INSTITUTO DE ECONOMÍA

TESIS de DOCTORADO

2010

A Generalized Three-Step Panel Data Estimator (G3SPD)

Diego Avanzini.

A Generalized Three-Step Panel Data Estimator (G3SPD)

por

Diego Avanzini

Contador Público Nacional, Universidad Nacional de San Luis, 1998
Licenciado en Administración, Universidad Nacional de San Luis, 2002
Magíster en Economía, Mención Políticas Sociales y Económicas (th.), Universidad Nacional de Tucumán, 2001

Esta tesis se presenta como requerimiento parcial
para optar al grado de

Doctor en Economía

Instituto de Economía
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

Comité:
Raimundo Soto (profesor guía)
Rodrigo Cerda
Francisco Gallego
Rómulo Chumacero

2010

La tesis de Diego Avanzini fue aprobada:

Raimundo Soto (profesor guía)

Rodrigo Cerda

Francisco Gallego

Rómulo Chumacero

Pontificia Universidad Católica de Chile

13 de enero de 2010

Abstract

A Generalized Three-Step Panel Data Estimator (G3SPD)

by

Diego Avanzini

Doctor en Economía, Pontificia Universidad Católica de Chile

Raimundo Soto (Chair)

I develop a panel data estimator, the Generalized Three-Step Panel Data (G3SPD) estimator, allowing for a consistent (or at least less biased) estimation of models with endogenous regressors and the inclusion of dimension-invariant variables. Standard estimators fail to account for one or both problems. Simulated experiments show it substantially outperforms classical estimation techniques in terms of bias.

I also analyze the behavior of the standard errors and how they can affect inference. I show that the standard errors obtained by standard methods are downward biased and propose an adjusted variance-covariance matrix for the 3rd Step of the G3SPD estimator. Using Monte Carlo simulations I find that the adjusted standard errors outperforms popular panel data estimators across sample sizes and degrees of heterogeneity of the unobservable effects, irrespective of assuming fixed- or random-effects designs.

Finally, I apply the G3SPD estimator to three classical panel data applications (namely, a gravity model of bilateral trade, economics of crime, and analysis of wage determinants and returns to schooling). I find that alternative estimators such as error-components or Hausman-Taylor-type estimators may be highly biased both in terms of coefficients estimation and fitting the sample. Additional Monte Carlo experiments applied to the estimation of the wage equation show that the G3SPD estimator is able to control the bias generated by omitted variables that are correlated with the included regressors (e.g. ability).

Raimundo Soto
Chair

Resumen

A Generalized Three-Step Panel Data Estimator (G3SPD)

por

Diego Avanzini

Doctor en Economía, Pontificia Universidad Católica de Chile

Raimundo Soto (profesor guía)

Desarrollo un estimador de datos de panel, el estimador generalizado de datos de panel en tres pasos (Generalized Three-Step Panel Data estimator, o G3SPD estimator), que permite la estimación consistente (o al menos, con menor sesgo) de modelos con regresores endógenos y la inclusión de variables invariantes en alguna de las dimensiones del panel. Los estimadores estándar no suelen contemplar alguno o ambos problemas. Los resultados de experimentos simulados muestran que este estimador se desempeña substancialmente mejor que las técnicas de estimación clásicas en términos de sesgo.

También analizo el comportamiento de los errores estándar y cómo éstos pueden afectar la inferencia. Allí muestro que los errores estándar obtenidos por métodos estándar se encuentran sesgados a la baja, y propongo una matriz de varianza-covarianza ajustada para el tercer paso del estimador G3SPD. Usando simulaciones de Monte Carlo, encuentro que los errores estándar ajustados se desempeñan mejor que los obtenidos por estimadores populares para datos de panel, para distintos tamaños de muestra y grados de heterogenei-

dad de los efectos no-observables, sin importar si se asume que el diseño corresponde a efectos fijos o aleatorios.

Finalmente, aplico el estimador G3SPD a tres problemas clásicos de datos de panel (particularmente, un modelo de gravedad de comercio bilateral, un modelo de economía del crimen, y el análisis de los determinantes del salario y los retornos a los años de escolaridad). Encuentro que los estimadores alternativos como el de componente de errores o los estimadores del tipo de Hausman y Taylor, pueden estar altamente sesgados tanto en términos de la estimación de los coeficientes como en el ajuste dentro de la muestra. Un conjunto adicional de experimentos de Monte Carlo aplicados a la estimación de la ecuación de salario muestra que el estimador G3SPD es capaz de controlar el sesgo generado por variables omitidas que están correlacionadas con los regresores incluidos (por ejemplo, habilidad).

Raimundo Soto
Profesor Guía

To Pancín and my family.

A Pancín y mi familia.

Contents

Acknowledgments

I wish to express my gratitude to my adviser, Raimundo Soto for all his helpful comments and support during the development of this dissertation. The experience he shared with me has been of great importance for my professional career. I also want to thank the members of my committee, Rodrigo Cerda and Francisco Gallego for their time and wise advice, as well as Rómulo Chumacero for his time and goodwill.

I want to thank my beloved wife Pamela who gave me energy, support, and motives to finish this enterprise from the very beginning. I am also grateful to my family, specially my parents Tito and Lillyana (R.I.P.) for always encouraging me to face new challenges, and my brother Elbio that patiently heard my explanations over and over again.

I want to thank my colleagues, friends, and classmates for all their time, concern and advice. Among them I would like to mention Carlos Salomón who kindly heard me and "helped me to think", Carmen Garcés for making my life easier during these years at the PUC, José Miguel Sánchez for giving me this opportunity in the first place, and those who anonymously supported me in hard times.

Finally, I acknowledge the financial support of the PUC (grant to pursue a doctorate degree, between Aug. 2004 and Feb. 2005, grant VRAID between Apr. and Jun., 2009) and CONICYT Chile (grant to pursue a doctorate degree, between Mar. 2005 and Jun. 2008).

Part I

A G3SPD Estimator

Chapter 1

A Generalized Three-Step Panel Data Estimator: Bias Issues

1.1 Introduction

The use of panel data has extended to almost every area of economics, motivated by both data availability and because unobservable features can be captured by cross-section and time-series estimations. Of particular interest is the application of panel data analysis to policy impact evaluation (Cornwell and Rupert, 1988; Baltagi and Khanti-Akom, 1990; McClellan and Staiger, 1999), regulation (Knight, 2002; Hazlett and Muñoz, 2009a, 2009b), international trade (Krishnakumar, 2002; Egger and Pfaffermayr, 2004; Serlenga and Shin, 2007), investment (Davies et al., 2005), stock returns (Bulkley et al., 2004), political economy (Iversen and Cusack, 2000; Green et al., 2001), marketing actions (Lam et al., 2001), firm efficiency (Sickles, 2005), criminal behavior (Cornwell and Trumbull, 1994; Baltagi, 2006), etc. These applications, to name just a few, focus on estimating the value of the parameters of the studied model in order to guide decision-making, emphasizing the need for consistency and forecasting power. However, the selection of an adequate estimation method relies on the set of assumptions concerning the endogeneity of regressors and the randomization of unobservable characteristics of the sample.

The selection of a suitable approach to panel data estimation ranges from "pooled"

OLS, to "fixed effects", to Hausman–Taylor–type procedures, to (random effects) error-components. The literature provides analyses of their advantages and disadvantages.[1] However, the presence of endogenous regressors (either in single-equation or system estimation) and the inclusion of dimension-invariant variables[2] pose special challenges to these panel data methods, which we address in this study.

We introduce a Generalized Three-Step Panel Data estimator (G3SPD) to estimate single- and simultaneous-equations models (both equation-by-equation and system estimation) with panel data when the "true model" is characterized by (i) a set of regressors including dimension-invariant variables, and (ii) some (or all) of the regressors are correlated with the unobservable effects.

The G3SPD estimator offers many advantages: (i) it deals with both endogenous dimension-varying and dimension-invariant variables in a simple and elegant way; (ii) it outperforms available alternative estimators in terms of bias; (iii) it yields a full set of estimates that can be used directly in forecasting, model evaluation, and inference; (iv) it yields a set of estimates for the unobservable effects, dubbed *pseudo-effects*; and (v) it can be applied to both fixed-effects and random-effects models. Moreover, its estimation procedure is simple, apparent, and quick, and can be easily implemented using standard econometric packages with minimum adjustments.

The estimator is based on a simple intuition: if the source of bias in a panel data regression is the correlation between the regressors and the unobservable effects, introducing

[1] This discussion and the description of the estimation techniques can be found in many articles and books, e.g. Hausman (1978), Mundlak (1978), Hausman and Taylor (1981), Hsiao (2002), Wooldridge (2002), Baltagi (2005), Krishnakumar (2006).

[2] Here we call "dimension" to each of the "ways" a panel can have. Usually a panel data has two ways or dimensions: individuals and time, but more dimensions can be analyzed (e.g. see Abowd et al., 2002; Horrace and Schnier, 2008).

an estimate of the latter in the model may contribute to eliminate, or at least attenuate, such bias. Consequently, obtaining a set of estimates of the unobservable effects is a key feature of this method that requires a multi-step regression procedure.[3] Briefly, the method comprises three estimation steps: the first and second steps are aimed at obtaininig a set of estimates of the coefficients for both the dimension-varying and the dimension-invariant covariates. These estimates are used to obtain the *pseudo-effects*, which are added to the dimension-varying and dimension-invariant regressors in a pooled regression in the third step. The G3SPD estimator yields less biased estimates of the coefficients; outcomes from a set of Monte Carlo simulations for different sample sizes and degrees of endogeneity are presented in order to compare the proposed estimator with classical panel data methods. Efficiency considerations regarding the G3SPD estimator are developed in a companion paper (see Avanzini, 2010, and chapter 2, below), suggesting that it yields more reliable standard errors estimates when compared with other available methods.

The rest of the chapter is organized as follows: the next section introduces three classical examples that present the problems we are interested in. The third section reviews the theoretical aspects of panel data estimation as well as the problems that may bias estimates. Section 1.4 develops the G3SPD and its relationship with classical models as well as some statistical features. Simulation outcomes and performance comparisons for both fixed- and random-effects simultaneous equations models are presented in Section 1.5. In the sixth section we apply the G3SPD estimator to the examples introduced in Section 1.2

[3]To our knowledge, Plümper and Troeger (2007) are the first to develop a single-equation panel data estimation procedure following this line of reasoning, which they call Fixed-Effects Vector Decomposition. However, their procedure does not take into account the potential correlation between regressors and unobservable components, and may yield highly biased results for both the coefficients and the standard errors when the regressors are correlated with the unobservable effects.

and analyze the new results. We also conduct an empirical exercise to show how the G3SPD estimator deal with omitted variables correlated with the included regressors. Section 1.7 concludes with some remarks. The appendices include proofs of the theorems and a section where we show how to set matrices in order to obtain classical estimation methods for panel data as special cases of the estimators in Section 1.4.

1.2 Three Classical Examples

The increasing availability of panel datasets has popularized the application of panel data methods to very diverse areas of economics such as labor and education, macroeconomics, trade, finance, marketing, health care and social security, demand-supply estimation, programs and public policy evaluation and regulation. A reason to use panel data is to account for the presence of unobservable information common to some or all units in the sample. This unobservable information, usually called "effects" in the panel data jargon, may bias estimates when correlated with the regressors.

To illustrate the questions researcher may be interested in answering when facing these problems, we revisit three widely-known examples: (i) gravity models of international trade flows, (ii) economics of crime, and (iii) wage determinants and returns to schooling. These examples appear in textbooks and are used to show the performance and validity of alternative estimation methods.[4] Another advantage if that the corresponding datasets are available from the J*ournal of Applied Econometrics* website, where the related articles were published.

[4] For example, Baltagi (2005, 2009) use these examples to illustrate the use of error-components and Hausman-Taylor-type estimators with different sets of instruments.

The first example refers to a gravity model of international trade flows which is widely used as a baseline model for estimating the impact of a variety of policy issues such as regional trading groups, currency unions and trade distortions. Serlenga and Shin (2007) estimate a gravity equation of bilateral trade flows among 15 European countries over the period 1960-2001 using a generalized Hausman-Taylor estimation methodology. They are particularly interested in evaluating the effect of time-invariant characteristics that are usually disregarded by the fixed-effects estimators used in this context. They regress the $log(volume\ of\ bilateral\ trade)$ on time-varying (real exchange rate, GDP, GDP per capita, similarity, shared currency, EU membership), time-invariant (distance, shared border, shared language), and time-specific unobservable effects. Among their findings is that the impact of country-specific variables (time-invariant) can be recovered using the Hausman-Taylor approach in addition to allowing such variables to be endogenous. In particular, they argue that shared language is a proxy for cultural, historical and linguistic proximity, and this in turn is highly correlated with country-specific effects (see Baltagi, 2009, ch. 7).

Our second example is taken from Cornwell and Trumbull (1994), and a revisit by Baltagi (2006). They are interested in estimating an economic model of crime using panel data. The empirical model –which follows Becker (1968), Ehrlich (1973), and others– relates $log(county\ crime\ rate)$ to dimension-varying (probability of arrest, number of police per capita, probability of conviction, sanction severity, sectoral wages, population characteristics), dimension-invariant (location, percent of minority), and time-specific effects. The reason for using panel data instead of cross-sectional information, as many other studies do,

"[...] is an inability to control for unobserved heterogeneity in the unit of observation. The use of 2SLS and 3SLS in these studies does not treat this problem. Neglected heterogeneity also may be correlated with the instrumental variables used to compute the 2SLS and 3SLS estimates. With panel data we can account for unobservable county characteristics by conditioning on county effects in estimation [...]" (Cornwell and Trumbull, 1994, pp. 361). Additionally, the probability of arrest and the number of police per capita are endogenous variables that must be instrumented to avoid simultaneity bias. The main problem with the empirical results is that after controlling for both sources of endogeneity, namely conventional simultaneity and neglected heteregeneity, none of the determinants appears to be significant. Baltagi (2006), in an attempt to overcome this problem, proposes the use of 2SLS error-components estimators supporting its validity with an alternative Hausman test (Baltagi, 2004).

The final example deals with the estimation of wage determinants, and particularly, the impact of schooling. The usual approach is to use a reduced-form equation inspired by Mincer (1974), controlling for the potential correlation between individual ability and education using different strategies (e.g., see Griliches, 1977; Lillard and Willis, 1978; Hausman and Taylor, 1981; Chowdhury and Nickell, 1985, Cornwell and Rupert, 1988; Baltagi and Khanti-Akom, 1990; among others). This framework is particularly appealing because it captures two distinct economic concepts: (i) a pricing equation for productive attributes such as schooling and work experience, and (ii) the rate of return to schooling which can be used to determine optimality of human capital investment (see Heckman et al., 2003). We focus on the study by Cornwell and Rupert (1988), who run regressions of the $log(wage_{it})$

on several time-varying (experience, occupation, unionization, location) and time-invariant regressors (gender, ethnics, years of schooling, marital status). In this context, years of schooling is the variable of interest charaterized by time-invariability and potential correlation with individual unobservable characteristics such as ability and ambition. Isolating the effect of years of schooling on wages serves to analyze the two economic implications stated before. Using a panel dataset from the Panel Study of Income Dynamics (PSID), Cornwell and Rupert evaluate the performance of different estimators including Hausman-Taylor-type with different sets of instruments. Baltagi and Khanti-Akom (1990) replicate Cornwell and Rupert's results but assuming a different correlation structure which in turn lead to different sets of instruments.

After developing the theoretical framework of the G3SPD and showing how it performs in terms of bias via Monte Carlo simulations, we will return to these examples to evaluate the results using G3SPD.

1.3 Theoretical Issues

The former examples have an important problem in common: endogeneity. To put this problem in formal terms, consider the following generic single-equation model for a bi-dimensional panel data:

$$y_{it} = X_{it}\beta + Z_i\gamma + Z_t\tau + [\varphi + \mu_i + \lambda_t + \varepsilon_{it}] \tag{1.1}$$

where y_{it} is a dimension-varying "dependent" variable, and individuals and time periods are identified by $i = 1, \ldots, N$ and $t = 1, \ldots, T$, respectively. The right-hand side includes a set of observable covariates and unobservable (or neglected) information which forms the usual residual of the model. The observable information includes dimension-varying covariates – X_{it}, which varies in both dimensions – and dimension-invariant regressors – Z_i and Z_t, which only vary in one of the dimensions of the panel. Assume that the unobservable information can be split into four components: an individual-specific effect μ_i (i.e. an effect that remains constant for each individual through all time periods), a time-specific effect λ_t (that does the proper for time periods), a constant term φ (i.e. it does not vary in all panel dimensions), and an stochastic error term, ε_{it}. The estimable model is:

$$y_{it} = X_{it}\beta + Z_i\gamma + Z_t\tau + \eta_{it} \tag{1.2}$$

where $\eta_{it} = \varphi + \mu_i + \lambda_t + \varepsilon_{it}$ is the residual part of the regression. When some regressors and η_{it} are correlated, the estimation of this model may yield biased and inconsistent estimates if not accounting for that correlation. Two possible sources of bias are (i) the correlation between the regressors and either μ or λ, and (ii) the correlation between the regressors and ε.

Assume now that equation [1.1] is one of the G equations of the following (reduced-form) system of simultaneous equations:

$$\begin{aligned} y_{1it} &= Y_{1it}\pi_1 + X_{1it}\beta_1 + Z_{1i}\gamma_1 + Z_{1t}\tau_1 + [\varphi_1 + \mu_{1i} + \lambda_{1t} + \varepsilon_{1it}] \\ &\vdots \\ y_{Git} &= Y_{Git}\pi_G + X_{Git}\beta_G + Z_{Gi}\gamma_G + Z_{Gt}\tau_G + [\varphi_G + \mu_{Gi} + \lambda_{Gt} + \varepsilon_{Git}] \end{aligned} \tag{1.3}$$

where Y_{git} is a set of endogenous variables correlated with both the unobservable effects and the stochastic error term, each one appearing as the left-hand-side-variable in the remaining equations of the system. Thus the latter regressors add a simultaneity bias to the problem of estimation. In summary, the system of equations [1.3] presents four sets of variables that can be correlated with either components of η, potentially biasing the estimates of the coefficients if not accounting for endogeneity and/or simultaneity.

1.3.1 Endogeneity bias

Consider the case when the regressors in equation [1.1] are only correlated with the unobservable effects μ or λ. The validity of the estimators depends on the assumptions the researcher makes about the data generating process and the type of inference required – i.e. distinguishing between inference based on "conditional" versus "marginal or unconditional" distributions of the events. Intuitively, when the unobservable effects in the sample contain information of interest to the researcher, *fixed effects* should be assumed and inference should be made *conditional on such effects*[5]. When the effects contain general information attributable to the population, then the effects should be considered *randomly distributed across the population*, i.e. inference should be carried out based on the *unconditional (marginal) distribution*.

In general, if we assume that the effects are *fixed*, three estimation methods apply: the fixed-effects (FE) or within-groups estimator, the least–squares dummy variables (LSDV) approach, and the estimator based on first-differences (FD) of equation [1.1]. The three approaches present problems when dealing with dimension-invariant variables: the FE and FD estimators wipe out the fixed effects together with other dimension-invariant components eliminating the bias problem originated by the correlation of the regressors with

[5] The fixed effects can also be thought as follows: when the differences in behavior between observations cannot be attributed to a pure chance mechanism, then it is more appropiate to view each observation as drawn from heterogeneous populations and the specific-dimension effect as representing the fundamental differences among the heterogeneous populations (Hsiao, 2002, Ch. 3). This interpretation allows for the fixed effects to be correlated with the regressors (as in Cornwell and Trumbull, 1994; or Baltagi, 2006), contrary to Krishnakumar (2006) that assumes that fixed effects are never correlated with the regressors.

the unobservable effects, at the cost of neglecting the estimation of the coefficients of the dimension-invariant regressors[6]. On the other hand, the LSDV approach introduces a set of dummy variables to represent the effects but they become perfectly correlated with the dimension-invariant regressors, making it impossible to obtain estimates of γ or τ.[7] Thus, assuming fixed effects prevents us to obtain consistent estimates of the parameters except for β unless we use a two-step procedure as proposed by Hsiao (2002, Section 3.6) or Krishnakumar (2006). The two-step procedure uses the within residuals obtained in the first step as the dependent variable in a second step reduced-"between-groups" regression that yields a set of estimates of γ and τ. As shown below, those estimates may be highly biased if the regressors (specially the dimension-invariant ones) are correlated with the effects.

Alternatively, when the effects are assumed to be *random*, error-components (EC) estimators may be a suitable choice for estimating the equation. Although we would be able to obtain estimates of β, γ, τ, all estimators will be biased given that the EC estimators do not eliminate the effects from the regression (they remain in the model mixed with the stochastic error term). This endogeneity bias can be solved, for example, using two-stage least squares (2SLS) methods. 2SLS requires the identification of a suitable set of instruments to correct the bias problem using an Aitken-type estimator. And this is exactly what the Hausman-Taylor-type procedures (HT) do[8]. In fact, the HT procedure consists of finding an adequate set of instruments[9] and estimating the regression by 2SLS EC. When

[6]First-differences and within-groups transformation approaches are textbook methods to obtain the estimates (see Hsiao, 2002; Wooldridge, 2002; Baltagi, 2005). A discussion about pros and cons of using these methods can be found in Wooldridge (2002, Ch. 10).

[7]This method poses some additional problems for the user such as large dummy variables matrixes that must be inverted, an accelerated loss of degrees of freedom, and the so-called "incidental parameter problem" (see Neyman and Scott, 1948; Lancaster, 2000).

[8]See Hausman and Taylor (1981), Amemiya and MaCurdy (1986), Cornwell and Rupert (1988), Breusch et al. (1989), Baltagi and Khanti-Akom (1990), Baltagi et al. (2003), among others.

[9]Hausman and Taylor (1981), Amemiya and MaCurdy (1986), Breusch et al. (1989), Balestra and

all regressors are correlated with the effects, the 2SLS EC estimator coincides with the FE estimator, though conceptually they are not the same[10]. Hence, obtaining consistent estimates of β, γ, τ relies on the availability and quality of instruments.

Finally, note that whenever the "true model" is one of fixed effects, the EC and HT estimators will yield biased and inconsistent estimates of the coefficients. The fixed effects approach is still viable in this case, i.e. it remains unbiased and consistent, though not necessarily efficient (see Baltagi, 2005, Ch. 2).

1.3.2 Simultaneity bias

The previously discussed methods deal only with the problem of the correlation between the regressors and the effects. But the correlation between the regressors and the stochastic error term needs a different approach such as using instrumental variables. To see this, consider the system of simultaneous equations in [1.3], and assume that neither the X's nor the Z's are correlated with $\eta = \varphi + \mu + \lambda + \varepsilon$. Note that if the corresponding structural model has a stochastic error term and individual and/or time effects in every structural equation, the unobservable effects – μ and λ – and the stochastic error terms – ε – of each reduced-form equation will be linear combinations of one another. Therefore, the solution for every endogenous variables Y_g will in general involve every structural error and also the individual- and time-specific effects from every structural equation. This means that all

Varadharajan-Krishnakumar (1987), Wyhowski (1994), to mention only a few, suggest different sets of instruments that can be recovered from the same panel, which largely reduce the effort associated with finding instrumental variables.

[10]Both estimators are sometimes confused with each other, specially based on the claims by Mundlak (1978). Nevertheless, FE and RE estimators (even in those cases where all regressors are correlated with the effects) rely on different assumptions, as previously stated. For a discussion on this, see Krishnakumar (2006).

endogenous variables in the reduced-form system are correlated with both the stochastic error term and the effects (see Breusch et al., 1989; Cornwell et al., 1992). Estimators that do not account for this twofold correlation may be affected by simultaneity bias.

Although methods such as three-stage least squares (3SLS) EC may be appropriate to avoid the bias due to the correlation between Y and ε, they are not aimed to control the remaining bias due to the correlation with the unobservable effects. In this case, it is necessary to explicitly account for this source of endogeneity, for example, using a within-groups (WI) estimator[11].

1.4 The G3SPD Estimator

The issues discussed in the previous section give rise to the question of whether it is possible to find an estimator that can account for all these problems and still be consistent (and ideally efficient). The Generalized Three-Step Panel Data estimator addresses these problems on the basis of an apparent reasoning: if standard panel data estimators may be biased due to the correlation between the regressors and the unobservable effects, then we could include an estimator of such effects and get back to the standard approaches to obtain a more reliable set of estimators.

Although the G3SPD estimator can be easily applied to individual- and time-specific effects, as well as to n-way unbalanced panel data, for exposition purposes we

[11]Notice that in the context of simultaneous equations with panel data, the within-groups estimator is treated as a part of the EC estimator given that the latter can be obtained as a weighted sum of a within-groups estimator and a between-groups estimator (see Baltagi, 1981b, 1984, 2005). Consistency and efficiency of EC estimators for simultaneous equations with panel data and random effects are addressed in Baltagi (1981b). Cornwell et al. (1992) do likewise for the within-groups estimator obtaining closed forms for its unconditional and conditional likelihood functions, but without tackling the efficiency issue. They also propose a generalized approach to simultaneous equations with panel data that comprises the estimation of within-groups transformed systems as a special case.

concentrate on balanced panel data configurations containing individual-specific effects. In this context, the inclusion of time-invariant variables pose a problem given that classical fixed-effects estimators wipe out those variables together with the unobservable individual effects, and OLS and EC estimators are biased and inconsistent, as discussed. Finally, HT estimators may be unable to account for the time-invariant regressors (or even may not exist) depending on the correlation between the regressors and the effects.[12]

1.4.1 General Setup

Consider the (reduced-form) system of simultaneous equations[13]:

$$y = Y\pi + X\beta + Z\gamma + [Z_{\varphi}\varphi + Z_{\mu}\mu + \varepsilon] \tag{1.4}$$

This system contains G equations, and each equation has N observations concerning households, firms, countries, etc., throughout T periods of time. Hereafter, we use subscript $g = 1, \ldots, G$ to identify cases in which we refer to a specific equation, whereas no subscript means the whole system. Z_{μ} is a selector matrix representing a set of *dummy* variables for each individual that arrange the μ_g allocating one for each individual for all time periods, and it is obtained as $Z_{\mu_g} = I_N \otimes \iota_T$, whereas for the whole system we have $Z_{\mu} = I_G \otimes Z_{\mu_g}$. Z_{φ} is a selector matrix representing a set of *dummy* variables that allocates φ_g in the corresponding equation, and is obtained as $Z_{\varphi} = I_G \otimes \iota_{NT}$.

[12] See Appendix 1.F for the rank conditions of different HT estimators to exist.

[13] For the sake of brevity, we explain the G3SPD estimator in the context of a system of simultaneous equations estimation given that it encompasses single-equation estimation.

The system comprises four sets of observable variables:

$$
\begin{aligned}
y &= \begin{bmatrix} y_1 \\ \vdots \\ y_G \end{bmatrix} \quad X = \begin{bmatrix} \left[X_1^{(1)}, X_1^{(2)}\right] & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left[X_G^{(1)}, X_G^{(2)}\right] \end{bmatrix} \\
Y &= \begin{bmatrix} Y_1 \\ \vdots \\ Y_G \end{bmatrix} \quad Z = \begin{bmatrix} \left[Z_1^{(1)}, Z_1^{(2)}\right] & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left[Z_G^{(1)}, Z_G^{(2)}\right] \end{bmatrix}
\end{aligned}
\tag{1.5}
$$

where y is a $GNT \times 1$ vector containing the set of endogenous left-hand side variables, and Y is a $GNT \times (G-1)$ matrix containing the $(G-1)$ set of remaining endogenous variables appearing on the right-hand side of each equation. Hence y_g is a $NT \times 1$ vector, and Y_g is a $NT \times (G-1)$ matrix. Aditionally, the right-hand side set of variables may include K dimension-varying covariates – the X's – and J dimension-invariant covariates – the Z's. These are block-diagonal matrices of size $GNT \times GK$ and $GNT \times GJ$, respectively.[14]

Regarding exogeneity, we adopt the Cornwell et al. (1992) classification, which separates the regressors in three categories: *doubly-exogenous* (covariates that are uncorrelated with both the stochastic term and the effects, identified with superscript (1), i.e. $X_g^{(1)}, Z_g^{(1)}$ $\forall g = 1, \ldots, G$), *singly-exogenous* (covariates that are uncorrelated with the stochastic term but correlated with the effects, identified with superscript (2), i.e. $X_g^{(2)}, Z_g^{(2)}$ $\forall g = 1, \ldots, G$), and *endogenous* covariates (those correlated with both the stochastic error and the effects, i.e. Y_g $\forall g = 1, \ldots, G$). We assume that the correlation between the regressors and the unobservable effects only involves their means.

[14]Note that the Z's do not vary across time periods remaining constant for each individual, which will imply perfect collinearity with the individual effects, as defined afterwards.

The corresponding set of coefficients in system [1.4] are block-diagonal arrangements of parameters:

$$\pi = \begin{bmatrix} \pi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \pi_G \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \beta_G \end{bmatrix} \quad \gamma = \begin{bmatrix} \gamma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma_G \end{bmatrix}$$

where π is a $G(G-1) \times G$ matrix, β is a $GK \times G$ matrix, and γ is a $GJ \times G$ matrix. This means that each π_g, β_g, γ_g is a column-vector with a coefficient for each covariate[15].

Regarding the unobservable information of the system, assume that the residual of each equation has three components: a constant φ_g (which can be thought as corresponding to the error mean), a set of individual effects μ_g (corresponding to a constant term for each individual), and a purely stochastic component ε_g. For the whole system of equations, the heteroskedastic residual

$$\eta = Z_\varphi \varphi + Z_\mu \mu + \varepsilon$$

has its components arranged in the following way:

$$Z_\varphi \varphi = \begin{bmatrix} \iota_{NT}\varphi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \iota_{NT}\varphi_G \end{bmatrix} \quad Z_\mu \mu = \begin{bmatrix} Z_{\mu_g}\mu_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Z_{\mu_g}\mu_G \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varepsilon_G \end{bmatrix}$$

where $\varphi, \mu, \varepsilon$ are $GNT \times G$ matrices each one. In what follows, ι_h represents a column-vector of ones of order h.

The stochastic error term distributes $\varepsilon_g \sim iid(0, \sigma^2_{\varepsilon_g})$, i.e. assuming homoskedasticity, the disturbances of the i^{th} individual at the t^{th} period, $(\varepsilon_{1it}, \ldots, \varepsilon_{Git})'$ distribute

[15] For exposition purposes, assume that all equations share the same number of dimension-varying and dimension-invariant regressors.

$N(0, \Sigma_\varepsilon)$, with Σ_ε being a $G \times G$ covariance matrix: $\Sigma_\varepsilon = E([\varepsilon_{1it}, \ldots, \varepsilon_{Git}]' [\varepsilon_{1it}, \ldots, \varepsilon_{Git}])$. Hence the stochastic error term of the system is $\varepsilon \sim iid\left(0, \Sigma_\varepsilon \otimes I_{NT}\right)$.

When fixed effects are assumed, the set of unobservable effects μ_g remains constant for each possible realization of y, while varying among individuals according to $\mu_g \sim iid(0, \sigma^2_{\mu_g})$.[16] This means that for the currently observed time-series for an individual $y_{gi} = (y_{gi1}, \ldots, y_{giT})$, the unobservable effect μ_{gi} is fixed. In this case, the stochastic part of η only corresponds to ε, and $\eta_g \sim iid\left(\varphi_g + \mu_g, \sigma^2_{\varepsilon_g}\right)$. For the whole system, $\eta \sim iid\left(Z_\varphi \varphi + Z_\mu \mu, \Sigma_\varepsilon \otimes I_{NT}\right)$.

Alternatively, when random effects are assumed, the set of unobservable effects μ_{gi} varies across the possible realizations of y : in this case $\mu_{gi} \sim iid(0, \sigma^2_{\mu_{gi}})$, which means that for each potential realization of y we have a corresponding set of random effects. We assume that $\sigma^2_{\mu_{gi}} = \sigma^2_{\mu_g}$ $\forall g = 1, \ldots, G$. Now, η_g distributes with mean φ_g and covariance matrix:

$$E\begin{pmatrix} \mu_g \\ \varepsilon_g \end{pmatrix} \begin{pmatrix} \mu'_h & \varepsilon'_h \end{pmatrix} = \begin{bmatrix} \sigma^2_{\mu_{gh}} I_N & 0 \\ 0 & \sigma^2_{\varepsilon_{gh}} I_{NT} \end{bmatrix} \qquad \forall g, h = 1, \ldots, G$$

It follows that for any pair of equations,

$$\Sigma_{gh} = E(\eta_g \eta'_h) = \sigma^2_{\mu_{gh}}(I_N \otimes J_T) + \sigma^2_{\varepsilon_{gh}} I_{NT}$$

while for the whole system,

$$\Sigma = E(\eta\eta') = \Sigma_\mu \otimes (I_N \otimes J_T) + \Sigma_\varepsilon \otimes I_{NT} \tag{1.6}$$

where $J_T = \iota_T \iota'_T$, and Σ_μ and Σ_ε are both $G \times G$ matrices. Using Wansbeek and Kapteyn's

[16] This configuration follows from the interpretation of the fixed effects by Hsiao (2002, Ch. 3). This also justifies the fact that even when the effects are considered fixed, they can be correlated with the regressors, as previously stated.

(1982a, 1982b, 1983) results, we can rewrite [1.6] as:[17]

$$\begin{aligned}\Sigma &= (T\Sigma_\mu + \Sigma_\varepsilon) \otimes \left(I_N \otimes \overline{J_T}\right) + \Sigma_\varepsilon \otimes (I_N \otimes E_T) \\ &= \Sigma_1 \otimes P_{\mu_g} + \Sigma_\varepsilon \otimes Q_{\mu_g}\end{aligned} \tag{1.7}$$

where $\overline{J_T} = (1/T)J_T$, $E_T = I_T - \overline{J_T}$, $\Sigma_1 = T\Sigma_\mu + \Sigma_\varepsilon$, and P_{μ_g} and Q_{μ_g} are two useful matrices obtained from the generalized projection matrix $P_A = A(A'A)^{-1}A'$ and its complement $Q_A = I_A - P_A$, respectively, after replacing A with Z_{μ_g}. Here I_A is an identity matrix with its order determined by the number of rows in A. Notice that P_{μ_g} is the *between-groups transformation* that returns the individual means from a panel dataset, and Q_{μ_g} is the *fixed effects or within-groups transformation* which returns deviations from the individual means, for a given equation g.[18] For the whole system, $P_\mu = I_G \otimes P_{\mu_g}$ and $Q_\mu = I_{GNT} - P_\mu$.

1.4.2 The G3SPD Procedure

The G3SPD procedure estimates the following system of simultaneous equations:

$$y = Y\pi + X\beta + Z\gamma + Z_\varphi\varphi + Z_\mu\mu + \varepsilon \tag{1.8}$$

where feasibility of estimation is solved replacing μ with an estimator. Obtaining such estimator of the unobservable effects is a key feature of G3SPD.[19] Inspired in two-step regression procedures, the first and second step of the G3SPD estimator are aimed to obtain adequate estimates that are used to generate the pseudo-effects, i.e. the set of estimated

[17]Baltagi (2005, Ch. 6) pointed out that the result in [1.7] corresponds to the spectral decomposition of Σ derived in Baltagi (1980), and which implies that $\Sigma^r = \Sigma_1^r \otimes P_{\mu_g} + \Sigma_\varepsilon^r \otimes Q_{\mu_g}$, where r is an arbitrary scalar. This result facilitates obtaining the covariance matrixes.

[18]Note that these projection matrixes are symmetric, idempotent, of rank $rank(P_{\mu_g}) = tr(P_{\mu_g}) = N$ and $rank(Q_{\mu_g}) = tr(Q_{\mu_g}) = N(T-1)$, orthogonal (i.e. $P_{\mu_g}Q_{\mu_g} = 0$), and sum to identity matrix (i.e. $P_{\mu_g} + Q_{\mu_g} = I_{NT}$).

[19]Notice that FE, EC, and HT estimators do not produce a set of estimates for these effects: obtaining them may require the use of two-step procedures which is the basis of the G3SPD estimator.

unobservable effects, $\hat{\mu}$. The third step of the G3SPD estimator is the central step and is intended to obtain unbiased and consistent estimates for the whole set of coefficients in the system [1.8], including the pseudo-effects as a new regressor. This helps to avoid (or at least attenuate) biasedness and inconsistency of estimates due to the correlation of the regressors with the unobservable effects, μ.

Two-step procedures for estimating the unobservable effects have been previously proposed in the literature (e.g. see Hsiao, 2002, Ch. 3; Baltagi, 2005, Ch. 2; Krishnakumar, 2006). The problem with these procedures is that they fail to adequately estimate the coefficients of the dimension-invariant regressors. This motivates using the following three-steps procedure:

1^{st} Step: Estimate a within-groups-transformed regression. *Transform the model [1.8] by pre-multiplying it by Q_{μ_g} or Q_{μ} depending on the type of estimation (equation-by-equation or system estimation, respectively) and estimate a regression with the transformed model*[20]*. This will yield $\hat{\pi}_{wi}$ and $\hat{\beta}_{wi}$.*

Recall that using a within-groups transformation when estimating the model allows to obtain consistent estimators for π and β, at the cost of wiping out the unobservable effects together with the dimension-invariant variables and the constant term. However, the advantage of this approach is that any arising bias generated by the correlation between endogenous and singly-exogenous dimension-varying regressors and the unobservable effects is avoided. But this transformation is not enough to eliminate the correlation between the endogenous covariates Y_g and the stochastic error term ε_g, because their deviations from the

[20] See the following subsection for general estimators, and the Appendix 1.F for a discussion of several alternative methods.

individual means remain correlated, i.e. $Q_{\mu_g} Y_g$ is still correlated with $Q_{\mu_g} \varepsilon_g$. An adequate treatment for this simultaneity problem would require the use of instruments; the next subsection presents alternative estimation methods accounting for this.

Consistency of the fixed-effects (within-groups) estimator for the system of simultaneous equations when the effects are fixed follows from Theorem 1.1:

Theorem 1.1 *The maximum likelihood estimators of* π *and* β *in*

$$y = Y\pi + X\beta + Z\gamma + Z_\varphi \varphi + Z_\mu \mu + \varepsilon$$

are the same as the maximum likelihood estimators obtained from the system of equations after a within-groups transformation, i.e.

$$Q_\mu y = Q_\mu Y\pi + Q_\mu X\beta + Q_\mu \varepsilon$$

when the effects μ *are assumed fixed. Moreover, their covariance matrices coincide.*

Proof. See Appendix 1.A. ■

This result is also useful because estimating a within-groups-transformed regression avoids the *incidental parameters problem* for linear panel data models while its asymptotic properties remain intact[21]. Moreover, the within-groups estimator is also efficient when the "true model" is fixed effects, as shown in this theorem, and also stated in Cornwell et al. (1992).

[21]See Neyman and Scott (1948). For a review of the literature on the subject, see Lancaster (2000). It is worth mentioning that the incidental parameters problem remains in fixed-effects models for non-linear panel data models, biasing the estimates because it is not possible to completly eliminate the unobservables from the model. A long literature has developed around this problem: e.g. Heckman (1981), Vella and Verbeek (1999), Honoré (2002), Lancaster (2002), Greene (2004), Arellano and Hahn (2006), Fernandez-Val and Vella (2009), to name only a few.

On the other hand, consistency of the within-groups estimator for the system of simultaneous equations when the effects are random follows from Theorem 1.2:

Theorem 1.2 *When the effects μ are assumed to be random, the maximum likelihood estimators of π and β in*

$$y = Y\pi + X\beta + Z\gamma + Z_{\varphi}\varphi + (Z_{\mu}\mu + \varepsilon)$$

are the same as the maximum likelihood estimators obtained from the system of equations after a within-groups transformation, i.e.

$$Q_{\mu}y = Q_{\mu}Y\pi + Q_{\mu}X\beta + Q_{\mu}\varepsilon.$$

However, their covariance matrices differ.

Proof. See Appendix 1.B. ■

This theorem ensures that the estimators of π and β wil be consistent even when the individual-means of Y and X are correlated with the random effects. Nevertheless, they will not be necessarily efficient because the within-transformation is neglecting the between-groups variability, namely σ_{μ}^2.

2^{nd} Step: Estimate a between-groups-transformed regression. *Now, estimate a regression using only the means for each individual obtained from the model [1.8], i.e. premultiply it by P_{μ_g} if it is a single-equation or an equation-by-equation estimation, or by P_{μ} when considering the entire system of simultaneous equations. Using the general estimators proposed in the following sub-section or those developed in Appendix 1.F may ease the work. This will yield $\hat{\pi}_{be}$, $\hat{\beta}_{be}$, $\hat{\gamma}_{be}$, and $\hat{\varphi}_{be}$.*

Contrary to 1^{st} Step, consistency (and unbiasedness) of the between-groups estimators relies on the correlation between the regressors and the unobservable effects (assuming the stochastic residuals have zero mean, or that a constant is included in the model). When the effects (fixed or random) are correlated with the regressors[22], the between-groups estimator will be biased and inconsistent. At least three alternative approaches are available: first, plain (biased) between-groups estimator; second, two-step regressions; and third, instrumental variables. We analyze each one in turn.

The first approach takes the between-groups estimator as is. Recall that this estimator is biased and inconsistent when the regressors are correlated with the unobservable effects because the effects remain in the error after the between-groups transformation.[23] In the present context (linear panel data models), the biases for each set of coefficients are stated in the following Theorem:

Theorem 1.3 *Assume that $\hat{Y}$ stands for the instrumentalized counterpart of Y such that plim $\left(\frac{1}{NT}\right)\hat{Y}'\varepsilon = 0$. When all the regressors are correlated with the unobservable individual-specific effects, μ, the biases of the estimated coefficients of the between-transformed linear panel data model*

$$\overline{y} = \overline{\widehat{Y}}\pi + \overline{X}\beta + Z\gamma + Z_{\mu}\mu + \varepsilon$$

[22]When effects are assumed fixed, there are at least to alternative positions regarding correlation between the regressors and the effects: (i) fixed effects imply that they will never be correlated with the regressors (as explicitly stated in Krishnakumar, 2006); (ii) fixed effects are actually random variables containing important information about the sample, according to the explanations of the previous section, and they may be correlated with the regressors. The latter argument, which we assume, puts the accent on the importance of those unobservable effects to understand the behavior of the sample; moreover, it may motivate obtaining estimates of those effects.

[23]Notice that EC estimators are also biased and inconsistent. This results from the fact that EC estimators can be interpreted as a weighted sum of WI and BE estimators (with the weights reflecting the relative amount of information contained in the within and the between variation), and the latter are biased and inconsistent.

are given by:

$$
\begin{aligned}
Bias\left(\hat{\pi}_{be}\right) &= \frac{M_{\overline{\widehat{Y}}\,\overline{X}}}{M_{\overline{\widehat{Y}}}}\beta + \frac{M_{\overline{\widehat{Y}}Z}}{M_{\overline{\widehat{Y}}}}\gamma + \frac{M_{\overline{\widehat{Y}}\mu}}{M_{\overline{\widehat{Y}}}} \\
Bias\left(\hat{\beta}_{be}\right) &= \frac{M_{\overline{X}\,\overline{\widehat{Y}}}}{M_{\overline{X}}}\pi + \frac{M_{\overline{X}Z}}{M_{\overline{X}}}\gamma + \frac{M_{\overline{X}\mu}}{M_{\overline{X}}} \\
Bias\left(\hat{\gamma}_{be}\right) &= \frac{M_{Z\overline{\widehat{Y}}}}{M_{Z}}\pi + \frac{M_{Z\overline{X}}}{M_{Z}}\beta + \frac{M_{Z\mu}}{M_{Z}}
\end{aligned}
$$

Proof. See Appendix 1.C. ■

Note that the biases of $\hat{\pi}_{be}$, $\hat{\beta}_{be}$ and $\hat{\gamma}_{be}$ contain three components each one: two related with specific observed regressors accompanied by the respective coefficients, and one accounting for the correlation between the endogenous regressor and the unobservable effect. This implies that the unobservable effects bias the coefficients estimates through two general channels: (i) directly, due to the omitted μ in the regression, and (ii) indirectly, due to the correlation of the regressors between them.

The second approach is based on two-step regression and concentrates in obtaining unbiased and consistent estimators for γ and φ, as suggested by Hsiao (2002, Section 3.6), Baltagi (2005, Ch. 2, and previous editions of the book), and more recently, with elegant proofs, by Krishnakumar (2006). The first step obtains the within-groups estimators $\hat{\pi}_{wi}$ and $\hat{\beta}_{wi}$. With these estimators it is possible to generate a set of "within residuals", namely $u = y - Y\hat{\pi}_{wi} - X\hat{\beta}_{wi} = Z\gamma + \iota_{NT}\varphi + Z_{\mu}\mu + \nu$. The second step estimates a reduced-between-groups (R-BE) regression using the means across individuals of those within residuals as the left-hand-side variable to obtain the estimates of γ and φ. Thus, using this two-step procedure it is possible to obtain the estimators for all the coefficients in the model. However, it is easy to see that the estimators $\hat{\gamma}_{R-BE}$ and $\hat{\varphi}_{R-BE}$ are even more biased than the plain between-groups ones. See Appendix 1.E for a detailed explanation.

The third approach is based on instrumental variables estimators. Although the use of instrumental variables, such as those proposed in the HT-type procedures, may help to alleviate this inconsistency problem, in some cases it can be aggravated due to low quality instruments, as shown by Stock and Yogo (2004): the worse part of this situation is that, contrary to the researcher beliefs, the inconsistency problem persists (even worsening), and the cost of this potential bias may be high. An additional problem with this approach is that instruments may not always be available: when the number of doubly-exogenous regressors is not enough to meet the rank conditions, obtaining estimates for the dimension-invariant covariates is not possible, and the instrumental variables estimator becomes the within-groups estimator. See Appendix 1.F for rank conditions for different instrumental-variables estimators to exist.

We consider that our best set of estimators in this case comes from the (plain) between-groups estimator, and these estimators will be used to obtain the pseudo-effects in the next step.

3^{rd} Step: Estimate a pooled regression including the *pseudo-effects* as a new regressor. *In this step, we estimate the following pooled regression:*

$$y = Y\pi + X\beta + Z\gamma + \iota_{NT}\varphi + [Z_\mu \hat{\mu}]\phi + \varepsilon \qquad (1.9)$$

i.e., model [1.8] after substituting $Z_\mu \mu$ with $[Z_\mu \hat{\mu}]$, which we call the pseudo-effects. The pseudo-effects are a proxy for the "true" unobservable effects, and obtaining them is a critical issue of the G3SPD estimation procedure. This step obtains the correct set of residuals that are used in inference, confidence-intervals estimation, forecasting, etc. Without this step, we only have dispersed estimates for the set of parameters in [1.9].

The 1^{st} and 2^{nd} Steps of G3SPD obtains two sets of estimators for π and β, and one set for γ and φ. We know that only $\hat{\pi}_{wi}$ and $\hat{\beta}_{wi}$ are unbiased and consistent for sure (after adequately controlling for simultaneity of Y). The between-groups estimates are subject to potential bias and inconsistency. To obtain the pseudo-effects we introduce a little trick: our best estimates for π and β come from the within-groups regression (1^{st} Step), and the less biased estimates for γ and φ come from the between-groups regression (2^{nd} Step). Therefore, we use both sets of estimates to generate a proxy for the unobservable effects:

$$\hat{\mu} = P_\mu y - P_\mu(Y\hat{\pi}_{wi} + X\hat{\beta}_{wi}) - (Z\hat{\gamma}_{be} + Z_\varphi\hat{\varphi}_{be})$$

It is straightforward that $\hat{\mu} = \mu - Bias(Z\hat{\gamma}_{be} + Z_\varphi\hat{\varphi}_{be})$, i.e. the proxy for the unobservable effects, or the *pseudo-effects*, is a (potentially) biased representation of the "true" unobservable effects. In the worst scenario, when all the dimension-varying and dimension-invariant regressors are correlated with the effects, $\hat{\gamma}_{be}$ and $\hat{\varphi}_{be}$ are functions of $P_\mu Y\pi$ and $P_\mu X\beta$ too. Thus, the bias is a function $bias = Bias(P_\mu Y, P_\mu X, Z, \mu)$.

The form and size of this bias are very complicated to obtain and may require a large amount of information, even if we assume very simple forms for the correlation between the regressors and the effects (see Greene, 2002, Section 5.6). In fact, the bias depends on (i) the degree of heterogeneity of the unobservable effects relative to the variance of the stochastic error term, measured by $\rho = \frac{\sigma_\mu^2}{\sigma_\varepsilon^2}$, (ii) the correlation between the regressors and the effects, and (iii) the correlation of the regressors one another. However, the good news are that now that we are introducing this proxy (that is only a function of the regressors and the effects of the current model), we are attenuating the bias generated by the correlation between the regressors and the unobservables. Also notice that the coefficient

ϕ should be statistically equal to 1 whenever the pseudo-effects are consistent estimates of the unobservable ones, which can be interpreted as a natural test of the adequacy of the regressor $[Z_\mu \hat{\mu}]$.

The complexity of the problem at hand leads us to address the biasedness of these 3^{rd} Step estimators using Monte Carlo Simulations, as explained in the next section. Efficiency issues of the G3SPD estimator are developed in a companion paper (see Avanzini, 2010, and chapter 2, below).

1.4.3 Estimators for the G3SPD procedure

We now turn to the problem of finding a suitable set of estimators for each step and for both approaches, equation-by-equation and system estimation. In what follows, we propose general estimators that under general conditions are able to accomodate classical estimation methods for panel data. The proposed estimators belong to the GMM class inspired by Amemiya (1977) and are able to accommodate different sets of instruments for different equations when needed[24].

Before presenting the estimators, some comments are in order. First, GMM estimators behave very differently in small versus large samples. While asymptotically they have very desirable properties, their behavior in small samples is not clear. Many studies tackle the small sample behavior using Monte Carlo simulations (see Podivinsky, 1999, for a review of the literature), or bootstrap methods (e.g. Ramalho, 2006). Others have obtained closed forms for the bias of the GMM estimators based on stochastic expansions and

[24]Particular cases of the proposed estimators have been discussed in the literature, e.g. Schmidt (1988, 1990), Cornwell et al. (1992), Wooldridge (1996, 2002).

generalized empirical likelihood (e.g. Newey and Smith, 2004). However, in the present context is difficult to define small and large samples, and we adopt Monte Carlo simulations to study the behavior of the estimators.

The second problem relates the well-known bias of the two-step (and iterated) GMM estimators (Hansen, 1982). To avoid this problem, the proposed estimators below are single-step GMM estimators. As Windmeijer (2005) pointed out, single-step GMM estimators obtain unbiased and consistent estimators of the coefficients, though not necessarily efficient ones. See Avanzini (2009) for details on the efficiency problem.

Equation-by-equation estimators

First consider the case of equation-by-equation (and single-equation) estimation. A general expression for the estimator of a generic vector of parameters θ_g is given by:

$$\hat{\theta}_g = \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1} R_g' W_g' P_{[A_g,B_g]} W_g y_g \qquad (1.10)$$

where R_g is the set of regressors included in the model, W_g is a weighting matrix, and $P_{[A_g,B_g]}$ is the projection matrix of instruments. The set of instruments $[A_g, B_g]$ is capable of arranging instruments for both dimension-varying, A_g, and dimension-invariant endogenous regressors, B_g. The weighting matrix W_g allows for heteroskedasticity of various forms and model transformations.

A general expression for the (homoskedastic) covariance matrix of $\hat{\theta}_g$ is given by:

$$\hat{\Omega}_g = \hat{\sigma}_g^2 \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1}$$

where $\hat{\sigma}_g^2 = (\hat{\varepsilon}_g' \hat{\varepsilon}_g)/dof$ is a consistent estimator for σ_g^2, $\hat{\varepsilon}_g = W_g y_g - W_g R_g \hat{\theta}_g$ is a consistent estimator for ε_g, and dof stands for the degrees-of-freedom adjustment[25].

Depending on the choice of the weighting matrix and the set of instruments, this estimator can accomodate most of the classical panel-data estimators. For example, when instruments are not required, set $P_{[A_g,B_g]} = I_{NT}$. By changing the definition of W_g we can obtain several estimators: if $W_g = I_{NT}$, we get the traditional OLS estimator; if $W_g = \hat{\Omega}_{gg}^{-\frac{1}{2}}$ we have the GLS estimator with $\hat{\Omega}_{gg}$ being a consistent, not necessarily efficient, estimator of the covariance matrix. Moreover, setting $W_g = Q_{\mu_g}$ we get the within-groups estimator, while setting it to P_{μ_g} we obtain the between-groups estimator (BE). Extensions to account for special estimators of the covariance matrix such as the case of error-components models are straightforward. On the other hand, if instruments are required and available, we can obtain two-stage least-squares estimators in the spirit of Hausman and Taylor (1981), Amemiya and McCurdy (1986), and Breusch et al. (1989). See Appendix 1.F for details.

System estimators

Estimating a system of equations allows us to avoid the loss of information implicit in equation-by-equation estimation. In fact, efficiency gains in system of equations estimation are the result of non-neglecting cross-equation correlations of the disturbances. However it must be emphasized that the consequences of model misspecification when estimating the whole system are more harmful than when estimating it equation by equation: a misspecified equation contaminates the entire system and all parameters might be biased, while estimating it equation by equation only affects the misspecified equation keeping the other estimated equations unaffected.

[25]See Theorem 5.2 and Eqs. 5-25 to 5-27 in Wooldridge (2002), and Section 5.4 in Greene (2002).

An extension of the previous estimator to systems of equations is given by:

$$\hat{\theta} = \left[R'W'P^{*}_{[A,B]}WR\right]^{-1} R'W'P^{*}_{[A,B]}Wy \tag{1.11}$$

with

$$P^{*}_{A} = A(A'\check{\Sigma}A)^{-1}A'$$

where $\check{\Sigma}$ is a consistent estimator of the covariance matrix. This estimator is capable of arranging different sets of instruments in different equations, as well as different covariance estimators for the whole system of equations[26]. The set of instruments $[A, B]$ is a block-diagonal matrix with each block containing the proper set of instruments for each equation. The covariance matrix is:

$$\hat{\Omega}_g = \left[R'W'P^{**}_{[A,B]}WR\right]^{-1}$$

with

$$\begin{aligned} P^{**}_{A} &= A(A'\hat{\Sigma}A)^{-1}A' \\ \hat{\Sigma} &= \left[\frac{\tilde{\varepsilon}'W'W\tilde{\varepsilon}}{dof}\right] \otimes W \end{aligned}$$

The residuals, $\hat{\varepsilon} = Wy - WR\hat{\theta}$, are arranged in a $NT \times G$ matrix, i.e. $\tilde{\varepsilon} = [\hat{\varepsilon}_1, \hat{\varepsilon}_2, \ldots, \hat{\varepsilon}_G]$. Asymptotically, $\hat{\Sigma} = \check{\Sigma}$, though in small samples they can differ.

As stated, this general feasible estimator is capable of accomodating other classical panel data system estimators such as pooled 3SLS (set $W = I_{GNT}$, obtain an estimator of the covariance matrix from 2SLS: $\hat{\Sigma} \otimes I_{NT}$, and define the set of instruments as $[A, B] = I_G \otimes [X_g^{(1)}, X_g^{(2)}, X_g^{(3)}, Z_g^{(1)}, Z_g^{(2)}, Z_g^{(3)}, \iota_{NT}]$). The FE system estimator obtains if setting

[26] See Cornwell et al. (1992), Hsiao (2002), Wooldridge (2002), Baltagi (2005), among others, for a detailed explanation on the procedure to obtain consistent estimates of Σ.

$W = Q_\mu$, covariance matrix to $\check{\Sigma} \otimes Q_{\mu_g}$, and the set of instruments as $[A, B] = I_G \otimes Q_{\mu_g}[X_g^{(1)}, X_g^{(2)}, X_g^{(3)}]$. See Appendix 1.F for details and extensions to EC and HT estimators.

1.4.4 Differences between the G3SPD estimator and other available estimators

Even when the G3SPD estimator incorporates information from within and between estimators (1^{st} and 2^{nd} steps), it follows a different logic from that of EC or HT estimators. Following Maddala (1971), Baltagi (1981b, 2005), we can establish that the G3SPD estimator weighs equally the within and between estimators, coinciding with the OLS estimator after including the pseudo-effects. This differs from error-components and Hausman-Taylor-types estimators which are weighted sums of within and between estimators, where the weights are given by the inverse of their corresponding variances.

Theorem 1.4 *The G3SPS estimator assigns equal weights to the within and between estimators, after including the pseudo-effects as a regressor, while the error-components and Hausman-Taylor-type estimators weigh the within and between estimators according to the inverse of their corresponding variances.*

Proof. See Appendix 1.D. ■

Contrary to other approaches (e.g. see Hausman and Taylor, 1981, section 2.3; Baltagi, 2003, section 2.2; Plümper and Troeger, 2007), the G3SPD estimator does not decompose the panel effects. Instead, it obtains an estimator of the effects, the *pseudo-effects*, from the residuals of a fitted model that combines within and between estimates of the coefficients. In this case, the weights assigned to the within and between estimators are not related with the variances, as in EC and HT estimators, but according to the following arrangement:

Proposition 1.1 *The pseudo-effects give weights equal to 1 to the within coefficients, weights equal to 0 to the corresponding between coefficients, and weights equal to 1 to the remaining between coefficients.*

1.5 Evaluating Bias Performance

Our aim is to compare the performance of G3SPD and HT estimators in terms of bias because, as discussed before, HT estimators are the available (and suggested) alternative to handle the problem of interest. We also evaluate the bias of other alternative popular estimators such as Pooled OLS, FE and BE estimators, and EC with different variance estimators. Estimators are applied to both fixed-effects and random-effects models.

We use Monte Carlo simulations to evaluate the bias performance of the estimation techniques, because by creating a controlled environment, they allow us to focus only on those specific problems we want to analyse. Moreover, we can check the behavior of different estimators applied to the same dataset. Finally, and more importantly, we can study the performance of the G3SPD (and the other estimators) in small samples, being possible to cover panel-data and time-series-cross-sections datasets.

We use three measures to evaluate the performance of the estimators: bias (measured as the mean absolute deviation of the estimated coefficients relative to the true parameter), variability (measured by the standard deviation of the estimated coefficients), and the root mean squared error (RMSE). In what follows, all the measures are presented in percentages.

1.5.1 Simulations setup

We run experiments for a system of two equations, i.e. $G = 2$, which is sufficient for evaluating bias performance. The i^{th} individual during the t^{th} period obeys the following system of simultaneous equations:

$$\begin{aligned} y_{1it} &= Y_{2it}\pi_1 + X_{1it}\beta_{11} + X_{2it}\beta_{12} + X_{3it}\beta_{13} + \\ &\quad + Z_{1i}\gamma_{11} + Z_{2i}\gamma_{12} + \varphi_1 + \mu_{1i} + \varepsilon_{1it} \end{aligned} \tag{1.12a}$$

$$\begin{aligned} y_{2it} &= Y_{1it}\pi_2 + X_{1it}\beta_{21} + X_{4it}\beta_{24} + X_{5it}\beta_{25} + \\ &\quad + Z_{1i}\gamma_{21} + Z_{3i}\gamma_{23} + \varphi_2 + \mu_{2i} + \varepsilon_{2it} \end{aligned} \tag{1.12b}$$

Note that X_1 and Z_1 are doubly exogenous variables common to both equations, while X_2, X_3 and Z_2 only appear in equation [1.12a]. X_2 is a doubly exogenous variable and X_3 and Z_2 are singly exogenous variables. Likewise for X_4, X_5 and Z_3 in equation [1.12b].

The π coefficients were obtained from a $U[-1, 1]$, while the remaining ones came from $U[0, 1] * 15$. The selected coefficients are:

$$\begin{array}{ccccccc} \pi_1 = .055 & \beta_{11} = 8 & \beta_{12} = 2 & \beta_{13} = 5 & \gamma_{11} = 7 & \gamma_{12} = 5 & \varphi_1 = 8 \\ \pi_2 = -0.2 & \beta_{21} = 12 & \beta_{24} = 7 & \beta_{25} = 3 & \gamma_{21} = 6 & \gamma_{23} = 5 & \varphi_2 = 4 \end{array}$$

To study the small and large sample properties of the estimators we vary sample sizes over N $(20, 60, 200)$ and T $(15, 30, 50)$. This yields nine cases allowing to study time-series cross-section (small fixed N, large T) and panel (large N, fixed small T) asymptotic behavior. However, due to computing limitations, for $N = 200$ we only run experiments for

$T = 15$. Each experiment was replicated 1000 times[27].

The regressors were generated following Nerlove (1971a), Im et al. (1999) and Baltagi et al. (2003) as:

Starting Values	Data Generating Process
$X_{1i1} = \frac{\eta_{1i1}}{(1-0.5^2)^{\frac{1}{2}}} + \frac{\phi_{1i}}{(1-0.5)}$	$X_{1it} = 0.5X_{1it-1} + \phi_{1i} + \eta_{1it}$
$X_{2i1} = \frac{\eta_{2i1}}{(1-0.2^2)^{\frac{1}{2}}} + \frac{\phi_{2i}}{(1-0.2)}$	$X_{2it} = 0.2X_{2it-1} + \phi_{2i} + \eta_{2it}$
$X_{3i1} = \frac{\eta_{3i1}}{(1-0.7^2)^{\frac{1}{2}}} + \frac{\mu_{1i}}{(1-0.7)}$	$X_{3it} = 0.7X_{3it-1} + \mu_{1i} + \eta_{3it}$
$X_{4i1} = \frac{\eta_{4i1}}{(1-0.2^2)^{\frac{1}{2}}} + \frac{\phi_{4i}}{(1-0.2)}$	$X_{4it} = 0.2X_{4it-1} + \phi_{4i} + \eta_{4it}$
$X_{5i1} = \frac{\eta_{5i1}}{(1-0.7^2)^{\frac{1}{2}}} + \frac{\mu_{2i}}{(1-0.7)}$	$X_{5it} = 0.7X_{5it-1} + \mu_{2i} + \eta_{5it}$
	$Z_{1i} = \xi_{1i}$
	$Z_{2i} = \mu_{1i} + \xi_{2i}$
	$Z_{3i} = \mu_{2i} + \xi_{3i}$

where $\phi_{ji}, \eta_{jit}, \xi_{ji} \sim U[-2, 2]\ \forall j$.

Regarding the individual effects, we focus on two designs:

1. *Fixed-effects case*, where μ follows $\mu_{gi} \sim N\left(0, \sigma_\mu^2\right)\ \forall g = 1, 2$, which implies that for the whole set of 1000 replications the set of μ is kept fixed.

2. *Random-effects case*, where μ follows $\mu_{gki} \sim N\left(0, \sigma_\mu^2\right)\ \forall g = 1, 2; k = 1, \ldots, 1000$. This means that we have a set of μ for each replication.

In both cases, the stochastic error terms follow $\varepsilon_{git} \sim N\left(0, \sigma_\varepsilon^2\right)\ \forall g = 1, 2$.

[27]Notice that for $N = 200$ and $T = 15$, estimating an experiment (1000 replications) takes about 62 hours using MATLAB R2008b's optimized set of commands to handle matrixes, running on an Intel Core 2 Duo CPU at 2.00 GHz with 3 Gb of physical memory. We run 77 of such experiments. Bigger panel datasets should be estimated using loops, a very slow processing strategy. Most of the experiments were conducted using MATLAB R2007b running on the following machines: an Intel Core 2 Duo CPU at 3.00 GHz with 3.25 Gb of physical memory; an Intel Core 2 CPU at 1.86 GHz with 1.00 Gb of physical memory; and an Intel Pentium IV CPU at 3.00 GHz with 2.00 Gb of physical memory.

Following previous literature, total variance for each equation is fixed at $\sigma^2 = \sigma_\mu^2 + \sigma_\varepsilon^2 = 3$. The degree of heterogeneity of the effects as measured by $\rho = \frac{\sigma_\mu^2}{\sigma_\varepsilon^2}$ varies over the following eleven values: $0, \frac{1}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, \frac{11}{12}, 1$. This allows us to study how the correlation between the unobservable effects and the regressors affects the bias of the estimators. It is apparent that X_3, Z_2, and μ_1 are correlated, likewise X_5, Z_3, and μ_2. However, when $\rho = 0$, individual effects are absent from the model, and we only have Y_1 and Y_2 as endogenous regressors due to their correlation with ε. Accordingly, variables for equation [1.12a] are classified as $X_1^{(1)} = [X_1, X_2]$, $X_1^{(2)} = X_3$, $Z_1^{(1)} = Z_1$, $Z_1^{(2)} = Z_2$, and for equation [1.12b], as $X_2^{(1)} = [X_1, X_4]$, $X_2^{(2)} = X_5$, $Z_2^{(1)} = Z_1$, $Z_2^{(2)} = Z_3$. We also identify two additional sets of variables for each equation, namely $X_g^{(3)}$ and $Z_g^{(3)}$, which contain the variables excluded from equation g that can be used as instruments. Hence, we have $X_1^{(3)} = [X_4, X_5]$, $Z_1^{(3)} = Z_3$, $X_2^{(3)} = [X_2, X_3]$, $Z_2^{(3)} = Z_2$.

1.5.2 Results

To understand the behavior of the estimators relative to sample sizes, heterogeneity of the unobservable effects, and assumptions about the "nature" of those effects, we turn to Figures 1 to 6, that display the results on the estimation for the parameters of the endogenous and singly exogenous regressors (time-varying and -invariant) corresponding to equation [1.12a][28]. We consider only the most relevant estimators in this context: G3SPD, OLS, ECC, and HTM. We present the equation-by-equation and system estimation in columns 1 and 2 of each Figure, respectively. Each three-dimensional chart shows:

[28] Results for equation [1.12b] are similar. These results, as well as the corresponding ones for the remaining estimators derived in Appendix 1.F, are not presented here for the sake of brevity though they are available upon request from the author.

the relative bias of the estimator (in absolute terms, such that higher values imply more biased estimators); the heterogeneity of the unobservable effects (i.e. *Rho,* that represents different levels of $\rho = \sigma_{\mu}^2/\sigma_{\varepsilon}^2$ where higher values imply higher heterogeneity of the effects, while $\rho = 0$ implies no effects at all); and sample size (refered to as (N, T)). Notice that discrete changes in (N, T) favors non-smooth changes in the surface (i.e. the "peaks" and "valleys" that can be observed in the charts). To facilitate comparisons, the vertical axis scale of all charts has been fixed to 0%-20%, while the color scale ranges from 0% to 35%.

[Insert Figures 1 and 2 about here]

Observe that the bias of the estimated parameters obtained by G3SPD is the lowest in all cases as compared with the other reported estimators, both in the fixed-effects and the random-effects cases. Also, as expected, equation-by-equation and system estimations do not differ significantly except for the HTM estimator, probably due to the system contamination that the weak instruments may be generating. The behavior of the G3SPD estimator relative to the sample size follows the (asymptotic) intuition: it quickly declines with increasing sample sizes; but interestingly enough, it appears to be no difference between augmenting N or T in terms of bias improvements. Regarding the heterogeneity of the effects, note that for the endogenous variable (first row in Figures 1 and 2), the bias is higher for lower levels of ρ which can be explained by the corresponding increasing size of ε, which originates the simultaneity bias. Thus, as the uncertainty associated with the effects increase, the simultaneity bias (for a fixed level of aggregated uncertainty) lessens, reflecting the way the G3SPD estimator is controlling for the μ's. On the other hand, the increasing importance of μ (in terms of uncertainty) biases more the estimates of the coefficients of the dimension-invariant variables (see Figures 3 and 4). Finally, the best bias control is associated with the dimension-varying singly-exogenous variables, which are correctly handled in the first step of the method using a within-groups estimator: as previously demonstrated, in all possible schemes of correlation, the coefficients associated with these variables are adequately obtained using within-groups estimators. See Figures 5 and 6.

[Insert Figure 3 and 4 about here]

Consider now the behavior of the *Pooled OLS estimator* which we know that under the current setup yields biased and inconsistent estimates. However, we use it as a benchmark in view of its simplicity and popularity. The results for this estimator are reported in the second row of each Figure. If we focus on the bias of the estimators of the coefficients associated to the endogenous variables, we can observe that the OLS estimators appear to be severely biased which of course is due to the fact that the estimator does not account for endogeneity and simultaneity: note that the bias is as high as 20% in small samples, and even when the sample is large, the bias can be substantial for low ρ's; see the second row of Figures 1 and 2). When there are no unobservable effects, $\rho = \frac{\sigma_\mu^2}{\sigma_\varepsilon^2} = 0$, the correlation between the stochastic error and the endogenous variable remains, given that the bias arises from this correlation only. For higher levels of ρ, the correlation between the endogenous variables and the unobservables increases, while the correlation with ε diminishes in relative terms. The bias caused by the correlation between the endogenous variable and the unobservable effects tends to dissipate due to the presence of the singly exogenous regressors which are correlated with the unobservable effects. Collinearity among the regressors due to a common component, μ, generates this result. Observing the bias of the pooled OLS estimator of the coefficients of Z_2 and X_3 reported in the second row of Figures 3 and 5 for the FE case, and Figures 4 and 6 for the RE case, it seems to be a sort of "crowding out" of the bias associated with higher ρ. On the other hand, when considering the OLS system estimator for Y_2 (Figures 1 and 2), we can see that since we are using a set of instruments in the spirit of a three-stage least-squares estimator, the bias associated to the endogenous variable attenuates. The remaining bias is attributable to the correlation between the endogenous variable and the unobservable effects rather than to the low quality of the instruments. Sample size responds to classical panel data asymptotics: important changes in bias accounted for by important changes in N, while changes in T have no major impact.

[Insert Figures 5 and 6 about here]

ECC estimators, depicted in the third row of each Figure, do not account for the endogeneity problem regarding μ but they use instruments to deal with the simultaneity bias due to ε. This estimator is equivalent to OLS with a heteroskedasticity correction, so the above mentioned conclusions on the bias and consistency of the estimators of the coefficients should not be affected by this improvement. However, the bias of the estimator associated with Y decreases for all ρ reflecting the role of instruments in controlling the bias arising from the correlation between Y and ε.

HT estimators have been widely accepted as a solution to the problem of endogeneity and dimension-invariant variables. These estimators nevertheless are an ECC estimator and, as discussed, may be strongly biased in the current setup. HTM estimates for the coefficient of the endogenous variables in Figures 1 and 2 are increasing in ρ, decreasing with the sample size, and between 8 and 13 times the bias of the G3SPD estimates (on the contrary, the G3SPD estimator becomes less biased with increasing ρ and in larger samples). This can be interpreted as the result of an inadequate set of instruments: the set of instruments used by HT estimators is not enough to overcome the bias generated by the correlation of the endogenous variables with the unobservable effects[29]. In fact, looking at Figures 5 and 6, we arrive to similar conclusions regarding the estimators of the coefficients accompanying X_3, though with lower bias given that this variable is only correlated with the unobservable effects. It seems to be the case that the set of instruments used by the HTM estimator is uncapable to account for both sources of endogeneity; moreover, in the fixed-effects case, it exacerbates the bias because it does not account for the true structure of the model, i.e. the existence of fixed effects. In that case, while the bias of the G3SPD estimator diminishes quickly with the aggravation of the correlation problem, the bias of the HTM estimator increases with ρ.

Although the graphical inspection is intuitive and allow us to form a general idea of the behavior of the bias of these estimators, it neglects an important issue: the trade-off between bias and variability. In general, the root mean square error (RMSE) is suggested as an adequate measure of this trade-off, interpreting the lower values as indicators of better quality estimators. Nevertheless, this trade-off comprises several features that are sometimes ignored: (i) a given RMSE value may represent low bias-high variability or high bias-low variability, changing from estimator to estimator, and from sample to sample; (ii) reliability of the RMSE depend on the sample size, the presence of outliers, and the particular configuration of the sample. When the estimators persistently control bias to the detriment of variability, or viceversa, the RMSE is no longer an adequate measure of the trade-off. These problems motivate us to study not only the bias, but also the variability

[29]Notice that there is also a quality problem associated with the instruments that is known as the "weak instruments" problem. Stock and Yogo (2004) discuss this problem and point out the effect that weak instruments may have on biasing the estimation of the coefficients.

and the RMSE.

The first observation regards the information the RMSE is contributing: when the researcher needs to choose a methodology, in the present context, the RMSE appears not to be as informative as expected; moreover, it could lead the practioner to choose, for example, the OLS estimator based on the results for the endogenous set of variables at the cost of highly biased estimates of the parameters (that in our simulated example can be as high as 20%). In general, the RMSE appears to be closely representing the variability of the estimates: when the estimates are highly volatile, the RMSE tends to overrepresent the role of the variability to the detriment of the bias; thus highly biased estimates may remain undercover by the behavior of the variability.

An important feature of the G3SPD estimator is that it performs well irrespective of whether the model contains fixed- or random-effects, while the error-components-based estimators (i.e. ECC and HTM) and Pooled OLS behave better when the model contains random-effects, as can be concluded from the results. Thus, from an estimation perspective, the assumptions about the type of effect appears to be irrelevant.[30]

For the sake of completeness, some comments follow regarding the estimation of the coefficients of $X_g^{(1)}$, $Z_g^{(1)}$, and φ_g. The G3SPD estimator obtains less biased estimates for those coefficients than the other methods, particularly HTM, for both fixed-effects and random-effects cases. Again, system contamination worsens the HTM estimators for the entire system of equations.[31] Though this is not the focus of this chapter, it is worth noting

[30]This means in no way that for inference, prediction, or economic interpretation, the distinction between fixed and random effects is not important. The point is that when estimating a model with the G3SPD estimator, the discussion is not centred anymore on the methodology associated with the assumptions about the unobservables, but on conditioning or not on a set of (estimated) effects, as we do with other variables in the model.

[31]Figures and tables for these coefficients are available upon request from the author.

that according to the simulation outcomes, estimates for the unobservable μ obtained from G3SPD procedures are strongly consistent for both fixed-effects and random-effects cases, as infered from the unbiasedness of the estimates of the parameter ϕ (introduced in the third step of the G3SPD estimator as in model [1.9]). One advantage of having an estimate of the unobservable effects is that they can be used later in models as a proxy for unaccounted events and characteristics of the observations (e.g. see Bulkley et al., 2004; Jensen and Webster, 2009).

1.6 Three Classical Examples, Revisited

In Section 1.2, we illustrated the problem at hand with three textbook-examples. Now we apply the G3SPD estimator to each of these problems to evaluate the behavior of the estimator as compared with other estimators used in the corresponding articles.

To evaluate the impact of the differences in the estimated coefficients, we look at the behavior of the fitted models: better estimates should imply better fitting of the data. In general, we find that the G3SPD set of estimates outperforms the other methods suggested in the original articles, showing in all cases a sensible lower bias in terms of fitted data. In what follows, we revisit each example in turn. In Subsection 1.6.4, we perform an empirical exercise to verify the behavior of the G3SPD estimator when the correlation of the regressors and the unobservables is due to omitted variables. We use the returns to schooling example as the benchmark model.

1.6.1 Gravity Model of Bilateral Trade Flows

Serlenga and Shin (2007) estimated a general gravity equation of bilateral trade flows of the form:

$$\begin{aligned} RTTF_{it} &= \beta' X_{it} + \gamma' Z_i + \varepsilon_{it}, i = 1, \ldots, N, t = 1, ..., T \\ \varepsilon_{it} &= \alpha_i + \theta_t + u_{it} \end{aligned}$$

where the index i represents each country-pair (home-foreign), and t represents time. The sample covers bilateral trade flows for 15 EU member countries (Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxemburg, Netherlands, Portugal, Spain, Sweden, United Kingdom) forming 91 country pairs ($N = 91$). The annual data covers a period of 42 years ($T = 42$), from 1960 to 2001.

The variables $RTTF_{it}$, X_{it}, and Z_i are defined as a combination of features of the countries in each pair.[32] The dependent variable, $RTTF_{it}$, corresponds to the logarithm of real total trade flow between any two countries. The time-varying variables, X_{it}, are all in logs (except for the dummy variables) and include the bilateral real exchange rate (RER), real output (GDP), the absolute value of the difference between per capita GDPs of trading countries (RLF), an index that captures the relative size of two countries in terms of GDP (SIM), a dummy variable that is equal to one when both countries belong to the European Community (CEE), and a dummy variable equaling one when both trading partners adopt the same currency (EMU). On the other hand, the time-invariant variables, Z_i, include the logarithm of the geographical distance between capital cities ($DIST$), a shared border dummy that is equal to one when the trading partners share a border (BOR), and a common

[32]See Serlenga and Shin (2007) for details on the construction of the variables. The dataset is available for download from the *Journal of Applied Econometrics* web site.

language dummy (LAN) that has a value equal to one when both countries speak the same official language. Finally, the model captures the time-specific effects using a set of 41 time dummies (1960 is the base year) corresponding to θ_t.

We estimate two versions of the same model, including and excluding the time dummies. Table 1 shows the results for the model specification that does not control for time-specific effects. The first 6 columns resemble those in Serlenga and Shin (2007) and were estimated for comparative and verification purposes. The last column shows the estimates obtained using the G3SPD estimator. Even when most of the coefficients are very similar to those reported by Serlenga and Shin, three important differences emerge. First, the variable of interest, LAN_i, appears to be non-significant implying that speaking the same language (indirectly, sharing similar cultural and historical backgrounds between trading countries) does not improve bilateral trade flows. This conflict with Serlanga and Shin's results. Second, the negative impact of the distance between capital cities (capturing transport costs) appears to be more strong and significative than reported in the article. Finally, even when inference issues are out of the scope of this chapter, notice that the standard errors of the time-invariant variables shrink relative to those reported by other methods.[33] In Figure 7, we depict the original $RTTF_{it}$ series for four country-pairs: Belgium-France, Netherlands-Sweden, Austria-Italy, and Denmark-Greece. The combinations of country-pairs account for cases different alternatives of common language and common border. As can be seen, while the G3SPD fitted data closely resembles the original data series, the Hausman-Taylor-type estimators present the poorest fitting.

[33] In a companion paper, Avanzini (2010), we show that such standard errors are more accurate than those reported by other methods, contributing to improve inference.

[Insert Table 1 and Figure 7 about here]

The remaining estimations, corresponding to the model that controls for time-specific effects, are reported in Table 2, even when the coefficients corresponding to the 41 time dummies are not reported there for the sake of brevity. Again, coefficients obtained by the G3SPD estimator and compiting Hausman-Taylor-type estimators appear to be very similar for the time-varying regressors, while important differences arise in the coefficients accompanying the time-invariant variables. For example, while $DIST_i$ appears to be non-significant under HTM, it is negative and significant at 1% implying distance (v.g. transport costs) may be imposing important barriers to bilateral trade. Having a common border is not relevant under the HTM estimation while significant at 1% and positive under G3SPD. Finally, having a common language is again non-significant, conflicting with Serlenga and Shin's findings. Figure 8 depicts the corresponding original series and fitted model for the previous pair of countries under the current model. The G3SPD estimates again produce a superior fitting of the data when compared to the other reported methods.

[Insert Table 2 and Figure 8 about here]

Even when the coefficients of the time-varying variables are very similar under various estimation approaches, the G3SPD estimator is by far more reliable in obtaining the coefficients (and fitting the data) for the time-invariant regressors.

1.6.2 Economics of Crime

The second example comes from Cornwell and Trumbull (1994) and Baltagi (2006). They estimate a crime model using the following specification:

$$R_{it} = \beta' X_{it} + \gamma' P_{it} + \alpha_i + \varepsilon_{it}, i = 1, \ldots, N; t = 1, \ldots, T,$$

where R_i is the crime rate, X_{it} contains variables which control for the relative return to legal opportunities, as well as other observable county characteristics that may be correlated with the crime rate, and P_{it} contains a set of deterrent variables which proxy for the probability of arrest (PA), the probability of conviction (usually conditional on arrest) (PC), the probability of imprisonment (usually conditional on conviction) (PP), and the severity of the punishment (S). The α_i are fixed effects which reflect unobservable county-specific characteristics that may be correlated with $[X_{it}, P_{it}]$. The annual data corresponds to 90 counties in North Carolina over the period 1981-1987.

The variables used in the estimation are the following: the crime rate, R_{it}, is the ratio of FBI index crimes to county population, while the probability of arrest, PA, is proxied by the ratio of arrests to offenses ($PRBARR$). Individuals' perceptions of the probabilities of conviction and prison, PC and PP, are proxied by the ratio of convictions to arrests ($PRBCONV$) and the proportion of total convictions resulting in prison sentences ($PRBPRIS$), respectively. The sanction severity, S, is measured by the average prison sentence length in days ($AVGSEN$). The number of police per capita ($POLPC$) is included in the control vector X as a measure of a county's ability to detect crime. Opportunities in the legal sector are captured by the average weekly wage in the county by industry: construction ($WCON$); transportation, utilities and communications ($WTUC$); wholesale and retail trade ($WTRD$); finance, insurance and real estate ($WEIR$); services ($WSER$); manufacturing ($WMFG$); and federal, state and local government ($WFED$, $WSTA$ and $WLOC$). County characteristics include population density ($DENSITY$), the proportion of county population that is male and between the ages of 15 and 24 ($PCTYMLE$), along with the proportion that is minority or nonwhite ($PCTMIN$). The latest variable is time-invariant, as well as $WEST$ and $CENTRAL$, two dummies that capture regional or cultural factors that may affect the crime rate in western and central counties. Also a dummy variable, URBAN, accounts for counties with populations over 50,000. All the non-dummy variables are used in logarithms in the estimation. The model also includes time-specific effects that are captured by time dummies (base year is 1981). However, in our presentation we concentrate in the simpler model, without accounting for time-specific effects.

According to Cornwell and Trumbull (1994), the estimation must control for both sources of endogeneity, i.e. conventional simultaneity relating $PRBARR$ and $POLPC$, and neglected heterogeneity (unobservable country-specific characteristics). Conventional error-components and OLS estimators are biased in this context. Because of that the authors use FE2SLS which allow for instrumental variables to control simultaneity bias, and fixed effects to control for neglected heterogeneity. The problem with their estimation outcomes is that none of the coefficients appear to be significant missing the policy objective of finding the determinants of crime (see Table 3, column 3, which corresponds to the chosen model).

[Insert Table 3 about here]

On the other hand, Baltagi (2006) takes the same dataset and applies a EC2SLS which resulted in much of the deterrent variables to be significant (except for severity of sanction). Baltagi argues that the problem is that Cornwell and Rupert (1994) discarded the error-components model because they use a Hausman test to select the model. This test, according to Baltagi, may be biased if there are endogenous regressors present. Instead, using an extension of the Hausman test to compare FE2SLS and EC2SLS (see Baltagi, 2004), he finds that the EC2SLS is not biased and that it is the suitable estimator for this problem. Results of Cornwell and Rupert (1994) and Baltagi (2006) estimations are reported in Table 3 for comparative purposes.

However, when we apply the G3SPD estimator to the same dataset Baltagi (2006) uses, we find that, coincident with Cornwell and Rupert (1994), the none of the coefficients are significant except for some time-invariant county characteristics (namely, $CENTRAL$ and $PCTMIN$). See Table 3, column 7. To see why this difference arise, we plot the fitted values for the logarithm of county crime rate in Figure 9 as obtained from the estimation coefficients reported in columns 6 and 7 of Table 3. The 45-degree line represents perfect fitting, while values below that line indicate that the fitted crime rate is below the actual one. As can be observed, while the G3SPD fitted values lie along the 45-degree line, the EC2SLS fitted values gathered in a cloud lying right below the 45-degree line. This indicates that the EC2SLS estimates are systematically downward biased.[34] The resulting problem here is that policymaking may be conducted by these outcomes, expecting important reductions in crime rate that are not feasible in the real world. The reasons for this non-significance of the coefficients may rely on multicollinearity among the regressors (for example, those relating wages), or an inadequate use of the deterrent variables.[35]

[34]Recall the information shown in Figure 9 corresponds to the fitted values of the logarithm of the crime rate. Hence, more negative values correspond to smaller positive (between 0 and 1) values.

[35]Multi-level or hierarchical modelling may help to disentagle the relationship between variables that are

[Insert Figure 9 about here]

The bottom line here is that even when asymptotic properties of tests (such as the extended Hausman test suggested by Baltagi, 2004) and estimators (such as EC2SLS suggested by Baltagi, 1981) may be appealing, in small samples the behavior of the estimators can differ largely, and the G3SPD is able to overcome these problems as it emerges from this example.

1.6.3 Wage Equations and Returns to Schooling

Our last revisited example comes from a long tradition in empirical microeconomics: the estimation of Mincerian wage equations and returns to schooling, experience and other observable characteristics of the individuals. The results of these estimations have been widely used to conduct human capital policymaking and to evaluate educational and training programs.

Even when many models have been estimated, we concentrate on Cornwell and Rupert (1988), and a replication by Baltagi and Khanti-Akom (1990). Both articles focus on determining the efficiency gains of estimating the wage equation using different assumptions under the Hausman-Taylor approach. Different assumptions on the correlation structure between observables and unobservables lead to different sets of instruments (e.g. the modified-Hausman and Taylor set, the Amemiya and McCurdy set, the Breusch, Mizon and Schmidt set, etc.). The Mincerian wage equation they estimate is the following:

$$\begin{aligned}\log(wage_{it}) &= \beta' X_{it}^{(1)} + \delta' X_{it}^{(2)} + \gamma' Z_i^{(1)} + \theta' Z_i^{(2)} + \varepsilon_{it}, i = 1, \ldots, N; t = 1, \ldots, T \\ \varepsilon_{it} &= \alpha_i + \lambda_t + u_{it}\end{aligned}$$

relevant in different circumstances.

where, as before, $X_{it}^{(1)}$ are the doubly-exogenous dimension-varying variables, $X_{it}^{(2)}$ are the singly-exogenous dimension-varying variables, $Z_i^{(1)}$ are the doubly-exogenous time-invariant variables, and $Z_i^{(2)}$ are the singly-exogenous time-invariant variables. They assume the stochastic error has three components: an individual-specific effect, a time-specific effect, and a totally stochastic component. The problem is that the individual-specific effects may include non-observed characteristics such as ability, ambition, family background, etc. The correlation between the years of education attended and these unobservable characteristics are a major concern because the measured returns to schooling may be biased, and more important, the direction and magnitude of such bias are unknown.[36] The dimension-varying variables include years of full-time work experience (EXP) and its square ($EXP2$), weeks worked (WKS), occupation ($OCC = 1$, if the individual has blue-collar occupation), industry ($IND = 1$, if the individual works in a manufacturing industry), residence ($SOUTH = 1$, $SMSA = 1$, if the individual resides in the south, or in a standard metropolitan statistical area), marital status ($MS = 1$, if the individual is married), and union coverage ($UNION = 1$, if the individual's wage is set by a union contract). The dimension-invariant variables account for gender ($FEM = 1$, if the individual is female), ethnics ($BLK = 1$, if the individual is black), and years of education (ED). The sample is a panel of 595 individuals observed over the period 1976-1982 drawn from the Panel Study of Income Dynamics (PSID), collected by Baltagi and Khanti-Akom (1990), and made available by the *Journal of Applied Econometrics*.

[36]Contradictory evidence on the direction and magnitude of the bias have been reported repeatedly. For example, see Chamberlain and Griliches (1975), Griliches (1977), Chamberlain (1978), Griliches et al. (1978), Griliches (1979), Hausman and Taylor (1981), among others. Griliches (1977) presents a very complete review of the topic.

We first analyze Cornwell and Rupert's suggestion. They assume that $X_{it}^{(1)} = \{WKS, SOUTH, SMSA, MS\}$, $X_{it}^{(2)} = \{EXP, EXP2, OCC, IND, UNION\}$, $Z_i^{(1)} = \{FEM, BLK\}$, and $Z_i^{(2)} = \{ED\}$. This leads to a set of estimates that are reported in Tables 4 and 5, with and without controlling for time-specific effects, respectively. According to Cornwell and Rupert (1988), important efficiency gains are obtained as they move from the HTM to the HTAM to the BMS estimator. These efficiency gains are more apparent for the time-invariant coefficients. When we apply the G3SPD estimator, we find that the coefficients accompanying the time-variant variables are very similar to those reported under different estimators. However, very important differences emerge for the time-invariant determinants of wage. First, the returns to education obtained by G3SPD are about 30% of those obtained using Hausman-Taylor-type estimators, even when we control for time effects. In fact, the return to schooling under G3SPD is very similar to that of OLS under both estimation approaches. Second, the gender gap is is more pronounced when estimating the models with the G3SPD estimator: without controlling for time effects, the gap increases by about 130%; when accounting for time effects, the gap increases by around 40% on average. The opposite effect can be observed for the ethnics gap: afroamerican people receives around 25% less salary on average (depending on the considered HT method) versus 15.8% under G3SPD, when we do not account for time-specific effects. Differences narrow when time dummies are introduced in the estimation (see Table 5).

[Insert Table 4 and 5 about here]

Which is the impact of these differences in estimated coefficients? In Figure 10, we plot the cumulative distribution of the absolute bias of the fitted values of the log of the real wages for the models reported in Tables 4 and 5. Given that each curve indicates the cumulative frequencies of the absolute bias of the fitted values, the quicker a curve accumulates, the less the total bias. As can be see in both panels of the figure, while G3SPD is the estimator that more rapidly accumulates bias, the HT family of estimators and GLS are the more chronic ones. In other terms, G3SPD first-order stochastically dominates the other estimators in terms of absolute bias, which implies that the mean absolute bias is the smaller for the set of estimators plotted in Figure 10.

[Insert Figure 10 about here]

Baltagi and Khanti-Akom (1990) also notice that the assumption about the singly-exogeneity of some of the regressors may be affecting not only efficiency but more importantly they may be biasing the coefficients. Hence, they suggest, estimate and test the same model but assuming the following arrangement of singly- and doubly-exogenous variables: $X_{it}^{(1)} = \{OCC, SOUTH, SMSA, IND\}$, $X_{it}^{(2)} = \{EXP, EXP2, WKS, MS, UNION\}$, $Z_i^{(1)} = \{FEM, BLK\}$, and $Z_i^{(2)} = \{ED\}$. This lead to different sets of instruments than those used by Cornwell and Rupert (1988). Moreover, they find that using all the available instruments under BMS may generate biased results because not all instruments are exogenous. With these new assumptions, Baltagi and Khanti-Akom estimate the same two versions of the wage equation, i.e. with and without controlling for time-specific effects, which we report in Tables 6 and 7, respectively, for comparative purposes. We do the proper applying the G3SPD estimator to the same models and report the results in the last column of each of those tables.

[Insert Tables 6 and 7 about here]

While the G3SPD estimates appear to be similar to those reported in Tables 4 and 5, HT-type estimators are different, specially for the time-invariant regressors. While the returns to education are now lower than before, even when controlling for time effects, the coefficient of ethnic discrimination increases in absolute terms while the gender gap shrinks. Even when Baltagi and Khanti-Akom (1990) explain the gains of changing the assumptions, when we look at the cumulative distribution of the absolute bias of the fitted values of the log of the wages, as we did before, we find that such changes do not help to narrow HT-type estimators bias (see Figure 11). Again, the G3SPD first-order stochastically dominates the other estimators in terms of absolute bias.

[Insert Figure 11 about here]

Notice that changes in the assumptions about exogeneity of the regressors can have important impact on efficiency but also on the actual estimated coefficients, and that even when the assumptions asymptotically ensures unbiasedness of the estimates, in small samples the behavior of the estimators may be unreliable. On the contrary, the G3SPD shows a stable behavior mainly due to the fact that it does not require any assumption about the distribution of the effects (whenever they are fixed or random) or the correlation of the regressors with the effects.

1.6.4 A Step Beyond: Dimension-specific Effects, Omitted Variables, and Pseudo-effects

Even when Monte Carlo simulations and the three revisited examples show that the G3SPD estimator outperforms some of the widely known panel data estimators, an important discussion is still missing. What are the pseudo-effects? Are they only dimension-specific effects? Do they also capture omitted variables?

Usually, we consider that the panel "effects" represent differences in the constant term (fixed or randomly distributed) among groups. The effects reflect unobservable characteristics that may influence the observed behavior or economic outcome. The pseudo-effects capture the same concepts but incorporating the fact that there exists a latent bias (coming from the second step estimation of the G3SPD) that may make the pseudo-effects differ from the "true" panel effects. This is the reason for calling them "pseudo-effects" instead of simply effects. For example, when estimating the wage equation, economic theory (and intuition) tell us that some unobservable characteristics of the person (ability, ambition, family background, luck) may be correlated with the attained education, and in turn, with work experience, the choice of the industry or economic sector, and other observable characteristics. Moreover, we can argue that even when ethnics or gender are totally exogenous to the individual, they may be correlated with the development of abilities and skills in early stages of the childhood, access to different quality levels of education, personal interactions and social networks, differential attitudes when facing life, etc. Many of these characteristics are unobservable to the researcher and are encompassed under a general concept, the effects. The G3SPD uses the first two steps results to construct an estimate of these effects that encompass a wide variety of individual characteristics unobservable to the researcher.

Resuming the discussion about the returns to schooling in the last subsection, notice that from an economic point of view, the G3SPD estimates shown there pose some important questions that need to be answered and that may help to understand what the G3SPD estimator is doing. Why are the returns to schooling (i.e. coefficient accompanying

ED) very similar under OLS, BE, and G3SPD? Why the returns to experience (i.e. the coefficients accompanying EXP and $EXP2$) are high, increasing, and similar to other estimates but very different from those of OLS? Why are the differences in estimated gender and ethnic gaps important? Are the pseudo-effects erroneously capturing information that generates the decline in the returns to schooling? What information are the pseudo-effects capturing?

To show how the G3SPD deals with the unobservable characteristics, specially when the omitted variables are correlated with the included regressors, and at the same time, to answer whether the G3SPD and its pseudo-effects are introducing additional distortions to the estimated coefficients, we perform a set of Monte Carlo simulations for the wage equation example.

Monte Carlo Simulations Our main objective is to show how the G3SPD reacts when the model contains an omitted variable correlated with the included regressors, and the panel effects are either fixed or random. We assume that our sample coincides with the panel of 595 individuals observed over the period 1976-1982 drawn from the Panel Study of Income Dynamics (PSID), collected by Baltagi and Khanti-Akom (1990), totalling 4165 observations. We also assume that the model structure corresponds to the one showed in Table 4, and that the "true" impact of each covariate coincide with those reported there for the G3SPD estimator. Additionally, the years of education of each individual are assumed to be determined by

$$\begin{aligned} ED_i &= f(ABILITY_i,\ other\ personal\ characteristics_i,\ random\ term_i) \qquad (1.13) \\ &= \rho ABIL_i +\ aggregate\ random\ term_i \end{aligned}$$

The variable $ABIL_i$ is a time-invariant variable representing individual's ability, and ρ is the correlation between ability and education. Technically speaking, equation [1.13] is the missing equation in the returns to schooling model responsible for the singly-exogeneity of ED, EXP, $EXP2$, OCC, IND, $UNION$ (and probably FEM and BLK). We consider three different levels of correlation between ED_i and $ABIL_i$: no correlation ($\rho = 0$), moderate correlation ($\rho = 0.35$), and high correlation ($\rho = 0.7$). Given that ED_i already exist, we generate $ABIL_i$ to accomplish these assumptions as follows:

$$\rho = 0 \quad \rightarrow \quad ABIL_i = \omega_i \times 8.5 \times 0.01$$

$$\rho = 0.35 \quad \rightarrow \quad ABIL_i = (0.5 \times ED_i + 0.5 \times \omega_i \times 7.5) \times 0.01$$

$$\rho = 0.7 \quad \rightarrow \quad ABIL_i = (0.5 \times ED_i + 0.5 \times \omega_i \times 2.8) \times 0.01$$

where ED_i is actual data, and $\omega_i \sim N(0, 1)$.

We generate 500 replications of each of the six experiments (three levels of correlation between education and ability, under fixed and random effects worlds). In the fixed-effects world, we add to each of the three sets of 500 replications the same set of effects, $FE_i \sim N(0, 0.02)$, while in the random-effects world, we add a set of effects, $RE_{ji} \sim N(0, 0.02)$, to each replication in each set, where $j = 1, \ldots, 500$ indicates the replication. Finally, we add a completely stochastic term to each observation given by $\eta_{jit} \sim N(0, 1)$. The logarithm of the wages has been obtained as follows:

Fixed-effects world:

$$\log(wage_{jit}) = \beta' X_{it}^{(1)} + \delta' X_{it}^{(2)} + \gamma' Z_i^{(1)} + \theta' Z_i^{(2)} + FE_i + ABIL_i^{(\rho)} + \eta_{jit}$$

Random-effects world:

$$\log(wage_{jit}) = \beta' X_{it}^{(1)} + \delta' X_{it}^{(2)} + \gamma' Z_i^{(1)} + \theta' Z_i^{(2)} + RE_{ji} + ABIL_i^{(\rho)} + \eta_{jit}$$

where $j = 1, \ldots, 500$, $i = 1, \ldots, N$, $t = 1, \ldots, T$ indicates the replication, the individual and the time period, respectively. The superindex (ρ) indicates the level of correlation between ED and $ABIL$.

Results In Table 8, we display the results for a selected set of regressors and four estimators, namely OLS, EC, HTM, and G3SPD.[37] The most important result relates to the returns to schooling. Under both fixed-effects and random-effects worlds, the G3SPD estimator outperforms the Hausman-Taylor estimator. When the correlation between of the omitted variable and ED is zero, G3SPD shows a maximum bias of 1% (under the fixed-effects world). On the other hand, the HTM estimator can be biased by more than 13%. When the correlation between ability and education is moderate ($\rho = 0.35$) or high ($\rho = 0.7$), the bias of the G3SPD estimator is smaller than the bias of other methods. Moreover, the HTM estimator is the most biased one. Its bias ranges between 2.5 and 3.6 times the G3SPD bias under the fixed-effects assumption, and between 1.6 and 1.9 times the G3SPD bias under the random-effects assumption.

[Insert Table 8 about here]

The results for the coefficient of work experience are almost unbiased in all cases under both effects assumptions for all methods. However, the HTM estimates of the coefficient accompanying the higher order term of experience is systematically more biased than those obtained by the G3SPD estimator.

Gender differences present mixed results for both HTM and G3SPD estimators, but the bias ranges between 0.5% and 3.5%. More problematic is the coefficient representing

[37] The corresponding report for the rest of the variables in the model, as well as the estimations using other methods like FE, BE, and GLS, are available upon request from the author.

ethnic differences: while the HTM estimators bias is over 15% in most of the cases, the G3SPD estimator bias is important when ability is not correlated with education (7.6% when assuming fixed effects, and 6.2% when assuming random effects). However, for stronger endogeneity of ED, the bias lowers to around 2% on average, being systematically higher under random effects assumptions (about 1.5% higher).

Notice also that the empirical standard error of the estimated coefficients is always larger for the HTM estimator than for the G3SPD estimator, indicating that beyond the problem of bias, it shows a problem of accuracy that may distort inference.

Summarizing, the discussed evidence shows that the G3SPD incorporates the omitted variables into the pseudo-effects, and adequately control for the emerging bias when such variables are correlated with the included regressors. On the other hand, the HTM estimator performs poorly in terms of bias and variability, specially when estimating the coefficients of endogenous variables. Regarding the questions about the adequacy of the estimates of the coefficients obtained by G3SPD, the evidence support the 5% return to each year of schooling, and the 11% return to each year of work experience. The problem with this outcome is that returns to experience more than double returns to education: a simple conclusion is that education does not pay off, which conflicts with the general belief. However, as pointed out by Heckman et al. (2003), the reason for such controversial results may reside in the theoretical model and not in the estimation method.

1.7 Concluding Remarks

We have considered the estimation of single-equation and systems of simultaneous equations models with panel data, when the "true model" contains unobservable effects. We studied the problem of obtaining consistent estimates when (i) the set of regressors includes dimension-invariant variables, and (ii) some (or all) of the regressors are correlated with the unobservable effects. To account for these issues we developed a three-step estimation procedure, the Generalized Three-Step Panel Data estimator (G3SPD). The method consists of three estimation steps: the first step is aimed to obtain consistent estimates of the coefficients of the dimension-varying covariates, while the second step obtains a set of estimators (not necessarily unbiased) for both the dimension-varying and the dimension-invariant covariates. Both sets of estimators are combined to obtain a set of estimated unobservable effects that we call *pseudo-effects*. The third step includes the pseudo-effects into a pooled regression together with dimension-varying and dimension-invariant covariates. According to our simulation outcomes, this three-step procedure yields less biased estimators for all coefficients when compared with other methods such as Pooled OLS, FE and BE, EC, and HT-type estimators.

The proposed G3SPD estimator has remarkable characteristics: first, it outperforms all other panel data estimators by a substantial measure, whether in terms of bias, variability, and RMSE. Second, the performance of the G3SPD estimator does not depend on assuming fixed or random effects, given that the procedure does not require a decision over a set of possible estimation methods as in the usual approaches. Third, the estimator handles both the *"panel"* and the *"time-series cross-section"* setups. Moreover, according to our simulation, it presents a good performance in finite samples, and its bias diminishes with increasing total sample size, irrespective of whether N or T increases. Fourth, classifying the regressors as singly- or doubly-exogenous (a key issue in HT procedures) is irrelevant because there is no need to instrumentalize the correlation between the regressors and the unobservable effects once we have introduced the pseudo-effects in the regression. Fifth, the residuals of the 3^{rd} Step of G3SPD estimator are closer to the true innovations of the model, which implies better forecasting, inference, and estimation of the $n-$dimensional confidence-intervals.

Beyond these characteristics, there are some practical advantages in implementing this estimator: first, to facilitate its application to single-equation and to simultaneous equations models (both equation-by-equation and system estimation), we present two estimators that can be used with the G3SPD estimator and are capable of accomodating standard methods as particular cases, while their statistical properties remain intact. Both estimators conveniently incorporate suitable sets of instruments to control pure endogeneity originated by the correlation between the regressors and the stochastic error (simultaneity bias). The G3SPD equation-by-equation estimator also proved to perform very similar to system estimation, given that we are analyzing it for the linear case. This implies that in this context, equation-by-equation modelling can be used whenever there is a risk of generalized bias arising in system estimation when one of the equations is misspecified.

Second, the method deals with dimension-invariant variables and endogeneity of regressors in a simple and elegant way, and it relies on widely known statistical properties, namely those regarding within- and between-groups estimators, and GMM estimators for systems of equations. Its estimation procedure is simple, apparent, and quick: estimating G3SPD implies running a within-groups estimator, a between-groups estimator, and a pooled regression. The estimator is able to accomodate several alternative specifications that account for SUR specifications, block-diagonal simultaneous equations (partitioned systems of equations), unbalanced panels, $n-$way or multidimensional panels, etc, and can be easily implemented using standard econometric packages.

To show the performance of the G3SPD estimator with actual data, we revisited three classical examples in panel data literature, namely a gravity model of international trade flows, an economic model of crime, and a Mincerian wage equation. Results differ from those reported in the literature specially for the coefficients accompanying the endogenous regressors. To study the "nature" of the pseudo-effects, we perform an empirical exercise to verify how the G3SPD reacts when omitted variables generate endogeneity problems. We use the wage equation as a benchmark, and find that the G3SPD estimator accurately (in terms of bias and variability) obtain estimators for time-varying and time-invariant regressors that are correlated with the omitted variable (in this case, ability) for different levels of correlation of the regressors and the unobservables. This evidence supports the former results from Monte Carlo simulations favoring the G3SPD estimator, and gives a clear picture of its functioning.

Bibliography

[1] Abowd, John, Robert Creecy, and Francis Kramarz (2002). "Computing person and firm effects using linked longitudinal employer-employee data", Cornwell University

Working Paper.

[2] Amemiya, Takeshi (1971). "The estimation of the variances in a variance-components model", International Economic Review, Vol. 12, pp. 1-13.

[3] Amemiya, Takeshi (1977). "The maximum likelihood and the nonlinear three-stage least squares estimator in the general nonlinear simultaneous equation model", Econometrica, Vol. 45, No. 4, pp. 955-968.

[4] Amemiya, Takeshi, and Thomas MaCurdy (1986). "Instrumental-variable estimation of an error-components model", Econometrica, Vol. 54, No. 4, pp. 869-880.

[5] Arellano, Manuel, and Jinyong Hahn (2006). "Understanding bias in nonlinear panel models= some recent developments", manuscript, CEMFI and UCLA.

[6] Avanzini, Diego (2010). "A Generalized Three-Step Panel Data Estimator: Efficiency Issues", manuscript, revised version, Catholic University of Chile.

[7] Balestra, Pietro and Jayalakshmi Varadharajan-Krishnakumar (1987). "Full information estimations of a system of simultaneous equations with error components structure", Econometric Theory, Vol. 3, pp. 223-246.

[8] Baltagi, Badi (1980). "On seemingly unrelated regressions with error components", Econometrica, Vol. 48, No. 6, pp. 1547-1551.

[9] Baltagi, Badi (1981a). "Pooling: an experimental study of alternative testing and estimation procedures in a two-way error components model", Journal of Econometrics, Vol. 17, pp. 21-49.

[10] Baltagi, Badi (1981b). "Simultaneous equations with error components", Journal of Econometrics, Vol. 17, pp. 189-200.

[11] Baltagi, Badi (1984). "A Monte Carlo study for pooling time series of cross-section data in the simultaneous equations model", International Economic Review, Vol. 25, No. 3, October, pp. 603-624.

[12] Baltagi, Badi (2005, 3rd ed.; also 2001, 2nd ed., reprint 2005). Econometric Analysis of Panel Data. John Wiley & Sons, LTD, New York.

[13] Baltagi, Badi (2006). "Estimating an economic model of crime using panel data from North Carolina", Journal of Applied Econometrics, Vol. 21, pp. 543-547.

[14] Baltagi, Badi (2009). A companion to Econometric Analysis of Panel Data. John Wiley & Sons, LTD, New York.

[15] Baltagi, Badi, and Sophon Khanti-Akom (1990). "On efficient estimation with panel data: an empirical comparison of instrumental variables estimators", Journal of Applied Econometrics, Vol. 5, pp. 401-406.

[16] Baltagi, Badi, Georges Bresson and Alain Pirotte (2003). "Fixed effects, random effects or Hausman-Taylor? A pretest estimator", Economics Letters, Vol. 79, pp. 361-369.

[17] Baltagi, Badi. (2004). “A Hausman test based on the difference between fixed effects two-stage least squares and error components two-stage least squares, problem 04.1.1”. Econometric Theory, Vol. 20, pp. 223–224.

[18] Becker, Gary. (1968). “Crime and punishment: an economic approach,” Journal of Political Economy, Vol. 76, pp. 169–217.

[19] Bellmann, Lutz, Jörg Breitung and Joachim Wagner (1989). "Bias correction and bootstrapping of error component models for panel data: theory and applications", Empirical Economics, Vol. 14, pp. 329-342.

[20] Breusch, Trevor, Grayham Mizon, and Peter Schmidt (1989). "Efficient estimation using panel data", Econometrica, Vol. 57, No. 3, pp. 695-700.

[21] Bulkley, George, Richard Harrif, and Renata Herrerias (2004). "Why does book-to-market value of equity forecast cross-section stock returns?". International Review of Financial Analysis, Vol. 13, pp. 153-160.

[22] Chamberlain, Gary (1978). “Omitted Variable Bias in Panel Data: Estimating the Returns to Schooling”, Annales de l’inséé, No. 30/31, The Econometrics of Panel Data (Apr. - Sep.), pp. 49-82.

[23] Chamberlain, Gary, and Zvi Griliches. (1975). “Unobservables with a Variance-Components Structure: Ability, Schooling, and the Economic Success of Brothers”, International Economic Review, Vol. 16, No. 2 (June), pp. 422-449.

[24] Chowdhury, G. and S. Nickell (1985). "Hourly earnings in the U.S.: another look at unionization, schooling, sickness, and unemployment using PSID data," Journal of Labor Economics, Vol. 3, pp. 38-69.

[25] Cornwell, Christopher, and Peter Rupert (1988). "Efficient estimation with panel data:

an empirical comparison of instrumental variables estimators", Journal of Applied Econometrics, Vol. 3, pp. 149-155.

[26] Cornwell, Christopher, and William Trumbull (1994). “Estimating the economic model of crime with panel data”, Review of Economics and Statistics, Vol. 76, No. 2, pp. 360-366.

[27] Cornwell, Christopher, Peter Schmidt and Donald Wyhowski (1992). "Simultaneous equations and panel data", Journal of Econometrics, Vol. 51, pp. 151-181.

[28] Davies, Stephen, Kenneth Erickson, Steve Vickner, Dana Hoag, and Richard Nehring (2005). "An Error-Components Three-Stage Least-Squares Model of Investment Allocation by Farm Households", unpublished paper, presented at the 2005 AAEA Annual Meeting.

[29] Egger, Peter and Michael Pfaffermayr (2004). "Distance, trade and FDI: a Hausman-Taylor SUR approach", Journal of Applied Econometrics, Vol. 19, No. 2, pp. 227-246.

[30] Ehrlich, Issac. (1973). “Participation in illegitimate activities: a theoretical and empirical investigation”, Journal of Political Economy, Vol. 81, pp. 521–567.

[31] Fernandez-Val, Ivan, and Francis Vella (2009). "Bias corrections for two-step fixed effects panel data estimators", manuscript, Boston University and Georgetown University.

[32] Green, Donald, Soo Yeon Kim and David Yoon. (2001) “Dirty Pool”, International Organization, Vol. 55, pp. 441-468.

[33] Greene, William (2002, 5th ed.). Econometric Analysis, Pearson Education, Prentice Hall, New Jersey.

[34] Greene, William (2004). "Fixed effects and bias due to the incidental parameters problem in the Tobit model", Econometric Reviews, Vol. 23, No. 2, pp. 125-147.

[35] Griliches, Zvi (1977). "Estimating the Returns to Schooling: Some Econometric Problems", Econometrica, Vol. 45, No. 1 (January), pp. 1-22.

[36] Griliches, Zvi (1979). "Sibling Models and Data in Economics: Beginnings of a Survey", The Journal of Political Economy, Vol. 87, No. 5, Part 2: Education and Income Distribution (October), pp. S37-S64.

[37] Griliches, Zvi, Bronwyn Hall and Jerry Hausman. (1978). "Missing Data and Self-Selection in Large Panels", Annales de l'inséé, No. 30/31, The Econometrics of Panel Data (Apr. - Sep.), pp. 137-176.

[38] Hansen, Lars (1982). "Large sample properties of Generalized Method of Moments estimators", Econometrica, Vol. 50, No. 4, pp. 1029-1054.

[39] Hausman, Jerry (1978). "Specification Tests in Econometrics", Econometrica, Vol. 46, pp. 1251-1371.

[40] Hausman, Jerry, and William Taylor (1981). "Panel data and unobservable individual effects", Econometrica, Vol. 49, No. 6, pp. 1377-1398.

[41] Hazlett, Thomas, and Roberto Muñoz (2009a). "A welfare analysis of spectrum allocation policies", Rand Journal of Economics, Vol. 40, No. 3, pp. 424-454.

[42] Hazlett, Thomas, and Roberto Muñoz (2009b). “Spectrum allocation policies in Latin America: a welfare analysis”, Information Economics and Policy, Vol. 21, No. 4, pp. 261-278.

[43] Heckman, James (1981). “The incidental parameters problem and the problem of initial conditions in estimating a discrete time-discrete data stochastic process”, in Charles Manski and Daniel McFadden (eds.), Structural analysis of discrete panel data with econometric applications, pp. 179-195.

[44] Heckman, James, Lance Lochner and Petra Todd (2003). “Fifty years of Mincer earnings regressions”, NBER Working Paper 9732 (May).

[45] Holly, Alberto (1982). “A remark on Hausman’s specification test”, Econometrica, Vol. 50, No. 3, pp. 749-759.

[46] Honoré, Bo (2002). "Non-linear models with panel data", The Institute for Fiscal Studies (Department of Economics, UCL), Cemmap Working Paper CWP13/02.

[47] Horrace, William and Kurt Schnier (2008). "Fixed-effect estimation of highly-mobile production techonologies", CPR Working Paper Series No. 87.

[48] Hsiao, Cheng (2002, 2nd ed.; also 1986, 1st ed.). Analysis of Panel Data. Cambridge University Press, Cambridge.

[49] Im, Kyung, Seung Ahn, Peter Schmidt and Jeffrey Wooldridge (1999). "Efficient estimation of panel data models with strictly exogenous explanatory variables", Journal of Econometrics, Vol. 93, pp. 177-201.

[50] Iversen, Torben, and Thomas Cusack. (2000) “The Causes of Welfare State Expansion: Deindustrialization or Globalization?”, World Politics, Vol. 52, No. 3, pp. 313-349.

[51] Jensen, Paul and Elizabeth Webster (2009). "What creates abnormal profits: Collusion, efficiency or strategy?", Intellectual Property Research Institute of Australia, Working Paper No. 09/09.

[52] Knigth, Brian (2002). "Endogenous federal grants and crowd-out of state government spending: theory and evidence from the Federal Highway Aid Program", American Economic Review, Vol. 92, No. 1, pp. 71-92.

[53] Krishnakumar, Jaya (2002). "A SUR-EC-AR system gravity model of trade", unpublished mimeo, University of Geneva.

[54] Krishnakumar, Jaya (2006). "Time invariant variables and panel data models: a Generalised Frisch-Waugh Theorem and its implications", in Badi Baltagi (ed.), "Panel Data Econometrics", Elsevier B.V. (Oxford, UK), pp. 119-132.

[55] Lam, Shun Yin, Mark Vandenbosch, John Hulland, and Michael Pearce (2001). "Evaluating promotions in shopping environments: decomposing sales response into attraction, conversion, and spending effects", Marketing Science, Vol. 20, No. 2, pp. 194-215.

[56] Lancaster, Tony (2000). "The incidental parameter problem since 1948", Journal of Econometrics, Vol. 95, pp. 391-413.

[57] Lancaster, Tony (2002). "Orthogonal parameters and panel data", Review of Economic Studies, Vol. 69, No. 3, pp. 647-666.

[58] Lillard, L. and R. Willis (1978). "Dynamic aspects of earnings mobility", Econometrica, Vol. 46, pp. 985-1012.

[59] Maddala, G. S. (1971). "The use of variance components models in pooling cross section and time series data", Econometrica, Vol. 39, pp. 341-358.

[60] Maddala, G. S. and T. D. Mount (1973). "A comparative study of alternative estimators for variance components models used in econometric applications", Journal of the American Statistical Association, Vol. 68, pp. 324-328.

[61] McClellan, Mark and Douglas Staiger (1999). "Comparing hospital quality at for-profit and not-for-profit hospitals", NBER, Working Paper 7324, August.

[62] Mincer, Jacob. (1974). Schooling, Experience, and Earnings. Columbia University Press, New York.

[63] Mundlak, Yair (1978). "On the Pooling of Time Series and Cross Section Data", Econometrica, Vol. 46, No. 1, pp. 69-85.

[64] Nerlove, Marc (1971a). "Further evidence on the estimation of dynamic economic relations from a time series of cross sections", Econometrica, Vol. 39, No. 2, pp. 359-382.

[65] Nerlove, Marc (1971b). "A note on error components models", Econometrica, Vol. 39, No. 2, pp. 383-396.

[66] Newey, Whitney, and Richard Smith (2004). "Higher order properties of GMM and Generalized Empirical Likelihood estimators", Econometrica, Vol. 72, No. 1, pp. 219-255.

[67] Neyman, J. and Elizabeth Scott (1948). "Consistent estimation from partially consistent observations", Econometrica, Vol. 16, No. 1, pp. 1-32.

[68] Plümper, Thomas, and Vera Troeger (2007). "Efficient Estimation of Time-Invariant and Rarely Changing Variables in Finite Sample Panel Analyses with Unit Fixed Effects", Political Analysis, Vol. 15, pp. 124-139.

[69] Podivinsky, Jan (1999). "Finite sample properties of GMM estimators and tests", chapter 5 in László Mátyás (ed.), Generalized Method of Moments Estimation, chapter 5, Cambridge University Press.

[70] Prucha, Ingmar (1985). "Maximum likelihood and instrumental variable estimation in simultaneous equation systems with error components", International Economic Review, Vol. 26, No. 2, pp. 491-506.

[71] Ramalho, Joaquim (2006). "Bootstrap bias-adjusted GMM estimators", Economics Letters, Vol. 92, pp. 149-155.

[72] Schmidt, Peter (1988). "Estimation of a fixed-effect Cobb-Douglas system using panel data", Journal of Econometrics, Vol. 37, pp. 361-380.

[73] Schmidt, Peter (1990). "Three-stage least squares with different instruments for different equations", Journal of Econometrics, Vol. 43, pp. 389-394.

[74] Serlenga, Laura and Yongcheol Shin (2007). "Gravity models of intra-EU trade: application of the CCEP-HT estimation in heterogeneous panels with unobserved common time-specific factors", Journal of Applied Econometrics, Vol. 22, pp. 361-381.

[75] Sickles, Robin (2005). "Panel estimators and the identification of firm-specific efficiency levels in parametric, semiparametric and nonparametric settings", Journal of Econometrics, Vol. 126, No. 2, pp. 305-324.

[76] Stock, James, and Motohiro Yogo (2004). "Testing for weak instruments in linear IV regression", manuscript, Harvard University (first draft: 2001; NBER Technical Working Paper 284: 2002).

[77] Swamy, P. A. V. B. and S. S. Arora (1972). "The exact finite sample properties of the estimators of coefficients in the error components regression models", Econometrica, Vol. 40, No. 2, pp 261-275.

[78] Taylor, William (1980). "Small sample considerations in estimation from panel data", Journal of Econometrics, Vol. 13, pp. 203-223.

[79] Vella, Francis and Marno Verbeek (1999). "Two-step estimation of panel data models with censored endogenous variables and selection bias", Journal of Econometrics, Vol. 90, pp. 239-263.

[80] Wallace, Thomas and Ashiq Hussain (1969). "The use of error components models in combining cross-section and time-series data", Econometrica, Vol. 37, pp. 55-72.

[81] Wansbeek, Tom and Arie Kapteyn (1982a). "A class of decompositions of the variance-covariance matrix of a generalized error components model", Econometrica, Vol. 50, No. 3, pp. 713-724.

[82] Wansbeek, Tom and Arie Kapteyn (1982b). "A simple way to obtain the spectral

decomposition of variance components models for balanced data", Communications in Statistics, Vol. A11, pp. 2105-2112.

[83] Wansbeek, Tom and Arie Kapteyn (1983). "A note on spectral decomposition and maximum likelihood estimation of ANOVA models with balanced data", Statistics and Probability Letters, Vol. 1, pp. 213-215.

[84] Wansbeek, Tom and Arie Kapteyn (1989). "Estimation of the error components model with incomplete panels", Journal of Econometrics, Vol. 41, pp. 341-361.

[85] Windmeijer, Frank (2005). "A finite sample correction for the variance of linear efficient two-step GMM estimators", Journal of Econometrics, Vol. 126, pp. 25-51.

[86] Wooldridge, Jeffrey (1996). "Estimating systems of equations with different instruments for different equations", Journal of Econometrics, Vol. 74, pp. 387-405.

[87] Wooldridge, Jeffrey (2002). Econometric Analysis of Cross Section and Panel Data, The MIT Press, Cambridge, MA.

[88] Wyhowski, Donald (1994). "Estimation of a panel data model in the presence of correlation between regressors and a two-way error-component", Econometric Theory, Vol. 10, pp. 130-139.

1.A Proof of Theorem 1.1

This proof follows from Cornwell et al. (1992, Theorem 1).

Proof. The normal log-likelihood of $y = Y\pi + X\beta + Z\gamma + Z_\varphi \varphi + Z_\mu \mu + \varepsilon$ is

$$\begin{aligned} \mathcal{L}_1 &= cons - \frac{NT}{2} \ln |\Sigma| - \\ &-\frac{1}{2} \left(y - Y\pi - X\beta - Z\gamma - Z_\varphi \varphi - Z_\mu \mu \right)' \Sigma^{-1} \left(y - Y\pi - X\beta - Z\gamma - Z_\varphi \varphi - Z_\mu \mu \right) \end{aligned}$$

Applying a "between-groups" and "within-groups" decomposition to the system of equations we get:

$$\begin{aligned} P_\mu y + Q_\mu y &= P_\mu Y\pi + Q_\mu Y\pi + P_\mu X\beta + Q_\mu X\beta + \\ &+ P_\mu Z\gamma + P_\mu Z_\varphi \varphi + P_\mu Z_\mu \mu + P_\mu \varepsilon + Q_\mu \varepsilon \end{aligned}$$

Replace Y with $\hat{Y}$, X with $\hat{X}$, and Z with $\hat{Z}$, to control for endogeneity (due to correlation with μ and ε) when needed, assuming the variables with a hat are the instrumentalized version of the corresponding ones. Hence, we rewrite $\mathcal{L}_1$ as:

$$\begin{aligned} \mathcal{L}_1 &= cons - \frac{NT}{2} \ln |\Sigma| - \frac{1}{2} (P_\mu y + Q_\mu y - P_\mu \hat{Y}\pi - Q_\mu \hat{Y}\pi - P_\mu \hat{X}\beta - Q_\mu \hat{X}\beta - \\ &- P_\mu \hat{Z}\gamma - P_\mu Z_\varphi \varphi - P_\mu Z_\mu \mu)' \Sigma^{-1} (P_\mu y + Q_\mu y - P_\mu \hat{Y}\pi - Q_\mu \hat{Y}\pi - \\ &- P_\mu \hat{X}\beta - Q_\mu \hat{X}\beta - P_\mu \hat{Z}\gamma - P_\mu Z_\varphi \varphi - P_\mu Z_\mu \mu) \end{aligned}$$

Grouping terms we have:

$$\begin{aligned} \mathcal{L}_1 &= cons - \frac{NT}{2} \ln |\Sigma| - \frac{1}{2} \left(Q_\mu y - Q_\mu F\delta \right)' \Sigma^{-1} \left(Q_\mu y - Q_\mu F\delta \right) - \\ &-\frac{1}{2} \left(P_\mu y - P_\mu F\delta - P_\mu H\alpha \right)' \Sigma^{-1} \left(P_\mu y - P_\mu F\delta - P_\mu H\alpha \right) \end{aligned}$$

where $F\delta = \hat{Y}\pi + \hat{X}\beta$ and $H\alpha = \hat{Z}\gamma + Z_\varphi \varphi + Z_\mu \mu$. Maximizing $\mathcal{L}_1$ over δ, α, Σ yields the following first order conditions:

$$\begin{aligned}
\frac{\partial \mathcal{L}_1}{\partial \delta} &= 0 = (Q_\mu F)'\Sigma^{-1}(Q_\mu y - Q_\mu F\delta) + (P_\mu F)'\Sigma^{-1}\left(P_\mu y - P_\mu F\delta - P_\mu H\alpha\right) && (1.14a)\\
\frac{\partial \mathcal{L}_1}{\partial \alpha} &= 0 = (P_\mu H)'\Sigma^{-1}\left(P_\mu y - P_\mu F\delta - P_\mu H\alpha\right) && (1.14b)\\
\frac{\partial \mathcal{L}_1}{\partial \Sigma} &= 0 = -\frac{NT}{|\Sigma|} + \left(Q_\mu y - Q_\mu F\delta\right)'\Sigma^{-2}\left(Q_\mu y - Q_\mu F\delta\right) + && (1.14c)\\
&\quad + \left(P_\mu y - P_\mu F\delta - P_\mu H\alpha\right)'\Sigma^{-2}\left(P_\mu y - P_\mu F\delta - P_\mu H\alpha\right)
\end{aligned}$$

From [1.14b] we have:

$$P_\mu y - P_\mu F\delta - P_\mu H\alpha = 0 \tag{1.15}$$

On the other hand, the normal log-likelihood of the transformed model $Q_\mu y = Q_\mu \hat{Y}\pi + Q_\mu \hat{X}\beta + Q_\mu \varepsilon$ is:

$$\mathcal{L}_2 = cons - \frac{NT}{2}\ln|\Sigma^*| - \frac{1}{2}\left(Q_\mu y - Q_\mu F\delta\right)'\Sigma^{*-1}\left(Q_\mu y - Q_\mu F\delta\right)$$

with F defined as before. The first order conditions of the maximization of $\mathcal{L}_2$ are:

$$\begin{aligned}
\frac{\partial \mathcal{L}_2}{\partial \delta} &= (Q_\mu F)'\Sigma^{*-1}(Q_\mu y - Q_\mu F\delta) = 0\\
\frac{\partial \mathcal{L}_2}{\partial \Sigma} &= -\frac{NT}{|\Sigma^*|} + \left(Q_\mu y - Q_\mu F\delta\right)'\Sigma^{*-2}\left(Q_\mu y - Q_\mu F\delta\right) = 0
\end{aligned}$$

which are identical to [1.14a] and [1.14c] after accounting for [1.15]. Thus, for a wide range of dimension-invariant covariates that can be written in a $H\alpha$ form, the maximum likelihood estimators of $\delta' = [\pi', \beta']'$ and Σ obtained from the original model and from the within-groups transformed model are equivalent. ■

1.B Proof of Theorem 1.2

The proof follows from the results in Baltagi (1981b), Prucha (1985), and Balestra and Varadharajan-Krishnakumar (1987).

Proof. Following the same reasoning as in the proof of Theorem 1.1, and assuming the effects μ are random, the normal log-likelihood of

$$y = Y\pi + X\beta + Z\gamma + Z_{\varphi}\varphi + (Z_{\mu}\mu + \varepsilon) \tag{1.16}$$

is given by

$$\begin{aligned}
\mathcal{L}_1 &= cons - \frac{NT}{2}\ln|\Sigma| - \\
&\quad -\frac{1}{2}\left(y - \hat{Y}\pi - \hat{X}\beta - \hat{Z}\gamma - Z_{\varphi}\varphi\right)'\Sigma^{-1}\left(y - \hat{Y}\pi - \hat{X}\beta - \hat{Z}\gamma - Z_{\varphi}\varphi\right)
\end{aligned}$$

with Σ defined as in [1.7]. Note that we have replaced Y with $\hat{Y}$, X with $\hat{X}$, and Z with $\hat{Z}$, to control for endogeneity and simultaneity when needed (i.e. as before, assume the variables with hat are the instrumentalized version of the corresponding ones). After applying "between-groups" and "within-groups" decompositions to [1.16], and grouping terms, we can rewrite $\mathcal{L}_1$ as:

$$\begin{aligned}
\mathcal{L}_1 &= cons - \frac{NT}{2}\ln|\Sigma| - \frac{1}{2}\left(Q_{\mu}y - Q_{\mu}F\delta\right)'\Sigma^{-1}\left(Q_{\mu}y - Q_{\mu}F\delta\right) - \\
&\quad -\frac{1}{2}\left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right)'\Sigma^{-1}\left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right)
\end{aligned}$$

where $F\delta = \hat{Y}\pi + \hat{X}\beta$ and $H\alpha = \hat{Z}\gamma + Z_{\varphi}\varphi$. Maximizing $\mathcal{L}_1$ over δ, α, Σ yields the following first order conditions:

$$\frac{\partial\mathcal{L}_1}{\partial\delta} = 0 = (Q_{\mu}F)'\Sigma^{-1}(Q_{\mu}y - Q_{\mu}F\delta) + (P_{\mu}F)'\Sigma^{-1}\left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right) \tag{1.17}$$

$$\frac{\partial\mathcal{L}_1}{\partial\alpha} = 0 = (P_{\mu}H)'\Sigma^{-1}\left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right) \tag{1.18}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}_1}{\partial\Sigma} &= 0 = \left[-\frac{NT}{|\Sigma|}\right] + \left(Q_{\mu}y - Q_{\mu}F\delta\right)'\Sigma^{-2}\left(Q_{\mu}y - Q_{\mu}F\delta\right) + \\
&\quad + \left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right)'\Sigma^{-2}\left(P_{\mu}y - P_{\mu}F\delta - P_{\mu}H\alpha\right)
\end{aligned} \tag{1.19}$$

From [1.18], it follows that $P_\mu y - P_\mu F\delta - P_\mu H\alpha = 0$, and first order condition [1.17] becomes:

$$\begin{aligned} 0 &= (Q_\mu F)'\Sigma^{-1}(Q_\mu y - Q_\mu F\delta) \\ &= F'Q_\mu\left[\Sigma_1^{-1}\otimes P_{\mu_g} + \Sigma_\varepsilon^{-1}\otimes Q_{\mu_g}\right](Q_\mu y - Q_\mu F\delta) \\ &= F'Q_\mu\left[\Sigma_\varepsilon^{-1}\otimes Q_{\mu_g}\right](Q_\mu y - Q_\mu F\delta) \\ &= F'Q_\mu\Sigma^{*-1}(Q_\mu y - Q_\mu F\delta) \end{aligned} \tag{1.20}$$

where $\Sigma^* = \Sigma_\varepsilon \otimes I_{NT}$. Note that this result follows from the orthogonality of P_{μ_g} and Q_{μ_g}.

On the other hand, consider the corresponding "within-groups" transformed model $Q_\mu y = Q_\mu \hat{Y}\pi + Q_\mu \hat{X}\beta + Q_\mu\varepsilon$. Its normal log-likelihood is given by

$$\mathcal{L}_2 = cons - \frac{NT}{2}\ln|\Sigma^*| - \frac{1}{2}(Q_\mu y - Q_\mu F\delta)'\,\Sigma^{*-1}\,(Q_\mu y - Q_\mu F\delta)$$

with F as previously defined. The first order conditions for the maximum of $\mathcal{L}_2$ are:

$$\frac{\partial \mathcal{L}_2}{\partial \delta} = (Q_\mu F)'\Sigma^{*-1}(Q_\mu y - Q_\mu F\delta) = 0 \tag{1.21}$$

$$\frac{\partial \mathcal{L}_2}{\partial \Sigma} = -\frac{NT}{|\Sigma^*|} + (Q_\mu y - Q_\mu F\delta)'\,\Sigma^{*-2}\,(Q_\mu y - Q_\mu F\delta) = 0 \tag{1.22}$$

As it is apparent, [1.21] is identical to [1.20]. However, the solution for the variance-covariance matrix differs between [1.19] and [1.22]: Σ^* is partially using the information contained in the model, neglecting the between-groups variation, which is the usual inefficiency result when comparing within-groups estimators *versus* error-components ones.

■

1.C Proof of Theorem 1.3

To facilitate the proof of the theorem, we introduce some notation and a theorem taken from Krishnakumar (2006).

Notation 1.1 *Assume that the true model is given by this modified version of the model in the theorem:*

$$y = Y^{*}\pi + X^{*}\beta + Z^{*}\gamma + [\mu\alpha + \nu] \tag{1.23}$$

where the starred regressors stand for the "true" uncorrelated regressors. To simplify exposition, we assume that the mean of the residuals is zero, i.e. $P_{\mu}\nu = 0$. *We also assume that* μ *is a* $NT \times 1$ *vector with the individual-specific effects already arranged (i.e. each row of* μ *is now equivalent to the horizontal sum of each row of* $Z_{\mu}\mu$*).*

To introduce endogeneity in the model, we assume that the observed regressors are functions of the true uncorrelated regressors and the unobservable effects, namely $Y = f(Y^{*}, \mu, \varepsilon)$, $X = f(X^{*}, \mu)$, *and* $Z = f(Z^{*}, \mu)$. *The instrumentalized version of* Y *follows* $\hat{Y} = f(Y^{*}, \mu)$. *Particularly, assume the simple structures:*

$$\begin{aligned} \hat{Y} &= Y^{*} + \mu = \tilde{Y}^{*} + \bar{Y}^{*} + \mu \\ X &= X^{*} + \mu = \tilde{X}^{*} + \bar{X}^{*} + \mu \\ Z &= Z^{*} + \mu \end{aligned} \tag{1.24}$$

Theorem 1.5 (The Generalised Frisch-Waugh Theorem, Krishnakumar, 2006, Theorem 5.1) *In the generalised regression model* $y = X_1\beta_1 + X_2\beta_2 + u$ *with* $E(u) = 0$ *and* $V(u) = V$, *positive definite, non-scalar, the generalised least square estimator of a subvector of the coefficients, say* β_2, *can be written as*

$$\beta_2^{GLS} = \left(R_2' V^{-1} R_2\right)^{-1} R_2' V^{-1} R_1$$

where

$$R_1 = y - X_1\left(X_1'V^{-1}X_1\right)^{-1}X_1'V^{-1}y$$

$$R_2 = X_2 - X_1\left(X_1'V^{-1}X_1\right)^{-1}X_1'V^{-1}X_2$$

Proof. See Appendix A5 in Krishnakumar (2006). ■

We now turn to the proof of Theorem 1.3:

Proof. Replacing [1.24] in [1.23] we obtain the true model under the current set of assumptions:

$$\begin{aligned} y &= \left[\hat{Y} - \mu\right]\pi + [X - \mu]\beta + [Z - \mu]\gamma + [\mu\alpha + \nu] \\ &= \hat{Y}\pi + X\beta + Z\gamma + [\mu(\alpha - \pi - \beta - \gamma) + \nu] \\ &= \hat{Y}\pi + X\beta + Z\gamma + \eta \end{aligned}$$

Assume that $(\alpha - \pi - \beta - \gamma) = 1$. The between-tranformed model is:

$$\bar{y} = \overline{\hat{Y}}\pi + \bar{X}\beta + Z\gamma + \mu$$

while its estimable version is:

$$\bar{y} = \overline{\hat{Y}}\pi + \bar{X}\beta + Z\gamma + \eta$$

Applying Theorem 1.5, the between-groups estimators for π, β and γ are given by:

$$\begin{aligned}
\hat{\pi}_{be} &= \left(\overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{\hat{Y}}\right)^{-1} \overline{\hat{Y}}' Q_{[\overline{X},Z]} \bar{y} \\
plim\left\{\hat{\pi}_{be}\right\} &= plim\left\{\left(\overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{\hat{Y}}\right)^{-1} \overline{\hat{Y}}' Q_{[\overline{X},Z]} \left[\overline{\hat{Y}}\pi + \bar{X}\beta + Z\gamma + \mu\right]\right\} \\
&= \pi + plim\left\{\left(\overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{\hat{Y}}\right)^{-1} \overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{X}\beta\right\} + \\
&\quad + plim\left\{\left(\overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{\hat{Y}}\right)^{-1} \overline{\hat{Y}}' Q_{[\overline{X},Z]} Z\gamma\right\} + \\
&\quad + plim\left\{\left(\overline{\hat{Y}}' Q_{[\overline{X},Z]} \overline{\hat{Y}}\right)^{-1} \overline{\hat{Y}}' Q_{[\overline{X},Z]} \mu\right\} \\
&= \beta + \frac{M_{\overline{\hat{Y}}\overline{X}}}{M_{\overline{\hat{Y}}}}\beta + \frac{M_{\overline{\hat{Y}}Z}}{M_{\overline{\hat{Y}}}}\gamma + \frac{M_{\overline{\hat{Y}}\mu}}{M_{\overline{\hat{Y}}}}
\end{aligned}$$

$$\begin{aligned}
\hat{\beta}_{be} &= \left(\bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{X}\right)^{-1} \bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{y} \\
plim\left\{\hat{\beta}_{be}\right\} &= plim\left\{\left(\bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{X}\right)^{-1} \bar{X}' Q_{[\overline{\hat{Y}},Z]} \left[\overline{\hat{Y}}\pi + \bar{X}\beta + Z\gamma + \mu\right]\right\} \\
&= \beta + plim\left\{\left(\bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{X}\right)^{-1} \bar{X}' Q_{[\overline{\hat{Y}},Z]} \overline{\hat{Y}}\pi\right\} + \\
&\quad + plim\left\{\left(\bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{X}\right)^{-1} \bar{X}' Q_{[\overline{\hat{Y}},Z]} Z\gamma\right\} + \\
&\quad + plim\left\{\left(\bar{X}' Q_{[\overline{\hat{Y}},Z]} \bar{X}\right)^{-1} \bar{X}' Q_{[\overline{\hat{Y}},Z]} \mu\right\} \\
&= \beta + \frac{M_{\overline{X}\overline{\hat{Y}}}}{M_{\overline{X}}}\pi + \frac{M_{\overline{X}Z}}{M_{\overline{X}}}\gamma + \frac{M_{\overline{X}\mu}}{M_{\overline{X}}}
\end{aligned}$$

$$
\begin{aligned}
\hat{\gamma}_{be} &= \left(Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}Z\right)^{-1} Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}\bar{y} \\
plim\left\{\hat{\gamma}_{be}\right\} &= plim\left\{\left(Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}Z\right)^{-1} Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}\left[\bar{X}\beta + Z\gamma + \mu\right]\right\} \\
&= \gamma + plim\left\{\left(Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}Z\right)^{-1} Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}\overline{\hat{Y}}\pi\right\} + \\
&\quad + plim\left\{\left(Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}Z\right)^{-1} Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}\bar{X}\beta\right\} + \\
&\quad + plim\left\{\left(Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}Z\right)^{-1} Z'Q_{\left[\overline{\hat{Y}},\bar{X}\right]}\mu\right\} \\
&= \gamma + \frac{M_{Z\overline{\hat{Y}}}}{M_Z}\pi + \frac{M_{Z\overline{X}}}{M_Z}\beta + \frac{M_{Z\mu}}{M_Z}
\end{aligned}
$$

whith $M_{BA} = B'Q_A A$ and Q_A as defined in Section 1.3. Hence, the biases are given by

$$
\begin{aligned}
Bias\left(\hat{\pi}_{be}\right) &= \frac{M_{\overline{\hat{Y}}\overline{X}}}{M_{\overline{\hat{Y}}}}\beta + \frac{M_{\overline{\hat{Y}}Z}}{M_{\overline{\hat{Y}}}}\gamma + \frac{M_{\overline{\hat{Y}}\mu}}{M_{\overline{\hat{Y}}}} \\
Bias\left(\hat{\beta}_{be}\right) &= \frac{M_{\overline{X}\overline{\hat{Y}}}}{M_{\overline{X}}}\pi + \frac{M_{\overline{X}Z}}{M_{\overline{X}}}\gamma + \frac{M_{\overline{X}\mu}}{M_{\overline{X}}} \\
Bias\left(\hat{\gamma}_{be}\right) &= \frac{M_{Z\overline{\hat{Y}}}}{M_Z}\pi + \frac{M_{Z\overline{X}}}{M_Z}\beta + \frac{M_{Z\mu}}{M_Z}
\end{aligned}
$$

■

1.D Proof of Theorem 1.4

The proof uses results from Maddala (1971), Baltagi (1981b), and Baltagi (2003, ch. 2).

Proof. First, we separate the theorem in two parts:

(1) "The G3SPS estimator assigns equal weights to the within and between estimators, after including the pseudo-effects as a regressor [...]",

(2) "[...] while the error-components and Hausman-Taylor-type estimators weigh the within and between estimators according to the inverse of their corresponding variances".

We start with the proof of the second part. Consider the following single-equation panel data model model:

$$\begin{aligned} y &= Y\pi + X^{(1)}\beta^{(1)} + X^{(2)}\beta^{(2)} + Z^{(1)}\gamma^{(1)} + Z^{(2)}\gamma^{(2)} + [Z_\mu \hat{\mu}]\,\phi + \eta \qquad (1.25) \\ \eta &= Z_\mu \mu + \varepsilon \end{aligned}$$

where μ is the dimension-specific effect such as an individual effect (time-invariant). Assume that $\eta \sim N(0, T\sigma_\mu^2 + \sigma_\varepsilon^2)$. The variables Y and X are time-varying variables, while the Z are time-invariant. Y is a set of endogenous variables. The superindex "(1)" indicates the variables are doubly-exogenous (not correlated with η), and the superindex "(2)" indicates the variables are singly-exogenous (they are correlated with μ but not with ε). We also assume that there may exist a third set of X and Z with superindex "(3) " that corresponds to the excluded variables from the equation. If we disregard singly-exogeneity, we can re-group variables and coefficients as follows:

$$\begin{aligned} X &= \left[X^{(1)}, X^{(2)}\right] \qquad \beta = \left[\beta^{(1)\prime}, \beta^{(2)\prime}\right]' \\ Z &= \left[Z^{(1)}, Z^{(2)}\right] \qquad \gamma = \left[\gamma^{(1)\prime}, \gamma^{(2)\prime}\right]' \end{aligned}$$

and the model [1.25] can be rewritten as:

$$\begin{aligned} y &= Y\pi + X\beta + Z\gamma + \eta \qquad (1.26) \\ \eta &= Z_\mu \mu + \varepsilon \end{aligned}$$

Following Maddala (1971), Baltagi (1981b, 2003), the corresponding 2SLS error-components estimator for this model is given by the weighted sum:

$$\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \end{pmatrix}_{EC2SLS} = W_1 \begin{pmatrix} \tilde{\pi} \\ \tilde{\beta} \\ 0 \end{pmatrix}_{Within2SLS} + W_2 \begin{pmatrix} \bar{\pi} \\ \bar{\beta} \\ \bar{\gamma} \end{pmatrix}_{Between2SLS}$$

where

$$\begin{aligned} \begin{pmatrix} \tilde{\pi} \\ \tilde{\beta} \\ 0 \end{pmatrix}_{W2SLS} &= \left[\begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z\right) \right]^{-1} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\tilde{H}} y \\ \begin{pmatrix} \bar{\pi} \\ \bar{\beta} \\ \bar{\gamma} \end{pmatrix}_{B2SLS} &= \left[\begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\bar{H}} \left(Y, X, Z\right) \right]^{-1} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\bar{H}} y \end{aligned}$$

and P_H is the projection matrix of the instruments, $P_H = H(H'H)^{-1}H'$, and the set of instruments are given by:

$$\begin{aligned} \tilde{H} &= Q_\mu \left[X^{(1)}, X^{(2)}, X^{(3)} \right] \\ \bar{H} &= P_\mu \left[X^{(1)}, X^{(2)}, X^{(3)}, Z^{(1)}, Z^{(2)}, Z^{(3)} \right] \end{aligned}$$

where

$$\begin{aligned} P_\mu &= Z_\mu \left(Z'_\mu Z_\mu \right)^{-1} Z'_\mu \\ Q_\mu &= I_{NT} - P_\mu \\ Z_\mu &= I_N \otimes \iota_T \end{aligned}$$

The weights are given by

$$W_1 = \left[\frac{1}{\sigma_\varepsilon^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\tilde{H}} (Y, X, Z) + \frac{1}{\sigma_1^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\bar{H}} (Y, X, Z) \right]^{-1} \times$$

$$\times \frac{1}{\sigma_\varepsilon^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\tilde{H}} (Y, X, Z)$$

$$W_2 = \left[\frac{1}{\sigma_\varepsilon^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\tilde{H}} (Y, X, Z) + \frac{1}{\sigma_1^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\bar{H}} (Y, X, Z) \right]^{-1} \times$$

$$\times \frac{1}{\sigma_1^2} \begin{pmatrix} Y' \\ X' \\ Z' \end{pmatrix} P_{\bar{H}} (Y, X, Z)$$

and $\sigma_1^2 = T\sigma_\mu^2 + \sigma_\varepsilon^2$.

If we go back to the model [1.25] and take into account the singly-exogeneity of $X^{(2)}$ and $Z^{(2)}$, we can adjust the set of instruments according to Hausman and Taylor (1981), i.e.

$$\tilde{H} = Q_\mu \left[X^{(1)}, X^{(3)} \right]$$
$$\bar{H} = P_\mu \left[X^{(1)}, X^{(3)}, Z^{(1)}, Z^{(3)} \right]$$

and the structure of the estimators will not not change, but the projection matrix of the instruments. Thus, like the EC2SLS estimator, the Hausman-Taylor estimator is a weighted sum of within and between estimators under a particular set of assumptions on the correlation of the regressors and the unobservables that lead to a different set of instruments from that of the EC2SLS estimator. This completes the proof of the second part of the theorem.

To prove the first part of the theorem, we add the pseudo-effects to the model in [1.26] to have:

$$y = Y\pi + X\beta + Z\gamma + [Z_\mu \hat{\mu}]\phi + \eta \tag{1.27}$$

We can rewrite the EC2SLS estimator to account for the new regressor as:

$$\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} = W_3 \begin{pmatrix} \tilde{\pi} \\ \tilde{\beta} \\ 0 \\ 0 \end{pmatrix}_{W2SLS} + W_4 \begin{pmatrix} \bar{\pi} \\ \bar{\beta} \\ \bar{\gamma} \\ \bar{\phi} \end{pmatrix}_{B2SLS} \tag{1.28}$$

Accordingly, the rest of the matrices are adjusted to include the new time-invariant regressor $[Z_\mu \hat{\mu}]$. Notice that the weights have to be adjusted as well. The new weights are given by:

$$\begin{aligned} W_3 \quad &= \quad \left[\frac{1}{\sigma_\varepsilon^2} \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}] \right) + \frac{1}{\sigma_1^2} \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}] \right) \right]^{-1} \times \\ &\times \frac{1}{\sigma_\varepsilon^2} \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}] \right) \end{aligned}$$

$$W_4 = \left[\frac{1}{\sigma_\varepsilon^2}\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \frac{1}{\sigma_1^2}\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right)\right]^{-1} \times$$
$$\times \frac{1}{\sigma_1^2}\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right)$$

However, under the G3SPD estimator, we assume that after including $[Z_\mu \hat{\mu}]$ in the regression, $\eta = \varepsilon$. Hence, $\sigma_1^2 = \sigma_\varepsilon^2$ and W_3 and W_4 are no longer dependant on σ_ε^2. Moreover, W_3 and W_4 become:

$$W_3 = \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right)\right]^{-1} \times$$
$$\times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}}\left(Y, X, Z, [Z_\mu \hat{\mu}]\right)$$

$$W_4 = \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times$$

$$\times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right)$$

Replacing these weights in the EC2SLS estimator [1.28], we obtain:

$$\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} = W_3 \begin{pmatrix} \tilde{\pi} \\ \tilde{\beta} \\ 0 \\ 0 \end{pmatrix}_{W2SLS} + W_4 \begin{pmatrix} \bar{\pi} \\ \bar{\beta} \\ \bar{\gamma} \\ \bar{\phi} \end{pmatrix}_{B2SLS}$$

$$
\begin{aligned}
\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} = & \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
& \times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
& \times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} y + \\
& + \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
& \times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
& \times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} y
\end{aligned}
$$

$$
\begin{aligned}
\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} &= \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
&\times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} y + \\
&+ \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \\
&\times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} y
\end{aligned}
$$

$$\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} = \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times$$

$$\left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\tilde{H}} y + \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_{\bar{H}} y \right]$$

$$\begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{EC2SLS} = \left[\begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_H \left(Y, X, Z, [Z_\mu \hat{\mu}]\right) \right]^{-1} \times \begin{pmatrix} Y' \\ X' \\ Z' \\ [Z_\mu \hat{\mu}] \end{pmatrix} P_H y$$

$$= \begin{pmatrix} \hat{\pi} \\ \hat{\beta} \\ \hat{\gamma} \\ \hat{\phi} \end{pmatrix}_{OLS(IV)}$$

where we use the result $Q + P = I$. Hence, the G3SPD can be seen as a weighted sum of the within and between estimators with equal weights for both estimators, a result that coincides with the OLS estimator (allowing for instrumental variables). This completes the proof of the first part of the theorem. ■

1.E A Note on Two-Step Regressions

Using the notation and assumptions introduced in Appendix 1.C, note that applying the two-step procedure to obtain an estimate of γ, based on the within residuals, results in:

$$\begin{aligned}
\hat{\gamma}_{R-BE} &= \left(Z'Z\right)^{-1} Z'\overline{\hat{u}} \\
plim\left\{\hat{\gamma}_{R-BE}\right\} &= plim\left\{\left(Z'Z\right)^{-1} Z'\left[Z\gamma+\mu\right]\right\} \\
&= \gamma + plim\left\{\left(Z'Z\right)^{-1} Z'\mu\right\} \\
&= \gamma + \frac{M^*_{Z\mu}}{M^*_Z}
\end{aligned}$$

This two-step procedure is not able to split up Z into its two components, but now $\hat{\gamma}_{R-BE}$ differs from $\hat{\gamma}_{be}$. At first sight, it is tempting to state that $\hat{\gamma}_{R-BE}$ is less biased than $\hat{\gamma}_{be}$. However, notice that the unobservable effects affect y through four channels: Y, X, Z, and μ. While $\hat{\gamma}_{be}$ accounts for all channels, $\hat{\gamma}_{R-BE}$ neglects the Y and X channels. Thus, using this two-step procedure will yield unbiased and consistent estimates for π and β (just like FE will) but at the cost of a still more biased set of estimators for the coefficients accompanying the dimension-invariant regressors.

It is worth mentioning that Krishnakumar (2006) extended the results of Mundlak (1978) to the case where the model also contains time-invariant regressors in the context of single-equation estimation (no endogenous Y included). Notice that her result differs from the one we have just shown: she concluded that using the two-step procedure would produce a set of unbiased and consistent estimators for β and a biased estimator for γ – due to the practical impossibility of splitting up the components of Z – in the spirit of

Mundlak's result. However, such bias only accounts for the correlation between Z and μ (represented by a parameter ϕ in her paper and by $\frac{M^*_{Z\mu}}{M^*_Z}$ in our exposition), neglecting any possible bias arising from the interaction of μ and X. The reason for that result is the particular form that is assumed for the correlation, following the one used by Mundlak. Therefore, Krishnakumar's (2006) result, though "elegant", is not as general as claimed, and only applies to the particular form of endogeneity presented in that paper.

1.F Estimation Procedures

Here we describe how to obtain six groups of estimators as particular cases of the equation-by-equation and system estimators in [1.10] and [1.11], respectively. Obtaining particular cases implies setting three components of the estimators: the set of regressors, a weighting matrix that permits incorporating covariance estimators and equation transformations, and a set of instruments (together with a consistent estimate of the covariance matrix in the case of the system estimator). This particular property of the estimators presented in this chapter is inherited from the fact that the GMM, ML and/or LS estimators coincides in the cases we review below, so we exploit this advantage.

1.F.1 Least Squares (Pooled OLS) Estimators

Under this approach, there is no considerations regarding panel dimensions effects, and the sample is treated as coming from a unique distribution that is not affected by dimensional particular shocks. To obtain the Pooled OLS estimator for a single equation, set $R_g^{OLS} = \left[Y_g, X_g^{(1)}, X_g^{(2)}, Z_g^{(1)}, Z_g^{(2)}, \iota_{NT}\right]$, $P_{[A_g,B_g]} = I_{NT}$ and $W_g = I_{NT}$, and plug them in equation [1.10]. The Pooled OLS estimator for the entire system of equations obtains rearranging regressors as in [1.11], setting $W = I_{GNT}$, the estimated variance to $\check{\Sigma} \otimes I_{NT}$, and the set of instruments to $[A, B] = I_G \otimes \left[X_g^{(1)}, X_g^{(2)}, X_g^{(3)}, Z_g^{(1)}, Z_g^{(2)}, Z_g^{(3)}, \iota_{NT}\right]$, and plugging them in [1.11]. $\check{\Sigma}$ is based on equation-by-equation estimation residuals and is given by $\check{\Sigma} = (\tilde{\varepsilon}'\tilde{\varepsilon})/NT$, where $\tilde{\varepsilon} = [\check{\varepsilon}_1, \check{\varepsilon}_2, \ldots, \check{\varepsilon}_G]$ is the $NT \times G$ matrix of estimated residuals, $\check{\varepsilon}_g = y_g - R_g\hat{\theta}_g$.

1.F.2 Fixed Effects (Within-groups) Estimators

FE models can be obtained setting $R_g = \widetilde{R_g} = \left[Y_g, X_g^{(1)}, X_g^{(2)}\right]$ and $W_g = Q_{\mu_g}$. However, we distinguished between a standard model (FEStd) with $P_{[A_g,B_g]} = I_{NT}$, and an enhaced model (FEB[38]) that incorporates instruments to account for simultaneity bias with $A_g = \widetilde{A_g} = Q_{\mu_g}\left[X_g^{(1)}, X_g^{(2)}, X_g^{(3)}\right]$ and $B_g = \widetilde{B_g} = \{\varnothing\}$. The system estimation counterpart can be obtained rearranging $\widetilde{R_g}$ $\forall g = 1, \ldots, G$ according to [1.5], setting $W = Q_\mu$, with instruments $A = I_G \otimes \widetilde{A_g}$ and $B = I_g \otimes \widetilde{B_g}$. The difference between FEStd and FEB for system estimation resides in the set of residuals used to estimate $\check{\Sigma} = (\tilde{\varepsilon}'Q_\mu\tilde{\varepsilon})/N(T-1)$, which come from their respective equation-by-equation estimates. The FE results we report are those corresponding to FEB.

Notice that none of the FE estimators is able to obtain the coefficients associated with dimension-invariant variables, due to the within-groups transformation that wipes out all dimension-invariant components.

[38]This FE estimator was proposed by Baltagi (2001, Ch. 7), and incorporates a set of instruments taken from the same panel after a transformation. The original set of instruments as posed by Baltagi is $A = Q_v X$, where $X = [X_1, X_2]$ is the set of exogenous variables: X_1 is the set of exogenous *included* variables, and X_2 is the set of exogenous *excluded* variables. The identification condition is accomplished when the rank of X_2 is greater than or equal to the number of endogenous variables in the equation. This means that we need an additional set of variables excluded from regression, the X_2, in order to get the Aitken estimator. Cornwell et al. (1992) extends the set of instruments to include the time-invariant regressors that are not correlated with the residuals. The extension to BE models is direct.

Finally, notice also that leaving almost dimension-invariant variables in the transformed regression produces the exacerbation of the associated variances, as well as inaccurate coefficient estimates. The suggestion in this case is to keep in the regression only the dimension-varying regressors to avoid problems with matrix inversion.

1.F.3 Between-groups Estimators

BE estimators are obtained using $R_g = R_g^{OLS}$ and $W_g = P_{\mu_g}$. The result is a set of T identical regressions for each individual given our matrix arrangement which implies that the regression has NT observations instead of the N observations of the cross-section regression. To remedy this, we premultiply the whole model by $\sqrt{T}$. As with FE estimators, we also distinguish between a standard model (BEStd) and an enhanced model (BEB). For equation-by-equation estimation, these estimators differ in their sets of instruments, being $P_{[A_g,B_g]} = I_{NT}$ for BEStd, and $A_g = \overline{A_g} = \{\varnothing\}$ and $B_g = \overline{B_g} = P_{\mu_g}\left[X_g^{(1)}, X_g^{(2)}, X_g^{(3)}, Z_g^{(1)}, Z_g^{(2)}, Z_g^{(3)}, \iota_{NT}\right]$ for BEB. The system counterpart can be obtained rearranging R_g $\forall g = 1, \ldots, G$ as in [1.5], setting $W = P_\mu$, $A = I_G \otimes \overline{A_g}$ and $B = I_G \otimes \overline{B_g}$. Again, both methods differ in their respective sets of residuals from equation-by-equation estimation used to estimate $\check{\Sigma} = (\tilde{\varepsilon}' P_\mu \tilde{\varepsilon})/N$.

1.F.4 Error-Components (Random Effects) Estimators

The error-components estimators assume that the unobservable effects come from a given common distribution (i.e. they are "random" effects). Baltagi (1981b, 1984) developed the estimator for systems of simultaneous equations. The EC estimator accounts for the dimension-varying and dimension-invariant variables, given that the system is not transformed as in the fixed effects case. Consequently random effects models are able to obtain those coefficients. However the assumptions (specially those regarding uncorrelatedness between regressors and effects) are hardly accomplished, and some arrangements are in order to get consistent estimates. A well-known specification test by Hausman (1978) may help to know the appropiateness of this approach when some or all of the regressors are suspected to be endogenous[39].

Single equation EC models can be obtained plugging $R_g = R_g^{OLS}$ and $W_g = \hat{\Omega}_g^{-\frac{1}{2}}$ in [1.10]. To select the instruments, we consider two cases: (i) not accounting for endogeneity of regressors, implying $P_{[A_g,B_g]} = I_{NT}$; and (ii) accounting for endogeneity of regressors (as in Baltagi, 1981b, 1984, 2005 ch. 7, and Cornwell et al., 1992), implying $A_g = \widetilde{A_g}$ and $B_g = \overline{B_g}$. The system estimation counterparts are obtained rearranging R_g $\forall g = 1, \ldots, G$ according to [1.5], with $W = I_{GNT}$ and instruments $[A, B] = I_G \otimes [A_g, B_g]$. The estimated covariance matrix based on equation-by-equation estimation is given by[40]:

$$\check{\Sigma} = \left[\frac{(\tilde{\varepsilon}' Q_\mu \tilde{\varepsilon})}{N(T-1)}\right] \otimes Q_{\mu_g} + \left[\frac{(\tilde{\varepsilon}' P_\mu \tilde{\varepsilon})}{N}\right] \otimes P_{\mu_g}.$$

Regarding the estimation of $\Omega_g = \sigma_{1,g}^2 Q_{\mu_g} + \sigma_{2,g}^2 P_{\mu_g}$, there are several feasible alternatives[41]. According to Cornwell et al. (1992), we only need consistent estimates

[39] It is worth remembering that under the random effects assumptions, this model obtains consistent and efficient estimates, while under the fixed effects model (that can be used instead when the regressors are endogenous), the estimates are consistent but less efficient. On the other hand, if the random effects model assumptions are not accomplished, then its estimates are biased, and the fixed effects model is still consistent, and now the efficient one.

[40] The expression for $\check{\Sigma}$ follows from the decomposition proposed by Wansbeek and Kapteyn (1982a, 1982b, 1983). For incomplete or unbalanced panel data estimators, the alternative proposed by Wansbeek and Kapteyn (1989) would be more appropriate.

[41] Some proposals to get a feasible estimate for Ω are: (i) Wallace and Hussain (1969), that substitutes the unknown errors with the residuals from an OLS regression; (ii) Amemiya (1971), that suggests using the LSDV residuals instead of the OLS residuals; (iii) Swamy and Arora (1972), that suggest running a "between" and "within" regressions and use those residuals for each component of the variance; (iv) Nerlove (1971b), that uses the mean squared error from the LSDV regression as a estimator of the needed variance, without adjusting by degrees of freedom.

of the $\sigma_{1,g}^2$ and $\sigma_{2,g}^2$ (not necessarily efficient ones) in order to maintain the asymptotic properties unaltered. Those estimates are given by $\widehat{\sigma}_{1,g}^2 = \frac{\tilde{u}_g' Q_{\mu_g} \tilde{u}_g}{N(T-1)}$ and $\widehat{\sigma}_{2,g}^2 = \frac{\bar{u}_g' P_{\mu_g} \bar{u}_g}{N}$. We estimate three EC models that differ in their definition of $\hat{\Omega}$:

- ECWH (based on Wallace and Hussain, 1969): $\tilde{u}_g = \bar{u}_g$ is obtained from an OLS regression.

- ECAm (based on Amemiya, 1971): the residuals come from an LSDV estimation, namely $\tilde{u}_g = \bar{u}_g = y_g - \hat{\alpha}\iota_{NT} - \widetilde{R_g}\hat{\theta}_{FEStd}$, where $\hat{\alpha} = \overline{(Dy)} - \overline{(D\widetilde{R_g})}\hat{\theta}_{FEStd}$, $D = I_N \otimes \iota_T'/T$, and the bar indicates means over N.

- ECSA (based on Swamy and Arora, 1972): $\tilde{u}_g$ comes from FEStd and $\bar{u}_g$ from BEStd.

We also estimate an EC model that accounts for endogeneity, that we call ECC, and that uses a Swamy and Arora's estimator for $\hat{\Omega}$.

1.F.5 Hausman-Taylor-type Estimators

The obvious disadvantage of the previously described EC estimators is the potential inconsistency of their estimates due to the correlation between the regressors and the effects. Several attempts can be found in the panel data literature to prevent this problem. The most well-known case is the work by Hausman and Taylor (1981) that integrates time-varying and time-invariant variables, with some of them correlated with the effects, and

Although the Wallace and Hussain's alternative is the simpler to estimate, it has been criticized by Amemiya (1971) showing that the estimators of the variance components have a different asymptotic distribution from that knowing the true disturbances. However, Bellmann et al. (1989) consider the bias of the Wallace and Hussain's variance estimator and find that the relevance of the bias depends on the particular problem at hand. To decide which variance estimate is more reliable, Taylor (1980) compared the performance of the Amemiya's and Swamy-Arora's feasible estimators and found that the Swamy-Arora estimator is more efficient (except for the fewest degrees of freedom), and that more efficient estimators of the variance components do not necessarily yield more efficient feasible GLS estimators. Other works, such as Maddala and Mount (1973) and Baltagi (1981a), have obtained similar results using Monte Carlo simultations.

proposes an instrumental variable approach to obtain consistent estimates. The proposed set of instruments is constructed using the same panel data simplifying the search for adequate instruments. Following the same line, Amemiya and MaCurdy (1986), Breusch et al. (1989), Wyhowski (1994), among others, extend the set of instruments to include other variables[42]. We use these sets of instruments to extend the ECC estimator to account for correlated dimension-varying and dimension-invariant regressors, considering three alternative definitions:

(i) Hausman-Taylor[43]: the modified Hausman and Taylor's (1981) set of instruments is given by $A_g = \widetilde{A_g}$ and $B_g = B_g^{HT} = P_{\mu_g}\left[X_g^{(1)}, X_g^{(3)}, Z_g^{(1)}, Z_g^{(3)}, \iota_{NT}\right]$, with instruments projection matrix $P_{\left[A_g, B_g^{HT}\right]} = P_{A_g} + P_{B_g^{HT}}$. The variance component $\sigma^2_{2,g}$ is consistently estimated by $\widehat{\sigma}^2_{2,g} = \frac{(\bar{u}_g^{HT})' P_{\mu_g} (\bar{u}_g^{HT})}{N-K-1}$, where $\bar{u}_g^{HT}$ is given by:

$$\bar{u}_g^{HT} = P_{\mu_g} y_g - P_{\mu_g} R_g \left(R_g' P_{\mu_g} P_{B_g^{HT}} P_{\mu_g} R_g \right)^{-1} R_g' P_{\mu_g} P_{B_g^{HT}} P_{\mu_g} y_g$$

and K is the number of regressors in the equation. The extension to system estimation implies rearranging R_g $\forall g = 1, \ldots, G$ as before, defining $W = I_{GNT}$ and setting the set of instruments to $A = I_G \otimes A_g$ and $B = diag\left[B_1^{HT}, B_2^{HT}, \ldots, B_G^{HT}\right]$.[44] We call this estimator HTM.

[42] Wyhowski (1994) extends the set of instruments proposed by Breusch et al. (1989) to the two-way panel data. We will not consider this case by now.

[43] Hausman and Taylor (1981) proposed a set of instruments for the single-equation model, namely $[Q_{\mu_g}, X_g^{(1)}, Z_g^{(1)}]$. However, Breusch et al. (1989) found that this set of instruments is not viable due to a potential bias arising when there exists endogenous variables (namely, the regression has a non-empty Y_g). Beyond that, the original set did not fit the matrix of instruments as needed in [1.10]. Instead of it, Breusch et al. (1989) propose the use of an equivalent set of instruments that is not equal to the Hausman and Taylor's set of instruments, but that yields the same estimator for their model (the Hausman-Taylor's). We consider this corrected set of intruments while we keep calling it the Hausman-Taylor's instrument set.

[44] It is easy to see that we have already mentioned two extreme cases of these sets of instruments. If there are no doubly exogenous variables, the instrument set B_g^{HT} is empty, and the resulting estimator reduces to FEBal. On the other hand, if there are no singly exogenous variables, then the resulting estimator corresponds to ECC. Similar conclusions apply to system estimators.

(ii) Amemiya-MaCurdy: Cornwell et al. (1992) propose a set of instruments based on Amemiya and MaCurdy (1986) that also includes instruments taken from the dimension-invariant set of regressors. It is given by $A_g = \widetilde{A_g}$ and $B_g = B_g^{AM} = P_{\mu_g}\left[\left(X_g^{(1)}\right)^*, \left(X_g^{(3)}\right)^*, Z_g^{(1)}, Z_g^{(3)}, \iota_{NT}\right]$, where $(\cdot)^*$ stands for a special arrangement of variables[45], and the projection matrix is $P_{\left[A_g, B_g^{AM}\right]} = P_{A_g} + P_{B_g^{AM}}$. The variance component $\sigma_{2,g}^2$ is consistently estimated by $\widehat{\sigma}_{2,g}^2 = \frac{\left(\bar{u}_g^{AM}\right)' P_{\mu_g}\left(\bar{u}_g^{AM}\right)}{N-K-1}$, where $\bar{u}_g^{AM}$ is given by:

$$\bar{u}_g^{AM} = P_{\mu_g} y_g - P_{\mu_g} R_g \left(R_g' P_{\mu_g} P_{B_g^{AM}} P_{\mu_g} R_g\right)^{-1} R_g' P_{\mu_g} P_{B_g^{AM}} P_{\mu_g} y_g$$

and K is the number of regressors in the equation. The extension to system estimation is similar to the Hausman-Taylor case[46]. The estimator is reported as HTAM.

[45] Breusch et al. (1989) explain the shape of the arrangement represented by $(\cdot)^*$: suppose we have a $NT \times K$ panel data matrix S, then there exists an $NT \times KT$ matrix S^* defined as follows:

$$S = \begin{bmatrix} S_{11} \\ \vdots \\ S_{1T} \\ \vdots \\ S_{N1} \\ \vdots \\ S_{NT} \end{bmatrix} \qquad S^* = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1T} \\ \vdots & \vdots & \ddots & \vdots \\ S_{11} & S_{12} & \cdots & S_{1T} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{NT} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{NT} \end{bmatrix}$$

This implies that a set of covariates H_g for the g^{th} equation has an associated H_g^* matrix that looks like:

$$H_g = \begin{bmatrix} H_{g11} \\ \vdots \\ H_{g1T} \\ \vdots \\ H_{gN1} \\ \vdots \\ H_{gNT} \end{bmatrix} \qquad H_g^* = \begin{bmatrix} H_{g11} & H_{g12} & \cdots & H_{g1T} \\ \vdots & \vdots & \ddots & \vdots \\ H_{g11} & H_{g12} & \cdots & H_{g1T} \\ \vdots & \vdots & \ddots & \vdots \\ H_{gN1} & H_{gN2} & \cdots & H_{gNT} \\ \vdots & \vdots & \ddots & \vdots \\ H_{gN1} & H_{gN2} & \cdots & H_{gNT} \end{bmatrix}$$

where $H_{git} = [H_{git1}, ..., H_{gitK}]$, and H_g is a $(NT \times K)$ matrix, H_{git} is a $(1 \times K)$ row-vector, and H_g^* is a $(NT \times KT)$ matrix of instruments. See Amemiya and MaCurdy (1986), Cornwell and Rupert (1988), and Breusch et al. (1989) for more details on this set of instruments.

[46] According to Cornwell and Rupert (1988) and Breusch et al. (1989), the consistency of this estimator requires a stronger exogeneity assumption for $X_g^{(1)}$ than does the consistency of the Hausman and Taylor

(iii) Breusch-Mizon-Schmidt: In Breusch et al. (1989), the authors extended the set of instruments proposed by Amemiya and MaCurdy (1986) to incorporate information from the set of singly-exogenous dimension-varying regressors. Under a configuration similar to HTM and HTAM, the corresponding set of instruments is given by $B_g = B_g^{BMS} = P_{\mu_g}\left[\left(X_g^{(1)}\right)^*, \left(X_g^{(3)}\right)^*, Z_g^{(1)}, Z_g^{(3)}, \left(Q_{\mu_g}X_g^{(2)}\right)^*, \iota_{NT}\right]$. This means that the set of deviations from the mean of the set of singly-exogenous dimension-varying variables are incorporated in an Amemiya-MaCurdy arrangement[47]. The projection

estimator. In fact, the Hausman and Taylor estimator requires only the means of the variables in $X_g^{(1)}$ to be uncorrelated with the effects, whereas Amemiya and MaCurdy estimator requires uncorrelatedness separately at every point in time. As Amemiya and MaCurdy (1986) argue, however, it is hard to think of cases where the Hausman and Taylor's exogeneity condition would hold but the Amemiya and MaCurdy's would not. In addition, the Amemiya and MaCurdy condition is required if the Hausman and Taylor estimator is to remain consistent when only subsets of the time periods $1, ..., T$ are used in estimation.

Although those exogeneity conditions are more restrictive, there exists an efficiency gain: EC models using Amemiya-MaCurdy instruments are no less efficient than those using Hausman-Taylor instruments, assuming the resulting EC model with Amemiya-MaCurdy instruments is consistent. The difference between the Hausman-Taylor and Amemiya-MaCurdy sets of instruments is that they treat differently the time-varying doubly exogenous variables: Hausman and Taylor include the means of such variables as single instruments, whereas Amemiya and MaCurdy include each of the T time-period values of such variables as a separate instrument.

A final comment on the consistency issue comes from Breusch et al. (1989) where they notice that the conditions under which the Hausman and Taylor or the Amemiya and MaCurdy estimators are efficient are testable, at least in principle, since they are just sets of linear restrictions on the reduced form equations. It may therefore be reasonable to test rather than assume the restrictions applying to each estimator in order to be consistent.

[47]Breusch et al. (1989) pointed out that the legitimacy of the set of instruments $\left(Q_{\mu_g}X_g^{(2)}\right)^*$ depends on the assumption about the nature of the correlation between $X_g^{(2)}$ and μ_g. Given they are assuming that time is the invariant dimension, the unobservable effects are constants attribuitable to each individial in the sample. In this case, Hausman and Taylor's (1981) definition of correlation between $X_g^{(2)}$ and μ_g is simply that the individual means of the variables in $X_g^{(2)}$ are correlated with the effects; thus $P_{\mu_g}X_g^{(2)}$ cannot be used as instruments.

On the other hand, $\left(Q_{\mu_g}X_g^{(2)}\right)^*$ may or may not be correlated with the effects, depending on what we assume in the reduced form equation for $X_g^{(2)}$:

- It is possible for $X_g^{(2)}$ to be correlated with the effects only because it contains a time-invariant component, and $\left(Q_{\mu_g}X_g^{(2)}\right)^*$ is legitimate.
- Or, if the part of $X_g^{(2)}$ correlated with the effects is not time-invariant, then $\left(Q_{\mu_g}X_g^{(2)}\right)^*$ is not legitimate. However, $\left(Q_{\mu_g}X_g^{(2)}\right)$ is still legitimate because $Q_{\mu_g}\mu_g = 0$.

The legitimacy of $\left(Q_{\mu_g}X_g^{(2)}\right)^*$ as an instrument set is testable. For example, a Hausman's specification test can be used to check-out the differences between HTAM and HTBMS. However, some problems might arise, and Holly (1982) discusses some considerations regarding the appropiateness of this constrast.

matrix is $P_{\left[A_g, B_g^{BMS}\right]} = P_{A_g} + P_{B_g^{BMS}}$. The variance component $\sigma^2_{2,g}$ is consistently estimated by $\widehat{\sigma}^2_{2,g} = \frac{\left(\bar{u}_g^{BMS}\right)' P_{\mu_g} \left(\bar{u}_g^{BMS}\right)}{N-K-1}$, where $\bar{u}_g^{BMS}$ is given by:

$$\bar{u}_g^{BMS} = P_{\mu_g} y_g - P_{\mu_g} R_g \left(R_g' P_{\mu_g} P_{B_g^{BMS}} P_{\mu_g} R_g \right)^{-1} R_g' P_{\mu_g} P_{B_g^{BMS}} P_{\mu_g} y_g$$

and K is the number of regressors in the equation. The extension to system estimation is straightforward following HTM. We call this estimator HTBMS.

According to Cornwell and Rupert (1988), the efficiency gains from using HTAM and HTBMS are *limited* to the estimated coefficients of the dimension-invariant variables. This can be explained because the *extra* instruments employed by those methods are dimension-invariant. Thus those instruments should be of relevance under our approach given our interest in estimating the coefficients of the dimension-invariant variables. However, Baltagi and Khanti-Akom (1990) find that the set of instruments proposed by Breusch et al. (1989) may contain some remaining correlated variables so the outcomes may be biased. They suggest a reduced set of instruments based on successive tests that comprises the ones in Hausman and Taylor (1981) and Amemiya and MaCurdy (1986), and that is not affected by remaining correlations with the unobserved individual effects.

They also point out the order conditions for these three HT estimators to exist:

$$\begin{aligned}
\text{HTM} \quad &\Rightarrow \quad K + k_1 + j_1 \geq K + J \Rightarrow k_1 \geq j_2 \\
\text{HTAM} \quad &\Rightarrow \quad K + Tk_1 + j_1 \geq K + J \Rightarrow Tk_1 \geq j_2 \\
\text{HTBMS} \quad &\Rightarrow \quad K + Tk_1 + (T-1)k_2 + j_1 \geq K + J \Rightarrow Tk_1 + (T-1)k_2 \geq j_2
\end{aligned}$$

where K and J are the number of dimension-varying and dimension-invariant regressors, respectively; and the subindex 1 indicates the set of doubly-exogenous variables, while 2 indicates the set of singly-exogenous variables, so $K = k_1 + k_2$, and $J = j_1 + j_2$. As can be seen, incorporating more instruments relaxes the order condition. Given that under our approach, the system estimator in [1.11] is an extension of the equation-by-equation estimator given by [1.10], the same conditions apply for each equation in the system. However, it is not possible to compensate the order conditions between equations. Efficiency ordering of these estimators is easy to figure out given that under the consistency assumptions, those estimators incorporating more instruments are at least as efficient as those with fewer instruments (see Breusch et al., 1989).

1.F.6 G3SPD-type Estimators

Given that G3SPD is a procedure that integrates three estimators on it, we describe them as incorporated in the method though they can be obtained directly from previous specifications. In this chapter we focus on five cases of G3SPD, and the suitability of each of them depends on the underlying assumptions about the endogeneity of the regressors. From a bias point of view, there are two general estimators: one that does not incorporate simultaneity or endogeneity considerations, and one based on a simultaneity-corrected approach. Although we are not interested in efficiency considerations here, we present four cases of the latter procedure using the sets of instruments developed for ECC, HTM, HTAM, and HTBMS.

G3SPD without Endogeneity Considerations Here we use no instruments at all, obtaining a FEStd estimator in the first step, a BEStd estimator in the second step, and a Pooled OLS estimator in the third step, which uses the pseudo-effects as regressors together with the dimension-invariant ones. To obtain these estimators, set:

$$\begin{aligned}
&1^{st}\text{ Step:} \quad R_g = \widetilde{R_g} \quad ; \quad W_g = Q_{\mu_g} \quad ; \quad P_{[A_g,B_g]} = I_{NT} \\
&2^{nd}\text{ Step:} \quad R_g = R_g^{OLS} \quad ; \quad W_g = P_{\mu_g} \quad ; \quad P_{[A_g,B_g]} = I_{NT} \qquad (1.29)\\
&3^{rd}\text{ Step:} \quad R_g = R_g^{*} \quad ; \quad W_g = I_{NT} \quad ; \quad P_{[A_g,B_g]} = I_{NT}
\end{aligned}$$

where $R_g^* = \left[Y_g, X_g^{(1)}, X_g^{(2)}, Z_g^{(1)}, Z_g^{(2)}, [Z_{\mu_g}\hat{\mu}_g], \iota_{NT}\right]$ is the "augmented" set of regressors for the third step. The covariance estimate for the system estimator in the 3^{rd} Step is obtained as $\check{\Sigma} = (\tilde{\varepsilon}'\tilde{\varepsilon})/NT$ where $\check{\varepsilon} = y - R\hat{\theta}$ is the set of residuals from a equation-by-equation estimation, arranged in a $NT \times G$ matrix $\tilde{\varepsilon} = [\check{\varepsilon}_1, \check{\varepsilon}_2, \ldots, \check{\varepsilon}_G]$. As can be seen, this covariance estimate coincides with the covariance matrix from the Pooled OLS previously described given that in the 3^{rd} Step we are running a plain pooled OLS regression. We call this estimator the G3SPD-Std.

G3SPD with Corrections for Endogeneity Consider now incorporating different sets of instruments to the estimation to obtain four estimators, namely G3SPD, G3SPD-HTM, G3SPD-AM, and G3SPD-BMS, resembling ECC, HTM, HTAM, and HTBMS. The general setup is similar to [1.29] though differing in the set of instruments: for the 1^{st} Step, all estimators use $A_g = \widetilde{A_g}$, while for the 2^{nd} Step they vary according to:

$$\begin{aligned}
&\text{G3SPD:} \quad B_g = \overline{B_g} \qquad &&\text{G3SPD-HT:} \quad B_g = B_g^{HT} \\
&\text{G3SPD-AM:} \quad B_g = B_g^{AM} \qquad &&\text{G3SPD-BMS:} \quad B_g = B_g^{BMS}
\end{aligned}$$

with the matrices defined above. The set of instruments for the 3^{rd} Step is given by:

$$C_g = \left[X_g^{(1)}, X_g^{(2)}, X_g^{(3)}, Z_g^{(1)}, Z_g^{(2)}, Z_g^{(3)}, [Z_{\mu_g}\hat{\mu}_g], \iota_{NT}\right],$$

This is possible because now that we have included $[Z_{\mu_g}\hat{\mu}_g]$ in the regressors set, $X_g^{(2)}$ and $Z_g^{(2)}$ become doubly exogenous. In this case, G3SPD, G3SPD-HTM, G3SPD-AM, and

G3SPD-BMS should yield the same because there is no additional gains from improving the set of instruments to control the endogeneity caused by the correlation of the regressors with the unobservable effects. Y_g is not included as an instruments because it remains correlated with ε_g. Finally, the inclusion of the pseudo-effects as a new instrument implies an additional gain from the endogeneity correction point of view that favours the consistency of the coefficient of Y_g.

1.G Figures and Tables

Figure 1.- Fixed-Effects Case: bias performance of the estimator for the coefficient of an endogenous variable (Y_2) in Equation 1.

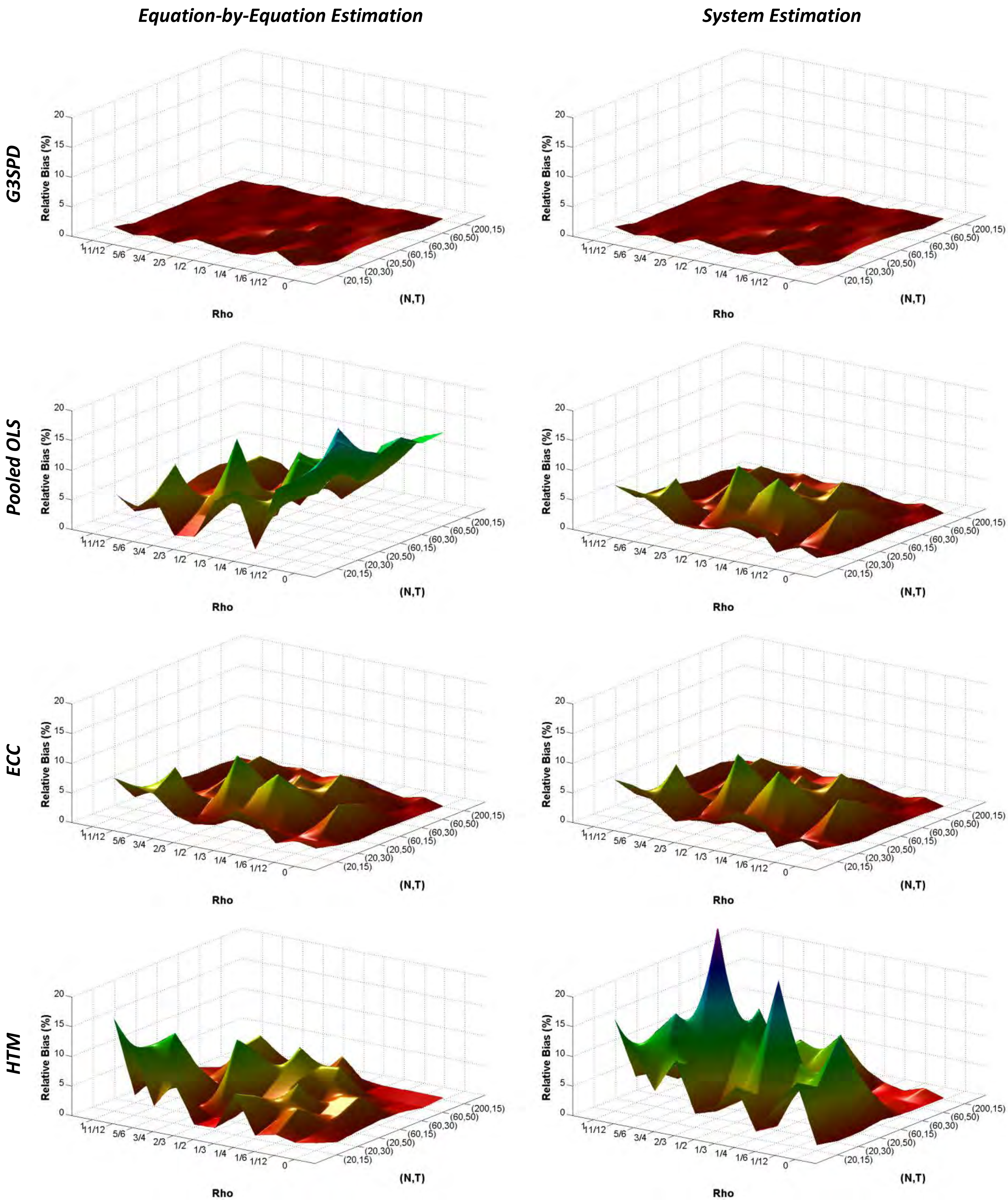


Figure 2.- Random-Effects Case: bias performance of the estimator for the coefficient of an endogenous variable (Y_2) in Equation 1.

Equation-by-Equation Estimation | *System Estimation*

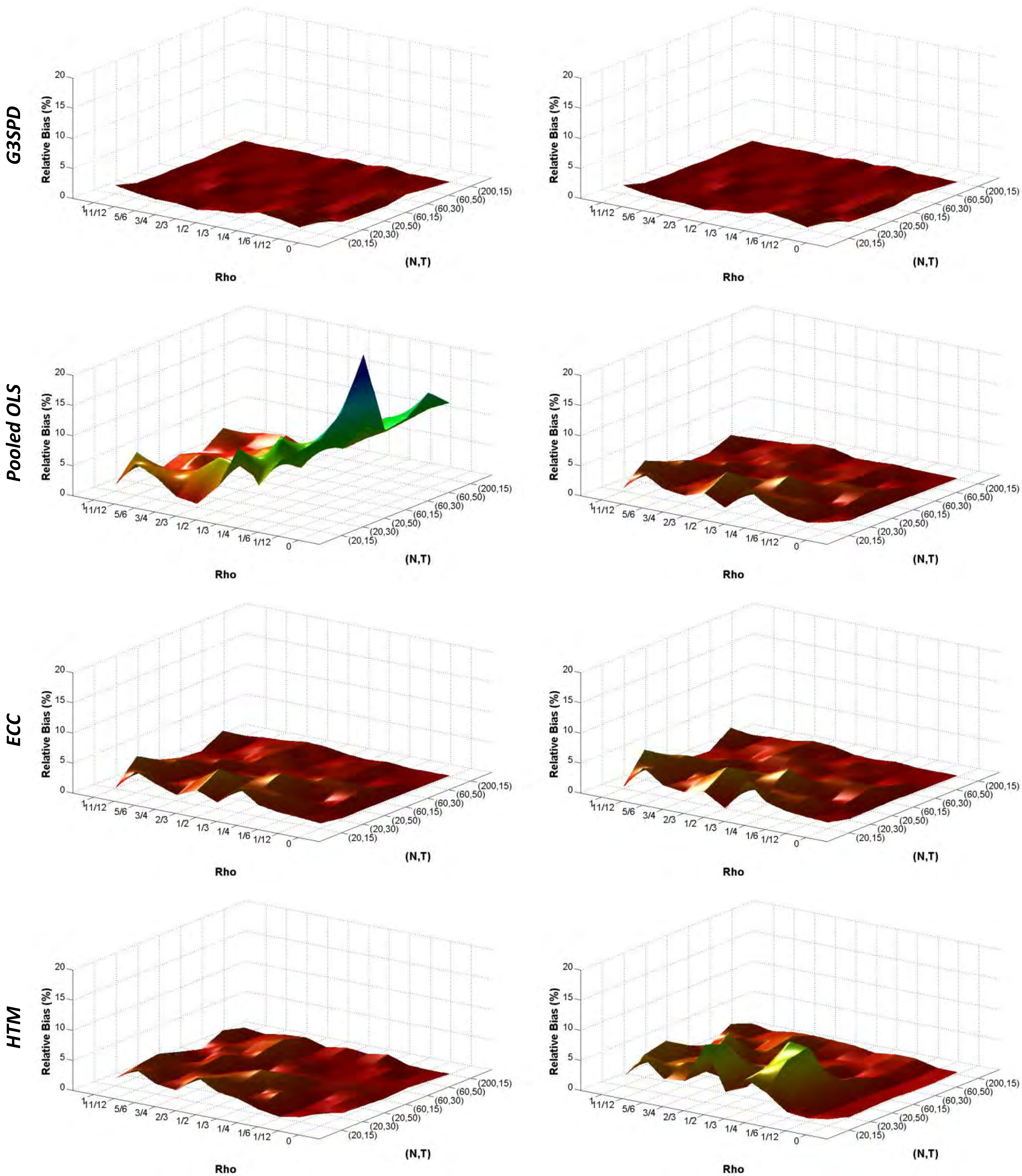


Figure 3.- Fixed-Effects Case: bias performance of the estimator for the coefficient of a singly-exogenous time-invariant variable (Z_2) in Equation 1.

Equation-by-Equation Estimation | *System Estimation*

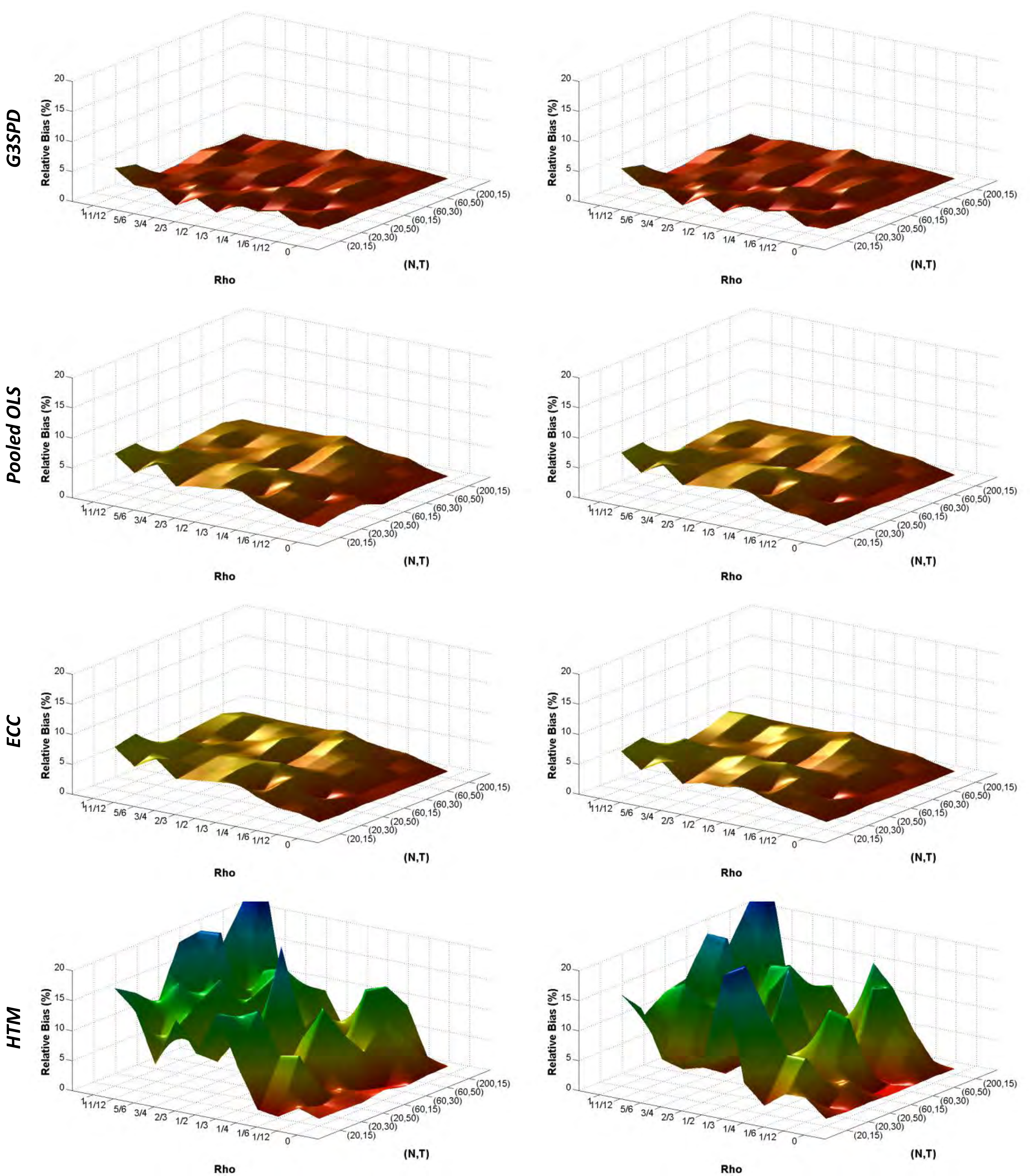


Figure 4.- Random-Effects Case: bias performance of the estimator for the coefficient of a singly-exogenous time-invariant variable (Z_2) in Equation 1.

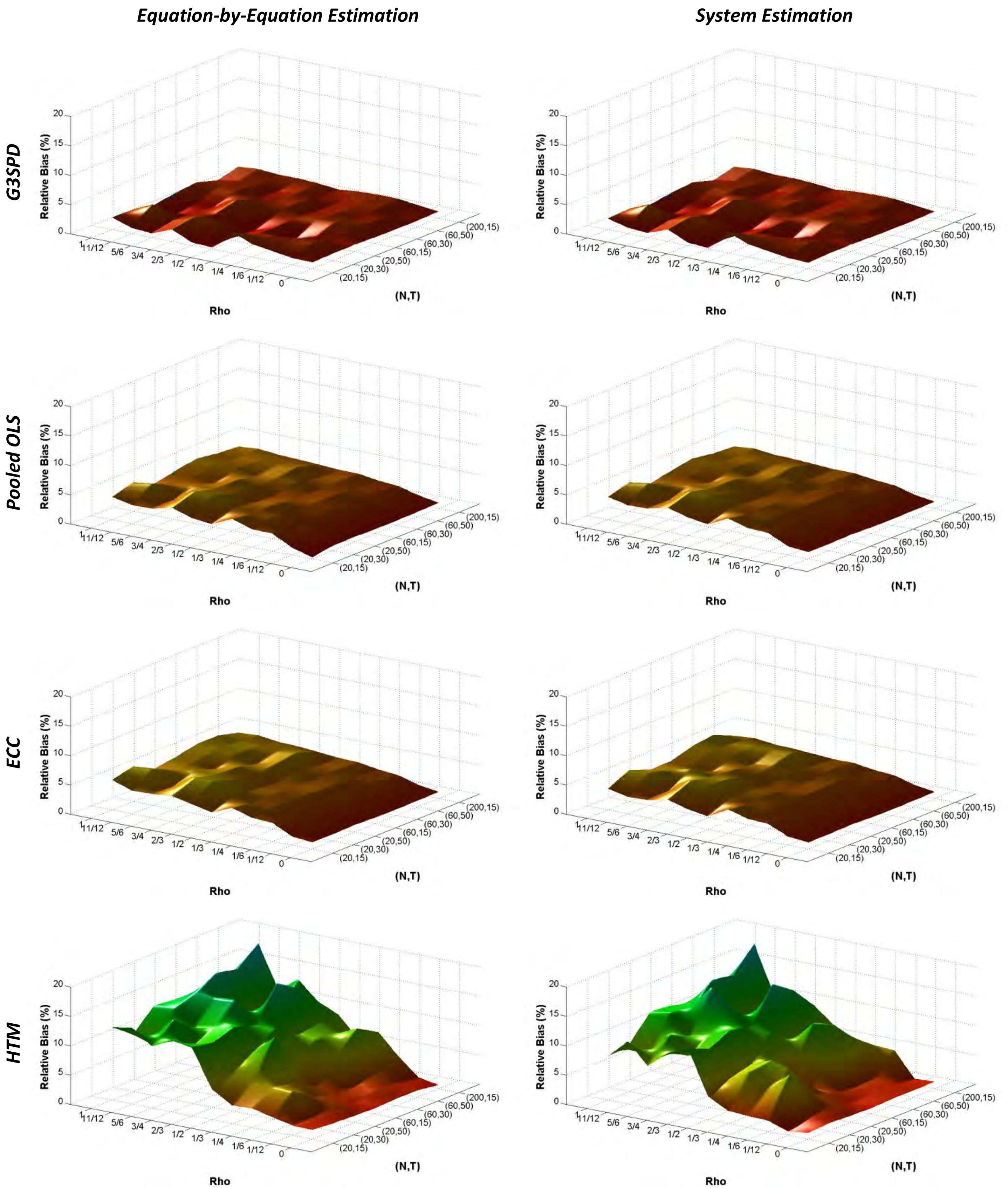


Figure 5.- Fixed-Effects Case: bias performance of the estimator for the coefficient of a singly-exogenous dimension-varying variable (X_3) in Equation 1.

Equation-by-Equation Estimation *System Estimation*

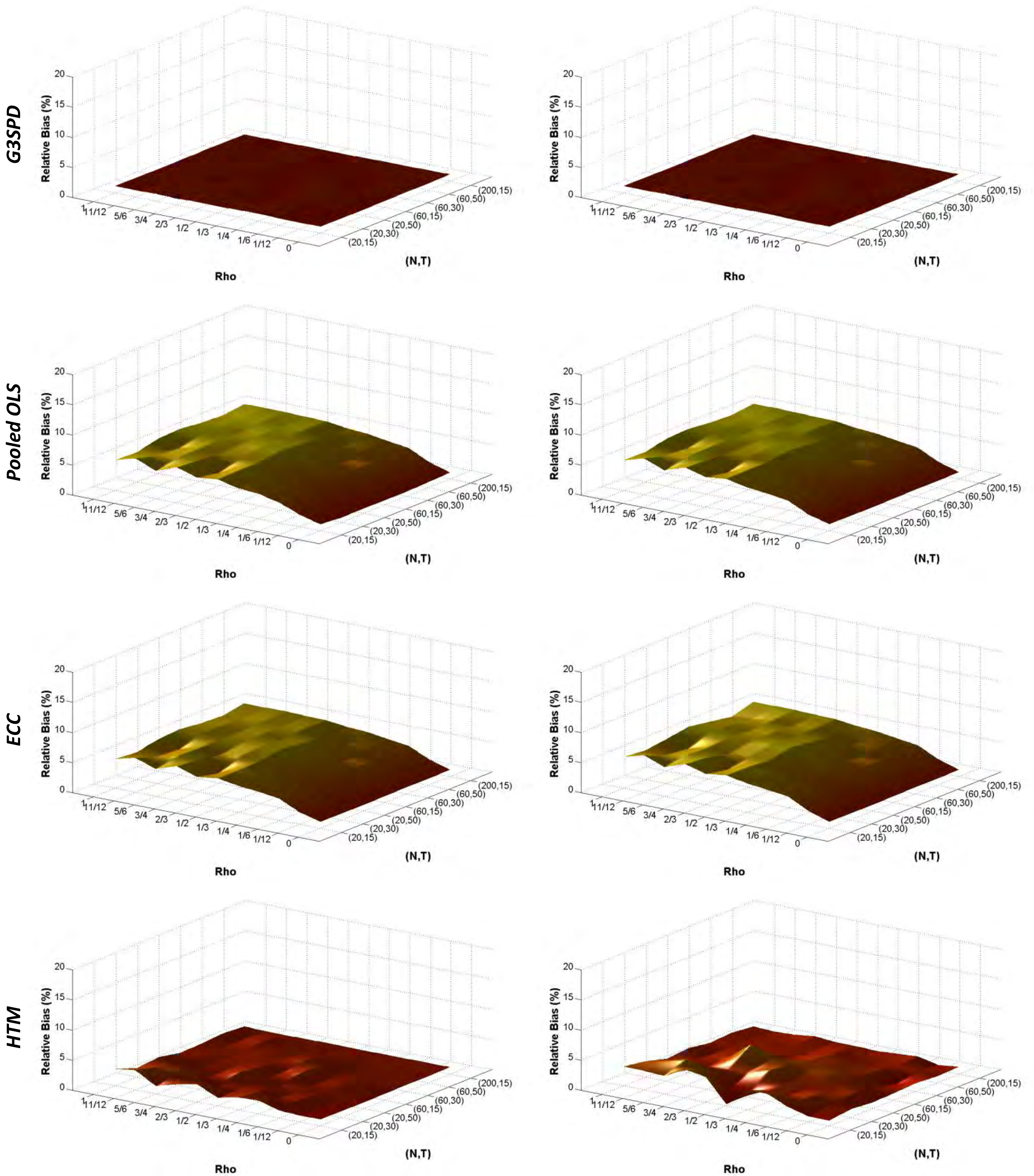


Figure 6.- Random-Effects Case: bias performance of the estimator for the coefficient of a singly-exogenous dimension-varying variable (X_3) in Equation 1.

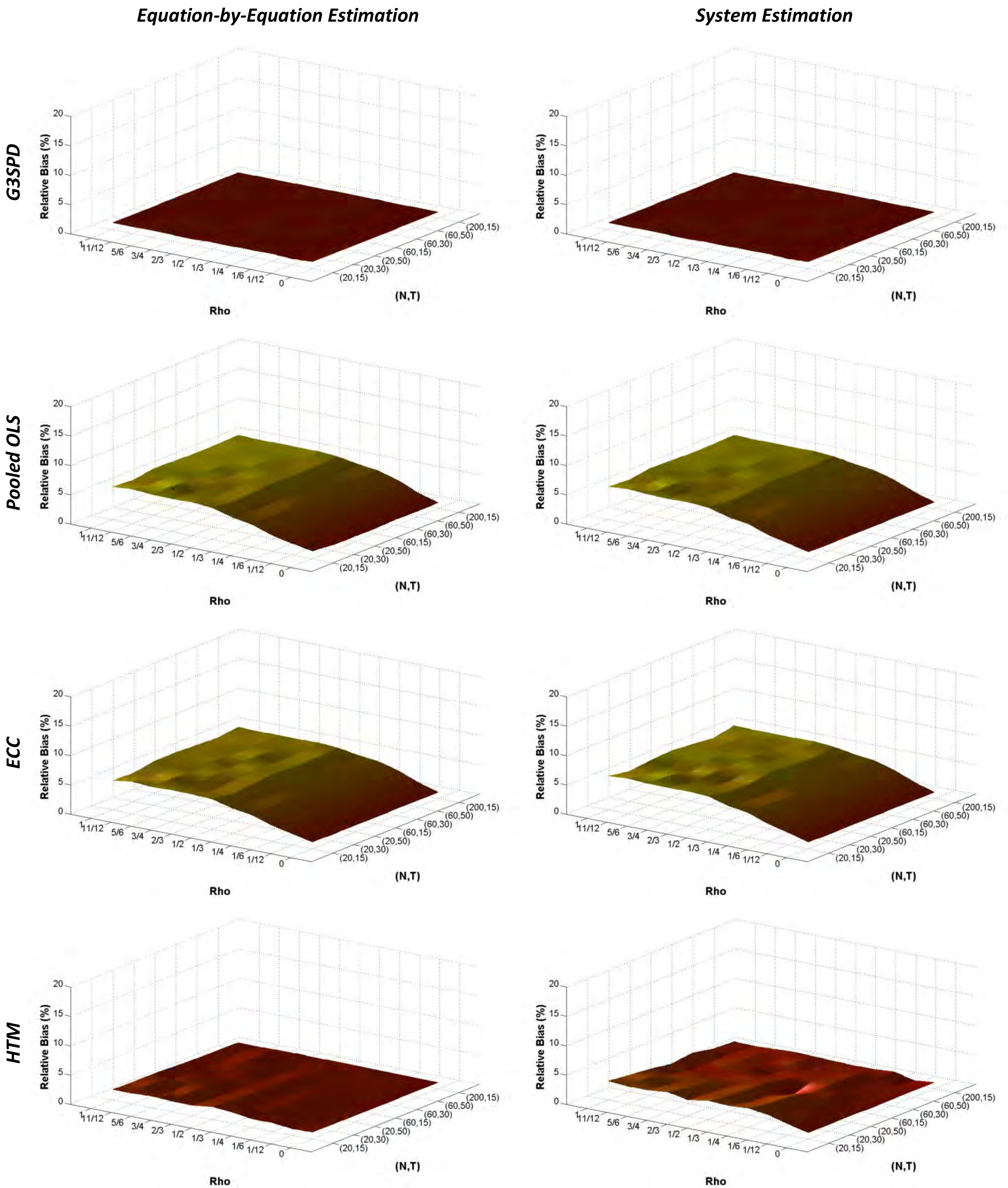


Table 1. A gravity model of intra-EU trade: estimation results for 15 EU members, 1960-2001.

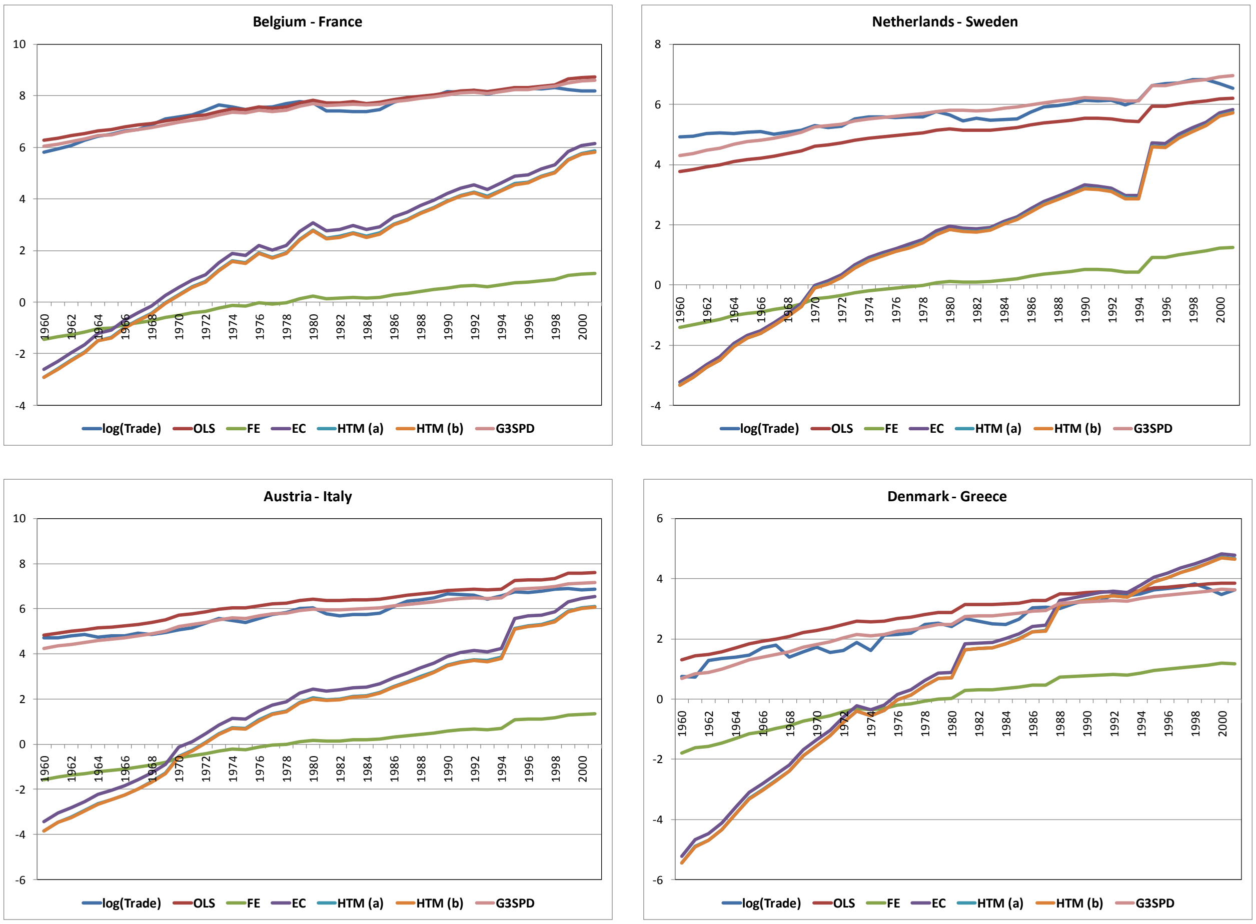
(whithout controlling for time effects)

	OLS [1]	FE [2]	BE [3]	EC [4]	HTM (a) [5]	HTM (b) [6]	**G3SPD [7]**
RER	0,099 *	0,061 *	0,087 *	0,069 *	0,061 *	0,063 *	**0,061 ***
	(0,004)	(0,009)	(0,023)	(0,008)	(0,009)	(0,009)	**(0,009)**
GDP	1,579 *	1,812 *	1,541 *	1,795 *	1,812 *	1,805 *	**1,812 ***
	(0,013)	(0,020)	(0,080)	(0,018)	(0,020)	(0,019)	**(0,021)**
RLF	0,032 *	0,033 *	0,011	0,033 *	0,033 *	0,033 *	**0,033 ***
	(0,008)	(0,008)	(0,060)	(0,008)	(0,008)	(0,008)	**(0,009)**
SIM	0,885 *	1,172 *	0,806 *	1,143 *	1,172 *	1,198 *	**1,172 ***
	(0,017)	(0,056)	(0,104)	(0,046)	(0,056)	(0,053)	**(0,056)**
CEE	0,318 *	0,309 *	-0,030	0,318 *	0,309 *	0,311 *	**0,309 ***
	(0,023)	(0,016)	(0,244)	(0,016)	(0,016)	(0,016)	**(0,017)**
EMU	0,204 *	0,085 *	-1,596	0,093 *	0,085 *	0,086 *	**0,085 ***
	(0,051)	(0,027)	(1,842)	(0,027)	(0,027)	(0,027)	**(0,027)**
DIST	-0,646 *		-0,710 *	-0,591 *	-0,382 **	-0,378 **	**-0,710 ***
	(0,022)		(0,136)	(0,117)	(0,192)	(0,188)	**(0,130)**
BOR	0,525 *		0,609 *	0,441 **	0,601 **	0,610 **	**0,609 ***
	(0,034)		(0,201)	(0,190)	(0,259)	(0,259)	**(0,193)**
LAN	0,234 *		0,236	0,417 **	1,559 **	1,558 **	**0,236**
	(0,034)		(0,201)	(0,185)	(0,707)	(0,666)	**(0,193)**
Const	-10,947 *		-9,706 *	-13,930 *	-15,760 *	-15,656 *	**-9,706 ***
	(0,247)		(1,552)	(0,889)	(1,503)	(1,475)	**(1,485)**
MuHat							**1,000 ***
							(0,009)

Notes: Left-hand-side variable is the logarithm of real total trade flows. The regressors DIST, BOR, and LAN are time-invariant regressors. MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by G3SPD. Hausman-Taylor estimates are obtained under two assumptions of exogeneity of regressors: (a) only RER is doubly-exogenous; (b) RER, GDP, and RLF are assumed doubly-exogenous. In all cases, DIST and BOR are considered doubly-exogenous while LAN is singly-exogenous. The dataset corresponds to Serlenga and Shin (2007) and is available from the *Journal of Applied Econometrics* web site. Sample: 91 country-pairs throughout 42 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Figure 7. Bilateral trade flows for selected EU country-pairs, 1960-2001.

(actual and fitted data based on estimates from Table 1)



Notes: All scales in logs. Fitted values corresponding to estimations in Table 1. For more details, see the notes in Table 1. Countries have been selected in order to account for common and non-common border, and common and non-common language. Actual data comes from Serlenga and Shin (2007) and is available from the *Journal of Applied Econometrics* web site.

Table 2. A gravity model of intra-EU trade: estimation results for 15 EU members, 1960-2001.

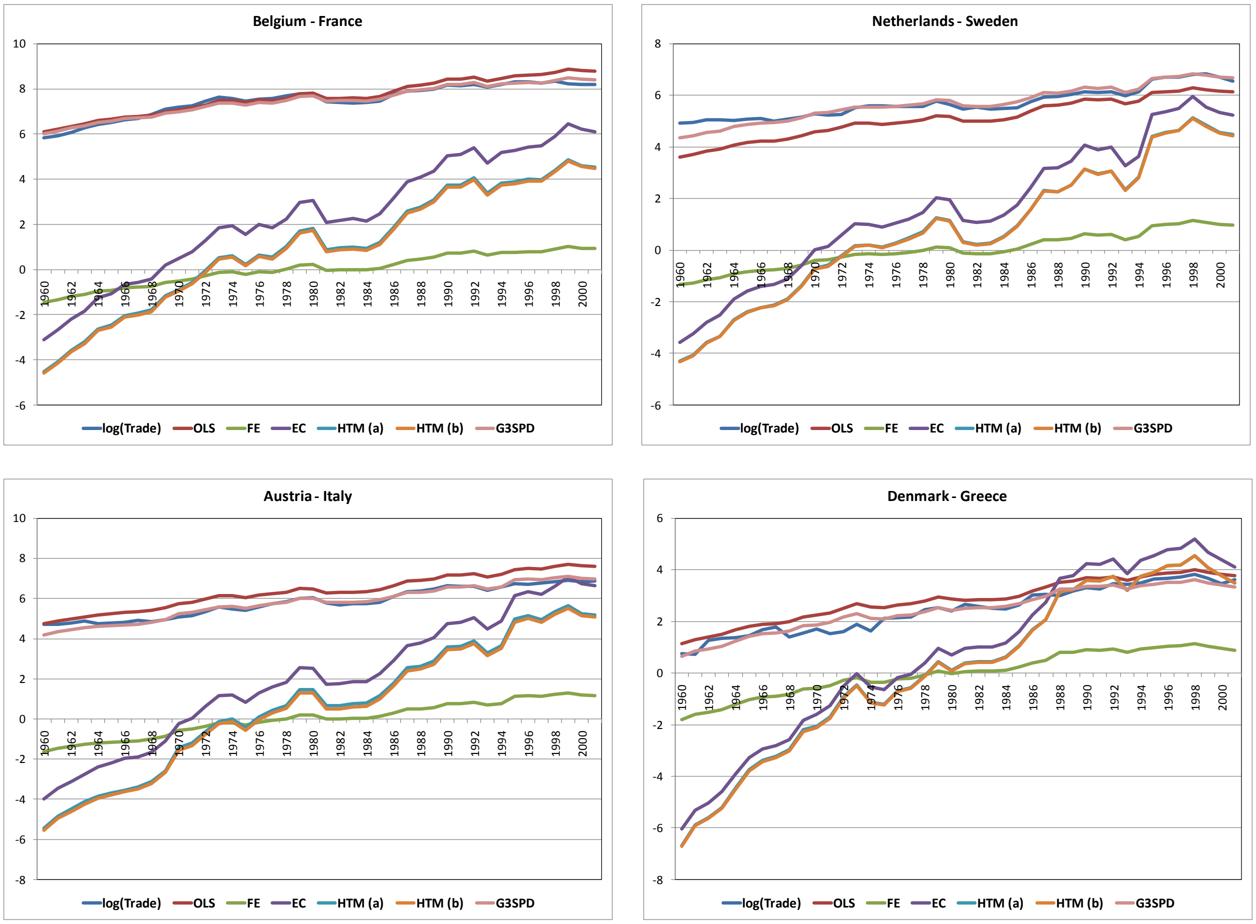
(controlling for time effects)

	OLS [1]	FE [2]	BE [3]	EC [4]	HTM (a) [5]	HTM (b) [6]	**G3SPD [7]**
RER	0,088 *	0,084 *	0,087 *	0,056 *	0,084 *	0,077 *	**0,084 ***
	(0,004)	(0,010)	(0,023)	(0,009)	(0,010)	(0,010)	**(0,010)**
GDP	1,538 *	3,053 *	1,541 *	2,224 *	3,053 *	2,927 *	**3,053 ***
	(0,013)	(0,079)	(0,080)	(0,054)	(0,079)	(0,075)	**(0,080)**
RLF	0,021 **	0,018 **	0,011	0,024 *	0,018 **	0,019 *	**0,018 ****
	(0,008)	(0,007)	(0,060)	(0,007)	(0,007)	(0,007)	**(0,008)**
SIM	0,839 *	1,422 *	0,806 *	1,279 *	1,422 *	1,417 *	**1,422 ***
	(0,017)	(0,055)	(0,104)	(0,050)	(0,055)	(0,055)	**(0,056)**
CEE	0,167 *	0,319 *	-0,030	0,305 *	0,319 *	0,315 *	**0,319 ***
	(0,026)	(0,017)	(0,244)	(0,017)	(0,017)	(0,017)	**(0,018)**
EMU	0,210 *	0,218 *	-1,596	0,274 *	0,218 *	0,226 *	**0,218 ***
	(0,070)	(0,034)	(1,842)	(0,035)	(0,034)	(0,034)	**(0,035)**
DIST	-0,698 *		-0,710 *	-0,439 *	0,377	0,612	**-0,710 ***
	(0,022)		(0,136)	(0,123)	(0,433)	(0,505)	**(0,130)**
BOR	0,536 *		0,609 *	0,277	0,391	0,665	**0,609 ***
	(0,033)		(0,201)	(0,196)	(0,586)	(0,698)	**(0,193)**
LAN	0,260 *		0,236	0,655 *	3,275 **	4,757 *	**0,236**
	(0,034)		(0,201)	(0,190)	(1,532)	(1,708)	**(0,193)**
Const	-10,202 *		-9,706 *	-20,188 *	-36,481 *	-36,787 *	**-8,715 ***
	(0,257)		(1,552)	(1,209)	(3,476)	(4,000)	**(1,485)**
MuHat							**1,000 ***
							(0,008)

Notes: Left-hand-side variable is the logarithm of real total trade flows. The regressors DIST, BOR, and LAN are time-invariant regressors. MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by G3SPD. Hausman-Taylor estimates are obtained under two assumptions of exogeneity of regressors: (a) only RER is doubly-exogenous; (b) RER, GDP, and RLF are assumed doubly-exogenous. In all cases, DIST and BOR are considered doubly-exogenous while LAN is singly-exogenous. The regresison includes 41 time-dummies, though they are not reported. The dataset corresponds to Serlenga and Shin (2007) and is available from the *Journal of Applied Econometrics* web site. Sample: 91 country-pairs throughout 42 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Figure 8. Bilateral trade flows for selected EU country-pairs, 1960-2001.

(actual and fitted data based on estimates from Table 2)



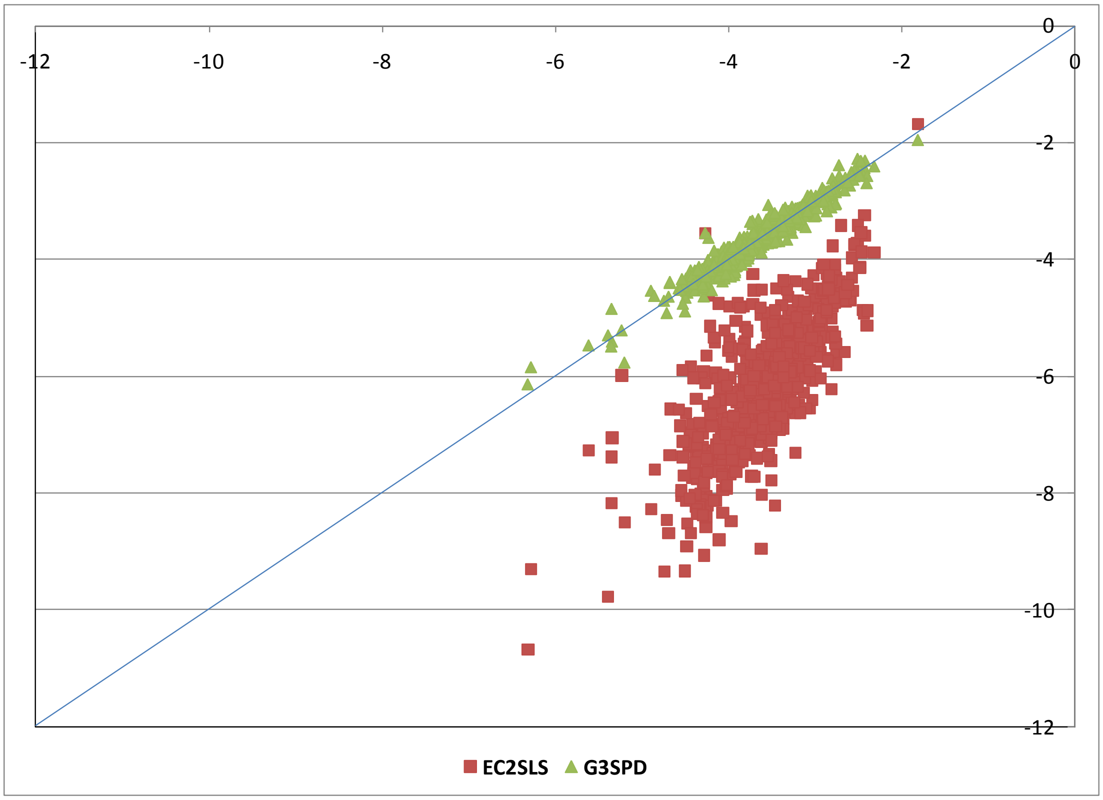
Notes: All scales in logs. Fitted values corresponding to estimations in Table 2. For more details, see the notes in Table 2. Countries have been selected in order to account for common and non-common border, and common and non-common language. Actual data comes from Serlenga and Shin (2007) and is available from the *Journal of Applied Econometrics* web site.

Table 3. Economics of Crime: estimates for North Carolina, 1981-1987.

	OLS	FE	FE 2SLS	BE	BE 2SLS	EC 2SLS	G3SPD
	[1]	[2]	[3]	[4]	[5]	[6]	[7]
LPRBARR	-0.537 *	-0.355 *	-0.576	-0.648 *	-0.503 **	-0.413 *	**-0.576**
	(0.030)	(0.032)	(0.802)	(0.088)	(0.241)	(0.097)	**(0.803)**
LPOLPC	0.360 *	0.413 *	0.658	0.364 *	0.408 **	0.435 *	**0.658**
	(0.022)	(0.027)	(0.847)	(0.060)	(0.193)	(0.090)	**(0.847)**
LPRBCONV	-0.431 *	-0.282 *	-0.423	-0.528 *	-0.525 *	-0.323 *	**-0.423**
	(0.022)	(0.021)	(0.502)	(0.067)	(0.100)	(0.054)	**(0.502)**
LPRBPRIS	-0.116 **	-0.173 *	-0.250	0.297	0.187	-0.186 *	**-0.250**
	(0.048)	(0.032)	(0.279)	(0.231)	(0.318)	(0.042)	**(0.280)**
LAVGSEN	-0.091 **	-0.002	0.009	-0.236	-0.227	-0.010	**0.009**
	(0.040)	(0.026)	(0.049)	(0.174)	(0.179)	(0.027)	**(0.050)**
LDENSITY	0.299 *	0.414	0.139	0.168 **	0.226 **	0.429 *	**0.139**
	(0.028)	(0.283)	(1.021)	(0.077)	(0.102)	(0.055)	**(1.021)**
LPCTYMLE	-0.145 **	0.627 ***	0.351	-0.095	-0.095	-0.108	**0.351**
	(0.063)	(0.364)	(1.011)	(0.158)	(0.192)	(0.140)	**(1.012)**
LWCON	0.085	-0.038	-0.029	0.195	0.314	-0.007	**-0.029**
	(0.055)	(0.039)	(0.054)	(0.210)	(0.259)	(0.040)	**(0.055)**
LWTUC	0.022	0.046 **	0.039	-0.196	-0.199	0.045 **	**0.039**
	(0.031)	(0.019)	(0.031)	(0.170)	(0.197)	(0.020)	**(0.031)**
LWTRD	0.043	-0.021	-0.018	0.129	0.054	-0.008	**-0.018**
	(0.062)	(0.040)	(0.045)	(0.278)	(0.296)	(0.041)	**(0.047)**
LWFIR	-0.002	-0.004	-0.009	0.113	0.042	-0.004	**-0.009**
	(0.045)	(0.028)	(0.037)	(0.220)	(0.306)	(0.029)	**(0.037)**
LWSER	-0.027	0.009	0.019	-0.106	-0.135	0.006	**0.019**
	(0.031)	(0.019)	(0.039)	(0.163)	(0.174)	(0.020)	**(0.039)**
LWMFG	-0.079	-0.360 *	-0.243	-0.025	-0.042	-0.204 **	**-0.243**
	(0.053)	(0.112)	(0.420)	(0.134)	(0.156)	(0.080)	**(0.420)**
LWFED	0.084	-0.309 ***	-0.451	0.156	0.148	-0.164	**-0.451**
	(0.114)	(0.176)	(0.527)	(0.287)	(0.326)	(0.159)	**(0.530)**
LWSTA	-0.202 **	0.053	-0.019	-0.284	-0.203	-0.054	**-0.019**
	(0.093)	(0.114)	(0.281)	(0.256)	(0.298)	(0.106)	**(0.284)**
LWLOC	0.055	0.182	0.263	0.010	0.044	0.163	**0.263**
	(0.140)	(0.118)	(0.312)	(0.463)	(0.494)	(0.120)	**(0.316)**
d1982	-0.001	0.023	0.038			0.011	**0.038**
	(0.040)	(0.026)	(0.062)			(0.026)	**(0.061)**
d1983	-0.085 ***	-0.041	-0.044			-0.084 *	**-0.044**
	(0.044)	(0.036)	(0.042)			(0.031)	**(0.042)**
d1984	-0.124 *	-0.043	-0.045			-0.103 *	**-0.045**
	(0.046)	(0.046)	(0.055)			(0.037)	**(0.055)**
d1985	-0.111 **	-0.017	-0.021			-0.096 ***	**-0.021**
	(0.055)	(0.063)	(0.074)			(0.049)	**(0.075)**
d1986	-0.084	0.035	0.006			-0.069	**0.006**
	(0.063)	(0.078)	(0.128)			(0.060)	**(0.129)**
d1987	-0.050	0.100	0.044			-0.031	**0.044**
	(0.071)	(0.093)	(0.216)			(0.071)	**(0.217)**
WEST	-0.229 *			-0.230 **	-0.205 ***	-0.227 **	**-0.205**
	(0.046)			(0.108)	(0.114)	(0.100)	**(0.175)**
CENTRAL	-0.178 *			-0.164 **	-0.173 **	-0.194 *	**-0.173** ***
	(0.028)			(0.064)	(0.067)	(0.060)	**(0.103)**
URBAN	-0.136 **			-0.035	-0.080	-0.225 ***	**-0.080**
	(0.054)			(0.132)	(0.144)	(0.116)	**(0.214)**
LPCTMIN	0.179 *			0.148 *	0.169 *	0.189 *	**0.169** **
	(0.020)			(0.049)	(0.053)	(0.041)	**(0.080)**
Const	-2.721 *			-2.097	-1.977	-0.954	**-1.974**
	(0.966)			(2.822)	(4.001)	(1.284)	**(5.246)**
MuHat							**1.000** *
							(0.033)

Notes: Left-hand-side variable is the logarithm of the county crime rate. The regressors WEST, CENTRAL, URBAN, AND LPCTMIN are time-invariant regressors. The regressors d1982-1987 represent time dummies. The MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by G3SPD. 2SLS estimators instrumentaliza LPRBARR AND LPOLPC. It is assumed that none of the regressors is singly-exogenous. The dataset corresponds to Baltagi (2006) and is available from the *Journal of Applied Econometrics* web site. Sample: 90 counties throughout 7 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Figure 9. Economics of Crime: fitted values for North Carolina, 1981-1987.



Notes: Fitted values for the logarithm of county crime rate as obtained from the estimations reported in Table 3. The blue line corresponds to the 45-degrees line. Sample: 90 counties throughout 7 years.

Table 4. Wage determinants and returns to schooling: estimates from a subsample of the PSID, 1976-1982.

(specification by Cornwell and Rupert, 1988, without controlling for time effects)

	OLS [1]	FE [2]	BE [3]	GLS [4]	EC [5]	**HTM [6]**	HTAM [7]	HTBMS (a) [8]	HTBMS (b) [9]	**G3SPD [10]**
WKS	0.004 *	0.001	0.009 **	0.001	0.001	**0.001**	0.001	0.001	0.001	**0.001**
	(0.001)	(0.001)	(0.004)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	***(0.001)***
SOUTH	-0.056 *	-0.002	-0.057 **	0.004	-0.017	**0.007**	0.009	0.012	0.010	**-0.002**
	(0.013)	(0.034)	(0.026)	(0.032)	(0.027)	**(0.033)**	(0.032)	(0.032)	(0.032)	***(0.032)***
SMSA	0.152 *	-0.042 **	0.176 *	-0.046 **	-0.014	**-0.042 ****	-0.043 **	-0.045 **	-0.044 **	**-0.042 ****
	(0.012)	(0.019)	(0.026)	(0.019)	(0.020)	**(0.019)**	(0.019)	(0.019)	(0.019)	***(0.019)***
MS	0.048 **	-0.030	0.115 **	-0.038 **	-0.075 *	**-0.036 *****	-0.036 ***	-0.036 ***	-0.036 ***	**-0.030**
	(0.021)	(0.019)	(0.048)	(0.019)	(0.023)	**(0.019)**	(0.019)	(0.019)	(0.019)	***(0.018)***
EXP	0.040 *	0.113 *	0.032 *	0.109 *	0.082 *	**0.113 ***	0.113 *	0.113 *	0.113 *	**0.113 ***
	(0.002)	(0.002)	(0.005)	(0.002)	(0.003)	**(0.002)**	(0.002)	(0.002)	(0.002)	***(0.003)***
EXP2	-0.001 *	0.000 *	-0.001 *	0.000 *	-0.001 *	**0.000 ***	0.000 *	0.000 *	0.000 *	**0.000 ***
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	***(0.000)***
OCC	-0.140 *	-0.021	-0.168 *	-0.024 ***	-0.050 *	**-0.021**	-0.022	-0.022	-0.022	**-0.021**
	(0.015)	(0.014)	(0.034)	(0.013)	(0.017)	**(0.014)**	(0.014)	(0.014)	(0.014)	***(0.014)***
IND	0.047 *	0.019	0.058 **	0.015	0.004	**0.019**	0.018	0.018	0.018	**0.019**
	(0.012)	(0.015)	(0.026)	(0.015)	(0.017)	**(0.015)**	(0.015)	(0.015)	(0.015)	***(0.015)***
UNION	0.093 *	0.033 **	0.109 *	0.037 **	0.063 *	**0.030 ****	0.030 **	0.030 **	0.030 **	**0.033 ****
	(0.013)	(0.015)	(0.029)	(0.015)	(0.017)	**(0.015)**	(0.015)	(0.015)	(0.015)	***(0.014)***
FEM	-0.368 *		-0.317 *	-0.161	-0.339 *	**-0.137**	-0.141	-0.149	-0.145	**-0.317 ***
	(0.025)		(0.055)	(0.136)	(0.051)	**(0.127)**	(0.127)	(0.128)	(0.127)	***(0.028)***
BLK	-0.167 *		-0.158 *	-0.266	-0.210 *	**-0.282**	-0.261	-0.215	-0.239	**-0.158 ***
	(0.022)		(0.045)	(0.166)	(0.058)	**(0.177)**	(0.166)	(0.163)	(0.163)	***(0.024)***
ED	0.057 *		0.051 *	0.138 *	0.100 *	**0.141 ****	0.155 *	0.190 *	0.172 *	**0.051 ***
	(0.003)		(0.006)	(0.015)	(0.006)	**(0.066)**	(0.048)	(0.040)	(0.042)	***(0.003)***
Const	5.251 *		5.121 *	3.030 *	4.264 *	**2.884 ***	2.702 *	2.259 *	2.492 *	**5.121 ***
	(0.071)		(0.204)	(0.209)	(0.098)	**(0.853)**	(0.628)	(0.516)	(0.548)	***(0.093)***
MuHat										**1.000 ***
										(0.006)

Notes: Left-hand-side variable is the logarithm of wage. The regressors FEM, BLK, and ED are time-invariant regressors. The MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by the G3SPD estimator. Hausman-Taylor estimates with the Breusch, Mizon and Schmidt's (1989) instruments (HTBMS) are obtained under two assumptions of exogeneity of regressors: (a) UNION is doubly-exogenous; (b) UNION is singly-exogenous. In all cases, FEM and BLK are considered doubly-exogenous while ED is singly-exogenous. The dataset corresponds to Baltagi and Khanti-Akom (1990) and is available from the *Journal of Applied Econometrics* web site. Sample: 595 individuals throughout 7 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Table 5. Wage determinants and returns to schooling: estimates from a subsample of the PSID, 1976-1982.

(specification by Cornwell and Rupert, 1988, controlling for time effects)

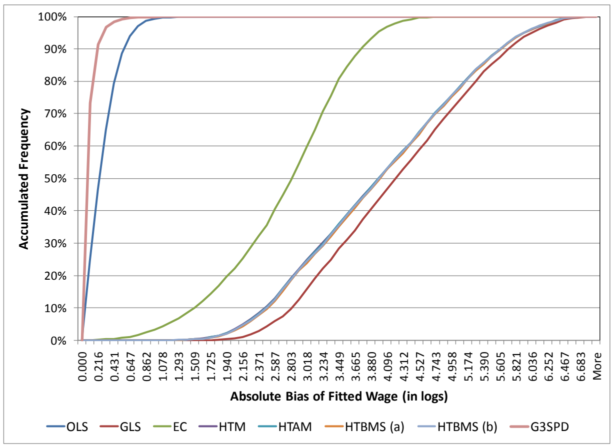
	OLS	FE	BE	GLS	EC	**HTM**	HTAM	HTBMS (a)	HTBMS (b)	**G3SPD**
	[1]	[2]	[3]	[4]	[5]	**[6]**	[7]	[8]	[9]	**[10]**
WKS	0.004 *	0.001	0.009 **	0.001	0.001 ***	0.001	0.001	0.001	0.001	0.001
	(0.001)	(0.001)	(0.004)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
SOUTH	-0.060 *	0.003	-0.057 **	-0.009	-0.058 *	0.011	0.008	0.008	0.007	0.003
	(0.011)	(0.034)	(0.026)	(0.030)	(0.021)	(0.032)	(0.031)	(0.031)	(0.031)	(0.032)
SMSA	0.165 *	-0.042 **	0.176 *	-0.029 ***	0.047 *	-0.040 **	-0.039 **	-0.039 **	-0.038 **	-0.042 **
	(0.011)	(0.019)	(0.026)	(0.018)	(0.016)	(0.019)	(0.019)	(0.018)	(0.018)	(0.019)
MS	0.081 *	-0.029	0.115 **	-0.031 ***	-0.018	-0.034 ***	-0.033 ***	-0.033 ***	-0.033 ***	-0.029
	(0.018)	(0.019)	(0.048)	(0.018)	(0.018)	(0.018)	(0.018)	(0.018)	(0.018)	(0.019)
EXP	0.032 *	0.104 *	0.032 *	0.039 *	0.030 *	0.091 *	0.081 *	0.078 *	0.079 *	0.104 *
	(0.002)	(0.009)	(0.005)	(0.004)	(0.002)	(0.008)	(0.007)	(0.007)	(0.007)	(0.011)
EXP2	-0.001 *	0.000 *	-0.001 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
OCC	-0.136 *	-0.019	-0.168 *	-0.022 ***	-0.044 *	-0.019	-0.019	-0.020	-0.020	-0.019
	(0.013)	(0.014)	(0.034)	(0.013)	(0.013)	(0.014)	(0.013)	(0.013)	(0.013)	(0.014)
IND	0.054 *	0.021	0.058 **	0.022	0.029 **	0.021	0.021	0.021	0.021	0.021
	(0.010)	(0.015)	(0.026)	(0.014)	(0.013)	(0.015)	(0.015)	(0.015)	(0.015)	(0.015)
UNION	0.085 *	0.030 **	0.109 *	0.029 **	0.041 *	0.027 ***	0.027 ***	0.026 ***	0.026 ***	0.030 **
	(0.011)	(0.015)	(0.029)	(0.014)	(0.013)	(0.015)	(0.014)	(0.014)	(0.014)	(0.014)
d1978	0.158 *	0.041 *		0.139 *	0.156 *	0.060 *	0.075 *	0.080 *	0.079 *	0.041 **
	(0.015)	(0.015)		(0.009)	(0.008)	(0.014)	(0.013)	(0.012)	(0.012)	(0.016)
d1979	0.246 *	0.052 **		0.215 *	0.243 *	0.083 *	0.108 *	0.117 *	0.115 *	0.052 **
	(0.015)	(0.023)		(0.011)	(0.008)	(0.021)	(0.019)	(0.018)	(0.018)	(0.025)
d1980	0.324 *	0.055 ***		0.283 *	0.321 *	0.099 *	0.134 *	0.147 *	0.143 *	0.055
	(0.015)	(0.032)		(0.013)	(0.009)	(0.028)	(0.025)	(0.024)	(0.025)	(0.034)
d1981	0.399 *	0.046		0.340 *	0.390 *	0.102 *	0.148 *	0.164 *	0.160 *	0.046
	(0.016)	(0.040)		(0.016)	(0.009)	(0.036)	(0.032)	(0.030)	(0.031)	(0.044)
d1982	0.482 *	0.046		0.406 *	0.468 *	0.115 *	0.171 *	0.190 *	0.185 *	0.046
	(0.016)	(0.049)		(0.019)	(0.010)	(0.043)	(0.039)	(0.037)	(0.038)	(0.053)
FEM	-0.353 *		-0.317 *	-0.389 *	-0.422 *	-0.217 ***	-0.252 **	-0.266 **	-0.262 **	-0.317 *
	(0.022)		(0.055)	(0.120)	(0.040)	(0.118)	(0.115)	(0.115)	(0.115)	(0.028)
BLK	-0.161 *		-0.158 *	-0.164	-0.153 *	-0.193	-0.178	-0.155	-0.168	-0.158 *
	(0.019)		(0.045)	(0.146)	(0.045)	(0.162)	(0.148)	(0.144)	(0.145)	(0.024)
ED	0.054 *		0.051 *	0.081 *	0.067 *	0.166 *	0.157 *	0.167 *	0.159 *	0.051 *
	(0.002)		(0.006)	(0.014)	(0.005)	(0.060)	(0.043)	(0.035)	(0.037)	(0.003)
Const	5.135 *		5.121 *	4.921 *	5.240 *	2.916 *	3.203 *	3.140 *	3.224 *	5.087 *
	(0.062)		(0.204)	(0.204)	(0.078)	(0.770)	(0.570)	(0.488)	(0.508)	(0.094)
MuHat										1.000 *
										(0.014)

Notes: Left-hand-side variable is the logarithm of wage. The regressors FEM, BLK, and ED are time-invariant regressors. The MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by the G3SPD estimator. Hausman-Taylor estimates with the Breusch, Mizon and Schmidt's (1989) instruments (HTBMS) are obtained under two assumptions of exogeneity of regressors: (a) UNION is doubly-exogenous; (b) UNION is singly-exogenous. In all cases, FEM and BLK are considered doubly-exogenous while ED is singly-exogenous. The regression includes 5 time-dummies, namely d1978-d1982, to control for time effects. The dataset corresponds to Baltagi and Khanti-Akom (1990) and is available from the *Journal of Applied Econometrics* web site. Sample: 595 individuals throughout 7 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

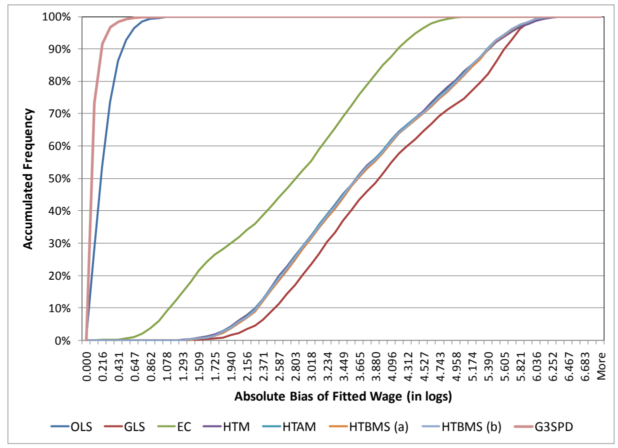
Figure 10. Wage determinants and returns to schooling: Cumulative distribution of the absolute bias of fitted models

(specification by Cornwell and Rupert, 1988)

Without controlling for time effects (fitted values based on estimates from Table 4)



Controlling for time effects (fitted values based on estimates from Table 5)



Notes: Fitted values corresponding to estimations in Table 4 and 5. For more details, see the corresponding notes. Frequencies computed as the number of observations in each bin. Sample: 595 individuals throughout 7 years.

Table 6. Wage determinants and returns to schooling: estimates from a subsample of the PSID, 1976-1982.

(specification by Baltagi and Khanti-Akom, 1990, without controlling for time effects)

	OLS [1]	FE [2]	BE [3]	GLS [4]	EC [5]	**HTM [6]**	HTAM [7]	HTBMS (a) [8]	HTBMS (b) [9]	**G3SPD [10]**
OCC	-0.140 *	-0.021	-0.168 *	-0.024 ***	-0.050 *	**-0.021**	-0.021	-0.020	-0.021	**-0.021**
	(0.015)	(0.014)	(0.034)	(0.013)	(0.017)	**(0.014)**	(0.014)	(0.014)	(0.014)	**(0.014)**
SOUTH	-0.056 *	-0.002	-0.057 **	0.004	-0.017	**0.007**	0.007	0.008	0.007	**-0.002**
	(0.013)	(0.034)	(0.026)	(0.032)	(0.027)	**(0.032)**	(0.032)	(0.032)	(0.032)	**(0.032)**
SMSA	0.152 *	-0.042 **	0.176 *	-0.046 **	-0.014	**-0.042 ****	-0.042 **	-0.042 **	-0.042 **	**-0.042 ****
	(0.012)	(0.019)	(0.026)	(0.019)	(0.020)	**(0.019)**	(0.019)	(0.019)	(0.019)	**(0.019)**
IND	0.047 *	0.019	0.058 **	0.015	0.004	**0.014**	0.014	0.014	0.014	**0.019**
	(0.012)	(0.015)	(0.026)	(0.015)	(0.017)	**(0.015)**	(0.015)	(0.015)	(0.015)	**(0.015)**
EXP	0.040 *	0.113 *	0.032 *	0.109 *	0.082 *	**0.113 ***	0.113 *	0.113 *	0.113 *	**0.113 ***
	(0.002)	(0.002)	(0.005)	(0.002)	(0.003)	**(0.002)**	(0.002)	(0.002)	(0.002)	**(0.003)**
EXP2	-0.001 *	0.000 *	-0.001 *	0.000 *	-0.001 *	**0.000 ***	0.000 *	0.000 *	0.000 *	**0.000 ***
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	**(0.000)**
WKS	0.004 *	0.001	0.009 **	0.001	0.001	**0.001**	0.001	0.001	0.001	**0.001**
	(0.001)	(0.001)	(0.004)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	**(0.001)**
MS	0.048 **	-0.030	0.115 **	-0.038 **	-0.075 *	**-0.030**	-0.030	-0.030	-0.030	**-0.030**
	(0.021)	(0.019)	(0.048)	(0.019)	(0.023)	**(0.019)**	(0.019)	(0.019)	(0.019)	**(0.018)**
UNION	0.093 *	0.033 **	0.109 *	0.037 **	0.063 *	**0.033 ****	0.032 **	0.033 **	0.032 **	**0.033 ****
	(0.013)	(0.015)	(0.029)	(0.015)	(0.017)	**(0.015)**	(0.015)	(0.015)	(0.015)	**(0.014)**
FEM	-0.368 *		-0.317 *	-0.161	-0.339 *	**-0.131**	-0.132	-0.134	-0.133	**-0.317 ***
	(0.025)		(0.055)	(0.136)	(0.051)	**(0.127)**	(0.127)	(0.127)	(0.127)	**(0.028)**
BLK	-0.167 *		-0.158 *	-0.266	-0.210 *	**-0.286 *****	-0.286 ***	-0.279 ***	-0.284 ***	**-0.158 ***
	(0.022)		(0.045)	(0.166)	(0.058)	**(0.156)**	(0.155)	(0.155)	(0.155)	**(0.024)**
ED	0.057 *		0.051 *	0.138 *	0.100 *	**0.138 ***	0.137 *	0.142 *	0.138 *	**0.051 ***
	(0.003)		(0.006)	(0.015)	(0.006)	**(0.021)**	(0.021)	(0.020)	(0.020)	**(0.003)**
Const	5.251 *		5.121 *	3.030 *	4.264 *	**2.913 ***	2.927 *	2.873 *	2.921 *	**5.121 ***
	(0.071)		(0.204)	(0.209)	(0.098)	**(0.284)**	(0.275)	(0.271)	(0.274)	**(0.093)**
MuHat										**1.000 ***
										(0.006)

Notes: Left-hand-side variable is the logarithm of wage. The regressors FEM, BLK, and ED are time-invariant regressors. The MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by the G3SPD estimator. Hausman-Taylor estimates with the Breusch, Mizon and Schmidt's (1989) instruments (HTBMS) are obtained under two assumptions of exogeneity of regressors: (a) UNION is doubly-exogenous; (b) UNION is singly-exogenous. In all cases, FEM and BLK are considered doubly-exogenous while ED is singly-exogenous. The dataset corresponds to Baltagi and Khanti-Akom (1990) and is available from the *Journal of Applied Econometrics* web site. Sample: 595 individuals throughout 7 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Table 7. Wage determinants and returns to schooling: estimates from a subsample of the PSID, 1976-1982.

(specification by Baltagi and Khanti-Akom, 1990, controlling for time effects)

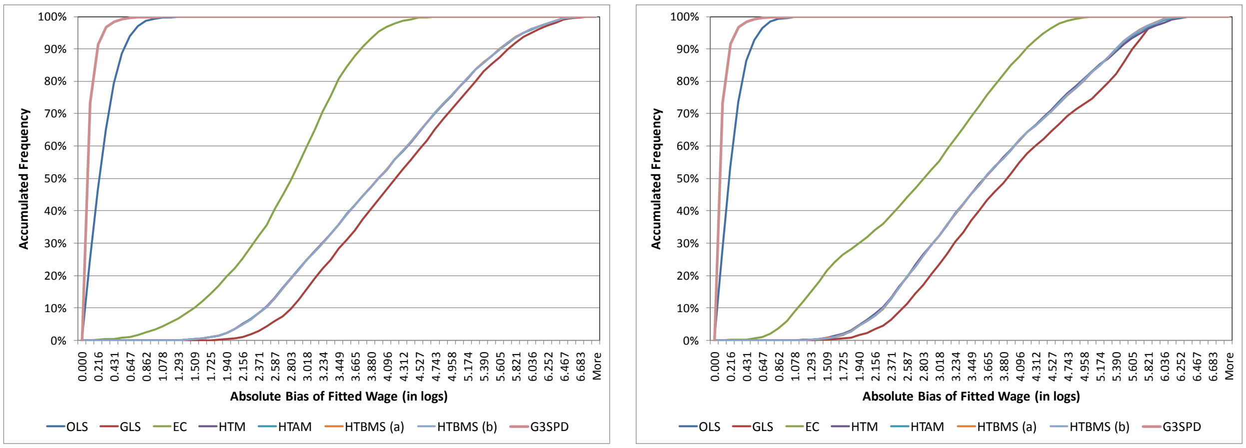
	OLS [1]	FE [2]	BE [3]	GLS [4]	EC [5]	**HTM [6]**	HTAM [7]	HTBMS (a) [8]	HTBMS (b) [9]	**G3SPD [10]**
OCC	-0.136 *	-0.019	-0.168 *	-0.022 ***	-0.044 *	-0.018	-0.019	-0.019	-0.019	-0.019
	(0.013)	(0.014)	(0.034)	(0.013)	(0.013)	(0.014)	(0.013)	(0.013)	(0.013)	(0.014)
SOUTH	-0.060 *	0.003	-0.057 **	-0.009	-0.058 *	0.009	0.006	0.004	0.004	0.003
	(0.011)	(0.034)	(0.026)	(0.030)	(0.021)	(0.031)	(0.031)	(0.031)	(0.031)	(0.032)
SMSA	0.165 *	-0.042 **	0.176 *	-0.029 ***	0.047 *	-0.039 **	-0.037 **	-0.036 **	-0.036 **	-0.042 **
	(0.011)	(0.019)	(0.026)	(0.018)	(0.016)	(0.019)	(0.018)	(0.018)	(0.018)	(0.019)
IND	0.054 *	0.021	0.058 **	0.022	0.029 **	0.016	0.017	0.018	0.018	0.021
	(0.010)	(0.015)	(0.026)	(0.014)	(0.013)	(0.015)	(0.015)	(0.015)	(0.015)	(0.015)
EXP	0.032 *	0.104 *	0.032 *	0.039 *	0.030 *	0.099 *	0.088 *	0.082 *	0.083 *	0.104 *
	(0.002)	(0.009)	(0.005)	(0.004)	(0.002)	(0.008)	(0.008)	(0.007)	(0.007)	(0.011)
EXP2	-0.001 *	0.000 *	-0.001 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *	0.000 *
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
WKS	0.004 *	0.001	0.009 **	0.001	0.001 ***	0.001	0.001	0.001	0.001	0.001
	(0.001)	(0.001)	(0.004)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
MS	0.081 *	-0.029	0.115 **	-0.031 ***	-0.018	-0.029	-0.029	-0.030	-0.030	-0.029
	(0.018)	(0.019)	(0.048)	(0.018)	(0.018)	(0.019)	(0.018)	(0.018)	(0.018)	(0.019)
UNION	0.085 *	0.030 **	0.109 *	0.029 **	0.041 *	0.030 **	0.029 **	0.029 **	0.029 **	0.030 **
	(0.011)	(0.015)	(0.029)	(0.014)	(0.013)	(0.015)	(0.015)	(0.014)	(0.014)	(0.014)
d1978	0.158 *	0.041 *		0.139 *	0.156 *	0.048 *	0.065 *	0.074 *	0.072 *	0.041 **
	(0.015)	(0.015)		(0.009)	(0.008)	(0.014)	(0.013)	(0.013)	(0.013)	(0.016)
d1979	0.246 *	0.052 **		0.215 *	0.243 *	0.063 *	0.091 *	0.107 *	0.104 *	0.052 **
	(0.015)	(0.023)		(0.011)	(0.008)	(0.022)	(0.020)	(0.019)	(0.019)	(0.025)
d1980	0.324 *	0.055 ***		0.283 *	0.321 *	0.071 **	0.111 *	0.133 *	0.128 *	0.055
	(0.015)	(0.032)		(0.013)	(0.009)	(0.029)	(0.027)	(0.025)	(0.026)	(0.034)
d1981	0.399 *	0.046		0.340 *	0.390 *	0.067 ***	0.117 *	0.146 *	0.140 *	0.046
	(0.016)	(0.040)		(0.016)	(0.009)	(0.037)	(0.034)	(0.032)	(0.032)	(0.044)
d1982	0.482 *	0.046		0.406 *	0.468 *	0.072	0.134 *	0.168 *	0.161 *	0.046
	(0.016)	(0.049)		(0.019)	(0.010)	(0.045)	(0.041)	(0.039)	(0.039)	(0.053)
FEM	-0.353 *		-0.317 *	-0.389 *	-0.422 *	-0.177	-0.217 ***	-0.240 **	-0.235 **	-0.317 *
	(0.022)		(0.055)	(0.120)	(0.040)	(0.118)	(0.116)	(0.114)	(0.115)	(0.028)
BLK	-0.161 *		-0.158 *	-0.164	-0.153 *	-0.260 ***	-0.239 ***	-0.224	-0.229 ***	-0.158 *
	(0.019)		(0.045)	(0.146)	(0.045)	(0.142)	(0.139)	(0.138)	(0.138)	(0.024)
ED	0.054 *		0.051 *	0.081 *	0.067 *	0.132 *	0.125 *	0.124 *	0.122 *	0.051 *
	(0.002)		(0.006)	(0.014)	(0.005)	(0.020)	(0.019)	(0.018)	(0.019)	(0.003)
Const	5.135 *		5.121 *	4.921 *	5.240 *	3.219 *	3.505 *	3.626 *	3.623 *	5.087 *
	(0.062)		(0.204)	(0.204)	(0.078)	(0.321)	(0.303)	(0.295)	(0.297)	(0.094)
MuHat										1.000 *
										(0.014)

Notes: Left-hand-side variable is the logarithm of wage. The regressors FEM, BLK, and ED are time-invariant regressors. The MuHat coefficient corresponds to the coefficient accompanying the set of pseudo-effects generated by the G3SPD estimator. Hausman-Taylor estimates with the Breusch, Mizon and Schmidt's (1989) instruments (HTBMS) are obtained under two assumptions of exogeneity of regressors: (a) UNION is doubly-exogenous; (b) UNION is singly-exogenous. In all cases, FEM and BLK are considered doubly-exogenous while ED is singly-exogenous. The regression includes 5 time-dummies, namely d1978-d1982, to control for time effects. The dataset corresponds to Baltagi and Khanti-Akom (1990) and is available from the *Journal of Applied Econometrics* web site. Sample: 595 individuals throughout 7 years. Numbers in parenthesis indicate the standard error. Significance at 1%, 5%, and 10% are represented by '*', '**', and '***', respectively.

Figure 11. Wage determinants and returns to schooling: Cumulative distribution of the absolute bias of fitted models

(specification by Baltagi and Khanti-Akom, 1990)

Without controlling for time effects (based on estimates from Table 6) ***Controlling for time effects (based on estimates from Table 7)***



Notes: Fitted values corresponding to estimations in Table 6 and 7. For more details, see the corresponding notes. Frequencies computed as the number of observations in each bin. Sample: 595 individuals throughout 7 years.

Table 8. Estimation results from Monte Carlo simulations for a wage equation with a time-invariant omitted variable correlated with included regressors

(Absolute bias, measured in %)

FIXED EFFECTS

	ρ=0				***ρ=0.35***				***ρ=0.70***			
Vars.	**OLS**	**EC**	**HTM**	***G3SPD***	**OLS**	**EC**	**HTM**	***G3SPD***	**OLS**	**EC**	**HTM**	***G3SPD***
EXP	1.101	1.081	0.367	***0.405***	0.118	0.118	0.419	***0.400***	0.036	0.042	0.476	***0.409***
	(5.58)	(5.59)	(14.85)	***(15.10)***	(5.58)	(5.59)	(14.83)	***(15.11)***	(5.58)	(5.57)	(14.90)	***(15.10)***
EXP2	8.125	8.002	4.161	***2.586***	1.025	0.997	3.970	***2.561***	0.058	0.009	3.868	***2.606***
	(33.26)	(33.33)	(92.81)	***(93.00)***	(33.26)	(33.31)	(92.77)	***(93.03)***	(33.26)	(33.19)	(93.12)	***(92.97)***
FEM	3.277	3.305	3.368	***3.404***	1.290	1.224	0.500	***1.960***	1.399	1.457	1.887	***0.934***
	(23.69)	(23.70)	(25.48)	***(24.71)***	(23.69)	(23.70)	(25.66)	***(24.71)***	(23.69)	(23.67)	(25.56)	***(24.71)***
BLK	7.813	7.799	16.774	***7.673***	0.222	0.229	16.568	***0.412***	1.302	1.314	17.815	***1.554***
	(40.29)	(40.28)	(131.22)	***(40.33)***	(40.29)	(40.28)	(143.62)	***(40.33)***	(40.29)	(40.28)	(134.01)	***(40.33)***
ED	0.147	0.148	13.287	***1.022***	9.896	9.811	23.155	***9.036***	9.357	9.302	28.141	***8.367***
	(14.37)	(14.34)	(257.61)	***(14.93)***	(14.37)	(14.47)	(285.70)	***(14.94)***	(14.37)	(14.49)	(267.71)	***(14.95)***

RANDOM EFFECTS

	ρ=0				***ρ=0.35***				***ρ=0.70***			
Vars.	**OLS**	**EC**	**HTM**	***G3SPD***	**OLS**	**EC**	**HTM**	***G3SPD***	**OLS**	**EC**	**HTM**	***G3SPD***
EXP	1.209	1.184	0.319	***0.408***	0.227	0.217	0.342	***0.414***	0.072	0.057	0.357	***0.411***
	(5.59)	(5.61)	(14.91)	***(15.10)***	(5.59)	(5.62)	(14.89)	***(15.10)***	(5.59)	(5.61)	(14.85)	***(15.09)***
EXP2	8.659	8.515	4.344	***2.603***	1.558	1.482	3.924	***2.632***	0.592	0.492	3.500	***2.617***
	(33.35)	(33.44)	(93.15)	***(93.02)***	(33.35)	(33.45)	(93.05)	***(93.00)***	(33.35)	(33.42)	(92.81)	***(92.95)***
FEM	2.211	2.253	2.759	***2.016***	2.356	2.285	1.186	***3.349***	0.333	0.383	1.027	***0.455***
	(23.81)	(23.81)	(25.78)	***(24.85)***	(23.81)	(23.80)	(25.71)	***(24.84)***	(23.81)	(23.81)	(25.68)	***(24.85)***
BLK	6.318	6.298	15.846	***6.178***	1.718	1.724	12.910	***1.905***	2.798	2.810	9.947	***3.048***
	(40.24)	(40.24)	(135.62)	***(40.29)***	(40.24)	(40.24)	(138.12)	***(40.29)***	(40.24)	(40.23)	(135.71)	***(40.29)***
ED	0.581	0.607	14.365	***0.525***	10.331	10.265	18.369	***9.493***	9.792	9.747	14.447	***8.877***
	(14.42)	(14.47)	(271.83)	***(15.04)***	(14.42)	(14.40)	(273.91)	***(14.99)***	(14.42)	(14.47)	(267.20)	***(15.04)***

Notes: Left-hand-side variable is the logarithm of wage. The estimated model coincides with the one showed in Table 4. The regressors FEM, BLK, and ED are time-invariant regressors. The original dataset corresponds to Baltagi and Khanti-Akom (1990) and is available from the *Journal of Applied Econometrics* web site. Complementary information as well as Monte Carlo simulations are explained in Section 6. The parameter ρ refers to the correlation of the omitted variable with ED. Numbers in parenthesis indicate the empirical standard error corresponding to each set of 500 estimated coefficients.

Chapter 2

A Generalized Three-Step Panel Data Estimator: Inference Issues

2.1 Introduction

In a companion paper (see Avanzini, 2010, and chapter 1, above), we have shown that the Generalized Three-Step Panel Data (G3SPD) estimator performs better in terms of bias than other available estimators when either the dimension-varying or the dimension-invariant regressors and the unobservable effects are correlated. However, as it is well known, there is a tradeoff between bias and variance that we need to address in order to have a complete picture of the behavior of the G3SPD estimator. Both biased estimates and big variances lead to inaccurate inference. While measures such as the root mean squared error (RMSE) capture the total tradeoff, it is not possible to know its sources. Given that the G3SPD estimator is less biased than other available estimators in this context, we need to show also that its accuracy for inference purposes has not been undermined as a by-product of the reduced bias.

The G3SPD estimator allows the estimation of single-equation and systems of simultaneous equations models with panel data, when the "true model" contains unobservable effects. Particularly, the G3SPD has proven to be very reliable en terms of bias when faced with the problem of obtaining consistent estimates when (i) the set of regressors includes dimension-invariant variables, and (ii) some (or all) of the regressors are correlated with the unobservable effects. In this context, the least squares methods (OLS) yield biased estimates when the true model contains fixed effects, and when the regressors are correlated with unobservable information. Error-components models (EC) also yield biased estimates when regressors are correlated with the effects: this results from the biased estimates obtained by the between-groups estimator (BE). Within-groups estimators (FE) obtain unbiased and consistent estimates even when the regressors are correlated with the unobservable effects, at the cost of neglecting the impact of the dimension-invariant variables (which are wiped out by the within-groups transformation). Finally, an intermediate solution by Hausman and Taylor (1981) uses instrumental variables taken from the same panel to control the bias generated by the correlation between the regressors and the unobservable effects.[1] Using Monte Carlo simulations, Avanzini (2010, and chapter 1, above) showed that all previous estimators perform poorly in terms of bias.

The G3SPD estimator consists of a three-step estimation procedure that obtains substantially less biased estimators for all the coefficients in the regression, irrespective of assuming fixed- or random-effects. This estimator introduces a new regressor, dubbed the *pseudo-effects*, to control the bias generated by the correlation between the regressors and the unobservable effects. The pseudo-effects are obtained from within- and between-groups regressions; the latter are not necessarily unbiased though, in the overall, they help to control the bias of the endogenous regressors.

In this chapter we present the exact variance-covariance of the G3SPD estimator

[1] In this case, obtaining estimates of the coefficients of the dimension-invariant regressors depends on the availability of totally exogenous dimension-varying regressors which are used as instruments.

and compares its accuracy against other available estimators. The exact variance-covariance matrix allows for the inclusion of generated regressors, i.e. the pseudo-effects. We show that the standard errors obtained from the residuals of the 3^{rd} Step regression of the G3SPD estimator are downward biased because they do not account for the stochasticity of the generated or imputed regressors. The proposed variance-covariance matrix is at least as big as the variance-covariance matrix of other panel data estimators. However, using Monte Carlo experiments, we show that the adjusted matrix obtains more reliable standard errors than other panel data estimators, as compared with the variability of the estimated coefficients.

We also analyze the behavior of the estimated standard errors and how they can affect inference. Particularly, we analyze the confidence intervals built on those estimated standard errors: using Monte Carlo simulations, we compare the size of the confidence intervals obtained by the G3SPD estimator and by the forementioned methods. We find that the G3SPD adjusted-standard errors outperforms the other ones in terms of the accuracy of the confidence intervals, with substantial improvements for the system estimator.

According to our experiments, alternative panel data methods perform poorly, exacerbating the size of the standard errors and the confidence intervals, specially the Hausman-Taylor-type procedures (HT). The problem aggravates when estimating the entire system of simultaneous equations. The pattern of behavior of these estimators hardly change with increasing sample size, while aggravating with higher correlation between the regressors and the unobservable effects. Hence, even when it may be tempting to use estimators with smaller standard errors, inference may be highly misleading, as it can be concluded from our Monte Carlo simulations. On the other hand, the G3SPD estimator obtains more reliable standard errors across different sample sizes and degrees of correlation of the regressors with the unobservable effects, for both fixed- and random-effects designs. Moreover, sensible improvements in the size of the confidence intervals and accuracy of the estimated standard errors can be achieved by applying the G3SPD estimator to the entire system of equations.

Our study develops in six sections including the current one. Generalities of the G3SPD estimator as well as a workhorse model are presented in Section 2.2. In Section 2.3 we discuss the estimation of the standard errors and introduce the exact variance-covariance matrix for the 3^{rd} Step estimator. The analysis of the behavior of the standard errors and the confidence intervals are performed in Section 2.4 based on Monte Carlo simulations. Section 2.5 contains comments regarding alternative strategies to estimate the standard errors, as well as possible paths to improve the efficiency of the G3SPD estimator. Section 2.6 concludes. In Appendix 2.A we describe the measures used in this chapter to evaluate the behavior of the standard errors and the confidence intervals.

2.2 The G3SPD estimator

2.2.1 Overview

In Avanzini (2010, and chapter 1, above), we developed a three-step estimation procedure, the Generalized Three-Step Panel Data estimator (G3SPD), that is capable to obtain substantially less biased estimates of the coefficients when the panel data model contains both dimension-varying and dimension-invariant regressors correlated with the unobservable effects, irrespective of assuming fixed or random effects. The G3SPD estimator is based on a simple intuition: if the source of bias in a panel data regression is the correlation between the regressors and the unobservable effects, introducing an estimate of the latter in the model may contribute to eliminate, or at least attenuate, such bias. Consequently, obtaining a set of estimates of the unobservable effects is a key feature of the G3SPD estimator, that requires a multi-step regression procedure. Thus, the method consists of three estimation steps: the first step is aimed to obtain consistent estimates of the coefficients of the dimension-varying covariates, while the second step obtains a set of estimators (not necessarily unbiased) for both the dimension-varying and the dimension-invariant covariates. Both sets of estimators are combined to obtain a set of estimated unobservable effects dubbed *pseudo-effects*. The third step includes the pseudo-effects into a pooled regression together with dimension-varying and dimension-invariant covariates. According to the Monte Carlo simulations, this three-step procedure yields less biased estimators for all coefficients when compared with other methods such as Pooled OLS, FE and BE, EC, and HT estimators.

The proposed G3SPD estimator has remarkable characteristics: first, it outperforms all other panel data estimators by a substantial measure, whether in terms of bias, variability, and RMSE. Second, the performance of the G3SPD estimator does not depend on assuming fixed or random effects, given that the procedure does not require a decision over a set of possible estimation methods as in the usual approaches. Third, the estimator handles both the *"panel"* and the *"time-series cross-section"* setups. Moreover, according to the Monte Carlo simulations, it presents a good performance in finite samples, and its bias diminishes with increases on the total sample size, irrespective of whether N or T increases. Fourth, classifying the regressors as singly- or doubly-exogenous (a key issue in

HT procedures) is irrelevant because there is no need to instrumentalize the correlation between the regressors and the unobservable effects once the pseudo-effects are introduced in the regression. Fifth, the residuals of the 3^{rd} Step of G3SPD estimator are closer to the true innovations of the model, which implies better forecasting, inference, and estimation of the $n-$dimensional confidence-intervals.

From a practical perspective, implementing the G3SPD estimator has other advantages: first, given that the G3SPD equation-by-equation estimator proved to perform very similar to the system estimator, equation-by-equation modelling can be used whenever there is a risk of generalized bias arising in system estimation when one of the equations is misspecified. Second, the method deals with dimension-invariant variables and endogeneity of regressors in a simple and elegant way, and it relies on widely known statistical properties, namely those regarding within- and between-groups estimators, and GMM estimators for systems of equations. Third, its estimation procedure is simple, apparent, and quick: estimating G3SPD implies running a within-groups estimator, a between-groups estimator, and a pooled regression. Finally, the estimator is able to accomodate several alternative specifications that account for SUR designs, block-diagonal simultaneous equations (partitioned systems of equations), unbalanced panels, $n-$way or multidimensional panels, etc, and can be easily implemented using standard econometric packages.

2.2.2 A workhorse model

Although the G3SPD estimator can be applied to $n-$dimensional panels, in order to simplify the exposition, we concentrate on one-way balanced panels with unobservable individual-specific effects. Consequently, consider the (reduced-form) system of simultaneous equations[2]:

$$y = Y\pi + X\beta + Z\gamma + \eta \tag{2.1}$$

where the unobservable part is assumed to follow:

$$\eta = Z_{\varphi}\varphi + Z_{\mu}\mu + \varepsilon \tag{2.2}$$

The system in [2.1] contains G equations, with N observations throughout T periods of time in each equation, where y is a $GNT \times 1$ vector containing the set of endogenous left-hand side variables, and Y is a $GNT \times (G-1)$ matrix containing the $(G-1)$ set of remaining endogenous variables appearing on the right-hand side of each equation. Hence y_g is a $NT \times 1$ vector, and Y_g is a $NT \times (G-1)$ matrix. Aditionally, the right-hand side set of variables may include K dimension-varying covariates – the X's – and J dimension-invariant covariates – the Z's. These are block-diagonal matrices of size $GNT \times GK$ and $GNT \times GJ$, respectively.[3] The corresponding set of coefficients are block-diagonal arrangements of parameters: π is a $G(G-1) \times G$ matrix, β is a $GK \times G$ matrix, and γ is a $GJ \times G$ matrix.[4]

Regarding exogeneity, we adopt the Cornwell et al. (1992) classification, which separates the regressors in three categories: *doubly exogenous* (covariates that are uncorrelated with both the stochastic term and the effects), *singly exogenous* (covariates that are uncorrelated with the stochastic term but correlated with the effects), and *endogenous* covariates (those correlated with both the stochastic error term and the effects). We assume that the correlation between the regressors and the unobservable individual-specific effects

[2]For the sake of brevity, we explain the G3SPD estimator in the context of a system of simultaneous equations estimation given that it encompasses single-equation estimation as well.

[3]Note that the Z's do not vary across time periods remaining constant for each individual, which will imply perfect collinearity with the individual-specific effects, as defined afterwards.

[4]For exposition purposes, assume that all equations share the same number of dimension-varying and dimension-invariant regressors.

only involve their individual means.

We assume that the unobservable information in [2.2] can be split into three components for each equation: a constant φ (which can be thought as corresponding to the error mean), a set of individual effects μ (corresponding to a constant term for each individual), and a purely stochastic component ε. Z_μ is a selector matrix representing a set of *dummy* variables for each individual that arrange the μ_g allocating one for each individual for all time periods, and it is obtained as $Z_{\mu_g} = I_N \otimes \iota_T$, whereas for the whole system we have $Z_\mu = I_G \otimes Z_{\mu_g}$. Z_φ is a selector matrix representing a set of *dummy* variables that allocates φ_g in the corresponding equation, and is obtained as $Z_\varphi = I_G \otimes \iota_{NT}$. The (homoskedastic) stochastic error term of each equation in the model distributes $\varepsilon_g \sim iid(0, \sigma^2_{\varepsilon_g})$. Thus, the stochastic error term of the system is $\varepsilon \sim iid\left(0, \Sigma_\varepsilon \otimes I_{NT}\right)$. When fixed effects are assumed, the set of unobservable effects μ_g remains constant for each possible realization of y, while varying among individuals according to $\mu_g \sim iid(0, \sigma^2_{\mu_g})$.[5] In this case, the stochastic part of η only corresponds to ε, and $\eta_g \sim iid\left(\varphi_g + \mu_g, \sigma^2_{\varepsilon_g}\right)$. For the whole system, $\eta \sim iid\left(Z_\varphi \varphi + Z_\mu \mu, \Sigma_\varepsilon \otimes I_{NT}\right)$. Alternatively, when random effects are assumed, the set of unobservable effects μ_{gi} varies across the possible realizations of y for each individual: in this case $\mu_{gi} \sim iid(0, \sigma^2_{\mu_{gi}})$. Assuming $\sigma^2_{\mu_{gi}} = \sigma^2_{\mu_g}\ \forall i = 1, \ldots, N; g = 1, \ldots, G$, it follows that for the whole system,

$$\Sigma = E(\eta\eta') = \Sigma_\mu \otimes (I_N \otimes J_T) + \Sigma_\varepsilon \otimes I_{NT} \tag{2.3}$$

where $J_T = \iota_T \iota_T'$, ι_T is a column-vector of ones of order T, and Σ_μ and Σ_ε are both $G \times G$ matrices. Using Wansbeek and Kapteyn's (1982a, 1982b, 1983) results, we can rewrite [2.3]

[5]This configuration follows from the interpretation of the fixed effects by Hsiao (2002, Ch. 3). This also justifies the fact that even when the effects are considered fixed, they can be correlated with the regressors (e.g. see Cornwell and Trumbull, 1994; and Baltagi, 2006).

as:

$$\begin{aligned}\Sigma &= (T\Sigma_\mu + \Sigma_\varepsilon) \otimes \left(I_N \otimes \overline{J_T}\right) + \Sigma_\varepsilon \otimes (I_N \otimes E_T) \\ &= \Sigma_1 \otimes P_{\mu_g} + \Sigma_\varepsilon \otimes Q_{\mu_g}\end{aligned} \tag{2.4}$$

where $\overline{J_T} = (1/T)J_T$, $E_T = I_T - \overline{J_T}$, $\Sigma_1 = T\Sigma_\mu + \Sigma_\varepsilon$, and P_{μ_g} and Q_{μ_g} are projection matrices of the form $P_A = A(A'A)^{-1}A'$ and $Q_A = I_A - P_A$, respectively, and I_A is a conformable identity matrix.[6] Notice that P_{μ_g} is the *between-groups transformation* that returns the individual means from a panel dataset, and Q_{μ_g} is the *fixed effects or within-groups transformation* which returns deviations from the individual means.[7] For the whole system, $P_\mu = I_G \otimes P_{\mu_g}$ and $Q_\mu = I_{GNT} - P_\mu$.

2.2.3 The G3SPD procedure

The G3SPD procedure estimates the following system of simultaneous equations:

$$y = Y\pi + X\beta + Z\gamma + Z_\varphi\varphi + Z_\mu\mu + \varepsilon \tag{2.5}$$

where feasibility of the estimation is solved replacing μ with an estimator $\hat{\mu}$ that we call the *pseudo-effects*. Briefly, the three steps of the method are:

1^{st} Step: Estimate a within-groups-transformed regression. Transform model [2.5] by pre-multiplying it by Q_{μ_g} or Q_μ depending on the type of estimation at hand (i.e.

[6] The result in [2.4] corresponds to the spectral decomposition of Σ derived by Baltagi (1980), and which implies that $\Sigma^r = \Sigma_1^r \otimes P_{\mu_g} + \Sigma_\varepsilon^r \otimes Q_{\mu_g}$, where r is an arbitrary scalar (see Baltagi, 2005, Ch. 6). This result facilitates the obtention of the covariance matrixes, and represents the known-form of the heteroskedasticity of η.

[7] Note that these projection matrices are symmetric, idempotent, of rank $rank(P_{\mu_g}) = tr(P_{\mu_g}) = N$ and $rank(Q_{\mu_g}) = tr(Q_{\mu_g}) = N(T-1)$, orthogonal (i.e. $P_{\mu_g}Q_{\mu_g} = 0$), and sum to identity matrix (i.e. $P_{\mu_g} + Q_{\mu_g} = I_{NT}$).

equation-by-equation or system estimation, respectively) and estimate a regression with the transformed model. This will yield consistent and unbiased estimators $\hat{\pi}^{wi}$ and $\hat{\beta}^{wi}$.

2^{nd} Step: Estimate a between-groups-transformed regression. Now, estimate a regression using only the means for each individual obtained from the model [2.5] after transforming it (i.e., pre-multiply it by P_{μ_g} if it is a single-equation or an equation-by-equation estimation, or by P_μ when considering the whole system of simultaneous equations). This will yield $\hat{\pi}^{be}$, $\hat{\beta}^{be}$, $\hat{\gamma}^{be}$, and $\hat{\varphi}^{be}$.

3^{rd} Step: Estimate a pooled regression including the *pseudo-effects* as a new regressor. Finally, estimate the following pooled regression:

$$y = Y\pi + X\beta + Z\gamma + Z_\varphi\varphi + [Z_\mu\hat{\mu}]\phi + \eta \tag{2.6}$$

where we have introduced as a new regressor, $[Z_\mu\hat{\mu}]$, the *pseudo-effects.* This step obtains the correct set of residuals that are used in inference, confidence-intervals estimation, forecasting, etc. Without this step, we only have dispersed estimates for the set of parameters in [2.5]. The pseudo-effects comprise a mix of within- and between-groups estimators as follows:

$$\hat{\mu} = P_\mu y - P_\mu(Y\hat{\pi}^{wi} + X\hat{\beta}^{wi}) - (Z\hat{\gamma}^{be} + Z_\varphi\hat{\varphi}^{be}) \tag{2.7}$$

2.3 Obtaining the standard errors

2.3.1 Standard approach

In Avanzini (2010) we present a set of estimators that can be used in the different steps of the G3SPD estimator. They belong to the GMM class inspired by Amemiya (1977) and are able to accommodate different sets of instruments for different equations when needed[8]. General expressions for the equation-by-equation estimator of a generic vector of parameters θ_g and the corresponding (homoskedastic) covariance matrix are given by:

$$\hat{\theta}_g^{GMM} = \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1} R_g' W_g' P_{[A_g,B_g]} W_g y_g \tag{2.8}$$

$$\hat{\Omega}_g^{GMM} = \hat{\sigma}_g^2 \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1} \tag{2.9}$$

where R_g is the set of regressors included in the equation, W_g is a weighting matrix, and $P_{[A_g,B_g]}$ is the projection matrix of instruments. The set of instruments $[A_g, B_g]$ is capable of arranging instruments for both dimension-varying, A_g, and dimension-invariant endogenous regressors, B_g. The weighting matrix W_g allows for heteroskedasticity of various forms and model transformations. $\hat{\sigma}_g^2 = (\hat{\varepsilon}_g' \hat{\varepsilon}_g)/dof$ is a consistent estimator for σ_g^2, $\hat{\varepsilon}_g = W_g y_g - W_g R_g \hat{\theta}_g^{GMM}$ is a consistent estimator for ε_g, and dof stands for the degrees-of-freedom adjustment[9].

Estimating a system of equations allows us to avoid the loss of information implicit in equation-by-equation estimation. In fact, efficiency gains in system of equations estimation are the result of non-neglecting cross-equation correlations of the disturbances. However it must be emphasized that the consequences of model misspecification when estimating the entire system are more harmful than when estimating it equation by equation: a misspecified equation contaminates the entire system and all parameters may be biased, while estimating it equation by equation only affects the misspecified equation keeping the other estimated equations unaffected. The extension of the previous estimator to systems

---

[8]Some aspects regarding biasedness of GMM estimators, specially in small samples, are discussed in Section 4 of Avanzini (2010), and Section 1.4, above.

[9]See Theorem 5.2 and Eqs. 5-25 to 5-27 in Wooldridge (2002), and Section 5.4 in Greene (2002).

of simultaneous equations is given by:

$$\hat{\theta}^{GMM} = \left[R'W'P^{*}_{[A,B]}WR\right]^{-1} R'W'P^{*}_{[A,B]}Wy \tag{2.10}$$

with $P^{*}_{A} = A(A'\check{\Sigma}A)^{-1}A'$, where $\check{\Sigma}$ is a consistent estimator of the covariance matrix[10]. The estimator in [2.10] is capable of arranging different sets of instruments in different equations, as well as different covariance estimators for the entire system of equations. The set of instruments $[A, B]$ is a block-diagonal matrix with each block containing the proper set of instruments for each equation. Different choices of the weighting matrix W and the set of instruments $[A, B]$ in $\hat{\theta}_{g}^{GMM}$ and $\hat{\theta}^{GMM}$ allow to accomodate most of the classical panel-data estimators.[11] The corresponding (homoskedastic) covariance matrix estimator is given by:

$$\hat{\Omega}^{GMM} = \left[R'W'P^{**}_{[A,B]}WR\right]^{-1} \tag{2.11}$$

with $P^{**}_{A} = A(A'\hat{\Sigma}A)^{-1}A'$ and $\hat{\Sigma} = \left[\frac{\tilde{\varepsilon}'W'W\tilde{\varepsilon}}{dof}\right] \otimes I_{NT}$. The residuals, $\hat{\varepsilon} = Wy - WR\hat{\theta}^{GMM}$, are arranged in a $NT \times G$ matrix, i.e. $\tilde{\varepsilon} = [\hat{\varepsilon}_1, \hat{\varepsilon}_2, \ldots, \hat{\varepsilon}_G]$. Asymptotically, $\hat{\Sigma} = \check{\Sigma}$, though in small samples they may differ.

The estimators in [2.8] and [2.10] are of the single-step-GMM type which obtains unbiased and consistent estimates of the coefficients, though not necessarily efficient ones. Alternative methods have been proposed to improve efficiency, specially in small samples (e.g. see Hansen, 1982; Hansen et al., 1996; Newey and Smith, 2004), and they are commented in Section 2.5.

---

[10]See Cornwell et al. (1992), Hsiao (2002), Wooldridge (2002), Baltagi (2005), among others, for a detailed explanation on the procedure to obtain consistent estimates of $\check{\Sigma}$.

[11]See Appendix F in Avanzini (2010), and Appendix 1.F.

2.3.2 Adjusting for the inclusion of generated regressors

Notice that even when [2.9] and [2.11] are good estimators for the variances of ε_g and ε, respectively, they are not adequate for η_g and η in [2.6]. The problem is that the residuals of the model [2.6] differs from ε due to the inclusion of "generated or imputed regressors".[12] Provided that the two-step model is adequately specified, and that the auxiliary regression obtains consistent estimates of the coefficients, the generated regressor will be a consistent estimate of the actual unobserved regressor it proxies. Moreover, the estimated coefficients in the final-step regression will be consistent as well. However, the standard errors will be biased and inconsistent (see Pagan, 1984; Hoffman, 1987).

The inconsistency of the standard errors is the result of not accounting for the fact that we are including estimates of the regressors instead of the actual ones. This implies that the residual of the regression including the generated regressors differs from the error of the "true" model. In fact, the use of generated regressors may precipitate a nonscalar disturbance matrix when they are treated as ordinary nonstochastic regressors. Consequently, the reliability of inference is questionable: Pagan (1984, 1986) suggests that the standard errors obtained from OLS and similar methods are inconsistent estimators of the true ones. Using Monte Carlo experiments designed to measure this bias in small samples, Hoffman et al. (1984) find that appreciable bias exists and that the number of spurious rejections is greatly reduced by accounting for "generated regressors bias" (see Hoffman, 1987). Moreover, Murphy and Topel (1985) pose that two-step procedures fail to account for the fact that generated regressors are measured with sampling error, thus

[12] Two-step regression models have been widely studied in the literature for the single-equation case (e.g. Pagan, 1984, 1986; Murphy and Topel, 1985; Hoffman, 1991; Gawande, 1997; McKenzie and McAleer, 1997) and for systems of equations (e.g. Hoffman, 1987).

hypothesis tests based on the estimated covariance matrix of the second-step estimator are biased even in large samples.

The G3SPD estimator, like other multi-step estimators, suffers from this bias and inconsistency of the standard errors as obtained by standard methods. Observe that the variance estimators in [2.9] and [2.11] assume that the residuals of the regression only include ε. However, from [2.6] we know that η does not equal ε. To see this, define $\bar{y} = P_\mu y$, $H = [Y, X] = P_\mu H + Q_\mu H = \bar{H} + \tilde{H}$, $F = [Z,\ Z_\varphi,\ Z_\mu \hat{\mu}]$, $\delta = [\pi',\ \beta']'$, $\lambda = [\gamma',\ \varphi', \phi']'$, and $\lambda^* = [\gamma',\ \varphi']'$. Thus, the system [2.6] can be rewritten as:

$$\begin{aligned} y &= H\delta + F\lambda + \eta \\ &= H\delta + F\lambda + [(Z_\mu \mu - Z_\mu \hat{\mu}\phi) + \varepsilon] \end{aligned}$$

Correspondingly, the pseudo-effects obtain as:

$$\hat{\mu} = \bar{y} - \bar{H}\hat{\delta}^{wi} - F\hat{\lambda}^{*be}$$

As explained in Avanzini (2010), even when the regressors in H are singly-exogenous, the within-groups estimator $\hat{\delta}^{wi}$ is consistent and unbiased. But this is not the case for $\hat{\lambda}^{*be}$: its general form is given by:

$$\hat{\lambda}^{*be} = \lambda + \frac{M_{F\bar{H}}^{\bar{H}}}{M_F^{\bar{H}}}\delta + \frac{M_{F\mu}^{\mu}}{M_F^{\mu}}$$

where the last two terms account for the bias generated by the correlation between F and μ. M_{AB} and M_A are given by $M_{AB}^B = A'Q_B B$ and $M_A^B = A'Q_B A$, respectively. Note that if F is uncorrelated with μ, both terms $\frac{M_{F\bar{H}}^{\bar{H}}}{M_F^{\bar{H}}}$ and $\frac{M_{F\mu}^{\mu}}{M_F^{\mu}}$ converge to zero. On the other hand, when F is correlated with the effects and $\bar{H}$ is not, $\frac{M_{F\bar{H}}^{\bar{H}}}{M_F^{\bar{H}}}$ vanishes and the bias is represented by $\frac{M^{\mu}_{F\mu}}{M^{\mu}_{F}}$. Finally, when both F and $\bar{H}$ are correlated with the effects, the total bias is given by $\frac{M^{\bar{H}}_{F\bar{H}}}{M^{H}_{F}}\delta + \frac{M^{\mu}_{F\mu}}{M^{\mu}_{F}}$.

Using [2.8] or [2.10] we can recover the estimators of the parameters in [2.6] corresponding to the 3^{rd} Step of the G3SPD estimator. Applying the Generalized Frisch-Waugh Theorem (Krishnakumar, 2006, Theorem 5.1), and after some calculations, we have:

$$\begin{aligned}
plim\ \hat{\delta}^{GMM} &= plim\ \hat{\delta}^{wi} = \delta \\
plim\ \hat{\lambda}^{*GMM} &= plim\ \hat{\lambda}^{*be} = \lambda^{*} + \frac{M^{\bar{H}}_{F\bar{H}}}{M^{\bar{H}}_{F}}\delta + \frac{M^{\mu}_{F\mu}}{M^{\mu}_{F}} \\
plim\ \hat{\phi}^{GMM} &= 1
\end{aligned}$$

Replacing $\hat{\mu}, \hat{\delta}^{wi}, \hat{\lambda}^{*be}, \hat{\phi}^{GMM}$ in η we have:

$$\begin{aligned}
\eta &= [Z_{\mu}\mu - Z_{\mu}\hat{\mu}\phi] + \varepsilon \\
&= Z_{\mu}\mu - Z_{\mu}\Big[\bar{y} - \bar{H}\delta - \bar{H}\left(\tilde{H}'\tilde{H}\right)^{-1}\tilde{H}'\tilde{\varepsilon} - F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}}F\lambda \\
&\quad - F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}}\bar{H}\delta - F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}}Z_{\mu}\mu - \\
&\quad - F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}}\bar{\varepsilon}\Big] + \varepsilon
\end{aligned}$$

By assumption, μ is uncorrelated with ε. Moreover, $\bar{\varepsilon} = 0$ by construction because we have included a constant term in each equation $(Z_{\varphi}\varphi)$ that eventually captures $\bar{\varepsilon} \neq 0$. Grouping terms we can obtain the "true" η for the system:

$$\eta^{GMM} = \varepsilon + \bar{H}\left(\tilde{H}'\tilde{H}\right)^{-1}\tilde{H}'\tilde{\varepsilon} + F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}}\left[\bar{H}\delta + Z_{\mu}\mu\right]$$

Notice that the second and third term of η^{GMM} correspond to the bias generated by the use of $\hat{\mu}$ instead of the true μ.

Assuming homoskedasticity, for a single-equation, the $NT \times NT$ variance-covariance matrix of η_g^{GMM} is given by:

$$Var(\eta_g^{GMM}) = E\left(\eta_g \eta_g'\right) = S_g \Sigma_g S_g'$$

where S_g is a $NT \times 3NT$ matrix,

$$S_g = \left[I_{NT} \;\middle|\; \bar{H}_g \left(\tilde{H}_g' \tilde{H}_g\right)^{-1} \tilde{H}_g' \;\middle|\; F_g \left(F_g' Q_{\bar{H}_g} F_g\right)^{-1} F_g' Q_{\bar{H}_g} \right]$$

Σ_g is a $3NT \times 3NT$ matrix,

$$\Sigma_g = E \begin{bmatrix} \varepsilon_g' \varepsilon_g & \varepsilon_g' \tilde{\varepsilon}_g & \varepsilon_g' \bar{\upsilon}_g \\ \tilde{\varepsilon}_g' \varepsilon_g & \tilde{\varepsilon}_g' \tilde{\varepsilon}_g & \tilde{\varepsilon}_g' \bar{\upsilon}_g \\ \bar{\upsilon}_g' \varepsilon_g & \bar{\upsilon}_g' \tilde{\varepsilon}_g & \bar{\upsilon}_g' \bar{\upsilon}_g \end{bmatrix} \otimes I_{NT}$$

and $\bar{\upsilon}_g = \bar{H}_g \delta_g + Z_{\mu_g} \mu_g$. It is apparent that $Var(\eta_g^{GMM}) \geq Var(\varepsilon_g)$, which implies that the standard errors obtained from [2.9] understate the "true" standard errors of the estimated coefficients. The corresponding estimator of the variance-covariance matrix, based on the "sandwich" estimator, is given by:

$$\begin{aligned} \Omega_g^{GMM-Adj} &= \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1} \times \\ &\quad \times \left(R_g' W_g' P_{[A_g,B_g]} W_g \left[S_g \Sigma_g S_g'\right] W_g' P_{[A_g,B_g]} W_g R_g\right) \times \\ &\quad \times \left[R_g' W_g' P_{[A_g,B_g]} W_g R_g\right]^{-1} \end{aligned} \tag{2.12}$$

Observe that when $\left[S_g \Sigma_g S_g'\right] = \Sigma_{\varepsilon_g} = \sigma_\varepsilon^2 I_{NT}$, $\Omega_g^{GMM-Adj}$ becomes Ω_g^{GMM} as defined in [2.9].

A feasible estimator of $\Omega_g^{GMM-Adj}$ can be obtained by replacing ε_g with the residuals from the 3^{rd} Step estimation, $\tilde{\varepsilon}_g$ with the residuals from the 1^{st} Step estimation, and $\bar{\upsilon}_g$ with $\widehat{\bar{\upsilon}}_g = \bar{H}_g \hat{\delta}_g^{wi} + Z_{\mu_g} \hat{\mu}_g$ or $\widehat{\bar{\upsilon}}_g = \bar{y}_g - F_g \hat{\lambda}_g^{be}$.

The preceding variance-covariance estimator can be extended to the system of equations. Define the $GNT \times 3GNT$ matrix S as

$$S = \left[I_{GNT} \quad \middle| \quad \bar{H}\left(\tilde{H}'\tilde{H}\right)^{-1}\tilde{H}' \quad \middle| \quad F\left(F'Q_{\bar{H}}F\right)^{-1}F'Q_{\bar{H}} \right],$$

Σ as a $3GNT \times 3GNT$ matrix,

$$\Sigma = E\begin{bmatrix} \varepsilon'\varepsilon & \varepsilon'\tilde{\varepsilon} & \varepsilon'\bar{\upsilon} \\ \tilde{\varepsilon}'\varepsilon & \tilde{\varepsilon}'\tilde{\varepsilon} & \tilde{\varepsilon}'\bar{\upsilon} \\ \bar{\upsilon}'\varepsilon & \bar{\upsilon}'\tilde{\varepsilon} & \bar{\upsilon}'\bar{\upsilon} \end{bmatrix} \otimes I_{NT}$$

and $\bar{\upsilon} = \bar{H}\delta + Z_\mu\mu$. The variance-covariance matrix of the residual part of the system of equations is given by:

$$Var(\eta^{GMM}) = E\left(\eta\eta'\right) = S\Sigma S'$$

and the corresponding variance-covariance matrix is:

$$\begin{aligned} \Omega^{GMM-Adj} \quad &= \quad \left[R'W'P^{**}_{[A,B]}WR\right]^{-1} \times \\ &\times \left(R'W'P^{**}_{[A,B]}W\left[S\Sigma S'\right]W'P^{**}_{[A,B]}WR\right) \times \\ &\times \left[R'W'P^{**}_{[A,B]}WR\right]^{-1} \end{aligned} \tag{2.13}$$

Again, the unknown components of Σ can be obtained from the three steps of the G3SPD estimator.

Finally, notice that the empirical counterparts used to obtain the feasible estimators $\hat{\Omega}_g^{GMM-Adj}$ and $\hat{\Omega}^{GMM-Adj}$ may be biased: as discussed in Avanzini (2010) and the previous chapter, identification problems prevent us to control that bias. However, we study the extent of the bias in small samples using Monte Carlo simulations, as shown below.

2.4 Evaluating the estimated standard errors

In this section, we analyze the behavior of the estimated standard errors. Particularly, we are interested in the accuracy of the standard errors and the size of the confidence intervals built on these estimates.

In general, the unbiased model with the smallest (root) mean square error (RMSE) is interpreted as best explaining the variability of the observations. However, when the bias is important, the RMSE is not a complete indicator of the variability of the estimated coefficients. As shown in Avanzini (2010) and chapter 1 above, the G3SPD estimator is less biased than several panel data estimators including Hausman-Taylor-type procedures (e.g. see Hausman and Taylor, 1981; Amemiya and McCurdy, 1986; Breusch et al., 1989; Wyhowski, 1994), though the RMSE is not necessarily smaller. In fact, using the RMSE criterion may lead to choose, for example, OLS or EC estimators that may be highly biased when regressors are correlated with the unobservable effects.

To gain insight, we complement the RMSE with two additional measures, namely "overconfidence" and the "true 95% size". Both are based on Beck and Katz (1995) and are explained in Appendix 2.A. Overconfidence measures the extent by which an estimator of the standard errors understates variability, comparing the true sampling variability (represented by the RMSE) with the reported standard errors. Values over 100% indicates that the reported variability understates the true variability. Accurate standard errors should yield overconfidence values below 100%; nevertheless, very low values call for efficiency improvements. On the other hand, the "true 95% size" measures the accuracy of the standard errors through its "true level" of reported 95% confidence intervals. Values beyond 95%

indicate that the standard errors are upward biased, while values below 95% indicate that the standard errors are downward biased. Any deviation from the 95% implies that the individual coefficients hypothesis tests may be biased and inference on these outcomes may be misleading. Note that these two measures evaluate the behavior of the estimated standard errors using first- and second-moment estimations, while the RMSE only uses first-moment estimations.

We compare the performance of the standard errors obtained by the G3SPD estimator against other available panel data estimators such as HT-type estimators, pooled OLS, FE and BE, and EC with different variance estimators. Recall that, in this context, OLS, EC, and BE methods may yield biased estimates of the coefficients when the true model contains fixed effects or the regressors are correlated with the unobservable effects. On the other hand, the FE estimator obtains consistent and unbiased estimates only for the dimension-varying regressors. Apparently, only the HT-type estimators are adequate to deal with the inclusion of endogenous dimension-varying and dimension-invariant regressors[13]. However, Avanzini (2010) shows that the G3SPD estimator outperforms these estimators in terms of bias (including the HT-type estimators). See also chapter 1, above.

We use Monte Carlo simulations to evaluate the behavior of the standard errors obtained by the G3SPD estimator and other usual panel data estimators. Monte Carlo simulations are a useful tool in this context because they allow us to focus on specific problems controlling the cases. Moreover, we can check the behavior of different estimators under the same dataset. Finally, and more importantly, we can study the performance of

[13]Recall that HT procedures undertake two-stage least squares error-components models with a set of instruments taken from the same panel data accounting for potential endogeneity of regressors due to their correlation with the unobservable effects.

the G3SPD and other estimators in small samples.

2.4.1 Simulations setup

We run experiments for a system of two equations, i.e. $G = 2$, which is sufficient for evaluating the performance of the standard errors. The i^{th} individual during the t^{th} period obeys the following system of simultaneous equations:

$$
\begin{aligned}
y_{1it} &= Y_{2it}\pi_1 + X_{1it}\beta_{11} + X_{2it}\beta_{12} + X_{3it}\beta_{13} + \qquad (2.14a)\\
&\quad + Z_{1i}\gamma_{11} + Z_{2i}\gamma_{12} + \varphi_1 + \mu_{1i} + \varepsilon_{1it}\\
y_{2it} &= Y_{1it}\pi_2 + X_{1it}\beta_{21} + X_{4it}\beta_{24} + X_{5it}\beta_{25} + \qquad (2.14b)\\
&\quad + Z_{1i}\gamma_{21} + Z_{3i}\gamma_{23} + \varphi_2 + \mu_{2i} + \varepsilon_{2it}
\end{aligned}
$$

Note that X_1 and Z_1 are doubly exogenous variables common to both equations, while X_2, X_3 and Z_2 only appear in equation [2.14a]. X_2 is a doubly exogenous variable and X_3 and Z_2 are singly exogenous variables. Likewise for X_4, X_5, and Z_3 in equation [2.14b].

The π coefficients were obtained from a $U[-1, 1]$, while the remaining ones came from $U[0, 1] * 15$. The selected coefficients are:

$$
\begin{array}{ccccccc}
\pi_1 = .055 & \beta_{11} = 8 & \beta_{12} = 2 & \beta_{13} = 5 & \gamma_{11} = 7 & \gamma_{12} = 5 & \varphi_1 = 8 \\
\pi_2 = -0.2 & \beta_{21} = 12 & \beta_{24} = 7 & \beta_{25} = 3 & \gamma_{21} = 6 & \gamma_{23} = 5 & \varphi_2 = 4
\end{array}
$$

We vary sample sizes over N $(20, 60)$ and T $(15, 30, 50)$. This yields six cases; however, due to computing limitations, for the equation-by-equation estimation, we do not run experiments for $N = 60$, $T = 50$, while for the system estimation, we also excluded the experiments for $N = 60$, $T = 30$. Each experiment was replicated 1000 times[14].

[14]Notice that for $N = 60$ and $T = 15$, estimating an experiment (1000 replications) takes more than 40

The regressors were generated following Nerlove (1971), Im et al. (1999) and Baltagi et al. (2003) as:

Starting Values	Data Generating Process
$X_{1i1} = \frac{\eta_{1i1}}{(1-0.5^2)^{\frac{1}{2}}} + \frac{\phi_{1i}}{(1-0.5)}$	$X_{1it} = 0.5X_{1it-1} + \phi_{1i} + \eta_{1it}$
$X_{2i1} = \frac{\eta_{2i1}}{(1-0.2^2)^{\frac{1}{2}}} + \frac{\phi_{2i}}{(1-0.2)}$	$X_{2it} = 0.2X_{2it-1} + \phi_{2i} + \eta_{2it}$
$X_{3i1} = \frac{\eta_{3i1}}{(1-0.7^2)^{\frac{1}{2}}} + \frac{\mu_{1i}}{(1-0.7)}$	$X_{3it} = 0.7X_{3it-1} + \mu_{1i} + \eta_{3it}$
$X_{4i1} = \frac{\eta_{4i1}}{(1-0.2^2)^{\frac{1}{2}}} + \frac{\phi_{4i}}{(1-0.2)}$	$X_{4it} = 0.2X_{4it-1} + \phi_{4i} + \eta_{4it}$
$X_{5i1} = \frac{\eta_{5i1}}{(1-0.7^2)^{\frac{1}{2}}} + \frac{\mu_{2i}}{(1-0.7)}$	$X_{5it} = 0.7X_{5it-1} + \mu_{2i} + \eta_{5it}$
	$Z_{1i} = \xi_{1i}$
	$Z_{2i} = \mu_{1i} + \xi_{2i}$
	$Z_{3i} = \mu_{2i} + \xi_{3i}$

where $\phi_{ji}, \eta_{jit}, \xi_{ji} \sim U[-2, 2]\ \forall j$.

Regarding the individual effects, we focus on two designs:

1. *Fixed-effects case*, where μ follows $\mu_{gi} \sim N\left(0, \sigma^2_\mu\right)\ \forall g = 1, 2$, which implies that for the whole set of 1000 replications the set of μ is kept fixed.

2. *Random-effects case*, where μ follows $\mu_{gki} \sim N\left(0, \sigma^2_\mu\right)\ \forall g = 1, 2; k = 1, \ldots, 1000$. This means that we have a set of μ for each replication.

In both cases, the stochastic error terms follow $\varepsilon_{git} \sim N\left(0, \sigma^2_\varepsilon\right)\ \forall g = 1, 2$.

Following previous literature, total variance for each equation is fixed at $\sigma^2 = \sigma^2_\mu + \sigma^2_\varepsilon = 3$. The degree of heterogeneity of the effects as measured by $\rho = \frac{\sigma^2_\mu}{\sigma^2_\varepsilon}$ varies over the

hours using MATLAB R2008b's optimized set of commands to handle matrixes, running on an Intel Core 2 Duo CPU at 2.00 GHz with 3 Gb of physical memory. We run 55 of such experiments. Bigger panel datasets should be estimated using loops and partioned matrixes, a very slow processing strategy. Most of the experiments were conducted using MATLAB R2007b running on the following machines: an Intel Core 2 Duo CPU at 3.00 GHz with 3.25 Gb of physical memory; an Intel Core 2 CPU at 1.86 GHz with 1.00 Gb of physical memory; and an Intel Pentium IV CPU at 3.00 GHz with 2.00 Gb of physical memory.

following eleven values: $0, \frac{1}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, \frac{11}{12}, 1$. This allows to study how the correlation between the unobservable effects and the regressors affects the bias of the estimators. It is apparent that X_3, Z_2, and μ_1 are correlated, likewise X_5, Z_3, and μ_2. However, when $\rho = 0$, individual effects are absent from the model, and we only have Y_1 and Y_2 as endogenous regressors due to their correlation with ε. Accordingly, variables for equation [2.14a] are classified as $X_1^{(1)} = [X_1, X_2]$, $X_1^{(2)} = X_3$, $Z_1^{(1)} = Z_1$, $Z_1^{(2)} = Z_2$, and for equation [2.14b], as $X_2^{(1)} = [X_1, X_4]$, $X_2^{(2)} = X_5$, $Z_2^{(1)} = Z_1$, $Z_2^{(2)} = Z_3$, where the superscripts (1) and (2) indicate doubly- and singly-exogeneity, respectively. We also identify two additional sets of variables for each equation, namely $X_g^{(3)}$ and $Z_g^{(3)}$, which contain the variables excluded from equation g that can be used as instruments. Hence, we have $X_1^{(3)} = [X_4, X_5]$, $Z_1^{(3)} = Z_3$, $X_2^{(3)} = [X_2, X_3]$, $Z_2^{(3)} = Z_2$.

2.4.2 Results

The first observation regards the information the RMSE may contribute: when the researcher needs to choose a methodology, in the present context, the RMSE appears not to be very informative; moreover, it could lead the practioner to choose, for example, the OLS estimator based on the results for the endogenous set of variables at the cost of highly biased estimates of the parameters (that in our simulated example can be as high as 20%, see Avanzini, 2010). The RMSE appears to be closely representing the variability of the estimates of the coefficients which we consider is not very relevant when they are highly biased. For this reason we use analyze the behavior of the G3SPD and other estimators using an overconfidence indicator and the "true " size of the 95% confidence interval. These measures are described in Appendix 2.A.

When looking at the overconfidence indicator and the "true 95% size", we observe that the accuracy gains of the estimated standard errors of the G3SPD estimator are apparent for the adjusted-GMM covariance matrix, as was expected based on the previous discussion. When these outcomes are compared with those from the FE estimators for the dimension-varying regressors, and with those from the BE estimators for the dimension-invariant regressors, the results are close with the largest differences coming from the standard errors of the estimated coefficients accompanying the dimension-invariant regressors. Two effects are undermining the accuracy of these standard errors: first, the remaining bias due to the correlation of the dimension-invariant regressors and the unobservable effects, and second, the practical impossibility of separating the dimension-invariant regressors and the unobservable effects given both are representing sets of information with similar dynamics.

To gain insight, we plot the overconfidence indicator and the "true 95% size" of the confidence intervals in Figures 1 to 12. We show the behavior of the estimated standard errors corresponding to the coefficients of an endogenous variable, Y_2, a singly-exogenous dimension-invariant variable, Z_2, and singly-exogenous dimension-varying variable, X_3, entering equation [2.14a] when their coefficients are estimated equation-by-equation (Figures 1 to 6) and by system estimation (Figures 7 to 12). Each three-dimensional chart in each Figure shows: the overconfidence or the "true 95% size" in the vertical axis, depending on the Figure; the heterogeneity of the unobservable effects (i.e. Rho, that represents different levels of $\rho = \sigma_\mu^2/\sigma_\varepsilon^2$, where higher values imply higher heterogeneity of the effects, while $\rho = 0$ implies no effects at all); and sample sizes (refered to as (N, T)). Notice that discrete changes in (N, T) favors non-smooth changes in the surface (i.e. tha "peaks" and "valleys" that can be observed in the charts). To facilitate comparisons, the vertical axis scale of all charts displaying the overconfidence measure has been fixed to 70%-130%, while for the "true 95% size", it has been fixed to 40%-100%. Color scales have been adjusted accordingly. The reported methods are G3SPD with the adjusted-GMM covariance matrix to account for the inclusion of generated regressors, OLS, HTM, and either FE for time-varying variables or BE for time-invariant ones.

[Insert Figure 1 about here]

In Figure 1, we can see that the behavior of the G3SPD and FE estimators are quite similar, presenting no special pattern of behavior across sample sizes or heterogeneity of the unobservable effects, for both fixed- and random-effects designs. Both sets of surfaces lie around the 100% level, as expected. On the other hand, the standard errors obtained by the OLS estimator tend to understate the true variability of the estimated coefficients: under the fixed-effects design, overconfidence diminishes with increasing heterogeneity of the unobservable effects, while declining very slow with increasing sample sizes; under the random-effects design, overconfidence increases with heterogeneity and with T, ranging from around 100% to near 150%, i.e. standard errors may understate true parameter variability by about 50%. The HTM estimator behaves similar to OLS but with a narrower range of variation.

Coincidental with the conclusions of Hoffman et al. (1984) and Murphy and Topel (1985), changes in sample size appear to have almost no effect in the improvement of the accuracy of the standard errors. Moreover, the (generally) tempting adoption of estimators with smaller standard errors may lead to incorrect inference when they are understating the true variability of the estimated coefficients.

Now we turn to Figure 2. Observe that again the true size of the confidence intervals at the 95% of significance obtained from the G3SPD and the FE estimators behave very closely, irrespective of the design of the effects, changes in sample size, and degrees of heterogeneity of the individual effects. Standard errors obtained by OLS give rise to estimated confidence intervals that are far beyond the theoretical 95%, being even lower than 50%. The HTM standard errors behave more uniformly, showing a deterioration with the increasing degrees of heterogeneity of the unobservable effects. Confidence intervals constructed with the standard errors obtained by OLS or HTM will tend to increase the number of spurious rejections of the hypothesis.

[Insert Figure 2 about here]

The behavior of the estimators of the coefficients accompanying the time-invariant variable Z_2 are more erratic. Notice that the G3SPD overconfidence change with sample size and correlation (see Figure 3). However, it behaves better than the HTM estimator under both assumptions about the effects. Also the 95% confidence interval appears to be more imprecise undermined by the reasons stated for Y_2. However, it is more reliable than the confidence interval obtained from the HTM estimator, as can be inferred from Figure 4. The HTM systematically worsens with increasing correlation of the regressor and the unobservable effects, despite the changes in sample size.

[Insert Figures 3 and 4 about here]

The behavior of the estimators relating to the singly-exogenous time-varying variable are similar to those of the endogenous variable. In Figure 5, the overconfidence of the

G3SPD stabilizes around 100% while the HTM estimator tends to deteriorate for increasing levels of correlation. On the other hand, the confidence intervals shrinks with higher levels of correlation of the covariate with the unobservable effects (see Figure 6). However, while the G3SPD estimator reach 80% on average for $\rho = 1$, the HTM estimator can be as low as 20%. The important downward bias of the standard errors of the HTM estimator may highly distort inference even if the coefficients are not biased (something we know is not true according to the results in Avanzini, 2010).

[Insert Figures 5 and 6 about here]

Regarding the differences between the equation-by-equation and system estimation, reported results favor the latter due to the fact that the system estimation takes into account the cross-equation information. In fact, system estimation appears to be at least as good as equation-by-equation estimation, as expected. Compare the outcomes in Figures 1 to 6 for the equation-by-equation estimation with their corresponding system estimations displayed in Figures 7 to 12.

[Insert Figures 7 to 12 about here]

On the overall, the standard errors obtained with the adjusted-GMM variance-covariance matrix for the 3^{rd} Step of the G3SPD estimator substantially outperform the ones obtained with other methods, across designs, sample sizes, and degrees of heterogeneity of the unobservable effects. This result contributes to conclude on the suitability of the G3SPD estimator we have posed in Avanzini (2010) for the cases when the model includes dimension-varying and dimension-invariant endogenous variables, irrespective of assuming fixed- of random-effects. The latter helps to avoid the choice of the design that is a critical issue in panel data estimation.

The HT-type procedures have been widely accepted and used as the suitable solution when we have dimension-varying and dimension-invariant endogenous regressors, as in our case. However, observe that the reported variability as measured by the estimated standard errors understate the true variability of the estimated coefficients. Moreover, the 95% confidence intervals built with those standard errors tend to not include the true value of the parameters most of the times. Notice also that, as pointed out in Avanzini (2010) and chapter 1 above, regarding the bias, the system estimation tends to exacerbate the poor performance of the HTM estimator possibly due to the problem of weak instruments (see Stock and Yogo, 2004) and the across-equations contamination.

Finally, though not reported here[15], the outcomes for the remaining coefficients (namely, those corresponding to X_2 and X_4, Z_1, φ_1 and φ_2, ϕ_1 and ϕ_2) follow the same patterns.

2.5 Further comments

Some additional issues are worth mentioning though related discussion is beyond the scope of this chapter.

Heteroskedastic disturbances. Even when we disregarded this problem, an extensive literature on efficiency issues regarding heteroskedastic residuals, including White (1980), MacKinnon and White (1985), Wooldridge (1996), Cribari-Neto et al. (2000), Long and Ervin (2000), to name just a few, proposes improved variance-covariance estimators that

[15]Reports for the remaining coefficients, as well as the outcomes for other panel data estimators, are available upon request from the author.

can be extended to account for the particular characteristics of the G3SPD estimator. Moreover, even when the disturbances are in fact homoskedastic, provided the third step of the G3SPD estimator is a pooled regression, the residuals may contain some remaining heteroskedasticity due to the panel structure or other sources of heterogeneity we have not consider yet (see Parks, 1967; Beck and Katz, 1995; Im et al., 1999). In fact, Hoffman (1987) suggests that the inclusion of generated regressors may precipitate a nonscalar disturbance matrix.

Notice that when estimating the system of simultaneous equations, we have assumed that the disturbances are homoskedastic as well as the residuals. This implies that

$$\hat{\Sigma}^{homo} = \left[\frac{\tilde{\varepsilon}'\tilde{\varepsilon}}{dof}\right] \otimes I_{NT} = \begin{bmatrix} \varepsilon_1^2 & 0 & \varepsilon_1'\varepsilon_G & 0 \\ & \ddots & \cdots & \ddots & \\ 0 & \varepsilon_1^2 & 0 & \varepsilon_1'\varepsilon_G \\ & \vdots & & \vdots & \\ \varepsilon_1'\varepsilon_G & 0 & \varepsilon_G^2 & 0 \\ & \ddots & \cdots & \ddots & \\ 0 & \varepsilon_1'\varepsilon_G & 0 & \varepsilon_G^2 \end{bmatrix}$$

which in turn implies that we are estimating a three-stage least squares regression (See Greene, 2002, pp. 410). Alternatively, a fully heteroskedastic covariance matrix, as suggested by Greene, should have the following structure:

$$\hat{\Sigma}^{hetero} = \begin{bmatrix}
\varepsilon_{111}^2 & & & 0 & & \varepsilon_{111}'\varepsilon_{G11} & & & 0 \\
& \varepsilon_{112}^2 & & & & & \varepsilon_{112}'\varepsilon_{G12} & & \\
& & \ddots & & \cdots & & & \ddots & \\
0 & & & \varepsilon_{1NT}^2 & & 0 & & & \varepsilon_{1NT}'\varepsilon_{GNT} \\
& \vdots & & & & & & \vdots & \\
\varepsilon_{111}'\varepsilon_{G11} & & & 0 & & \varepsilon_{G11}^2 & & & 0 \\
& \varepsilon_{112}'\varepsilon_{G12} & & & & & \varepsilon_{G12}^2 & & \\
& & \ddots & & \cdots & & & \ddots & \\
0 & & & \varepsilon_{1NT}'\varepsilon_{GNT} & & 0 & & & \varepsilon_{GNT}^2
\end{bmatrix}$$

This estimator gives rise to the system-GMM efficient estimator. As explained by Greene, this estimator is at least as efficient as the three-stage leasts squares. Using this configuration for the covariance matrix can obtain additional efficiency gains, specially when the unobservable effects are present in the structural system of equations. In this case, they may be spread away in the reduced-form system of equations that we are estimating and some heteroskedasticity of unknown form may remain.

Alternative variance-covariance estimators. Our treatment of the variance-covariance matrices in [2.12] and [2.13] follows Hoffman (1987), in an attempt to obtain closed forms for Ω, that may help to understand the sources of bias. However, other approaches can be used, such the one by Adrian Pagan (Pagan, 1984, 1986), or the exact covariance matrix based on Murphy and Topel (1985). Those approaches may be difficult to implement specially when estimators are (potentially) biased, as in our case. Alternatively, bootstrap methods can also be applied to obtain more accurate standard errors. Viable bootstrap methods in this context range from traditional ones (e.g. Efron, 1979; Efron and Tibshirani, 1993; Cribari-Neto and Zarkos, 1999; MacKinnon, 2002; Davidson and MacKinnon, 2004; Flachaire, 2005; Davidson and Flachaire, 2008) to some specific methods that exploit panel data structure (e.g. Rilstone and Veall, 1996; Kapetanios, 2008).

Iterated G3SPD 3^{rd} Step Estimator. Note that efficiency improvements can be obtained by iterating the estimators [2.8] and [2.10]. However, given that we are using GMM-based estimators, we know that the two-step GMM (Hansen, 1982) and iterated-GMM estimators (continuos updating estimators or CUE, see Hansen et al., 1996) obtain downward biased estimates of the standard errors in small samples (see Windmeijer, 2000, 2005; Ramalho, 2005; Windmeijer suggests analytical corrections of the biases). The bias is a consequence of the use of estimated weighting matrices built on the basis of estimated parameters from previous iterations of the estimator. Thus, iterating the G3SPD 3^{rd} Step estimator would require an additional correction of the covariance matrix. This is the reason for choosing the single-step GMM for the current study. Various alternative methods are available in the literature to overcome the bias and efficiency problems such as (generalized) empirical likelihood (Owen, 1988; Qin and Lawless, 1994; Imbens, 1997; Smith, 1997; Newey and Smith, 2004), exponential tilting (Kitamura and Stutzer, 1997; Imbens et al., 1998), efficient method of moments (Gallant and Tauchen, 1996; Chumacero, 1997), bootstrap methods (Hall and Horowitz, 1996; Brown and Newey, 2002; Ramalho, 2006).

2.6 Concluding Remarks

In a companion paper (see Avanzini, 2010, and chapter 1, above), we have considered the estimation of single-equation and system of simultaneous equations models with panel data, when the "true model" contains unobservable effects. We studied the problem of obtaining consistent estimates when (i) the set of regressors includes dimension-invariant variables, and (ii) some (or all) of the regressors are correlated with the unobservable effects. To account for these issues we developed a three-step estimation procedure, the Generalized Three-Step Panel Data estimator (G3SPD).

In this chapter we study the behavior of the standard errors obtained by the G3SPD estimator. We find that the standard errors obtained in the 3^{rd} Step of the procedure are downward biased due to the inclusion of generated regressors, the *pseudo-effects*. To obtain the adequate standard errors, we develop a variance-covariance matrix estimator that accounts for this source of bias.

We analyze the behavior of these adjusted standard errors both from the perspective of its validity to adequately state the variability of the estimated coefficients, as well as the size of the confidence intervals of the individual coefficients. We use the overconfidence indicator and the "true 95% size" of the confidence intervals to measure the two aspects of the accuracy of the standard errors.

The evidence from Monte Carlo simulations supports the validity of the adjusted standard errors estimator, and shows that the third step standard errors of the G3SPD estimator perform very good irrespective of assuming fixed- or random-effects designs, sample sizes, and degrees of heterogeneity of the unobservable effects. On the contrary, standard errors obtained by widely known panel data estimators perform poorly, and the performance does not improve significantly with bigger samples.

Bibliography

[1] Amemiya, Takeshi (1977). "The maximum likelihood and the nonlinear three-stage least squares estimator in the general nonlinear simultaneous equation model", Econometrica, Vol. 45, No. 4, May, pp. 955-968.

[2] Amemiya, Takeshi, and Thomas MaCurdy (1986). "Instrumental-variable estimation of an error-components model", Econometrica, Vol. 54, No. 4, pp. 869-880.

[3] Avanzini, Diego (2010). “A Generalized Three-Step Panel Data Estimator: Bias Issues”, manuscript, Catholic University of Chile.

[4] Baltagi, Badi (1980). “On seemingly unrelated regressions with error components”, Econometrica, Vol. 48, No. 6, pp. 1547-1551.

[5] Baltagi, Badi (2005, 3rd ed.; also 2001, 2nd ed.). Econometric Analysis of Panel Data. John Wiley & Sons, LTD, New York.

[6] Baltagi, Badi (2006). "Estimating an economic model of crime using panel data from North Carolina", Journal of Applied Econometrics, Vol. 21, pp. 543-547.

[7] Baltagi, Badi, Georges Bresson and Alain Pirotte (2003). "Fixed effects, random effects or Hausman-Taylor? A pretest estimator", Economics Letters, Vol. 79, pp. 361-369.

[8] Beck, Nathaniel and Jonathan Katz (1995). "What to do (and not to do) with time-

series cross-section data", The American Political Science Review, Vol. 89, No. 3, pp. 634-647.

[9] Breusch, Trevor, Grayham Mizon, and Peter Schmidt (1989). "Efficient estimation using panel data", Econometrica, Vol. 57, No. 3, pp. 695-700.

[10] Brown, Bryan and Whitney Newey (2002). "Generalised method of moments, efficient bootstrapping, and improved inference", Journal of Business & Economic Statistics, Vol. 20, No. 4, pp. 507–517.

[11] Chumacero, Romulo (1997). “Finite sample properties of the efficient method of moments”, Studies in Nonlinear Dynamics and Econometrics, Vol. 2, No. 2, pp. 35-51.

[12] Cornwell, Christopher, and William Trumbull (1994). “Estimating the economic model of crime with panel data”, Review of Economics and Statistics, Vol. 76, No. 2, pp. 360-366.

[13] Cornwell, Christopher, Peter Schmidt and Donald Wyhowski (1992). "Simultaneous equations and panel data", Journal of Econometrics, Vol. 51, pp. 151-181.

[14] Cribari-Neto, Francisco and Spyros Zarkos (1999). "Bootstrap methods for heteroskedastic regression models: evidence on estimation and testing", Econometric Reviews, Vol. 18, No. 2, pp. 211-228.

[15] Cribari-Neto, Francisco, Silvia Ferrari and Gauss Cordeiro (2000). "Improved heteroskedasticity-consistent covariance matrix estimators", Biometrika, Vol. 87, No. 4, pp. 907-918.

[16] Davidson, Russell and Emmanuel Flachaire (2008). "The wild bootstrap, tamed at last", Journal of Econometrics, Vol. 146, pp. 162-169.

[17] Davidson, Russell and James MacKinnon (2004). "Bootstrap methods in Econometrics", manuscript, McGill University and Queen's University.

[18] Efron, Bradley (1979). "Bootstrap methods: another look at the jacknife," The Annals of Statistics, Vol. 7, No. 1, pp. 1-26.

[19] Efron, Bradley and Robert Tibshirani (1993). An introduction to the bootstrap, Chapman and Hall, London.

[20] Flachaire, Emmanuel (2005). "Bootstrapping heteroskedastic regression models: wild bootstrap vs. pairs bootstrap", Computational Statistics & Data Analysis, Vol. 49, pp. 361-376.

[21] Gallant, A. Ronald and George Tauchen (1996). "Which moments to match?", Econometric Theory, Vol. 12, No. 4, pp. 657-681.

[22] Gawande, Kishore (1997). "Generated regressors in linear and nonlinear models", Economics Letters, Vol. 54, pp. 119-126.

[23] Greene, William (2002, 5th ed.). Econometric Analysis, Pearson Education, Prentice Hall, New Jersey.

[24] Hall, Peter and Joel Horowitz (1996). "Bootstrap critical values for tests based on generalised-method-of-moments estimators", Econometrica, Vol. 64, No. 4, pp. 891–916.

[25] Hansen, Lars (1982). "Large sample properties of Generalized Method of Moments estimators", Econometrica, Vol. 50, No. 4, pp. 1029-1054.

[26] Hansen, Lars, John Heaton and Amir Yaron (1996). "Finite-sample properties of some alternative GMM estimators", Journal of Business & Economic Statistics, Vol. 14, No. 3, pp. 262-280.

[27] Hausman, Jerry, and William Taylor (1981). "Panel data and unobservable individual effects", Econometrica, Vol. 49, No. 6, pp. 1377-1398.

[28] Hoffman, Dennis (1987). "Two-step Generalized Least Squares Estimators in multi-equation generated regressor models", The Review of Economics and Statistics, Vol. 69, No. 2, pp. 336-346.

[29] Hoffman, Dennis (1991). "Two-step and related estimators in contemporary rational-expectations models: an analysis of small-sample properties", Journal of Business and Economic Statistics, Vol. 9, No. 1, pp. 51-61.

[30] Hoffman, Dennis, Stuart Low and Don Schlagenhauf (1984). "Tests of rationality, neutrality and market efficiency: a Monte Carlo analysis of alternative test statistics", Journal of Monetary Economics, Vol. 14, No. 3, pp. 339-363.

[31] Hsiao, Cheng (2002, 2nd ed.; also 1986, 1st ed.). Analysis of Panel Data. Cambridge University Press, Cambridge.

[32] Im, Kyung, Seung Ahn, Peter Schmidt and Jeffrey Wooldridge (1999). "Efficient estimation of panel data models with strictly exogenous explanatory variables", Journal of Econometrics, Vol. 93, pp. 177-201.

[33] Imbens, Guido (1997). "One-step estimators for over-identified Generalized Method of Moments models", The Review of Economic Studies, Vol. 64, No. 3, pp. 359-383.

[34] Imbens, Guido, Richard Spady, and Phillip Johnson (1998). "Information theoretic approaches to inference in moment condition models", Econometrica, Vol. 66, No. 2, pp. 333-357.

[35] Kapetanios, G. (2008). "A bootstrap procedure for panel data sets with many cross-sectional units", Econometrics Journal, Vol. 11, pp. 377-395.

[36] Kitamura, Yuichi and Michael Stutzer (1997). "An information-theoretic alternative to Generalized Method of Moments estimation", Econometrica, Vol. 65, No. 4, pp. 861-874.

[37] Krishnakumar, Jaya (2006). "Time invariant variables and panel data models: a Generalised Frisch-Waugh Theorem and its implications", in Badi Baltagi (ed.), "Panel Data Econometrics", Elsevier B.V. (Oxford, UK), pp. 119-132.

[38] Kristensen, Ida and Gregory Wawro (2003). "Lagging the dog? The robustness of Panel Corrected Standard Errors in the presence of serial correlation and observation specific effects", manuscript prepared for presentation at the 2003 Summer Methods Conference, Department of Political Science, Columbia University.

[39] Long, J. Scott and Laurie Ervin (2000). "Using heteroscedasticity consistent standard errors in the linear regression model", The American Statistician, Vol. 54, No. 3, pp. 217-224.

[40] MacKinnon, James (2002). "Bootstrap inference in Econometrics", paper presented

as the Presidential Address at the 2002 Annual Meeting of the Canadian Economics Association.

[41] MacKinnon, James, and Halbert White (1985). "Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties", Journal of Econometrics, Vol. 29, pp. 305-325.

[42] McKenzie, Colin, and Michael McAleer (1997). "On efficient estimation and correct inference in models with generated regressors: a general approach", The Japanese Economic Review, Vol. 48, No. 4, pp. 368-389.

[43] Murphy, Kevin, and Robert Topel (1985). "Estimation and inference in two-step econometric models", Journal of Business and Economic Statistics, Vol. 3, No. 4, pp. 88-97.

[44] Nerlove, Marc (1971). "Further evidence on the estimation of dynamic economic relations from a time series of cross sections", Econometrica, Vol. 39, No. 2, pp. 359-382.

[45] Newey, Whitney, and Richard Smith (2004). "Higher order properties of GMM and Generalized Empirical Likelihood estimators", Econometrica, Vol. 72, No. 1, pp. 219-255.

[46] Owen, Art (1988). "Empirical Likelihood Ratio confidence intervals for a single functional", Biometrika, Vol. 75, No. 2, pp. 237-249.

[47] Pagan, Adrian (1984). "Econometric issues in the analysis of regressions with generated regressors", International Economic Review, Vol. 25, No. 1, pp. 221-247.

[48] Pagan, Adrian (1986). “Two stage and related estimators and their applications”, Review of Economic Studies, Vol. 53, pp. 517-538.

[49] Parks, Richard (1967). "Efficient estimation of a system of regression equations when disturbances are both serially and contemporaneously correlated", Journal of the American Statistical Association, Vol. 62, No. 318, pp. 500-509.

[50] Qin, Jin and Jerry Lawless (1994). "Empirical likelihood and general estimating equations", The Annals of Statistics, Vol. 22, No. 1, pp. 300–325.

[51] Ramalho, Joaquim (2005). "Small sample bias of alternative estimation methods for moment condition models: Monte Carlo evidence for covariance structures". Studies in Nonlinear Dynamics and Econometrics, Vol. 9, No. 1, Article 3. Also available as Documento de Trabalho 2003/09, Departamento de Economia, Universidade de Evora.

[52] Ramalho, Joaquim (2006). “Bootstrap bias-adjusted GMM estimators”, Economics Letters, Vol. 92, pp. 149-155.

[53] Rilstone, P., and M. R. Veall (1996). “Using bootstrapped confidence intervals for improved inferences with seemingly unrelated regression equations,” Econometric Theory, Vol. 12, pp. 569–580.

[54] Smith, Richard (1997). "Alternative semi-parametric likelihood approaches to Generalised Method of Moments estimation", The Economic Journal, Vol. 107, No. 441, pp. 503-519.

[55] Stock, James, and Motohiro Yogo (2004). "Testing for weak instruments in linear IV

regression", manuscript, Harvard University (first draft: 2001; NBER Technical Working Paper 284: 2002).

[56] Wansbeek, Tom and Arie Kapteyn (1982a). "A class of decompositions of the variance-covariance matrix of a generalized error components model", Econometrica, Vol. 50, No. 3, pp. 713-724.

[57] Wansbeek, Tom and Arie Kapteyn (1982b). "A simple way to obtain the spectral decomposition of variance components models for balanced data", Communications in Statistics, Vol. A11, pp. 2105-2112.

[58] Wansbeek, Tom and Arie Kapteyn (1983). "A note on spectral decomposition and maximum likelihood estimation of ANOVA models with balanced data", Statistics and Probability Letters, Vol. 1, pp. 213-215.

[59] White, Halbert (1980). "A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity", Econometrica, Vol. 48, No. 4, pp. 817-838.

[60] Windmeijer, Frank (2000). "A finite sample correction for the variance of linear efficient two-step GMM estimators", Institute for Fiscal Studies Working Paper Series No. W00/19, London.

[61] Windmeijer, Frank (2005). "A finite sample correction for the variance of linear efficient two-step GMM estimators", Journal of Econometrics, Vol. 126, pp. 25-51.

[62] Wooldridge, Jeffrey (1996). "Estimating systems of equations with different instruments for different equations", Journal of Econometrics, Vol. 74, pp. 387-405.

[63] Wooldridge, Jeffrey (2002). Econometric Analysis of Cross Section and Panel Data, The MIT Press, Cambridge, MA.

[64] Wyhowski, Donald (1994). "Estimation of a panel data model in the presence of correlation between regressors and a two-way error-component", Econometric Theory, Vol. 10, pp. 130-139.

2.A Performance Measures

2.A.1 Root Mean Squared Error

The (root) mean squared error (RMSE) accounts for the trade-off between bias and variance of a estimator. When the true value of the parameter θ is known, the RMSE of the estimator $\hat{\theta}$ can be obtained as:

$$RMSE = \left[\frac{\sum_{m=1}^{M} \left[\hat{\theta}_m - \theta \right]^2}{(M-1)} \right]^{1/2}$$

where M is the number of replications. The relative RMSE is given by:

$$RMSE_{(\%)} = \left| \frac{\left[\frac{\sum_{m=1}^{M} \left[\hat{\theta}_m - \theta \right]^2}{(M-1)} \right]^{1/2}}{\theta} \right| \times 100$$

Usually, smaller values of the RMSE are interpreted as improved quality of the estimator. However, the solely use of the RMSE as a criterion to choose a estimator may be misleading. Complementary measures can help to make a right decision specially when the estimator may be biased.

2.A.2 Overconfidence

This measure is proposed by Beck and Katz (1995)[16]. It is intended to measure the performance of the estimated standard errors. According to the authors, an accurate measure of the sampling variability of each estimator is the standard deviation of the set of $\hat{\theta}$'s. The quality of the estimates of the variability can then be assessed by comparing the *root mean square average of the M estimated standard errors* with the corresponding *standard deviation of the M estimates*, where M is the number of replications. The corresponding measure is called *overconfidence*, and is defined as the percentage by which an estimator understates variability, i.e.,

$$OC_{(\%)} = \frac{\sqrt{\sum_{m=1}^{M}\left(\hat{\theta}_m - \bar{\hat{\theta}}\right)^2}}{\sqrt{\sum_{m=1}^{M}\left(stder(\hat{\theta}_m)\right)^2}} \times 100$$

For example, a reported overconfidence of 200% indicates that the true sampling variability of an estimator is, on average, twice the reported estimate of that variability.

2.A.3 "True 95% size"

The accuracy of the standard errors and the corresponding hypothesis tests built on these estimates can be evaluated through its "true level" of reported 95% confidence intervals. This measure consists of counting the number of times the true parameter belongs to the 95% confidence interval computed using the estimated standard errors. The problem is that, in general, the confidence intervals are computed assuming a $t-Student$ distribution of the residuals (instead of a Normal distribution, honoring the small sample problem), but when the residuals are not distributed as assumed, the tests may be biased.

[16] Kristensen and Wawro (2003) called this measure "optimism".

To compute this measure, define the upper and lower bounds of the confidence interval evaluated at 95% for the m^{th} replication as:

$$\begin{aligned} UB_m &= \hat{\theta}_m + \left[\left|t^{0.975}_{NT-K}\right| \times stder(\hat{\theta}_m)\right] \\ LB_m &= \hat{\theta}_m - \left[\left|t^{0.975}_{NT-K}\right| \times stder(\hat{\theta}_m)\right] \end{aligned}$$

Then

$$\text{"95\% Size"} = \left[\frac{\#^{M}_{m=1}(LB_m \leq \theta \leq UB_m)}{M}\right] \times 100$$

where # stands for "count the number of times the condition in brackets is true", and K is the number of regressors in the regression. Whenever the measure is under 95%, the estimated variability leads to overconfidence. Moreover, values above and beyond 95% lead to under- and over-rejection of the null hypothesis.

2.B Figures

Figure 1.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (Y_2) in Equation 1, equation-by-equation estimation.

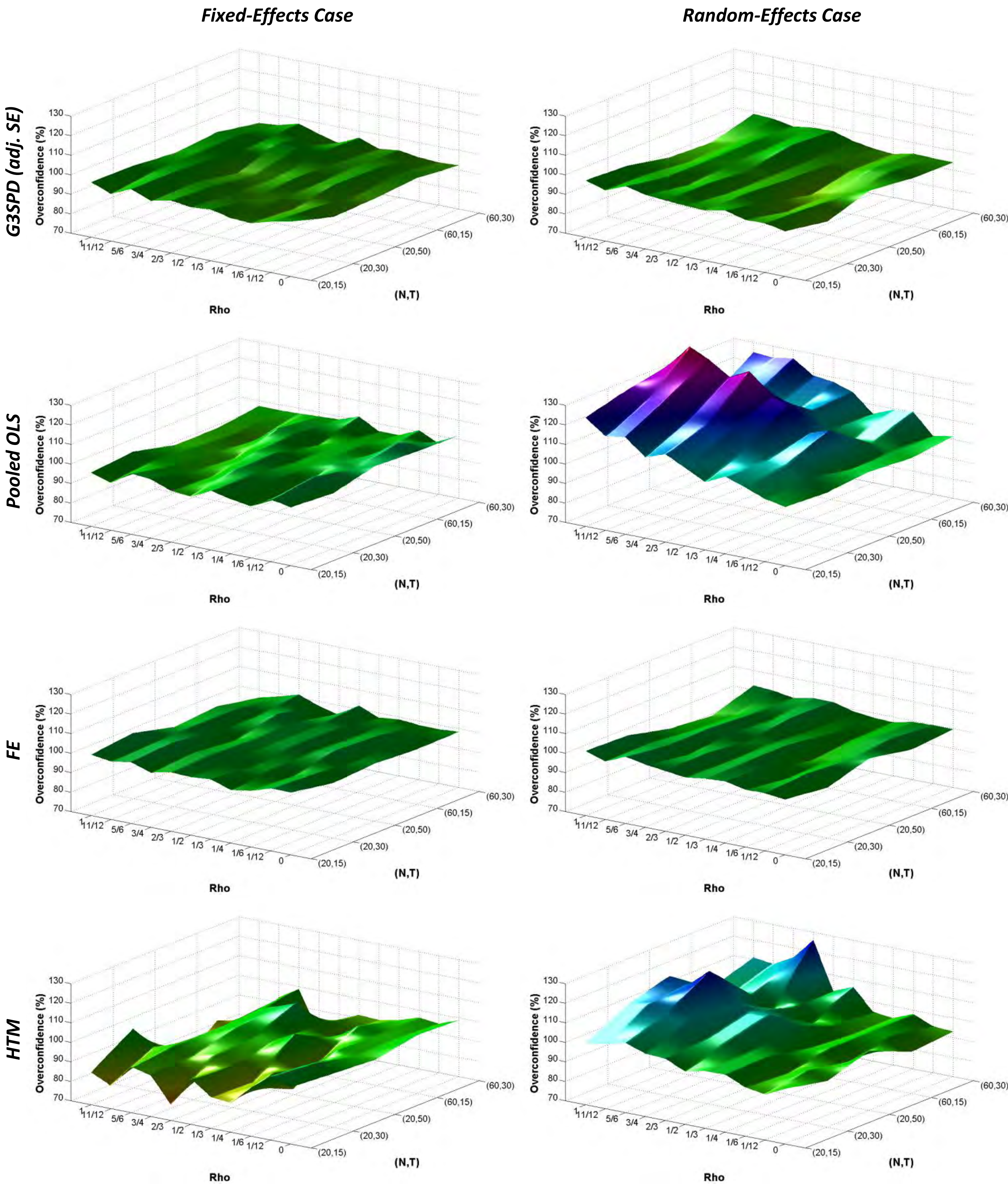


Figure 2.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (Y_2) in Equation 1, equation-by-equation estimation.

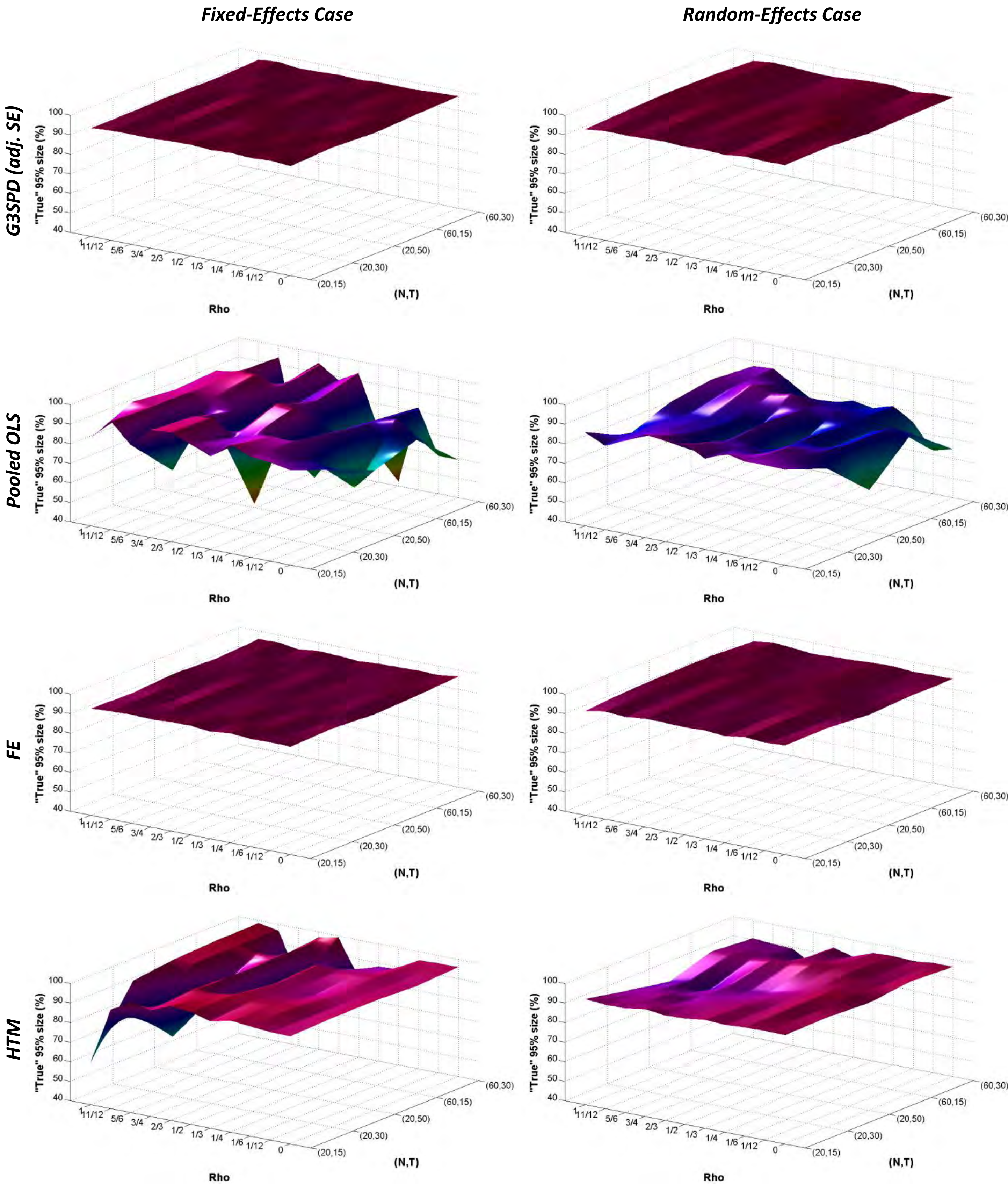


Figure 3.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (Z_2) in Equation 1, equation-by-equation estimation.

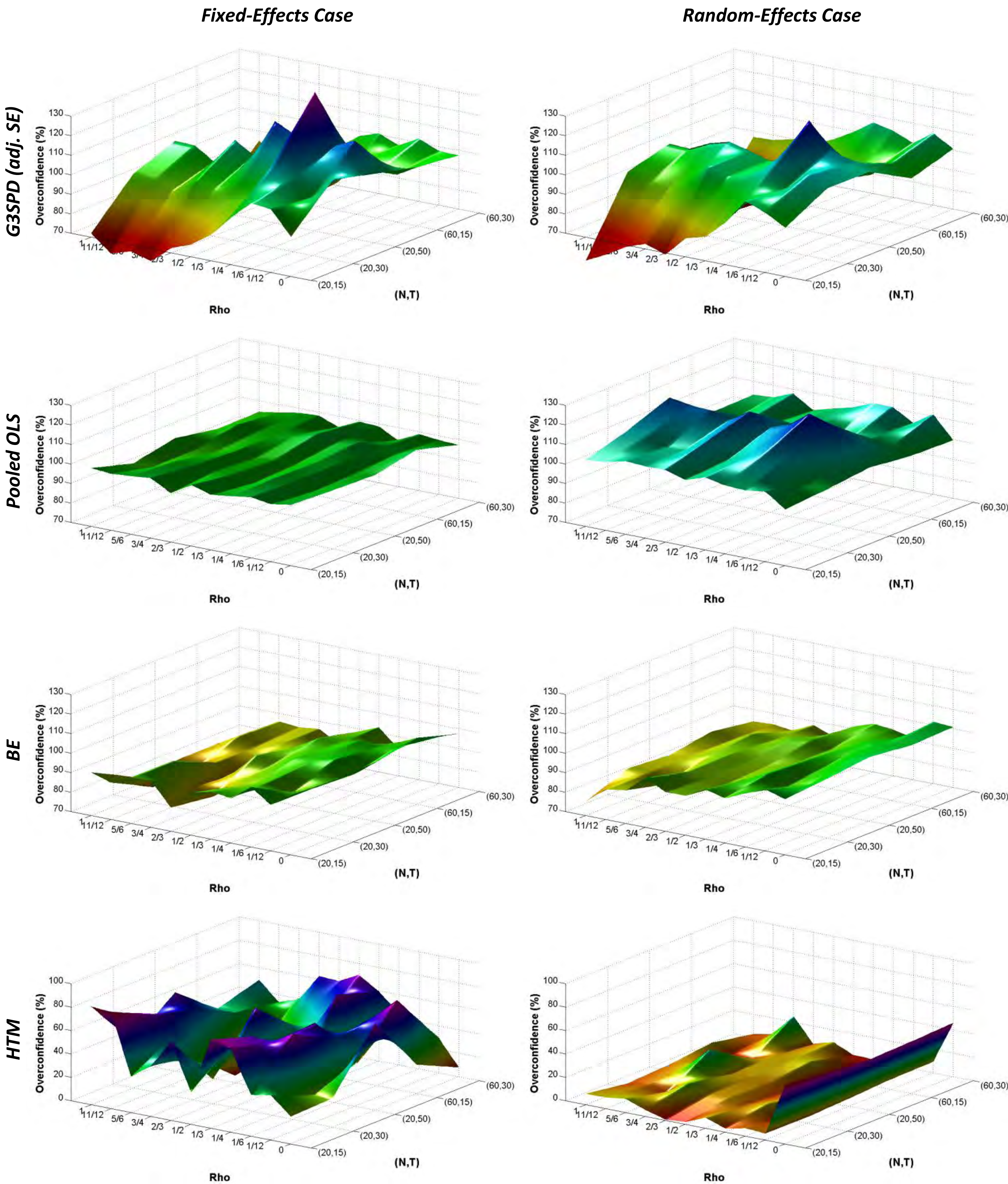


Figure 4.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (Z_2) in Equation 1, equation-by-equation estimation.

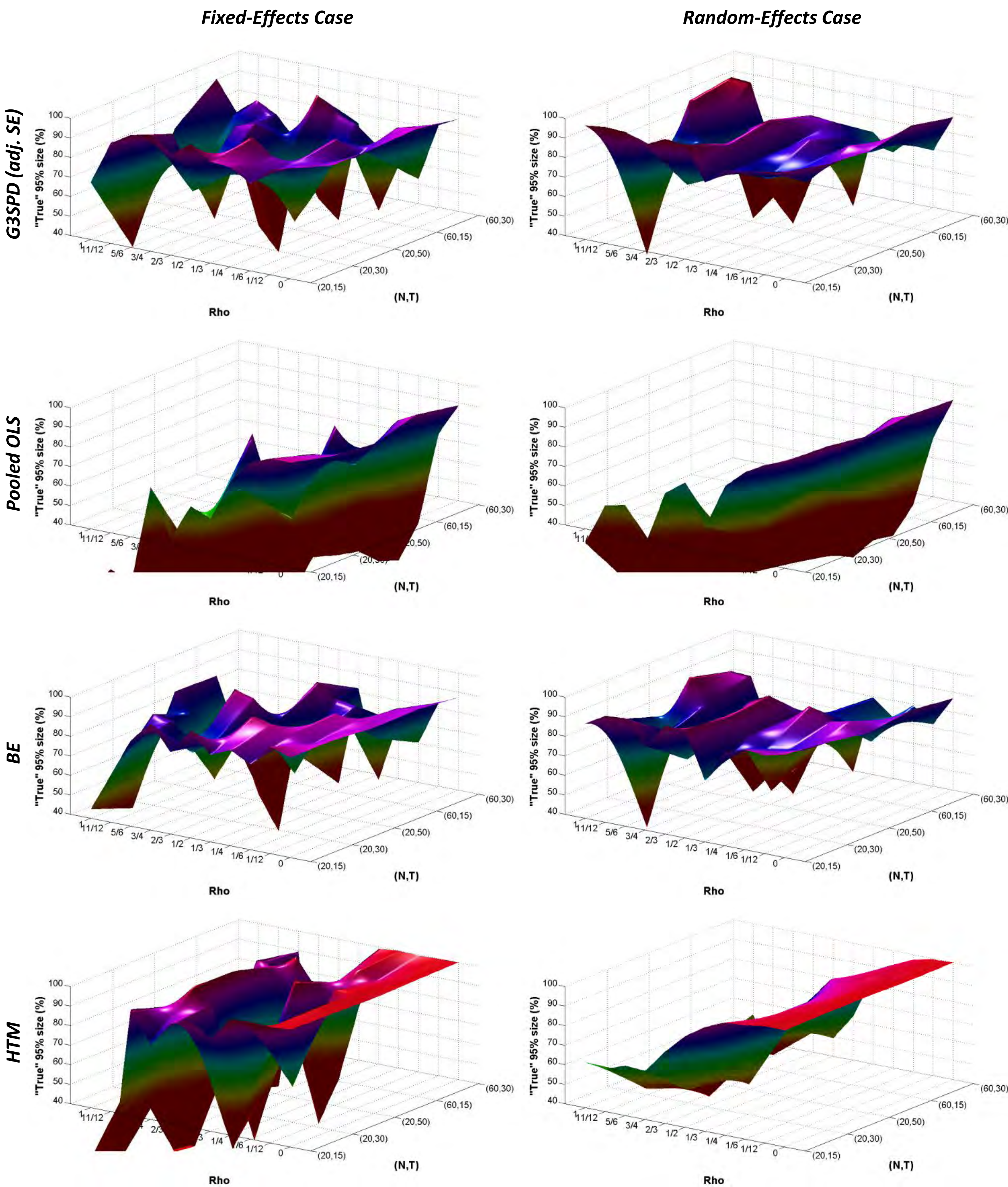


Figure 5.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (X_3) in Equation 1, equation-by-equation estimation.

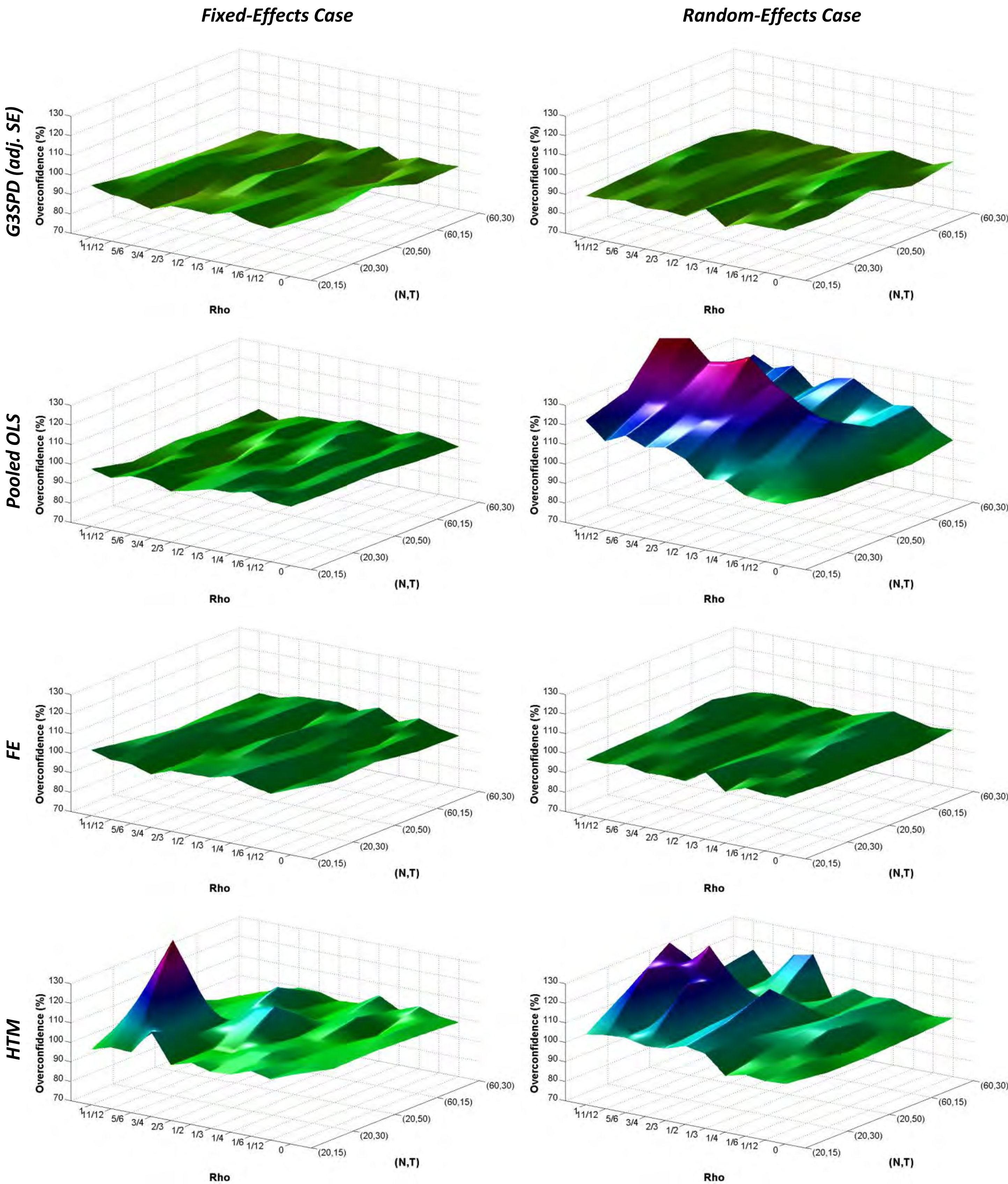


Figure 6.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (X_3) in Equation 1, equation-by-equation estimation.

Fixed-Effects Case | ***Random-Effects Case***

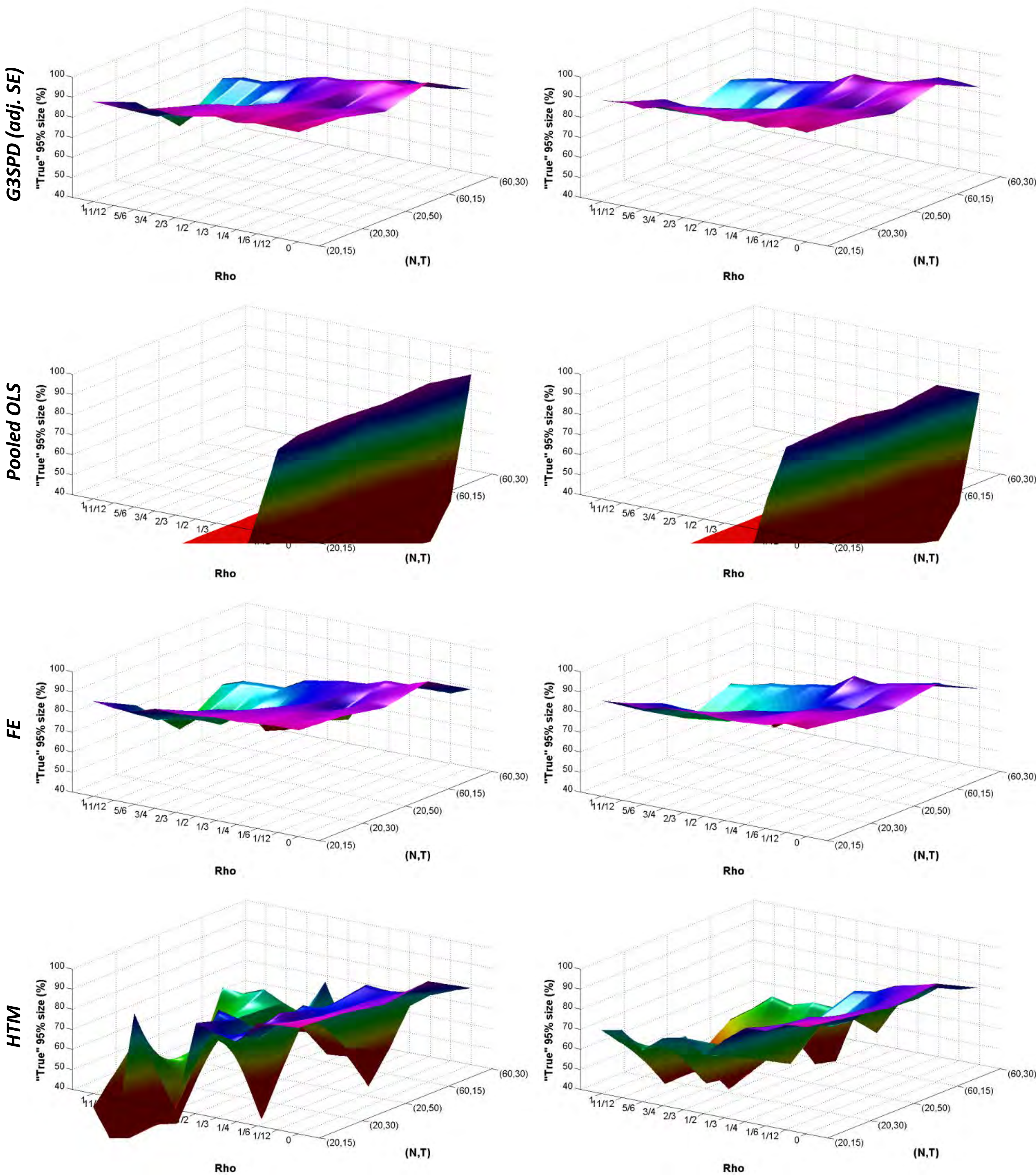


Figure 7.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (Y_2) in Equation 1, system estimation.

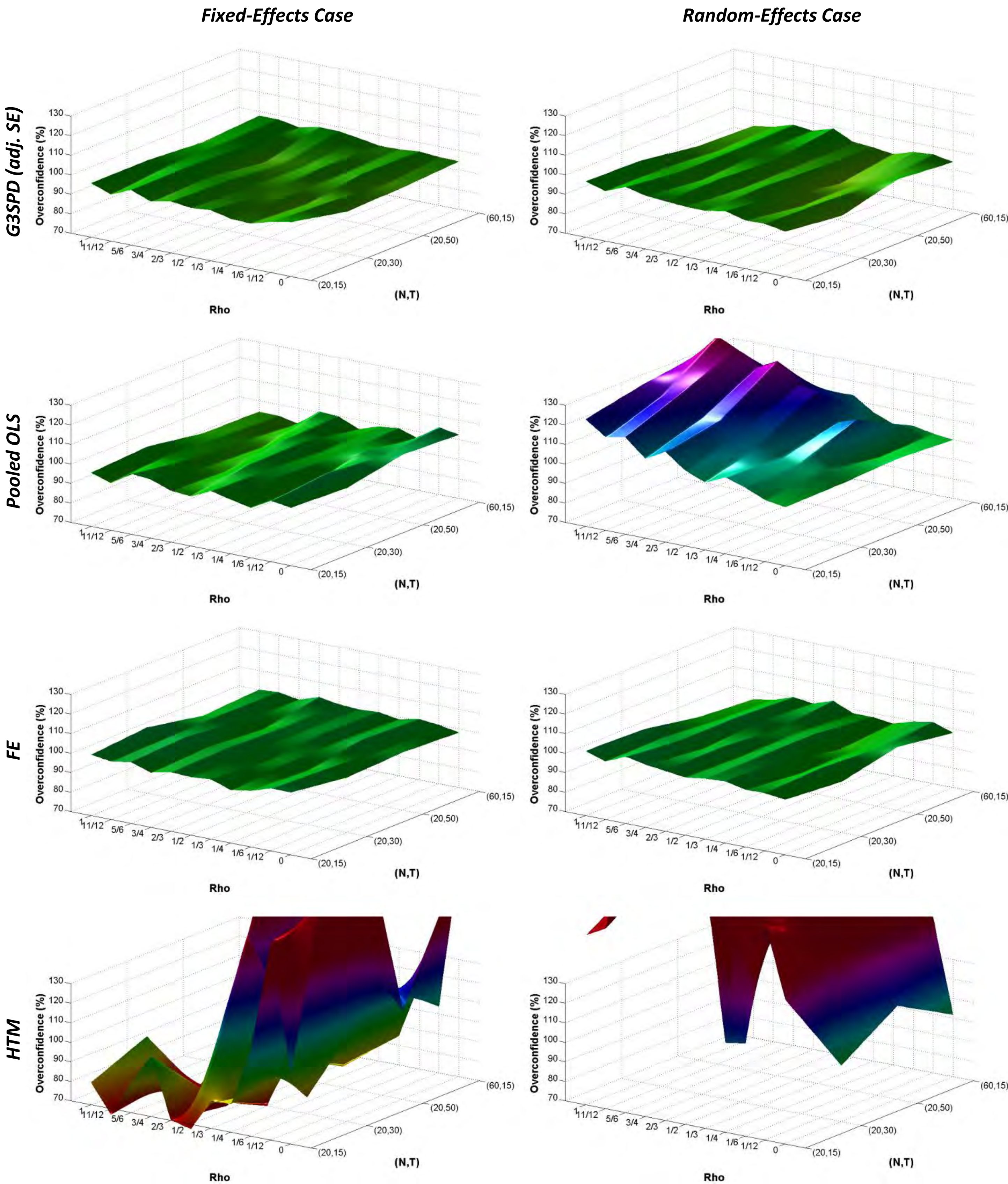


Figure 8.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (Y_2) in Equation 1, system estimation.

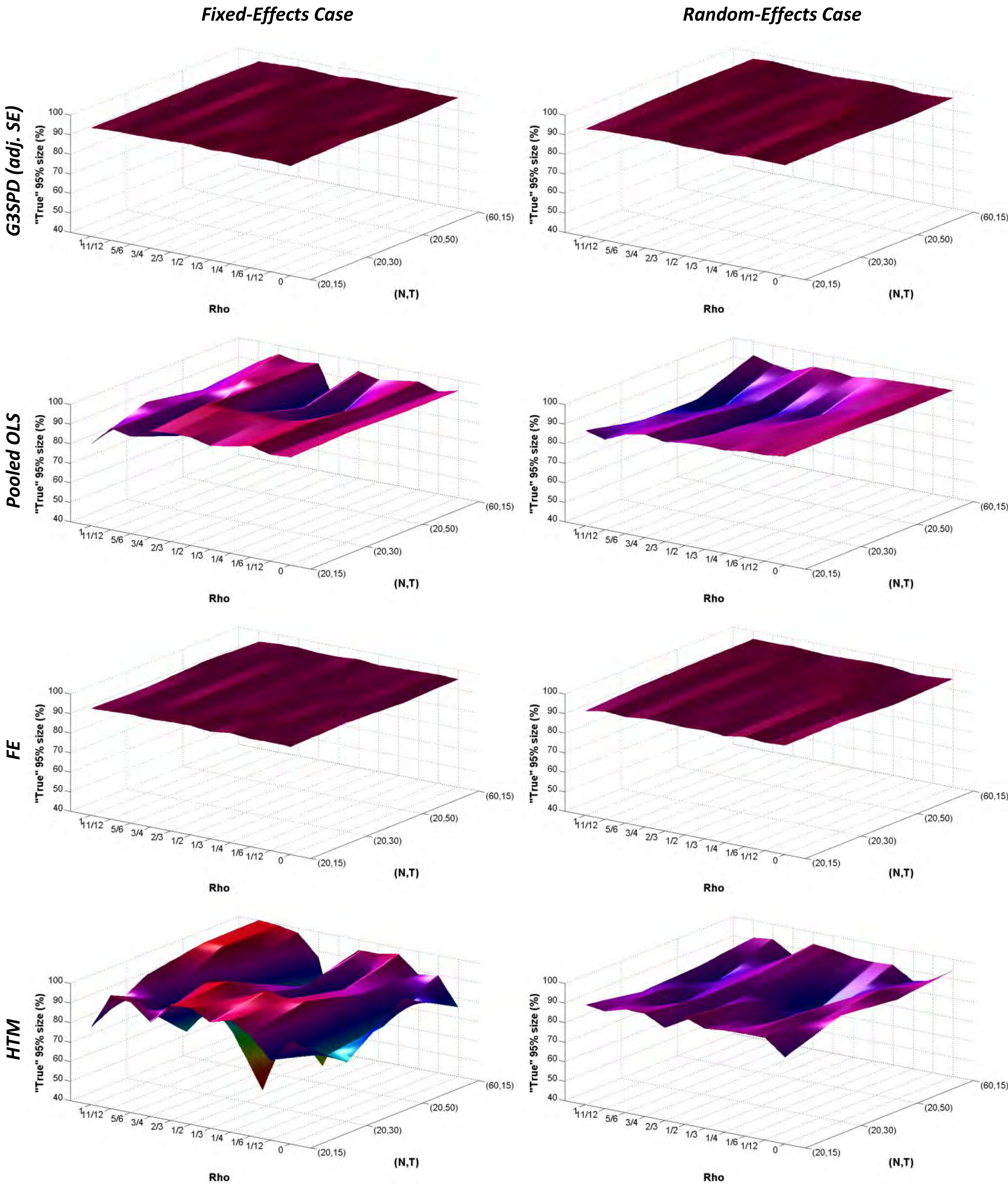


Figure 9.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (Z_2) in Equation 1, system estimation.

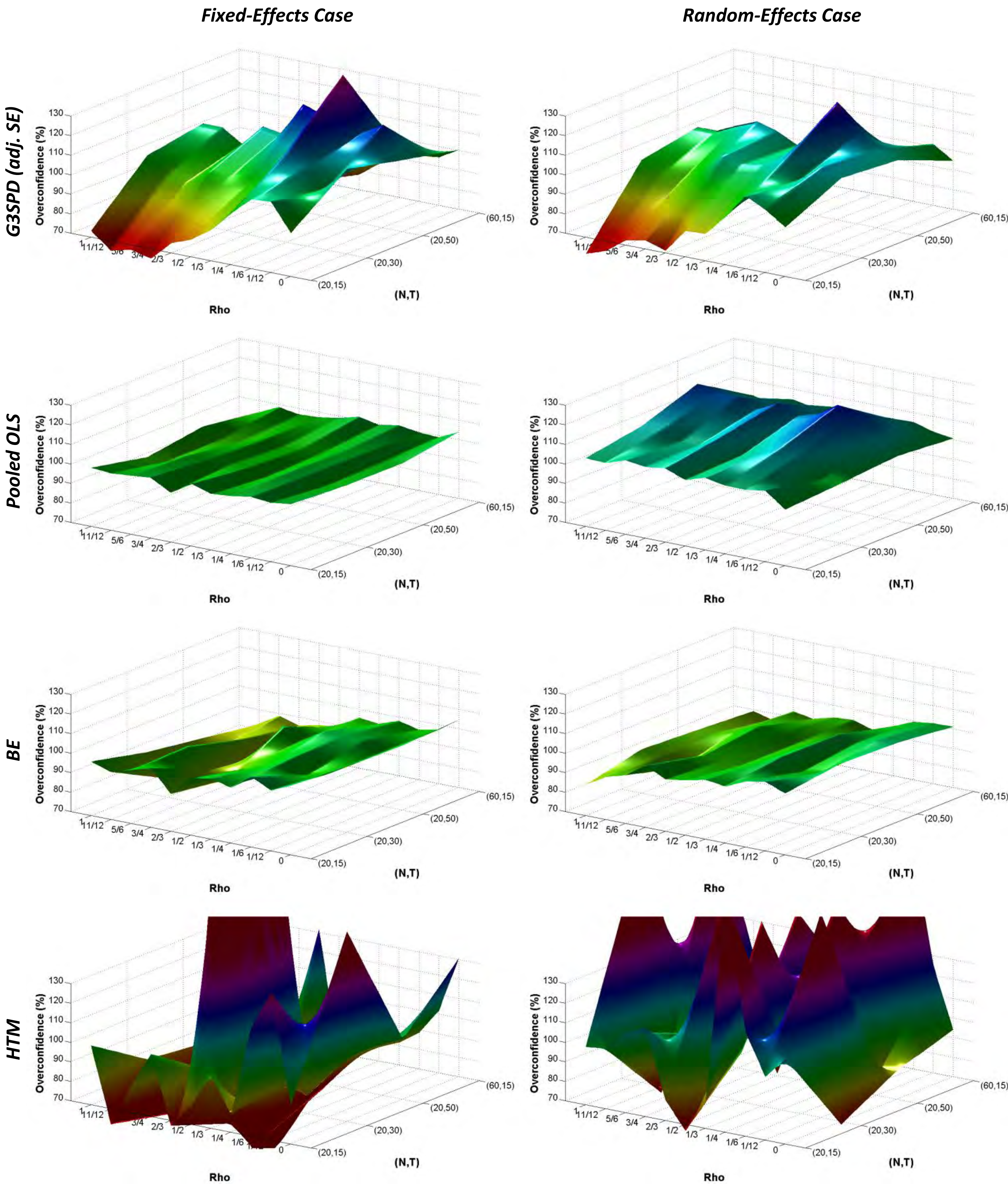


Figure 10.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (Z_2) in Equation 1, system estimation.

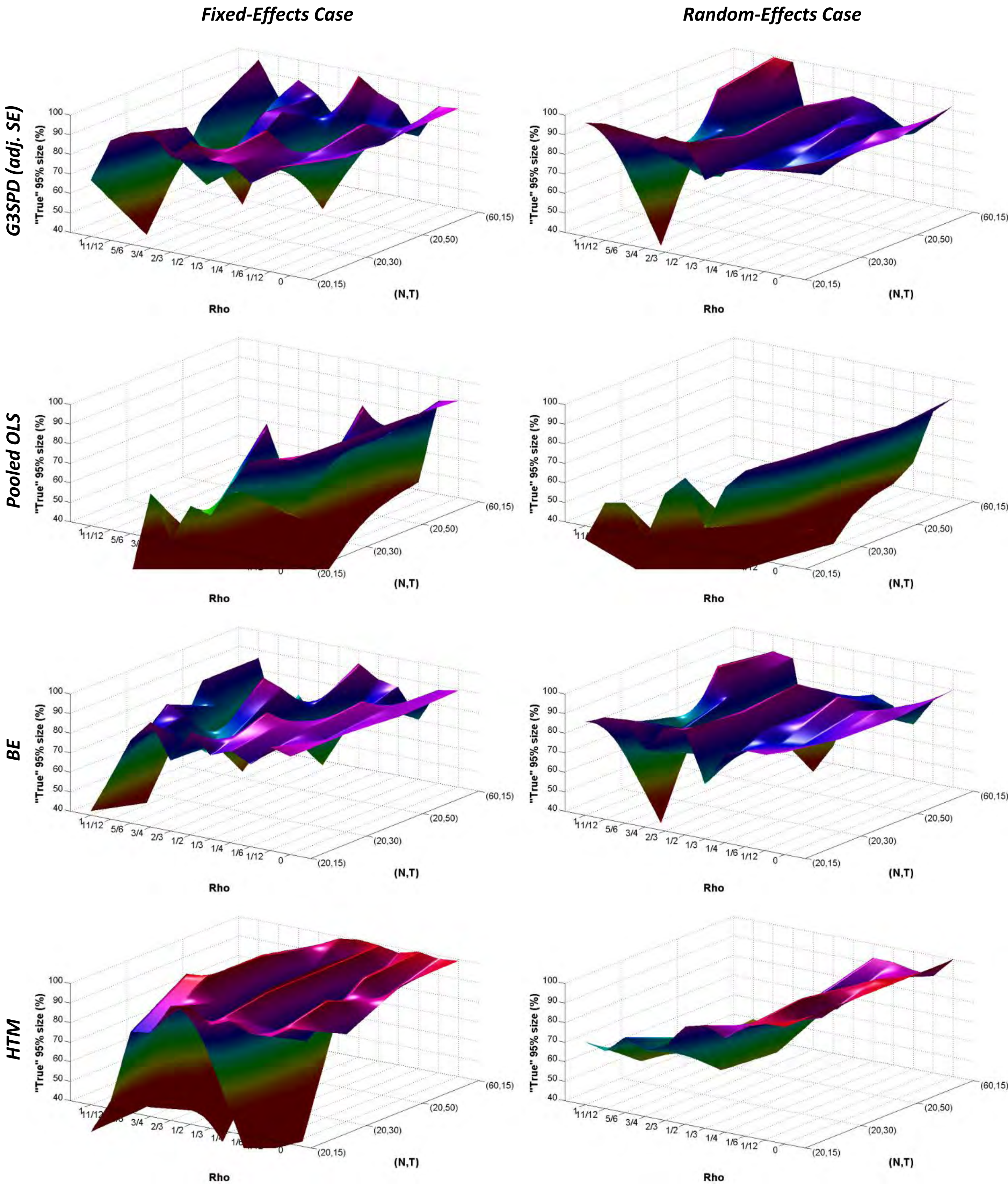


Figure 11.- Overconfidence of the estimated standard errors for the coefficient of an endogenous variable (X_3) in Equation 1, system estimation.

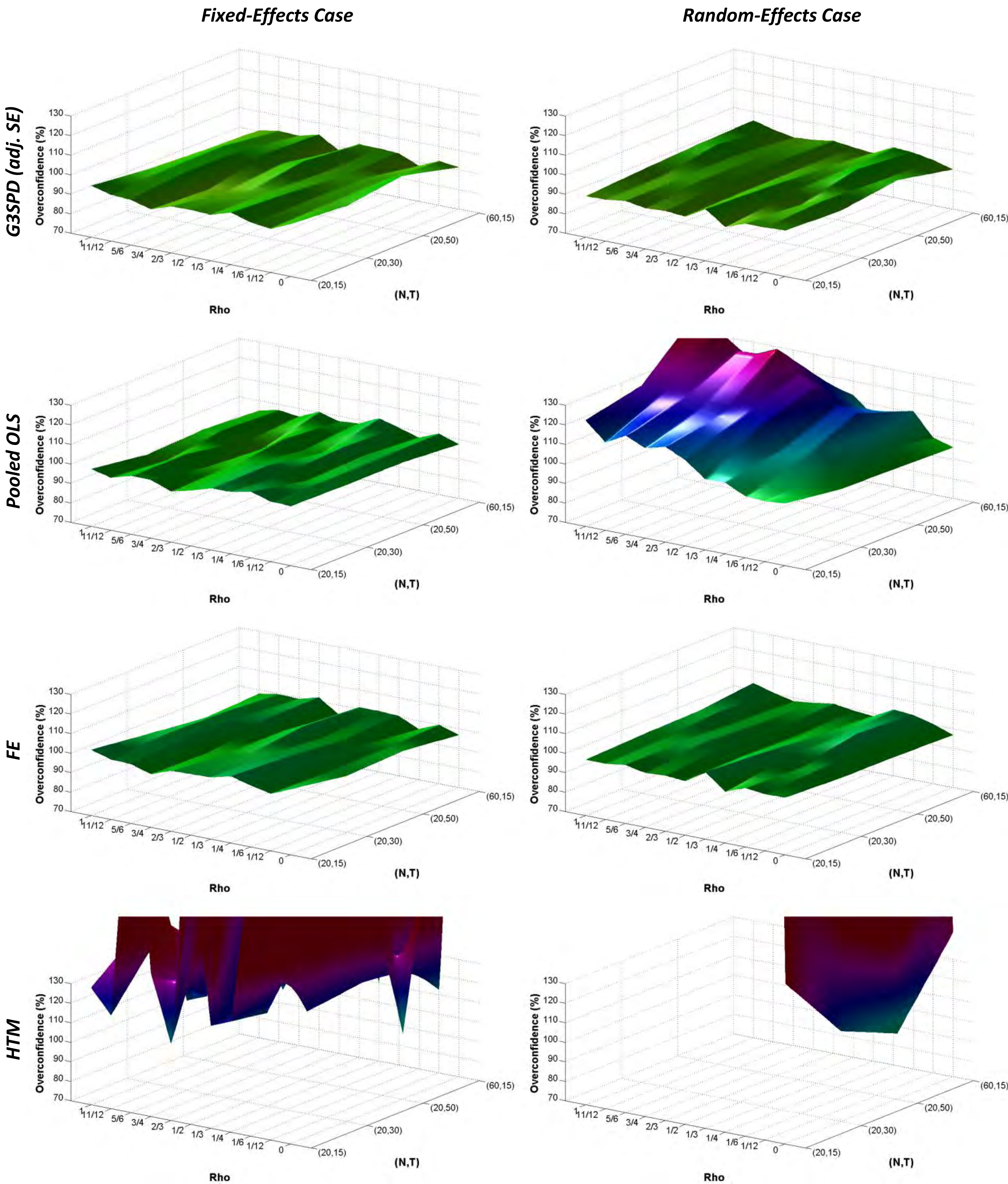
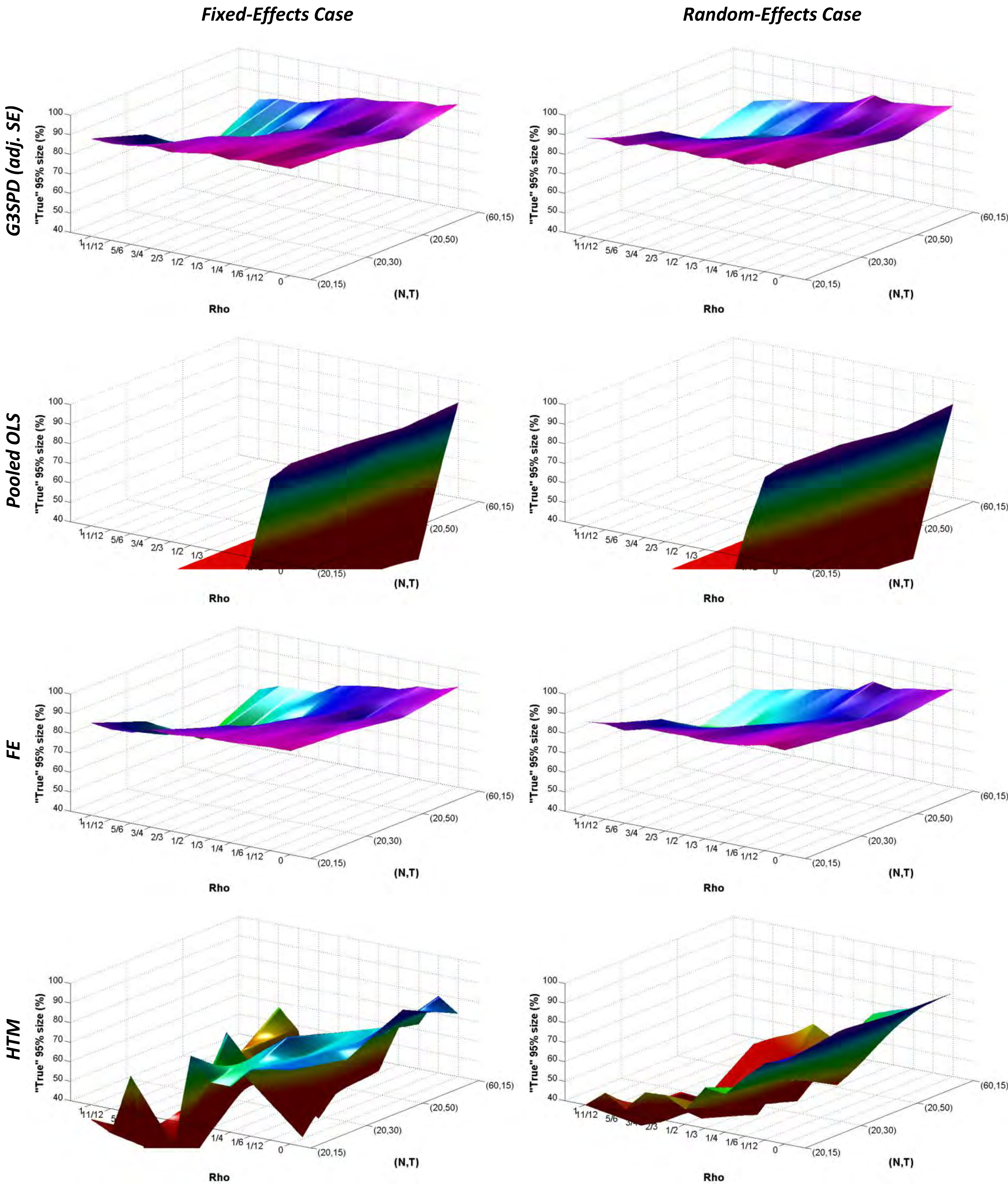


Figure 12.- True size of the estimated 95% confidence interval for the coefficient of an endogenous variable (X_3) in Equation 1, system estimation.



Part II

Software

Chapter 3

A Generalized Three-Step Panel Data Estimator: Software Appendix

3.1 General Comments

The set of programs used to estimate all the methods and reported statistics in this thesis has been compilated in this Appendix. The programs are separated according to the design of the Monte Carlo experiment. I generated two samples, one assuming the panel effects are fixed, and the other assuming them to be random.

The estimation of all available methods (even those not reported in the thesis itself), has been performed estimating first the first equation of the system, then the second equation, and finally, estimating the whole system. The reason for this is that the system estimation uses information from equation-by-equation estimation. Additional procedures have been used to estimate the coefficients and standard errors. They are included in a separate set.

All the estimated coefficients and standard errors have been grouped using a set of programs to gather that information. The reported statistics for the whole set of estimations are obtained from the compiled information.

As can be seen, the programs were adjusted several times to fit all the theoretical and empirical restrictions. Executing them in the correct order contributes to obtain the correct set of estimators. Below there is a list of the programs separated by topics (and when convenient, with the proper number to order the execution). The actual code can be found in the following pages in the same sequence.

3.2 Sample Generation

- generador_FE.m
- generador_RE.m

3.3 Estimation of Equation 1

The following programs, run in order, obtain the estimated coefficients reported in this thesis.

1. estimaeq1_v1.m
2. estimaeq1_v2.m
3. estimaeq1_v3.m
4. estimaeq1_v4.m
5. estimaeq1_v5.m
6. estimaeq1_v6.m
7. estimaeq1_v7.m
8. estimaeq1_alter_v1.m

9. estimaeq1_alter_v2.m
10. estimaeq1_alter_v3.m
11. estimaeq1_alter_v4.m

3.4 Estimation of Equation 2

The following programs, run in order, obtain the estimated coefficients reported in this thesis.

1. estimaeq2_v1.m
2. estimaeq2_v2.m
3. estimaeq2_v3.m
4. estimaeq2_v4.m
5. estimaeq2_v5.m
6. estimaeq2_v6.m
7. estimaeq2_v7.m
8. estimaeq2_alter_v1.m
9. estimaeq2_alter_v2.m
10. estimaeq2_alter_v3.m
11. estimaeq2_alter_v4.m

3.5 System Estimation

The following programs, run in order, obtain the estimated coefficients reported in this thesis.

1. estimasys_v1.m
2. estimasys_v2.m
3. estimasys_v3.m
4. estimasys_v4.m
5. estimasys_v5.m
6. estimasys_alter_v4.m
7. estimasys_hetero.m
8. estimateVARI_alter.m
9. estimateVARI_hetero.m
10. estimateVARI_homo.m

3.6 Additional Estimation Procedures

- SA.m
- G2SLS.m
- G2SLSM.m

- G3SLS2.m
- G3SLSM.m
- G3SLSM_hetero.m

3.7 Information Compilation

- compila.m
- compila_eq1.m
- compila_eq1_newvar.m
- compila_eq2.m
- compila_eq2_newvar.m
- compila_sys.m
- compila_sys_newvar.m

3.8 Statistics Calculation

- stats_eq1.m
- stats_eq1_newvar.m
- stats_eq2.m
- stats_eq2_newvar.m
- stats_sys.m

- stats_sys_newvar.m

```
%=======================================================================
% Function GENERADOR_FE
% This program generates Monte Carlo samples assuming Random Effects
% Author: Diego Avanzini
%===========================================================================

function generador_FE
clc
clear

%single equation
G=2; %number of equations
nt=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rho=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]; %b/w ratio
sigmae=3./(1.+rho);
sigmamu=3.-sigmae;

%coefficients
delta=[1,-.200;0.055,1];
beta=[8,12;2,0;5,0;0,7;0,3];
gamma=[7,6;5,0;0,5];
varphi=[8,0;0,4];

for i=1:1:size(nt,1)
    for k=1:1:size(rho,2)
        clear datos
        N=nt(i,1);
        T=nt(i,2);
        NT=N*T;
        Zmu=kron(eye(N),ones(T,1));

        %generate individual effects
        FE1=normrnd(0,sigmamu(1,k),N,1);
        FE2=normrnd(0,sigmamu(1,k),N,1);
        mu1=sum(Zmu*FE1,2);
        mu2=sum(Zmu*FE2,2);

        %generate X
        phi1=unifrnd(-2,2,N,1);
        phi2=unifrnd(-2,2,N,1);
        phi4=unifrnd(-2,2,N,1);
        eta1=unifrnd(-2,2,NT,1);
        eta2=unifrnd(-2,2,NT,1);
        eta3=unifrnd(-2,2,NT,1);
        eta4=unifrnd(-2,2,NT,1);
        eta5=unifrnd(-2,2,NT,1);
        X1=zeros(NT,1);
        X2=zeros(NT,1);
        X3=zeros(NT,1);
        X4=zeros(NT,1);
        X5=zeros(NT,1);
        for n=1:1:N
```

```
            % define initial values of series for each individual
            X1((n-1)*T+1,1)=(eta1((n-1)*T+1,1)/((1-0.5^2)^0.5))+(phi1(n,1)/(1-0.5));
            X2((n-1)*T+1,1)=(eta2((n-1)*T+1,1)/((1-0.2^2)^0.5))+(phi2(n,1)/(1-0.2));
            X3((n-1)*T+1,1)=(eta3((n-1)*T+1,1)/((1-0.7^2)^0.5))+(FE1(n,1)/(1-0.7));
            X4((n-1)*T+1,1)=(eta4((n-1)*T+1,1)/((1-0.2^2)^0.5))+(phi4(n,1)/(1-0.2));
            X5((n-1)*T+1,1)=(eta5((n-1)*T+1,1)/((1-0.7^2)^0.5))+(FE2(n,1)/(1-0.7));
            % generate the rest of each serie
            for t=2:1:T
                X1((n-1)*T+t,1)=0.5*X1((n-1)*T+t-1,1)+phi1(n,1)+eta1((n-1)*T+t,1);
                X2((n-1)*T+t,1)=0.2*X2((n-1)*T+t-1,1)+phi2(n,1)+eta2((n-1)*T+t,1);
                X3((n-1)*T+t,1)=0.7*X3((n-1)*T+t-1,1)+FE1(n,1)+eta3((n-1)*T+t,1);
                X4((n-1)*T+t,1)=0.2*X4((n-1)*T+t-1,1)+phi4(n,1)+eta4((n-1)*T+t,1);
                X5((n-1)*T+t,1)=0.7*X5((n-1)*T+t-1,1)+FE2(n,1)+eta5((n-1)*T+t,1);
            end
        end

        % generate Z
        xi1=unifrnd(-2,2,N,1);
        xi2=unifrnd(-2,2,N,1);
        xi3=unifrnd(-2,2,N,1);
        Z1=sum(Zmu*xi1,2);
        Z2=sum(Zmu*(FE1+xi2),2);
        Z3=sum(Zmu*(FE2+xi3),2);

        % generate dependent variables
        H=[X1,X2,X3,X4,X5]*beta;
        H=H+([Z1,Z2,Z3]*gamma);
        H=H+(ones(NT,2)*varphi);
        H=H+[mu1,mu2];

        %correlations
        correl.X3mu1=corr(X3,mu1);
        correl.Z2mu1=corr(Z2,mu1);
        correl.X5mu2=corr(X5,mu2);
        correl.Z3mu2=corr(Z3,mu2);

        % save base variables for each case
        filename=['base_' num2str(i,'%3.0f') '_' num2str(k,'%3.0f')];
        save(filename,'N','T','FE1','FE2','mu1','mu2','phi1','phi2','phi4',...
            'eta1','eta2','eta3','eta4','eta5','X1','X2','X3','X4','X5',...
            'xi1','xi2','xi3','Z1','Z2','Z3','H','correl');
        disp([filename ' guardado. Continuar...']);
        clear('FE1','FE2','mu1','mu2','phi1','phi2','phi4','eta1','eta2',...
            'eta3','eta4','eta5','X1','X2','X3','X4','X5','xi1','xi2',...
            'xi3','Z1','Z2','Z3','correl');

        % add stochastic error
        for g=1:1:1000
            e1=normrnd(0,sigmae(1,k),NT,1);
            e2=normrnd(0,sigmae(1,k),NT,1);
            datos(1,g).Y=(H+[e1,e2])*delta;
        end
```

```
        filename=['case_' num2str(i,'%3.0f') '_' num2str(k,'%3.0f')];
        save(filename,'datos');
        disp([filename ' Saved. Continue...']);
    end
end
end
```

```
%=======================================================================
% Function GENERADOR_RE
% This program generates Monte Carlo samples assuming Random Effects
% Author: Diego Avanzini
%===========================================================================

function generador_RE
% This small program generate the set of covariates for the basic set
% of experiments.

clc
clear
tic
G=2; %number of equations
ntcases=single([20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]); %↙
combinations of (N,T)
rhocases=single([0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]); %b/w ratio
sigmae=3./(1.+rhocases);
sigmamu=3.-sigmae;

%coefficients
delta=single([1,-.200;0.055,1]);
beta=single([8,12;2,0;5,0;0,7;0,3]); %time-varying vars.
gamma=single([7,6;5,0;0,5]);             %time-invariant vars.
varphi=single([8,0;0,4]);                       %constant


for nt=1:1:size(ntcases,1) %loop over number of individuals
% for nt=1:1:3 %loop over number of individuals

    for rho=1:1:size(rhocases,2) %loop over b/w ratio
%     for rho=1:1:3 %loop over b/w ratio

        clear('datos','H');
        N=ntcases(nt,1);
        T=ntcases(nt,2);
        NT=N*T;
        Zmu=kron(eye(N,'single'),ones(T,1,'single'));

        % Generate fixed variables: X1, X2, X4, Z1
        %=========================================================
        phi1=unifrnd(-2,2,N,1); %for X1
        phi2=unifrnd(-2,2,N,1); %for X2
        phi4=unifrnd(-2,2,N,1); %for X4
        eta1=unifrnd(-2,2,NT,1); %initial value for X1
        eta2=unifrnd(-2,2,NT,1); %initial value for X2
        eta4=unifrnd(-2,2,NT,1); %initial value for X4
        X1=zeros(NT,1);
        X2=zeros(NT,1);
        X4=zeros(NT,1);
        %generates fixed Xs
        for n=1:1:N
```

```
            % define initial values of series for each individual
            X1((n-1)*T+1,1)=(eta1((n-1)*T+1,1)/((1-0.5^2)^0.5))+(phi1(n,1)/(1-0.5));
            X2((n-1)*T+1,1)=(eta2((n-1)*T+1,1)/((1-0.2^2)^0.5))+(phi2(n,1)/(1-0.2));
            X4((n-1)*T+1,1)=(eta4((n-1)*T+1,1)/((1-0.2^2)^0.5))+(phi4(n,1)/(1-0.2));
            % generate the rest of the periods of each serie
            for t=2:1:T
                X1((n-1)*T+t,1)=0.5*X1((n-1)*T+t-1,1)+phi1(n,1)+eta1((n-1)*T+t,1);
                X2((n-1)*T+t,1)=0.2*X2((n-1)*T+t-1,1)+phi2(n,1)+eta2((n-1)*T+t,1);
                X4((n-1)*T+t,1)=0.2*X4((n-1)*T+t-1,1)+phi4(n,1)+eta4((n-1)*T+t,1);
            end
        end
        % generate fixed Z
        xi1=unifrnd(-2,2,N,1);
        Z1=sum(Zmu*xi1,2);


        % Generate changing variables: X3, X5, Z2, Z3, RE1, RE2
        %=================================================================
        %generate individual random effects
        RE1=single(normrnd(0,sigmamu(1,rho),N,1000)); %individual effects for the 1st.↙
equa
        RE2=single(normrnd(0,sigmamu(1,rho),N,1000)); %individual effects for the 2nd.↙
equa
        %generate X
        %initial values are fixed for all observations 1..1000
        eta3=unifrnd(-2,2,NT,1);
        eta5=unifrnd(-2,2,NT,1);
        X3=zeros(NT,1000);
        X5=zeros(NT,1000);
        for obs=1:1:1000
            for n=1:1:N
                % define initial values of series for each individual
                X3((n-1)*T+1,obs)=(eta3((n-1)*T+1,1)/((1-0.7^2)^0.5))+(RE1(n,obs)/(1-↙
0;7)
                X5((n-1)*T+1,obs)=(eta5((n-1)*T+1,1)/((1-0.7^2)^0.5))+(RE2(n,obs)/(1-↙
0;7)
                % generate the rest of each serie
                for t=2:1:T
                    X3((n-1)*T+t,obs)=0.7*X3((n-1)*T+t-1,obs)+RE1(n,obs)+eta3((n-1)*T+t,↙
1);
                    X5((n-1)*T+t,obs)=0.7*X5((n-1)*T+t-1,obs)+RE2(n,obs)+eta5((n-1)*T+t,↙
1);
                end
            end
        end
        % generate Z
        %initial values are fixed for all observations 1..1000
        xi2=unifrnd(-2,2,N,1);
        xi3=unifrnd(-2,2,N,1);
        Z2=zeros(NT,1000);
        Z3=zeros(NT,1000);
        for obs=1:1:1000
```

```
            Z2(:,obs)=sum(Zmu*(RE1(:,obs)+xi2),2);
            Z3(:,obs)=sum(Zmu*(RE2(:,obs)+xi3),2);
        end
        % generates dependent variables
        H=zeros(NT,2,1000,'single');
        for obs=1:1:1000
            %generate individual random effect
            mu1=sum(Zmu*RE1(:,obs),2);
            mu2=sum(Zmu*RE2(:,obs),2);
            %generate esencial part of dependent variables Y
            H(:,:,obs)=[X1,X2,X3(:,obs),X4,X5(:,obs)]*beta;
            H(:,:,obs)=H(:,:,obs)+([Z1,Z2(:,obs),Z3(:,obs)]*gamma);
            H(:,:,obs)=H(:,:,obs)+(ones(NT,2,'single')*varphi); %initialization with a↙
const per equation
            H(:,:,obs)=H(:,:,obs)+[mu1,mu2];
        end

        % Save variables for each case
        %============================================================
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        save(filename,'N','T','RE1','RE2','phi1','phi2','phi4',...
            'eta1','eta2','eta3','eta4','eta5','X1','X2','X3','X4','X5',...
            'xi1','xi2','xi3','Z1','Z2','Z3','H');
        disp([filename ' guardado. Continuar...']);
        clear('RE1','RE2','phi1','phi2','phi4','eta1','eta2',...
            'eta3','eta4','eta5','X1','X2','X3','X4','X5','xi1','xi2',...
            'xi3','Z1','Z2','Z3');

        for obs=1:1:1000 %loop over 1000 iterations
            %generate error
            e1=normrnd(0,sigmae(1,rho),NT,1); %error for the 1st. equation
            e2=normrnd(0,sigmae(1,rho),NT,1); %error for the 2nd. equation
            % obtain Y
            datos(1,obs).Y=single((H(:,:,obs)+[e1,e2])*delta); %endogenous vars.: each↙
column is the dependet var. of an equation
        end
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        save(filename,'datos');
        disp([filename ' guardado. Continuar...']);
    end
end
toc
```

```
%=======================================================================
% Function ESTIMAEQ1_V1
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%===========================================================================

function estimaeq1_v1
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '========================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '========================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    D=single(kron(eye(N),ones(1,T)*(1/T))); %reduce matrix from NT*1 to N*1

   for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3); mu1=single(mu1); mu2=single(mu2);
        %create new folder for containing 1000 obs.
        if exist(['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')],'dir')
~=7
            mkdir(['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end

        %predetermined matrixes
        %----------------------
        W=eye(N*T);
        P=eye(N*T);
        Wwave=Qmu;
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        Wbar=Pmu;
```

```
        A=Qmu*([X1,X2,X3,X4,X5]); %within instruments
        PA=A*inv(A'*A)*A';
        B=Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]); %between instruments
        PB=B*inv(B'*B)*B';
        PAB=PA+PB; %projection matrix of instruments
        BHT=Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]; %between instruments
        if rank(BHT)<size(BHT,2);BHT=BHT(:,1:rank(BHT));end
        PBHT=BHT*inv(BHT'*BHT)*BHT';
        PABHT=PA+PBHT; %projection matrix of instruments
        BAM=Pmu*[SA([X1,X2],N,T),SA([X4,X5],N,T),Z1,Z3,ones(N*T,1)]; %between↙
instruments
        if rank(BAM)<size(BAM,2);BAM=BAM(:,1:rank(BAM));end
        PBAM=BAM*inv(BAM'*BAM)*BAM';
        PABAM=PA+PBAM; %projection matrix of instruments
        BBMS=Pmu*[SA([X1,X2],N,T),SA([X4,X5],N,T),Z1,Z3,SA(Qmu*X3,N,T),ones(N*T,1)]; %↙
between instruments
        if rank(BBMS)<size(BBMS,2);BBMS=BBMS(:,1:rank(BBMS));end
        PBBMS=BBMS*inv(BBMS'*BBMS)*BBMS';
        PABBMS=PA+PBBMS; %projection matrix of instruments

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1));
            Y=single(datos(1,obs).Y(:,2));
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
endogenous
            K=size(R,2);
            Rwave=[Y X1 X2 X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);
            clear rdo

%===========================================================================
            %Method 1: OLS
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,eye(N*T),eye(N*T),y,N,T);
            rdoeq1(1).coef=[ NaN(1,2);         %y1: empty
                                 rdo.coef(1,:);    %y2
                                 rdo.coef(2,:);    %X1
                                 rdo.coef(3,:);    %X2
                                 rdo.coef(4,:);    %X3
                                 NaN(2,2);         %X4,X5: empty
                                 rdo.coef(5,:);    %Z1
                                 rdo.coef(6,:);    %Z2
                                 NaN(1,2);         %Z3: empty
                                 rdo.coef(7,:);    %phi1
                                 NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(1).resi=rdo.resi;
            rdoeq1(1).stats=rdo.stats;

%===========================================================================
            %Method 2: FEStd
%===========================================================================
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(Rwave,Qmu,eye(N*T),y,N,T);
            betaFEStd=rdo(1).coef(:,1); %to be used in ECAm
            rdoeq1(2).coef=[ NaN(1,2);          %y1: empty
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
                                rdo.coef(4,:);    %X3
                                NaN(9,2)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq1(2).resi=rdo.resi;
            rdoeq1(2).stats=rdo.stats;

%===========================================================================
            %Method 3: FEBal
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(Rwave,Qmu,PA,y,N,T);
            rdoeq1(3).coef=[ NaN(1,2);          %y1: empty
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
                                rdo.coef(4,:);    %X3
                                NaN(9,2)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq1(3).resi=rdo.resi;
            rdoeq1(3).stats=rdo.stats;

%===========================================================================
            %Method 4: BEStd
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,Pmu,eye(N*T),y,N,T);
            rdoeq1(4).coef=[ NaN(1,2);          %y1: empty
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
                                rdo.coef(4,:);    %X3
                                NaN(2,2);         %X4,X5: empty
                                rdo.coef(5,:);    %Z1
                                rdo.coef(6,:);    %Z2
                                NaN(1,2);         %Z3: empty
                                rdo.coef(7,:);    %phi1
                                NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(4).resi=rdo.resi;
            rdoeq1(4).stats=rdo.stats;

%===========================================================================
            %Method 5: BEBal
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,Pmu,PB,y,N,T);
            rdoeq1(5).coef=[ NaN(1,2);          %y1: empty
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
```

```
                                   rdo.coef(4,:);    %X3
                                   NaN(2,2);         %X4,X5: empty
                                   rdo.coef(5,:);    %Z1
                                   rdo.coef(6,:);    %Z2
                                   NaN(1,2);         %Z3: empty
                                   rdo.coef(7,:);    %phi1
                                   NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(5).resi=rdo.resi;
            rdoeq1(5).stats=rdo.stats;

%==========================================================================
            %Method 6: ECWH
%==========================================================================
            u=y-R*inv(R'*R)*R'*y; %OLS residuals
            sigma1=((u'*Qmu*u)/(N*(T-1)))^0.5; %"within" component of variance
            sigma2=((u'*Pmu*u)/N)^0.5; %"between" component of variance
            WWH=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % omega^(-1/2)
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WWH,eye(N*T),y,N,T);
            rdoeq1(6).coef=[ NaN(1,2);         %y1: empty
                                   rdo.coef(1,:);    %y2
                                   rdo.coef(2,:);    %X1
                                   rdo.coef(3,:);    %X2
                                   rdo.coef(4,:);    %X3
                                   NaN(2,2);         %X4,X5: empty
                                   rdo.coef(5,:);    %Z1
                                   rdo.coef(6,:);    %Z2
                                   NaN(1,2);         %Z3: empty
                                   rdo.coef(7,:);    %phi1
                                   NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(6).resi=rdo.resi;
            rdoeq1(6).stats=rdo.stats;

%==========================================================================
            %Method 7: ECAm
%==========================================================================
            alpha=mean(D*y)-mean(D*Rwave)*betaFEStd; %constant term
            u=y-alpha*ones(N*T,1)-Rwave*betaFEStd; %residuals of a LSDV regression
            sigma1=((u'*Qmu*u)/(N*(T-1)))^0.5; %"within" component of variance
            sigma2=((u'*Pmu*u)/N)^0.5; %"between" component of variance
            WAm=(1/sigma1)*Qmu+(1/sigma2)*Pmu; %omega ^ (-1/2)
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WAm,eye(N*T),y,N,T);
            rdoeq1(7).coef=[ NaN(1,2);         %y1: empty
                                   rdo.coef(1,:);    %y2
                                   rdo.coef(2,:);    %X1
                                   rdo.coef(3,:);    %X2
                                   rdo.coef(4,:);    %X3
                                   NaN(2,2);         %X4,X5: empty
                                   rdo.coef(5,:);    %Z1
                                   rdo.coef(6,:);    %Z2
                                   NaN(1,2);         %Z3: empty
                                   rdo.coef(7,:);    %phi1
                                   NaN(3,2)];        %phi2,mu1,mu2: empty
```

```
            rdoeq1(7).resi=rdo.resi;
            rdoeq1(7).stats=rdo.stats;

%===========================================================================
            %Method 8: ECSA
%===========================================================================
            %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu*y;
            ewave=squeeze(rdoeq1(2).resi);
            sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
            %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*R)*R'*Pmu*y;
            ebar=squeeze(rdoeq1(4).resi);
            sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
            WSA=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WSA,eye(N*T),y,N,T);
            rdoeq1(8).coef=[ NaN(1,2);         %y1: empty
                                   rdo.coef(1,:);    %y2
                                   rdo.coef(2,:);    %X1
                                   rdo.coef(3,:);    %X2
                                   rdo.coef(4,:);    %X3
                                   NaN(2,2);         %X4,X5: empty
                                   rdo.coef(5,:);    %Z1
                                   rdo.coef(6,:);    %Z2
                                   NaN(1,2);         %Z3: empty
                                   rdo.coef(7,:);    %phi1
                                   NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(8).resi=rdo.resi;
            rdoeq1(8).stats=rdo.stats;

%===========================================================================
            %Method 9: ECCorn
%===========================================================================
            %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
            ewave=rdoeq1(3).resi; %taken from FEBal
            sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
            %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
            ebar=rdoeq1(5).resi; %taken from BEBal
            sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
            WCorn=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WCorn,PAB,y,N,T);
            rdoeq1(9).coef=[ NaN(1,2);         %y1: empty
                                   rdo.coef(1,:);    %y2
                                   rdo.coef(2,:);    %X1
                                   rdo.coef(3,:);    %X2
                                   rdo.coef(4,:);    %X3
                                   NaN(2,2);         %X4,X5: empty
                                   rdo.coef(5,:);    %Z1
                                   rdo.coef(6,:);    %Z2
                                   NaN(1,2);         %Z3: empty
                                   rdo.coef(7,:);    %phi1
                                   NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(9).resi=rdo.resi;
            rdoeq1(9).stats=rdo.stats;
```

```
%===========================================================================
            %Method 10: HTHT
%===========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=((ebarHT'*Pmu*ebarHT)/(N-K-1))^0.5;
            WHT=(1/sigma1)*Qmu+(1/sigma2HT)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WHT,PABHT,y,N,T);
            rdoeq1(10).coef=[ NaN(1,2);         %y1: empty
                                    rdo.coef(1,:);    %y2
                                    rdo.coef(2,:);    %X1
                                    rdo.coef(3,:);    %X2
                                    rdo.coef(4,:);    %X3
                                    NaN(2,2);         %X4,X5: empty
                                    rdo.coef(5,:);    %Z1
                                    rdo.coef(6,:);    %Z2
                                    NaN(1,2);         %Z3: empty
                                    rdo.coef(7,:);    %phi1
                                    NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(10).resi=rdo.resi;
            rdoeq1(10).stats=rdo.stats;

%===========================================================================
            %Method 11: HTAM
%===========================================================================
            ebarAM=Pmu*y-Pmu*R*inv(R'*Pmu*PBAM*Pmu*R)*R'*Pmu*PBAM*Pmu*y;
            sigma2AM=((ebarAM'*Pmu*ebarAM)/(N-K-1))^0.5;
            WAM=(1/sigma1)*Qmu+(1/sigma2AM)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WAM,PABAM,y,N,T);
            rdoeq1(11).coef=[ NaN(1,2);         %y1: empty
                                    rdo.coef(1,:);    %y2
                                    rdo.coef(2,:);    %X1
                                    rdo.coef(3,:);    %X2
                                    rdo.coef(4,:);    %X3
                                    NaN(2,2);         %X4,X5: empty
                                    rdo.coef(5,:);    %Z1
                                    rdo.coef(6,:);    %Z2
                                    NaN(1,2);         %Z3: empty
                                    rdo.coef(7,:);    %phi1
                                    NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(11).resi=rdo.resi;
            rdoeq1(11).stats=rdo.stats;

%===========================================================================
            %Method 12: HTBMS
%===========================================================================
            ebarBMS=Pmu*y-Pmu*R*inv(R'*Pmu*PBBMS*Pmu*R)*R'*Pmu*PBBMS*Pmu*y;
            sigma2BMS=((ebarBMS'*Pmu*ebarBMS)/(N-K-1))^0.5;
            WBMS=(1/sigma1)*Qmu+(1/sigma2BMS)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WBMS,PABBMS,y,N,T);
            rdoeq1(12).coef=[ NaN(1,2);         %y1: empty
                                    rdo.coef(1,:);    %y2
```

```
                                                rdo.coef(2,:);    %X1
                                                rdo.coef(3,:);    %X2
                                                rdo.coef(4,:);    %X3
                                                NaN(2,2);         %X4,X5: empty
                                                rdo.coef(5,:);    %Z1
                                                rdo.coef(6,:);    %Z2
                                                NaN(1,2);         %Z3: empty
                                                rdo.coef(7,:);    %phi1
                                                NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(12).resi=rdo.resi;
            rdoeq1(12).stats=rdo.stats;

%==========================================================================
            %Method 13,14,15: G3SPD_FEStd
%==========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G2SLS(Rwave,Qmu,eye(N*T),y,N,T);
            rdoeq1(13).coef=[ NaN(1,2);            %y1: empty
                                                rdo(1).coef(1,:);    %y2
                                                rdo(1).coef(2,:);    %X1
                                                rdo(1).coef(3,:);    %X2
                                                rdo(1).coef(4,:);    %X3
                                                NaN(9,2)];           %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,↙
mu2: empty
            rdoeq1(13).resi=rdo(1).resi;
            rdoeq1(13).stats=rdo(1).stats;

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %vector of mean residuals (dependent↙
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,eye(N*T),uhat,N,T);
            rdoeq1(14).coef=[ NaN(7,2);            %y1,y2,X1,X2,X3,X4,X5: empty
                                                rdo(2).coef(1,:);    %Z1
                                                rdo(2).coef(2,:);    %Z2
                                                NaN(1,2);            %Z3: empty
                                                rdo(2).coef(3,:);    %phi1
                                                NaN(3,2)];           %phi2,mu1,mu2: empty
            rdoeq1(14).resi=rdo(2).resi;
            rdoeq1(14).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdo(2).resi,ones(N*T,1)]; %full dataset, including↙
all the original vars. plus the estimated Fixed Effects
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),eye(N*T),y,N,T,↙
-N*T);
            rdoeq1(15).coef=[ NaN(1,4);            %y1: empty
                                                rdo(3).coef(1,:);    %y2
                                                rdo(3).coef(2,:);    %X1
                                                rdo(3).coef(3,:);    %X2
                                                rdo(3).coef(4,:);    %X3
                                                NaN(2,4);            %X4,X5: empty
                                                rdo(3).coef(5,:);    %Z1
```

```
                                    rdo(3).coef(6,:);     %Z2
                                    NaN(1,4);             %Z3: empty
                                    rdo(3).coef(8,:);     %phi1
                                    NaN(1,4);             %phi2: empty
                                    rdo(3).coef(7,:);     %mu1
                                    NaN(1,4)];            %mu2: empty
            rdoeq1(15).resi=rdo(3).resi;
            rdoeq1(15).stats=rdo(3).stats;

%===========================================================================
            %Method 16,17,18: G3SPD_FECorn
%===========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G2SLS(Rwave,Qmu,PA,y,N,T);
            rdoeq1(16).coef=[ NaN(1,2);             %y1: empty
                                    rdo(1).coef(1,:);     %y2
                                    rdo(1).coef(2,:);     %X1
                                    rdo(1).coef(3,:);     %X2
                                    rdo(1).coef(4,:);     %X3
                                    NaN(9,2)];            %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq1(16).resi=rdo(1).resi;
            rdoeq1(16).stats=rdo(1).stats;

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PB,uhat,N,T);
            rdoeq1(17).coef=[ NaN(7,2);             %y1,y2,X1,X2,X3,X4,X5: empty
                                    rdo(2).coef(1,:);     %Z1
                                    rdo(2).coef(2,:);     %Z2
                                    NaN(1,2);             %Z3: empty
                                    rdo(2).coef(3,:);     %phi1
                                    NaN(3,2)];            %phi2,mu1,mu2: empty
            rdoeq1(17).resi=rdo(2).resi;
            rdoeq1(17).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdo(2).resi,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T,(N*T-8));
            rdoeq1(18).coef=[ NaN(1,4);             %y1: empty
                                    rdo(3).coef(1,:);     %y2
                                    rdo(3).coef(2,:);     %X1
                                    rdo(3).coef(3,:);     %X2
                                    rdo(3).coef(4,:);     %X3
                                    NaN(2,4);             %X4,X5: empty
                                    rdo(3).coef(5,:);     %Z1
                                    rdo(3).coef(6,:);     %Z2
```

```
                                    NaN(1,4);               %Z3: empty
                                    rdo(3).coef(8,:);       %phi1
                                    NaN(1,4);               %phi2: empty
                                    rdo(3).coef(7,:);       %mu1
                                    NaN(1,4)];              %mu2: empty
            rdoeq1(18).resi=rdo(3).resi;
            rdoeq1(18).stats=rdo(3).stats;

%===========================================================================
            %Method 19,20: G3SPD_HTHT
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent↙
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBHT,uhat,N,T);
            rdoeq1(19).coef=[ NaN(7,2);             %y1,y2,X1,X2,X3,X4,X5: empty
                                    rdo(2).coef(1,:);       %Z1
                                    rdo(2).coef(2,:);       %Z2
                                    NaN(1,2);               %Z3: empty
                                    rdo(2).coef(3,:);       %phi1
                                    NaN(3,2)];              %phi2,mu1,mu2: empty
            rdoeq1(19).resi=rdo(2).resi;
            rdoeq1(19).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdo(2).resi,ones(N*T,1)]; %full dataset, including↙
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T,(N*T-↙
8));
            rdoeq1(20).coef=[ NaN(1,4);             %y1: empty
                                    rdo(3).coef(1,:);       %y2
                                    rdo(3).coef(2,:);       %X1
                                    rdo(3).coef(3,:);       %X2
                                    rdo(3).coef(4,:);       %X3
                                    NaN(2,4);               %X4,X5: empty
                                    rdo(3).coef(5,:);       %Z1
                                    rdo(3).coef(6,:);       %Z2
                                    NaN(1,4);               %Z3: empty
                                    rdo(3).coef(8,:);       %phi1
                                    NaN(1,4);               %phi2: empty
                                    rdo(3).coef(7,:);       %mu1
                                    NaN(1,4)];              %mu2: empty
            rdoeq1(20).resi=rdo(3).resi;
            rdoeq1(20).stats=rdo(3).stats;

%===========================================================================
            %Method 21,22: G3SPD_HTAM
```

```
%======================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBAM,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar)); ----> ajuste de
varianza     erroneo!!!!!
            rdoeq1(21).coef=[ NaN(7,2);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    rdo(2).coef(1,:);    %Z1
                                    rdo(2).coef(2,:);    %Z2
                                    NaN(1,2);            %Z3: empty
                                    rdo(2).coef(3,:);    %phi1
                                    NaN(3,2)];           %phi2,mu1,mu2: empty
            rdoeq1(21).resi=rdo(2).resi;
            rdoeq1(21).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdo(2).resi,ones(N*T,1)]; %full dataset, including
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq1(22).coef=[ NaN(1,4);            %y1: empty
                                    rdo(3).coef(1,:);    %y2
                                    rdo(3).coef(2,:);    %X1
                                    rdo(3).coef(3,:);    %X2
                                    rdo(3).coef(4,:);    %X3
                                    NaN(2,4);            %X4,X5: empty
                                    rdo(3).coef(5,:);    %Z1
                                    rdo(3).coef(6,:);    %Z2
                                    NaN(1,4);            %Z3: empty
                                    rdo(3).coef(8,:);    %phi1
                                    NaN(1,4);            %phi2: empty
                                    rdo(3).coef(7,:);    %mu1
                                    NaN(1,4)];           %mu2: empty
            rdoeq1(22).resi=rdo(3).resi;
            rdoeq1(22).stats=rdo(3).stats;

%==========================================================================
            %Method 23,24: G3SPD_HTBMS
%==========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBBMS,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar)); ----> ajuste de
```

```
varianza     erroneo!!!!!
            rdoeq1(23).coef=[ NaN(7,2);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    rdo(2).coef(1,:);    %Z1
                                    rdo(2).coef(2,:);    %Z2
                                    NaN(1,2);            %Z3: empty
                                    rdo(2).coef(3,:);    %phi1
                                    NaN(3,2)];           %phi2,mu1,mu2: empty
            rdoeq1(23).resi=rdo(2).resi;
            rdoeq1(23).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdo(2).resi,ones(N*T,1)]; %full dataset, including
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq1(24).coef=[ NaN(1,4);            %y1: empty
                                    rdo(3).coef(1,:);    %y2
                                    rdo(3).coef(2,:);    %X1
                                    rdo(3).coef(3,:);    %X2
                                    rdo(3).coef(4,:);    %X3
                                    NaN(2,4);            %X4,X5: empty
                                    rdo(3).coef(5,:);    %Z1
                                    rdo(3).coef(6,:);    %Z2
                                    NaN(1,4);            %Z3: empty
                                    rdo(3).coef(8,:);    %phi1
                                    NaN(1,4);            %phi2: empty
                                    rdo(3).coef(7,:);    %mu1
                                    NaN(1,4)];           %mu2: empty
            rdoeq1(24).resi=rdo(3).resi;
            rdoeq1(24).stats=rdo(3).stats;

            % Save results
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');
            keyboard
        end
        toc
    end
end
```

```
%==========================================================================
% Function ESTIMAEQ1_V2
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%============================================================================

function estimaeq1_v2
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    D=single(kron(eye(N),ones(1,T)*(1/T))); %reduce matrix from NT*1 to N*1

    for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3); mu1=single(mu1); mu2=single(mu2);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
            invZPZ=inv(Rbar'*Pmu*Rbar); % this matrix is the same for all 1000 observations and all methods!!!
            PZ_invZPZ_ZP=Pmu*Rbar*invZPZ*Rbar'*Pmu; % this matrix is the same for all 1000 observations and all methods!!!

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1));
            Y=single(datos(1,obs).Y(:,2));
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are
```

```
genous
            Rwave=[Y X1 X2 X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            %open data for the observation
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            load(filename,'rdoeq1');

%=========================================================================
            %Method 13,14,15: G3SPD_FEStd
%=========================================================================
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq1(14).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq1(14).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    stdernew(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stdernew(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty

            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2,ones(N*T,1)]; %full dataset, including all the
orig vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq1(15).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq1(15).coef(13,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq1(15).coef(13,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq1(15).coef(:,5)=[ NaN(1,1);            %y1: empty
                            stderMT(1,:);    %y2
                            stderMT(2,:);    %X1
                            stderMT(3,:);    %X2
                            stderMT(4,:);    %X3
                            NaN(2,1);            %X4,X5: empty
                            stderMT(5,:);    %Z1
                            stderMT(6,:);    %Z2
                            NaN(1,1);            %Z3: empty
                            stderMT(8,:);    %phi1
                            NaN(1,1);            %phi2: empty
                            stderMT(7,:);    %mu1
                            NaN(1,1)];           %mu2: empty

%=========================================================================
            %Method 16,17,18: G3SPD_FECorn
%=========================================================================
```

```
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq1(17).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq1(17).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    stdernew(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stdernew(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty

            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2,ones(N*T,1)]; %full dataset, including all the
original vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq1(18).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq1(18).coef(13,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq1(18).coef(13,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq1(18).coef(:,5)=[ NaN(1,1);            %y1: empty
                            stderMT(1,:);    %y2
                            stderMT(2,:);    %X1
                            stderMT(3,:);    %X2
                            stderMT(4,:);    %X3
                            NaN(2,1);            %X4,X5: empty
                            stderMT(5,:);    %Z1
                            stderMT(6,:);    %Z2
                            NaN(1,1);            %Z3: empty
                            stderMT(8,:);    %phi1
                            NaN(1,1);            %phi2: empty
                            stderMT(7,:);    %mu1
                            NaN(1,1)];           %mu2: empty

%===========================================================================
            %Method 19,20: G3SPD_HTHT
%===========================================================================
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq1(19).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq1(19).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    stdernew(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stdernew(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty
```

```
            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq1(20).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq1(20).coef(13,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq1(20).coef(13,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq1(20).coef(:,5)=[ NaN(1,1);            %y1: empty
                                stderMT(1,:);    %y2
                                stderMT(2,:);    %X1
                                stderMT(3,:);    %X2
                                stderMT(4,:);    %X3
                                NaN(2,1);            %X4,X5: empty
                                stderMT(5,:);    %Z1
                                stderMT(6,:);    %Z2
                                NaN(1,1);            %Z3: empty
                                stderMT(8,:);    %phi1
                                NaN(1,1);            %phi2: empty
                                stderMT(7,:);    %mu1
                                NaN(1,1)];           %mu2: empty

            % Save results
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');
        keyboard
        end
        toc
    end
end
```

```
%===================================================================
% Function ESTIMAEQ1_V3
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=======================================================================

function estimaeq1_v3
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear('ZMu');

    for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1));
            Y=single(datos(1,obs).Y(:,2));
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are
endogenous
            Rwave=[Y X1 X2 X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
                invXQX=inv(Rwave'*Qmu*Rwave);
                invRPR=inv(R'*Pmu*R);
            %open data for the observation
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
```

```
            load(filename,'rdoeq1');

%==========================================================================
            %Method 2: FEStd
%==========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(2).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq1(2).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%==========================================================================
            %Method 3: FEBal
%==========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(3).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq1(3).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%==========================================================================
            %Method 4: BEStd
%==========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(4).resi); %residuals
            vari=(resi'*resi)/(N-size(R,2));
            stdernew=diag(vari*invRPR).^0.5; %new std. errors
            rdoeq1(4).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stdernew(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty

%==========================================================================
```

```
            %Method 5: BEBal
%==========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(5).resi); %residuals
            vari=(resi'*resi)/(N-size(R,2));
            stdernew=diag(vari*invRPR).^0.5; %new std. errors
            rdoeq1(5).coef(:,2)=[   NaN(1,1);         %y1: empty
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,1);         %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,1);         %Z3: empty
                                stdernew(7,:);    %phi1
                                NaN(3,1)];        %phi2,mu1,mu2: empty

%==========================================================================
            %Method 13: G3SPD_FEStd
%==========================================================================
            % 1st stage: reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(13).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq1(13).coef(:,2)=[  NaN(1,1);         %y1: empty
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2:↙
empty

%==========================================================================
            %Method 16: G3SPD_FECorn
%==========================================================================
            % 1st stage: reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq1(16).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq1(16).coef(:,2)=[  NaN(1,1);         %y1: empty
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2:↙
empty


            % Save results
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
```

```
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');
        end
        toc
    end
end
```

```
%==========================================================================
% Function ESTIMAEQ1_V4
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%==========================================================================

function estimaeq1_v4
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    clear('ZMu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                    Observation: ' num2str(obs)]);
            Y=single(datos(1,obs).Y(:,2));

            %open data for the observation
            filename=['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
```

```
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');

%===========================================================================
            %Method 13,14,15: G3SPD_Std
%===========================================================================
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdoeq1(14).resi,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covariance matrix
            sigma3rd=(rdoeq1(15).resi'*rdoeq1(15).resi)/(N*T-K);
            omegaGMM=sigma3rd*inv(Rfull'*Rfull);

            % WH-HC1: stored in column 6
            sigmaHC1=diag((rdoeq1(15).resi).^2);
            omegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(omegaHC1).^0.5;
            clear('sigmaHC1','omegaHC1');

            % WH-HC2: stored in column 7
            sigmaHC2=diag(((rdoeq1(15).resi).^2)./(1-diag(PR)));
            omegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq1(15).resi)./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            omegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            omegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % BK: stored in column 10
            E=reshape(rdoeq1(15).resi,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix for each time period.
            omegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(omegaBK).^0.5;
            clear('E','sigmaBK','omegaBK');

            % MT covariance matrix: stored in column 11
            sigma2nd=((rdoeq1(14).resi'*rdoeq1(14).resi))/(N-Kbar);
            Psi1=omegaGMM;
```

```
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*rdoeq1(15).resi)*...
                (rdoeq1(14).resi'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            omegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(omegaMT).^0.5;
            clear('sigma3rd','sigma2nd','omegaGMM','Psi1','Psi2','Psi3','omegaMT');

            % store standard errors
            stder=[stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(15).coef(:,6:11)=[   NaN(1,6);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,6);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,6);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,6);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,6)];     %mu2: empty


% %=============================================================================
%             %Method 16,17,18: G3SPD_Corn
% %=============================================================================
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdoeq1(17).resi,ones(N*T,1)]; %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdoeq1(17).resi,ones(N*T,1)]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag(((rdoeq1(18).resi).^2)./(1-diag(PR)))*C;
            omegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq1(18).resi)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            omegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            omegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
```

```
(RCinvCC*C'*Rfull);
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % Shift BK and MT one column to the right: stored in columns 10
            % and 11
            rdoeq1(18).coef(:,10:11)=rdoeq1(18).coef(:,9:10);

            % store standard errors
            stder=[stderHC2,stderLE,stderHC3];
            rdoeq1(18).coef(:,7:9)=[    NaN(1,3);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,3);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,3);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,3);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,3)];     %mu2: empty

% %======================================================================
%              %Method 19,20: G3SPD_HTM
% %======================================================================
            Rfull=[Y,X1,X2,X3,Z1,Z2,rdoeq1(19).resi,ones(N*T,1)]; %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdoeq1(19).resi,ones(N*T,1)]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag(((rdoeq1(20).resi).^2)./(1-diag(PR)))*C;
            omegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq1(20).resi)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            omegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            omegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
```

```
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % Shift BK and MT one column to the right: stored in columns 10
            % and 11
            rdoeq1(20).coef(:,10:11)=rdoeq1(20).coef(:,9:10);

            % store standard errors
            stder=[stderHC2,stderLE,stderHC3];
            rdoeq1(20).coef(:,7:9)=[    NaN(1,3);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,3);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,3);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,3);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,3)];     %mu2: empty

            % Save results
            filename=['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');

        end
        toc
    end
end
```

```
%===================================================================
% Function ESTIMAEQ1_V5
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=====================================================================

function estimaeq1_v5
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        PAB=PA+PB; %projection matrix of instruments
        BHT=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';
```

```
        PABHT=PA+PBHT; %projection matrix of instruments

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
genous
            K=size(R,2);
            Rwave=[Y,X1,X2,X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

%===========================================================================
            %Method 1: OLS
%===========================================================================
            theta=inv(R'*R)*R*y;
            resi=y-R*theta;
            vari=(resi'*resi)/(N*T-K);
            stder=diag(vari*inv(R'*R)).^0.5; %new std. errors
            esti=[theta,stder];
            rdoeq1(1).coef=[ NaN(1,2);         %y1: empty
                                       rdo.coef(1,:);    %y2
                                       rdo.coef(2,:);    %X1
                                       rdo.coef(3,:);    %X2
                                       rdo.coef(4,:);    %X3
                                       NaN(2,2);         %X4,X5: empty
                                       rdo.coef(5,:);    %Z1
                                       rdo.coef(6,:);    %Z2
                                       NaN(1,2);         %Z3: empty
                                       rdo.coef(7,:);    %phi1
                                       NaN(3,2)];        %phi2,mu1,mu2: empty
            rdoeq1(1).resi=rdo.resi;
            rdoeq1(1).stats=rdo.stats;

%===========================================================================
            %Method 2: FEStd
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(Rwave,Qmu,eye(N*T),y,N,T);
            betaFEStd=rdo(1).coef(:,1); %to be used in ECAm
            rdoeq1(2).coef=[ NaN(1,2);         %y1: empty
                               rdo.coef(1,:);    %y2
                               rdo.coef(2,:);    %X1
                               rdo.coef(3,:);    %X2
                               rdo.coef(4,:);    %X3
                               NaN(9,2)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq1(2).resi=resi;
            rdoeq1(2).stats=[];

%===========================================================================
%           Method 3: FEBal
%===========================================================================
            resi=squeeze(rdoeq1(3).resi); %residuals
```

```
            vari=(resi'*Qmu*resi)/(N*(T-1)-Kwave);
            stdernew=diag(vari*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq1(3).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%===========================================================================
%           Method 5: BEBal
%===========================================================================
            resi=squeeze(rdoeq1(5).resi); %residuals
            vari=(resi'*Pmu*resi)/(N-K);
            stdernew=diag(vari*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            rdoeq1(5).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stdernew(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty

%===========================================================================
%           Method 6: ECWH
%===========================================================================
            resi=squeeze(rdoeq1(6).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq1(6).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stdernew(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty

%===========================================================================
%           Method 7: ECAm
%===========================================================================
            resi=squeeze(rdoeq1(7).resi); %residuals
```

```
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq1(7).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stdernew(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty

%==========================================================================
%           Method 8: ECSA
%==========================================================================
            resi=squeeze(rdoeq1(8).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq1(8).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stdernew(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty

%==========================================================================
%           Method 9: ECCorn
%==========================================================================
            resi=squeeze(rdoeq1(9).resi); %residuals
            sigma1=((resi'*Qmu*resi)/(N*(T-1)-K))^0.5; %"within" component of variance
            sigma2=((resi'*Pmu*resi)/(N-K))^0.5; %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1/2)
            stdernew=diag(inv(R'*invsigma*PAB*invsigma*R)).^0.5; %new std. errors
            rdoeq1(9).coef(:,2)=[   NaN(1,1);         %y1: empty
                                   stdernew(1,:);    %y2
                                   stdernew(2,:);    %X1
                                   stdernew(3,:);    %X2
                                   stdernew(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stdernew(5,:);    %Z1
                                   stdernew(6,:);    %Z2
```

```
                                NaN(1,1);           %Z3: empty
                                stdernew(7,:);      %phi1
                                NaN(3,1)];          %phi2,mu1,mu2: empty

%==========================================================================
%           Method 10: HTM
%==========================================================================
            resi=squeeze(rdoeq1(10).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*PABHT*invsigma*R)).^0.5; %new std. errors
            rdoeq1(10).coef(:,2)=[   NaN(1,1);          %y1: empty
                                stdernew(1,:);      %y2
                                stdernew(2,:);      %X1
                                stdernew(3,:);      %X2
                                stdernew(4,:);      %X3
                                NaN(2,1);           %X4,X5: empty
                                stdernew(5,:);      %Z1
                                stdernew(6,:);      %Z2
                                NaN(1,1);           %Z3: empty
                                stdernew(7,:);      %phi1
                                NaN(3,1)];          %phi2,mu1,mu2: empty

%==========================================================================
%           Method 16,17,18: G3SPD_Corn
%==========================================================================
        %1st Step
            resi=single(squeeze(rdoeq1(16).resi)); %residuals
            vari=(resi'*Qmu*resi)/(N*(T-1)-Kwave);
            stdernew=diag(vari*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq1(16).coef(:,2)=[  NaN(1,1);          %y1: empty
                                stdernew(1,:);      %y2
                                stdernew(2,:);      %X1
                                stdernew(3,:);      %X2
                                stdernew(4,:);      %X3
                                NaN(9,1)];          %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2:↙
empty

        % 2nd Step
            resi2nd=single(squeeze(rdoeq1(17).resi)); %residuals 2nd stage
            sigma2nd=(resi2nd'*Pmu*resi2nd)/(N-Kbar);
            stdernew=diag(sigma2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            rdoeq1(17).coef(:,2)=[  NaN(7,1);             %y1,y2,X1,X2,X3,X4,X5: empty
                                     stdernew(1,:);    %Z1
                                     stdernew(2,:);    %Z2
                                     NaN(1,1);            %Z3: empty
                                     stdernew(3,:);    %phi1
                                     NaN(3,1)];           %phi2,mu1,mu2: empty

        % 3rd Step
            resi3rd=single(squeeze(rdoeq1(18).resi)); %residuals 2nd stage
```

```
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all↙
the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';

            % GMM:
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(Rfull'*PC*Rfull);
            stdernew=diag(OmegaGMM).^0.5;
            rdoeq1(18).coef(:,2)=[  NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

            % MT
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PB*Pmu*Rbar*inv↙
(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PB*Pmu*Rbar*inv↙
(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stdernew=diag(OmegaMT).^0.5;
            rdoeq1(18).coef(:,11)=[ NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

%==========================================================================
%           Method 19,20: G3SPD_HTM
%==========================================================================
        % 2nd Step
            resi2nd=single(squeeze(rdoeq1(19).resi)); %residuals 2nd stage
            sigma2nd=(resi2nd'*Pmu*resi2nd)/(N-Kbar);
            stdernew=diag(sigma2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
```

```
            rdoeq1(19).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    stdernew(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stdernew(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty

        % 3rd Step
            resi3rd=single(squeeze(rdoeq1(20).resi)); %residuals 2nd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';

            % GMM:
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(Rfull'*PC*Rfull);
            stdernew=diag(OmegaGMM).^0.5;
            rdoeq1(20).coef(:,2)=[  NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

            % MT
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stdernew=diag(OmegaMT).^0.5;
            rdoeq1(20).coef(:,11)=[ NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
```

```
                                        NaN(1,1)];     %mu2: empty



%===========================================================================
%           Method 25: G3SPD_HTM new!!!
%==========================================================================
        % 3rd Step
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all↙
the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resi2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);

            % coefficients: stored in column 1
            theta=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi3rd=y-Rfull*theta; %residuals

            % GMM: stored in column 2
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1: stored in column 6
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv↙
(RCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('OmegaHC1');

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(resi3rd)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');
```

```
            % BK: stored in column 10
            E=reshape(resi3rd,T,N); %arrange the set of OLS residuals
            VBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix for each time period.
            sigmaBK=C'*VBK*C;
            OmegaBK=inv(Rfull'*PC*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(Rfull'*PC*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','VBK','sigmaBK','OmegaBK');

            % MT: stored in column 11
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;

            % store standard errors and residuals
            stder=[theta,stderGMM,zeros(8,3),stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(25).coef=[NaN(1,11);      %y1: empty
                             stder(1,:);    %y2
                             stder(2,:);    %X1
                             stder(3,:);    %X2
                             stder(4,:);    %X3
                             NaN(2,11);      %X4,X5: empty
                             stder(5,:);    %Z1
                             stder(6,:);    %Z2
                             NaN(1,11);      %Z3: empty
                             stder(8,:);    %phi1
                             NaN(1,11);      %phi2: empty
                             stder(7,:);    %mu1
                             NaN(1,11)];     %mu2: empty
            rdoeq1(25).resi=resi3rd;
            rdoeq1(25).stats=[];


            % Save results
            filename=['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');

        end
        toc
    end
```

```
end
```

```
%===============================================================
% Function ESTIMAEQ1_V6
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%===================================================================

function estimaeq1_v6
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        BHT=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';
```

```
        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                    Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are
endogenous
            K=size(R,2);
            Rwave=[Y,X1,X2,X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

%==========================================================================
%             Method 1: OLS
%==========================================================================
            thetaOLS=inv(R'*R)*R'*y;
            resiOLS=y-R*thetaOLS;
            sigmaOLS=(resiOLS'*resiOLS)/(N*T-K);
            stderOLS=diag(sigmaOLS*inv(R'*R)).^0.5; %new std. errors
            estimadores=[thetaOLS,stderOLS];
            rdoeq1(1).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq1(1).resi=resiOLS;
            clear('thetaOLS','resiOLS','sigmaOLS','stderOLS','estimadores');


%==========================================================================
%           Method 2: FEStd
%==========================================================================
            thetaFEStd=inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu*y;
            resiFEStd=Qmu*y-Qmu*Rwave*thetaFEStd;
            sigmaFEStd=(resiFEStd'*Qmu*resiFEStd)/(N*(T-1)-Kwave);
            stderFEStd=diag(sigmaFEStd*inv(Rwave'*Qmu*Rwave)).^0.5; %new std. errors
            estimadores=[thetaFEStd,stderFEStd];
            rdoeq1(2).coef=[estimadores; NaN(4,2)]; %Z1, Z2, VarPhi1, Phi1: empty
            rdoeq1(2).resi=resiFEStd;
            clear('resiFEStd','sigmaFEStd','stderFEStd','estimadores');


%==========================================================================
%           Method 3: FEBal
%==========================================================================
            thetaFEBal=inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
            resiFEBal=Qmu*y-Qmu*Rwave*thetaFEBal;
            sigmaFEBal=(resiFEBal'*Qmu*resiFEBal)/(N*(T-1)-Kwave);
            stderFEBal=diag(sigmaFEBal*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std.
errors
            estimadores=[thetaFEBal,stderFEBal];
            rdoeq1(3).coef=[estimadores; NaN(4,2)]; %Z1, Z2, VarPhi1, Phi1: empty
            rdoeq1(3).resi=resiFEBal;
            clear('resiFEBal','stderFEBal','estimadores');


%==========================================================================
```

```
%           Method 4: BEBal
%===========================================================================
            thetaBEBal=inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
            resiBEBal=Pmu*y-Pmu*R*thetaBEBal;
            sigmaBEBal=(resiBEBal'*Pmu*resiBEBal)/(N-K);
            stderBEBal=diag(sigmaBEBal*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            estimadores=[thetaBEBal,stderBEBal];
            rdoeq1(4).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq1(4).resi=resiBEBal;
            clear('thetaBEBal','resiBEBal','stderBEBal','estimadores');


%===========================================================================
%           Method 5: ECCorn
%===========================================================================
            WCorn=((1/sigmaFEBal)^0.5)*Qmu+((1/sigmaBEBal)^0.5)*Pmu; %sigma ^ (-1/2)
            thetaECCorn=inv(R'*WCorn*(PA+PB)*WCorn*R)*R'*WCorn*(PA+PB)*WCorn*y;
            resiECCorn=WCorn*y-WCorn*R*thetaECCorn;
            sigmaECCorn=(resiECCorn'*resiECCorn)/(N*T-K); %"within" component of
variance
            stderECCorn=diag(sigmaECCorn*inv(R'*WCorn*(PA+PB)*WCorn*R)).^0.5; %new std.
errors
            estimadores=[thetaECCorn,stderECCorn];
            rdoeq1(5).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq1(5).resi=resiECCorn;
            clear('WCorn','thetaECCorn','resiECCorn','sigmaECCorn',...
                'stderECCorn','sigmaBEBal','estimadores');


%===========================================================================
%           Method 6: HTM
%===========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=(ebarHT'*Pmu*ebarHT)/(N-K);
            WHT=((1/sigmaFEBal)^0.5)*Qmu+((1/sigma2HT)^0.5)*Pmu;
            thetaHT=inv(R'*WHT*(PA+PBHT)*WHT*R)*R'*WHT*(PA+PBHT)*WHT*y; %coefficients
            resiHT=WHT*y-WHT*R*thetaHT; %residuals
            sigmaHT=(resiHT'*resiHT)/(N*T-K); %"within" component of variance
            stderHT=diag(sigmaHT*inv(R'*WHT*(PA+PBHT)*WHT*R)).^0.5; %new std. errors
            estimadores=[thetaHT,stderHT];
            rdoeq1(6).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq1(6).resi=resiHT;
            clear('ebarHT','sigmaFEBal','sigma2HT','WHT','thetaHT',...
                'resiHT','sigmaHT','sigmaFEBal','stderHT','estimadores');


%===========================================================================
%             Method 7,8: G3SPD_Std
%===========================================================================
        % 1st stage:
            % idem FEStd.
```

```
        % 2nd Step:
            uhatStd=Pmu*y-Pmu*Rwave*thetaFEStd; %vector of mean residuals (dependent
variable)
            thetaStd=inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*uhatStd;
            resiStd2nd=Pmu*uhatStd-Pmu*Rbar*thetaStd;
            sigmaStd2nd=(resiStd2nd'*Pmu*resiStd2nd)/(N-Kbar);
            stderStd=diag(sigmaStd2nd*inv(Rbar'*Pmu*Rbar)).^0.5;
            estimadores=[thetaStd,stderStd];
            rdoeq1(7).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq1(7).resi=resiStd2nd;
            clear('uhatStd','thetaFEStd','stderStd','estimadores');

        % 3rd Step:
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),resiStd2nd]); %full dataset,
including all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            %Coefficients: stored in column 1
            thetaStd=inv(Rfull'*Rfull)*Rfull'*y;
            resiStd3rd=y-Rfull*thetaStd;

            % GMM covar.: stored in column 2
            sigmaStd3rd=(resiStd3rd'*resiStd3rd)/(N*T-K);
            OmegaGMM=sigmaStd3rd*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=diag(resiStd3rd.^2);
            OmegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=diag((resiStd3rd.^2)./(1-diag(PR)));
            OmegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiStd3rd./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            OmegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            OmegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');
```

```
            % BK covar.: stored in column 7
            E=reshape(resiStd3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix↙
for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaStd2nd/(sigmaStd3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigmaStd3rd^2))*(Rfull'*resiStd3rd)*...
                (resiStd2nd'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaStd3rd','sigmaStd2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiStd2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq1(8).coef=[thetaStd,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,↙
stderBK,stderMT];
            rdoeq1(8).resi=resiStd3rd;
            clear('resiStd3rd','thetaStd','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','estimadores','Rfull');


%==========================================================================
%           Method 9,10: G3SPD_Corn
%==========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            uhatCorn=Pmu*y-Pmu*Rwave*thetaFEBal; %vector of mean residuals (dependent↙
variable)
            thetaCorn=inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*Pmu*uhatCorn;
            resiCorn2nd=Pmu*uhatCorn-Pmu*Rbar*thetaCorn; %residuals 2nd stage
            sigmaCorn2nd=(resiCorn2nd'*Pmu*resiCorn2nd)/(N-Kbar);
            stderCorn=diag(sigmaCorn2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            estimadores=[thetaCorn,stderCorn];
            rdoeq1(9).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq1(9).resi=resiCorn2nd;
            clear('uhatCorn','thetaCorn','stderCorn','estimadores');

        % 3rd Step:
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),resiCorn2nd]); %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),resiCorn2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
```

```
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaCorn=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiCorn3rd=y-Rfull*thetaCorn;

            % GMM covar.: stored in column 2
            sigmaCorn3rd=(resiCorn3rd'*resiCorn3rd)/(N*T-K);
            OmegaGMM=sigmaCorn3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiCorn3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiCorn3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiCorn3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiCorn3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaCorn2nd/(sigmaCorn3rd^2))* Rfull'*PC*PB*Pmu*Rbar*...
```

```
                inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigmaCorn3rd^2))*(Rfull'*PC*resiCorn3rd)*...
                (resiCorn2nd'*PB*Pmu*Rbar*inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaCorn3rd','sigmaCorn2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiCorn2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq1(10).coef=[thetaCorn,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(10).resi=resiCorn3rd;
            clear('resiCorn3rd','thetaCorn','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');


%==========================================================================
%           Method 11,12: G3SPD_HTM (Old version using the whole set of
%                                   instruments in the 3rd Step estimation)
%==========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            uhatHTM=Pmu*y-Pmu*Rwave*thetaFEBal; %vector of mean residuals (dependent variable)
            thetaHTM=inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*Pmu*uhatHTM;
            resiHTM2nd=Pmu*uhatHTM-Pmu*Rbar*thetaHTM; %residuals 2nd stage
            sigmaHTM2nd=(resiHTM2nd'*Pmu*resiHTM2nd)/(N-Kbar);
            stderHTM=diag(sigmaHTM2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
            estimadores=[thetaHTM,stderHTM];
            rdoeq1(11).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq1(11).resi=resiHTM2nd;
            clear('uhatHTM','thetaHTM','stderHTM','estimadores','thetaFEBal');

        % 3rd Step:
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),resiHTM2nd]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTM=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTM3rd=y-Rfull*thetaHTM;

            % GMM covar.: stored in column 2
            sigmaHTM3rd=(resiHTM3rd'*resiHTM3rd)/(N*T-K);
```

```
            OmegaGMM=sigmaHTM3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTM3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTM3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTM3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTM3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTM3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTM3rd^2))*(Rfull'*PC*resiHTM3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)↙
*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTM3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT');
```

```
            % store coefficients and standard errors (8 columns)
            rdoeq1(12).coef=[thetaHTM,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(12).resi=resiHTM3rd;
            clear('resiHTM3rd','thetaHTM','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');



%==========================================================================
%           Method 13: G3SPD_HTMNew (New version using the HT set of
%                                   instruments in the 3rd Step estimation)
%==========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            % idem G3SPD_HTM (Old version)

        % 3rd Step
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),resiHTM2nd]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTMNEW=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTMNEW3rd=y-Rfull*thetaHTMNEW;

            % GMM covar.: stored in column 2
            sigmaHTMNEW3rd=(resiHTMNEW3rd'*resiHTMNEW3rd)/(N*T-K);
            OmegaGMM=sigmaHTMNEW3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTMNEW3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTMNEW3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');
```

```
            % WH-LE covar.: stored in column 5
            epsilonast=resiHTMNEW3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTMNEW3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTMNEW3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTMNEW3rd^2))*(Rfull'*PC*resiHTMNEW3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)
*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTMNEW3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiHTM2nd','sigmaHTM2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq1(13).coef=[thetaHTMNEW,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,
stderBK,stderMT];
            rdoeq1(13).resi=resiHTMNEW3rd;
            clear('resiHTMNEW3rd','thetaHTMNEW','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');

%=====================================================================
%          Save results
%=====================================================================
            if exist(['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')],'dir' )~=7
                mkdir(['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
            end
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
```

```
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');

        end
        toc
    end
end
```

```
%===================================================================
% Function ESTIMAEQ1_V7
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=======================================================================

function estimaeq1_v7
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        BHT=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);
            Y=single(datos(1,obs).Y(:,2));
            y=single(datos(1,obs).Y(:,1));
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);
            Rwave=[Y,X1,X2,X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

            %open data for the observation
            filename=['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');

% ==========================================================================
%           Method 3: FEBal
% ==========================================================================
            resiFEBal=rdoeq1(3).resi;
            sigmaFEBal=(resiFEBal'*Qmu*resiFEBal)/(N*(T-1)-Kwave);
            stderFEBal=diag(sigmaFEBal*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq1(3).coef(:,2)=[  NaN(1,1);         %y1: empty
                                   stderFEBal(1,:);    %y2
                                   stderFEBal(2,:);    %X1
                                   stderFEBal(3,:);    %X2
                                   stderFEBal(4,:);    %X3
                                   NaN(9,1)];        %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            clear('resiFEBal','sigmaFEBal','stderFEBal');

% ==========================================================================
%           Method 5: BEBal
% ==========================================================================
            resiBEBal=rdoeq1(5).resi;
            sigmaBEBal=(resiBEBal'*Pmu*resiBEBal)/(N-K);
            stderBEBal=diag(sigmaBEBal*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            rdoeq1(5).coef(:,2)=[  NaN(1,1);         %y1: empty
                                   stderBEBal(1,:);    %y2
                                   stderBEBal(2,:);    %X1
                                   stderBEBal(3,:);    %X2
                                   stderBEBal(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stderBEBal(5,:);    %Z1
                                   stderBEBal(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stderBEBal(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty
            clear('resiBEBal','sigmaBEBal','stderBEBal');

% ==========================================================================
```

```
%           Method 9: ECCorn
% ===========================================================================
            %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
            ewave=rdoeq1(3).resi; %taken from FEBal
            sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
            %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
            ebar=rdoeq1(5).resi; %taken from BEBal
            sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
            WCorn=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
            resiECCorn=rdoeq1(9).resi;
            sigmaECCorn=(resiECCorn'*resiECCorn)/(N*T-K); %"within" component of variance
            stderECCorn=diag(sigmaECCorn*inv(R'*WCorn*(PA+PB)*WCorn*R)).^0.5; %new std. errors
            rdoeq1(9).coef(:,2)=[  NaN(1,1);         %y1: empty
                                   stderECCorn(1,:);    %y2
                                   stderECCorn(2,:);    %X1
                                   stderECCorn(3,:);    %X2
                                   stderECCorn(4,:);    %X3
                                   NaN(2,1);         %X4,X5: empty
                                   stderECCorn(5,:);    %Z1
                                   stderECCorn(6,:);    %Z2
                                   NaN(1,1);         %Z3: empty
                                   stderECCorn(7,:);    %phi1
                                   NaN(3,1)];        %phi2,mu1,mu2: empty
            clear('ewave','ebar','sigma2','WCorn','resiECCorn',...
                'sigmaECCorn','stderECCorn');

% ===========================================================================
%           Method 10: HTM
% ===========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=((ebarHT'*Pmu*ebarHT)/(N-K-1))^0.5;
            WHT=(1/sigma1)*Qmu+(1/sigma2HT)*Pmu;
            resiHT=rdoeq1(10).resi; %residuals
            sigmaHT=(resiHT'*resiHT)/(N*T-K); %"within" component of variance
            stderHT=diag(sigmaHT*inv(R'*WHT*(PA+PBHT)*WHT*R)).^0.5; %new std. errors
            rdoeq1(10).coef(:,2)=[  NaN(1,1);         %y1: empty
                                    stderHT(1,:);    %y2
                                    stderHT(2,:);    %X1
                                    stderHT(3,:);    %X2
                                    stderHT(4,:);    %X3
                                    NaN(2,1);         %X4,X5: empty
                                    stderHT(5,:);    %Z1
                                    stderHT(6,:);    %Z2
                                    NaN(1,1);         %Z3: empty
                                    stderHT(7,:);    %phi1
                                    NaN(3,1)];        %phi2,mu1,mu2: empty
            clear('ebarHT','sigma1','sigma2HT','WHT','resiHT','sigmaHT','stderHT');

% ==========================================================================
%           Method 15: G3SPD_Std 3rd Step
```

```
%======================================================================
            resiStd2nd=rdoeq1(14).resi;
            sigmaStd2nd=(resiStd2nd'*resiStd2nd)/(N-Kbar);
            resiStd3rd=rdoeq1(15).resi;
            Rfull=single([Y,X1,X2,X3,Z1,Z2,resiStd2nd,ones(N*T,1)]); %full dataset, using all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaStd3rd=(resiStd3rd'*resiStd3rd)/(N*T-K);
            OmegaGMM=sigmaStd3rd*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=diag(resiStd3rd.^2);
            OmegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=diag((resiStd3rd.^2)./(1-diag(PR)));
            OmegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiStd3rd./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            OmegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            OmegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiStd3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaStd2nd/(sigmaStd3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigmaStd3rd^2))*(Rfull'*resiStd3rd)*...
```

```
                (resiStd2nd'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaStd3rd','sigmaStd2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiStd2nd');

            % store coefficients and standard errors (8 columns)
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(15).coef=rdoeq1(15).coef(:,1);
            rdoeq1(15).coef(:,2:8)=[NaN(1,7);      %y1: empty
                                    stder(1,:);    %y2
                                    stder(2,:);    %X1
                                    stder(3,:);    %X2
                                    stder(4,:);    %X3
                                    NaN(2,7);      %X4,X5: empty
                                    stder(5,:);    %Z1
                                    stder(6,:);    %Z2
                                    NaN(1,7);      %Z3: empty
                                    stder(8,:);    %phi1
                                    NaN(1,7);      %phi2: empty
                                    stder(7,:);    %mu1
                                    NaN(1,7)];     %mu2: empty
            clear('resiStd3rd','stderGMM','stderHC1','stderHC2','PR',...
                'stderLE','stderHC3','stderBK','stderMT','Rfull','stder');

%==========================================================================
%           Method 16,17,18: G3SPD_Corn
%==========================================================================
        % 1st stage
            resiCorn1st=rdoeq1(16).resi;
            sigmaCorn1st=(resiCorn1st'*Qmu*resiCorn1st)/(N*(T-1)-Kwave);
            stderCorn1st=diag(sigmaCorn1st*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5;
            rdoeq1(16).coef(:,2)=[  NaN(1,1);             %y1: empty
                                    stderCorn1st(1,:);    %y2
                                    stderCorn1st(2,:);    %X1
                                    stderCorn1st(3,:);    %X2
                                    stderCorn1st(4,:);    %X3
                                    NaN(9,1)];            %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            clear('resiCorn1st','sigmaCorn1st','stderCorn1st');

        % 2nd stage
            resiCorn2nd=rdoeq1(17).resi;
            sigmaCorn2nd=(resiCorn2nd'*Pmu*resiCorn2nd)/(N-Kbar);
            stderCorn2nd=diag(sigmaCorn2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            rdoeq1(17).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stderCorn2nd(1,:);    %Z1
                                    stderCorn2nd(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stderCorn2nd(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty
            clear('stderCorn2nd');
```

```
        % 3rd stage
            resiCorn3rd=rdoeq1(18).resi;
            Rfull=[Y,X1,X2,X3,Z1,Z2,resiCorn2nd,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resiCorn2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaCorn3rd=(resiCorn3rd'*resiCorn3rd)/(N*T-K);
            OmegaGMM=sigmaCorn3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiCorn3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiCorn3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiCorn3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiCorn3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
```

```
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaCorn2nd/(sigmaCorn3rd^2))* Rfull'*PC*PB*Pmu*Rbar*...
                inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigmaCorn3rd^2))*(Rfull'*PC*resiCorn3rd)*...
                (resiCorn2nd'*PB*Pmu*Rbar*inv(Rbar'*Pmu*PB*Pmu*Rbar)↙
*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaCorn3rd','sigmaCorn2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiCorn2nd');

            % store standard errors
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(18).coef=rdoeq1(18).coef(:,1);
            rdoeq1(18).coef(:,2:8)=[NaN(1,7);      %y1: empty
                                    stder(1,:);    %y2
                                    stder(2,:);    %X1
                                    stder(3,:);    %X2
                                    stder(4,:);    %X3
                                    NaN(2,7);      %X4,X5: empty
                                    stder(5,:);    %Z1
                                    stder(6,:);    %Z2
                                    NaN(1,7);      %Z3: empty
                                    stder(8,:);    %phi1
                                    NaN(1,7);      %phi2: empty
                                    stder(7,:);    %mu1
                                    NaN(1,7)];     %mu2: empty
            clear('resiCorn3rd','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull','stder');

% %==========================================================================
%              %Method 19,20: G3SPD_HTM
% %==========================================================================
        % 2nd stage
            resiHTM2nd=rdoeq1(19).resi;
            sigmaHTM2nd=(resiHTM2nd'*Pmu*resiHTM2nd)/(N-Kbar);
            stderHTM2nd=diag(sigmaHTM2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
            rdoeq1(19).coef(:,2)=[  NaN(7,1);              %y1,y2,X1,X2,X3,X4,X5: empty
                                    stderHTM2nd(1,:);    %Z1
                                    stderHTM2nd(2,:);    %Z2
                                    NaN(1,1);              %Z3: empty
                                    stderHTM2nd(3,:);    %phi1
                                    NaN(3,1)];             %phi2,mu1,mu2: empty
            clear('stderHTM2nd');

        % 3rd Step:
            resiHTM3rd=rdoeq1(20).resi;
            Rfull=[Y,X1,X2,X3,Z1,Z2,resiHTM2nd,ones(N*T,1)]; %full dataset, including↙
```

```
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resiHTM2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaHTM3rd=(resiHTM3rd'*resiHTM3rd)/(N*T-K);
            OmegaGMM=sigmaHTM3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTM3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTM3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTM3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTM3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
```

```
            Psi2=(sigmaHTM2nd/(sigmaHTM3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTM3rd^2))*(Rfull'*PC*resiHTM3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)↙
*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTM3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT');

            % store standard errors
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq1(20).coef=rdoeq1(20).coef(:,1);
            rdoeq1(20).coef(:,2:8)=[NaN(1,7);      %y1: empty
                                    stder(1,:);    %y2
                                    stder(2,:);    %X1
                                    stder(3,:);    %X2
                                    stder(4,:);    %X3
                                    NaN(2,7);      %X4,X5: empty
                                    stder(5,:);    %Z1
                                    stder(6,:);    %Z2
                                    NaN(1,7);      %Z3: empty
                                    stder(8,:);    %phi1
                                    NaN(1,7);      %phi2: empty
                                    stder(7,:);    %mu1
                                    NaN(1,7)];     %mu2: empty
            clear('resiHTM3rd','stder','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','Rfull');

%==========================================================================
%           Method 25: G3SPD_HTMNew (New version using the HT set of
%                                   instruments in the 3rd Step estimation)
%==========================================================================
        % 1st Step:
            % idem G3SPD_HTM

        % 2nd Step:
            % idem G3SPD_HTM

        % 3rd Step
            Rfull=single([Y,X1,X2,X3,Z1,Z2,resiHTM2nd,ones(N*T,1)]); %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTMNEW=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTMNEW3rd=y-Rfull*thetaHTMNEW;
```

```
            % GMM covar.: stored in column 2
            sigmaHTMNEW3rd=(resiHTMNEW3rd'*resiHTMNEW3rd)/(N*T-K);
            OmegaGMM=sigmaHTMNEW3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTMNEW3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTMNEW3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTMNEW3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTMNEW3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTMNEW3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTMNEW3rd^2))*(Rfull'*PC*resiHTMNEW3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
```

```
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTMNEW3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiHTM2nd','sigmaHTM2nd');

            % store coefficients and standard errors (8 columns)
            stder=[thetaHTMNEW,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,↙
stderMT];
            rdoeq1(25).coef=[NaN(1,8);      %y1: empty
                            stder(1,:);    %y2
                            stder(2,:);    %X1
                            stder(3,:);    %X2
                            stder(4,:);    %X3
                            NaN(2,8);      %X4,X5: empty
                            stder(5,:);    %Z1
                            stder(6,:);    %Z2
                            NaN(1,8);      %Z3: empty
                            stder(8,:);    %phi1
                            NaN(1,8);      %phi2: empty
                            stder(7,:);    %mu1
                            NaN(1,8)];     %mu2: empty
            rdoeq1(25).resi=resiHTMNEW3rd;
            clear('resiHTMNEW3rd','thetaHTMNEW','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');

            % Save results
            if exist(['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')],'dir')~=7
                mkdir(['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')]);
            end
            filename=['C:\Diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');

        end
        toc
    end
end
```

```
%==========================================================================
% Function ESTIMAEQ1_ALTER_V1
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%==========================================================================

function estimaeq1_alter_v1
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        BHT=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';
```

```
        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);
            Rwave=[Y,X1,X2,X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');

%==========================================================================
%           1st Step: FEBal
%==========================================================================
            redo(1).coef=rdoeq1(3).coef;
            redo(1).resi=rdoeq1(3).resi;
            thetaFE=redo(1).coef(2:5,1);
%==========================================================================
%           2nd Step: BEBal
%==========================================================================
            redo(2).coef=rdoeq1(5).coef;
            redo(2).resi=rdoeq1(5).resi;
            thetaBE=[redo(2).coef(8:9,1);
                redo(2).coef(11,1)];

%==========================================================================
%           3rd Step: MuHat
%==========================================================================
            uhat=Pmu*y-Pmu*Rwave*thetaFE;
            MuHat=uhat-Rbar*thetaBE;
            redo(3).resi=MuHat;

%==========================================================================
%           4th Step: Pooled OLS
%==========================================================================
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaCorn=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi=y-Rfull*thetaCorn;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
```

```
            OmegaGMM=sigmaGMM*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            redo(4).coef=[thetaCorn,stderGMM,stderBK];
            redo(4).resi=resi;
            clear('resi','thetaCorn','stderGMM','stderBK','Rfull',...
                'C','RCinvCC');

%======================================================================
%           Save results
%======================================================================
            if exist(['.\eq1_alter\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')],'dir' )~=7
                mkdir(['.\eq1_alter\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')]);
            end
            filename=['.\eq1_alter\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'redo');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1','redo');

        end
        toc
    end
end
```

```
%=======================================================================
% Function ESTIMAEQ1_ALTER_V2
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%===========================================================================

function estimaeq1_alter_v2
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio


%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)


for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['c:\diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['c:\diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        BHT=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
            R=[Y,X1,X2,X3,Z1,Z2,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);

            filename=['c:\diego\SimNewFE\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');


%============================================================================
%           Move Methods (6 to 25) to (7 to 26)
%============================================================================
            for met=25:(-1):6
                if size(rdoeq1(met).coef,2)>0
                    rdoeq1(met+1).coef=[rdoeq1(met).coef(2:5,:);    %y2,X1,X2,X3
                                        rdoeq1(met).coef(8:9,:);    %Z1,Z2
                                        rdoeq1(met).coef(11,:);     %Varphi1
                                        rdoeq1(met).coef(13,:)];    %Phi1
                    rdoeq1(met+1).resi=rdoeq1(met).resi;
                    rdoeq1(met+1).stats=rdoeq1(met).stats;
                else
                    rdoeq1(met+1).coef=[];
                    rdoeq1(met+1).resi=[];
                    rdoeq1(met+1).stats=[];
                end
            end


%============================================================================
%           Method 6: BEHTM
%============================================================================
            thetaBEHTM=inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            resiBEHTM=Pmu*y-Pmu*R*thetaBEHTM;
            sigmaBEHTM=(resiBEHTM'*Pmu*resiBEHTM)/(N-K);
            stderBEHTM=diag(sigmaBEHTM*inv(R'*Pmu*PBHT*Pmu*R)).^0.5; %new std. errors
            estimadores=[thetaBEHTM,stderBEHTM];
            rdoeq1(6).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq1(6).resi=resiBEHTM;
            clear('thetaBEHTM','resiBEHTM','stderBEHTM','estimadores');

%============================================================================
%           Arrange Methods (1 to 5)
%============================================================================
            for met=1:1:5
                if size(rdoeq1(met).coef,2)>0
                    rdoeq1(met).coef=[  rdoeq1(met).coef(2:5,:);    %y2,X1,X2,X3
                                        rdoeq1(met).coef(8:9,:);    %Z1,Z2
```

```
                                               rdoeq1(met).coef(11,:);     %Varphi1
                                               rdoeq1(met).coef(13,:)];    %Phi1
                end
            end


%==========================================================================
%          Save results
%==========================================================================
            if exist(['c:\diego\SimLastFE\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_'
num2str(rho,    '%3.0f')],'dir')~=7
                mkdir(['c:\diego\SimLastFE\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_'
num2str(rho,    '%3.0f')]);
            end
            filename=['c:\diego\SimLastFE\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_'
num2str(rho,    '%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1');

        end
        toc
    end
end
```

```
%==========================================================================
% Function ESTIMAEQ1_ALTER_V3
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%===========================================================================

function estimaeq1_alter_v3
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio


%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)


for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %---------------------
        Rbar=single([Z1,Z2,ones(N*T,1)]);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
```

```
            Rwave=[Y,X1,X2,X3]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');
            filename=['.\eq1_alter\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'redo');


%=========================================================================
%           Method 27,28: G3SPD_Std_alter 3rd Step
%=========================================================================
            %generates pseudo-effects (stored in 27)
            thetaFEStd=rdoeq1(2).coef(1:4,1);
            thetaBEStd=rdoeq1(4).coef(5:7,1);
            MuHat=Pmu*y-Pmu*Rwave*thetaFEStd-Rbar*thetaBEStd;
            rdoeq1(27).resi=MuHat;

            %estimation (stored in 28)
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaStd=inv(Rfull'*Rfull)*Rfull'*y;
            resi=y-Rfull*thetaStd;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=kron((E'*E)./(T-K),eye(T));%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            rdoeq1(28).coef=[thetaStd,stderGMM,stderBK];
            rdoeq1(28).resi=resi;
            clear('resi','thetaFEStd','thetaBEStd','MuHat','Rfull',...
                'thetaStd','stderGMM','stderBK');
```

```
%==========================================================================
%           Method 29,30: G3SPD_Corn_alter 3rd Step
%==========================================================================
            %generates pseudo-effects (stored in 29)
            rdoeq1(29).resi=redo(3).resi;

            % store coefficients and standard errors (3 columns) (stored in 30)
            rdoeq1(30).coef=[redo(4).coef(:,1:2),NaN(8,4),redo(4).coef(:,3),NaN(8,1)];
            rdoeq1(30).resi=redo(4).resi;

%==========================================================================
%           Method 31,32: G3SPD_HTM_alter 3rd Step
%==========================================================================
            %generates pseudo-effects (stored in 31)
            thetaFEBal=rdoeq1(3).coef(1:4,1);
            thetaBEHTM=rdoeq1(6).coef(5:7,1);
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEHTM;
            rdoeq1(31).resi=MuHat;

            %estimation (stored in 32)
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTM=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi=y-Rfull*thetaHTM;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            rdoeq1(32).coef=[thetaHTM,stderGMM,NaN(8,4),stderBK,NaN(8,1)];
            rdoeq1(32).resi=resi;
            clear('resi','thetaHTM','stderGMM','stderBK','C','RCinvCC',...
                'thetaFEBal','thetaBEHTM');
```

```
%===================================================================
%          Save results
%===================================================================
            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1','redo');

        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMAEQ1_ALTER_V4
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=============================================================================

function estimaeq1_alter_v4
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);
        Rbar=single([Z1,Z2,ones(N*T,1)]);
        Kbar=size(Rbar,2);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,1)); %dependent var.
            Y=single(datos(1,obs).Y(:,2)); %endogenous var.
            Rwave=single([Y,X1,X2,X3]);
            Kwave=size(Rwave,2);

            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
```

```
f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            load(filename,'rdoeq1');

%==========================================================================
%           Method 30: G3SPD_Corn_alter 3rd Step
%==========================================================================
            MuHat=rdoeq1(29).resi;
            Rfull=single([Y,X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat]); %full dataset, including
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            PC=C*inv(C'*C)*C';
            K=size(Rfull,2);

            % obtain residuals and V
            resi1=squeeze(rdoeq1(30).resi);
            resi2=squeeze(rdoeq1(3).resi);
            resi3=(Pmu*Rwave*squeeze(rdoeq1(3).coef(1:4,1))+MuHat)/T;
            resi=[resi1,resi2,resi3]; %residuals (OLS 3rd step, Within, ~between)
            dof1=N*T-K;
            dof2=N*(T-1)-Kwave;
            dof3=(N*T-(K-1));
            dof=[dof1,    dof1,    dof1;
                 dof1,    dof2,    dof2;
                 dof1,    dof2,    dof3];

            V=(resi'*resi)./dof;
            clear('resi1','resi2','resi3','resi','dof');

            %obtain S
            QRwavebar=eye(N*T)-Pmu*Rwave*inv(Rwave'*Pmu*Rwave)*Rwave'*Pmu;
            S1=eye(N*T);
            S2=Pmu*Rwave*inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu;
            S3=Rbar*inv(Rbar'*QRwavebar*Rbar)*Rbar'*QRwavebar;
            S=[S1 S2 S3];
            clear('S1','S2','S3','QRwavebar');

            %obtain Sigma
            sigmaAdjOLS=S*kron(V,eye(N*T))*S';
            OmegaAdjOLS=inv(Rfull'*PC*Rfull)*(Rfull'*PC*sigmaAdjOLS*PC*Rfull)*inv
(Rfull'*PC*Rfull);
            clear('V','S','sigmaAdjOLS');
            sterdnew=diag(OmegaAdjOLS).^0.5; %new std. errors
            rdoeq1(30).coef(:,9)=sterdnew; %store new std. errors in column 9

%======================================================================
%          Save results
%======================================================================
            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
```

```
(obs '%4.0f')];
            save(filename,'rdoeq1');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq1','redo');

        end
        toc
    end
end
```

```
%===================================================================
% Function ESTIMAEQ2_V1
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%=====================================================================

function estimaeq2_v1
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    D=single(kron(eye(N),ones(1,T)*(1/T))); %reduce matrix from NT*1 to N*1

   for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3); mu1=single(mu1); mu2=single(mu2);
        %create new folder for containing 1000 obs.
        if exist(['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')],'dir')↙
~=7
            mkdir(['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end

        %predetermined matrixes
        %----------------------
        W=eye(N*T);
        P=eye(N*T);
        Wwave=Qmu;
        Rbar=[Z1,Z3,ones(N*T,1)];
        Kbar=size(Rbar,2);
        Wbar=Pmu;
        A=Qmu*([X1,X2,X3,X4,X5]); %within instruments
```

```
        PA=A*inv(A'*A)*A';
        B=Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]); %between instruments
        PB=B*inv(B'*B)*B';
        PAB=PA+PB; %projection matrix of instruments
        BHT=Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]; %between instruments
        if rank(BHT)<size(BHT,2);BHT=BHT(:,1:rank(BHT));end
        PBHT=BHT*inv(BHT'*BHT)*BHT';
        PABHT=PA+PBHT; %projection matrix of instruments
        BAM=Pmu*[SA([X1,X4],N,T),SA([X2,X3],N,T),Z1,Z2,ones(N*T,1)]; %between instruments
        if rank(BAM)<size(BAM,2);BAM=BAM(:,1:rank(BAM));end
        PBAM=BAM*inv(BAM'*BAM)*BAM';
        PABAM=PA+PBAM; %projection matrix of instruments
        BBMS=Pmu*[SA([X1,X4],N,T),SA([X2,X3],N,T),Z1,Z2,SA(Qmu*X5,N,T),ones(N*T,1)]; %between instruments
        if rank(BBMS)<size(BBMS,2);BBMS=BBMS(:,1:rank(BBMS));end
        PBBMS=BBMS*inv(BBMS'*BBMS)*BBMS';
        PABBMS=PA+PBBMS; %projection matrix of instruments

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                    Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2));
            Y=single(datos(1,obs).Y(:,1));
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);
            Rwave=[Y X1 X4 X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);
            clear rdo

%===========================================================================
            %Method 1: OLS
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,eye(N*T),eye(N*T),y,N,T);
            rdoeq2(1).coef=[rdo.coef(1,:);  %y1
                                   NaN(1,2);          %y2: empty
                                   rdo.coef(2,:);     %X1
                                   NaN(2,2);          %X2,X3: empty
                                   rdo.coef(3,:);     %X4
                                   rdo.coef(4,:);     %X5
                                   rdo.coef(5,:);     %Z1
                                   NaN(1,2);          %Z2: empty
                                   rdo.coef(6,:);     %Z3
                                   NaN(1,2);          %phi1
                                   rdo.coef(7,:);     %phi2
                                   NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(1).resi=rdo.resi;
            rdoeq2(1).stats=rdo.stats;

%===========================================================================
            %Method 2: FEStd
%===========================================================================
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(Rwave,Qmu,eye(N*T),y,N,T);
            betaFEStd=rdo(1).coef(:,1); %to be used in ECAm
            rdoeq2(2).coef=[rdo.coef(1,:);  %y1
                                   NaN(1,2);          %y2: empty
                                   rdo.coef(2,:);     %X1
                                   NaN(2,2);          %X2,X3:empty
                                   rdo.coef(3,:);     %X4
                                   rdo.coef(4,:);     %X5
                                   NaN(7,2)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq2(2).resi=rdo.resi;
            rdoeq2(2).stats=rdo.stats;

%==========================================================================
            %Method 3: FEBal
%==========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(Rwave,Qmu,PA,y,N,T);
            rdoeq2(3).coef=[rdo.coef(1,:);  %y1
                                   NaN(1,2);          %y2: empty
                                   rdo.coef(2,:);     %X1
                                   NaN(2,2);          %X2,X3:empty
                                   rdo.coef(3,:);     %X4
                                   rdo.coef(4,:);     %X5
                                   NaN(7,2)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq2(3).resi=rdo.resi;
            rdoeq2(3).stats=rdo.stats;

%==========================================================================
            %Method 4: BEStd
%==========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,Pmu,eye(N*T),y,N,T);
            rdoeq2(4).coef=[rdo.coef(1,:);  %y1
                                   NaN(1,2);          %y2: empty
                                   rdo.coef(2,:);     %X1
                                   NaN(2,2);          %X2,X3: empty
                                   rdo.coef(3,:);     %X4
                                   rdo.coef(4,:);     %X5
                                   rdo.coef(5,:);     %Z1
                                   NaN(1,2);          %Z2: empty
                                   rdo.coef(6,:);     %Z3
                                   NaN(1,2);          %phi1
                                   rdo.coef(7,:);     %phi2
                                   NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(4).resi=rdo.resi;
            rdoeq2(4).stats=rdo.stats;

%==========================================================================
            %Method 5: BEBal
%==========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,Pmu,PB,y,N,T);
            rdoeq2(5).coef=[rdo.coef(1,:);  %y1
                                   NaN(1,2);          %y2: empty
                                   rdo.coef(2,:);     %X1
```

```
                             NaN(2,2);           %X2,X3: empty
                             rdo.coef(3,:);      %X4
                             rdo.coef(4,:);      %X5
                             rdo.coef(5,:);      %Z1
                             NaN(1,2);           %Z2: empty
                             rdo.coef(6,:);      %Z3
                             NaN(1,2);           %phi1
                             rdo.coef(7,:);      %phi2
                             NaN(2,2)];          %mu1,mu2: empty
            rdoeq2(5).resi=rdo.resi;
            rdoeq2(5).stats=rdo.stats;

%==========================================================================
            %Method 6: ECWH
%==========================================================================
            u=y-R*inv(R'*R)*R'*y; %OLS residuals
            sigma1=((u'*Qmu*u)/(N*(T-1)))^0.5; %"within" component of variance
            sigma2=((u'*Pmu*u)/N)^0.5; %"between" component of variance
            WWH=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % omega^(-1/2)
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WWH,eye(N*T),y,N,T);
            rdoeq2(6).coef=[rdo.coef(1,:);  %y1
                             NaN(1,2);           %y2: empty
                             rdo.coef(2,:);      %X1
                             NaN(2,2);           %X2,X3: empty
                             rdo.coef(3,:);      %X4
                             rdo.coef(4,:);      %X5
                             rdo.coef(5,:);      %Z1
                             NaN(1,2);           %Z2: empty
                             rdo.coef(6,:);      %Z3
                             NaN(1,2);           %phi1
                             rdo.coef(7,:);      %phi2
                             NaN(2,2)];          %mu1,mu2: empty
            rdoeq2(6).resi=rdo.resi;
            rdoeq2(6).stats=rdo.stats;

%==========================================================================
            %Method 7: ECAm
%==========================================================================
            alpha=mean(D*y)-mean(D*Rwave)*betaFEStd; %constant term
            u=y-alpha*ones(N*T,1)-Rwave*betaFEStd; %residuals of a LSDV regression
            sigma1=((u'*Qmu*u)/(N*(T-1)))^0.5; %"within" component of variance
            sigma2=((u'*Pmu*u)/N)^0.5; %"between" component of variance
            WAm=(1/sigma1)*Qmu+(1/sigma2)*Pmu; %omega ^ (-1/2)
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WAm,eye(N*T),y,N,T);
            rdoeq2(7).coef=[rdo.coef(1,:);  %y1
                             NaN(1,2);           %y2: empty
                             rdo.coef(2,:);      %X1
                             NaN(2,2);           %X2,X3: empty
                             rdo.coef(3,:);      %X4
                             rdo.coef(4,:);      %X5
                             rdo.coef(5,:);      %Z1
                             NaN(1,2);           %Z2: empty
```

```
                              rdo.coef(6,:);    %Z3
                              NaN(1,2);         %phi1
                              rdo.coef(7,:);    %phi2
                              NaN(2,2)];        %mu1,mu2: empty
        rdoeq2(7).resi=rdo.resi;
        rdoeq2(7).stats=rdo.stats;

%===========================================================================
        %Method 8: ECSA
%===========================================================================
        %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu*y;
        ewave=squeeze(rdoeq2(2).resi);
        sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
        %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*R)*R'*Pmu*y;
        ebar=squeeze(rdoeq2(4).resi);
        sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
        WSA=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
        [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WSA,eye(N*T),y,N,T);
        rdoeq2(8).coef=[rdo.coef(1,:);  %y1
                              NaN(1,2);         %y2: empty
                              rdo.coef(2,:);    %X1
                              NaN(2,2);         %X2,X3: empty
                              rdo.coef(3,:);    %X4
                              rdo.coef(4,:);    %X5
                              rdo.coef(5,:);    %Z1
                              NaN(1,2);         %Z2: empty
                              rdo.coef(6,:);    %Z3
                              NaN(1,2);         %phi1
                              rdo.coef(7,:);    %phi2
                              NaN(2,2)];        %mu1,mu2: empty
        rdoeq2(8).resi=rdo.resi;
        rdoeq2(8).stats=rdo.stats;

%===========================================================================
        %Method 9: ECCorn
%===========================================================================
        %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
        ewave=rdoeq2(3).resi; %taken from FEBal
        sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
        %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
        ebar=rdoeq2(5).resi; %taken from BEBal
        sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
        WCorn=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
        [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WCorn,PAB,y,N,T);
        rdoeq2(9).coef=[rdo.coef(1,:);  %y1
                              NaN(1,2);         %y2: empty
                              rdo.coef(2,:);    %X1
                              NaN(2,2);         %X2,X3: empty
                              rdo.coef(3,:);    %X4
                              rdo.coef(4,:);    %X5
                              rdo.coef(5,:);    %Z1
                              NaN(1,2);         %Z2: empty
```

```
                       rdo.coef(6,:);     %Z3
                       NaN(1,2);          %phi1
                       rdo.coef(7,:);     %phi2
                       NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(9).resi=rdo.resi;
            rdoeq2(9).stats=rdo.stats;

%===========================================================================
            %Method 10: HTHT
%===========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=((ebarHT'*Pmu*ebarHT)/(N-K-1))^0.5;
            WHT=(1/sigma1)*Qmu+(1/sigma2HT)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WHT,PABHT,y,N,T);
            rdoeq2(10).coef=[rdo.coef(1,:); %y1
                       NaN(1,2);          %y2: empty
                       rdo.coef(2,:);     %X1
                       NaN(2,2);          %X2,X3: empty
                       rdo.coef(3,:);     %X4
                       rdo.coef(4,:);     %X5
                       rdo.coef(5,:);     %Z1
                       NaN(1,2);          %Z2: empty
                       rdo.coef(6,:);     %Z3
                       NaN(1,2);          %phi1
                       rdo.coef(7,:);     %phi2
                       NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(10).resi=rdo.resi;
            rdoeq2(10).stats=rdo.stats;

%===========================================================================
            %Method 11: HTAM
%===========================================================================
            ebarAM=Pmu*y-Pmu*R*inv(R'*Pmu*PBAM*Pmu*R)*R'*Pmu*PBAM*Pmu*y;
            sigma2AM=((ebarAM'*Pmu*ebarAM)/(N-K-1))^0.5;
            WAM=(1/sigma1)*Qmu+(1/sigma2AM)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WAM,PABAM,y,N,T);
            rdoeq2(11).coef=[rdo.coef(1,:); %y1
                       NaN(1,2);          %y2: empty
                       rdo.coef(2,:);     %X1
                       NaN(2,2);          %X2,X3: empty
                       rdo.coef(3,:);     %X4
                       rdo.coef(4,:);     %X5
                       rdo.coef(5,:);     %Z1
                       NaN(1,2);          %Z2: empty
                       rdo.coef(6,:);     %Z3
                       NaN(1,2);          %phi1
                       rdo.coef(7,:);     %phi2
                       NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(11).resi=rdo.resi;
            rdoeq2(11).stats=rdo.stats;

%===========================================================================
```

```
            %Method 12: HTBMS
%===========================================================================
            ebarBMS=Pmu*y-Pmu*R*inv(R'*Pmu*PBBMS*Pmu*R)*R'*Pmu*PBBMS*Pmu*y;
            sigma2BMS=((ebarBMS'*Pmu*ebarBMS)/(N-K-1))^0.5;
            WBMS=(1/sigma1)*Qmu+(1/sigma2BMS)*Pmu;
            [rdo.coef,rdo.resi,rdo.stats]=G2SLS(R,WBMS,PABBMS,y,N,T);
            rdoeq2(12).coef=[rdo.coef(1,:); %y1
                                       NaN(1,2);          %y2: empty
                                       rdo.coef(2,:);     %X1
                                       NaN(2,2);          %X2,X3: empty
                                       rdo.coef(3,:);     %X4
                                       rdo.coef(4,:);     %X5
                                       rdo.coef(5,:);     %Z1
                                       NaN(1,2);          %Z2: empty
                                       rdo.coef(6,:);     %Z3
                                       NaN(1,2);          %phi1
                                       rdo.coef(7,:);     %phi2
                                       NaN(2,2)];         %mu1,mu2: empty
            rdoeq2(12).resi=rdo.resi;
            rdoeq2(12).stats=rdo.stats;

%===========================================================================
            %Method 13,14,15: G3SPD_FEStd
%===========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G2SLS(Rwave,Qmu,eye(N*T),y,N,T);
            rdoeq2(13).coef=[rdo(1).coef(1,:); %y1
                                       NaN(1,2);          %y2: empty
                                       rdo(1).coef(2,:);     %X1
                                       NaN(2,2);          %X2,X3:empty
                                       rdo(1).coef(3,:);     %X4
                                       rdo(1).coef(4,:);     %X5
                                       NaN(7,2)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq2(13).resi=rdo(1).resi;
            rdoeq2(13).stats=rdo(1).stats;

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %vector of mean residuals (dependent↙
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,eye(N*T),uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar));
            rdoeq2(14).coef=[ NaN(7,2);          %y1,y2,X1,X2,X3,X4,X5: empty
                                        rdo(2).coef(1,:);     %Z1
                                        NaN(1,2);             %Z2: empty
                                        rdo(2).coef(2,:);     %Z3
                                        NaN(1,2);             %phi1: empty
                                        rdo(2).coef(3,:);     %phi2
                                        NaN(2,2)];            %mu1,mu2: empty
            rdoeq2(14).resi=rdo(2).resi;
            rdoeq2(14).stats=rdo(2).stats;

            % 3rd stage
```

```
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdo(2).resi,ones(N*T,1)]; %full dataset, including↙
all the original vars. plus the estimated Fixed Effects
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),eye(N*T),y,N,↙
T);
            rdoeq2(15).coef=[rdo(3).coef(1,:); %y1
                                   NaN(1,4);         %y2: empty
                                   rdo(3).coef(2,:); %X1
                                   NaN(2,4);         %X2,X3: empty
                                   rdo(3).coef(3,:); %X4
                                   rdo(3).coef(4,:); %X5
                                   rdo(3).coef(5,:); %Z1
                                   NaN(1,4);         %Z2: empty
                                   rdo(3).coef(6,:); %Z3
                                   NaN(1,4);         %phi1: empty
                                   rdo(3).coef(8,:); %phi2
                                   NaN(1,4);         %mu1: empty
                                   rdo(3).coef(7,:)];%mu2
            rdoeq2(15).resi=rdo(3).resi;
            rdoeq2(15).stats=rdo(3).stats;

%==========================================================================
            %Method 16,17,18: G3SPD_FECornWI
%==========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G2SLS(Rwave,Qmu,PA,y,N,T);
            rdoeq2(16).coef=[rdo(1).coef(1,:);  %y1
                                   NaN(1,2);         %y2: empty
                                   rdo(1).coef(2,:);    %X1
                                   NaN(2,2);         %X2,X3:empty
                                   rdo(1).coef(3,:);    %X4
                                   rdo(1).coef(4,:);    %X5
                                   NaN(7,2)];        %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdoeq2(16).resi=rdo(1).resi;
            rdoeq2(16).stats=rdo(1).stats;

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent↙
var)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PB,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar));
            rdoeq2(17).coef=[ NaN(7,2);         %y1,y2,X1,X2,X3,X4,X5: empty
                                    rdo(2).coef(1,:);    %Z1
                                    NaN(1,2);            %Z2: empty
                                    rdo(2).coef(2,:);    %Z3
                                    NaN(1,2);            %phi1: empty
                                    rdo(2).coef(3,:);    %phi2
                                    NaN(2,2)];           %mu1,mu2: empty
            rdoeq2(17).resi=rdo(2).resi;
            rdoeq2(17).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdo(2).resi,ones(N*T,1)]; %full dataset, including↙
```

```
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq2(18).coef=[rdo(3).coef(1,:); %y1
                                   NaN(1,4);         %y2: empty
                                   rdo(3).coef(2,:); %X1
                                   NaN(2,4);         %X2,X3: empty
                                   rdo(3).coef(3,:); %X4
                                   rdo(3).coef(4,:); %X5
                                   rdo(3).coef(5,:); %Z1
                                   NaN(1,4);         %Z2: empty
                                   rdo(3).coef(6,:); %Z3
                                   NaN(1,4);         %phi1: empty
                                   rdo(3).coef(8,:); %phi2
                                   NaN(1,4);         %mu1: empty
                                   rdo(3).coef(7,:)];%mu2
            rdoeq2(18).resi=rdo(3).resi;
            rdoeq2(18).stats=rdo(3).stats;

%==========================================================================
            %Method 19,20: G3SPD_HTHT
%==========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBHT,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar));
            rdoeq2(19).coef=[ NaN(7,2);          %y1,y2,X1,X2,X3,X4,X5: empty
                                   rdo(2).coef(1,:);    %Z1
                                   NaN(1,2);            %Z2: empty
                                   rdo(2).coef(2,:);    %Z3
                                   NaN(1,2);            %phi1: empty
                                   rdo(2).coef(3,:);    %phi2
                                   NaN(2,2)];           %mu1,mu2: empty
            rdoeq2(19).resi=rdo(2).resi;
            rdoeq2(19).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdo(2).resi,ones(N*T,1)]; %full dataset, including
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq2(20).coef=[rdo(3).coef(1,:); %y1
                                   NaN(1,4);         %y2: empty
                                   rdo(3).coef(2,:); %X1
                                   NaN(2,4);         %X2,X3: empty
                                   rdo(3).coef(3,:); %X4
```

```
                                rdo(3).coef(4,:); %X5
                                rdo(3).coef(5,:); %Z1
                                NaN(1,4);         %Z2: empty
                                rdo(3).coef(6,:); %Z3
                                NaN(1,4);         %phi1: empty
                                rdo(3).coef(8,:); %phi2
                                NaN(1,4);         %mu1: empty
                                rdo(3).coef(7,:)];%mu2
            rdoeq2(20).resi=rdo(3).resi;
            rdoeq2(20).stats=rdo(3).stats;

%===========================================================================
            %Method 21,22: G3SPD_HTAM
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent↙
variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBAM,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar));
            rdoeq2(21).coef=[ NaN(7,2);         %y1,y2,X1,X2,X3,X4,X5: empty
                                 rdo(2).coef(1,:);    %Z1
                                 NaN(1,2);            %Z2: empty
                                 rdo(2).coef(2,:);    %Z3
                                 NaN(1,2);            %phi1: empty
                                 rdo(2).coef(3,:);    %phi2
                                 NaN(2,2)];           %mu1,mu2: empty
            rdoeq2(21).resi=rdo(2).resi;
            rdoeq2(21).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdo(2).resi,ones(N*T,1)]; %full dataset, including↙
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq2(22).coef=[rdo(3).coef(1,:); %y1
                                NaN(1,4);         %y2: empty
                                rdo(3).coef(2,:); %X1
                                NaN(2,4);         %X2,X3: empty
                                rdo(3).coef(3,:); %X4
                                rdo(3).coef(4,:); %X5
                                rdo(3).coef(5,:); %Z1
                                NaN(1,4);         %Z2: empty
                                rdo(3).coef(6,:); %Z3
                                NaN(1,4);         %phi1: empty
                                rdo(3).coef(8,:); %phi2
                                NaN(1,4);         %mu1: empty
                                rdo(3).coef(7,:)];%mu2
            rdoeq2(22).resi=rdo(3).resi;
```

```
            rdoeq2(22).stats=rdo(3).stats;

%===========================================================================
            %Method 23,24: G3SPD_HTBMSWI
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=Pmu*y-Pmu*Rwave*rdo(1).coef(:,1); %matrix of residuals (dependent variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G2SLS(Rbar,Pmu,PBBMS,uhat,N,T);
            rdo(2).coef(:,2)=rdo(2).coef(:,2)*((N*T-Kbar)/(N-Kbar));
            rdoeq2(23).coef=[ NaN(7,2);         %y1,y2,X1,X2,X3,X4,X5: empty
                                rdo(2).coef(1,:);    %Z1
                                NaN(1,2);            %Z2: empty
                                rdo(2).coef(2,:);    %Z3
                                NaN(1,2);            %phi1: empty
                                rdo(2).coef(3,:);    %phi2
                                NaN(2,2)];           %mu1,mu2: empty
            rdoeq2(23).resi=rdo(2).resi;
            rdoeq2(23).stats=rdo(2).stats;

            % 3rd stage
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdo(2).resi,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G2SLSM(Rfull,eye(N*T),PC,y,N,T);
            rdoeq2(24).coef=[rdo(3).coef(1,:); %y1
                               NaN(1,4);         %y2: empty
                               rdo(3).coef(2,:); %X1
                               NaN(2,4);         %X2,X3: empty
                               rdo(3).coef(3,:); %X4
                               rdo(3).coef(4,:); %X5
                               rdo(3).coef(5,:); %Z1
                               NaN(1,4);         %Z2: empty
                               rdo(3).coef(6,:); %Z3
                               NaN(1,4);         %phi1: empty
                               rdo(3).coef(8,:); %phi2
                               NaN(1,4);         %mu1: empty
                               rdo(3).coef(7,:)];%mu2
            rdoeq2(24).resi=rdo(3).resi;
            rdoeq2(24).stats=rdo(3).stats;

            % Save results
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');
```

```
        end
        toc
    end
end
```

```
%==============================================================
% Function ESTIMAEQ2_V2
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%================================================================

function estimaeq2_v2
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    D=single(kron(eye(N),ones(1,T)*(1/T))); %reduce matrix from NT*1 to N*1

    for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3); mu1=single(mu1); mu2=single(mu2);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=Qmu*([X1,X2,X3,X4,X5]); %within instruments
        PA=A*inv(A'*A)*A';
        B=Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]); %between instruments
        PB=B*inv(B'*B)*B';
        PAB=PA+PB; %projection matrix of instruments
        BHT=Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]; %between instruments
        if rank(BHT)<size(BHT,2);BHT=BHT(:,1:rank(BHT));end
        PBHT=BHT*inv(BHT'*BHT)*BHT';
        PABHT=PA+PBHT; %projection matrix of instruments
```

```
            invZPZ=inv(Rbar'*Pmu*Rbar); % this matrix is the same for all 1000 observations and all methods!!!
            PZ_invZPZ_ZP=Pmu*Rbar*invZPZ*Rbar'*Pmu; % this matrix is the same for all 1000 observations and all methods!!!

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2));
            Y=single(datos(1,obs).Y(:,1));
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);
            Rwave=[Y X1 X4 X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);
            %open data for the observation
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq2');

%==========================================================================
            %Method 13,14,15: G3SPD_FEStd
%==========================================================================
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq2(14).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq2(14).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    NaN(1,1);            %Z2: empty
                                    stdernew(2,:);    %Z3
                                    NaN(1,1);            %phi1: empty
                                    stdernew(3,:);    %phi2
                                    NaN(2,1)];           %mu1,mu2: empty

            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X4,X5,Z1,Z3,resi2,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq2(15).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq2(15).coef(14,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq2(15).coef(14,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq2(15).coef(:,5)=[stderMT(1,:); %y1
                                  NaN(1,1);         %y2: empty
                                  stderMT(2,:); %X1
                                  NaN(2,1);         %X2,X3: empty
```

```
                                        stderMT(3,:); %X4
                                        stderMT(4,:); %X5
                                        stderMT(5,:); %Z1
                                        NaN(1,1);         %Z2: empty
                                        stderMT(6,:); %Z3
                                        NaN(1,1);         %phi1: empty
                                        stderMT(8,:); %phi2
                                        NaN(1,1);         %mu1: empty
                                        stderMT(7,:)];%mu2

%===========================================================================
            %Method 16,17,18: G3SPD_FECorn
%===========================================================================
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq2(17).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq2(17).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    NaN(1,1);            %Z2: empty
                                    stdernew(2,:);    %Z3
                                    NaN(1,1);            %phi1: empty
                                    stdernew(3,:);    %phi2
                                    NaN(2,1)];           %mu1,mu2: empty

            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X4,X5,Z1,Z3,resi2,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq2(18).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq2(18).coef(14,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq2(18).coef(14,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq2(18).coef(:,5)=[stderMT(1,:); %y1
                                  NaN(1,1);         %y2: empty
                                  stderMT(2,:); %X1
                                  NaN(2,1);         %X2,X3: empty
                                  stderMT(3,:); %X4
                                  stderMT(4,:); %X5
                                  stderMT(5,:); %Z1
                                  NaN(1,1);         %Z2: empty
                                  stderMT(6,:); %Z3
                                  NaN(1,1);         %phi1: empty
                                  stderMT(8,:); %phi2
                                  NaN(1,1);         %mu1: empty
                                  stderMT(7,:)];%mu2
```

```
%=======================================================================
            %Method 19,20: G3SPD_HTHT
%=======================================================================
            % 2nd stage: reestimacion varianza comun (ajuste de dof)
            resi2=squeeze(rdoeq2(19).resi); %residuals 2nd stage
            vari2=(resi2'*resi2)/(N-size(Rbar,2));
            stdernew=diag(vari2*invZPZ).^0.5;
            rdoeq2(19).coef(:,2)=[  NaN(7,1);           %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    NaN(1,1);            %Z2: empty
                                    stdernew(2,:);    %Z3
                                    NaN(1,1);            %phi1: empty
                                    stdernew(3,:);    %phi2
                                    NaN(2,1)];           %mu1,mu2: empty

            % 3rd stage: varianza Murphy & Topel (1985)
            Rfull=[Y,X1,X4,X5,Z1,Z3,resi2,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            resi3=squeeze(rdoeq2(20).resi);
            vari3=(resi3'*resi3)/(N*T-size(Rfull,2));
            invWW_W=inv(Rfull'*Rfull)*Rfull';
            parte1=vari3*inv(Rfull'*Rfull);
            comp1=(squeeze(rdoeq2(20).coef(14,1))^2)*vari2;
            parte2=comp1*invWW_W*PZ_invZPZ_ZP*invWW_W';
            comp2=2*squeeze(rdoeq2(20).coef(14,1));
            comp3=resi3*resi2';
            parte3=comp2*invWW_W*comp3*PZ_invZPZ_ZP*invWW_W';
            stderMT=diag(parte1+parte2+parte3).^0.5;
            rdoeq2(20).coef(:,5)=[stderMT(1,:); %y1
                                  NaN(1,1);         %y2: empty
                                  stderMT(2,:); %X1
                                  NaN(2,1);         %X2,X3: empty
                                  stderMT(3,:); %X4
                                  stderMT(4,:); %X5
                                  stderMT(5,:); %Z1
                                  NaN(1,1);         %Z2: empty
                                  stderMT(6,:); %Z3
                                  NaN(1,1);         %phi1: empty
                                  stderMT(8,:); %phi2
                                  NaN(1,1);         %mu1: empty
                                  stderMT(7,:)];%mu2

            % Save results
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');
        end
        toc
    end
```

```
end
```

```
%=======================================================================
% Function ESTIMAEQ2_V3
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%===========================================================================

function estimaeq2_v3
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %up to 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear('ZMu');

    for rho=1:1:size(rhocases,1)  % eleven rho (rhomu/rhoepsilon) coefficients
        disp( '         --------------------------------');
        disp(['         Rho= ' num2str(rho)]);
        disp( '         --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2));
            Y=single(datos(1,obs).Y(:,1));
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
endogenous
            Rwave=[Y X1 X4 X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
                invXQX=inv(Rwave'*Qmu*Rwave);
                invRPR=inv(R'*Pmu*R);
            %open data for the observation
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
```

```
            load(filename,'rdoeq2');

%=========================================================================
            %Method 2: FEStd
%=========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(2).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq2(2).coef(:,2)=[stdernew(1,:);  %y1
                                   NaN(1,1);          %y2: empty
                                   stdernew(2,:);     %X1
                                   NaN(2,1);          %X2,X3:empty
                                   stdernew(3,:);     %X4
                                   stdernew(4,:);     %X5
                                   NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%=========================================================================
            %Method 3: FEBal
%=========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(3).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq2(3).coef(:,2)=[stdernew(1,:);  %y1
                                   NaN(1,1);          %y2: empty
                                   stdernew(2,:);     %X1
                                   NaN(2,1);          %X2,X3:empty
                                   stdernew(3,:);     %X4
                                   stdernew(4,:);     %X5
                                   NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%=========================================================================
            %Method 4: BEStd
%=========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(4).resi); %residuals
            vari=(resi'*resi)/(N-size(R,2));
            stdernew=diag(vari*invRPR).^0.5; %new std. errors
            rdoeq2(4).coef(:,2)=[stdernew(1,:);  %y1
                                   NaN(1,1);          %y2: empty
                                   stdernew(2,:);     %X1
                                   NaN(2,1);          %X2,X3: empty
                                   stdernew(3,:);     %X4
                                   stdernew(4,:);     %X5
                                   stdernew(5,:);     %Z1
                                   NaN(1,1);          %Z2: empty
                                   stdernew(6,:);     %Z3
                                   NaN(1,1);          %phi1
                                   stdernew(7,:);     %phi2
                                   NaN(2,1)];         %mu1,mu2: empty
```

```
%======================================================================
            %Method 5: BEBal
%==========================================================================
            % reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(5).resi); %residuals
            vari=(resi'*resi)/(N-size(R,2));
            stdernew=diag(vari*invRPR).^0.5; %new std. errors
            rdoeq2(5).coef(:,2)=[stdernew(1,:);  %y1
                                  NaN(1,1);          %y2: empty
                                  stdernew(2,:);     %X1
                                  NaN(2,1);          %X2,X3: empty
                                  stdernew(3,:);     %X4
                                  stdernew(4,:);     %X5
                                  stdernew(5,:);     %Z1
                                  NaN(1,1);          %Z2: empty
                                  stdernew(6,:);     %Z3
                                  NaN(1,1);          %phi1
                                  stdernew(7,:);     %phi2
                                  NaN(2,1)];         %mu1,mu2: empty

%==========================================================================
            %Method 13: G3SPD_FEStd
%==========================================================================
            % 1st stage: reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(13).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq2(13).coef(:,2)=[stdernew(1,:); %y1
                                  NaN(1,1);          %y2: empty
                                  stdernew(2,:);     %X1
                                  NaN(2,1);          %X2,X3:empty
                                  stdernew(3,:);     %X4
                                  stdernew(4,:);     %X5
                                  NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%==========================================================================
            %Method 16: G3SPD_FECorn
%==========================================================================
            % 1st stage: reestimacion varianza comun (ajuste de dof)
            resi=squeeze(rdoeq2(16).resi); %residuals
            vari=(resi'*resi)/(N*(T-1)-size(Rwave,2));
            stdernew=diag(vari*invXQX).^0.5; %new std. errors
            rdoeq2(16).coef(:,2)=[stdernew(1,:); %y1
                                  NaN(1,1);          %y2: empty
                                  stdernew(2,:);     %X1
                                  NaN(2,1);          %X2,X3:empty
                                  stdernew(3,:);     %X4
                                  stdernew(4,:);     %X5
                                  NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty


            % Save results
```

```
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');
        end
        toc
    end
end
```

```
%==========================================================================
% Function ESTIMAEQ2_V4
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%==========================================================================

function estimaeq2_v4
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    clear('ZMu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            Y=single(datos(1,obs).Y(:,1));

            %open data for the observation
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
```

```
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');

%===========================================================================
            %Method 13,14,15: G3SPD_Std
%===========================================================================
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdoeq2(14).resi,ones(N*T,1)]; %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covariance matrix
            sigma3rd=(rdoeq2(15).resi'*rdoeq2(15).resi)/(N*T-K);
            omegaGMM=sigma3rd*inv(Rfull'*Rfull);

            % WH-HC1: stored in column 6
            sigmaHC1=diag((rdoeq2(15).resi).^2);
            omegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(omegaHC1).^0.5;
            clear('sigmaHC1','omegaHC1');

            % WH-HC2: stored in column 7
            sigmaHC2=diag(((rdoeq2(15).resi).^2)./(1-diag(PR)));
            omegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq2(15).resi)./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            omegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            omegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % BK: stored in column 10
            E=reshape(rdoeq2(15).resi,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix↙
for each time period.
            omegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(omegaBK).^0.5;
            clear('E','sigmaBK','omegaBK');

            % MT covariance matrix: stored in column 11
            sigma2nd=((rdoeq2(14).resi'*rdoeq2(14).resi))/(N-Kbar);
            Psi1=omegaGMM;
```

```
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*rdoeq2(15).resi)*...
                (rdoeq2(14).resi'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            omegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(omegaMT).^0.5;
            clear('sigma3rd','sigma2nd','omegaGMM','Psi1','Psi2','Psi3','omegaMT');

            % store standard errors
            stder=[stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(15).coef(:,6:11)=[   NaN(1,6);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,6);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,6);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,6);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,6)];     %mu2: empty


% %==========================================================================
%             %Method 16,17,18: G3SPD_Corn
% %==========================================================================
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdoeq2(17).resi,ones(N*T,1)]; %full dataset,
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdoeq2(17).resi,ones(N*T,1)]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag(((rdoeq2(18).resi).^2)./(1-diag(PR)))*C;
            omegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq2(18).resi)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            omegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            omegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv
```

```
(RCinvCC*C'*Rfull);
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % Shift BK and MT one column to the right: stored in columns 10
            % and 11
            rdoeq2(18).coef(:,10:11)=rdoeq2(18).coef(:,9:10);

            % store standard errors
            stder=[stderHC2,stderLE,stderHC3];
            rdoeq2(18).coef(:,7:9)=[    NaN(1,3);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,3);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,3);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,3);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,3)];     %mu2: empty

% %=============================================================================
%              %Method 19,20: G3SPD_HTM
% %=============================================================================
            Rfull=[Y,X1,X4,X5,Z1,Z3,rdoeq2(19).resi,ones(N*T,1)]; %full dataset,
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdoeq2(19).resi,ones(N*T,1)]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag(((rdoeq2(20).resi).^2)./(1-diag(PR)))*C;
            omegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC2=diag(omegaHC2).^0.5;
            clear('sigmaHC2','omegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(rdoeq2(20).resi)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            omegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderLE=diag(omegaLE).^0.5;
            clear('omegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            omegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv
(RCinvCC*C'*Rfull);
```

```
            stderHC3=diag(omegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','omegaHC3');

            % Shift BK and MT one column to the right: stored in columns 10
            % and 11
            rdoeq2(20).coef(:,10:11)=rdoeq2(20).coef(:,9:10);

            % store standard errors
            stder=[stderHC2,stderLE,stderHC3];
            rdoeq2(20).coef(:,7:9)=[    NaN(1,3);      %y1: empty
                                        stder(1,:);    %y2
                                        stder(2,:);    %X1
                                        stder(3,:);    %X2
                                        stder(4,:);    %X3
                                        NaN(2,3);      %X4,X5: empty
                                        stder(5,:);    %Z1
                                        stder(6,:);    %Z2
                                        NaN(1,3);      %Z3: empty
                                        stder(8,:);    %phi1
                                        NaN(1,3);      %phi2: empty
                                        stder(7,:);    %mu1
                                        NaN(1,3)];     %mu2: empty

            % Save results
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');

        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMAEQ2_V5
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%=============================================================================

function estimaeq2_v5
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        PAB=PA+PB; %projection matrix of instruments
        BHT=single(Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';
```

```
        PABHT=PA+PBHT; %projection matrix of instruments

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                    Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
endogenous
            K=size(R,2);
            Rwave=[Y,X1,X4,X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

            %open data for the observation
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');

%==========================================================================
%           Method 3: FEBal
%==========================================================================
            resi=squeeze(rdoeq2(3).resi); %residuals
            vari=(resi'*Qmu*resi)/(N*(T-1)-Kwave);
            stdernew=diag(vari*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq2(3).coef(:,2)=[stdernew(1,:);  %y1
                                 NaN(1,1);          %y2: empty
                                 stdernew(2,:);     %X1
                                 NaN(2,1);          %X2,X3:empty
                                 stdernew(3,:);     %X4
                                 stdernew(4,:);     %X5
                                 NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

%==========================================================================
%           Method 5: BEBal
%==========================================================================
            resi=squeeze(rdoeq2(5).resi); %residuals
            vari=(resi'*Pmu*resi)/(N-K);
            stdernew=diag(vari*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            rdoeq2(5).coef(:,2)=[stdernew(1,:);  %y1
                                 NaN(1,1);          %y2: empty
                                 stdernew(2,:);     %X1
                                 NaN(2,1);          %X2,X3: empty
                                 stdernew(3,:);     %X4
                                 stdernew(4,:);     %X5
                                 stdernew(5,:);     %Z1
                                 NaN(1,1);          %Z2: empty
                                 stdernew(6,:);     %Z3
                                 NaN(1,1);          %phi1
                                 stdernew(7,:);     %phi2
                                 NaN(2,1)];         %mu1,mu2: empty
```

```
%======================================================================
%            Method 6: ECWH
%==========================================================================
            resi=squeeze(rdoeq2(6).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq2(6).coef(:,2)=[stdernew(1,:);  %y1
                                 NaN(1,1);         %y2: empty
                                 stdernew(2,:);    %X1
                                 NaN(2,1);         %X2,X3: empty
                                 stdernew(3,:);    %X4
                                 stdernew(4,:);    %X5
                                 stdernew(5,:);    %Z1
                                 NaN(1,1);         %Z2: empty
                                 stdernew(6,:);    %Z3
                                 NaN(1,1);         %phi1
                                 stdernew(7,:);    %phi2
                                 NaN(2,1)];        %mu1,mu2: empty

%==========================================================================
%            Method 7: ECAm
%==========================================================================
            resi=squeeze(rdoeq2(7).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq2(7).coef(:,2)=[stdernew(1,:);  %y1
                                 NaN(1,1);         %y2: empty
                                 stdernew(2,:);    %X1
                                 NaN(2,1);         %X2,X3: empty
                                 stdernew(3,:);    %X4
                                 stdernew(4,:);    %X5
                                 stdernew(5,:);    %Z1
                                 NaN(1,1);         %Z2: empty
                                 stdernew(6,:);    %Z3
                                 NaN(1,1);         %phi1
                                 stdernew(7,:);    %phi2
                                 NaN(2,1)];        %mu1,mu2: empty

%==========================================================================
%            Method 8: ECSA
%==========================================================================
            resi=squeeze(rdoeq2(8).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*R)).^0.5; %new std. errors
            rdoeq2(8).coef(:,2)=[stdernew(1,:);  %y1
                                 NaN(1,1);         %y2: empty
```

```
                                      stdernew(2,:);    %X1
                                      NaN(2,1);         %X2,X3: empty
                                      stdernew(3,:);    %X4
                                      stdernew(4,:);    %X5
                                      stdernew(5,:);    %Z1
                                      NaN(1,1);         %Z2: empty
                                      stdernew(6,:);    %Z3
                                      NaN(1,1);         %phi1
                                      stdernew(7,:);    %phi2
                                      NaN(2,1)];        %mu1,mu2: empty

%===========================================================================
%            Method 9: ECCorn
%===========================================================================
            resi=squeeze(rdoeq2(9).resi); %residuals
            sigma1=((resi'*Qmu*resi)/(N*(T-1)-K))^0.5; %"within" component of variance
            sigma2=((resi'*Pmu*resi)/(N-K))^0.5; %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1/2)
            stdernew=diag(inv(R'*invsigma*PAB*invsigma*R)).^0.5; %new std. errors
            rdoeq2(9).coef(:,2)=[stdernew(1,:);  %y1
                                      NaN(1,1);         %y2: empty
                                      stdernew(2,:);    %X1
                                      NaN(2,1);         %X2,X3: empty
                                      stdernew(3,:);    %X4
                                      stdernew(4,:);    %X5
                                      stdernew(5,:);    %Z1
                                      NaN(1,1);         %Z2: empty
                                      stdernew(6,:);    %Z3
                                      NaN(1,1);         %phi1
                                      stdernew(7,:);    %phi2
                                      NaN(2,1)];        %mu1,mu2: empty

%===========================================================================
%            Method 10: HTM
%===========================================================================
            resi=squeeze(rdoeq2(10).resi); %residuals
            sigma1=(resi'*Qmu*resi)/(N*(T-1)-K); %"within" component of variance
            sigma2=(resi'*Pmu*resi)/(N-K); %"between" component of variance
            invsigma=(1/sigma1)*Qmu+(1/sigma2)*Pmu; % sigma^(-1)
            stdernew=diag(inv(R'*invsigma*PABHT*invsigma*R)).^0.5; %new std. errors
            rdoeq2(10).coef(:,2)=[stdernew(1,:);  %y1
                                      NaN(1,1);         %y2: empty
                                      stdernew(2,:);    %X1
                                      NaN(2,1);         %X2,X3: empty
                                      stdernew(3,:);    %X4
                                      stdernew(4,:);    %X5
                                      stdernew(5,:);    %Z1
                                      NaN(1,1);         %Z2: empty
                                      stdernew(6,:);    %Z3
                                      NaN(1,1);         %phi1
                                      stdernew(7,:);    %phi2
                                      NaN(2,1)];        %mu1,mu2: empty
```

```
%=========================================================================
%            Method 16,17,18: G3SPD_Corn
%=========================================================================
        %1st Step
            resi=single(squeeze(rdoeq2(16).resi)); %residuals
            vari=(resi'*Qmu*resi)/(N*(T-1)-Kwave);
            stdernew=diag(vari*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq2(16).coef(:,2)=[stdernew(1,:);  %y1
                                   NaN(1,1);          %y2: empty
                                   stdernew(2,:);     %X1
                                   NaN(2,1);          %X2,X3:empty
                                   stdernew(3,:);     %X4
                                   stdernew(4,:);     %X5
                                   NaN(7,1)];         %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty

        % 2nd Step
            resi2nd=single(squeeze(rdoeq2(17).resi)); %residuals 2nd stage
            sigma2nd=(resi2nd'*Pmu*resi2nd)/(N-Kbar);
            stdernew=diag(sigma2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            rdoeq2(17).coef(:,2)=[ NaN(7,1);         %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    NaN(1,1);            %Z2: empty
                                    stdernew(2,:);    %Z3
                                    NaN(1,1);            %phi1: empty
                                    stdernew(3,:);    %phi2
                                    NaN(2,1)];           %mu1,mu2: empty

        % 3rd Step
            resi3rd=single(squeeze(rdoeq2(18).resi)); %residuals 2nd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all↙
the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';

            % GMM:
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(Rfull'*PC*Rfull);
            stdernew=diag(OmegaGMM).^0.5;
            rdoeq2(18).coef(:,2)=[    NaN(1,1);      %y1: empty
                                        stdernew(1,:);    %y2
                                        stdernew(2,:);    %X1
                                        stdernew(3,:);    %X2
                                        stdernew(4,:);    %X3
                                        NaN(2,1);      %X4,X5: empty
                                        stdernew(5,:);    %Z1
                                        stdernew(6,:);    %Z2
                                        NaN(1,1);      %Z3: empty
                                        stdernew(8,:);    %phi1
                                        NaN(1,1);      %phi2: empty
                                        stdernew(7,:);    %mu1
                                        NaN(1,1)];     %mu2: empty
```

```
            % MT
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PB*Pmu*Rbar*inv↙
(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PB*Pmu*Rbar*inv↙
(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stdernew=diag(OmegaMT).^0.5;
            rdoeq2(18).coef(:,11)=[ NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

%==========================================================================
%           Method 19,20: G3SPD_HTM
%==========================================================================
        % 2nd Step
            resi2nd=single(squeeze(rdoeq2(19).resi)); %residuals 2nd stage
            sigma2nd=(resi2nd'*Pmu*resi2nd)/(N-Kbar);
            stdernew=diag(sigma2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
            rdoeq2(19).coef(:,2)=[  NaN(7,1);            %y1,y2,X1,X2,X3,X4,X5: empty
                                    stdernew(1,:);    %Z1
                                    stdernew(2,:);    %Z2
                                    NaN(1,1);            %Z3: empty
                                    stdernew(3,:);    %phi1
                                    NaN(3,1)];           %phi2,mu1,mu2: empty

        % 3rd Step
            resi3rd=single(squeeze(rdoeq2(20).resi)); %residuals 2nd stage
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all↙
the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';

            % GMM:
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(Rfull'*PC*Rfull);
            stdernew=diag(OmegaGMM).^0.5;
            rdoeq2(20).coef(:,2)=[  NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
```

```
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

            % MT
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*inv↙
(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PBHT*Pmu*Rbar*inv↙
(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stdernew=diag(OmegaMT).^0.5;
            rdoeq2(20).coef(:,11)=[ NaN(1,1);      %y1: empty
                                    stdernew(1,:);    %y2
                                    stdernew(2,:);    %X1
                                    stdernew(3,:);    %X2
                                    stdernew(4,:);    %X3
                                    NaN(2,1);      %X4,X5: empty
                                    stdernew(5,:);    %Z1
                                    stdernew(6,:);    %Z2
                                    NaN(1,1);      %Z3: empty
                                    stdernew(8,:);    %phi1
                                    NaN(1,1);      %phi2: empty
                                    stdernew(7,:);    %mu1
                                    NaN(1,1)];     %mu2: empty

%==========================================================================
%           Method 25: G3SPD_HTMnew
%=========================================================================
        % 3rd Step
            Rfull=[Y,X1,X2,X3,Z1,Z2,resi2nd,ones(N*T,1)]; %full dataset, including all↙
the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resi2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);

            % coefficients: stored in column 1
            theta=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi3rd=y-Rfull*theta; %residuals

            % GMM: stored in column 2
            sigma3rd=(resi3rd'*resi3rd)/(N*T-K);
            OmegaGMM=sigma3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
```

```
            % WH-HC1: stored in column 6
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv
(RC1*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('OmegaHC1');

            % WH-HC2: stored in column 7
            sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv
(RC1*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE: stored in column 8
            epsilonast=(resi3rd)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv
(RC1*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3: stored in column 9
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv
(RC1*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK: stored in column 10
            E=reshape(resi3rd,T,N); %arrange the set of OLS residuals
            VBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix for
each time period.
            sigmaBK=C'*VBK*C;
            OmegaBK=inv(Rfull'*PC*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv
(Rfull'*PC*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','VBK','sigmaBK','OmegaBK');

            % MT: stored in column 11
            Psi1=OmegaGMM;
            Psi2=(sigma2nd/(sigma3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*inv
(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigma3rd^2))*(Rfull'*PC*resi3rd)*(resi2nd'*PBHT*Pmu*Rbar*inv
(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;

            % store standard errors and residuals
            stder=[theta,stderGMM,zeros(8,3),stderHC1,stderHC2,stderLE,stderHC3,stderBK,
stdMT];
            rdoeq2(25).coef=[NaN(1,11);      %y1: empty
```

```
                                stder(1,:);    %y2
                                stder(2,:);    %X1
                                stder(3,:);    %X2
                                stder(4,:);    %X3
                                NaN(2,11);      %X4,X5: empty
                                stder(5,:);    %Z1
                                stder(6,:);    %Z2
                                NaN(1,11);      %Z3: empty
                                stder(8,:);    %phi1
                                NaN(1,11);      %phi2: empty
                                stder(7,:);    %mu1
                                NaN(1,11)];     %mu2: empty
            rdoeq2(25).resi=resi3rd;
            rdoeq2(25).stats=[];


            % Save results
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');

        end
        toc
    end
end
```

```
%=====================================================================
% Function ESTIMAEQ2_V6
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%=========================================================================

function estimaeq2_v6
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y1, X1, X4, X5, Z1, Z3, VarPhi2 (constant), Phi2 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        BHT=single(Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';
```

```
        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
endogenous
            K=size(R,2);
            Rwave=[Y,X1,X4,X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

%==========================================================================
%              Method 1: OLS
%==========================================================================
            thetaOLS=inv(R'*R)*R'*y;
            resiOLS=y-R*thetaOLS;
            sigmaOLS=(resiOLS'*resiOLS)/(N*T-K);
            stderOLS=diag(sigmaOLS*inv(R'*R)).^0.5; %new std. errors
            estimadores=[thetaOLS,stderOLS];
            rdoeq2(1).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq2(1).resi=resiOLS;
            clear('thetaOLS','resiOLS','sigmaOLS','stderOLS','estimadores');

%==========================================================================
%            Method 2: FEStd
%==========================================================================
            thetaFEStd=inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu*y;
            resiFEStd=Qmu*y-Qmu*Rwave*thetaFEStd;
            sigmaFEStd=(resiFEStd'*Qmu*resiFEStd)/(N*(T-1)-Kwave);
            stderFEStd=diag(sigmaFEStd*inv(Rwave'*Qmu*Rwave)).^0.5; %new std. errors
            estimadores=[thetaFEStd,stderFEStd];
            rdoeq2(2).coef=[estimadores; NaN(4,2)]; %Z1, Z2, VarPhi1, Phi1: empty
            rdoeq2(2).resi=resiFEStd;
            clear('resiFEStd','sigmaFEStd','stderFEStd','estimadores');

%==========================================================================
%            Method 3: FEBal
%==========================================================================
            thetaFEBal=inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
            resiFEBal=Qmu*y-Qmu*Rwave*thetaFEBal;
            sigmaFEBal=(resiFEBal'*Qmu*resiFEBal)/(N*(T-1)-Kwave);
            stderFEBal=diag(sigmaFEBal*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std.↙
errors
            estimadores=[thetaFEBal,stderFEBal];
            rdoeq2(3).coef=[estimadores; NaN(4,2)]; %Z1, Z2, VarPhi1, Phi1: empty
            rdoeq2(3).resi=resiFEBal;
            clear('resiFEBal','stderFEBal','estimadores');

%==========================================================================
%            Method 4: BEBal
%==========================================================================
            thetaBEBal=inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
```

```
            resiBEBal=Pmu*y-Pmu*R*thetaBEBal;
            sigmaBEBal=(resiBEBal'*Pmu*resiBEBal)/(N-K);
            stderBEBal=diag(sigmaBEBal*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            estimadores=[thetaBEBal,stderBEBal];
            rdoeq2(4).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq2(4).resi=resiBEBal;
            clear('thetaBEBal','resiBEBal','stderBEBal','estimadores');

%===========================================================================
%           Method 5: ECCorn
%===========================================================================
            WCorn=((1/sigmaFEBal)^0.5)*Qmu+((1/sigmaBEBal)^0.5)*Pmu; %sigma ^ (-1/2)
            thetaECCorn=inv(R'*WCorn*(PA+PB)*WCorn*R)*R'*WCorn*(PA+PB)*WCorn*y;
            resiECCorn=WCorn*y-WCorn*R*thetaECCorn;
            sigmaECCorn=(resiECCorn'*resiECCorn)/(N*T-K); %"within" component of↙
variance
            stderECCorn=diag(sigmaECCorn*inv(R'*WCorn*(PA+PB)*WCorn*R)).^0.5; %new std.↙
errors
            estimadores=[thetaECCorn,stderECCorn];
            rdoeq2(5).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq2(5).resi=resiECCorn;
            clear('WCorn','thetaECCorn','resiECCorn','sigmaECCorn',...
                'stderECCorn','sigmaBEBal','estimadores');

%===========================================================================
%           Method 6: HTM
%===========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=(ebarHT'*Pmu*ebarHT)/(N-K);
            WHT=((1/sigmaFEBal)^0.5)*Qmu+((1/sigma2HT)^0.5)*Pmu;
            thetaHT=inv(R'*WHT*(PA+PBHT)*WHT*R)*R'*WHT*(PA+PBHT)*WHT*y; %coefficients
            resiHT=WHT*y-WHT*R*thetaHT; %residuals
            sigmaHT=(resiHT'*resiHT)/(N*T-K); %"within" component of variance
            stderHT=diag(sigmaHT*inv(R'*WHT*(PA+PBHT)*WHT*R)).^0.5; %new std. errors
            estimadores=[thetaHT,stderHT];
            rdoeq2(6).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq2(6).resi=resiHT;
            clear('ebarHT','sigmaFEBal','sigma2HT','WHT','thetaHT',...
                'resiHT','sigmaHT','sigmaFEBal','stderHT','estimadores');

%===========================================================================
%             Method 7,8: G3SPD_Std
%===========================================================================
        % 1st stage:
            % idem FEStd.

        % 2nd Step:
            uhatStd=Pmu*y-Pmu*Rwave*thetaFEStd; %vector of mean residuals (dependent↙
variable)
            thetaStd=inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*uhatStd;
            resiStd2nd=Pmu*uhatStd-Pmu*Rbar*thetaStd;
            sigmaStd2nd=(resiStd2nd'*Pmu*resiStd2nd)/(N-Kbar);
```

```
            stderStd=diag(sigmaStd2nd*inv(Rbar'*Pmu*Rbar)).^0.5;
            estimadores=[thetaStd,stderStd];
            rdoeq2(7).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq2(7).resi=resiStd2nd;
            clear('uhatStd','thetaFEStd','stderStd','estimadores');

        % 3rd Step:
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),resiStd2nd]); %full dataset,↙
including all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            %Coefficients: stored in column 1
            thetaStd=inv(Rfull'*Rfull)*Rfull'*y;
            resiStd3rd=y-Rfull*thetaStd;

            % GMM covar.: stored in column 2
            sigmaStd3rd=(resiStd3rd'*resiStd3rd)/(N*T-K);
            OmegaGMM=sigmaStd3rd*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=diag(resiStd3rd.^2);
            OmegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=diag((resiStd3rd.^2)./(1-diag(PR)));
            OmegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiStd3rd./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            OmegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            OmegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiStd3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix↙
for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
```

```
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaStd2nd/(sigmaStd3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigmaStd3rd^2))*(Rfull'*resiStd3rd)*...
                (resiStd2nd'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaStd3rd','sigmaStd2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiStd2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq2(8).coef=[thetaStd,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,
stderBK,stderMT];
            rdoeq2(8).resi=resiStd3rd;
            clear('resiStd3rd','thetaStd','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','estimadores','Rfull');

%===========================================================================
%           Method 9,10: G3SPD_Corn
%===========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            uhatCorn=Pmu*y-Pmu*Rwave*thetaFEBal; %vector of mean residuals (dependent
variable)
            thetaCorn=inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*Pmu*uhatCorn;
            resiCorn2nd=Pmu*uhatCorn-Pmu*Rbar*thetaCorn; %residuals 2nd stage
            sigmaCorn2nd=(resiCorn2nd'*Pmu*resiCorn2nd)/(N-Kbar);
            stderCorn=diag(sigmaCorn2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            estimadores=[thetaCorn,stderCorn];
            rdoeq2(9).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq2(9).resi=resiCorn2nd;
            clear('uhatCorn','thetaCorn','stderCorn','estimadores');

        % 3rd Step:
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),resiCorn2nd]); %full dataset,
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),resiCorn2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaCorn=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiCorn3rd=y-Rfull*thetaCorn;

            % GMM covar.: stored in column 2
```

```
            sigmaCorn3rd=(resiCorn3rd'*resiCorn3rd)/(N*T-K);
            OmegaGMM=sigmaCorn3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiCorn3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiCorn3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiCorn3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiCorn3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaCorn2nd/(sigmaCorn3rd^2))* Rfull'*PC*PB*Pmu*Rbar*...
                inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigmaCorn3rd^2))*(Rfull'*PC*resiCorn3rd)*...
                (resiCorn2nd'*PB*Pmu*Rbar*inv(Rbar'*Pmu*PB*Pmu*Rbar)↙
*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaCorn3rd','sigmaCorn2nd','OmegaGMM','Psi1','Psi2',...
```

```
                'Psi3','OmegaMT','resiCorn2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq2(10).coef=[thetaCorn,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,
stderBK,stderMT];
            rdoeq2(10).resi=resiCorn3rd;
            clear('resiCorn3rd','thetaCorn','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');

%==========================================================================
%           Method 11,12: G3SPD_HTM (Old version using the whole set of
%                                   instruments in the 3rd Step estimation)
%==========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            uhatHTM=Pmu*y-Pmu*Rwave*thetaFEBal; %vector of mean residuals (dependent
variable)
            thetaHTM=inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*Pmu*uhatHTM;
            resiHTM2nd=Pmu*uhatHTM-Pmu*Rbar*thetaHTM; %residuals 2nd stage
            sigmaHTM2nd=(resiHTM2nd'*Pmu*resiHTM2nd)/(N-Kbar);
            stderHTM=diag(sigmaHTM2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
            estimadores=[thetaHTM,stderHTM];
            rdoeq2(11).coef=[NaN(4,2);estimadores; NaN(1,2)]; %Y,X1,X2,X3,Phi1: empty
            rdoeq2(11).resi=resiHTM2nd;
            clear('uhatHTM','thetaHTM','stderHTM','estimadores','thetaFEBal');

        % 3rd Step:
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),resiHTM2nd]); %full dataset,
including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTM=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTM3rd=y-Rfull*thetaHTM;

            % GMM covar.: stored in column 2
            sigmaHTM3rd=(resiHTM3rd'*resiHTM3rd)/(N*T-K);
            OmegaGMM=sigmaHTM3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTM3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
```

```
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTM3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTM3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTM3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTM3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTM3rd^2))*(Rfull'*PC*resiHTM3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTM3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT');

            % store coefficients and standard errors (8 columns)
            rdoeq2(12).coef=[thetaHTM,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(12).resi=resiHTM3rd;
            clear('resiHTM3rd','thetaHTM','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');
```

```
%========================================================================
%           Method 13: G3SPD_HTMNew (New version using the HT set of
%                                   instruments in the 3rd Step estimation)
%========================================================================
        % 1st Step:
            % idem FEBal.

        % 2nd Step:
            % idem G3SPD_HTM (Old version)

        % 3rd Step
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),resiHTM2nd]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTMNEW=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTMNEW3rd=y-Rfull*thetaHTMNEW;

            % GMM covar.: stored in column 2
            sigmaHTMNEW3rd=(resiHTMNEW3rd'*resiHTMNEW3rd)/(N*T-K);
            OmegaGMM=sigmaHTMNEW3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTMNEW3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTMNEW3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTMNEW3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');
```

```
            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTMNEW3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTMNEW3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTMNEW3rd^2))*(Rfull'*PC*resiHTMNEW3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTMNEW3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiHTM2nd','sigmaHTM2nd');

            % store coefficients and standard errors (8 columns)
            rdoeq2(13).coef=[thetaHTMNEW,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(13).resi=resiHTMNEW3rd;
            clear('resiHTMNEW3rd','thetaHTMNEW','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');

%=====================================================================
%           Save results
%=====================================================================
            if exist(['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')],'dir' )~=7
                mkdir(['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
            end
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');

        end
        toc
```

```
    end
end
```

```
%=======================================================================
% Function ESTIMAEQ2_V7
% This program perform complementary estimations for EQUATION 2
% Author: Diego Avanzini
%===========================================================================

function estimaeq2_v7
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['C:\Diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f' )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['C:\Diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f' )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);
        A=single(Qmu*([X1,X2,X3,X4,X5])); %within instruments
        PA=A*inv(A'*A)*A';
        B=single(Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)])); %between instruments
        PB=B*inv(B'*B)*B';
        BHT=single(Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);
            Y=single(datos(1,obs).Y(:,1));
            y=single(datos(1,obs).Y(:,2));
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            K=size(R,2);
            Rwave=[Y,X1,X4,X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);

            %open data for the observation
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq2');

% ===========================================================================
%           Method 3: FEBal
% ===========================================================================
            resiFEBal=rdoeq2(3).resi;
            sigmaFEBal=(resiFEBal'*Qmu*resiFEBal)/(N*(T-1)-Kwave);
            stderFEBal=diag(sigmaFEBal*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5; %new std. errors
            rdoeq2(3).coef(:,2)=[  stderFEBal(1,:);  %y1
                                   NaN(1,1);         %y2: empty
                                   stderFEBal(2,:);  %X1
                                   NaN(2,1);         %X2,X3:empty
                                   stderFEBal(3,:);  %X4
                                   stderFEBal(4,:);  %X5
                                   NaN(7,1)];        %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            clear('resiFEBal','sigmaFEBal','stderFEBal');

% ===========================================================================
%           Method 5: BEBal
% ===========================================================================
            resiBEBal=rdoeq2(5).resi;
            sigmaBEBal=(resiBEBal'*Pmu*resiBEBal)/(N-K);
            stderBEBal=diag(sigmaBEBal*inv(R'*Pmu*PB*Pmu*R)).^0.5; %new std. errors
            rdoeq2(5).coef(:,2)=[  stderBEBal(1,:);  %y1
                                   NaN(1,1);         %y2: empty
                                   stderBEBal(2,:);  %X1
                                   NaN(2,1);         %X2,X3: empty
                                   stderBEBal(3,:);  %X4
                                   stderBEBal(4,:);  %X5
                                   stderBEBal(5,:);  %Z1
                                   NaN(1,1);         %Z2: empty
                                   stderBEBal(6,:);  %Z3
                                   NaN(1,1);         %phi1
                                   stderBEBal(7,:);  %phi2
                                   NaN(2,1)];        %mu1,mu2: empty
            clear('resiBEBal','sigmaBEBal','stderBEBal');
```

```
%==========================================================================
%           Method 9: ECCorn
% ==========================================================================
            %ewave=Qmu*y-Qmu*Rwave*inv(Rwave'*Qmu*PA*Qmu*Rwave)*Rwave'*Qmu*PA*Qmu*y;
            ewave=rdoeq2(3).resi; %taken from FEBal
            sigma1=((ewave'*Qmu*ewave)/(N*(T-1)-Kwave))^0.5;
            %ebar=Pmu*y-Pmu*R*inv(R'*Pmu*PB*Pmu*R)*R'*Pmu*PB*Pmu*y;
            ebar=rdoeq2(5).resi; %taken from BEBal
            sigma2=((ebar'*Pmu*ebar)/(N-K-1))^0.5;
            WCorn=(1/sigma1)*Qmu+(1/sigma2)*Pmu;
            resiECCorn=rdoeq2(9).resi;
            sigmaECCorn=(resiECCorn'*resiECCorn)/(N*T-K); %"within" component of↙
variance
            stderECCorn=diag(sigmaECCorn*inv(R'*WCorn*(PA+PB)*WCorn*R)).^0.5; %new std.↙
errors
            rdoeq2(9).coef(:,2)=[stderECCorn(1,:);  %y1
                                   NaN(1,1);         %y2: empty
                                   stderECCorn(2,:);    %X1
                                   NaN(2,1);         %X2,X3: empty
                                   stderECCorn(3,:);    %X4
                                   stderECCorn(4,:);    %X5
                                   stderECCorn(5,:);    %Z1
                                   NaN(1,1);         %Z2: empty
                                   stderECCorn(6,:);    %Z3
                                   NaN(1,1);         %phi1
                                   stderECCorn(7,:);    %phi2
                                   NaN(2,1)];        %mu1,mu2: empty
            clear('ewave','ebar','sigma2','WCorn','resiECCorn',...
                'sigmaECCorn','stderECCorn');

% ==========================================================================
%           Method 10: HTM
% ==========================================================================
            ebarHT=Pmu*y-Pmu*R*inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            sigma2HT=((ebarHT'*Pmu*ebarHT)/(N-K-1))^0.5;
            WHT=(1/sigma1)*Qmu+(1/sigma2HT)*Pmu;
            resiHT=rdoeq2(10).resi; %residuals
            sigmaHT=(resiHT'*resiHT)/(N*T-K); %"within" component of variance
            stderHT=diag(sigmaHT*inv(R'*WHT*(PA+PBHT)*WHT*R)).^0.5; %new std. errors
            rdoeq2(10).coef(:,2)=[ stderHT(1,:); %y1
                                   NaN(1,1);         %y2: empty
                                   stderHT(2,:);    %X1
                                   NaN(2,1);         %X2,X3: empty
                                   stderHT(3,:);    %X4
                                   stderHT(4,:);    %X5
                                   stderHT(5,:);    %Z1
                                   NaN(1,1);         %Z2: empty
                                   stderHT(6,:);    %Z3
                                   NaN(1,1);         %phi1
                                   stderHT(7,:);    %phi2
                                   NaN(2,1)];        %mu1,mu2: empty
            clear('ebarHT','sigma1','sigma2HT','WHT','resiHT','sigmaHT','stderHT');
```

```
%========================================================================
%            Method 15: G3SPD_Std 3rd Step
% ========================================================================
            resiStd2nd=rdoeq2(14).resi;
            sigmaStd2nd=(resiStd2nd'*resiStd2nd)/(N-Kbar);
            resiStd3rd=rdoeq2(15).resi;
            Rfull=single([Y,X1,X4,X5,Z1,Z3,resiStd2nd,ones(N*T,1)]); %full dataset, ↙
including all the original vars. plus the estimated Fixed Effects
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaStd3rd=(resiStd3rd'*resiStd3rd)/(N*T-K);
            OmegaGMM=sigmaStd3rd*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=diag(resiStd3rd.^2);
            OmegaHC1=inv(Rfull'*Rfull)*(Rfull'*sigmaHC1*Rfull)*inv(Rfull'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=diag((resiStd3rd.^2)./(1-diag(PR)));
            OmegaHC2=inv(Rfull'*Rfull)*(Rfull'*sigmaHC2*Rfull)*inv(Rfull'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiStd3rd./(1-diag(PR)); %epsilon*
            sigmaLE=diag((epsilonast.^2));
            OmegaLE=inv(Rfull'*Rfull)*(Rfull'*sigmaLE*Rfull)*inv(Rfull'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(epsilonast*epsilonast');
            OmegaHC3=inv(Rfull'*Rfull)*(Rfull'*sigmaHC3*Rfull)*inv(Rfull'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiStd3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=(kron((E'*E)./(T-K),eye(T))); %Beck & Katz (1995) covariance matrix↙
for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
```

```
            Psi2=(sigmaStd2nd/(sigmaStd3rd^2))* Rfull'*...
                Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull;
            Psi3=(1/(sigmaStd3rd^2))*(Rfull'*resiStd3rd)*...
                (resiStd2nd'*Pmu*Rbar*inv(Rbar'*Pmu*Rbar)*Rbar'*Pmu*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaStd3rd','sigmaStd2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiStd2nd');

            % store coefficients and standard errors (8 columns)
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(15).coef=rdoeq2(15).coef(:,1);
            rdoeq2(15).coef(:,2:8)=[stder(1,:); %y1
                                   NaN(1,7);         %y2: empty
                                   stder(2,:); %X1
                                   NaN(2,7);         %X2,X3: empty
                                   stder(3,:); %X4
                                   stder(4,:); %X5
                                   stder(5,:); %Z1
                                   NaN(1,7);         %Z2: empty
                                   stder(6,:); %Z3
                                   NaN(1,7);         %phi1: empty
                                   stder(8,:); %phi2
                                   NaN(1,7);         %mu1: empty
                                   stder(7,:)];%mu2
            clear('resiStd3rd','stderGMM','stderHC1','stderHC2','PR',...
                'stderLE','stderHC3','stderBK','stderMT','Rfull','stder');

%==========================================================================
%           Method 16,17,18: G3SPD_Corn
%==========================================================================
        % 1st stage
            resiCorn1st=rdoeq2(16).resi;
            sigmaCorn1st=(resiCorn1st'*Qmu*resiCorn1st)/(N*(T-1)-Kwave);
            stderCorn1st=diag(sigmaCorn1st*inv(Rwave'*Qmu*PA*Qmu*Rwave)).^0.5;
            rdoeq2(16).coef(:,2)=[stderCorn1st(1,:);  %y1
                                   NaN(1,1);         %y2: empty
                                   stderCorn1st(2,:);    %X1
                                   NaN(2,1);         %X2,X3:empty
                                   stderCorn1st(3,:);    %X4
                                   stderCorn1st(4,:);    %X5
                                   NaN(7,1)];        %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            clear('resiCorn1st','sigmaCorn1st','stderCorn1st');

        % 2nd stage
            resiCorn2nd=rdoeq2(17).resi;
            sigmaCorn2nd=(resiCorn2nd'*Pmu*resiCorn2nd)/(N-Kbar);
            stderCorn2nd=diag(sigmaCorn2nd*inv(Rbar'*Pmu*PB*Pmu*Rbar)).^0.5;
            rdoeq2(17).coef(:,2)=[  NaN(7,1);         %y1,y2,X1,X2,X3,X4,X5: empty
                                    stderCorn2nd(1,:);    %Z1
                                    NaN(1,1);            %Z2: empty
                                    stderCorn2nd(2,:);    %Z3
```

```
                                NaN(1,1);               %phi1: empty
                                stderCorn2nd(3,:);       %phi2
                                NaN(2,1)];              %mu1,mu2: empty
            clear('stderCorn2nd');

        % 3rd stage
            resiCorn3rd=rdoeq2(18).resi;
            Rfull=[Y,X1,X4,X5,Z1,Z3,resiCorn2nd,ones(N*T,1)]; %full dataset, including
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resiCorn2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaCorn3rd=(resiCorn3rd'*resiCorn3rd)/(N*T-K);
            OmegaGMM=sigmaCorn3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiCorn3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiCorn3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiCorn3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv
(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiCorn3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance
```

```
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaCorn2nd/(sigmaCorn3rd^2))* Rfull'*PC*PB*Pmu*Rbar*...
                inv(Rbar'*Pmu*PB*Pmu*Rbar)*Rbar'*Pmu*PB*PC*Rfull;
            Psi3=(1/(sigmaCorn3rd^2))*(Rfull'*PC*resiCorn3rd)*...
                (resiCorn2nd'*PB*Pmu*Rbar*inv(Rbar'*Pmu*PB*Pmu*Rbar)↙
*Rbar'*Pmu*PB*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaCorn3rd','sigmaCorn2nd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiCorn2nd');

            % store standard errors
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(18).coef=rdoeq2(18).coef(:,1);
            rdoeq2(18).coef(:,2:8)=[stder(1,:); %y1
                                    NaN(1,7);         %y2: empty
                                    stder(2,:); %X1
                                    NaN(2,7);         %X2,X3: empty
                                    stder(3,:); %X4
                                    stder(4,:); %X5
                                    stder(5,:); %Z1
                                    NaN(1,7);         %Z2: empty
                                    stder(6,:); %Z3
                                    NaN(1,7);         %phi1: empty
                                    stder(8,:); %phi2
                                    NaN(1,7);         %mu1: empty
                                    stder(7,:)];%mu2
            clear('resiCorn3rd','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull','stder');

% %=====================================================================
%             %Method 19,20: G3SPD_HTM
% %=====================================================================
        % 2nd stage
            resiHTM2nd=rdoeq2(19).resi;
            sigmaHTM2nd=(resiHTM2nd'*Pmu*resiHTM2nd)/(N-Kbar);
            stderHTM2nd=diag(sigmaHTM2nd*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)).^0.5;
            rdoeq2(19).coef(:,2)=[  NaN(7,1);           %y1,y2,X1,X2,X3,X4,X5: empty
                                    stderHTM2nd(1,:);   %Z1
                                    NaN(1,1);           %Z2: empty
                                    stderHTM2nd(2,:);   %Z3
                                    NaN(1,1);           %phi1: empty
                                    stderHTM2nd(3,:);   %phi2
                                    NaN(2,1)];          %mu1,mu2: empty
```

```
            clear('stderHTM2nd');

        % 3rd Step:
            resiHTM3rd=rdoeq2(20).resi;
            Rfull=[Y,X1,X4,X5,Z1,Z3,resiHTM2nd,ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resiHTM2nd,ones(N*T,1)]; %set of instruments
            PC=C*inv(C'*C)*C';
            RCinvCC=Rfull'*C*inv(C'*C);
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            K=size(Rfull,2);

            % GMM covar.: stored in column 2
            sigmaHTM3rd=(resiHTM3rd'*resiHTM3rd)/(N*T-K);
            OmegaGMM=sigmaHTM3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTM3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTM3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTM3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTM3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
```

```
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTM3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTM3rd^2))*(Rfull'*PC*resiHTM3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)
*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTM3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT');

            % store standard errors
            stder=[stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,stderMT];
            rdoeq2(20).coef=rdoeq2(20).coef(:,1);
            rdoeq2(20).coef(:,2:8)=[stder(1,:); %y1
                                    NaN(1,7);         %y2: empty
                                    stder(2,:); %X1
                                    NaN(2,7);         %X2,X3: empty
                                    stder(3,:); %X4
                                    stder(4,:); %X5
                                    stder(5,:); %Z1
                                    NaN(1,7);         %Z2: empty
                                    stder(6,:); %Z3
                                    NaN(1,7);         %phi1: empty
                                    stder(8,:); %phi2
                                    NaN(1,7);         %mu1: empty
                                    stder(7,:)];%mu2
            clear('resiHTM3rd','stder','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','Rfull');

%===========================================================================
%           Method 25: G3SPD_HTMNew (New version using the HT set of
%                                   instruments in the 3rd Step estimation)
%===========================================================================
        % 1st Step:
            % idem G3SPD_HTM

        % 2nd Step:
            % idem G3SPD_HTM

        % 3rd Step
            Rfull=single([Y,X1,X4,X5,Z1,Z3,resiHTM2nd,ones(N*T,1)]); %full dataset,
including all the original vars. plus the estimated Fixed Effects
            C=[A,BHT,resiHTM2nd]; %set of instruments
            PC=C*inv(C'*C)*C';
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors
            RCinvCC=Rfull'*C*inv(C'*C);
```

```
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaHTMNEW=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resiHTMNEW3rd=y-Rfull*thetaHTMNEW;

            % GMM covar.: stored in column 2
            sigmaHTMNEW3rd=(resiHTMNEW3rd'*resiHTMNEW3rd)/(N*T-K);
            OmegaGMM=sigmaHTMNEW3rd*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;

            % WH-HC1 covar.: stored in column 3
            sigmaHC1=C'*diag(resiHTMNEW3rd.^2)*C;
            OmegaHC1=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC1*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % WH-HC2 covar.: stored in column 4
            sigmaHC2=C'*diag((resiHTMNEW3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC2*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE covar.: stored in column 5
            epsilonast=resiHTMNEW3rd./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(epsilonast.^2)*C;
            OmegaLE=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaLE*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3 covar.: stored in column 6
            sigmaHC3=sigmaLE-(C'*epsilonast*epsilonast'*C);
            OmegaHC3=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaHC3*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderHC3=diag(OmegaHC3).^0.5;
            clear('epsilonast','sigmaLE','sigmaHC3','OmegaHC3');

            % BK covar.: stored in column 7
            E=reshape(resiHTMNEW3rd,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % MT covar.: stored in column 8
            Psi1=OmegaGMM;
            Psi2=(sigmaHTM2nd/(sigmaHTMNEW3rd^2))* Rfull'*PC*PBHT*Pmu*Rbar*...
```

```
                inv(Rbar'*Pmu*PBHT*Pmu*Rbar)*Rbar'*Pmu*PBHT*PC*Rfull;
            Psi3=(1/(sigmaHTMNEW3rd^2))*(Rfull'*PC*resiHTMNEW3rd)*...
                (resiHTM2nd'*PBHT*Pmu*Rbar*inv(Rbar'*Pmu*PBHT*Pmu*Rbar)↙
*Rbar'*Pmu*PBHT*PC*Rfull);
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('sigmaHTMNEW3rd','OmegaGMM','Psi1','Psi2',...
                'Psi3','OmegaMT','resiHTM2nd','sigmaHTM2nd');

            % store coefficients and standard errors (8 columns)
            stder=[thetaHTMNEW,stderGMM,stderHC1,stderHC2,stderLE,stderHC3,stderBK,↙
stderMT];
            rdoeq2(25).coef=[stder(1,:); %y1
                             NaN(1,8);         %y2: empty
                             stder(2,:); %X1
                             NaN(2,8);         %X2,X3: empty
                             stder(3,:); %X4
                             stder(4,:); %X5
                             stder(5,:); %Z1
                             NaN(1,8);         %Z2: empty
                             stder(6,:); %Z3
                             NaN(1,8);         %phi1: empty
                             stder(8,:); %phi2
                             NaN(1,8);         %mu1: empty
                             stder(7,:)];%mu2
            rdoeq2(25).resi=resiHTMNEW3rd;
            clear('resiHTMNEW3rd','thetaHTMNEW','stderGMM','stderHC1','stderHC2',...
                'stderLE','stderHC3','stderBK','stderMT','PR','PC','Rfull',...
                'C','RCinvCC','estimadores','Rfull');

            % Save results
            if exist(['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')],'dir')~=7
                mkdir(['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')]);
            end
            filename=['C:\Diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');

        end
        toc
    end
end
```

```
%========================================================================
% Function ESTIMAEQ2_ALTER_V1
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%============================================================================

function estimaeq2_alter_v1
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y1, X1, X4, X5, Z1, Z3, VarPhi2 (constant), Phi2 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            Rwave=[Y,X1,X4,X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);
```

```
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq2');

%==========================================================================
%           1st Step: FEBal
%==========================================================================
            redo(1).coef=rdoeq2(3).coef;
            redo(1).resi=rdoeq2(3).resi;
            thetaFE=redo(1).coef(2:5,1);
%==========================================================================
%           2nd Step: BEBal
%==========================================================================
            redo(2).coef=rdoeq2(5).coef;
            redo(2).resi=rdoeq2(5).resi;
            thetaBE=[redo(2).coef(8:9,1);
            redo(2).coef(11,1)];

%==========================================================================
%           3rd Step: MuHat
%==========================================================================
            uhat=Pmu*y-Pmu*Rwave*thetaFE;
            MuHat=uhat-Rbar*thetaBE;
            redo(3).resi=MuHat;

%==========================================================================
%           4th Step: Pooled OLS
%==========================================================================
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaCorn=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi=y-Rfull*thetaCorn;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of OLS residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
```

```
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            redo(4).coef=[thetaCorn,stderGMM,stderBK];
            redo(4).resi=resi;
            clear('resi','thetaCorn','stderGMM','stderBK','Rfull',...
                'C','RCinvCC');

%=====================================================================
%           Save results
%=====================================================================
            if exist(['.\eq2_alter\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')],'dir' )~=7
                mkdir(['.\eq2_alter\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')]);
            end
            filename=['.\eq2_alter\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            save(filename,'redo');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2','redo');

        end
        toc
    end
end
```

```
%=======================================================================
% Function ESTIMAEQ2_ALTER_V2
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%===========================================================================

function estimaeq2_alter_v2
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 2:
% y1, X1, X4, X5, Z1, Z3, VarPhi2 (constant), Phi2 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['c:\diego\SimNewFE\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f )];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['c:\diego\SimNewFE\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f )];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        BHT=single(Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]); %between instruments
        PBHT=BHT*inv(BHT'*BHT)*BHT';

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
```

```
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            R=[Y,X1,X4,X5,Z1,Z3,ones(N*T,1)]; %Eq. 1, all regressors: Y, x3, z3 are↙
endogenous
            K=size(R,2);

            filename=['c:\diego\SimNewFE\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');


%==========================================================================
%           Move Methods (6 to 25) to (7 to 26)
%==========================================================================
            for met=25:(-1):6
                if size(rdoeq2(met).coef,2)>0
                    rdoeq2(met+1).coef=[rdoeq2(met).coef(1,:);    %y1
                                        rdoeq2(met).coef(3,:);    %X1
                                        rdoeq2(met).coef(6:8,:);  %X4,X5,Z1
                                        rdoeq2(met).coef(10,:);   %Z3
                                        rdoeq2(met).coef(12,:);   %Varphi2
                                        rdoeq2(met).coef(14,:)];  %Phi2
                    rdoeq2(met+1).resi=rdoeq2(met).resi;
                    rdoeq2(met+1).stats=rdoeq2(met).stats;
                else
                    rdoeq2(met+1).coef=[];
                    rdoeq2(met+1).resi=[];
                    rdoeq2(met+1).stats=[];
                end
            end


%==========================================================================
%           Method 6: BEHTM
%==========================================================================
            thetaBEHTM=inv(R'*Pmu*PBHT*Pmu*R)*R'*Pmu*PBHT*Pmu*y;
            resiBEHTM=Pmu*y-Pmu*R*thetaBEHTM;
            sigmaBEHTM=(resiBEHTM'*Pmu*resiBEHTM)/(N-K);
            stderBEHTM=diag(sigmaBEHTM*inv(R'*Pmu*PBHT*Pmu*R)).^0.5; %new std. errors
            estimadores=[thetaBEHTM,stderBEHTM];
            rdoeq2(6).coef=[estimadores; NaN(1,2)]; %Phi1 is empty
            rdoeq2(6).resi=resiBEHTM;
            clear('thetaBEHTM','resiBEHTM','stderBEHTM','estimadores');

%==========================================================================
%           Arrange Methods (1 to 5)
%==========================================================================
            for met=1:1:5
                if size(rdoeq2(met).coef,2)>0
                    rdoeq2(met).coef=[  rdoeq2(met).coef(1,:);    %y1
                                        rdoeq2(met).coef(3,:);    %X1
```

```
                                        rdoeq2(met).coef(6:8,:);  %X4,X5,Z1
                                        rdoeq2(met).coef(10,:);   %Z3
                                        rdoeq2(met).coef(12,:);   %Varphi2
                                        rdoeq2(met).coef(14,:)];  %Phi2
                end
            end


%======================================================================
%          Save results
%======================================================================
            if exist(['c:\diego\SimLastFE\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,    '%3.0f')],'dir')~=7
                mkdir(['c:\diego\SimLastFE\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,    '%3.0f')]);
            end
            filename=['c:\diego\SimLastFE\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,    '%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2');

        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMAEQ2_ALTER_V3
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=============================================================================

function estimaeq2_alter_v3
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio

%Order of coefficients in Equation 1:
% y2, X1, X2, X3, Z1, Z2, VarPhi1 (constant), Phi1 (Mui)

for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %predetermined matrixes
        %----------------------
        Rbar=single([Z1,Z3,ones(N*T,1)]);

        for obs=1:1:1000 %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            Rwave=[Y,X1,X4,X5]; %Eq. 1, all regressors: Y, x3, z3 are endogenous
            Kwave=size(Rwave,2);
```

```
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f')];
            load(filename,'rdoeq2');


%==========================================================================
%           Method 27,28: G3SPD_Std_alter 3rd Step
%==========================================================================
            %generates pseudo-effects (stored in 27)
            thetaFEStd=rdoeq2(2).coef(1:4,1);
            thetaBEStd=rdoeq2(4).coef(5:7,1);
            MuHat=Pmu*y-Pmu*Rwave*thetaFEStd-Rbar*thetaBEStd;
            rdoeq2(27).resi=MuHat;

            %estimation (stored in 28)
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaStd=inv(Rfull'*Rfull)*Rfull'*y;
            resi=y-Rfull*thetaStd;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(Rfull'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=kron((E'*E)./(T-K),eye(T));%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(Rfull'*Rfull)*(Rfull'*sigmaBK*Rfull)*inv(Rfull'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            rdoeq2(28).coef=[thetaStd,stderGMM,stderBK];
            rdoeq2(28).resi=resi;
            clear('resi','thetaFEStd','thetaBEStd','MuHat','Rfull',...
                'thetaStd','stderGMM','stderBK');

%==========================================================================
%           Method 29,30: G3SPD_Corn_alter 3rd Step
%==========================================================================
            %generates pseudo-effects (stored in 29)
            thetaFEBal=rdoeq2(3).coef(1:4,1);
            thetaBEBal=rdoeq2(5).coef(5:7,1);
```

```
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEBal;
            rdoeq2(29).resi=MuHat;

            %estimation (stored in 32)
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);

            % Coefficients: stored in column 1
            thetaCorn=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi=y-Rfull*thetaCorn;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            rdoeq2(30).coef=[thetaCorn,stderGMM,NaN(8,4),stderBK,NaN(8,1)];
            rdoeq2(30).resi=resi;
            clear('resi','thetaCorn','stderGMM','stderBK','C','RCinvCC',...
                'thetaFEBal','thetaBECorn');


%==========================================================================
%           Method 31,32: G3SPD_HTM_alter 3rd Step
%==========================================================================
            %generates pseudo-effects (stored in 31)
            thetaFEBal=rdoeq2(3).coef(1:4,1);
            thetaBEHTM=rdoeq2(6).coef(5:7,1);
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEHTM;
            rdoeq2(31).resi=MuHat;

            %estimation (stored in 32)
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat]); %full dataset, including all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            RCinvCC=Rfull'*C*inv(C'*C);
            K=size(Rfull,2);
```

```
            % Coefficients: stored in column 1
            thetaHTM=inv(RCinvCC*C'*Rfull)*RCinvCC*C'*y;
            resi=y-Rfull*thetaHTM;

            % GMM covar.: stored in column 2
            sigmaGMM=(resi'*resi)/(N*T-K);
            OmegaGMM=sigmaGMM*inv(RCinvCC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5;
            clear('sigmaGMM','OmegaGMM');

            % BK covar.: stored in column 3
            E=reshape(resi,T,N); %arrange the set of residuals
            sigmaBK=C'*kron((E'*E)./(T-K),eye(T))*C;%Beck & Katz (1995) covariance↙
matrix for each time period.
            OmegaBK=inv(RCinvCC*C'*Rfull)*(RCinvCC*sigmaBK*RCinvCC')*inv↙
(RCinvCC*C'*Rfull);
            stderBK=diag(OmegaBK).^0.5;
            clear('E','sigmaBK','OmegaBK');

            % store coefficients and standard errors (3 columns)
            rdoeq2(32).coef=[thetaHTM,stderGMM,NaN(8,4),stderBK,NaN(8,1)];
            rdoeq2(32).resi=resi;
            clear('resi','thetaHTM','stderGMM','stderBK','C','RCinvCC',...
                'thetaFEBal','thetaBEHTM');


%======================================================================
%          Save results
%======================================================================
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2','redo');

        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMAEQ2_ALTER_V4
% This program perform complementary estimations for EQUATION 1
% Author: Diego Avanzini
%=============================================================================

function estimaeq2_alter_v4
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio


%Order of coefficients in Equation 1:
% y1, X1, X4, X5, Z1, Z3, VarPhi2 (constant), Phi2 (Mui)


for nt=1:1:size(ntcases,1) %seven combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    Zmu=single(kron(eye(N),ones(T,1)));
    Pmu=single(Zmu*inv(Zmu'*Zmu)*Zmu');
    Qmu=single(eye(N*T)-Pmu);
    clear('Zmu');

    for rho=1:1:size(rhocases,1)  % eleven rho (sigmamu/sigmaepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        % open files containing variables and convert to single
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','datos');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);
        Rbar=single([Z1,Z3,ones(N*T,1)]);
        Kbar=size(Rbar,2);

        %predetermined matrixes
        %----------------------
        iniciobs=1;
%         if nt==1 && rho==2;iniciobs=187;end

        for obs=iniciobs:1:1000 %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                  Observation: ' num2str(obs)]);
            y=single(datos(1,obs).Y(:,2)); %dependent var.
            Y=single(datos(1,obs).Y(:,1)); %endogenous var.
            Rwave=single([Y,X1,X4,X5]);
            Kwave=size(Rwave,2);

            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');

%==========================================================================
%           Method 30: G3SPD_Corn_alter 3rd Step
%==========================================================================
            MuHat=rdoeq2(29).resi;
            Rfull=single([Y,X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat]); %full dataset, including↙
all the original vars. plus the estimated Fixed Effects
            C=[X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1),MuHat]; %set of instruments
            PC=C*inv(C'*C)*C';
            K=size(Rfull,2);

            % obtain residuals and V
            resi1=squeeze(rdoeq2(30).resi);
            resi2=squeeze(rdoeq2(3).resi);
%             resi3=Pmu*y-Rbar*squeeze(rdoeq2(5).coef(5:7,1));
            resi3=(Pmu*Rwave*squeeze(rdoeq2(3).coef(1:4,1))+MuHat)/T;
            resi=[resi1,resi2,resi3]; %residuals (OLS 3rd step, Within, ~between)


            dof1=N*T-K;
            dof2=N*(T-1)-Kwave;
            dof3=(N*T-(K-1));
            dof=[dof1,    dof1,    dof1;
                 dof1,    dof2,    dof2;
                 dof1,    dof2,    dof3];

            V=(resi'*resi)./dof;
            clear('resi1','resi2','resi3','resi','dof');

            %obtain S
            QRwavebar=eye(N*T)-Pmu*Rwave*inv(Rwave'*Pmu*Rwave)*Rwave'*Pmu;
            S1=eye(N*T);
            S2=Pmu*Rwave*inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu;
            S3=Rbar*inv(Rbar'*QRwavebar*Rbar)*Rbar'*QRwavebar;
            S=[S1 S2 S3];
            clear('S1','S2','S3','QRwavebar');

            %obtain Sigma
            sigmaAdjOLS=S*kron(V,eye(N*T))*S';
            OmegaAdjOLS=inv(Rfull'*PC*Rfull)*(Rfull'*PC*sigmaAdjOLS*PC*Rfull)*inv↙
(Rfull'*PC*Rfull);
```

```
            clear('V','S','sigmaAdjOLS');
            stdernew=diag(OmegaAdjOLS).^0.5; %new std. errors
            rdoeq2(30).coef(:,9)=stdernew; %store new std. errors in column 9

%=======================================================================
%          Save results
%=======================================================================
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            save(filename,'rdoeq2');
            disp([filename ' guardado. Continuar...']);
            clear('rdoeq2','redo');

        end
        toc
    end
end
```

```
%==================================================================
% Function ESTIMASYS_V1
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%======================================================================

function estimasys_v1
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic

        %load variables and estimated covariance matrixes
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        if exist(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')],'dir' )~=7
            mkdir(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end

        % predetermined matrixes
        % ----------------------
        % vars. for the 2nd. stage of G3SPD
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        % instruments for pooled regression
        A=single(kron(eye(G),([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]))); %all instruments
```

```
        % instruments for the within regression
        Awave=single(kron(eye(G),Qmu*([X1,X2,X3,X4,X5]))); %within instruments
        % instruments for the between regression
        Bbar=single(kron(eye(G),Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]))); %between↙
instruments
        % instruments for the HT between regression
        BHT1=single(Pmu*[X1,X2,X4,X5,Z1,Z3,ones(N*T,1)]); %eq. 1: HT between instruments
        if rank(BHT1)<size(BHT1,2);BHT1=BHT1(:,1:rank(BHT1));end
        BHT2=single(Pmu*[X1,X4,X2,X3,Z1,Z2,ones(N*T,1)]); %eq. 2: HT between instruments
        if rank(BHT2)<size(BHT2,2);BHT2=BHT2(:,1:rank(BHT2));end
        BHT=single(blkdiag(BHT1,BHT2)); %system: HT 'between' instruments
        % instruments for the AM between regression
        BAM1=single(Pmu*[SA([X1,X2],N,T),SA([X4,X5],N,T),Z1,Z3,ones(N*T,1)]); %eq. 1: AM↙
between instruments
        if rank(BAM1)<size(BAM1,2);BAM1=BAM1(:,1:rank(BAM1));end
        BAM2=single(Pmu*[SA([X1,X4],N,T),SA([X2,X3],N,T),Z1,Z2,ones(N*T,1)]); %eq. 2: AM↙
between instruments
        if rank(BAM2)<size(BAM2,2);BAM2=BAM2(:,1:rank(BAM2));end
        BAM=single(blkdiag(BAM1,BAM2)); %system: AM 'between' instruments
         instruments for the BMS between regression
        BBMS1=single(Pmu*[SA([X1,X2],N,T),SA([X4,X5],N,T),Z1,Z3,SA(Qmu*X3,N,T),ones(N*T,↙
1)])  %eq. 1: BMS between instruments
        if rank(BBMS1)<size(BBMS1,2);BBMS1=BBMS1(:,1:rank(BBMS1));end
        BBMS2=single(Pmu*[SA([X1,X4],N,T),SA([X2,X3],N,T),Z1,Z2,SA(Qmu*X5,N,T),ones(N*T,↙
1)])  %eq. 2: BMS between instruments
        if rank(BBMS2)<size(BBMS2,2);BBMS2=BBMS2(:,1:rank(BBMS2));end
        BBMS=single(blkdiag(BBMS1,BBMS2)); %system: BMS 'between' instruments

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            filename=['.\varianza\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')↙
...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f )];
            load(filename,'bigsigma');
            y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
            R1=single([datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1)]); %eq. 1 all↙
regressors
            R2=single([datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1)]); %eq. 2 all↙
regressors
            R=single(blkdiag(R1,R2)); %system all regressors
            K=single(size(R,2));
            RbarBE=single(blkdiag(Pmu*R1,Pmu*R2)); %to estimate BE
            R1wave=single([datos(1,obs).Y(:,2) X1 X2 X3]); %Eq. 1 within regressors
            R2wave=single([datos(1,obs).Y(:,1) X1 X4 X5]); %Eq. 2 within regressors
            Rwave=single(blkdiag(R1wave,R2wave)); %system within regressors
            Kwave=single(size(Rwave,2));
            clear rdo

%===========================================================================
            %Method 1: OLS
%===========================================================================
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(1).vari,eye
(N*T)),A,y,G,N,T);
            rdosys(1).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(2,2);         %X4,X5: empty
                            rdo.coef(5,:);    %Z1
                            rdo.coef(6,:);    %Z2
                            NaN(1,2);         %Z3: empty
                            rdo.coef(7,:);    %phi1
                            NaN(3,2);         %phi2,mu1,mu2: empty
                            rdo.coef(8,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(9,:);    %X1
                            NaN(2,2);         %X2,X3: empty
                            rdo.coef(10,:);   %X4
                            rdo.coef(11,:);   %X5
                            rdo.coef(12,:);   %Z1
                            NaN(1,2);         %Z2: empty
                            rdo.coef(13,:);   %Z3
                            NaN(1,2);         %phi1
                            rdo.coef(14,:);   %phi2
                            NaN(2,2)];        %mu1,mu2: empty
            rdosys(1).resi=rdo.resi;
            rdosys(1).stats=rdo.stats;

%===========================================================================
            %Method 2: FEStd
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(Rwave,Qmu,kron(bigsigma(2).vari,Qmu),
Awave,y,G,N,T);
            rdosys(2).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(9,2);         %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
                            rdo.coef(5,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(6,:);    %X1
                            NaN(2,2);         %X2,X3:empty
                            rdo.coef(7,:);    %X4
                            rdo.coef(8,:);    %X5
                            NaN(7,2)];        %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdosys(2).resi=rdo.resi;
            rdosys(2).stats=rdo.stats;

%===========================================================================
            %Method 3: FEBal
%===========================================================================
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(Rwave,Qmu,kron(bigsigma(3).vari,Qmu),↙
Awave,y,G,N,T);
            rdosys(3).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(9,2);         %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
                            rdo.coef(5,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(6,:);    %X1
                            NaN(2,2);         %X2,X3:empty
                            rdo.coef(7,:);    %X4
                            rdo.coef(8,:);    %X5
                            NaN(7,2)];        %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdosys(3).resi=rdo.resi;
            rdosys(3).stats=rdo.stats;

%===========================================================================
            %Method 4: BEStd
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(RbarBE,Pmu,kron(bigsigma(4).vari,Pmu),↙
Bbar,y,G,N,T);
            rdosys(4).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(2,2);         %X4,X5: empty
                            rdo.coef(5,:);    %Z1
                            rdo.coef(6,:);    %Z2
                            NaN(1,2);         %Z3: empty
                            rdo.coef(7,:);    %phi1
                            NaN(3,2);         %phi2,mu1,mu2: empty
                            rdo.coef(8,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(9,:);    %X1
                            NaN(2,2);         %X2,X3: empty
                            rdo.coef(10,:);   %X4
                            rdo.coef(11,:);   %X5
                            rdo.coef(12,:);   %Z1
                            NaN(1,2);         %Z2: empty
                            rdo.coef(13,:);   %Z3
                            NaN(1,2);         %phi1
                            rdo.coef(14,:);   %phi2
                            NaN(2,2)];        %mu1,mu2: empty
            rdosys(4).resi=rdo.resi;
            rdosys(4).stats=rdo.stats;

%===========================================================================
            %Method 5: BEBal
%===========================================================================
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(RbarBE,Pmu,kron(bigsigma(5).vari,Pmu),↙
Bbar,y,G,N,T);
            rdosys(5).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(2,2);         %X4,X5: empty
                            rdo.coef(5,:);    %Z1
                            rdo.coef(6,:);    %Z2
                            NaN(1,2);         %Z3: empty
                            rdo.coef(7,:);    %phi1
                            NaN(3,2);         %phi2,mu1,mu2: empty
                            rdo.coef(8,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(9,:);    %X1
                            NaN(2,2);         %X2,X3: empty
                            rdo.coef(10,:);   %X4
                            rdo.coef(11,:);   %X5
                            rdo.coef(12,:);   %Z1
                            NaN(1,2);         %Z2: empty
                            rdo.coef(13,:);   %Z3
                            NaN(1,2);         %phi1
                            rdo.coef(14,:);   %phi2
                            NaN(2,2)];        %mu1,mu2: empty
            rdosys(5).resi=rdo.resi;
            rdosys(5).stats=rdo.stats;

%===========================================================================
            %Method 6: ECWH
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(6).vari,Qmu)↙
+kron(bigsigma(7).vari,Pmu),A,y,G,N,T);
            rdosys(6).coef=[NaN(1,2);         %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(2,2);         %X4,X5: empty
                            rdo.coef(5,:);    %Z1
                            rdo.coef(6,:);    %Z2
                            NaN(1,2);         %Z3: empty
                            rdo.coef(7,:);    %phi1
                            NaN(3,2);         %phi2,mu1,mu2: empty
                            rdo.coef(8,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(9,:);    %X1
                            NaN(2,2);         %X2,X3: empty
                            rdo.coef(10,:);   %X4
                            rdo.coef(11,:);   %X5
                            rdo.coef(12,:);   %Z1
                            NaN(1,2);         %Z2: empty
```

```
                                rdo.coef(13,:);   %Z3
                                NaN(1,2);         %phi1
                                rdo.coef(14,:);   %phi2
                                NaN(2,2)];        %mu1,mu2: empty
            rdosys(6).resi=rdo.resi;
            rdosys(6).stats=rdo.stats;

%==========================================================================
            %Method 7: ECAm
%==========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(8).vari,Qmu)↙
+kron(bigsigma(9).vari,Pmu),A,y,G,N,T);
            rdosys(7).coef=[NaN(1,2);         %y1: empty    EQ. 1
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
                                rdo.coef(4,:);    %X3
                                NaN(2,2);         %X4,X5: empty
                                rdo.coef(5,:);    %Z1
                                rdo.coef(6,:);    %Z2
                                NaN(1,2);         %Z3: empty
                                rdo.coef(7,:);    %phi1
                                NaN(3,2);         %phi2,mu1,mu2: empty
                                rdo.coef(8,:);    %y1           EQ. 2
                                NaN(1,2);         %y2: empty
                                rdo.coef(9,:);    %X1
                                NaN(2,2);         %X2,X3: empty
                                rdo.coef(10,:);   %X4
                                rdo.coef(11,:);   %X5
                                rdo.coef(12,:);   %Z1
                                NaN(1,2);         %Z2: empty
                                rdo.coef(13,:);   %Z3
                                NaN(1,2);         %phi1
                                rdo.coef(14,:);   %phi2
                                NaN(2,2)];        %mu1,mu2: empty
            rdosys(7).resi=rdo.resi;
            rdosys(7).stats=rdo.stats;

%==========================================================================
            %Method 8: ECSA
%==========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(10).vari,Qmu)↙
+kron(bigsigma(11).vari,Pmu),A,y,G,N,T);
            rdosys(8).coef=[NaN(1,2);         %y1: empty    EQ. 1
                                rdo.coef(1,:);    %y2
                                rdo.coef(2,:);    %X1
                                rdo.coef(3,:);    %X2
                                rdo.coef(4,:);    %X3
                                NaN(2,2);         %X4,X5: empty
                                rdo.coef(5,:);    %Z1
                                rdo.coef(6,:);    %Z2
                                NaN(1,2);         %Z3: empty
```

```
                                rdo.coef(7,:);     %phi1
                                NaN(3,2);          %phi2,mu1,mu2: empty
                                rdo.coef(8,:);     %y1           EQ. 2
                                NaN(1,2);          %y2: empty
                                rdo.coef(9,:);     %X1
                                NaN(2,2);          %X2,X3: empty
                                rdo.coef(10,:);    %X4
                                rdo.coef(11,:);    %X5
                                rdo.coef(12,:);    %Z1
                                NaN(1,2);          %Z2: empty
                                rdo.coef(13,:);    %Z3
                                NaN(1,2);          %phi1
                                rdo.coef(14,:);    %phi2
                                NaN(2,2)];         %mu1,mu2: empty
            rdosys(8).resi=rdo.resi;
            rdosys(8).stats=rdo.stats;

%===========================================================================
            %Method 9: ECCorn
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(12).vari,Qmu)↙
+kron(bigsigma(13).vari,Pmu),[Awave,Bbar],y,G,N,T);
            rdosys(9).coef=[NaN(1,2);          %y1: empty    EQ. 1
                                rdo.coef(1,:);     %y2
                                rdo.coef(2,:);     %X1
                                rdo.coef(3,:);     %X2
                                rdo.coef(4,:);     %X3
                                NaN(2,2);          %X4,X5: empty
                                rdo.coef(5,:);     %Z1
                                rdo.coef(6,:);     %Z2
                                NaN(1,2);          %Z3: empty
                                rdo.coef(7,:);     %phi1
                                NaN(3,2);          %phi2,mu1,mu2: empty
                                rdo.coef(8,:);     %y1           EQ. 2
                                NaN(1,2);          %y2: empty
                                rdo.coef(9,:);     %X1
                                NaN(2,2);          %X2,X3: empty
                                rdo.coef(10,:);    %X4
                                rdo.coef(11,:);    %X5
                                rdo.coef(12,:);    %Z1
                                NaN(1,2);          %Z2: empty
                                rdo.coef(13,:);    %Z3
                                NaN(1,2);          %phi1
                                rdo.coef(14,:);    %phi2
                                NaN(2,2)];         %mu1,mu2: empty
            rdosys(9).resi=rdo.resi;
            rdosys(9).stats=rdo.stats;

%===========================================================================
            %Method 10: HTHT
%===========================================================================
            [rdo.coef,rdo.resi,rdo.stats]=G3SLS2(R,eye(N*T),kron(bigsigma(14).vari,Qmu)↙
```

```
n(bigsigma(15).vari,Pmu),[Awave,BHT],y,G,N,T);
            rdosys(10).coef=[NaN(1,2);        %y1: empty    EQ. 1
                            rdo.coef(1,:);    %y2
                            rdo.coef(2,:);    %X1
                            rdo.coef(3,:);    %X2
                            rdo.coef(4,:);    %X3
                            NaN(2,2);         %X4,X5: empty
                            rdo.coef(5,:);    %Z1
                            rdo.coef(6,:);    %Z2
                            NaN(1,2);         %Z3: empty
                            rdo.coef(7,:);    %phi1
                            NaN(3,2);         %phi2,mu1,mu2: empty
                            rdo.coef(8,:);    %y1           EQ. 2
                            NaN(1,2);         %y2: empty
                            rdo.coef(9,:);    %X1
                            NaN(2,2);         %X2,X3: empty
                            rdo.coef(10,:);   %X4
                            rdo.coef(11,:);   %X5
                            rdo.coef(12,:);   %Z1
                            NaN(1,2);         %Z2: empty
                            rdo.coef(13,:);   %Z3
                            NaN(1,2);         %phi1
                            rdo.coef(14,:);   %phi2
                            NaN(2,2)];        %mu1,mu2: empty
            rdosys(10).resi=rdo.resi;
            rdosys(10).stats=rdo.stats;

%==========================================================================
%             Method 11: HTAM
%==========================================================================
            rdosys(11).coef=[];
            rdosys(11).resi=[];
            rdosys(11).stats=[];

%==========================================================================
%           Method 12: HTBMS
%==========================================================================
            rdosys(12).coef=[];
            rdosys(12).resi=[];
            rdosys(12).stats=[];

%==========================================================================
            %Method 13,14,15: G3SPD_FEStd
%==========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G3SLS2(Rwave,Qmu,kron(bigsigma(20).↙
Qmu),Awave,y,G,N,T);
            rdosys(13).coef=[NaN(1,2);           %y1: empty    EQ. 1
                            rdo(1).coef(1,:);    %y2
                            rdo(1).coef(2,:);    %X1
                            rdo(1).coef(3,:);    %X2
                            rdo(1).coef(4,:);    %X3
```

```
                                NaN(9,2);               %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2:
empty
                                rdo(1).coef(5,:);    %y1           EQ. 2
                                NaN(1,2);               %y2: empty
                                rdo(1).coef(6,:);    %X1
                                NaN(2,2);               %X2,X3:empty
                                rdo(1).coef(7,:);    %X4
                                rdo(1).coef(8,:);    %X5
                                NaN(7,2)];              %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdosys(13).resi=rdo(1).resi;
            rdosys(13).stats=rdo(1).stats;

            % 2nd stage
            uhat=kron(eye(G),Pmu)*y-kron(eye(G),Pmu)*Rwave*rdo(1).coef(:,1); %vector of
residuals (dependent variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G3SLS2(Rbar,Pmu,kron(bigsigma(21).
vari,Pmu),Bbar,uhat,G,N,T);
            rdosys(14).coef=[NaN(7,2);              %y1,y2,X1,X2,X3,X4,X5: empty
                                rdo(2).coef(1,:);    %Z1
                                rdo(2).coef(2,:);    %Z2
                                NaN(1,2);               %Z3: empty
                                rdo(2).coef(3,:);    %phi1
                                NaN(3,2);               %phi2,mu1,mu2: empty
                                NaN(7,2);               %y1,y2,X1,X2,X3,X4,X5: empty
                                rdo(2).coef(4,:);    %Z1
                                NaN(1,2);               %Z2: empty
                                rdo(2).coef(5,:);    %Z3
                                NaN(1,2);               %phi1: empty
                                rdo(2).coef(6,:);    %phi2
                                NaN(2,2)];              %mu1,mu2: empty
            rdosys(14).resi=rdo(2).resi;
            rdosys(14).stats=rdo(2).stats;

            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdo(2).resi(1:(N*T),1),ones(N*T,
1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdo(2).resi((N*T+1):(2*N*T),1),
ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed
Effects
            Rfull=blkdiag(R1full,R2full);
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %
set of instruments
            C=single(blkdiag(C1,C2));
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G3SLSM(Rfull,kron(bigsigma(22).vari,
eye(N*T)),C,y,G,N,T,size(R1full,2));
            rdosys(15).coef=[NaN(1,6);              %y1: empty    EQ. 1
                                rdo(3).coef(1,:);    %y2
                                rdo(3).coef(2,:);    %X1
                                rdo(3).coef(3,:);    %X2
                                rdo(3).coef(4,:);    %X3
```

```
                                NaN(2,6);               %X4,X5: empty
                                rdo(3).coef(5,:);       %Z1
                                rdo(3).coef(6,:);       %Z2
                                NaN(1,6);               %Z3: empty
                                rdo(3).coef(8,:);       %phi1
                                NaN(1,6);               %phi2: empty
                                rdo(3).coef(7,:);       %mu1
                                NaN(1,6);               %mu2: empty
                                rdo(3).coef(9,:);       %y1           EQ. 2
                                NaN(1,6);               %y2: empty
                                rdo(3).coef(10,:);      %X1
                                NaN(2,6);               %X2,X3: empty
                                rdo(3).coef(11,:);      %X4
                                rdo(3).coef(12,:);      %X5
                                rdo(3).coef(13,:);      %Z1
                                NaN(1,6);               %Z2: empty
                                rdo(3).coef(14,:);      %Z3
                                NaN(1,6);               %phi1
                                rdo(3).coef(16,:);      %phi2
                                NaN(1,6);               %mu1: empty
                                rdo(3).coef(15,:)];     %mu2
            rdosys(15).resi=rdo(3).resi;
            rdosys(15).stats=rdo(3).stats;

%===========================================================================
            %Method 16,17,18: G3SPD_FECorn
%===========================================================================
            % 1st stage
            [rdo(1).coef,rdo(1).resi,rdo(1).stats]=G3SLS2(Rwave,Qmu,kron(bigsigma(23).↙
y,Qmu),Awave,y,G,N,T);
            rdosys(16).coef=[NaN(1,2);              %y1: empty    EQ. 1
                                rdo(1).coef(1,:);       %y2
                                rdo(1).coef(2,:);       %X1
                                rdo(1).coef(3,:);       %X2
                                rdo(1).coef(4,:);       %X3
                                NaN(9,2);               %X4,X5,Z1,Z2,Z3,phi1,phi2,mu1,mu2:↙
empty
                                rdo(1).coef(5,:);       %y1           EQ. 2
                                NaN(1,2);               %y2: empty
                                rdo(1).coef(6,:);       %X1
                                NaN(2,2);               %X2,X3:empty
                                rdo(1).coef(7,:);       %X4
                                rdo(1).coef(8,:);       %X5
                                NaN(7,2)];              %Z1,Z2,Z3,phi1,phi2,mu1,mu2: empty
            rdosys(16).resi=rdo(1).resi;
            rdosys(16).stats=rdo(1).stats;

            % 2nd stage
            uhat=kron(eye(G),Pmu)*y-kron(eye(G),Pmu)*Rwave*rdo(1).coef(:,1); %vector of↙
mean residuals (dependent variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G3SLS2(Rbar,Pmu,kron(bigsigma(24).↙
y,Pmu),Bbar,uhat,G,N,T);
```

```
            rdosys(17).coef=[NaN(7,2);              %y1,y2,X1,X2,X3,X4,X5: empty
                            rdo(2).coef(1,:);    %Z1
                            rdo(2).coef(2,:);    %Z2
                            NaN(1,2);            %Z3: empty
                            rdo(2).coef(3,:);    %phi1
                            NaN(3,2);            %phi2,mu1,mu2: empty
                            NaN(7,2);            %y1,y2,X1,X2,X3,X4,X5: empty
                            rdo(2).coef(4,:);    %Z1
                            NaN(1,2);            %Z2: empty
                            rdo(2).coef(5,:);    %Z3
                            NaN(1,2);            %phi1: empty
                            rdo(2).coef(6,:);    %phi2
                            NaN(2,2)];           %mu1,mu2: empty
            rdosys(17).resi=rdo(2).resi;
            rdosys(17).stats=rdo(2).stats;

            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdo(2).resi(1:(N*T),1),ones(N*T,1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdo(2).resi((N*T+1):(2*N*T),1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed Effects
            Rfull=blkdiag(R1full,R2full);
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2));
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G3SLSM(Rfull,kron(bigsigma(25).vari,eye(N*T)),C,y,G,N,T,size(R1full,2));
            rdosys(18).coef=[NaN(1,6);              %y1: empty    EQ. 1
                            rdo(3).coef(1,:);    %y2
                            rdo(3).coef(2,:);    %X1
                            rdo(3).coef(3,:);    %X2
                            rdo(3).coef(4,:);    %X3
                            NaN(2,6);            %X4,X5: empty
                            rdo(3).coef(5,:);    %Z1
                            rdo(3).coef(6,:);    %Z2
                            NaN(1,6);            %Z3: empty
                            rdo(3).coef(8,:);    %phi1
                            NaN(1,6);            %phi2: empty
                            rdo(3).coef(7,:);    %mu1
                            NaN(1,6);            %mu2: empty
                            rdo(3).coef(9,:);    %y1           EQ. 2
                            NaN(1,6);            %y2: empty
                            rdo(3).coef(10,:);   %X1
                            NaN(2,6);            %X2,X3: empty
                            rdo(3).coef(11,:);   %X4
                            rdo(3).coef(12,:);   %X5
                            rdo(3).coef(13,:);   %Z1
                            NaN(1,6);            %Z2: empty
                            rdo(3).coef(14,:);   %Z3
```

```
                                NaN(1,6);              %phi1
                                rdo(3).coef(16,:);     %phi2
                                NaN(1,6);              %mu1: empty
                                rdo(3).coef(15,:)];    %mu2
            rdosys(18).resi=rdo(3).resi;
            rdosys(18).stats=rdo(3).stats;

%===========================================================================
            %Method 19,20: G3SPD_HTHT
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            uhat=kron(eye(G),Pmu)*y-kron(eye(G),Pmu)*Rwave*rdo(1).coef(:,1); %vector of
means residuals (dependent variable)
            [rdo(2).coef,rdo(2).resi,rdo(2).stats]=G3SLS2(Rbar,Pmu,kron(bigsigma(26).
vari,Pmu),BHT,uhat,G,N,T);
            rdosys(19).coef=[NaN(7,2);             %y1,y2,X1,X2,X3,X4,X5: empty
                                rdo(2).coef(1,:);      %Z1
                                rdo(2).coef(2,:);      %Z2
                                NaN(1,2);              %Z3: empty
                                rdo(2).coef(3,:);      %phi1
                                NaN(3,2);              %phi2,mu1,mu2: empty
                                NaN(7,2);              %y1,y2,X1,X2,X3,X4,X5: empty
                                rdo(2).coef(4,:);      %Z1
                                NaN(1,2);              %Z2: empty
                                rdo(2).coef(5,:);      %Z3
                                NaN(1,2);              %phi1: empty
                                rdo(2).coef(6,:);      %phi2
                                NaN(2,2)];             %mu1,mu2: empty
            rdosys(19).resi=rdo(2).resi;
            rdosys(19).stats=rdo(2).stats;

            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdo(2).resi(1:(N*T),1),ones(N*T,
1)]; %full dataset, including all the original vars. plus the estimated Fixed Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdo(2).resi((N*T+1):(2*N*T),1),
ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed
Effects
            Rfull=blkdiag(R1full,R2full);
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdo(2).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %
set of instruments
            C=single(blkdiag(C1,C2));
            [rdo(3).coef,rdo(3).resi,rdo(3).stats]=G3SLSM(Rfull,kron(bigsigma(27).vari,
eye(N*T)),C,y,G,N,T,size(R1full,2));
            rdosys(20).coef=[NaN(1,6);             %y1: empty    EQ. 1
                                rdo(3).coef(1,:);      %y2
                                rdo(3).coef(2,:);      %X1
                                rdo(3).coef(3,:);      %X2
```

```
                                rdo(3).coef(4,:);    %X3
                                NaN(2,6);            %X4,X5: empty
                                rdo(3).coef(5,:);    %Z1
                                rdo(3).coef(6,:);    %Z2
                                NaN(1,6);            %Z3: empty
                                rdo(3).coef(8,:);    %phi1
                                NaN(1,6);            %phi2: empty
                                rdo(3).coef(7,:);    %mu1
                                NaN(1,6);            %mu2: empty
                                rdo(3).coef(9,:);    %y1            EQ. 2
                                NaN(1,6);            %y2: empty
                                rdo(3).coef(10,:);    %X1
                                NaN(2,6);            %X2,X3: empty
                                rdo(3).coef(11,:);   %X4
                                rdo(3).coef(12,:);   %X5
                                rdo(3).coef(13,:);   %Z1
                                NaN(1,6);            %Z2: empty
                                rdo(3).coef(14,:);   %Z3
                                NaN(1,6);            %phi1
                                rdo(3).coef(16,:);   %phi2
                                NaN(1,6);            %mu1: empty
                                rdo(3).coef(15,:)];  %mu2
            rdosys(20).resi=rdo(3).resi;
            rdosys(20).stats=rdo(3).stats;

%===========================================================================
%             Method 21,22: G3SPD_HTAM
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            rdosys(21).coef=[];
            rdosys(21).resi=[];
            rdosys(21).stats=[];

            % 3rd stage
            rdosys(22).coef=[];
            rdosys(22).resi=[];
            rdosys(22).stats=[];

%===========================================================================
%             Method 23,24: G3SPD_HTBMS
%===========================================================================
            % 1st stage
            % results are identical to Method 16

            % 2nd stage
            rdosys(23).coef=[];
            rdosys(23).resi=[];
            rdosys(23).stats=[];
```

```
            % 3rd stage
            rdosys(24).coef=[];
            rdosys(24).resi=[];
            rdosys(24).stats=[];

            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys');
        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMASYS_V2
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%============================================================================

function estimasys_v2
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        %load variables and estimated covariance matrixes
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','datos','correl');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        if exist(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')],'dir' )~=7
            mkdir(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                   Observation: ' num2str(obs)]);
            %open data for the observation
            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')↙
...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdosys');
```

```
%=======================================================================
            %Method 15: G3SPD_FEStd
%===========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(14).resi(1:(N*T),1),ones
(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed
Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(14).resi((N*T+1):(2*N*T),
1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated
Fixed Effects
            K=size(R1full,2);
            Rfull=blkdiag(R1full,R2full);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(14).resi(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(14).resi((N*T+1):(2*N*T),1),ones(N*T,1)];
%set of instruments
            A=single(blkdiag(C1,C2));
            clear('C1','C2');
            % Omega 1: GMM-3SLS asymptotic covariance matrix with estimated 3SLS
covariance matrix
            ENT=reshape(rdosys(15).resi,N*T,G); %arrange the set of G vectors of 3SLS
residuals
            V3SLS=(ENT'*ENT)./(N*T);
            instr3SLS=single(A*inv(A'*(kron(V3SLS,eye(N*T)))*A)*A');
            inv3SLS=inv(Rfull'*instr3SLS*Rfull);
            stder1=diag(inv3SLS).^0.5; %std. errors
            % Omega 2: GMM-3SLS with White-Huber VCV matrix (White, 1980) and 3SLS GMM
            %           weighting matrix
            VCVWH=diag(rdosys(15).resi.^2)./(N*T-K);  % VCVWH makes computing White's
covariance matrix easy
            VCV_White=(N*T-K) * inv3SLS * (Rfull'*instr3SLS*VCVWH*instr3SLS*Rfull) *
inv3SLS     % White's VCV
            stder2=diag(VCV_White).^0.5;
            clear('VCV_White','VCVWH');
            % Omega 3: GMM-3SLS with PCSEs (Beck & Katz, 1995)
            VCVBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VCVBK=blkdiag(VCVBK,kron((ET'*ET)./(T-K),eye(T)));
            end
            VCV_BK=inv3SLS * (Rfull'*instr3SLS*VCVBK*instr3SLS*Rfull) * inv3SLS;
            stder3=diag(VCV_BK).^0.5;
            clear('VCVBK','VCV_BK');
            % replace existing estimation
            stdernew=[stder1,stder2,stder3];
            rdosys(15).coef=squeeze(rdosys(15).coef(:,1));
            rdosys(15).coef(:,2:4)=[NaN(1,3);          %y1: empty    EQ. 1
                            stdernew(1,:);    %y2
                            stdernew(2,:);    %X1
                            stdernew(3,:);    %X2
                            stdernew(4,:);    %X3
```

```
                                   NaN(2,3);              %X4,X5: empty
                                   stdernew(5,:);      %Z1
                                   stdernew(6,:);      %Z2
                                   NaN(1,3);              %Z3: empty
                                   stdernew(8,:);      %phi1
                                   NaN(1,3);              %phi2: empty
                                   stdernew(7,:);      %mu1
                                   NaN(1,3);              %mu2: empty
                                   stdernew(9,:);      %y1          EQ. 2
                                   NaN(1,3);              %y2: empty
                                   stdernew(10,:);     %X1
                                   NaN(2,3);              %X2,X3: empty
                                   stdernew(11,:);     %X4
                                   stdernew(12,:);     %X5
                                   stdernew(13,:);     %Z1
                                   NaN(1,3);              %Z2: empty
                                   stdernew(14,:);     %Z3
                                   NaN(1,3);              %phi1
                                   stdernew(16,:);     %phi2
                                   NaN(1,3);              %mu1: empty
                                   stdernew(15,:)];    %mu2

%===========================================================================
            %Method 18: G3SPD_FECorn
%===========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(17).resi(1:(N*T),1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(17).resi((N*T+1):(2*N*T),1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed Effects
            K=size(R1full,2);
            Rfull=blkdiag(R1full,R2full);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            A=single(blkdiag(C1,C2));
            clear('C1','C2');
            % Omega 1: GMM-3SLS asymptotic covariance matrix with estimated 3SLS covariance matrix
            ENT=reshape(rdosys(18).resi,N*T,G); %arrange the set of G vectors of 3SLS residuals
            V3SLS=(ENT'*ENT)./(N*T);
            instr3SLS=single(A*inv(A'*(kron(V3SLS,eye(N*T)))*A)*A');
            inv3SLS=inv(Rfull'*instr3SLS*Rfull);
            stder1=diag(inv3SLS).^0.5; %std. errors
            % Omega 2: GMM-3SLS with White-Huber VCV matrix (White, 1980) and 3SLS GMM
            %          weighting matrix
            VCVWH=diag(rdosys(18).resi.^2)./(N*T-K);  % VCVWH makes computing White's
```

```
riance matrix easy
            VCV_White=(N*T-K) * inv3SLS * (Rfull'*instr3SLS*VCVWH*instr3SLS*Rfull) *
3SLS;       % White's VCV
            stder2=diag(VCV_White).^0.5;
            clear('VCV_White','VCVWH');
            % Omega 3: GMM-3SLS with PCSEs (Beck & Katz, 1995)
            VCVBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VCVBK=blkdiag(VCVBK,kron((ET'*ET)./(T-K),eye(T)));
            end
            VCV_BK=inv3SLS * (Rfull'*instr3SLS*VCVBK*instr3SLS*Rfull) * inv3SLS;
            stder3=diag(VCV_BK).^0.5;
            clear('VCVBK','VCV_BK');
            % replace existing estimation
            stdernew=[stder1,stder2,stder3];
            rdosys(18).coef=squeeze(rdosys(18).coef(:,1));
            rdosys(18).coef(:,2:4)=[NaN(1,3);           %y1: empty    EQ. 1
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,3);            %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,3);            %Z3: empty
                                stdernew(8,:);    %phi1
                                NaN(1,3);            %phi2: empty
                                stdernew(7,:);    %mu1
                                NaN(1,3);            %mu2: empty
                                stdernew(9,:);    %y1           EQ. 2
                                NaN(1,3);            %y2: empty
                                stdernew(10,:);   %X1
                                NaN(2,3);            %X2,X3: empty
                                stdernew(11,:);   %X4
                                stdernew(12,:);   %X5
                                stdernew(13,:);   %Z1
                                NaN(1,3);            %Z2: empty
                                stdernew(14,:);   %Z3
                                NaN(1,3);            %phi1
                                stdernew(16,:);   %phi2
                                NaN(1,3);            %mu1: empty
                                stdernew(15,:)];  %mu2

%==========================================================================
            %Method 19,20: G3SPD_HTHT
%==========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(19).resi(1:(N*T),1),ones
(N*T,1)];    %full dataset, including all the original vars. plus the estimated Fixed
Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(19).resi((N*T+1):(2*N*T),
```

```
ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated
Effects
            K=size(R1full,2);
            Rfull=blkdiag(R1full,R2full);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi((N*T+1):(2*N*T),1),ones(N*T,1)];
%set of instruments
            A=single(blkdiag(C1,C2));
            clear('C1','C2');
            % Omega 1: GMM-3SLS asymptotic covariance matrix with estimated 3SLS
covariance matrix
            ENT=reshape(rdosys(20).resi,N*T,G); %arrange the set of G vectors of 3SLS
residuals
            V3SLS=(ENT'*ENT)./(N*T);
            instr3SLS=single(A*inv(A'*(kron(V3SLS,eye(N*T)))*A)*A');
            inv3SLS=inv(Rfull'*instr3SLS*Rfull);
            stder1=diag(inv3SLS).^0.5; %std. errors
            % Omega 2: GMM-3SLS with White-Huber VCV matrix (White, 1980) and 3SLS GMM
            %           weighting matrix
            VCVWH=diag(rdosys(20).resi.^2)./(N*T-K);  % VCVWH makes computing White's
covariance matrix easy
            VCV_White=(N*T-K) * inv3SLS * (Rfull'*instr3SLS*VCVWH*instr3SLS*Rfull) *
inv3SLS;     % White's VCV
            stder2=diag(VCV_White).^0.5;
            clear('VCV_White','VCVWH');
            % Omega 3: GMM-3SLS with PCSEs (Beck & Katz, 1995)
            VCVBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VCVBK=blkdiag(VCVBK,kron((ET'*ET)./(T-K),eye(T)));
            end
            VCV_BK=inv3SLS * (Rfull'*instr3SLS*VCVBK*instr3SLS*Rfull) * inv3SLS;
            stder3=diag(VCV_BK).^0.5;
            clear('VCVBK','VCV_BK');
            % replace existing estimation
            stdernew=[stder1,stder2,stder3];
            rdosys(20).coef=squeeze(rdosys(20).coef(:,1));
            rdosys(20).coef(:,2:4)=[NaN(1,3);            %y1: empty    EQ. 1
                            stdernew(1,:);    %y2
                            stdernew(2,:);    %X1
                            stdernew(3,:);    %X2
                            stdernew(4,:);    %X3
                            NaN(2,3);            %X4,X5: empty
                            stdernew(5,:);    %Z1
                            stdernew(6,:);    %Z2
                            NaN(1,3);            %Z3: empty
                            stdernew(8,:);    %phi1
                            NaN(1,3);            %phi2: empty
                            stdernew(7,:);    %mu1
                            NaN(1,3);            %mu2: empty
```

```
                            stdernew(9,:);     %y1           EQ. 2
                            NaN(1,3);             %y2: empty
                            stdernew(10,:);    %X1
                            NaN(2,3);             %X2,X3: empty
                            stdernew(11,:);    %X4
                            stdernew(12,:);    %X5
                            stdernew(13,:);    %Z1
                            NaN(1,3);             %Z2: empty
                            stdernew(14,:);    %Z3
                            NaN(1,3);             %phi1
                            stdernew(16,:);    %phi2
                            NaN(1,3);             %mu1: empty
                            stdernew(15,:)];   %mu2

            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys');
        end
        toc
    end
end
```

```
%===================================================================
% Function ESTIMASYS_V3
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%=======================================================================

function estimasys_v3
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '         -------------------------------');
        disp(['         Rho= ' num2str(rho)]);
        disp( '         -------------------------------');
        tic

        %load variables and estimated covariance matrixes
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        clear('R1bar','R2bar');
        Bbar=single(kron(eye(G),Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]))); %between
moments

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                  Observation: ' num2str(obs)]);
            %load system estimation
            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')
```

```
...
            '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
        load(filename,'rdosys');
        %load variance estimates
        filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
            '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f' )];
        load(filename,'bigsigma');
        %load dependent variable
        y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
        clear('rdo');

%===========================================================================
        %Method 15: G3SPD_Std 3rd Step
%===========================================================================
        % Covariance estimation asumming Homoskedasticity
        %---------------------------------------------------------------
        resi2nd=single(rdosys(14).resi); %residuals from the 2nd Step estimation
        resi3rd=single(rdosys(15).resi); %residuals from the 3rd Step estimation
        R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,resi2nd(1:(N*T),1),ones(N*T,1)];
        R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,resi2nd((N*T+1):(2*N*T),1),ones(N*T,1)];
        Rfull=blkdiag(R1full,R2full); %set of regressors of the system
        Kfull=size(R1full,2);
        clear('R1full','R2full');
        clear('R1full','R2full');
        C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd(1:(N*T),1),ones(N*T,1)]; %set of instruments
        C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
        C=single(blkdiag(C1,C2)); %set of instruments of the system
        clear('C1','C2');
        PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

        % OmegaGMM: stored in column 2
        ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
        V=(ENT'*ENT)./(N*T-Kfull);
        RCinvCVC=single(Rfull'*C*inv(C'*(kron(V,eye(N*T)))*C));
        OmegaGMM=inv(RCinvCVC*C'*Rfull);
        stderGMM=diag(OmegaGMM).^0.5; %std. errors
        clear('V');
        %Coefficients
        thetaStd=inv(Rfull'*
        % OmegaGMM: stored in column 2
        ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
        V=(ENT'*ENT)./(N*T-Kfull);
        sigmaGMM=kron(V,eye(N*T));
        RCinvCVC=single(Rfull'*C*inv(C'*()*C));
        OmegaGMM=inv(RCinvCVC*C'*Rfull);
        stderGMM=diag(OmegaGMM).^0.5; %std. errors
```

```
        clear('V');

        % WH-HC1: stored in column 3
        sigmaHC1=C'*diag(resi3rd.^2)*C;
        OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
        stderHC1=diag(OmegaHC1).^0.5;
        clear('sigmaHC1','OmegaHC1');

        % WH-HC2: stored in column 4
        sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
        OmegaHC2=OmegaGMM * (RCinvCVC*sigmaHC2*RCinvCVC') * OmegaGMM;
        stderHC2=diag(OmegaHC2).^0.5;
        clear('sigmaHC2','OmegaHC2');

        % WH-LE: stored in column 5
        resi3rdmodi=(resi3rd)./(1-diag(PR)); %epsilon**
        sigmaLE=C'*diag(resi3rdmodi.^2)*C;
        OmegaLE=OmegaGMM * (RCinvCVC*sigmaLE*RCinvCVC') * OmegaGMM;
        stderLE=diag(OmegaLE).^0.5;
        clear('OmegaLE');

        % WH-HC3: stored in column 6
        sigmaHC3=sigmaLE-(C'*resi3rdmodi*resi3rdmodi'*C);
        OmegaHC3=OmegaGMM * (RCinvCVC*sigmaHC3*RCinvCVC') * OmegaGMM;
        stderHC3=diag(OmegaHC3).^0.5;
        clear('sigmaLE','resi3rdmodi','sigmaHC3','OmegaHC3');

        % BK: stored in column 7
        VBK=[];
        for g=1:1:G
            ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
            VBK=blkdiag(VBK,kron((ET'*ET)./(T-K),eye(T)));
        end
        sigmaBK=C'*VBK*C;
        OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
        stderBK=diag(OmegaBK).^0.5;
        clear('ET','ENT','VBK','sigmaBK','OmegaBK');

        % MT: stored in column 8
        ENT2nd=reshape(resi2nd,N*T,G); %arrange the set of G vectors of 2nd Step↙
esation

        RBinvBVB=single(Rbar'*Bbar*inv(Bbar'*(kron((ENT2nd'*Pmu*ENT2nd)./(N-Kbar),↙
Pmu)*Bbar));
        OmegaGMM2nd=inv(RBinvBVB*Bbar'*Rbar);
        CEEB=C'*resi3rd*resi3rd'*Bbar;
        CEHB=C'*resi3rd*resi2nd'*Bbar;
        clear('Rbar','Kbar','Bbar','ENT2nd');

        Psi1=OmegaGMM; %OmegaGMM 3rd Step
        Psi2=(RCinvCVC*CEEB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
        Psi3=(RCinvCVC*CEHB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
        OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
```

```
            stderMT=diag(OmegaMT).^0.5;
            clear('RBinvBVB','OmegaGMM2nd','CEEB','CEHB','Psi1','Psi2',...
                'Psi3','RCinvCVC','OmegaGMM','OmegaMT');

            % replace existing estimation
            stdernew=[rdosys(15).coef(:,1),stderGMM,stderHC1,stderHC2,stderLE,stderHC3,↙
stderBE,stderMT];
            rdosys(15).coef(:)=[NaN(1,8);           %y1: empty    EQ. 1
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,8);            %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,8);            %Z3: empty
                                stdernew(8,:);    %phi1
                                NaN(1,8);            %phi2: empty
                                stdernew(7,:);    %mu1
                                NaN(1,8);            %mu2: empty
                                stdernew(9,:);    %y1           EQ. 2
                                NaN(1,8);            %y2: empty
                                stdernew(10,:);   %X1
                                NaN(2,8);            %X2,X3: empty
                                stdernew(11,:);   %X4
                                stdernew(12,:);   %X5
                                stdernew(13,:);   %Z1
                                NaN(1,8);            %Z2: empty
                                stdernew(14,:);   %Z3
                                NaN(1,8);            %phi1
                                stdernew(16,:);   %phi2
                                NaN(1,8);            %mu1: empty
                                stdernew(15,:)];  %mu2

%===========================================================================
%           Method 25: G3SPD_Std 3rd Step: fully heteroskedastic
%===========================================================================
            % Coefficients and Covariance estimation asumming intra-eq.↙
homoskedasticity
            %↙
--------------------------------------------------------------------------
            V=[diag(bigsigma(1).vari11),diag(bigsigma(1).vari12);...
               diag(bigsigma(1).vari12),diag(bigsigma(1).vari22)]; %consistent↙
covariance estimator (from 2SLS)
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            % residuals
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            bigsigmanew(1).vari11=resi3rd(:,1).^2;
            bigsigmanew(1).vari12=resi3rd(:,1).*resi3rd(:,2);
```

```
        bigsigmanew(1).vari22=resi3rd(:,2).^2;
        V=[diag(bigsigmanew(1).vari11),diag(bigsigmanew(1).vari12);...
           diag(bigsigmanew(1).vari12),diag(bigsigmanew(1).vari22)];
        RCinvCVC=single(Rfull'*C*inv(C'*V*C));
        OmegaGMM=inv(RCinvCVC*C'*Rfull);
        stderGMM=diag(OmegaGMM).^0.5; %std. errors
        clear('V');

        % WH-HC1: stored in column 3
        sigmaHC1=C'*diag(resi3rd.^2)*C;
        OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
        stderHC1=diag(OmegaHC1).^0.5;
        clear('sigmaHC1','OmegaHC1');

        % WH-HC2: stored in column 4
        sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
        OmegaHC2=OmegaGMM * (RCinvCVC*sigmaHC2*RCinvCVC') * OmegaGMM;
        stderHC2=diag(OmegaHC2).^0.5;
        clear('sigmaHC2','OmegaHC2');

        % WH-LE: stored in column 5
        resi3rdmodi=(resi3rd)./(1-diag(PR)); %epsilon**
        sigmaLE=C'*diag(resi3rdmodi.^2)*C;
        OmegaLE=OmegaGMM * (RCinvCVC*sigmaLE*RCinvCVC') * OmegaGMM;
        stderLE=diag(OmegaLE).^0.5;
        clear('OmegaLE');

        % WH-HC3: stored in column 6
        sigmaHC3=sigmaLE-(C'*resi3rdmodi*resi3rdmodi'*C);
        OmegaHC3=OmegaGMM * (RCinvCVC*sigmaHC3*RCinvCVC') * OmegaGMM;
        stderHC3=diag(OmegaHC3).^0.5;
        clear('sigmaLE','resi3rdmodi','sigmaHC3','OmegaHC3');

        % BK: stored in column 7
        VBK=[];
        for g=1:1:G
            ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
            VBK=blkdiag(VBK,kron((ET'*ET)./(T-K),eye(T)));
        end
        sigmaBK=C'*VBK*C;
        OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
        stderBK=diag(OmegaBK).^0.5;
        clear('ET','ENT','VBK','sigmaBK','OmegaBK');

        % MT: stored in column 8
        ENT2nd=reshape(resi2nd,N*T,G); %arrange the set of G vectors of 2nd Step↙
mation
        RBinvBVB=single(Rbar'*Bbar*inv(Bbar'*(kron((ENT2nd'*Pmu*ENT2nd)./(N-Kbar),↙
Pmu)*Bbar));
        OmegaGMM2nd=inv(RBinvBVB*Bbar'*Rbar);
        CEEB=C'*resi3rd*resi3rd'*Bbar;
        CEHB=C'*resi3rd*resi2nd'*Bbar;
```

```
            clear('Rbar','Kbar','Bbar','ENT2nd');

            Psi1=OmegaGMM; %OmegaGMM 3rd Step
            Psi2=(RCinvCVC*CEEB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
            Psi3=(RCinvCVC*CEHB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('RBinvBVB','OmegaGMM2nd','CEEB','CEHB','Psi1','Psi2',...
                'Psi3','RCinvCVC','OmegaGMM','OmegaMT');

            % replace existing estimation
            stdernew=[rdosys(15).coef(:,1),stderGMM,stderHC1,stderHC2,stderLE,stderHC3,↙
stdBd,stderMT];
            rdosys(25).coef(:)=[NaN(1,8);          %y1: empty    EQ. 1
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,8);            %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,8);            %Z3: empty
                                stdernew(8,:);    %phi1
                                NaN(1,8);            %phi2: empty
                                stdernew(7,:);    %mu1
                                NaN(1,8);            %mu2: empty
                                stdernew(9,:);    %y1            EQ. 2
                                NaN(1,8);            %y2: empty
                                stdernew(10,:);   %X1
                                NaN(2,8);            %X2,X3: empty
                                stdernew(11,:);   %X4
                                stdernew(12,:);   %X5
                                stdernew(13,:);   %Z1
                                NaN(1,8);            %Z2: empty
                                stdernew(14,:);   %Z3
                                NaN(1,8);            %phi1
                                stdernew(16,:);   %phi2
                                NaN(1,8);            %mu1: empty
                                stdernew(15,:)];  %mu2
            rdosys(25).resi=resi3rd;
            rdosys(25).stats=[];

%===========================================================================
            %Method 16,17,18: G3SPD_FECorn
%===========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(17).resi(1:(N*T),1),ones↙
(N*T)];     %full dataset, including all the original vars. plus the estimated Fixed↙
Effe
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(17).resi((N*T+1):(2*N*T),↙
ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated↙
Fixed Effects
```

```
            Rfull=blkdiag(R1full,R2full);
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi((N*T+1):(2*N*T),1),ones(N*T,1)];
%set of instruments
            C=single(blkdiag(C1,C2));
            clear('C1','C2');
            [rdo.coef,rdo.resi,rdo.stats]=G3SLSM_hetero(Rfull,bigsigma(2).vari,C,y,G,N,
T,Kf);
            rdosys(18).coef(:)=[NaN(1,6);           %y1: empty    EQ. 1
                            rdo(3).coef(1,:);    %y2
                            rdo(3).coef(2,:);    %X1
                            rdo(3).coef(3,:);    %X2
                            rdo(3).coef(4,:);    %X3
                            NaN(2,6);            %X4,X5: empty
                            rdo(3).coef(5,:);    %Z1
                            rdo(3).coef(6,:);    %Z2
                            NaN(1,6);            %Z3: empty
                            rdo(3).coef(8,:);    %phi1
                            NaN(1,6);            %phi2: empty
                            rdo(3).coef(7,:);    %mu1
                            NaN(1,6);            %mu2: empty
                            rdo(3).coef(9,:);    %y1           EQ. 2
                            NaN(1,6);            %y2: empty
                            rdo(3).coef(10,:);   %X1
                            NaN(2,6);            %X2,X3: empty
                            rdo(3).coef(11,:);   %X4
                            rdo(3).coef(12,:);   %X5
                            rdo(3).coef(13,:);   %Z1
                            NaN(1,6);            %Z2: empty
                            rdo(3).coef(14,:);   %Z3
                            NaN(1,6);            %phi1
                            rdo(3).coef(16,:);   %phi2
                            NaN(1,6);            %mu1: empty
                            rdo(3).coef(15,:)];  %mu2
            rdosys(18).resi=rdo(3).resi;
            rdosys(18).stats=rdo(3).stats;

%===========================================================================
            %Method 19,20: G3SPD_HTHT
%===========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(19).resi(1:(N*T),1),ones
(N*T)];     %full dataset, including all the original vars. plus the estimated Fixed
Effe
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(19).resi((N*T+1):(2*N*T),
1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated
Fixed Effects
            Rfull=blkdiag(R1full,R2full);
            Kfull=size(R1full,2);
```

```
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2));
            clear('C1','C2');
            [rdo.coef,rdo.resi,rdo.stats]=G3SLSM_hetero(Rfull,bigsigma(3).vari,C,y,G,N,T,K);
            rdosys(20).coef(:,)=[NaN(1,6);           %y1: empty     EQ. 1
                            rdo(3).coef(1,:);    %y2
                            rdo(3).coef(2,:);    %X1
                            rdo(3).coef(3,:);    %X2
                            rdo(3).coef(4,:);    %X3
                            NaN(2,6);            %X4,X5: empty
                            rdo(3).coef(5,:);    %Z1
                            rdo(3).coef(6,:);    %Z2
                            NaN(1,6);            %Z3: empty
                            rdo(3).coef(8,:);    %phi1
                            NaN(1,6);            %phi2: empty
                            rdo(3).coef(7,:);    %mu1
                            NaN(1,6);            %mu2: empty
                            rdo(3).coef(9,:);    %y1            EQ. 2
                            NaN(1,6);            %y2: empty
                            rdo(3).coef(10,:);    %X1
                            NaN(2,6);            %X2,X3: empty
                            rdo(3).coef(11,:);   %X4
                            rdo(3).coef(12,:);   %X5
                            rdo(3).coef(13,:);   %Z1
                            NaN(1,6);            %Z2: empty
                            rdo(3).coef(14,:);   %Z3
                            NaN(1,6);            %phi1
                            rdo(3).coef(16,:);   %phi2
                            NaN(1,6);            %mu1: empty
                            rdo(3).coef(15,:)];  %mu2
            rdosys(20).resi=rdo(3).resi;
            rdosys(20).stats=rdo(3).stats;

            %Save results
            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys','Rfull','C');
        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMASYS_V4
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%=============================================================================

function estimasys_v4
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu
    %Redefine Pmu and Qmu for a system of 2 equations
    Pmu=blkdiag(Pmu,Pmu);
    Qmu=blkdiag(Qmu,Qmu);

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        %load variables
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %Arrange regressors
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        clear('R1bar','R2bar');

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);

            %load system estimation
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            load(filename,'rdosys');
            %load homoskedastic variance estimates
            filename=['.\varianza\vari_homo\vari_' num2str(nt,'%3.0f') '_' num2str
(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%
4.0f )];
            load(filename,'bigsigma'); bshomo=bigsigma;
            %load heteroskedastic variance estimates
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str
(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%
4.0f )];
            load(filename,'bigsigma'); bshete=bigsigma;
            clear bigsigma

            %arrange variables
            y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
            R1wave=single([datos(1,obs).Y(:,2) X1 X2 X3]); %Eq. 1 within regressors
            R2wave=single([datos(1,obs).Y(:,1) X1 X4 X5]); %Eq. 2 within regressors
            Rwave=single(blkdiag(R1wave,R2wave)); %system within regressors
            Kwave=single(size(Rwave,2));
            clear('R1wave','R2wave');

%==========================================================================
%           Move Methods (6 to 25) to (7 to 26)
%           Eliminate NaN from sets of coefficients and std. errors (all
%           methods)
%           Introduce methods 6 (BEHTM) and 26 (G3SPD_HTMnew 3rd Step)
%==========================================================================
% Methods 26 (G3SPD_HTMnew 3rd Step) and 6 (BEHTM) are not included in
% system estimation, though their place are accounted for and filled with
% NaN.

            % Method 26: G3SPD_HTMnew 3rd Step
            % place new estimation (pending)
            rdosys(26).coef=[];
            rdosys(26).resi=[];
            rdosys(26).stats=[];

            % Methods 24 to 6
            for met=24:(-1):6
                if size(rdosys(met).coef,2)>0
                    rdosys(met+1).coef=[rdosys(met).coef(2:5,:);    %y2,X1,X2,X3
Equation 1
                                        rdosys(met).coef(8:9,:);    %Z1,Z2
```

```
                                        rdosys(met).coef(11,:);     %Varphi1
                                        rdosys(met).coef(13,:);     %Phi1
                                        rdosys(met).coef(15,:);     %y1↙
Equa 2
                                        rdosys(met).coef(17,:);     %X1
                                        rdosys(met).coef(20:22,:);  %X4,X5,Z1
                                        rdosys(met).coef(24,:);     %Z3
                                        rdosys(met).coef(26,:);     %Varphi2
                                        rdosys(met).coef(28,:)];    %Phi2
                rdosys(met+1).resi=rdosys(met).resi;
                rdosys(met+1).stats=rdosys(met).stats;
            else
                rdosys(met+1).coef=[];
                rdosys(met+1).resi=[];
                rdosys(met+1).stats=[];
            end
        end

        % Method 6: BEHTM
        % place new estimation (pending)
        rdosys(6).coef=[];
        rdosys(6).resi=[];
        rdosys(6).stats=[];

        % Methods 1 to 5
        for met=1:1:5
            if size(rdosys(met).coef,2)>0
                rdosys(met).coef=[rdosys(met).coef(2:5,:);    %y2,X1,X2,X3↙
Equa 1
                                        rdosys(met).coef(8:9,:);    %Z1,Z2
                                        rdosys(met).coef(11,:);     %Varphi1
                                        rdosys(met).coef(13,:);     %Phi1
                                        rdosys(met).coef(15,:);     %y1↙
Equa 2
                                        rdosys(met).coef(17,:);     %X1
                                        rdosys(met).coef(20:22,:);  %X4,X5,Z1
                                        rdosys(met).coef(24,:);     %Z3
                                        rdosys(met).coef(26,:);     %Varphi2
                                        rdosys(met).coef(28,:)];    %Phi2
                rdosys(met).resi=rdosys(met).resi;
                rdosys(met).stats=rdosys(met).stats;
            else
                rdosys(met).coef=[];
                rdosys(met).resi=[];
                rdosys(met).stats=[];
            end
        end

%===========================================================================
%           Method 16: G3SPD_Std 3rd Step -- Homoskedastic Covariances
%===========================================================================
        MuHat=single(rdosys(15).resi); %residuals from the 2nd Step estimation
```

```
            resi3rd=single(rdosys(16).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T)); %homoskedastic intra-equation variance
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(16).coef=[rdosys(16).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];

%=========================================================================
%           Method 33: G3SPD_Std 3rd Step -- Heteroskedastic Covariances
%=========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(1).vari11),diag(bshete(1).vari12);...
               diag(bshete(1).vari12),diag(bshete(1).vari22)];
```

```
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic
mations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
anot estimated
            rdosys(33).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(33).resi=resi3rd;
            rdosys(33).stats=[];

%=========================================================================
%           Method 19: G3SPD_Corn 3rd Step -- Homoskedastic Covariances
%=========================================================================
            MuHat=single(rdosys(18).resi); %residuals from the 2nd Step estimation
            resi3rd=single(rdosys(19).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):
(2*N1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
```

```
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of
instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
are not estimated
            rdosys(19).coef=[rdosys(19).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,
NaN(16,1)];

%======================================================================
%           Method 34: G3SPD_Corn 3rd Step -- Heteroskedastic Covariances
%======================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(2).vari11),diag(bshete(2).vari12);...
               diag(bshete(2).vari12),diag(bshete(2).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
```

```
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic↙
equations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)↙
are not estimated
            rdosys(34).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(34).resi=resi3rd;
            rdosys(34).stats=[];

%==========================================================================
%           Method 21: G3SPD_HTM 3rd Step -- Homoskedastic Covariances
%==========================================================================
            MuHat=single(rdosys(20).resi); %residuals from the 2nd Step estimation
            resi3rd=single(rdosys(21).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):↙
(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of↙
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of↙
instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
```

```
            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(21).coef=[rdosys(21).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];

%=========================================================================
%           Method 35: G3SPD_HTM 3rd Step -- Heteroskedastic Covariances
%=========================================================================
            % place new estimation (pending)
            rdosys(35).coef=[];
            rdosys(35).resi=[];
            rdosys(35).stats=[];

%=========================================================================
%           Method 27,28: G3SPD_Std_alter 3rd Step -- Homoskedastic Cov.
%=========================================================================
            %generates pseudo-effects (stored in 27)
            thetaFEStd=[rdosys(2).coef(1:4,1);rdosys(2).coef(9:12,1)]; %(Y1,X1,X2,X3) &(Y2,X4,X5)
            thetaBEStd=[rdosys(4).coef(5:7,1);rdosys(4).coef(13:15,1)]; %(Z1,Z2,Varphi1)&(Z1,Z3,Varphi2)
            MuHat=Pmu*y-Pmu*Rwave*thetaFEStd-Rbar*thetaBEStd;
            rdoeq1(27).resi=MuHat;

            % consistent covariance estimator (from 2SLS)
```

```
            V=kron(bshomo(32).vari,eye(N*T));
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            clear('V');

            % Coefficients (stored in 28, column 1) and Residuals
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(28).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(28).resi=resi3rd;
            rdosys(28).stats=[];
```

```
%=======================================================================
%            Method 36: G3SPD_Std_alter 3rd Step -- Heteroskedastic Cov.
%=======================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(4).vari11),diag(bshete(4).vari12);...
               diag(bshete(4).vari12),diag(bshete(4).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic↙
mations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)↙
are not estimated
            rdosys(36).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(36).resi=resi3rd;
            rdosys(36).stats=[];

%=======================================================================
%            Method 29,30: G3SPD_Corn_alter 3rd Step -- Homoskedastic Cov.
%=======================================================================
            %generates pseudo-effects (stored in 29)
```

```
            thetaFEBal=[rdosys(3).coef(1:4,1);rdosys(3).coef(9:12,1)];%(Y1,X1,X2,X3) &(Y2,X4,X5)
            thetaBEBal=[rdosys(5).coef(5:7,1);rdosys(5).coef(13:15,1)];%(Z1,Z2,Varphi1)&(Z1,Z3,Varphi2)
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEBal;
            rdoeq1(29).resi=MuHat;

            % consistent covariance estimator (from 2SLS)
            V=kron(bshomo(33).vari,eye(N*T));
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            clear('V');

            % Coefficients (stored in 30, column 1) and Residuals
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
```

```
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
are not estimated
            rdosys(30).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(30).resi=resi3rd;
            rdosys(30).stats=[];

%===========================================================================
%           Method 37: G3SPD_HTM_alter 3rd Step -- Heteroskedastic Cov.
%===========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(5).vari11),diag(bshete(5).vari12);...
               diag(bshete(5).vari12),diag(bshete(5).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic
equations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
are not estimated
```

```
            rdosys(37).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(37).resi=resi3rd;
            rdosys(37).stats=[];

%==========================================================================
%           Method 31,32: G3SPD_HTM_alter 3rd Step -- Homoskedastic Cov.
%==========================================================================
            % new MuHat
            rdosys(31).resi=[];
            % place new estimation (pending)
            rdosys(32).coef=[];
            rdosys(32).resi=[];
            rdosys(32).stats=[];

%==========================================================================
%           Method 38: G3SPD_HTM_alter 3rd Step -- Heteroskedastic Cov.
%==========================================================================
            % place new estimation (pending)
            rdosys(38).coef=[];
            rdosys(38).resi=[];
            rdosys(38).stats=[];

%==========================================================================
%           Save results
%==========================================================================
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys','Rfull','C');
        end
        toc
    end
end
```

```
%======================================================================
% Function ESTIMASYS_V5
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%==========================================================================

function estimasys_v5
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu
    %Redefine Pmu and Qmu for a system of 2 equations
    Pmu=blkdiag(Pmu,Pmu);
    Qmu=blkdiag(Qmu,Qmu);

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '         --------------------------------');
        disp(['         Rho= ' num2str(rho)]);
        disp( '         --------------------------------');
        tic

        %load variables
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %Arrange regressors
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        clear('R1bar','R2bar');

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);

            %load system estimation
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdosys');
            %load homoskedastic variance estimates
            filename=['.\varianza\vari_homo\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f' )];
            load(filename,'bigsigma'); bshomo=bigsigma;
            %load heteroskedastic variance estimates
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f' )];
            load(filename,'bigsigma'); bshete=bigsigma;
            clear bigsigma

            %arrange variables
            y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
            R1wave=single([datos(1,obs).Y(:,2) X1 X2 X3]); %Eq. 1 within regressors
            R2wave=single([datos(1,obs).Y(:,1) X1 X4 X5]); %Eq. 2 within regressors
            Rwave=single(blkdiag(R1wave,R2wave)); %system within regressors
            Kwave=single(size(Rwave,2));
            clear('R1wave','R2wave');

%===========================================================================
%           Move Methods (6 to 25) to (7 to 26)
%           Eliminate NaN from sets of coefficients and std. errors (all
%           methods)
%           Introduce methods 6 (BEHTM) and 26 (G3SPD_HTMnew 3rd Step)
%===========================================================================
% Methods 26 (G3SPD_HTMnew 3rd Step) and 6 (BEHTM) are not included in
% system estimation, though their place are accounted for and filled with
% NaN.

            % Method 26: G3SPD_HTMnew 3rd Step
            % place new estimation (pending)
            rdosys(26).coef=[];
            rdosys(26).resi=[];
            rdosys(26).stats=[];

            % Methods 24 to 6
            for met=24:(-1):6
                if size(rdosys(met).coef,2)>0
                    rdosys(met+1).coef=[rdosys(met).coef(2:5,:);    %y2,X1,X2,X3↙
Equa 1
                                        rdosys(met).coef(8:9,:);    %Z1,Z2
```

```
                                        rdosys(met).coef(11,:);     %Varphi1
                                        rdosys(met).coef(13,:);     %Phi1
                                        rdosys(met).coef(15,:);     %y1↙
Equa 2
                                        rdosys(met).coef(17,:);     %X1
                                        rdosys(met).coef(20:22,:);  %X4,X5,Z1
                                        rdosys(met).coef(24,:);     %Z3
                                        rdosys(met).coef(26,:);     %Varphi2
                                        rdosys(met).coef(28,:)];    %Phi2
                    rdosys(met+1).resi=rdosys(met).resi;
                    rdosys(met+1).stats=rdosys(met).stats;
                else
                    rdosys(met+1).coef=[];
                    rdosys(met+1).resi=[];
                    rdosys(met+1).stats=[];
                end
            end

            % Method 6: BEHTM
            % place new estimation (pending)
            rdosys(6).coef=[];
            rdosys(6).resi=[];
            rdosys(6).stats=[];

            % Methods 1 to 5
            for met=1:1:5
                if size(rdosys(met).coef,2)>0
                    rdosys(met).coef=[rdosys(met).coef(2:5,:);    %y2,X1,X2,X3↙
Equa 1
                                        rdosys(met).coef(8:9,:);    %Z1,Z2
                                        rdosys(met).coef(11,:);     %Varphi1
                                        rdosys(met).coef(13,:);     %Phi1
                                        rdosys(met).coef(15,:);     %y1↙
Equa 2
                                        rdosys(met).coef(17,:);     %X1
                                        rdosys(met).coef(20:22,:);  %X4,X5,Z1
                                        rdosys(met).coef(24,:);     %Z3
                                        rdosys(met).coef(26,:);     %Varphi2
                                        rdosys(met).coef(28,:)];    %Phi2
                    rdosys(met).resi=rdosys(met).resi;
                    rdosys(met).stats=rdosys(met).stats;
                else
                    rdosys(met).coef=[];
                    rdosys(met).resi=[];
                    rdosys(met).stats=[];
                end
            end

%===========================================================================
%           Method 16: G3SPD_Std 3rd Step -- Homoskedastic Covariances
%===========================================================================
            MuHat=single(rdosys(15).resi); %residuals from the 2nd Step estimation
```

```
            resi3rd=single(rdosys(16).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T)); %homoskedastic intra-equation variance
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(16).coef=[rdosys(16).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];

%========================================================================
%            Method 33: G3SPD_Std 3rd Step -- Heteroskedastic Covariances
%========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(1).vari11),diag(bshete(1).vari12);...
               diag(bshete(1).vari12),diag(bshete(1).vari22)];
```

```
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic↙
mations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)↙
are not estimated
            rdosys(33).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(33).resi=resi3rd;
            rdosys(33).stats=[];

%===========================================================================
%           Method 19: G3SPD_Corn 3rd Step -- Homoskedastic Covariances
%===========================================================================
            MuHat=single(rdosys(18).resi); %residuals from the 2nd Step estimation
            resi3rd=single(rdosys(19).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):↙
(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
```

```
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(19).coef=[rdosys(19).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];

%===========================================================================
%           Method 34: G3SPD_Corn 3rd Step -- Heteroskedastic Covariances
%===========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(2).vari11),diag(bshete(2).vari12);...
               diag(bshete(2).vari12),diag(bshete(2).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
```

```
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic↙
estimations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)↙
are not estimated
            rdosys(34).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(34).resi=resi3rd;
            rdosys(34).stats=[];

%=========================================================================
%           Method 21: G3SPD_HTM 3rd Step -- Homoskedastic Covariances
%=========================================================================
            MuHat=single(rdosys(20).resi); %residuals from the 2nd Step estimation
            resi3rd=single(rdosys(21).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):↙
(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of↙
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of↙
instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
```

```
            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V','sigmaGMM');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % replace existing estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
are not estimated
            rdosys(21).coef=[rdosys(21).coef(:,1),stderGMM,stderHC1,NaN(16,3),stderBK,
NaN(16,1)];

%===========================================================================
%           Method 35: G3SPD_HTM 3rd Step -- Heteroskedastic Covariances
%===========================================================================
            % place new estimation (pending)
            rdosys(35).coef=[];
            rdosys(35).resi=[];
            rdosys(35).stats=[];

%===========================================================================
%           Method 27,28: G3SPD_Std_alter 3rd Step -- Homoskedastic Cov.
%===========================================================================
            %generates pseudo-effects (stored in 27)
            thetaFEStd=[rdosys(2).coef(1:4,1);rdosys(2).coef(9:12,1)]; %(Y1,X1,X2,X3) &
(Y2,X4,X5)
            thetaBEStd=[rdosys(4).coef(5:7,1);rdosys(4).coef(13:15,1)]; %(Z1,Z2,Varphi1)
&(Z1,Z3,Varphi2)
            MuHat=Pmu*y-Pmu*Rwave*thetaFEStd-Rbar*thetaBEStd;
            rdoeq1(27).resi=MuHat;

            % consistent covariance estimator (from 2SLS)
```

```
            V=kron(bshomo(32).vari,eye(N*T));
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            clear('V');

            % Coefficients (stored in 28, column 1) and Residuals
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8) are not estimated
            rdosys(28).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(28).resi=resi3rd;
            rdosys(28).stats=[];
```

```
%=========================================================================
%           Method 36: G3SPD_Std_alter 3rd Step -- Heteroskedastic Cov.
%=========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(4).vari11),diag(bshete(4).vari12);...
               diag(bshete(4).vari12),diag(bshete(4).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic↙
mations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)↙
anot estimated
            rdosys(36).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(36).resi=resi3rd;
            rdosys(36).stats=[];

%=========================================================================
%           Method 29,30: G3SPD_Corn_alter 3rd Step -- Homoskedastic Cov.
%=========================================================================
            %generates pseudo-effects (stored in 29)
```

```
            thetaFEBal=[rdosys(3).coef(1:4,1);rdosys(3).coef(9:12,1)];%(Y1,X1,X2,X3) & (Y2,X1,X4,X5)
            thetaBEBal=[rdosys(5).coef(5:7,1);rdosys(5).coef(13:15,1)];%(Z1,Z2,Varphi1) &(Z1,Z3,Varphi2)
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEBal;
            rdoeq1(29).resi=MuHat;

            % consistent covariance estimator (from 2SLS)
            V=kron(bshomo(33).vari,eye(N*T));
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            clear('V');

            % Coefficients (stored in 30, column 1) and Residuals
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVC=single(Rfull'*C*inv(C'*sigmaGMM*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
```

```
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
not estimated
            rdosys(30).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(30).resi=resi3rd;
            rdosys(30).stats=[];

%==========================================================================
%           Method 37: G3SPD_HTM_alter 3rd Step -- Heteroskedastic Cov.
%==========================================================================
            % consistent covariance estimator (from 2SLS)
            V=[diag(bshete(5).vari11),diag(bshete(5).vari12);...
               diag(bshete(5).vari12),diag(bshete(5).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));

            % Coefficients, Residuals, and New Heteroskedastic Covariance
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            resi3rd=y-Rfull*theta;
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            bsnew.vari11=ENT(:,1).^2;
            bsnew.vari12=ENT(:,1).*ENT(:,2);
            bsnew.vari22=ENT(:,2).^2;
            V=[diag(bsnew.vari11),diag(bsnew.vari12);...
               diag(bsnew.vari12),diag(bsnew.vari22)];
            clear('bsnew');

            % OmegaGMM: stored in column 2
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            stderHC1=stderGMM; %HC1 and GMM are identical for heteroskedastic
equations

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-Kfull),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK','OmegaGMM','RCinvCVC');

            % place new estimation
            % Covariance matrixes HC2 (col 4), LE (col 5), HC3 (col 6), and MT (col 8)
not estimated
```

```
            rdosys(37).coef=[theta,stderGMM,stderHC1,NaN(16,3),stderBK,NaN(16,1)];
            rdosys(37).resi=resi3rd;
            rdosys(37).stats=[];

%===========================================================================
%           Method 31,32: G3SPD_HTM_alter 3rd Step -- Homoskedastic Cov.
%===========================================================================
            % new MuHat
            rdosys(31).resi=[];
            % place new estimation (pending)
            rdosys(32).coef=[];
            rdosys(32).resi=[];
            rdosys(32).stats=[];

%===========================================================================
%           Method 38: G3SPD_HTM_alter 3rd Step -- Heteroskedastic Cov.
%===========================================================================
            % place new estimation (pending)
            rdosys(38).coef=[];
            rdosys(38).resi=[];
            rdosys(38).stats=[];

%===========================================================================
%           Save results
%===========================================================================
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys','Rfull','C');
        end
        toc
    end
end
```

```
%=======================================================================
% Function ESTIMASYS_ALTER_V4
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%===========================================================================

function estimasys_alter_v4
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu
    %Redefine Pmu and Qmu for a system of 2 equations
    Pmu=blkdiag(Pmu,Pmu);
    Qmu=blkdiag(Qmu,Qmu);

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        %load variables
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);

        %Arrange regressors
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        clear('R1bar','R2bar');

        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
```

```
            disp(['                   Observation: ' num2str(obs)]);

            %load system estimation
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdosys');
            %load homoskedastic variance estimates
            filename=['.\varianza\vari_homo\vari_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f' )];
            load(filename,'bigsigma'); bshomo=bigsigma;
            clear bigsigma

            %arrange variables
            y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
            R1wave=single([datos(1,obs).Y(:,2) X1 X2 X3]); %Eq. 1 within regressors
            R2wave=single([datos(1,obs).Y(:,1) X1 X4 X5]); %Eq. 2 within regressors
            Rwave=single(blkdiag(R1wave,R2wave)); %system within regressors
            Kwave=single(size(Rwave,2));
            clear('R1wave','R2wave');

%==========================================================================
%           Method 30: G3SPD_Corn_alter 3rd Step -- Homoskedastic Cov.
%==========================================================================
            %generates pseudo-effects (stored in 29)
            thetaFEBal=[rdosys(3).coef(1:4,1);rdosys(3).coef(9:12,1)];%(Y1,X1,X2,X3) &(Y2,X4,X5)
            thetaBEBal=[rdosys(5).coef(5:7,1);rdosys(5).coef(13:15,1)];%(Z1,Z2,Varphi1) &(Z1,Z3,Varphi2)
            MuHat=Pmu*y-Pmu*Rwave*thetaFEBal-Rbar*thetaBEBal;
            rdosys(29).resi=MuHat;

            % obtain residuals
            resi1=squeeze(rdosys(30).resi);
            resi2=squeeze(rdosys(3).resi);
            resi3=(Pmu*Rwave*squeeze(thetaFEBal)+MuHat)/T;
            resi=[resi1,resi2,resi3]; %residuals (OLS 3rd step, Within, ~between)

            % obtain R'Cinv(C'VC)C' and OmegaGMM
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,ones(N*T,1),MuHat(1:(N*T),1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,ones(N*T,1),MuHat((N*T+1):(2*N*T),1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,MuHat((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
```

```
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            ENT=reshape(resi1,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            sigmaGMM=kron(V,eye(N*T));
            RCinvCVCC=single(Rfull'*C*inv(C'*sigmaGMM*C)*C');
            clear('V','C','ENT','sigmaGMM');
            OmegaGMM=inv(RCinvCVCC*Rfull);

            % Obtain SigmaGMM-Adjusted
            % (1) degrees of freedom
            dof1=N*T-Kfull;
            dof2=N*(T-1)-Kwave;
            dof3=(N*T-(Kfull-1));
            dof=[dof1,    dof1,    dof1;
                 dof1,    dof2,    dof2;
                 dof1,    dof2,    dof3]; %dof for one equatin
            dof=kron(dof,ones(G)); %dof for the whole system of eqs.
            % (2) new V
            resi=reshape(resi,N*T,3*G);
            newV=(resi'*resi)./dof;
            clear('resi1','resi2','resi3','resi','dof');
            % (3) obtain S
            QRwavebar=eye(G*N*T)-Pmu*Rwave*inv(Rwave'*Pmu*Rwave)*Rwave'*Pmu;
            S1=eye(G*N*T);
            S2=Pmu*Rwave*inv(Rwave'*Qmu*Rwave)*Rwave'*Qmu;
            S3=Rbar*inv(Rbar'*QRwavebar*Rbar)*Rbar'*QRwavebar;
            S=[S1 S2 S3];
            clear('S1','S2','S3','QRwavebar');
            % (4) obtain Sigma
            sigmaGMMAdj=S*kron(newV,eye(N*T))*S';
            OmegaGMMAdj=OmegaGMM * (RCinvCVCC*sigmaGMMAdj*RCinvCVCC') * OmegaGMM;
            clear('newV','S','sigmaAdjOLS');
            stdernew=diag(OmegaGMMAdj).^0.5; %new std. errors
            rdosys(30).coef(:,9)=stdernew; %store new std. errors in column 9

%==========================================================================
%           Save results
%==========================================================================
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys','Rfull','C','Rwave');
        end
        toc
    end
end
```

```
%=========================================================================
% Function ESTIMASYS_HETERO
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%=============================================================================

function estimasys_hetero
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        %load variables and estimated covariance matrixes
        clear('X1','X2','X3','X4','X5','Z1','Z2','Z3');
        filename=['.\base\base_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'X1','X2','X3','X4','X5','Z1','Z2','Z3','mu1','mu2','correl');
        filename=['.\case\case_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'datos');
        X1=single(X1); X2=single(X2); X3=single(X3); X4=single(X4); X5=single(X5);
        Z1=single(Z1); Z2=single(Z2); Z3=single(Z3);
        R1bar=single([Z1,Z2,ones(N*T,1)]); %eq. 1 between regressors
        R2bar=single([Z1,Z3,ones(N*T,1)]); %eq. 2 between regressors
        Rbar=single(blkdiag(R1bar,R2bar)); %system 'between' regressors
        Kbar=single(size(R1bar,2));
        clear('R1bar','R2bar');
        Bbar=single(kron(eye(G),Pmu*([X1,X2,X3,X4,X5,Z1,Z2,Z3,ones(N*T,1)]))); %between moments
        if exist(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')],'dir' )~=7
            mkdir(['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end
```

```
        for obs=1:1:1000  %1000 experiments for each combination of (nt,rho)
            disp(['                         Observation: ' num2str(obs)]);
            %load system estimation
            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdosys');
            %load variance estimates
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f') ...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f')];
            load(filename,'bigsigma');
            %load dependent variable
            y=single([datos(1,obs).Y(:,1);datos(1,obs).Y(:,2)]); %stacked y
            clear('rdo');

%===========================================================================
            %Method 15: G3SPD_Std 3rd Step
%===========================================================================
            % Covariance estimation asumming intra-eq. homoskedasticity
            %---------------------------------------------------------------
            resi2nd=single(rdosys(14).resi); %residuals from the 2nd Step estimation
            resi3rd=single(rdosys(15).resi); %residuals from the 3rd Step estimation
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,resi2nd(1:(N*T),1),ones(N*T,1)];
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,resi2nd((N*T+1):(2*N*T),1),ones(N*T,1)];
            Rfull=blkdiag(R1full,R2full); %set of regressors of the system
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,resi2nd((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2)); %set of instruments of the system
            clear('C1','C2');
            PR=Rfull*inv(Rfull'*Rfull)*Rfull'; %projection matrix of the regressors

            % OmegaGMM: stored in column 2
            ENT=reshape(resi3rd,N*T,G); %arrange the set of G vectors of residuals
            V=(ENT'*ENT)./(N*T-Kfull);
            RCinvCVC=single(Rfull'*C*inv(C'*(kron(V,eye(N*T)))*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');

            % WH-HC1: stored in column 3
            sigmaHC1=C'*diag(resi3rd.^2)*C;
            OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
            stderHC1=diag(OmegaHC1).^0.5;
            clear('sigmaHC1','OmegaHC1');
```

```
            % WH-HC2: stored in column 4
            sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
            OmegaHC2=OmegaGMM * (RCinvCVC*sigmaHC2*RCinvCVC') * OmegaGMM;
            stderHC2=diag(OmegaHC2).^0.5;
            clear('sigmaHC2','OmegaHC2');

            % WH-LE: stored in column 5
            resi3rdmodi=(resi3rd)./(1-diag(PR)); %epsilon**
            sigmaLE=C'*diag(resi3rdmodi.^2)*C;
            OmegaLE=OmegaGMM * (RCinvCVC*sigmaLE*RCinvCVC') * OmegaGMM;
            stderLE=diag(OmegaLE).^0.5;
            clear('OmegaLE');

            % WH-HC3: stored in column 6
            sigmaHC3=sigmaLE-(C'*resi3rdmodi*resi3rdmodi'*C);
            OmegaHC3=OmegaGMM * (RCinvCVC*sigmaHC3*RCinvCVC') * OmegaGMM;
            stderHC3=diag(OmegaHC3).^0.5;
            clear('sigmaLE','resi3rdmodi','sigmaHC3','OmegaHC3');

            % BK: stored in column 7
            VBK=[];
            for g=1:1:G
                ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
                VBK=blkdiag(VBK,kron((ET'*ET)./(T-K),eye(T)));
            end
            sigmaBK=C'*VBK*C;
            OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
            stderBK=diag(OmegaBK).^0.5;
            clear('ET','ENT','VBK','sigmaBK','OmegaBK');

            % MT: stored in column 8
            ENT2nd=reshape(resi2nd,N*T,G); %arrange the set of G vectors of 2nd Step↙
estimation
            RBinvBVB=single(Rbar'*Bbar*inv(Bbar'*(kron((ENT2nd'*Pmu*ENT2nd)./(N-Kbar),↙
Pmu)*Bbar));
            OmegaGMM2nd=inv(RBinvBVB*Bbar'*Rbar);
            CEEB=C'*resi3rd*resi3rd'*Bbar;
            CEHB=C'*resi3rd*resi2nd'*Bbar;
            clear('Rbar','Kbar','Bbar','ENT2nd');

            Psi1=OmegaGMM; %OmegaGMM 3rd Step
            Psi2=(RCinvCVC*CEEB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
            Psi3=(RCinvCVC*CEHB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
            OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
            stderMT=diag(OmegaMT).^0.5;
            clear('RBinvBVB','OmegaGMM2nd','CEEB','CEHB','Psi1','Psi2',...
                'Psi3','RCinvCVC','OmegaGMM','OmegaMT');

            % replace existing estimation
            stdernew=[rdosys(15).coef(:,1),stderGMM,stderHC1,stderHC2,stderLE,stderHC3,↙
stderBK,stderMT];
```

```
            rdosys(15).coef(:)=[NaN(1,8);           %y1: empty    EQ. 1
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,8);            %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,8);            %Z3: empty
                                stdernew(8,:);    %phi1
                                NaN(1,8);            %phi2: empty
                                stdernew(7,:);    %mu1
                                NaN(1,8);            %mu2: empty
                                stdernew(9,:);    %y1            EQ. 2
                                NaN(1,8);            %y2: empty
                                stdernew(10,:);   %X1
                                NaN(2,8);            %X2,X3: empty
                                stdernew(11,:);   %X4
                                stdernew(12,:);   %X5
                                stdernew(13,:);   %Z1
                                NaN(1,8);            %Z2: empty
                                stdernew(14,:);   %Z3
                                NaN(1,8);            %phi1
                                stdernew(16,:);   %phi2
                                NaN(1,8);            %mu1: empty
                                stdernew(15,:)];  %mu2

%==========================================================================
%           Method 25: G3SPD_Std 3rd Step: fully heteroskedastic
%==========================================================================
            % Coefficients and Covariance estimation asumming intra-eq. heteroskedasticity
            % -------------------------------------------------------------------------
            V=[diag(bigsigma(1).vari11),diag(bigsigma(1).vari12);...
               diag(bigsigma(1).vari12),diag(bigsigma(1).vari22)]; %consistent covariance estimator (from 2SLS)
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            theta=inv(RCinvCVC*C'*Rfull) * (RCinvCVC*C'*y);
            % residuals
            resi3rd=y-Rfull*theta;

            % OmegaGMM: stored in column 2
            bigsigmanew(1).vari11=resi3rd(:,1).^2;
            bigsigmanew(1).vari12=resi3rd(:,1).*resi3rd(:,2);
            bigsigmanew(1).vari22=resi3rd(:,2).^2;
            V=[diag(bigsigmanew(1).vari11),diag(bigsigmanew(1).vari12);...
               diag(bigsigmanew(1).vari12),diag(bigsigmanew(1).vari22)];
            RCinvCVC=single(Rfull'*C*inv(C'*V*C));
            OmegaGMM=inv(RCinvCVC*C'*Rfull);
            stderGMM=diag(OmegaGMM).^0.5; %std. errors
            clear('V');
```

```
        % WH-HC1: stored in column 3
        sigmaHC1=C'*diag(resi3rd.^2)*C;
        OmegaHC1=OmegaGMM * (RCinvCVC*sigmaHC1*RCinvCVC') * OmegaGMM;
        stderHC1=diag(OmegaHC1).^0.5;
        clear('sigmaHC1','OmegaHC1');

        % WH-HC2: stored in column 4
        sigmaHC2=C'*diag((resi3rd.^2)./(1-diag(PR)))*C;
        OmegaHC2=OmegaGMM * (RCinvCVC*sigmaHC2*RCinvCVC') * OmegaGMM;
        stderHC2=diag(OmegaHC2).^0.5;
        clear('sigmaHC2','OmegaHC2');

        % WH-LE: stored in column 5
        resi3rdmodi=(resi3rd)./(1-diag(PR)); %epsilon**
        sigmaLE=C'*diag(resi3rdmodi.^2)*C;
        OmegaLE=OmegaGMM * (RCinvCVC*sigmaLE*RCinvCVC') * OmegaGMM;
        stderLE=diag(OmegaLE).^0.5;
        clear('OmegaLE');

        % WH-HC3: stored in column 6
        sigmaHC3=sigmaLE-(C'*resi3rdmodi*resi3rdmodi'*C);
        OmegaHC3=OmegaGMM * (RCinvCVC*sigmaHC3*RCinvCVC') * OmegaGMM;
        stderHC3=diag(OmegaHC3).^0.5;
        clear('sigmaLE','resi3rdmodi','sigmaHC3','OmegaHC3');

        % BK: stored in column 7
        VBK=[];
        for g=1:1:G
            ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
            VBK=blkdiag(VBK,kron((ET'*ET)./(T-K),eye(T)));
        end
        sigmaBK=C'*VBK*C;
        OmegaBK=OmegaGMM * (RCinvCVC*sigmaBK*RCinvCVC') * OmegaGMM;
        stderBK=diag(OmegaBK).^0.5;
        clear('ET','ENT','VBK','sigmaBK','OmegaBK');

        % MT: stored in column 8
        ENT2nd=reshape(resi2nd,N*T,G); %arrange the set of G vectors of 2nd Step↙
mation
        RBinvBVB=single(Rbar'*Bbar*inv(Bbar'*(kron((ENT2nd'*Pmu*ENT2nd)./(N-Kbar),↙
Pmu)*Bbar));
        OmegaGMM2nd=inv(RBinvBVB*Bbar'*Rbar);
        CEEB=C'*resi3rd*resi3rd'*Bbar;
        CEHB=C'*resi3rd*resi2nd'*Bbar;
        clear('Rbar','Kbar','Bbar','ENT2nd');

        Psi1=OmegaGMM; %OmegaGMM 3rd Step
        Psi2=(RCinvCVC*CEEB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
        Psi3=(RCinvCVC*CEHB*RBinvBVB')*OmegaGMM2nd*(RBinvBVB*CEEB'*RCinvCVC');
        OmegaMT=Psi1+Psi1*(Psi2-2*Psi3)*Psi1;
        stderMT=diag(OmegaMT).^0.5;
```

```
            clear('RBinvBVB','OmegaGMM2nd','CEEB','CEHB','Psi1','Psi2',...
                'Psi3','RCinvCVC','OmegaGMM','OmegaMT');

            % replace existing estimation
            stdernew=[rdosys(15).coef(:,1),stderGMM,stderHC1,stderHC2,stderLE,stderHC3,↙
sBK,stderMT];
            rdosys(25).coef(:)=[NaN(1,8);           %y1: empty    EQ. 1
                                stdernew(1,:);    %y2
                                stdernew(2,:);    %X1
                                stdernew(3,:);    %X2
                                stdernew(4,:);    %X3
                                NaN(2,8);            %X4,X5: empty
                                stdernew(5,:);    %Z1
                                stdernew(6,:);    %Z2
                                NaN(1,8);            %Z3: empty
                                stdernew(8,:);    %phi1
                                NaN(1,8);            %phi2: empty
                                stdernew(7,:);    %mu1
                                NaN(1,8);            %mu2: empty
                                stdernew(9,:);    %y1           EQ. 2
                                NaN(1,8);            %y2: empty
                                stdernew(10,:);   %X1
                                NaN(2,8);            %X2,X3: empty
                                stdernew(11,:);   %X4
                                stdernew(12,:);   %X5
                                stdernew(13,:);   %Z1
                                NaN(1,8);            %Z2: empty
                                stdernew(14,:);   %Z3
                                NaN(1,8);            %phi1
                                stdernew(16,:);   %phi2
                                NaN(1,8);            %mu1: empty
                                stdernew(15,:)];  %mu2
            rdosys(25).resi=resi3rd;
            rdosys(25).stats=[];

%=========================================================================
            %Method 16,17,18: G3SPD_FECorn
%=========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(17).resi(1:(N*T),1),ones↙
(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed↙
Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(17).resi((N*T+1):(2*N*T),↙
1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated↙
Fixed Effects
            Rfull=blkdiag(R1full,R2full);
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi(1:(N*T),1),ones(N*T,1)]; %set of↙
instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(17).resi((N*T+1):(2*N*T),1),ones(N*T,1)];↙
%set of instruments
```

```
            C=single(blkdiag(C1,C2));
            clear('C1','C2');
            [rdo.coef,rdo.resi,rdo.stats]=G3SLSM_hetero(Rfull,bigsigma(2).vari,C,y,G,N,T,Kfull);
            rdosys(18).coef(:)=[NaN(1,6);           %y1: empty    EQ. 1
                             rdo(3).coef(1,:);    %y2
                             rdo(3).coef(2,:);    %X1
                             rdo(3).coef(3,:);    %X2
                             rdo(3).coef(4,:);    %X3
                             NaN(2,6);            %X4,X5: empty
                             rdo(3).coef(5,:);    %Z1
                             rdo(3).coef(6,:);    %Z2
                             NaN(1,6);            %Z3: empty
                             rdo(3).coef(8,:);    %phi1
                             NaN(1,6);            %phi2: empty
                             rdo(3).coef(7,:);    %mu1
                             NaN(1,6);            %mu2: empty
                             rdo(3).coef(9,:);    %y1          EQ. 2
                             NaN(1,6);            %y2: empty
                             rdo(3).coef(10,:);   %X1
                             NaN(2,6);            %X2,X3: empty
                             rdo(3).coef(11,:);   %X4
                             rdo(3).coef(12,:);   %X5
                             rdo(3).coef(13,:);   %Z1
                             NaN(1,6);            %Z2: empty
                             rdo(3).coef(14,:);   %Z3
                             NaN(1,6);            %phi1
                             rdo(3).coef(16,:);   %phi2
                             NaN(1,6);            %mu1: empty
                             rdo(3).coef(15,:)];  %mu2
            rdosys(18).resi=rdo(3).resi;
            rdosys(18).stats=rdo(3).stats;

%==========================================================================
            %Method 19,20: G3SPD_HTHT
%==========================================================================
            % 3rd stage
            R1full=[datos(1,obs).Y(:,2),X1,X2,X3,Z1,Z2,rdosys(19).resi(1:(N*T),1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed Effects
            R2full=[datos(1,obs).Y(:,1),X1,X4,X5,Z1,Z3,rdosys(19).resi((N*T+1):(2*N*T),1),ones(N*T,1)];     %full dataset, including all the original vars. plus the estimated Fixed Effects
            Rfull=blkdiag(R1full,R2full);
            Kfull=size(R1full,2);
            clear('R1full','R2full');
            C1=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi(1:(N*T),1),ones(N*T,1)]; %set of instruments
            C2=[X1,X2,X3,X4,X5,Z1,Z2,Z3,rdosys(19).resi((N*T+1):(2*N*T),1),ones(N*T,1)]; %set of instruments
            C=single(blkdiag(C1,C2));
            clear('C1','C2');
```

```
            [rdo.coef,rdo.resi,rdo.stats]=G3SLSM_hetero(Rfull,bigsigma(3).vari,C,y,G,N,↙
T,K);
            rdosys(20).coef(:,)=[NaN(1,6);           %y1: empty    EQ. 1
                            rdo(3).coef(1,:);    %y2
                            rdo(3).coef(2,:);    %X1
                            rdo(3).coef(3,:);    %X2
                            rdo(3).coef(4,:);    %X3
                            NaN(2,6);            %X4,X5: empty
                            rdo(3).coef(5,:);    %Z1
                            rdo(3).coef(6,:);    %Z2
                            NaN(1,6);            %Z3: empty
                            rdo(3).coef(8,:);    %phi1
                            NaN(1,6);            %phi2: empty
                            rdo(3).coef(7,:);    %mu1
                            NaN(1,6);            %mu2: empty
                            rdo(3).coef(9,:);    %y1           EQ. 2
                            NaN(1,6);            %y2: empty
                            rdo(3).coef(10,:);    %X1
                            NaN(2,6);            %X2,X3: empty
                            rdo(3).coef(11,:);   %X4
                            rdo(3).coef(12,:);   %X5
                            rdo(3).coef(13,:);   %Z1
                            NaN(1,6);            %Z2: empty
                            rdo(3).coef(14,:);   %Z3
                            NaN(1,6);            %phi1
                            rdo(3).coef(16,:);   %phi2
                            NaN(1,6);            %mu1: empty
                            rdo(3).coef(15,:)];  %mu2
            rdosys(20).resi=rdo(3).resi;
            rdosys(20).stats=rdo(3).stats;

            %Save results
            filename=['.\sistema\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            save(filename,'rdosys');
            disp([filename ' guardado. Continuar...']);
            clear('rdosys','Rfull','C');
        end
        toc
    end
end
```

```
%=====================================================================
% Function ESTIMATEVARI_ALTER
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%=====================================================================

function estimateVARI_alter
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        disp(' ')
        disp( '        Estimating covariance matrixes...')

        for obs=1:1:1000 % 1000 cases for each combination of nt and rho
            disp(['                   Observation: ' num2str(obs)]);

            %load residuals
            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq1');
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');


%=====================================================================
%            Estimation of Homoskedastic Variance
%=====================================================================
            %retrieve estimated variances
            filename=['.\varianza\vari_homo\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
```

```
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f')];
            load(filename,'bigsigma');

            % G3SPD_Std_alter 3rd Step (method 28, stored in section 32)
            resi1=single([rdoeq1(28).resi,rdoeq2(28).resi]);
            bigsigma(32).vari=resi1'*resi1/(N*T);

            % G3SPD_Corn_alter 3rd Step (method 30, stored in section 33)
            resi2=single([rdoeq1(30).resi,rdoeq2(30).resi]);
            bigsigma(33).vari=resi2'*resi2/(N*T);

            % G3SPD_HTM_alter 3rd Step (method 32, stored in section 34)
            resi3=single([rdoeq1(32).resi,rdoeq2(32).resi]);
            bigsigma(34).vari=resi3'*resi3/(N*T);

            %save estimated variances
            filename=['.\varianza\vari_homo\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho,'%3.0f')...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f')];
            save(filename,'bigsigma');
            disp([filename ' guardado. Continuar...']);
            clear('bigsigma');


%===========================================================================
%           Estimation of Heteroskedastic Variance
%===========================================================================
            %retrieve estimated variances
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho,'%3.0f')...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f')];
            load(filename,'bigsigma');

            % G3SPD_Std_alter 3rd Step (method 28, stored in section 4)
            bigsigma(4).vari11=resi1(:,1).^2;
            bigsigma(4).vari12=resi1(:,1).*resi1(:,2);
            bigsigma(4).vari22=resi1(:,2).^2;


            % G3SPD_Corn_alter 3rd Step(method 30, stored in section 5)
            bigsigma(5).vari11=resi2(:,1).^2;
            bigsigma(5).vari12=resi2(:,1).*resi2(:,2);
            bigsigma(5).vari22=resi2(:,2).^2;

            % G3SPD_HTM_alter (method 32, stored in section 6)
            bigsigma(6).vari11=resi3(:,1).^2;
            bigsigma(6).vari12=resi3(:,1).*resi3(:,2);
            bigsigma(6).vari22=resi3(:,2).^2;
```

```
            % save estimated covariance matrixes
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%4.0f )];
            save(filename,'bigsigma');
            disp([filename ' guardado.']);
            clear('bigsigma');
        end

        toc
    end
end
```

```
%=====================================================================
% Function ESTIMATEVARI_HETERO
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%=========================================================================

function estimateVARI_hetero
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        disp(' ')
        disp( '        Estimating covariance matrixes...')

        for obs=1:1:1000 % 1000 cases for each combination of nt and rho
            disp(['                   Observation: ' num2str(obs)]);

            %load residuals
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq1');
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs '%4.0f')];
            load(filename,'rdoeq2');

            % FEVD_FEStd
            % 3rd Stage (1)
            resi=single([rdoeq1(15).resi,rdoeq2(15).resi]);
            bigsigma(1).vari11=resi(:,1).^2;
            bigsigma(1).vari12=resi(:,1).*resi(:,2);
```

```
            bigsigma(1).vari22=resi(:,2).^2;

            % FEVD_FECorn
            % 3rd Stage (2)
            resi=single([rdoeq1(18).resi,rdoeq2(18).resi]);
            bigsigma(2).vari11=resi(:,1).^2;
            bigsigma(2).vari12=resi(:,1).*resi(:,2);
            bigsigma(2).vari22=resi(:,2).^2;

            % FEVD_HTHT
            % 3rd Stage (3)
            resi=single([rdoeq1(20).resi,rdoeq2(20).resi]);
            bigsigma(3).vari11=resi(:,1).^2;
            bigsigma(3).vari12=resi(:,1).*resi(:,2);
            bigsigma(3).vari22=resi(:,2).^2;

            % save estimated covariance matrixes
            if exist(['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')],'dir')~=7
                mkdir(['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')]);
            end
            filename=['.\varianza\vari_hetero\vari_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f )];
            save(filename,'bigsigma');
            disp([filename ' guardado.']);
            clear('bigsigma');
        end

        toc
    end
end
```

```
%=======================================================================
% Function ESTIMATEVARI_HOMO
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%===========================================================================

function estimateVARI_homo
clc
clear
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2;

for nt=1:1:size(ntcases,1) % 9 combinations of N and T
    % auxiliary matrixes
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '========================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '========================================================');
    ZMu=single(kron(eye(N),ones(T,1)));
    Pmu=single(ZMu*inv(ZMu'*ZMu)*ZMu');
    Qmu=single(eye(N*T)-Pmu);
    clear ZMu

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic
        disp(' ')
        disp( '        Estimating covariance matrixes...')
        if exist(['.\varianza\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0
f')],'dir' )~=7
            mkdir(['.\varianza\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')]);
        end

        % estimate covariance matrixes
        for obs=1:1:1000 % 1000 cases for each combination of nt and rho
            disp(['                  Observation: ' num2str(obs)]);

            %load residuals
            filename=['.\eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            load(filename,'rdoeq1');
            filename=['.\eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str
(obs '%4.0f')];
            load(filename,'rdoeq2');
```

```
% OLS (1)
resi=single([rdoeq1(1).resi,rdoeq2(1).resi]);
bigsigma(1).vari=resi'*resi/(N*T);

% FEStd (2)
resi=single([rdoeq1(2).resi,rdoeq2(2).resi]);
bigsigma(2).vari=resi'*Qmu*resi/(N*(T-1));

% FEBal (3)
resi=single([rdoeq1(3).resi,rdoeq2(3).resi]);
bigsigma(3).vari=resi'*Qmu*resi/(N*(T-1));

% BEStd (4)
resi=single([rdoeq1(4).resi,rdoeq2(4).resi]);
bigsigma(4).vari=resi'*Pmu*resi/(N);

% BEBal (5)
resi=single([rdoeq1(5).resi,rdoeq2(5).resi]);
bigsigma(5).vari=resi'*Pmu*resi/(N);

% ECWH (6)
resi=single([rdoeq1(6).resi,rdoeq2(6).resi]);
bigsigma(6).vari=resi'*Qmu*resi/(N*(T-1)); %within variance
bigsigma(7).vari=resi'*Pmu*resi/(N); %between variance

% ECAm (7)
resi=single([rdoeq1(7).resi,rdoeq2(7).resi]);
bigsigma(8).vari=resi'*Qmu*resi/(N*(T-1)); %within variance
bigsigma(9).vari=resi'*Pmu*resi/(N); %between variance

% ECSA (8)
resi=single([rdoeq1(8).resi,rdoeq2(8).resi]);
bigsigma(10).vari=resi'*Qmu*resi/(N*(T-1)); %within variance
bigsigma(11).vari=resi'*Pmu*resi/(N); %between variance

% ECCorn (9)
resi=single([rdoeq1(9).resi,rdoeq2(9).resi]);
bigsigma(12).vari=resi'*Qmu*resi/(N*(T-1));
bigsigma(13).vari=resi'*Pmu*resi/(N);

% HTHT (10)
resi=single([rdoeq1(10).resi,rdoeq2(10).resi]);
bigsigma(14).vari=resi'*Qmu*resi/(N*(T-1));
bigsigma(15).vari=resi'*Pmu*resi/(N);

% HTAM (11)
resi=single([rdoeq1(11).resi,rdoeq2(11).resi]);
if isempty(resi)
    bigsigma(16).vari=[];
    bigsigma(17).vari=[];
else
    bigsigma(16).vari=resi'*Qmu*resi/(N*(T-1));
```

```
            bigsigma(17).vari=resi'*Pmu*resi/(N);
        end

        % HTBMS (12)
        resi=single([rdoeq1(12).resi,rdoeq2(12).resi]);
        if isempty(resi)
            bigsigma(18).vari=[];
            bigsigma(19).vari=[];
        else
            bigsigma(18).vari=resi'*Qmu*resi/(N*(T-1));
            bigsigma(19).vari=resi'*Pmu*resi/(N);
        end

        % G3SPD_FEStd
        % 1st Stage (13)
        resi=single([rdoeq1(13).resi,rdoeq2(13).resi]);
        bigsigma(20).vari=resi'*Qmu*resi/(N*(T-1));
        % 2nd Stage (14)
        resi=single([rdoeq1(14).resi,rdoeq2(14).resi]);
        bigsigma(21).vari=resi'*Pmu*resi/(N);
        % 3rd Stage (15)
        resi=single([rdoeq1(15).resi,rdoeq2(15).resi]);
        bigsigma(22).vari=resi'*resi/(N*T);

        % G3SPD_FECorn
        % 1st Stage (16)
        resi=single([rdoeq1(16).resi,rdoeq2(16).resi]);
        bigsigma(23).vari=resi'*Qmu*resi/(N*(T-1));
        % 2nd Stage (17)
        resi=single([rdoeq1(17).resi,rdoeq2(17).resi]);
        bigsigma(24).vari=resi'*Pmu*resi/(N);
        % 3rd Stage (18)
        resi=single([rdoeq1(18).resi,rdoeq2(18).resi]);
        bigsigma(25).vari=resi'*resi/(N*T);

        % G3SPD_HTHT
        % 2nd Stage (19)
        resi=single([rdoeq1(19).resi,rdoeq2(19).resi]);
        bigsigma(26).vari=resi'*Pmu*resi/(N);
        % 3rd Stage (20)
        resi=single([rdoeq1(20).resi,rdoeq2(20).resi]);
        bigsigma(27).vari=resi'*resi/(N*T);

        % G3SPD_HTAM
        % 2nd Stage (21)
        resi=single([rdoeq1(21).resi,rdoeq2(21).resi]);
        if isempty(resi)
            bigsigma(28).vari=[];
        else
            bigsigma(28).vari=resi'*Pmu*resi/(N);
        end
        % 3rd Stage (22)
```

```
            resi=single([rdoeq1(22).resi,rdoeq2(22).resi]);
            if isempty(resi)
                bigsigma(29).vari=[];
            else
                bigsigma(29).vari=resi'*resi/(N*T);
            end

            % G3SPD_HTBMS
            % 2nd Stage (23)
            resi=single([rdoeq1(23).resi,rdoeq2(23).resi]);
            if isempty(resi)
                bigsigma(30).vari=[];
            else
                bigsigma(30).vari=resi'*Pmu*resi/(N);
            end
            % 3rd Stage (24)
            resi=single([rdoeq1(24).resi,rdoeq2(24).resi]);
            if isempty(resi)
                bigsigma(31).vari=[];
            else
                bigsigma(31).vari=resi'*resi/(N*T);
            end


            % save estimated covariance matrixes
            filename=['.\varianza\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')...
                '\vari_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str(obs,'%↙
4.0f')];
            save(filename,'bigsigma');
            disp([filename ' guardado. Continuar...']);
            clear('bigsigma');
        end

        toc
    end
end
```

```
%-------------------------------------------------------------------
% Function SA
% Input:
%       i: case number (1...192)
%       exper: experiment number (1...1000)
%-----------------------------------------------------------------------

function [S]=SA(s,N,T) %generates Amemiya-MaCurdy special matrix of instruments
S=zeros(N*T,1);
for j=1:1:size(s,2)
    sa=kron(reshape(s(:,j),T,N)',ones(T,1));
    S=[S,sa];
end
S=S(:,2:end);
```

```
%----------------------------------------------------------------------
% Function G2SLS
%       Estimates Generalized 2SLS.
%
% Input:
%       R: set of covariates
%       W: omega ^ (-1/2)
%       P: projection matrix of instruments
%       y: dependent variable
%       N: number of individuals
%       T: number of time periods
%
% Output:
%       theta: coefficients
%       stder: standard errors of coefficients
%       resi: residuals
%       stats: structure containing:
%           .r2: R-squared
%           .r2adj: adjusted R-squared
%           .witvar: sigma1-squared ('within' variance)
%           .betvar: sigma2-squared ('between' variance)
%           .aic: Akaike Information Criterion
%           .bic: Bayesian (Schwarz) Information Criterion
%
%--------------------------------------------------------------------------

function [coef,resi,stats]=G2SLS(R,W,P,y,N,T,dof) %perform regression and returns
coefficients, residuals, and some statistics
% auxiliary matrixes
ZMu=kron(eye(N),ones(T,1));
Pmu=ZMu*inv(ZMu'*ZMu)*ZMu';
Qmu=eye(N*T)-Pmu;

% thetas
theta=inv(R'*W'*P*W*R)*R'*W'*P*W*y;

% residuals
resi=W*y-W*R*theta;

% standard errors
K=size(R,2);
vari=((resi'*resi)/(N*T-K));
stder=diag(vari*inv(R'*W'*W*R)).^0.5; %std. errors

% statistics
SST=(W*y-mean(W*y))'*(W*y-mean(W*y)); %Total Sum of Squares
SSR=resi'*resi; %Total Sum of Residuals
stats.r2=1-SSR/SST; % Rsq
stats.r2adj=1-((SSR/(N*T-K))/(SST/(N*T-1))); % Rsq
if T==1; dof=N-1; else dof=N*(T-1); end
stats.witvar=(resi'*Qmu*resi)/dof; %"within" variance
stats.betvar=(resi'*Pmu*resi)/N; %"between" variance
```

```
stats.aic=log(SSR/(N*T))+(2*(K+1)/(N*T));     %Akaike Information Criterion
stats.bic=log(SSR/(N*T))+((K+1)*log(N*T)/(N*T)); %Bayesian (Schwartz) Information
Criterion
coef=[theta,stder];
```

```
%-----------------------------------------------------------------------
% Function G2SLSM
%       Estimates Generalized 2SLS with multiple corrections of the
%       covariance matrix (no-correction, White-Huber, and PCSEs)
%
% Input:
%       R: set of covariates
%       W: omega ^ (-1/2)
%       P: projection matrix of instruments
%       y: dependent variable
%       N: number of individuals
%       T: number of time periods
%
% Output:
%       theta: coefficients
%       stder: standard errors of coefficients
%       resi: residuals
%       stats: structure containing:
%           .r2: R-squared
%           .r2adj: adjusted R-squared
%           .witvar: sigma1-squared ('within' variance)
%           .betvar: sigma2-squared ('between' variance)
%           .aic: Akaike Information Criterion
%           .bic: Bayesian (Schwarz) Information Criterion
%
%---------------------------------------------------------------------------

function [coef,resi,stats]=G2SLSM(R,W,P,y,N,T,dof) %perform regression and returns
coefficients, residuals, and some statistics
% auxiliary matrixes
ZMu=kron(eye(N),ones(T,1));
Pmu=ZMu*inv(ZMu'*ZMu)*ZMu';
Qmu=eye(N*T)-Pmu;

% thetas
theta=inv(R'*W'*P*W*R)*R'*W'*P*W*y;

% residuals
resi=W*y-W*R*theta;

% Omega 1: identity matrix (OLS)
K=size(R,2);
VCVI=((resi'*resi)/dof);
stderI=diag(VCVI*inv(R'*W'*W*R)).^0.5; %std. errors

% Omega 2: OLS with White-Huber VCV matrix (White, 1980)
VCVW=diag(resi.^2)./dof;  % VCVW makes computing White's covariance matrix easy
VCV_White=(N*T-K)*inv(R'*P*R) * (R'*P'*VCVW*P*R) * inv(R'*P*R); % White's VCV
stderW=diag(VCV_White).^0.5;

% Omega 3: OLS with PCSEs (Beck & Katz, 1995)
% if (T<N)    %if T<N then Parks' correction and PCSEs can't be
```

```
%   return   % calculated, or if T<=K then the model is underidentified
% end        % and we can estimate single equations for each individual
E=reshape(resi,T,N); %arrange the set of OLS residuals
VCVBK=kron((E'*E)./(T-K),eye(T)); %Beck & Katz (1995) covariance matrix for each time
period
VCV_BK=inv(R'*P*R) * (R'*P*VCVBK*P*R) * inv(R'*P*R); % Beck & Katz (1995) PC Covariance
Matrix (eq. 5 & endnote 15).
stderBK=diag(VCV_BK).^0.5;

% statistics
SST=(W*y-mean(W*y))'*(W*y-mean(W*y)); %Total Sum of Squares
SSR=resi'*resi; %Total Sum of Residuals
stats.r2=1-SSR/SST; % Rsq
stats.r2adj=1-((SSR/dof)/(SST/(dof+K-1))); % Rsq
if T==1; dof2=N-1; else dof2=N*(T-1); end
stats.witvar=(resi'*Qmu*resi)/dof2; %"within" variance
stats.betvar=(resi'*Pmu*resi)/N; %"between" variance
stats.aic=log(SSR/(N*T))+(2*(K+1)/(N*T)); %Akaike Information Criterion
stats.bic=log(SSR/(N*T))+((K+1)*log(N*T)/(N*T)); %Bayesian (Schwartz) Information
Criterion
coef=[theta,stderI,stderW,stderBK];
```

```
%------------------------------------------------------------------------
% Function G3SLS
%       Estimates GMM 3SLS.
%
% Input:
%       R: set of covariates
%       W: omega ^ (-1/2) (includes Pmu and Qmu)
%       V: Omega (GMM-3SLS weighting matrix)
%       A: matrix of instruments
%       y: dependent variable
%       N: number of individuals
%       T: number of time periods
%
% Output:
%       theta: coefficients
%       stder: standard errors of coefficients
%       resi: residuals
%       stats: structure containing:
%           .r2: R-squared
%           .r2adj: adjusted R-squared
%           .witvar: sigma1-squared ('within' variance)
%           .betvar: sigma2-squared ('between' variance)
%           .aic: Akaike Information Criterion
%           .bic: Bayesian (Schwarz) Information Criterion
%
%----------------------------------------------------------------------------

function [coef,resi,stats]=G3SLS2(R,W,V,A,y,G,N,T) %perform regression and returns
coefficients, residuals, and some statistics
% auxiliary matrixes
ZMu=kron(eye(N),ones(T,1));
Pmu=ZMu*inv(ZMu'*ZMu)*ZMu';
Qmu=eye(N*T)-Pmu;
clear ZMu
%W=kron(eye(G),W);
W=[W,zeros(size(W,1),size(W,2));zeros(size(W,1),size(W,2)),W];

% thetas
instr2SLS=A*inv(A'*V*A)*A';
inv2SLS=inv(R'*W*instr2SLS*W*R);
theta=inv2SLS * (R'*W*instr2SLS*W*y);

% residuals
resi=W*y-W*R*theta;

% standard errors
stder=diag(inv2SLS).^0.5; %std. errors

% statistics for each equation
if T==1; dof=N-1; else dof=N*(T-1); end
K=size(R,2)/G; %assuming all equations have the same number of regressors!!!
sysresi=reshape(resi,N*T,G);
```

```
for g=1:1:G
    suby=y((g-1)*N*T+1:g*N*T,1);
    subresi=sysresi(:,g);
    SST=(suby-mean(suby))'*(suby-mean(suby)); %Total Sum of Squares
    SSR=subresi'*subresi; %Total Sum of Residuals
    stats(g).r2=1-SSR/SST; % Rsq
    stats(g).r2adj=1-((SSR/(N*T-K))/(SST/(N*T-1))); % Rsq
    stats(g).witvar=(subresi'*Qmu*subresi)/dof; %"within" variance
    stats(g).betvar=(subresi'*Pmu*subresi)/N; %"between" variance
    stats(g).aic=log(SSR/(N*T))+(2*(K+1)/(N*T)); %Akaike Information Criterion
    stats(g).bic=log(SSR/(N*T))+((K+1)*log(N*T)/(N*T)); %Bayesian (Schwartz) Information Criterion
end
coef=[theta,stder];


% feature('memstats')
% whos


%McElroy's R^2 (McElroy, 1977) and Akaike IC for SEM
sigmachico=single(sysresi'*sysresi/(N*T));
sigma2=[sigmachico(1,1)*Qmu,sigmachico(1,2)*Qmu;sigmachico(2,1)*Qmu,sigmachico(2,2)*Qmu]
stats(G+1).r2=1-(resi'*V*resi)/(y'*sigma2*y);
stats(G+1).r2adj=1-((resi'*V*resi)/(G*N*T-K*G))/((y'*sigma2*y)/(G*N*T-G));
stats(G+1).aic=(G*N*T)*log(det(sysresi'*sysresi/(N*T)))+2*G*K+G*(G+1);
d=G*N*T/(G*N*T-(G+K+1));
stats(G+1).bic=(G*N*T)*log(det(sysresi'*sysresi/(N*T)))+2*d*(G*K+0.5*G*(G+1));   %instead of BIC, for the system we estimate AICC
```

```
%------------------------------------------------------------------------
% Function G3SLS
%       Estimates GMM 3SLS.
%
% Input:
%       R: set of covariates
%       W: omega ^ (-1/2) (includes Qmu and Pmu)
%       V: omega (GMM-3SLS weighting matrix)
%       A: matrix of instruments
%       y: dependent variable
%       N: number of individuals
%       T: number of time periods
%
% Output:
%       theta: coefficients
%       stder: standard errors of coefficients
%       resi: residuals
%       stats: structure containing:
%           .r2: R-squared
%           .r2adj: adjusted R-squared
%           .witvar: sigma1-squared ('within' variance)
%           .betvar: sigma2-squared ('between' variance)
%           .aic: Akaike Information Criterion
%           .bic: Bayesian (Schwarz) Information Criterion
%
%----------------------------------------------------------------------------

function [coef,resi,stats]=G3SLSM(R,V,A,y,G,N,T,K) %perform regression and returns coefients, residuals, and some statistics
% auxiliary matrixes
 ZMu=kron(eye(N),ones(T,1));
 Pmu=ZMu*inv(ZMu'*ZMu)*ZMu';
 Qmu=eye(N*T)-Pmu;
 clear ZMu Pmu

% thetas
instr2SLS=A*inv(A'*V*A)*A'; %weighted matrix of instruments (GMM)
inv2SLS=inv(R'*instr2SLS*R);
theta=inv2SLS * (R'*instr2SLS*y);
clear('instr2SLS','inv2SLS');

% residuals
resi=y-R*theta;

% GMM-weighting covariance matrix based on GMM-3SLS residuals (current
% estimation)
ENT=reshape(resi,N*T,G); %arrange the set of G vectors of 3SLS residuals
    %V3SLS=kron((ENT'*ENT)./df,eye(N*T));
instr3SLS=single(A*inv(A'*(kron((ENT'*ENT)./(N*T),eye(N*T)))*A)*A');
inv3SLS=inv(R'*instr3SLS*R);

% White-Huber VCV matrix based on GMM-3SLS residuals (current estimation)
```

```
VCVWH=diag(resi.^2)./(N*T-K);      % VCVWH makes computing White's covariance matrix easy
instrWH3SLS=single(A*inv(A'*VCVWH*A)*A'); %weighted matrix of instruments (GMM)

% Omega 1: GMM-3SLS asymptotic covariance matrix with estimated 3SLS covariance matrix
stder1=diag(inv3SLS).^0.5; %std. errors

% Omega 2: GMM-3SLS with White-Huber VCV matrix (White, 1980) and 3SLS GMM
%           weighting matrix
VCV_White=(N*T-K) * inv3SLS * (R'*instr3SLS*VCVWH*instr3SLS*R) * inv3SLS; % White's VCV
stder2=diag(VCV_White).^0.5;

% Omega 3: GMM-3SLS with GMM-weighted White-Huber VCV matrix (White, 1980) and 3SLS GMM
%           weighting matrix
VCV_White=inv3SLS * (R'*instr3SLS*instrWH3SLS*instr3SLS*R) * inv3SLS; % White's VCV
stder3=diag(VCV_White).^0.5;
clear('VCV_White','instrWH3SLS','VCVWH');

% Beck-Katz VCV matrix based on GMM-3SLS residuals (current estimation)
VCVBK=[];
for g=1:1:G
    ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
    VCVBK=blkdiag(VCVBK,kron((ET'*ET)./(T-K),eye(T)));
end
instrBK3SLS=A*inv(A'*VCVBK*A)*A';

% Omega 4: GMM-3SLS with PCSEs (Beck & Katz, 1995)
VCV_BK=inv3SLS * (R'*instr3SLS*VCVBK*instr3SLS*R) * inv3SLS;
stder4=diag(VCV_BK).^0.5;

% Omega 5: GMM-3SLS with GMM-weighted PCSEs (Beck & Katz, 1995)
VCV_BK=(N*T-K) * inv3SLS * (R'*instr3SLS*instrBK3SLS*instr3SLS*R) * inv3SLS;
stder5=diag(VCV_BK).^0.5;
clear('instrBK3SLS','VCVBK','VCV_BK');

% statistics for each equation
%if T==1; dof=N-1; else dof=N*(T-1); end
sysresi=reshape(resi,N*T,G);
for g=1:1:G
    suby=single(y((g-1)*N*T+1:g*N*T,1));
    subresi=sysresi(:,g);
    SST=(suby-mean(suby))'*(suby-mean(suby)); %Total Sum of Squares
    SSR=subresi'*subresi; %Total Sum of Residuals
    stats(g).r2=1-SSR/SST; % Rsq
    stats(g).r2adj=1-((SSR/(N*T-K))/(SST/(N*T-1))); % Rsq
    stats(g).witvar=[]; %"within" variance
    stats(g).betvar=[]; %"between" variance
    stats(g).aic=log(SSR/(N*T))+(2*(K+1)/(N*T)); %Akaike Information Criterion
    stats(g).bic=log(SSR/(N*T))+((K+1)*log(N*T)/(N*T)); %Bayesian (Schwartz) Information↙
Criterion
end
coef=[theta,stder1,stder2,stder3,stder4,stder5];
```

```
%McElroy's R^2 (McElroy, 1977) and Akaike IC for SEM
sigma2=kron(sysresi'*sysresi/(N*T),Qmu);
stats(G+1).r2=1-(resi'*V*resi)/(y'*sigma2*y);
stats(G+1).r2adj=1-((resi'*V*resi)/(G*N*T-K*G))/((y'*sigma2*y)/(G*N*T-G));
stats(G+1).aic=(G*N*T)*log(det(sysresi'*sysresi/(N*T)))+2*G*K+G*(G+1);
d=G*N*T/(G*N*T-(G+K+1));
stats(G+1).bic=(G*N*T)*log(det(sysresi'*sysresi/(N*T)))+2*d*(G*K+0.5*G*(G+1));   %
instead of BIC, for the system we estimate AICC
```

```
%-----------------------------------------------------------------------
% Function G3SLS
%       Estimates GMM 3SLS.
%
% Input:
%       R: set of covariates
%       W: omega ^ (-1/2) (includes Qmu and Pmu)
%       V: omega (GMM-3SLS weighting matrix)
%       A: matrix of instruments
%       y: dependent variable
%       N: number of individuals
%       T: number of time periods
%
% Output:
%       theta: coefficients
%       stder: standard errors of coefficients
%       resi: residuals
%       stats: structure containing:
%           .r2: R-squared
%           .r2adj: adjusted R-squared
%           .witvar: sigma1-squared ('within' variance)
%           .betvar: sigma2-squared ('between' variance)
%           .aic: Akaike Information Criterion
%           .bic: Bayesian (Schwarz) Information Criterion
%
%---------------------------------------------------------------------------

function [coef,resi,stats]=G3SLSM_hetero(R,V,A,y,G,N,T,K) %perform regression and
return coefficients, residuals, and some statistics

% thetas
instr2SLS=A*inv(A'*V*A)*A'; %weighted matrix of instruments (GMM)
inv2SLS=inv(R'*instr2SLS*R);
theta=inv2SLS * (R'*instr2SLS*y);
clear('instr2SLS','inv2SLS');

% residuals
resi=y-R*theta;

% GMM-weighting covariance matrix based on GMM-3SLS residuals (current
% estimation)
ENT=reshape(resi,N*T,G); %arrange the set of G vectors of 3SLS residuals
    %V3SLS=kron((ENT'*ENT)./df,eye(N*T));
instr3SLS=single(A*inv(A'*(kron((ENT'*ENT)./(N*T),eye(N*T)))*A)*A');
inv3SLS=inv(R'*instr3SLS*R);

% White-Huber VCV matrix based on GMM-3SLS residuals (current estimation)
VCVWH=diag(resi.^2)./(N*T-K);  % VCVWH makes computing White's covariance matrix easy
instrWH3SLS=single(A*inv(A'*VCVWH*A)*A'); %weighted matrix of instruments (GMM)

% Omega 1: GMM-3SLS asymptotic covariance matrix with estimated 3SLS covariance matrix
stder1=diag(inv3SLS).^0.5; %std. errors
```

```
%Omega 2: GMM-3SLS with White-Huber VCV matrix (White, 1980) and 3SLS GMM
%          weighting matrix
VCV_White=(N*T-K) * inv3SLS * (R'*instr3SLS*VCVWH*instr3SLS*R) * inv3SLS; % White's VCV
stder2=diag(VCV_White).^0.5;

% Omega 3: GMM-3SLS with GMM-weighted White-Huber VCV matrix (White, 1980) and 3SLS GMM
%          weighting matrix
VCV_White=inv3SLS * (R'*instr3SLS*instrWH3SLS*instr3SLS*R) * inv3SLS; % White's VCV
stder3=diag(VCV_White).^0.5;
clear('VCV_White','instrWH3SLS','VCVWH');

% Beck-Katz VCV matrix based on GMM-3SLS residuals (current estimation)
VCVBK=[];
for g=1:1:G
    ET=reshape(ENT(:,g),T,N); %arrange the set of GMM-3SLS residuals
    VCVBK=blkdiag(VCVBK,kron((ET'*ET)./(T-K),eye(T)));
end
instrBK3SLS=A*inv(A'*VCVBK*A)*A';

% Omega 4: GMM-3SLS with PCSEs (Beck & Katz, 1995)
VCV_BK=inv3SLS * (R'*instr3SLS*VCVBK*instr3SLS*R) * inv3SLS;
stder4=diag(VCV_BK).^0.5;

% Omega 5: GMM-3SLS with GMM-weighted PCSEs (Beck & Katz, 1995)
VCV_BK=(N*T-K) * inv3SLS * (R'*instr3SLS*instrBK3SLS*instr3SLS*R) * inv3SLS;
stder5=diag(VCV_BK).^0.5;
clear('instrBK3SLS','VCVBK','VCV_BK');

% statistics for each equation
for g=1:1:(G+1)
    stats(g).r2=[]; % Rsq
    stats(g).r2adj=[]; % Rsq
    stats(g).witvar=[]; %"within" variance
    stats(g).betvar=[]; %"between" variance
    stats(g).aic=[]; %Akaike Information Criterion
    stats(g).bic=[]; %Bayesian (Schwartz) Information Criterion
end
coef=[theta,stder1,stder2,stder3,stder4,stder5];
```

```
%======================================================================
% function COMPILA
% This function compiles information from all estimations.
% Author: Diego Avanzini
%==========================================================================

function compila
FE_compila_eq1;
FE_compila_eq1_newvar;
FE_compila_eq2;
FE_compila_eq2_newvar;
FE_compila_sys;
FE_compila_sys_newvar;
RE_compila_eq1;
RE_compila_eq1_newvar;
RE_compila_eq2;
RE_compila_eq2_newvar;
RE_compila_sys;
RE_compila_sys_newvar;
end
```

```
%======================================================================
% Function COMPILA_EQ1
% This program compilates information from EQUATION 1
% Author: Diego Avanzini
%==========================================================================

function compila_eq1
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:11,14:21,26,28,30,32]; %skip HTAM(12), HTBMS(13), G3SPD_AM(22,23), G3SPD_BMS↙
(24),
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '======================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq1');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                rdo(met).coef(:,obs)=single(rdoeq1(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdoeq1(pmet).coef(:,2));
                if size(rdoeq1(pmet).coef,2)>2
                    for i=3:1:size(rdoeq1(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdoeq1(pmet).coef(:,1));
                        rdo(adicionales).stder(:,obs)=single(rdoeq1(pmet).coef(:,i));
                    end
                end
            end
        end

        filename=['.\resultados\compi_eq1\compi_eq1_' num2str(nt,'%3.0f') '_' num2str↙
(rho '%3.0f')];
```

```
        save(filename,'rdo');
        disp([filename ' Saved. Continue...']);
        clear('rdo');
        toc
    end
end
end
```

```
%===============================================================
% Function COMPILA_EQ1_NEWVAR
% This program compilates information from EQUATION 1
% Author: Diego Avanzini
%===================================================================

function compila_eq1_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:5,7:11,14:21,30]; % skip BEHTM(6), HTAM(12), HTBMS(13), G3SPD_AM(22,23),
     % G3SPD_BMS(24,25), G3SPD_HTMnew(26), G3SPD_Std_alter(27,28),
     % G3SPD_Corn_alter Pseudo-Effects (29), G3SPD_HTM_alter (31,32)
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '========================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '========================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_eq1\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq1');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                rdo(met).coef(:,obs)=single(rdoeq1(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdoeq1(pmet).coef(:,2));
                if size(rdoeq1(pmet).coef,2)>2
                    for i=3:1:size(rdoeq1(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdoeq1(pmet).coef(:,1));
                        rdo(adicionales).stder(:,obs)=single(rdoeq1(pmet).coef(:,i));
                    end
                end
            end
        end

        filename=['.\resu\compi_eq1\compi_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
```

```
3.0f )];
        save(filename,'rdo');
        disp([filename ' Saved. Continue...']);
        clear('rdo');
        toc
    end
end
end
```

```
%==========================================================================
% Function COMPILA_EQ2
% This program compilates information from EQUATION 2
% Author: Diego Avanzini
%============================================================================

function compila_eq2

clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:11,14:21,26,28,30,32]; %skip HTAM(12), HTBMS(13), G3SPD_AM(22,23), G3SPD_BMS↙
(24),
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                rdo(met).coef(:,obs)=single(rdoeq2(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdoeq2(pmet).coef(:,2));
                if size(rdoeq2(pmet).coef,2)>2
                    for i=3:1:size(rdoeq2(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdoeq2(pmet).coef(:,1));
                        rdo(adicionales).stder(:,obs)=single(rdoeq2(pmet).coef(:,i));
                    end
                end
            end
        end

        filename=['.\resultados\compi_eq2\compi_eq2_' num2str(nt,'%3.0f') '_' num2str↙
```

```
(rho '%3.0f')];
        save(filename,'rdo');
        disp([filename ' Saved. Continue...']);
        clear('rdo');
        toc
    end
end
```

```
%======================================================================
% Function COMPILA_EQ2_NEWVAR
% This program compilates information from EQUATION 2
% Author: Diego Avanzini
%==========================================================================

function compila_eq2_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:5,7:11,14:21,30]; %skip BEHTM(6),HTAM(12), HTBMS(13), G3SFE_AM(22,23),
        % G3SFE_BMS(24,25), G3SFE_HTMnew(26), G3SPD_Std(27,28),
        % G3SPD_Corn Pseudo-Effects(29), G3SPD_HTM(31,32)
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_eq2\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdoeq2');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                rdo(met).coef(:,obs)=single(rdoeq2(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdoeq2(pmet).coef(:,2));
                if size(rdoeq2(pmet).coef,2)>2
                    for i=3:1:size(rdoeq2(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdoeq2(pmet).coef(:,1));
                        rdo(adicionales).stder(:,obs)=single(rdoeq2(pmet).coef(:,i));
                    end
                end
            end
        end

        filename=['.\resu\compi_eq2\compi_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
```

```
3.0f )];
        save(filename,'rdo');
        disp([filename ' Saved. Continue...']);
        clear('rdo');
        toc
    end
end
end
```

```
%==================================================================
% Function COMPILA_SYS
% This program compilates information from SYSTEM
% Author: Diego Avanzini
%======================================================================

function compila_sys
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:5,7:11,14:21,28,30,33,34,36,37]; % skip BEHTM (6), HTAM(12),
                % HTBMS(13), G3SPD_AM(22,23), G3SPD_BMS(24,25), G3SPD_HTMnew(26),
                % G3SPD_Std_alter FE(27), G3SPD_Corn_alter FE(29),
                % G3SPD_HTM_alter Homo(31,32), G3SPD_HTM Hetero(35),
                % G3SPD_HTM_alter Hetero(38)
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdosys');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                if size(rdosys(pmet).coef,2)==0
                    rdosys(pmet).coef=NaN(16,8);
                end
                rdo(met).coef(:,obs)=single(rdosys(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdosys(pmet).coef(:,2));
                if size(rdosys(pmet).coef,2)>2
                    for i=3:1:size(rdosys(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdosys(pmet).coef(:,1));
                        rdo(adicionales).stder(:,obs)=single(rdosys(pmet).coef(:,i));
                    end
```

```
                end
            end
        end
        filename=['.\resultados\compi_sys\compi_sys_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')];
        save(filename,'rdo');
        disp([filename ' Saved. Continue...']);
        clear('rdo');
        toc
    end
end
end
```

```
%==========================================================================
% Function COMPILA_SYS_NEWVAR
% This program compilates information from SYSTEM
% Author: Diego Avanzini
%==============================================================================

function compila_sys_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50];
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';
G=2;
metodo=[1:5,7:11,14:21,30]; % skip BEHTM (6), HTAM(12),
                % HTBMS(13), G3SPD_AM(22,23), G3SPD_BMS(24,25), G3SPD_HTMnew(26),
                % G3SPD_Std_alter FE(27,28), G3SPD_Corn_alter FE(29),
                % G3SPD_HTM_alter Homo(31,32), G3SPD_Std Hetero(33),
                % G3SPD_Corn Hetero(34),G3SPD_HTM Hetero(35),
                % G3SPD_Std_alter Hetero(36)G3SPD_Corn_alter Hetero(37),
                % G3SPD_HTM_alter Hetero(38)
nmetodo=size(metodo,2);

for nt=1:1:size(ntcases,1)
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        for obs=1:1:1000
            filename=['.\esti_sys\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0↙
f') ...
                '\esti_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f') '_' num2str↙
(obs '%4.0f')];
            load(filename,'rdosys');
            adicionales=nmetodo;
            for met=1:1:nmetodo
                pmet=squeeze(metodo(1,met));
                if size(rdosys(pmet).coef,2)==0
                    rdosys(pmet).coef=NaN(16,8);
                end
                rdo(met).coef(:,obs)=single(rdosys(pmet).coef(:,1));
                rdo(met).stder(:,obs)=single(rdosys(pmet).coef(:,2));
                if size(rdosys(pmet).coef,2)>2
                    for i=3:1:size(rdosys(pmet).coef,2)
                        adicionales=adicionales+1;
                        rdo(adicionales).coef(:,obs)=single(rdosys(pmet).coef(:,1));
```

```
                    rdo(adicionales).stder(:,obs)=single(rdosys(pmet).coef(:,i));
                end
            end
        end
    end
    filename=['.\resu\compi_sys\compi_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f' )];
    save(filename,'rdo');
    disp([filename ' Saved. Continue...']);
    clear('rdo');
    toc
  end
end
end
```

```matlab
%=========================================================================
% Function STATS_EQ1
% This program perform complementary estimations for the EQUATION 1
% Author: Diego Avanzini
%=============================================================================

function stats_eq1
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxeq1=7; %maximum number of estimated NT combinations
metodo=[1:5,7:11,15:17,22,49,53];
nmetodo=size(metodo,2); %50 methods
statseq1=single(zeros(ntmaxeq1,11,nmetodo,8,9)); %? NT combinations, 11 RHOs, 50 methods, 8 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,3)*8]; %includes the constant term

for nt=1:1:ntmaxeq1 % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        ---------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        ---------------------------------');
        tic

        filename=['.\resultados\compi_eq1\compi_eq1_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')];
        load(filename,'rdo');

        truepar=[   0.055;  %y2
                    8;      %x1
                    2;      %x2
                    5;      %x3
                    7;      %z1
                    5;      %z2
                    8;      %phi1
                    1];      %mu1

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:8
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';
```

```
                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statseq1(nt,rho,met,param,1)=SE/K; %Bias
                statseq1(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statseq1(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %↙
Relative Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statseq1(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statseq1(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )↙
*100;  %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statseq1(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statseq1(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )↙
*100;  %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statseq1(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails,↙
95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met '↙
num2str(met)     '  param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statseq1(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of↙
times the true parameter belong to the estimated interval

%                 % Aditional measures
%                 %----------------------
%
%                 % Alternative RMSE -- statis: 10,11
%                 statseq1(nt,rho,met,param,10)=(statseq1(nt,rho,met,param,1)^2+statseq1↙
(nt,rho,met,param,4)^2)^.5; %alternative RMSE
%                 statseq1(nt,rho,met,param,11)=abs(statseq1(nt,rho,met,param,10) /↙
truepar(param,1) )*100; %alternative Relative RMSE
%
%                 % ASE ("Average" Standard Error) -- statis: 12
%                 statseq1(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard↙
```

```
Erro
            end
        end

    end
    toc
end
filename='.\resultados\stats\stats_eq1';
save(filename,'statseq1');
disp([filename ' guardado. Continuar...']);
```

```
%==========================================================================
% Function STATS_EQ1_NEWVAR
% This program perform complementary estimations for the EQUATION 1
% Author: Diego Avanzini
%==========================================================================

function stats_eq1_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxeq1=5; %maximum number of estimated NT combinations
metodo=[1:10,14:16,19,38,42,44];
nmetodo=size(metodo,2);
statseq1=single(zeros(ntmaxeq1,11,nmetodo,8,9)); %? NT combinations, 11 RHOs, ##
methods, 8 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,4)*8]; %includes the constant term

for nt=1:1:ntmaxeq1 % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '========================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '========================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        filename=['.\resu\compi_eq1\compi_eq1_' num2str(nt,'%3.0f') '_' num2str(rho,'%
3.0f' )];
        load(filename,'rdo');

        truepar=[   0.055;  %y2
                    8;      %x1
                    2;      %x2
                    5;      %x3
                    7;      %z1
                    5;      %z2
                    8;      %phi1
                    1];      %mu1

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:8
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';
```

```
                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statseq1(nt,rho,met,param,1)=SE/K; %Bias
                statseq1(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statseq1(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %Relative Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statseq1(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statseq1(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )*100; %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statseq1(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statseq1(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )*100; %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statseq1(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails, 95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met ' num2str(met) ' param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statseq1(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of times the true parameter belong to the estimated interval

                % Bias of the Std. Error -- statis: 10
                statseq1(nt,rho,met,param,10)=(std(coef)-(sum(stder)/K))/std(coef)*100; %BSE (Bias of the Standard Error)


%                 % Aditional measures
%                 %----------------------
%
%                 % Alternative RMSE -- statis: 10,11
%                 statseq1(nt,rho,met,param,10)=(statseq1(nt,rho,met,param,1)^2+statseq1(nt,rho,met,param,4)^2)^.5; %alternative RMSE
```

```
%                 statseq1(nt,rho,met,param,11)=abs(statseq1(nt,rho,met,param,10) / 
true(param,1) )*100; %alternative Relative RMSE
%
%                  % ASE ("Average" Standard Error) -- statis: 12
%                  statseq1(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard
Error)
            end
        end

    end
    toc
end
filename='.\resu\stats\stats_eq1';
save(filename,'statseq1');
disp([filename ' guardado. Continuar...']);
```

```
%======================================================================
% Function STATS_EQ2
% This program perform complementary estimations for the EQUATION 2
% Author: Diego Avanzini
%==========================================================================

function stats_eq2
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxeq2=7; %maximum number of estimated NT combinations
metodo=[1:5,7:11,15:17,22,49,53];
nmetodo=size(metodo,2); %50 methods
statseq2=single(zeros(ntmaxeq2,11,nmetodo,8,9)); %? NT combinations, 11 RHOs, 50 methods, 8 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,3)*8]; %includes the constant term

for nt=1:1:ntmaxeq2 % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        filename=['.\resultados\compi_eq2\compi_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f')];
        load(filename,'rdo');

        truepar=[   -0.2;   %y1
                    12;     %x1
                    7;      %x4
                    3;      %x5
                    6;      %z1
                    5;      %z3
                    4;      %varphi2
                    1];     %phi2

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:8
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';
```

```
                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statseq2(nt,rho,met,param,1)=SE/K; %Bias
                statseq2(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statseq2(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %
Rela Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statseq2(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statseq2(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )
*100;  %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statseq2(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statseq2(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )
*100;  %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statseq2(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails,
95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met '
num2str(met)     ' param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statseq2(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of
times the true parameter belong to the estimated interval

%                 % Aditional measures
%                 %----------------------
%
%                 % Alternative RMSE -- statis: 10,11
%                 statseq2(nt,rho,met,param,10)=(statseq2(nt,rho,met,param,1)^2+statseq2
(nt,rho,met,param,4)^2)^.5; %alternative RMSE
%                 statseq2(nt,rho,met,param,11)=abs(statseq2(nt,rho,met,param,10) /
truepar(param,1) )*100; %alternative Relative RMSE
%
%                 % ASE ("Average" Standard Error) -- statis: 12
%                 statseq2(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard
```

```
Erro)
            end
        end

    end
    toc
end
filename='.\resultados\stats\stats_eq2';
save(filename,'statseq2');
disp([filename ' guardado. Continuar...']);
```

```
%=======================================================================
% Function STATS_EQ2_NEWVAR
% This program perform complementary estimations for the EQUATION 2
% Author: Diego Avanzini
%===========================================================================

function stats_eq2_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,↙
T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxeq2=5; %maximum number of estimated NT combinations
metodo=[1:10,14:16,19,38,42,44];
nmetodo=size(metodo,2);
statseq2=single(zeros(ntmaxeq2,11,nmetodo,8,9)); %? NT combinations, 11 RHOs, ##↙
methods, 8 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,4)*8]; %includes the constant term

for nt=1:1:ntmaxeq2 % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        filename=['.\resu\compi_eq2\compi_eq2_' num2str(nt,'%3.0f') '_' num2str(rho,'%↙
3.0f' )];
        load(filename,'rdo');

        truepar=[   -0.2;   %y1
                    12;     %x1
                    7;      %x4
                    3;      %x5
                    6;      %z1
                    5;      %z3
                    4;      %varphi2
                    1];     %phi2

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:8
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';
```

```
                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statseq2(nt,rho,met,param,1)=SE/K; %Bias
                statseq2(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statseq2(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %Relative Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statseq2(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statseq2(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )*100;  %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statseq2(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statseq2(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )*100;  %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statseq2(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails, 95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met ' num2str(met)     '  param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statseq2(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of times the true parameter belong to the estimated interval

                % Bias of the Std. Error
                serep=var(coef)^0.5; %std. error of the estimated coefficient over replications
                statseq2(nt,rho,met,param,10)=(serep-(sum(stder)/K))/serep*100; % BSE (Bias of the Standard Error)

%                 % Aditional measures
%                 %----------------------
%
%                 % Alternative RMSE -- statis: 10,11
%                 statseq2(nt,rho,met,param,10)=(statseq2(nt,rho,met,param,1)^2+statseq2
```

```
(nt,rho,met,param,4)^2)^.5; %alternative RMSE
%                  statseq2(nt,rho,met,param,11)=abs(statseq2(nt,rho,met,param,10) /
true(param,1) )*100; %alternative Relative RMSE
%
%                  % ASE ("Average" Standard Error) -- statis: 12
%                  statseq2(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard
Error)
            end
        end

    end
    toc
end
filename='.\resu\stats\stats_eq2';
save(filename,'statseq2');
disp([filename ' guardado. Continuar...']);
```

```
%=======================================================================
% Function STAT_SYS
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%===========================================================================

function stat_sys
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxsys=7; %maximum number of estimated NT combinations
metodo=[1:10,14:16,20,49,53];
nmetodo=size(metodo,2); %50 methods
statssys=single(zeros(ntmaxsys,11,nmetodo,16,9)); %? NT combinations, 11 RHOs, 50 meth, 16 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,3)*8]; %includes the constant term

for nt=1:1:ntmaxsys % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        filename=['.\resultados\compi_sys\compi_sys_' num2str(nt,'%3.0f') '_' num2str(rho '%3.0f')];
        load(filename,'rdo');

        truepar=[   0.055;  %y2
                    8;      %x1
                    2;      %x2
                    5;      %x3
                    7;      %z1
                    5;      %z2
                    8;      %varphi1
                    1;      %phi1
                    -0.2;   %y1
                    12;     %x1
                    7;      %x4
                    3;      %x5
                    6;      %z1
                    5;      %z3
```

```
                    4;      %varphi2
                    1];     %phi2

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:16
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';

                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statssys(nt,rho,met,param,1)=SE/K; %Bias
                statssys(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statssys(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %Relative Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statssys(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statssys(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )*100;  %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statssys(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statssys(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )*100;  %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statssys(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails, 95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met ' num2str(met) ' param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statssys(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of times the true parameter belong to the estimated interval

%                   % Aditional measures
%                   %----------------------
%
```

```
%                 % Alternative RMSE -- statis: 10,11
%                  statssys(nt,rho,met,param,10)=(statssys(nt,rho,met,param,1)^2+statssys
(nt,rho,met,param,4)^2)^.5; %alternative RMSE
%                  statssys(nt,rho,met,param,11)=abs(statssys(nt,rho,met,param,10) /
par_true(param,1) )*100; %alternative Relative RMSE
%
%                  % ASE ("Average" Standard Error) -- statis: 12
%                  statssys(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard
Error)
            end
        end

    end
    toc
end
filename='.\resultados\stats\stats_sys';
save(filename,'statssys');
disp([filename ' guardado. Continuar...']);
```

```
%=======================================================================
% Function STATS_SYS_NEWVAR
% This program perform complementary estimations for the SYSTEM
% Author: Diego Avanzini
%===========================================================================

function stats_sys_newvar
clc
clear all
ntcases=[20,15;20,30;20,50;60,15;60,30;60,50;200,15;200,30;200,50]; %combinations of (N,T)
rhocases=[0,1/12,1/6,1/4,1/3,1/2,2/3,3/4,5/6,11/12,1]';     %b/w ratio
G=2; %number of equations
K=1000; %number of iterations
ntmaxsys=4; %maximum number of estimated NT combinations
metodo=[1:10,14:16,19,38,42,44];
nmetodo=size(metodo,2);
statssys=single(zeros(ntmaxsys,11,nmetodo,16,9)); %? NT combinations, 11 RHOs, ##methods, 16 parameters, 9 measures
nvars=[7,4,4,ones(1,7)*7,4,3,8,ones(1,4)*8]; %includes the constant term

for nt=1:1:ntmaxsys % up to 9 combinations of N and T
    N=ntcases(nt,1);
    T=ntcases(nt,2);
    disp( '=======================================================');
    disp(['Caso: ' num2str(nt) '    N= ' num2str(N) '   T= ' num2str(T)]);
    disp( '=======================================================');

    for rho=1:1:size(rhocases,1)  % 11 rho (rhomu/rhoepsilon) coefficients
        disp( '        --------------------------------');
        disp(['        Rho= ' num2str(rho)]);
        disp( '        --------------------------------');
        tic

        filename=['.\resu\compi_sys\compi_sys_' num2str(nt,'%3.0f') '_' num2str(rho,'%3.0f' )];
        load(filename,'rdo');

        truepar=[   0.055;  %y2
                    8;      %x1
                    2;      %x2
                    5;      %x3
                    7;      %z1
                    5;      %z2
                    8;      %varphi1
                    1;      %phi1
                    -0.2;   %y1
                    12;     %x1
                    7;      %x4
                    3;      %x5
                    6;      %z1
                    5;      %z3
```

```
                4;      %varphi2
                1];     %phi2

        for met=1:1:nmetodo
            pmet=squeeze(metodo(1,met)); %puntero de metodo
            for param=1:1:16
                coef=single(squeeze(rdo(pmet).coef(param,:)))';
                stder=single(squeeze(rdo(pmet).stder(param,:)))';

                % SSSE (Sum of Sample Squared Errors)
                SSSE=(coef-mean(coef))'*(coef-mean(coef));

                % SSE (Sum of Squared Errors)
                SSE=sum((coef-truepar(param,1)).^2);

                % SE (Sum of Errors)
                SE=sum(coef-truepar(param,1));

                % B (Bias) -- statis: 1,2,3
                statssys(nt,rho,met,param,1)=SE/K; %Bias
                statssys(nt,rho,met,param,2)=(SE/K).^2; %Squared Bias
                statssys(nt,rho,met,param,3)=abs( (SE/K) / truepar(param,1) )*100; %Relative Absolute Bias

                % E (Efficiency) -- statis: 4,5
                statssys(nt,rho,met,param,4)=(SSSE/(K-1))^.5; %Efficiency
                statssys(nt,rho,met,param,5)=abs( ((SSSE/(K-1))^.5) / truepar(param,1) )*100;  %Relative Efficiency

                % RMSE (Root Mean Squared Error) -- statis: 6,7
                statssys(nt,rho,met,param,6)=(SSE/(K-1))^.5; %RMSE
                statssys(nt,rho,met,param,7)=abs( ((SSE/(K-1))^.5 ) / truepar(param,1) )*100;  %Relative RMSE

                % OC (Overconfidence) -- statis: 8
                SumVar=sum(stder.^2); %suma de las varianzas de los coeficientes
                statssys(nt,rho,met,param,8)=( (SSSE^.5)/(SumVar^.5) )*100;

                % "95% size" -- statis: 9
                t_value=tinv(0.975,N*T-nvars(1,met)); % t value for a 't' Test, 2 tails, 95% confidence; nvars includes the constant term
                UB=coef+(t_value*stder); %Interval upper bound
                LB=coef-(t_value*stder); %Interval lower bound
                if ~isreal(UB) || ~isreal(LB);disp(['Non-real numbers in stder: met ' num2str(met)     '  param ' num2str(param)]);end
                condi=(truepar(param,1)<=UB)&(truepar(param,1)>=LB);
                statssys(nt,rho,met,param,9)=( sum(condi)/K )*100; %counts the number of times the true parameter belong to the estimated interval

                % Bias of the Std. Error
                serep=var(coef)^0.5; %std. error of the estimated coefficient over repetitions
```

```
                statssys(nt,rho,met,param,10)=(serep-(sum(stder)/K))/serep*100; % BSE (Bias of the Standard Error)

%                  % Aditional measures
%                  %----------------------
%
%                  % Alternative RMSE -- statis: 10,11
%                  statssys(nt,rho,met,param,10)=(statssys(nt,rho,met,param,1)^2+statssys(nt,rho,met,param,4)^2)^.5; %alternative RMSE
%                  statssys(nt,rho,met,param,11)=abs(statssys(nt,rho,met,param,10) / true(param,1) )*100; %alternative Relative RMSE
%
%                  % ASE ("Average" Standard Error) -- statis: 12
%                  statssys(nt,rho,met,param,12)=sum(stder)/K; % ASE ("Average" Standard Error)
            end
        end

    end
    toc
end
filename='.\resu\stats\stats_sys';
save(filename,'statssys');
disp([filename ' guardado. Continuar...']);
```