systemfit: A Package to Estimate Simultaneous Equation Systems in R

Arne Henningsen, University of Kiel
Jeff D. Hamann, Forest Informatics, Inc.

March 15, 2006

Abstract

Many statistical analyses are based on models containing systems of structurally related equations. In cases where cross-equation disturbances are correlated, full information methods are required (Zellner, 1962). If exogenous variables are stochastically dependent on the disturbances in the system, then instrumental variable estimation methods should be used (Zellner and Theil, 1962) The package **systemfit** provides the capability to estimate systems of linear equations within the R programming environment.

Keywords: *R, simultaneous equations systems, seemingly unrelated regression, two-stage least squares, three-stage least squares*

1 Introduction

Many theoretical models that are econometrically estimated consist of more than one equation. The disturbance terms of these equations are likely to be contemporaneously correlated, because unconsidered factors that influence the disturbance term in one equation probably influence the disturbance terms in other equations. Ignoring this contemporaneous correlation and estimating these equations separately leads to inefficient parameter estimates. However, estimating all equations simultaneously, taking the covariance structure of the residuals into account, leads to efficient estimates. This estimation procedure is generally called "Seemingly Unrelated Regression" (SUR) (Zellner, 1962). Another reason to estimate an equation system simultaneously are cross-equation parameter restrictions.[1] These restrictions can be tested and/or imposed only in a simultaneous estimation approach.

Furthermore, these models can contain variables that appear on the left-hand side in one equation and on the right-hand side of another equation. Ignoring the endogeneity of these variables can lead to inconsistent parameter estimates. This simultaneity bias can be corrected for in each equation by applying a "Two-Stage Least Squares" (2SLS) method or for all equations simultaneously when combined with SUR resulting in a "Three-Stage Least Squares" (3SLS) estimation of the system of equations.

The **systemfit** package provides the capability to estimate linear equation systems in R (R Development Core Team, 2005). Although linear equation systems can be estimated with several other statistical and econometric software packages (e.g. SAS, EViews, TSP), **systemfit** has several advantages. First, all estimation procedures are publicly available in the source code. Second, the estimation algorithms can be easily modified to meet specific requirements. Third,

[1] Especially the economic theory suggests many cross-equation parameter restrictions (e.g. the symmetry restriction in demand models).

the (advanced) user can control estimation details generally not available in other software packages by overriding reasonable defaults.

In Section 2 we introduce the statistical background of estimating equation systems. The implementation of the statistical procedures in R is shortly explained in Section 3. Section 4 demonstrates how to run **systemfit** and how some of the features presented in the previous section can be utilized. In Section 5 the reliability of the results from **systemfit** are presented. Finally, a summary and outlook are presented in Section 6.

2 Statistical background

In this section we provide the statistical background of the functionality provided by the **systemfit** package. After introducing notations and assumptions, we provide the formulas to estimate systems of linear equations. We then demonstrate how to impose linear restrictions on parameters. Finally, we present additional relevant issues about estimation of equation systems.

Consider a system of G equations, where the ith equation is of the form

$$y_i = X_i \beta_i + u_i, \quad i = 1, 2, \ldots, G \tag{1}$$

where y_i is a vector of the dependent variable, X_i is a matrix of the exogenous variables, β_i is the coefficient vector and u_i is a vector of the disturbance terms of the ith equation.

We can write the "stacked" system as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_G \end{bmatrix} = \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & X_G \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_G \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_G \end{bmatrix} \tag{2}$$

or more simply as

$$y = X\beta + u \tag{3}$$

We assume that there is no correlation of the disturbance terms across observations:

$$E\left(u_{it}\, u_{jt^*}\right) = 0 \;\forall\; t \neq t^* \tag{4}$$

where i and j indicate the equation number and t and t^* denote the observation number.

However, we explicitly allow for contemporaneous correlation:

$$E\left(u_{it}\, u_{jt}\right) = \sigma_{ij} \tag{5}$$

Thus, the covariance matrix of the total system is

$$E\left(u\, u'\right) = \Omega = \Sigma \otimes I \tag{6}$$

where $\Sigma = [\sigma_{ij}]$ is the residual covariance matrix and I is an identity matrix.

2.1 Estimation

2.1.1 Ordinary least squares (OLS)

The Ordinary Least Squares (OLS) estimator of the system is obtained by

$$\widehat{\beta}_{OLS} = \left(X'X\right)^{-1} X'y \tag{7}$$

These estimates are efficient only if the disturbance terms are not contemporaneously correlated, which means $\sigma_{ij} = 0 \; \forall \; i \neq j$. If the whole system is treated as one single equation, the covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{OLS}\right] = \sigma^2 \left(X'X\right)^{-1} \tag{8}$$

with $\sigma^2 = E\left(u'u\right)$. This assumes that the disturbances of all equations have the same variance.

If the disturbance terms of the individual equations are allowed to have different variances, the covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{OLS}\right] = \left(X'\Omega^{-1}X\right)^{-1} \tag{9}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \; \forall \; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$.

If no cross-equation parameter restrictions are imposed, the simultaneous OLS estimation of the system leads to the same parameter estimates as an equation-wise OLS estimation. The covariance matrix of the parameters from an equation-wise OLS estimation is equal to the covariance matrix obtained by equation (9).

2.1.2 Weighted least squares (WLS)

The Weighted Least Squares (WLS) estimator of the system is obtained by

$$\widehat{\beta}_{WLS} = \left(X'\Omega^{-1}X\right)^{-1} X'\Omega^{-1}y \tag{10}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \; \forall \; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$. Like the OLS estimates these estimates are only efficient if the disturbance terms are not contemporaneously correlated. The covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{WLS}\right] = \left(X'\Omega^{-1}X\right)^{-1} \tag{11}$$

If no cross-equation parameter restrictions are imposed, the parameter estimates are equal to the OLS estimates.

2.1.3 Seemingly unrelated regression (SUR)

When the disturbances are contemporaneously correlated, a Generalized Least Squares (GLS) estimation leads to efficient parameter estimates. In this case, the GLS is generally called "Seemingly Unrelated Regression" (SUR) (Zellner, 1962). It should be noted that while an unbiased OLS or WLS estimation requires only that the regressors and the disturbance terms of each single equation are uncorrelated ($E\left[u_i | X_i\right] = 0 \;\; \forall \;\; i$), a consistent SUR estimation requires that all disturbance terms and all regressors are uncorrelated ($E\left[u | X\right] = 0$).

The SUR estimator can be obtained by:

$$\widehat{\beta}_{SUR} = \left(X'\Omega^{-1}X\right)^{-1} X'\Omega^{-1}y \tag{12}$$

with $\Omega = \Sigma \otimes I$ and $\sigma_{ij} = E\left(u_i' u_j\right)$. And the covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{SUR}\right] = \left(X'\Omega^{-1}X\right)^{-1} \tag{13}$$

2.1.4 Two-stage least squares (2SLS)

If the regressors of one or more equations are correlated with the disturbances ($E\left(u_i | X_i\right) \neq 0$), the estimated coefficients are biased. This can be circumvented by an instrumental variable (IV) two-stage least squares (2SLS) estimation. The instrumental variables for each equation H_i can be either different or identical for all equations. The instrumental variables of each equation may not be correlated with the disturbance terms of the corresponding equation ($E\left(u_i|H_i\right)=0$).

At the first stage new ('fitted') regressors are obtained by

$$\widehat{X_i}=H_i\left(H_i'H_i\right)^{-1}H_i'X \tag{14}$$

At the second stage the unbiased two-stage least squares estimates of β are obtained by:

$$\widehat{\beta}_{2SLS}=\left(\widehat{X}'\widehat{X}\right)^{-1}\widehat{X}'y \tag{15}$$

If the whole system is treated as one single equation, the covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{2SLS}\right]=\sigma^2\left(\widehat{X}'\widehat{X}\right)^{-1} \tag{16}$$

with $\sigma^2=E\left(u'u\right)$. If the disturbance terms of the individual equations are allowed to have different variances, the covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{2SLS}\right]=\left(\widehat{X}'\Omega^{-1}\widehat{X}\right)^{-1} \tag{17}$$

with $\Omega=\Sigma\otimes I$, $\sigma_{ij}=0\;\forall\;i\neq j$ and $\sigma_{ii}=E\left(u_i'u_i\right)$.

2.1.5 Weighted two-stage least squares (W2SLS)

The Weighted Two-Stage Least Squares (W2SLS) estimator of the system is obtained by

$$\widehat{\beta}_{W2SLS}=\left(\widehat{X}'\Omega^{-1}\widehat{X}\right)^{-1}\widehat{X}'\Omega^{-1}y \tag{18}$$

with $\Omega=\Sigma\otimes I$, $\sigma_{ij}=0\;\forall\;i\neq j$ and $\sigma_{ii}=E\left(u_i'u_i\right)$. The covariance matrix of the estimated parameters is

$$\text{Cov}\left[\widehat{\beta}_{W2SLS}\right]=\left(\widehat{X}'\Omega^{-1}\widehat{X}\right)^{-1} \tag{19}$$

2.1.6 Three-stage least squares (3SLS)

If the regressors are correlated with the disturbances ($E\left(u|X\right)\neq 0$) and the disturbances are contemporaneously correlated, a Generalized Least Squares (GLS) version of the two-stage least squares estimation leads to consistent and efficient estimates. This estimation procedure is generally called "Three-stage Least Squares" (3SLS) (Zellner and Theil, 1962).

The standard 3SLS estimator can be obtained by:

$$\widehat{\beta}_{3SLS}=\left(\widehat{X}'\Omega^{-1}\widehat{X}\right)^{-1}\widehat{X}'\Omega^{-1}y \tag{20}$$

with $\Omega=\Sigma\otimes I$ and $\sigma_{ij}=E\left(u_i'u_j\right)$. Its covariance matrix is:

$$\text{Cov}\left[\widehat{\beta}_{3SLS}\right]=\left(\widehat{X}'\Omega^{-1}\widehat{X}\right)^{-1} \tag{21}$$

While an unbiased 2SLS or W2SLS estimation requires only that the instrumental variables and the disturbance terms of each single equation are uncorrelated ($E\left[u_i|H_i\right]=0\;\forall\;i$), Schmidt (1990) points out that this estimator is only consistent if all disturbance terms and all instrumental variables are uncorrelated ($E\left[u|H\right]=0$) with

$$H=\begin{bmatrix} H_1 & 0 & \cdots & 0 \\ 0 & H_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_G \end{bmatrix} \tag{22}$$

Since there might be occasions where this cannot be avoided, Schmidt (1990) analyses other approaches to obtain 3SLS estimators:

One of these approaches is based on instrumental variable estimation (3SLS-IV):

$$\widehat{\beta}_{3SLS-IV} = \left(\widehat{X}' \Omega^{-1} X \right)^{-1} \widehat{X}' \Omega^{-1} y \tag{23}$$

The covariance matrix of this 3SLS-IV estimator is:

$$\operatorname{Cov}\left[\widehat{\beta}_{3SLS-IV} \right] = \left(\widehat{X}' \Omega^{-1} X \right)^{-1} \tag{24}$$

Another approach is based on the Generalized Method of Moments (GMM) estimator (3SLS-GMM):

$$\widehat{\beta}_{3SLS-GMM} = \left(X'H \left(H' \Omega H \right)^{-1} H'X \right)^{-1} X'H \left(H' \Omega H \right)^{-1} H'y \tag{25}$$

The covariance matrix of the 3SLS-GMM estimator is:

$$\operatorname{Cov}\left[\widehat{\beta}_{3SLS-GMM} \right] = \left(X'H \left(H' \Omega H \right)^{-1} H'X \right)^{-1} \tag{26}$$

A fourth approach developed by Schmidt (1990) himself is:

$$\widehat{\beta}_{3SLS-Schmidt} = \left(\widehat{X}' \Omega^{-1} \widehat{X} \right)^{-1} \widehat{X}' \Omega^{-1} H \left(H'H \right)^{-1} H'y \tag{27}$$

The covariance matrix of this estimator is:

$$\operatorname{Cov}\left[\widehat{\beta}_{3SLS-Schmidt} \right] = \left(\widehat{X}' \Omega^{-1} \widehat{X} \right)^{-1} \widehat{X}' \Omega^{-1} H \left(H'H \right)^{-1} H' \Omega H \left(H'H \right)^{-1} H' \Omega^{-1} \widehat{X} \left(\widehat{X}' \Omega^{-1} \widehat{X} \right)^{-1} \tag{28}$$

The econometrics software EViews uses following approach:

$$\widehat{\beta}_{3SLS-EViews} = \widehat{\beta}_{2SLS} + \left(\widehat{X}' \Omega^{-1} \widehat{X} \right)^{-1} \widehat{X}' \Omega^{-1} \left(y - X \widehat{\beta}_{2SLS} \right) \tag{29}$$

where $\widehat{\beta}_{2SLS}$ is the two-stage least squares estimator as defined by (15). EViews uses the standard 3SLS formula (21) to calculate the covariance matrix of the 3SLS estimator.

If the same instrumental variables are used in all equations ($H_1 = H_2 = \ldots = H_G$), all the above mentioned approaches lead to identical parameter estimates. However, if this is not the case, the results depend on the method used (Schmidt, 1990). The only reason to use different instruments for different equations is a correlation of the instruments of one equation with the disturbance terms of another equation. Otherwise, one could simply use all instruments in every equation (Schmidt, 1990). In this case, only the 3SLS-GMM (25) and the 3SLS estimator developed by Schmidt (1990) (27) are consistent.

2.2 Imposing linear restrictions

It is common to perform hypothesis tests by imposing restrictions on the parameter estimates. There are two ways to impose linear parameter restrictions. First, a matrix T can be specified that

$$\beta = T \cdot \beta^* \tag{30}$$

where β^* is a vector of restricted (linear independent) coefficients, and T is a matrix with the number of rows equal to the number of unrestricted coefficients (β) and the number of columns equal to the number of restricted coefficients (β^*). T can be used to map each unrestricted coefficient to one or more restricted coefficients.

To impose these restrictions, the X matrix is (post-)multiplied by this T matrix.

$$X^* = X \cdot T \tag{31}$$

Then, X^* is substituted for X and a standard estimation as described in the previous section is done (equations 7–29). This results in the linear independent parameter estimates β^* and their covariance matrix. The original parameters can be obtained by equation (30) and the covariance matrix of the original parameters can be obtained by:

$$\text{Cov}\left[\widehat{\beta}\right] = T \cdot \text{Cov}\left[\widehat{\beta}^*\right] \cdot T' \tag{32}$$

The second way to impose linear parameter restrictions can be formulated by

$$R\beta^0 = q \tag{33}$$

where β^0 is the vector of the restricted coefficients, and R and q are a matrix and vector, respectively, to impose the restrictions (see Greene, 2003, p. 100). Each linear independent restriction is represented by one row of R and the corresponding element of q.

The first way is less flexible than this latter one[2], but the first way is preferable if equality constraints for coefficients across many equations of the system are imposed. Of course, these restrictions can be also imposed using the latter method. However, while the latter method increases the dimension of the matrices to be inverted during estimation, the first reduces it. Thus, in some cases the latter way leads to estimation problems (e.g. (near) singularity of the matrices to be inverted), while the first doesn't.

These two methods can be combined. In this case the restrictions imposed using the latter method are imposed on the linear independent parameters due to the restrictions imposed using the first method:

$$R\beta^{*0} = q \tag{34}$$

where β^{*0} is the vector of the restricted β^* coefficients.

2.2.1 Restricted OLS estimation

The OLS estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{OLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'X & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} X'y \\ q \end{bmatrix} \tag{35}$$

where λ is a vector of the Lagrangean multipliers of the restrictions. If the whole system is treated as one single equation, the covariance matrix of the estimated parameters is

$$\text{Cov}\begin{bmatrix} \widehat{\beta}^0_{OLS} \\ \widehat{\lambda} \end{bmatrix} = \sigma^2 \begin{bmatrix} X'X & R' \\ R & 0 \end{bmatrix}^{-1} \tag{36}$$

with $\sigma^2 = E\left(u'u\right)$. If the disturbance terms of the individual equations are allowed to have different variances, the covariance matrix of the estimated parameters is

$$\text{Cov}\begin{bmatrix} \widehat{\beta}^0_{OLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \tag{37}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \; \forall \; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$.

[2] While restrictions like $\beta_1 = 2\beta_2$ can be imposed by both methods, restrictions like $\beta_1 + \beta_2 = 4$ can be imposed only by the second method.

2.2.2 Restricted WLS estimation

The WLS estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{WLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} X'\Omega^{-1}y \\ q \end{bmatrix} \tag{38}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \;\forall\; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$. The covariance matrix of the estimated parameters is

$$\operatorname{Cov} \begin{bmatrix} \widehat{\beta}^0_{WLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \tag{39}$$

2.2.3 Restricted SUR estimation

The SUR estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{SUR} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} X'\Omega^{-1}y \\ q \end{bmatrix} \tag{40}$$

with $\Omega = \Sigma \otimes I$ and $\sigma_{ij} = E\left(u_i' u_j\right)$. The covariance matrix of the estimated parameters is

$$\operatorname{Cov} \begin{bmatrix} \widehat{\beta}^0_{SUR} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \tag{41}$$

2.2.4 Restricted 2SLS estimation

The 2SLS estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{2SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'y \\ q \end{bmatrix} \tag{42}$$

If the whole system is treated as one single equation, the covariance matrix of the estimated parameters is

$$\operatorname{Cov} \begin{bmatrix} \widehat{\beta}^0_{2SLS} \\ \widehat{\lambda} \end{bmatrix} = \sigma^2 \begin{bmatrix} \widehat{X}'\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \tag{43}$$

with $\sigma^2 = E\left(u'u\right)$. If the disturbance terms of the individual equations are allowed to have different variances, the covariance matrix of the estimated parameters is

$$\operatorname{Cov} \begin{bmatrix} \widehat{\beta}^0_{2SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \tag{44}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \;\forall\; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$.

2.2.5 Restricted W2SLS estimation

The W2SLS estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{W2SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}y \\ q \end{bmatrix} \tag{45}$$

with $\Omega = \Sigma \otimes I$, $\sigma_{ij} = 0 \;\forall\; i \neq j$ and $\sigma_{ii} = E\left(u_i' u_i\right)$. The covariance matrix of the estimated parameters is

$$\operatorname{Cov} \begin{bmatrix} \widehat{\beta}^0_{W2SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \tag{46}$$

2.2.6 Restricted 3SLS estimation

The standard 3SLS estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{3SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}y \\ q \end{bmatrix} \tag{47}$$

with $\Omega = \Sigma \otimes I$ and $\sigma_{ij} = E\left(u_i' u_j\right)$. The covariance matrix of this estimator is

$$\text{Cov}\begin{bmatrix} \widehat{\beta}^0_{3SLS} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \tag{48}$$

The 3SLS-IV estimator restricted by $R\beta^0 = q$ can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{3SLS-IV} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}X & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}y \\ q \end{bmatrix} \tag{49}$$

with

$$\text{Cov}\begin{bmatrix} \widehat{\beta}^0_{3SLS-IV} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \tag{50}$$

The restricted 3SLS-GMM estimator can be obtained by

$$\begin{bmatrix} \widehat{\beta}^0_{3SLS-GMM} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'H\left(H'\Omega H\right)^{-1}H'X & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} X'H\left(H\Omega H\right)^{-1}H'y \\ q \end{bmatrix} \tag{51}$$

with

$$\text{Cov}\begin{bmatrix} \widehat{\beta}^0_{3SLS-GMM} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} X'H\left(H'\Omega H\right)^{-1}H'X & R' \\ R & 0 \end{bmatrix}^{-1} \tag{52}$$

The restricted 3SLS estimator based on the suggestion of Schmidt (1990) is:

$$\begin{bmatrix} \widehat{\beta}^0_{3SLS-Schmidt} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}H\left(H'H\right)^{-1}H'y \\ q \end{bmatrix} \tag{53}$$

with

$$\begin{aligned} \text{Cov}\begin{bmatrix} \widehat{\beta}^0_{3SLS-Schmidt} \\ \widehat{\lambda} \end{bmatrix} \quad = \quad & \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \\ & \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}H\left(H'H\right)^{-1}H'\Omega H\left(H'H\right)^{-1}H'\Omega^{-1}\widehat{X} & 0' \\ 0 & 0 \end{bmatrix}^{-1} \\ & \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \end{aligned} \tag{54}$$

The econometrics software EViews calculates the restricted 3SLS estimator by:

$$\begin{bmatrix} \widehat{\beta}^0_{3SLS-EViews} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{X}'\Omega^{-1}\widehat{X} & R' \\ R & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \widehat{X}'\Omega^{-1}\left(y - X\widehat{\beta}^0_{2SLS}\right) \\ q \end{bmatrix} \tag{55}$$

where $\widehat{\beta}^0_{2SLS}$ is the restricted 2SLS estimator calculated by equation (42). To calculate the covariance matrix EViews uses the standard formula of the restricted 3SLS estimator (48).

If the same instrumental variables are used in all equations ($H_1 = H_2 = \ldots = H_G$), all the above mentioned approaches lead to identical parameter estimates and identical covariance matrices of the estimated parameters.

2.3 Residual covariance matrix

Since the true residuals of the estimated equations are generally not known, the true covariance matrix of the residuals cannot be determined. Thus, this covariance matrix must be calculated from the *estimated* residuals. Generally, the estimated covariance matrix of the residuals ($\widehat{\Sigma} = [\widehat{\sigma}_{ij}]$) can be calculated from the residuals of a first-step OLS or 2SLS estimation. The following formula is often applied:

$$\widehat{\sigma}_{ij} = \frac{\widehat{u}_i' \widehat{u}_j}{T} \tag{56}$$

where T is the number of observations in each equation. However, in finite samples this estimator is biased, because it is not corrected for degrees of freedom. The usual single-equation procedure to correct for degrees of freedom cannot always be applied, because the number of regressors in each equation might differ. Two alternative approaches to calculate the residual covariance matrix are

$$\widehat{\sigma}_{ij} = \frac{\widehat{u}_i' \widehat{u}_j}{\sqrt{(T - K_i) \cdot (T - K_j)}} \tag{57}$$

and

$$\widehat{\sigma}_{ij} = \frac{\widehat{u}_i' \widehat{u}_j}{T - \max(K_i, K_j)} \tag{58}$$

where K_i and K_j are the number of regressors in equation i and j, respectively. However, these formulas yield unbiased estimators only if $K_i = K_j$ (Judge *et al.*, 1985, p. 469).

A further approach to obtain the estimated residual covariance matrix is (Zellner and Huang, 1962, p. 309)

$$\begin{aligned}
\widehat{\sigma}_{ij} &= \frac{\widehat{u}_i' \widehat{u}_j}{T - K_i - K_j + tr\left[X_i \left(X_i' X_i\right)^{-1} X_i' X_j \left(X_j' X_j\right)^{-1} X_j'\right]} && (59)\\
&= \frac{\widehat{u}_i' \widehat{u}_j}{T - K_i - K_j + tr\left[\left(X_i' X_i\right)^{-1} X_i' X_j \left(X_j' X_j\right)^{-1} X_j' X_i\right]} && (60)
\end{aligned}$$

This yields an unbiased estimator for all elements of $\widehat{\Sigma}$, but even if $\widehat{\Sigma}$ is an unbiased estimator of Σ, its inverse $\widehat{\Sigma}^{-1}$ is not an unbiased estimator of Σ^{-1} (Theil, 1971, p. 322). Furthermore, the covariance matrix calculated by (59) is not necessarily positive semidefinite (Theil, 1971, p. 322). Hence, "it is doubtful whether [this formula] is really superior to [(56)]" (Theil, 1971, p. 322).

The WLS, SUR, W2SLS and 3SLS parameter estimates are consistent, if the estimated residual covariance matrix is calculated using the residuals from a first-step OLS or 2SLS estimation. There exists also an alternative slightly different approach.[3] This alternative approach uses the residuals of a first-step OLS or 2SLS estimation to apply a WLS or W2SLS estimation on a second step. Then, it calculates the residual covariance matrix from the residuals of the second-step estimation to estimates the model by SUR or 3SLS in a third step. If no cross-equation restrictions are imposed, the parameter estimates of OLS and WLS as well as 2SLS and W2SLS are identical. Hence, in this case both approaches generate the same results.

It is also possible to iterate WLS, SUR, W2SLS and 3SLS estimations. At each iteration the residual covariance matrix is calculated from the residuals of the previous iteration. If equation (56) is applied to calculate the estimated residual covariance matrix, an iterated SUR estimation converges to maximum likelihood (Greene, 2003, p. 345).

[3] For instance, this approach is applied by the command "TSCS" of the software LIMDEP that carries out SUR estimations in which all coefficient vectors are constrained to be equal (Greene, 2006).

In some uncommon cases, for instance in pooled estimations, where the coefficients are restricted to be equal in all equations, the means of the residuals of each equation are not equal to zero ($\overline{\widehat{u}}_i \neq 0$). Therefore, it might be argued that the residual covariance matrix should be calculated by subtracting the means from the residuals and substituting $\widehat{u}_i - \overline{\widehat{u}}_i$ for $\widehat{u}_i$ in (56–59).

2.4 Degrees of freedom

To our knowledge the question about how to determine the degrees of freedom for single-parameter t-tests is not comprehensively discussed in the literature. While sometimes the degrees of freedom of the entire system (total number of observations in all equations minus total number of estimated parameters) are applied, in other cases the degrees of freedom of each single equation (number of observations in the equations minus number of estimated parameters in the equation) are used. Asymptotically, this distinction doesn't make a difference. However, in many empirical applications, the number of observations of each equation is rather small, and therefore, it matters.

If a system of equations is estimated by an unrestricted OLS and the covariance matrix of the parameters is calculated by (9), the estimated parameters and their standard errors are identical to an equation-wise OLS estimation. In this case, it is reasonable to use the degrees of freedom of each single equation, because this yields the same p-values as the equation-wise OLS estimation.

In contrast, if a system of equations is estimated with many cross-equation restrictions and the covariance matrix of an OLS estimation is calculated by (8), the system estimation is similar to a single equation estimation. Therefore, in this case, it seems to be reasonable to use the degrees of freedom of whole system.

2.5 Goodness of fit

The goodness of fit of each single equation can be measured by the traditional R^2 values:

$$R_i^2 = 1 - \frac{\widehat{u}_i' \widehat{u}_i}{(y_i - \overline{y_i})'(y_i - \overline{y_i})} \tag{61}$$

where R_i^2 is the R^2 value of the ith equation and $\overline{y_i}$ is the mean value of y_i.

The goodness of fit of the whole system can be measured by the McElroy's R^2 value (McElroy, 1977):

$$R_*^2 = 1 - \frac{\widehat{u}' \widehat{\Omega}^{-1} \widehat{u}}{y' \left(\widehat{\Sigma}^{-1} \otimes \left(I - \frac{ii'}{T} \right) \right) y} \tag{62}$$

where T is the number of observations in each equation, I is an $T \times T$ identity matrix and i is a column vector of T ones.

2.6 Testing linear restrictions

Linear restrictions can be tested by an F test, Wald test or likelihood-ratio (LR) test.

The F-statistic for systems of equations is

$$F = \frac{(R\hat{\beta} - q)'(R(X'(\hat{\Sigma} \otimes I)^{-1} X)^{-1} R')^{-1} (R\hat{\beta} - q)/j}{\hat{u}'(\Sigma \otimes I)^{-1} \hat{u}/(M \cdot T - K)} \tag{63}$$

where j is the number of restrictions, M is the number of equations, T is the number of observations per equation, K is the total number of estimated coefficients, and $\hat{\Sigma}$ is the estimated residual covariance matrix used in the estimation. Under the null hypothesis, F has an F-distribution with j and $M \cdot T - K$ degrees of freedom (Theil, 1971, p. 314).

The Wald-statistic for systems of equations is

$$W = (R\hat{\beta} - q)'(R\widehat{Cov}[\hat{\beta}]R')^{-1}(R\hat{\beta} - q) \tag{64}$$

Asymptotically, W has a χ^2 distribution with j degrees of freedom under the null hypothesis (Greene, 2003, p. 347).

The LR-statistic for systems of equations is

$$LR = T \cdot \left(log \left| \hat{\hat{\Sigma}}_r \right| - log \left| \hat{\hat{\Sigma}}_u \right| \right) \tag{65}$$

where T is the number of observations per equation, and $\hat{\hat{\Sigma}}_r$ and $\hat{\hat{\Sigma}}_u$ are the residual covariance matrices calculated by formula (56) of the restricted and unrestricted estimation, respectively. Asymptotically, LR has a χ^2 distribution with j degrees of freedom under the null hypothesis (Greene, 2003, p. 349).

2.7 Hausman test

Hausman (1978) developed a test for misspecification. The null hypotheses of the test is that all exogenous variables are uncorrelated with all disturbance terms. Under this hypothesis both the 2SLS and the 3SLS estimator are consistent but only the 3SLS estimator is (asymptotically) efficient. Under the alternative hypothesis the 2SLS estimator is consistent but the 3SLS estimator is inconsistent. The Hausman test statistic is,

$$m = \left(\hat{\beta}_{2SLS} - \hat{\beta}_{3SLS} \right)' \left(\text{Cov}\left[\hat{\beta}_{2SLS} \right] - \text{Cov}\left[\hat{\beta}_{3SLS} \right] \right) \left(\hat{\beta}_{2SLS} - \hat{\beta}_{3SLS} \right) \tag{66}$$

where $\hat{\beta}_{2SLS}$ and $\text{Cov}\left[\hat{\beta}_{2SLS}\right]$ are the estimated coefficient and covariance matrix from 2SLS estimation, and $\hat{\beta}_{3SLS}$ and $\text{Cov}\left[\hat{\beta}_{3SLS}\right]$ are the estimated coefficients and covariance matrix from 3SLS estimation. Under the null hypotheses this test statistic has a χ^2 distribution with degrees of freedom equal to the number of estimated parameters.

3 Source code

The **systemfit** package includes functions to estimate systems of equations (`systemfit`, `systemfitClassic`) and to test hypotheses in these systems (`ftest.systemfit`, `waldtest.systemfit`, `lrtest.systemfit`, `hausman.systemfit`). Furthermore, this package provides some helper functions e.g. to extract the estimated coefficients (`coef.systemfit`) or to calculate predicted values (`predict.systemfit`).

The source code of the **systemfit** is publicly available for download from "CRAN" (The Comprehensive R Archive Network, `http://cran.r-project.org/src/contrib/Descriptions/systemfit.html`). Since the whole package has more than 2,100 lines of code, it is not presented in this article. Furthermore, the code corresponds exactly to the procedures and formulas described in the previous section.

3.1 The basic function `systemfit`

The basic functionality of this package is provided by the function `systemfit`. This function estimates the equation system as described in sections 2.1. If parameter restrictions are provided, the formulas in section 2.2 are applied. Furthermore, the user can control several details of the estimation. For instance, the formula to calculate the residual covariance matrix (see section 2.3), the degrees of freedom for the t-tests (see section 2.4), or the formula for the 3SLS estimation (see sections 2.1 and 2.2) can be specified by the user. The `systemfit` function returns many objects that users might be interest in. A complete list is available in the documentation of this function that is included in the package.

3.2 The wrapper function `systemfitClassic`

Furthermore, the **systemfit** package includes the function `systemfitClassic`. This is a wrapper function for `systemfit` that can be applied to (classical) panel-like data in long format if the regressors are the same for all equations. The data are reshaped and the formulas are modified to enable an estimation using the standard `systemfit` function. The user can specify whether the coefficients should be restricted to be equal in all equations.

3.3 Statistical tests

The statistical tests described in sections 2.6 and 2.7 are implemented as specified in these sections. The functions `ftest.systemfit`, `waldtest.systemfit` and `lrtest.systemfit` test linear restrictions on the estimated parameters. On the other hand, the function `hausman.systemfit` tests the consistency of the 3SLS estimator. All functions return the empirical test statistic, the degree(s) of freedom, and the p-value.

3.4 Efficiency of the code

We have followed Bates (2004) to make the code faster and more stable. First, if a formula contains an inverse of a matrix that is post-multiplied by a vector, we use `solve(A, b)` instead of `solve(A) %*% b`. Second, we calculate crossproducts by `crossprod(X)` or `crossprod(X, y)` instead of `t(X) %*% X` or `t(X) %*% y`, respectively.

The matrix Ω^{-1} that is used to compute the estimated coefficients and their covariance matrix is of size $(G \cdot T) \times (G \cdot T)$ (see sections 2.1 and 2.2). In case of large data sets, this matrix Ω^{-1} gets really huge and needs a lot of memory. Therefore, we use the following transformation and compute $X'\Omega^{-1}$ by deviding the X matrix into submatrices, doing some calculations with these submatrices, adding up some of these submatrices, and finally putting the submatrices together:

$$X'\Omega^{-1} = \sum_{i=1} \begin{bmatrix} \sigma^{1i} X^1 \\ \sigma^{2i} X^2 \\ \vdots \\ \sigma^{Gi} X^G \end{bmatrix}' \tag{67}$$

where σ^{ij} are the elements of the matrix Σ^{-1}, and X^i is a submatrix of X that contains the rows that belong to the i's equation.

4 Using systemfit

In this section we demonstrate how to use the **systemfit** package. First, we show the standard usage of `systemfit` by a simple example. Second, several options that can be specified by the user are presented. Then, the wrapper function `systemfitClassic` is described. Finally, we demonstrate how to apply some statistical tests.

4.1 Standard usage of `systemfit`

As described in the previous section, equation systems can be econometrically estimated with the function `systemfit`. It is generally called by

```
> systemfit(method, eqns)
```

There are two mandatory arguments: `method` and `eqns`. The argument `method` is a string determining the estimation method. It must be either “OLS”, “WLS”, “SUR”, “WSUR”, “2SLS”, “W2SLS”, “3SLS”, or “W3SLS”. While six of these methods correspond to the estimation methods described in sections 2.1 and 2.2, the methods “WSUR” and “W3SLS” are “SUR” and “3SLS” estimations using the residual covariance matrices from "WLS" and "W2SLS" estimations, respectively (see section 2.3). The other mandatory argument, `eqns`, is a list of the equations to be estimated. Each equation is a standard formula in R and starts with a dependent variable on the left hand side. After a tilde (∼) the regressors are listed, separated by plus signs[4].

The following demonstration uses an example taken from Kmenta (1986, p. 685). We want to estimate a small model of US the food market:

$$\texttt{consump} = \beta_1 + \beta_2 * \texttt{price} + \beta_3 * \texttt{income} \tag{68}$$

$$\texttt{consump} = \beta_4 + \beta_5 * \texttt{price} + \beta_6 * \texttt{farmPrice} + \beta_7 * \texttt{trend} \tag{69}$$

The first equation represents the demand side of the food market. Variable `consump` (food consumption per capita) is the dependant variable. The regressors are `price` (ratio of food prices to general consumer prices) and `income` (disposable income) as well as a constant. The second equation specifies the supply side of the food market. Variable `consump` is the dependant variable of this equation as well. The regressors are again `price` (ratio of food prices to general consumer prices) and a constant as well as `farmPrice` (ratio of preceding year's prices received by farmers) and `trend` (a time trend in years). These equations can be estimated as SUR in R by

```
> library(systemfit)
> data(Kmenta)
> attach(Kmenta)
> eqDemand <- consump ~ price + income
> eqSupply <- consump ~ price + farmPrice + trend
> fitsur <- systemfit("SUR", list(demand = eqDemand, supply = eqSupply))
```

The first line loads the **systemfit** package. The second line loads example data that are included with the package. They are attached to the R search path in line three. In the fourth and fifth line, the demand and supply equations are specified, respectively[5]. Finally, in the last line, a seemingly unrelated regression is performed and the regression result is assigned to the variable `fitsur`.

Summary results can be printed by

```
> summary(fitsur)

systemfit results
method: SUR

        N DF      SSR     MSE    RMSE       R2   Adj R2
demand 20 17  65.6829 3.86370 1.96563 0.755019 0.726198
supply 20 16 104.0584 6.50365 2.55023 0.611888 0.539117

The covariance matrix of the residuals used for estimation
        demand  supply
demand 3.72539 4.13696
supply 4.13696 5.78444

The covariance matrix of the residuals
        demand  supply
demand 3.86370 4.92431
supply 4.92431 6.50365

The correlations of the residuals
```

[4] For Details see the R help files to `formula`.

[5] A regression constant is always implied if not explicitly omitted.

```
         demand   supply
demand 1.000000 0.982348
supply 0.982348 1.000000

The determinant of the residual covariance matrix: 0.879285
OLS R-squared value of the system: 0.683453
McElroy's R-squared value for the system: 0.788722

SUR estimates for 'demand' (equation 1)
Model Formula: consump ~ price + income

             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 99.332894   7.514452 13.218913        0 ***
price       -0.275486   0.088509 -3.112513 0.006332  **
income        0.29855   0.041945  7.117605    2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.96563 on 17 degrees of freedom
Number of observations: 20 Degrees of Freedom: 17
SSR: 65.682902 MSE: 3.8637 Root MSE: 1.96563
Multiple R-Squared: 0.755019 Adjusted R-Squared: 0.726198


SUR estimates for 'supply' (equation 2)
Model Formula: consump ~ price + farmPrice + trend

             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 61.966166   11.08079 5.592215    4e-05 ***
price        0.146884   0.094435 1.555397 0.139408
farmPrice    0.214004   0.039868 5.367761  6.3e-05 ***
trend        0.339304   0.067911 4.996283 0.000132 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.550226 on 16 degrees of freedom
Number of observations: 20 Degrees of Freedom: 16
SSR: 104.05843 MSE: 6.503652 Root MSE: 2.550226
Multiple R-Squared: 0.611888 Adjusted R-Squared: 0.539117
```

First, the estimation method is reported and a few summary statistics for each equation are given. Then, some results regarding the whole equation system are printed: covariance matrix and correlations of the residuals, log of the determinant of the residual covariance matrix, R^2 value of the whole system, and McElroy's R^2 value. If the model is estimated by (W)SUR or (W)3SLS, the covariance matrix used for estimation is printed additionally. Finally, the estimation results of each equation are reported: formula of the estimated equation, estimated parameters, their standard errors, t-values, p-values and codes indicating their statistical significance, and some other statistics like standard error of the residuals or R^2 value of the equation.

4.2 User options of `systemfit`

The user can modify the default estimation method by providing additional optional arguments, e.g. to specify instrumental variables or to impose parameter restrictions. All optional arguments are described in the following:

4.2.1 Equation labels

The optional argument `eqnlabels` allows the user to label the equations. It has to be a vector of strings. If this argument is not provided, the labels are taken from the names of the equations in argument `eqns`. And if the equations have no names, they are numbered consecutively. Hence, the following command has the same effect as the command above.

```
> fitsur <- systemfit("SUR", list(eqDemand, eqSupply), eqnlabels = c("demand",
+     "supply"))
```

4.2.2 Instrumental variables

The instruments for a 2SLS, W2SLS or 3SLS estimation can be specified by the argument `inst`. If the same instruments should be used for all equations, `inst` must be a one-sided formula. If different instruments should be used for the equations, `inst` must be a list that contains a one-sided formula for each equation. The first example uses the same instruments for all equations, and the second uses different instruments:

```
> fit3sls <- systemfit("3SLS", list(demand = eqDemand, supply = eqSupply),
+     inst = ~income + farmPrice + trend)
> fit3sls2 <- systemfit("3SLS", list(demand = eqDemand, supply = eqSupply),
+     inst = list(~farmPrice + trend, ~income + farmPrice + trend))
```

4.2.3 Data

Having all data in the global environment or attached to the search path is often inconvenient. Therefore, a data frame `data` can be provided that contains the variables of the model. In the following example we do not need to attach the data frame `Kmenta` before calling systemfit:

```
> fitsur <- systemfit("SUR", list(eqDemand, eqSupply), data = Kmenta)
```

4.2.4 Parameter restrictions

As outlined in section 2.2, parameter restrictions can be imposed in two ways. The first way is to use the transformation matrix T that can be specified by the argument `TX`. If we want to impose the restriction, say $\beta_2 = -\beta_6$, we can do this as follows

```
> tx <- matrix(0, nrow = 7, ncol = 6)
> tx[1, 1] <- 1
> tx[2, 2] <- 1
> tx[3, 3] <- 1
> tx[4, 4] <- 1
> tx[5, 5] <- 1
> tx[6, 2] <- -1
> tx[7, 6] <- 1
> fitsurTx <- systemfit("SUR", list(eqDemand, eqSupply), TX = tx)
```

The first line creates a 7×6 matrix of zeros, where 7 is the number of unrestricted coefficients and 6 is the number of restricted coefficients. The following seven lines specify this matrix in a way that the unrestricted coefficients (β) are assigned to the restricted coefficients (β^*) with $\beta_1 = \beta_1^*$, $\beta_2 = \beta_2^*$, $\beta_3 = \beta_3^*$, $\beta_4 = \beta_4^*$, $\beta_5 = \beta_5^*$, $\beta_6 = -\beta_2^*$, and $\beta_7 = \beta_6^*$. Finally the model is estimated with restriction $\beta_2^* = \beta_2 = -\beta_6$ imposed.

The second way to impose parameter restrictions is to use the matrix R and the vector q (see section 2.2). Matrix R can be specified with the argument `R.restr` and vector q with argument `q.restr`. We convert the restriction specified above to $\beta_2 + \beta_6 = 0$ and impose it in the second way:

```
> Rmat <- matrix(0, nrow = 1, ncol = 7)
> Rmat[1, 2] <- 1
> Rmat[1, 6] <- 1
> qvec <- c(0)
> fitsurRmat <- systemfit("SUR", list(eqDemand, eqSupply), R.restr = Rmat,
+     q.restr = qvec)
```

The first line creates a 1×7 matrix of zeros, where 1 is the number of restrictions and 7 is the number of unrestricted coefficients. The following two lines specify this matrix in a way that the multiplication with the parameter vector results in $\beta_2 + \beta_6$. The fourth line creates a vector with a single element that contains the left hand side of the restriction, i.e. zero. Finally the model is estimated with restriction $\beta_2 + \beta_6 = 0$ imposed.

4.2.5 Iteration control

The estimation methods WLS, SUR, W2SLS and 3SLS need a covariance matrix of the residuals that can be calculated from a first-step OLS or 2SLS estimation (see section 2.3). If the argument `maxiter` is set to a number large than one, this procedure is iterated and at each iteration the covariance matrix is calculated from the previous step estimation. This iteration is repeated until the maximum number of iterations is reached or the parameter estimates have converged. The maximum number of iterations is specified by the argument `maxiter`. Its default value is one, which means no iteration. The convergence criterion is

$$\sqrt{\frac{\sum_i (b_{i,g} - b_{i,g-1})^2}{\sum_i b_{i,g-1}^2}} < \texttt{tol} \tag{70}$$

where $b_{i,g}$ is the ith coefficient of the gth iteration. The default value of `tol` is 10^{-5}.

4.2.6 Residual covariance matrix

It was explained in section 2.3 that several different formulas have been proposed to calculate the residual covariance matrix. The user can specify, which formula **systemfit** should use. Possible values of the argument `rcovformula` are presented in table 1. By default, **systemfit** uses equation (57).

Table 1: Possible values of argument `rcovformula`

argument `rcovformula`	equation
0	56
1 or 'geomean'	57
2 or 'Theil'	59
3 or 'max'	58

Furthermore, the user can specify whether the means should be subtracted from the residuals before (56), (57), (58), or (59) are applied to calculate the residual covariance matrix (see section 2.3). The corresponding argument is called `centerResiduals`. It must be either "`TRUE`" (subtract the means) or "`FALSE`" (take the unmodified residuals). The default value of `centerResiduals` is "`FALSE`".

4.2.7 3SLS formula

As discussed in sections 2.1 and 2.2, there exist several different formulas to perform a 3SLS estimation. The user can specify the applied formula by the argument `formula3sls`. Possible values are presented in table 2. The default value is 'GLS'.

Table 2: Possible values of argument `formula3sls`

argument `formula3sls`	equation (unrestricted)	equation (restricted)
'GLS'	20	47
'IV'	23	49
'GMM'	25	51
'Schmidt'	27	53
'EViews'	29	55

4.2.8 Degrees of freedom for t-tests

There exist two different approaches to determine the degrees of freedom to perform t-tests on the estimated parameters (section 2.4). This can be specified with argument `probdfsys`. If it is `TRUE`, the degrees of freedom of the whole system are taken. In contrast, if `probdfsys` is `FALSE`, the degrees of freedom of the single equation are taken. By default, `probdfsys` is `TRUE`, if any restrictions are specified using either the argument `R.restr` or the argument `TX`, and it is `FALSE` otherwise.

4.2.9 Sigma squared

In case of OLS or 2SLS estimations, argument `single.eq.sigma` can be used to specify, whether different σ^2s for each single equation or the same σ^2 for all equations should be used. If argument `single.eq.sigma` is `TRUE`, equations (9) and (17) are applied. In contrast, if argument `single.eq.sigma` is `FALSE`, equations (8) and (16) are applied. By default, `single.eq.sigma` is `FALSE`, if any restrictions are specified using either the argument `R.restr` or the argument `TX`, and it is `TRUE` otherwise.

4.2.10 System options

Finally, two options regarding some internal calculations are available. First, argument `solvetol` specifies the tolerance level for detecting linear dependencies when inverting a matrix or calculating a determinant (using functions `solve` and `det`). The default value depends on the used computer system and is equal to the default tolerance level of `solve` and `det`. Second, argument `saveMemory` can be used in case of large data sets to accelerate the estimation by omitting some calculation that are not crucial for the basic estimation. Currently, only the calculation of McElroy's R^2 is omitted. The default value of argument `saveMemory` is `TRUE`, if the argument `data` is specified and the number of observations times the number of equations is larger than 1000, and it is `FALSE` otherwise.

4.3 The wrapper function `systemfitClassic`

The wrapper function `systemfitClassic` can be applied to (classical) panel-like data in long format[6] if the regressors are the same for all equations. This function is called by

```
> systemfitClassic(method, formula, eqnVar, timeVar, data)
```

[6] Panel data can be either in "long format" (different individuals are arranged below each other) or in "wide format" (different individuals are arranged next to each other).

The mandatory arguments are `method`, `formula`, `eqnVar`, and `timeVar`. Argument `method` is the same as in `systemfit` (see section 4.1). The second argument `formula` is a standard formula in R that will be applied to all equations. Argument `eqnVar` specifies the variable name indicating the equation to which the observation belongs, and argument `timeVar` specifies the variable name indicating the time. Finally, `data` is a `data.frame` that contains all required data.

We demonstrate the usage of `systemfitClassic` using an example taken from Theil (1971, pp. 295, 300) that is based on Grunfeld (1958). We want to estimate a model for gross investment of 2 US firms in the years 1935–1954:

$$\texttt{invest}_{it} = \beta_1 + \beta_2 * \texttt{value}_{it} + \beta_3 * \texttt{capital}_{it} \tag{71}$$

where `invest` is the gross investment of firm i in year t, `value` is the market value of the firm at the end of the previous year, and `capital` is the capital stock of the firm at the end of the previous year.

This model can be estimated by

```
> data("GrunfeldTheil")
> theilSur <- systemfitClassic("SUR", invest ~ value + capital,
+     "firm", "year", data = GrunfeldTheil)
```

The first line loads example data that are included with the package. And then, a seemingly unrelated regression is performed, where the variable "firm" indicates the firm and the variable "year" indicates the time.

The function `systemfitClassic` has also an optional argument `pooled` that is a logical variable indicating whether the coefficients should be restricted to be equal in all equations. By default, this argument is set to "`FALSE`". In addition all optional arguments of `systemfit` (see section 4.2) except for `eqnLabels` and `TX` can be used with `systemfitClassic`, too.

4.4 Testing linear restrictions

As described in section 2.6, linear restrictions can be tested by an F test, Wald test or LR test. The corresponding functions in package **systemfit** are `ftest.systemfit`, `waldtest.systemfit`, and `lrtest.systemfit`, respectively.

We will now test the restriction $\beta_2 = -\beta_6$ that was specified by the matrix `Rmat` and the vector `qvec` in the example above (p. 16).

```
> ftest.systemfit(fitsur, Rmat, qvec)

F-test for linear parameter restrictions in equation systems
F-statistic: 0.9322
degrees of freedom of the numerator: 1
degrees of freedom of the denominator: 33
p-value: 0.3413

> waldtest.systemfit(fitsur, Rmat, qvec)

Wald-test for linear parameter restrictions in equation systems
Wald-statistic: 0.6092
degrees of freedom: 1
p-value: 0.4351

> lrtest.systemfit(fitsurRmat, fitsur)

Likelihood-Ratio-test for parameter restrictions in equation systems
LR-statistic: 1.004
degrees of freedom: 1
p-value: 0.3163
```

The linear restrictions are tested by an F test first, then by a Wald test, and finally by an LR test. The functions `ftest.systemfit` and `waldtest.systemfit` have three arguments. The first argument must be an unrestricted regression returned by `systemfit`. The second and third argument are the restriction matrix R and the vector q as described in section 2.2. In contrast, the function `lrtest.systemfit` needs two arguments. The first argument must be a restricted and the second an unrestricted regression returned by `systemfit`.

All tests print a short explanation first. Then the empirical test statistic and the degree(s) of freedom are reported. Finally the p-value is printed. While there is some variation of the p-values across the three different tests, all tests suggest the same decision: The null hypothesis $\beta_2 = -\beta_6$ cannot be rejected at any reasonable level of significance.

4.5 Hausman test

A Hausman test, which is described in section 2.7, can be carried out with following commands:

```
> fit2sls <- systemfit("2SLS", list(demand = eqDemand, supply = eqSupply),
+     inst = ~income + farmPrice + trend, data = Kmenta)
> fit3sls <- systemfit("3SLS", list(demand = eqDemand, supply = eqSupply),
+     inst = ~income + farmPrice + trend, data = Kmenta)
> hausman.systemfit(fit2sls, fit3sls)

        Hausman specification test for consistency of the 3SLS estimation

data:  Kmenta
Hausman = 2.5357, df = 7, p-value = 0.9244
```

First of all, the model is estimated by 2SLS and then by 3SLS. Finally, in the last line the test is carried out by the command `hausman.systemfit`. This function requires two arguments: the result of a 2SLS estimation and the result of a 3SLS estimation. The Hausman test statistic is 2.536, which has a χ^2 distribution with 7 degrees of freedom under the null hypothesis. The corresponding p-value is 0.924. This shows that the null hypothesis is not rejected at any reasonable level of significance. Hence, we can assume that the 3SLS estimator is consistent.

5 Testing reliability

In this section we test the reliability of the results from `systemfit` and `systemfitClassic`.

5.1 Kmenta (1986): Example on p. 685 (food market)

First, we reproduce an example taken from Kmenta (1986, p. 685). The data are available from Table 13-1 (p. 687), and the results are presented in Table 13-2 (p. 712) of this book.

Before starting the estimation, we load the data and specify the two formulas to estimate as well as the instrumental variables. Then the equation system ist estimated by OLS, 2SLS, 3SLS, and iterated 3SLS. After each estimation the estimated coefficients are reported.

```
> data("Kmenta")
> eqDemand <- consump ~ price + income
> eqSupply <- consump ~ price + farmPrice + trend
> inst <- ~income + farmPrice + trend
> system <- list(demand = eqDemand, supply = eqSupply)
```

OLS estimation:

```
> fitOls <- systemfit("OLS", system, data = Kmenta)
> round(coef(summary(fitOls)), digits = 4)
```

```
                     Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)  99.8954     7.5194 13.2851   0.0000
eq 1 price        -0.3163     0.0907 -3.4882   0.0028
eq 1 income        0.3346     0.0454  7.3673   0.0000
eq 2 (Intercept)  58.2754    11.4629  5.0838   0.0001
eq 2 price         0.1604     0.0949  1.6901   0.1104
eq 2 farmPrice     0.2481     0.0462  5.3723   0.0001
eq 2 trend         0.2483     0.0975  2.5462   0.0216
```

2SLS estimation:

```
> fit2sls <- systemfit("2SLS", system, inst = inst, data = Kmenta)
> round(coef(summary(fit2sls)), digits = 4)
```

```
                     Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)  94.6333     7.9208 11.9474   0.0000
eq 1 price        -0.2436     0.0965 -2.5243   0.0218
eq 1 income        0.3140     0.0469  6.6887   0.0000
eq 2 (Intercept)  49.5324    12.0105  4.1241   0.0008
eq 2 price         0.2401     0.0999  2.4023   0.0288
eq 2 farmPrice     0.2556     0.0473  5.4096   0.0001
eq 2 trend         0.2529     0.0997  2.5380   0.0219
```

3SLS estimation:

```
> fit3sls <- systemfit("3SLS", system, inst = inst, data = Kmenta)
> round(coef(summary(fit3sls)), digits = 4)
```

```
                     Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)  94.6333     7.9208 11.9474   0.0000
eq 1 price        -0.2436     0.0965 -2.5243   0.0218
eq 1 income        0.3140     0.0469  6.6887   0.0000
eq 2 (Intercept)  52.1972    11.8934  4.3888   0.0005
eq 2 price         0.2286     0.0997  2.2934   0.0357
eq 2 farmPrice     0.2282     0.0440  5.1861   0.0001
eq 2 trend         0.3611     0.0729  4.9546   0.0001
```

Iterated 3SLS estimation:

```
> fitI3sls <- systemfit("3SLS", system, inst = inst, data = Kmenta,
+     maxit = 250)
> round(coef(summary(fitI3sls)), digits = 4)
```

```
                     Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)  94.6333     7.9208 11.9474   0.0000
eq 1 price        -0.2436     0.0965 -2.5243   0.0218
eq 1 income        0.3140     0.0469  6.6887   0.0000
eq 2 (Intercept)  52.6618    12.8051  4.1126   0.0008
eq 2 price         0.2266     0.1075  2.1086   0.0511
eq 2 farmPrice     0.2234     0.0468  4.7756   0.0002
eq 2 trend         0.3800     0.0720  5.2771   0.0001
```

The results above show that `systemfit` returns exactly the same coefficients and standard errors as published in Kmenta (1986, p. 712) except for two minor exemptions. Two standard errors of the 2SLS estimation deviate by 0.0001. However, this difference is likely due to rounding errors in `systemfit` or Kmenta (1986) and is so small that it empirically doesn't matter.

5.2 Greene (2003): Example 15.1 (Klein's model I)

Second, we try to replicate Klein's Model I (Klein, 1950) that is described in Greene (2003, pp. 381). The data are available from the online complements to Greene (2003), Table F15.1 (http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm), and the estimation results are presented in Table 15.3 (p. 412).

Initially, the data are loaded and three equations as well as the instrumental variables are specified. As in the example before, the equation system ist estimated by OLS, 2SLS, 3SLS, and iterated 3SLS, and estimated coefficients of each estimation are reported.

```
> data("KleinI")
> eqConsump <- consump ~ corpProf + corpProfLag + wages
> eqInvest <- invest ~ corpProf + corpProfLag + capitalLag
> eqPrivWage <- privWage ~ gnp + gnpLag + trend
> inst <- ~govExp + taxes + govWage + trend + capitalLag + corpProfLag +
+     gnpLag
> system <- list(Consumption = eqConsump, Investment = eqInvest,
+     "Private Wages" = eqPrivWage)
```

OLS estimation:

```
> kleinOls <- systemfit("OLS", system, data = KleinI)
> round(coef(summary(kleinOls)), digits = 3)
```

```
                 Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)   16.237      1.303  12.464    0.000
eq 1 corpProf       0.193      0.091   2.115    0.049
eq 1 corpProfLag    0.090      0.091   0.992    0.335
eq 1 wages          0.796      0.040  19.933    0.000
eq 2 (Intercept)   10.126      5.466   1.853    0.081
eq 2 corpProf       0.480      0.097   4.939    0.000
eq 2 corpProfLag    0.333      0.101   3.302    0.004
eq 2 capitalLag    -0.112      0.027  -4.183    0.001
eq 3 (Intercept)    1.497      1.270   1.179    0.255
eq 3 gnp            0.439      0.032  13.561    0.000
eq 3 gnpLag         0.146      0.037   3.904    0.001
eq 3 trend          0.130      0.032   4.082    0.001
```

2SLS estimation:

```
> klein2sls <- systemfit("2SLS", system, inst = inst, data = KleinI,
+     rcovformula = 0)
> round(coef(summary(klein2sls)), digits = 3)
```

```
                 Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)   16.555      1.321  12.534    0.000
eq 1 corpProf       0.017      0.118   0.147    0.885
eq 1 corpProfLag    0.216      0.107   2.016    0.060
eq 1 wages          0.810      0.040  20.129    0.000
eq 2 (Intercept)   20.278      7.543   2.688    0.016
eq 2 corpProf       0.150      0.173   0.867    0.398
eq 2 corpProfLag    0.616      0.163   3.784    0.001
eq 2 capitalLag    -0.158      0.036  -4.368    0.000
eq 3 (Intercept)    1.500      1.148   1.307    0.209
eq 3 gnp            0.439      0.036  12.316    0.000
eq 3 gnpLag         0.147      0.039   3.777    0.002
eq 3 trend          0.130      0.029   4.475    0.000
```

3SLS estimation:

```
> klein3sls <- systemfit("3SLS", system, inst = inst, data = KleinI,
+     rcovformula = 0)
> round(coef(summary(klein3sls)), digits = 3)
```

```
                 Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)   16.441      1.305  12.603    0.000
eq 1 corpProf       0.125      0.108   1.155    0.264
eq 1 corpProfLag    0.163      0.100   1.624    0.123
eq 1 wages          0.790      0.038  20.826    0.000
eq 2 (Intercept)   28.178      6.794   4.148    0.001
eq 2 corpProf      -0.013      0.162  -0.081    0.937
eq 2 corpProfLag    0.756      0.153   4.942    0.000
eq 2 capitalLag    -0.195      0.033  -5.990    0.000
eq 3 (Intercept)    1.797      1.116   1.611    0.126
eq 3 gnp            0.400      0.032  12.589    0.000
eq 3 gnpLag         0.181      0.034   5.307    0.000
eq 3 trend          0.150      0.028   5.358    0.000
```

iterated 3SLS estimation:

```
> kleinI3sls <- systemfit("3SLS", system, inst = inst, data = KleinI,
+     rcovformula = 0, maxit = 500)
> round(coef(summary(kleinI3sls)), digits = 3)
```

```
                 Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)   16.559      1.224  13.524    0.000
eq 1 corpProf       0.165      0.096   1.710    0.105
eq 1 corpProfLag    0.177      0.090   1.960    0.067
eq 1 wages          0.766      0.035  22.031    0.000
eq 2 (Intercept)   42.896     10.594   4.049    0.001
eq 2 corpProf      -0.357      0.260  -1.370    0.188
eq 2 corpProfLag    1.011      0.249   4.065    0.001
eq 2 capitalLag    -0.260      0.051  -5.115    0.000
eq 3 (Intercept)    2.625      1.196   2.195    0.042
eq 3 gnp            0.375      0.031  12.050    0.000
eq 3 gnpLag         0.194      0.032   5.977    0.000
eq 3 trend          0.168      0.029   5.805    0.000
```

Again, the results show that `systemfit` returns the same results as published in Greene (2003).[7] Also in this case we have two minor deviations, where only the last digit is different.

5.3 Theil (1971): Example on p. 295 (General Electric and Westinghouse)

Third, we estimate an example taken from Theil (1971, p. 295) that is based on Grunfeld (1958). The data are available from Table 7.1 (p. 296), and the results are presented on pages 295 and 300 of this book.

After loading the data and specifying the formula, the model is estimated by OLS and SUR. The coefficients of each estimation are reported.

```
> data("GrunfeldTheil")
> formulaGrunfeld <- invest ~ value + capital
```

[7] There are two typos in Table 15.3 (p. 412). Please take a look at the errata (`http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm`).

OLS estimation (page 295)

```
> theilOls <- systemfitClassic("OLS", formulaGrunfeld, "firm",
+     "year", data = GrunfeldTheil)
> round(coef(summary(theilOls)), digits = 3)

                            Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)              -9.956     31.374  -0.317    0.755
eq 1 value.General.Electric    0.027      0.016   1.706    0.106
eq 1 capital.General.Electric  0.152      0.026   5.902    0.000
eq 2 (Intercept)              -0.509      8.015  -0.064    0.950
eq 2 value.Westinghouse        0.053      0.016   3.368    0.004
eq 2 capital.Westinghouse      0.092      0.056   1.647    0.118
```

SUR estimation (page 300)

```
> theilSur <- systemfitClassic("SUR", formulaGrunfeld, "firm",
+     "year", data = GrunfeldTheil, rcovformula = 0)
> round(coef(summary(theilSur)), digits = 3)

                            Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)             -27.719     27.033  -1.025    0.320
eq 1 value.General.Electric    0.038      0.013   2.883    0.010
eq 1 capital.General.Electric  0.139      0.023   6.036    0.000
eq 2 (Intercept)              -1.252      6.956  -0.180    0.859
eq 2 value.Westinghouse        0.058      0.013   4.297    0.000
eq 2 capital.Westinghouse      0.064      0.049   1.308    0.208
```

The function `systemfitClassic`, which is a wrapper function to `systemfit` returns exactly the same results as published in Theil (1971, pp. 295, 300).

5.4 Greene (2003): Example 14.1 (Grunfeld's investment data)

Finally, we reproduce Example 14.1 of Greene (2003, p. 340) that is also based on Grunfeld (1958). The data are available from the online complements to Greene (2003), Table F13.1 (http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm), and the estimation results are presented in Tables 14.1 and 14.2 (p. 351).

First, we load the data and specify the formula to estimate. Then, the systems is estimated by OLS, pooled OLS, SUR, and pooled SUR. Immediately after each estimation, the estimated coefficients are reported. Furthermore, the σ^2 values of the OLS estimations, and the residual covariance matrix as well as the residual correlation matrix of the SUR estimations are printed.

```
> data("GrunfeldGreene")
> formulaGrunfeld <- invest ~ value + capital
```

OLS estimation (Table 14.2):

```
> greeneOls <- systemfitClassic("OLS", formulaGrunfeld, "firm",
+     "year", data = GrunfeldGreene)
> round(coef(summary(greeneOls)), digits = 4)

                            Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)             -6.1900    13.5065 -0.4583   0.6525
eq 1 value.Chrysler           0.0779     0.0200  3.9026   0.0011
eq 1 capital.Chrysler         0.3157     0.0288 10.9574   0.0000
eq 2 (Intercept)             -9.9563    31.3742 -0.3173   0.7548
eq 2 value.General.Electric   0.0266     0.0156  1.7057   0.1063
eq 2 capital.General.Electric 0.1517     0.0257  5.9015   0.0000
```

```
eq 3 (Intercept)            -149.7825   105.8421 -1.4151   0.1751
eq 3 value.General.Motors      0.1193     0.0258  4.6172   0.0002
eq 3 capital.General.Motors    0.3714     0.0371 10.0193   0.0000
eq 4 (Intercept)             -30.3685   157.0477 -0.1934   0.8490
eq 4 value.US.Steel            0.1566     0.0789  1.9848   0.0635
eq 4 capital.US.Steel          0.4239     0.1552  2.7308   0.0142
eq 5 (Intercept)              -0.5094     8.0153 -0.0636   0.9501
eq 5 value.Westinghouse        0.0529     0.0157  3.3677   0.0037
eq 5 capital.Westinghouse      0.0924     0.0561  1.6472   0.1179
```

```
> round(sapply(greeneOls$eq, function(x) {
+     return(x$ssr/20)
+ }), digits = 3)
```

```
[1]  149.872  660.829 7160.294 8896.416   88.662
```

pooled OLS (Table 14.2):

```
> greeneOlsPooled <- systemfitClassic("OLS", formulaGrunfeld, "firm",
+     "year", data = GrunfeldGreene, pooled = TRUE)
> round(coef(summary(greeneOlsPooled$eq[[1]])), digits = 4)
```

```
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      -48.0297    21.4802 -2.2360   0.0276
value.Chrysler     0.1051     0.0114  9.2360   0.0000
capital.Chrysler   0.3054     0.0435  7.0186   0.0000
```

```
> sum(sapply(greeneOlsPooled$eq, function(x) {
+     return(x$ssr)
+ }))/100
```

```
[1] 15708.84
```

SUR estimation (Table 14.1):

```
> greeneSur <- systemfitClassic("SUR", formulaGrunfeld, "firm",
+     "year", data = GrunfeldGreene, rcovformula = 0)
> round(coef(summary(greeneSur)), digits = 4)
```

```
                              Estimate Std. Error t value Pr(>|t|)
eq 1 (Intercept)                0.5043    11.5128  0.0438   0.9656
eq 1 value.Chrysler             0.0695     0.0169  4.1157   0.0007
eq 1 capital.Chrysler           0.3085     0.0259 11.9297   0.0000
eq 2 (Intercept)              -22.4389    25.5186 -0.8793   0.3915
eq 2 value.General.Electric     0.0373     0.0123  3.0409   0.0074
eq 2 capital.General.Electric   0.1308     0.0220  5.9313   0.0000
eq 3 (Intercept)             -162.3641    89.4592 -1.8150   0.0872
eq 3 value.General.Motors       0.1205     0.0216  5.5709   0.0000
eq 3 capital.General.Motors     0.3827     0.0328 11.6805   0.0000
eq 4 (Intercept)               85.4233   111.8774  0.7635   0.4556
eq 4 value.US.Steel             0.1015     0.0548  1.8523   0.0814
eq 4 capital.US.Steel           0.4000     0.1278  3.1300   0.0061
eq 5 (Intercept)                1.0889     6.2588  0.1740   0.8639
eq 5 value.Westinghouse         0.0570     0.0114  5.0174   0.0001
eq 5 capital.Westinghouse       0.0415     0.0412  1.0074   0.3279
```

```
> round(greeneSur$rcov, digits = 3)
```

```
                  Chrysler General Electric General Motors  US Steel
Chrysler           152.849            2.047       -313.704   455.089
General Electric     2.047          700.456        605.336  1224.405
General Motors    -313.704          605.336       7216.044 -2686.517
US Steel           455.089         1224.405      -2686.517  9188.151
Westinghouse        16.661          200.316        129.887   652.716
                  Westinghouse
Chrysler                16.661
General Electric       200.316
General Motors         129.887
US Steel               652.716
Westinghouse            94.912
```

```
> round(summary(greeneSur)$rcor, digits = 3)
```

```
                  Chrysler General Electric General Motors US Steel Westinghouse
Chrysler             1.000            0.006         -0.299    0.384        0.138
General Electric     0.006            1.000          0.269    0.483        0.777
General Motors      -0.299            0.269          1.000   -0.330        0.157
US Steel             0.384            0.483         -0.330    1.000        0.699
Westinghouse         0.138            0.777          0.157    0.699        1.000
```

pooled SUR estimation (Table 14.1):

```
> greeneSurPooled <- systemfitClassic("WSUR", formulaGrunfeld,
+     "firm", "year", data = GrunfeldGreene, pooled = TRUE, rcovformula = 0)
> round(coef(summary(greeneSurPooled$eq[[1]])), digits = 4)
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -28.2467     4.8882 -5.7785        0
value.Chrysler      0.0891     0.0051 17.5663        0
capital.Chrysler    0.3340     0.0167 19.9859        0
```

```
> round(greeneSurPooled$rcov, digits = 3)
```

```
                   Chrysler General Electric General Motors   US Steel
Chrysler            305.610        -1966.648         -4.805   2158.595
General Electric  -1966.648        34556.603      -7160.667 -28722.006
General Motors       -4.805        -7160.667      10050.525   4439.989
US Steel           2158.595       -28722.006       4439.989  34468.976
Westinghouse       -123.920         4274.000      -1400.747  -2893.733
                  Westinghouse
Chrysler              -123.920
General Electric      4274.000
General Motors       -1400.747
US Steel             -2893.733
Westinghouse           833.357
```

```
> round(cov(residuals(greeneSurPooled)), digits = 3)
```

```
                  Chrysler General Electric General Motors  US Steel
Chrysler           167.403          174.038       -432.370   260.431
General Electric   174.038         3733.260      -1322.236  -972.145
General Motors    -432.370        -1322.236       9396.058  -897.950
US Steel           260.431         -972.145       -897.950 10052.045
Westinghouse        89.336         1302.231       -865.791  -180.385
                  Westinghouse
```

```
Chrysler               89.336
General Electric     1302.231
General Motors       -865.791
US Steel             -180.385
Westinghouse          564.156
```

```
> round(summary(greeneSurPooled)$rcor, digits = 3)
```

```
                 Chrysler General Electric General Motors US Steel Westinghouse
Chrysler            1.000            0.220         -0.345    0.201        0.291
General Electric    0.220            1.000         -0.223   -0.159        0.897
General Motors     -0.345           -0.223          1.000   -0.092       -0.376
US Steel            0.201           -0.159         -0.092    1.000       -0.076
Westinghouse        0.291            0.897         -0.376   -0.076        1.000
```

For this example, the function `systemfitClassic` returns nearly the same results as published in Greene (2003).[8] Two different residual covariance matrices of the pooled SUR estimation are presented. The first is calculated without centering the results (see section 2.3). It is equal to the one published in the book (Greene, 2003, p. 351). The second residual covariance matrix is calculated after centering the results. It is equal to the one published in the errata (`http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm`).

6 Summary and outlook

In this article, we have described some of the basic features of the **systemfit** package for estimation of linear systems of equations. Many details of the estimation can be controlled by the user. Furthermore, the package provides some statistical tests for parameter restrictions and consistency of 3SLS estimation. It has been tested on a variety of datasets and has produced satisfactory for a few years. While the **systemfit** package performs the basic fitting methods, more sophisticated tools exist. We hope to implement missing functionalities in the near future.

Unbalanced datasets

Currently, the **systemfit** package requires that all equations have the same number of observations. However, many data sets have unbalanced observations.[9] Simply dropping data points that do not contain observations for all equations may reduce the number of observations considerably, and thus, the information utilized in the estimation. Hence, it is our intention to include the capability for estimations with unbalanced data sets as described in Schmidt (1977) in future releases of **systemfit**.

Serial correlation and heteroscedasticity

For all of the methods developed in the package, the disturbances of the individual equations are assumed to be independent and identically distributed (iid). The package could be enhanced by the inclusion of methods to fit equations with serially correlated and heteroscedastic disturbances (Parks, 1967).

[8] There are several typos and errors in Table 14.1 (p. 412). Please take a look at the errata of this book (`http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm`).

[9] For instance, forestry datasets typically contain many observations of inexpensive variables (stem diameter, tree count) and few expensive variables such as stem height or volume.

Estimation methods

In the future, we wish to include more sophisticated estimation methods such as limited-information maximum likelihood (LIML), full-information maximum likelihood (FIML), generalized methods of moments (GMM) and spatial econometric methods.

Non-linear estimation

Finally, the **systemfit** package provides a function to estimate systems of non-linear estimations. However, the function `nlsystemfit` is currently under development and the results are not yet always reliable due to convergence difficulties.

References

Bates D (2004). "Least Squares Calculations in R." *R News*, **4**(1), 17–20. URL http://CRAN.R-project.org/doc/Rnews/.

Greene WH (2003). *Econometric Analysis*. Prentice Hall, 5th edition.

Greene WH (2006). "Information about SUR estimation in LIMDEP." Personal email on 2006/02/16.

Grunfeld Y (1958). *The Determinants of Corporate Investment*. Ph.D. thesis, University of Chicago.

Hausman JA (1978). "Specification Test in Econometrics." *Econometrica*, **46**, 1251–1272.

Judge GG, Griffiths WE, Hill RC, Lütkepohl H, Lee TC (1985). *The Theory and Practice of Econometrics*. John Wiley and Sons, 2nd edition.

Klein L (1950). *Economic Fluctuations in the United States, 1921–1941*. John Wiley, New York.

Kmenta J (1986). *Elements of econometrics*. Macmillan, New York, 2 edition.

McElroy MB (1977). "Goodness of Fit for Seemingly Unrelated Regressions." *Journal of Econometrics*, **6**, 381–387.

Parks RW (1967). "Effcient estimation of a system of regression equations when disturbances are both serially and contemporaneously correlated." *Journal of the American Statistical Association*, **62**, 500–509.

R Development Core Team (2005). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Schmidt P (1977). "Estimation of Seemingly Unrelated Regressions with Unequal Numbers of Observations." *Journal of Econometrics*, **5**, 365–377.

Schmidt P (1990). "Three-Stage Least Squares with Different Instruments for Different Equations." *Journal of Econometrics*, **43**, 389–394.

Theil H (1971). *Principles of Econometrics*. Wiley, New York.

Zellner A (1962). "An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias." *Journal of the American Statistical Association*, **57**, 348–368.

Zellner A, Huang DS (1962). "Further Properties of Efficient Estimators for Seemingly Unrelated Regression Equations." *International Economic Review*, **3**(3), 300–313.

Zellner A, Theil H (1962). "Three-Stage Least Squares: Simultaneous Estimation of Simultaneous Equations." *Econometrica*, **30**(1), 54–78.