Data Estimation Methods
Application with Mathematica

Tugrul Temel
Development Research Institute (IVO)
Tilburg University, Tilburg
The Netherlands
t.temel@uvt.nl
5 November 2007

November 6, 2007

Abstract

This paper presents 4 data estimation methods and illustrates their application using *Mathematica* program. The methods include the Bayesian estimation, least-squares estimation (restricted and unrestricted estimations), best linear unbiased estimation and linear programming. The paper has been prepared for use in training programs within IVO's projects in developing countries.

Key words: system of national accounts, data compilation, Bayesian estimation, least-squares estimation, best linear unbiased estimation, linear programming

1 Introduction

This paper presents 4 methods for data estimation and illustrates their application using *Mathematica* (version 6). The methods include the Bayesian estimation, least-squares estimation (restricted and unrestricted estimations), best linear unbiased estimation and linear programming. The paper has been prepared for use in training programs within IVO's projects in developing countries.

The paper is divided into 5 sections. Following this brief Introduction, Section 2 formulates the data estimation problem concerned. Section 3 describes the estimation problem within the context of an example system of linear equations. Section 4 first presents a mathematical description of the estimation problem and then develops a *Mathematica* program for its solution. Section 5 concludes the paper with some remarks on the efficiency of *Mathematica* for solving large linear systems.

2 The data estimation problem

We formulate the data estimation problem using an example system of linear equations. For clarity in the formulation of the problem, some notations and assumptions are in order:

- Notations

  $\tilde{x}_i^t$ denotes time t value of the i^{th} variable with $i = 1, 2, ..., \tilde{n}$ and $t = 0, 1$.

  $\tilde{x}^t = (\tilde{x}_1^t, \tilde{x}_2^t, ..., \tilde{x}_{\tilde{n}}^t)'$ is a column vector of $\tilde{n}$ variables at time t.

  $f_k(\tilde{x}^t) = 0$ defines the k^{th} identity as a function of $\tilde{x}^t$.

  $R_{ij}^0 = \left(\frac{\tilde{x}_i^0}{\tilde{x}_j^0}\right)$ is a prior indicator ratio evaluated using benchmark data (i.e., data at $t = 0$).

- Assumptions

  i) $\tilde{x}_i \sim N(\mu_i, \sigma_i^2)$ for all i.

  ii) $R_{ij}^0 = R_{ij}^1$ for $\tilde{m}_1$ indicator ratios. Reliability levels and coefficients for $\tilde{p}$ observations made at $t = 1$ and for $\tilde{m}_1$ indicator ratios: {Fixed = 0, Strong = 0.01, High = 0.03, Medium = 0.06, Low = 0.12, Poor = 0.24}.

  iii) $f_k(\tilde{x}^t) = 0$ holds for all t and all $k = 1, 2, ..., \tilde{m}_2$

- Data

  i) At $t = 0$, data are available on $\tilde{n}$ variables: $\tilde{x}^0 = (\tilde{x}_1^0, \tilde{x}_2^0, ..., \tilde{x}_{\tilde{n}}^0)'$.

  ii) At $t = 1$, $\tilde{m}_1$ indicator ratios and $\tilde{m}_2$ linear identities are available.

  iii) At $t = 1$, data are available on $\tilde{p}$ variables: $(\tilde{x}_1^1, \tilde{x}_2^1, ..., \tilde{x}_{\tilde{p}}^1)'$ with $\tilde{p} < \tilde{n}$.

- The problem is to estimate the mean and variance of $\tilde{x}^1$ given data and assumptions.

3 An example system of linear equations

3.1 Data and prior information

Suppose that at time $t = 0$ a benchmark (or reference) data set is available for a vector of $\tilde{n} = 10$ latent variables, denoted by:

$$\begin{aligned} \tilde{x}^0 &\equiv (\tilde{x}_1^0, \tilde{x}_2^0, \tilde{x}_3^0, \tilde{x}_4^0, \tilde{x}_5^0, \tilde{x}_6^0, \tilde{x}_7^0, \tilde{x}_8^0, \tilde{x}_9^0, \tilde{x}_{10}^0)' \\ &= (P^0, M^0, I^0, K^0, X^0, C^0, Y^0, R^0, S^0, B^0)' \\ &= (100, 90, 40, 30, 70, 50, 60, 60, 10, 20)'. \end{aligned}$$

At time $t = 1$, data are available only for 4 variables ($\tilde{p} = 4$):

$$\begin{aligned} (\tilde{x}_1^1, \tilde{x}_2^1, \tilde{x}_4^1, \tilde{x}_5^1)' &\equiv (P^1, M^1, K^1, X^1)' \\ &= (107.12, 93.64, 32.14, 74.98)' \end{aligned}$$

Furthermore, the benckmark values of the following 5 prior indicator ratios ($\tilde{m}_1 = 5$) are assumed to remain about the same over the period from $t = 0$ to $t = 1$, implying that the economy has been in the state of equilibrium during that period.

$$\begin{aligned} (\frac{I^0}{P^0}) &= \frac{40}{100} = 0.4 \\ (\frac{K^0}{P^0 + M^0}) &= \frac{30}{190} = 0.158 \\ (\frac{M^0}{P^0}) &= \frac{90}{100} = 0.9 \\ (\frac{C^0}{R^0}) &= \frac{50}{60} = 0.833 \\ (\frac{X^0}{P^0}) &= \frac{70}{100} = 0.7 \end{aligned}$$

As a last piece of information, we assume that the following 6 linear identities ($\tilde{m}_2 = 6$) hold across all t, reflecting the basic accounting relations between the $\tilde{n}$ variables:

$$\begin{aligned} Y^t - P^t + I^t &\equiv 0 \\ S^t - R^t + C^t &\equiv 0 \\ B^t - M^t + X^t &\equiv 0 \\ Y^t - R^t &\equiv 0 \\ K^t - S^t - B^t &\equiv 0 \\ P^t + M^t - I^t - C^t - K^t - X^t &\equiv 0. \end{aligned}$$

3.2 The system of linear equations

Data available ($\tilde{p} = 4$) at $t = 1$ are expressed as:

$$\begin{aligned}
\tilde{D}_1\tilde{x}^1 &= \tilde{d}_1 \text{ where} \\
\tilde{D}_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\tilde{x}^1 &= (P^1, M^1, I^1, K^1, X^1, C^1, Y^1, R^1, S^1, B^1)' \\
\tilde{d}_1 &= (107.12, 93.64, 32.14, 74.98)'
\end{aligned}$$

The "linearized" indicator ratios ($\tilde{m}_1 = 5$) assumed to hold for all t are expressed as

$$\begin{aligned}
\tilde{A}_1\tilde{x}^t &\equiv \tilde{h}_1 \text{ where} \\
\tilde{A}_1 &= \begin{bmatrix} -0.4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.158 & -0.158 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.833 & 0 & 0 \\ -0.7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\tilde{x}^t &= (P^t, M^t, I^t, K^t, X^t, C^t, Y^t, R^t, S^t, B^t)' \\
\tilde{h}_1 &= (0, 0, 0, 0, 0)'
\end{aligned}$$

The linear identities ($\tilde{m}_2 = 6$) are:

$$\begin{aligned}
\tilde{A}_2\tilde{x}^t &\equiv \tilde{h}_2 \text{ for all } t, \text{ where} \\
\tilde{A}_2 &= \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\tilde{x}^t &= (P^t, M^t, I^t, K^t, X^t, C^t, Y^t, R^t, S^t, B^t)' \\
\tilde{h}_2 &= (0, 0, 0, 0, 0, 0)'
\end{aligned}$$

3.3 The modified system of linear equations

Owing to the numerator $(P^t + M^t)$ of the 2^{nd} indicator ratio above, we define a composite variable $Z^t \equiv (P^t + M^t)$ where $Z^0 = 190$ and $Z^1 = 200.76$. The introduction of this composite variable requires some modifications in the linear system described in Step 2. The first modification takes place in $\tilde{D}_1\tilde{x}^1 = \tilde{d}_1$ as follows:

$$
\begin{aligned}
D_1 x^1 &= d_1 \quad \text{where} \\
D_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
x^1 &= (P^1, M^1, I^1, K^1, X^1, C^1, Y^1, R^1, S^1, B^1, Z^1)' \\
d_1 &= (107.12, 93.64, 32.14, 74.98, 200.76)'
\end{aligned}
$$

The second modification takes place in $\tilde{A}_1 \tilde{x}^t \equiv \tilde{h}_1$ as follows:

$$
\begin{aligned}
A_1 x^t &\equiv h_1 \quad \text{where} \\
A_1 &= \begin{bmatrix} -0.4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.158 \\ -0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.833 & 0 & 0 & 0 \\ -0.7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
x^t &= (P^t, M^t, I^t, K^t, X^t, C^t, Y^t, R^t, S^t, B^t, Z^t)' \\
h_1 &= (0, 0, 0, 0, 0)'
\end{aligned}
$$

The third modification takes place in $\tilde{A}_2 \tilde{x}^t \equiv \tilde{h}_2$ by introducing the new identity $Z^t \equiv (P^t + M^t)$:

$$
\begin{aligned}
A_2 x^t &\equiv h_2 \quad \text{where} \\
A_2 &= \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
x^t &\equiv (P^t, M^t, I^t, K^t, X^t, C^t, Y^t, R^t, S^t, B^t, Z^t)' \\
h_2 &= (0, 0, 0, 0, 0, 0, 0)'
\end{aligned}
$$

The resulting system is:

$$
\begin{aligned}
D_1 x^1 &= d_1 \text{ with } \; d_1 \mid x^t \sim N_p(D_1 x^t, \Sigma_1) \\
A_1 x^1 &\equiv h_1 \text{ with } \; A_1 \sim N_{m_1}(h_1, H_1) \\
A_2 x^1 &= h_2 \text{ with } A_2 \sim N_{m_2}(h_2, H_2) \text{ (asy.)} \\
\text{where } \; p &= (\tilde{p} + 1) = 5 \\
n &= (\tilde{n} + 1) = 11 \\
m_1 &= \tilde{m}_1 = 5 \\
m_2 &= (\tilde{m}_2 + 1) = 7 \\
m &= (m_1 + m_2) = 12
\end{aligned}
$$

which will be employed in the application of the data estimation methods described in the next section.

4 Estimations with *Mathematica*

Following Danilov and Magnus (2007), five alternative methods are presented below. Although they all yield the same estimations, the performance of their computerized solution algorithms differ substantially depending on the size and sparsitiy of the linear system concerned. We apply *Mathematica* software version 6.0 to obtain the estimations.

4.1 The Bayesian estimation

Assume (i) $d_1|x^1 \sim N_p(D_1x^1, \Sigma_1)$ where $D_{1,(p,n)}$ has full row-rank and Σ_1 is positive definite (hence non-singular); (ii) $Ax^1 \sim N_m(h, H)$ where $A = (A_1 : A_2)$, a column vector $h = (h_1, h_2)$, a block diagonal matrix $H = (H_1, H_2)$ with H_1 associated with A_1 and H_2 with A_2; (iii) A has full row-rank and H may be singular. If $m < n$, let L be a semi-orthogonal $(n, n-m)$ matrix such that $L^TL = I_{n-m}$ and $AL = 0$, and assume that the identifiability condition $r(A) + r(D_1L) = n$ is satisfied. Then the posterior distribution of x^1 is given by $x^1|d_1 \sim N_n(\mu, V)$ with

$$
\begin{aligned}
V &= A^+HA^{+'} - A^+HA^{+'}D_1'\Sigma_0^{-1}D_1A^+HA^{+'} + CKC' \\
\mu &= A^+h - (A^+HA^{+'} + CK)D_1'\Sigma_0^{-1}(D_1A^+h - d_1) \\
\text{where } A^+ &= A'(AA')^{-1} \quad \text{"the Moore-Penrose inverse"} \\
\Sigma_0 &= \Sigma_1 + D_1A^+HA^{+'}D_1' \\
C &= I_n - A^+HA^{+'}D_1'\Sigma_0^{-1}D_1 \\
K &= \left\{ \begin{array}{ll} L(L'D_1'\Sigma_0^{-1}D_1L)^{-1}L' & \text{if } m < n \quad \text{(Lemma A2)} \\ 0 & \text{if } m = n \quad \text{(Lemma A1)} \end{array} \right\}
\end{aligned}
$$

(see Theorem 1 in *Magnus, Tongeren and Vos (2000)*). A^+ denotes the Moore-Penrose (MP) inverse of A. All the variables with superscript $(+)$ stand for the MP inverse.

4.1.1 Implementation with *Mathematica*

- Identifiability condition: $r(A)+r(D_1L) = n$. This condition is necessary and sufficient for the existence of a solution. The rank of A is 11, which is also equal to the number of variables in the system; that is, $r(A) = n = 11$. This shows that the identifiability condition is satisfied but there is one redundant equation in A because $(m - n) = (12 - 11) = 1$. The task is to find out that redundant equation and eliminate it from the system.

- Having applied the Gram-Schmidt method to the set of 7 identities, we find out that 6 identities (excluding $Z^t \equiv P^t + M^t$) are linearly dependent. Hence, dropping any one of the 6 identities from A leads to a system of 11 equations. This implies that we have 6 alternative systems. The question is to identify and eliminate from A the single identity that leads to the most inefficient estimations. Here is a sketch of how to perform this task.

Step 1: Determine the rank of of A with dimension of $(m, n) = (12, 11)$ where m=the number of rows, n=the number of columns.

Step 2: $rk(A) = 11$ implies that one of the equations is redundant, which needs to be eliminated from A for the unique solution to exist.

Step 3: Apply Gram-Schmidt process to identify the linearly dependent equation. The process would generate a zero row for the dependent equation. Since dependency is a property of a group of equations, not a property of a single equation, Gram-Schmidt process results in a different dependent equation every time we change the order of rows in A. Thus, we obtain 6 alternative systems, each of which has 11 equations and has a non-zero determinant. (Note that a non-zero determinant implies that the system of equations concerned comprises a linearly independent set.)

Step 4: There are 6 non-zero determinants. This implies that the final Bayesian estimation should be performed for each one of 6 systems separately and the one that minimizes posterior standard deviation should be used in the final analysis.

- Due to the elimination of the dependent equation from A, necessary adjustments are made in the vector h and the variance-covariance matrix H.

- The identifiability condition is satisfied with $r(A) = m = n = 11$ and one redundant equation is dropped from A. The reduced A is then used in all the operations presented in the following sections.

- The Bayesian data estimation approach allows for the deviation from the "true" values of the benchmark values of the indicator ratios and the data observations. Reliability levels assigned to each ratio and each one of the 4 observations imply that the "true" values are most likely to lie within the confidence intervals implied by the reliability levels. The concept of reliability bridges the gap between the "true" and "observed" values of a variable. We set reliability levels as: {Fixed (F), Strong (S), High (H), Medium (M), Low (L), Poor (P)}, with arbitrary reliability coefficients of {0, 0.01, 0.03, 0.06, 0.12, 0.24}, respectively. Coefficient of variation, defined by the ratio of standard error to mean, represents reliability coefficient. Given the reliability coefficient and the mean value at $t = 0$ of the variable of interest (see Table 1), we calculate prior standard error of that variable. In the rest of the paper, matrix notation is used for convenience. Estimate the variance-covariance matrices (Σ_1, H) for time $t = 1$ observations and for the indicator ratios, respectively. In the estimation of Σ_1 associated with d_1, the following reliability levels and coefficients are assumed:

Table 1

$Variable$	$R-coeff$	$Mean$	$Prior\ s.e.$	$Prior\ var.$
P^1	M=0.06	107.12	6.43	41.3
M^1	H=0.03	93.64	2.81	7.9
K^1	L=0.12	32.14	3.86	14.9
X^1	H=0.03	74.98	2.25	5.1
Z^1	MH=0.04	200.76	8.03	64.5

Consider, for example, P^1, which is assumed to be observed at the Medium level, with a corresponding reliability coefficient of 0.06. Applying the definition of coefficient of variation

$= \frac{se_{P^1}}{Mean\ of\ P^1}$ would then yield prior standard error $se_{P^1} = 0.06 * 107.12 = 6.427\,2$. Thus, the prior variance is 41.3. This operation yields

$$\Sigma_1 = \begin{bmatrix} 41.31 & 0 & 0 & 0 & 0 \\ 0 & 7.89 & 0 & 0 & 0 \\ 0 & 0 & 14.87 & 0 & 0 \\ 0 & 0 & 0 & 5.06 & 0 \\ 0 & 0 & 0 & 0 & 64.49 \end{bmatrix}.$$

The estimation of H associated with A is a bit complex. H_1 corresponding to the indicator ratios is a (5,5) diagonal matrix, elements of which are $\sigma_{ij}^2 B_{ij}^2$, whereas H_2 corresponding to the identities is a (6,6) zero matrix. Table 2 shows how to derive H_1 :

Table 2

Indicator Ratios	*R− coeff*	$E(\frac{x_i^t}{x_j^t}) = r_{ij}$	*Prior s.e.*	*Prior var* (σ_{ij}^2)	B_{ij}^2	$\sigma_{ij}^2 B_{ij}^2$
$\frac{x_3^t}{x_1^t} = \frac{I^t}{P^t}$	H=0.03	0.4	0.012	0.00014	10000	1.4
$\frac{x_4^t}{x_{11}^t} = \frac{K^t}{Z^t}$	H=0.03	0.16	0.005	0.00002	36076	0.8
$\frac{x_2^t}{x_1^t} = \frac{M^t}{P^t}$	M=0.06	0.9	0.05	0.0029	10000	29
$\frac{x_6^t}{x_8^t} = \frac{C^t}{R^t}$	L=0.12	0.83	0.1	0.0099	3612	36
$\frac{x_5^t}{x_1^t} = \frac{X^t}{P^t}$	M=0.06	0.7	0.04	0.0018	10000	18

The ratios are assumed to be distributed as $(\frac{x_i^t}{x_j^t}) \sim N_{m_2}(r_{ij}, \sigma_{ij}^2)$, while the linearized ratios as $(x_i^t - r_{ij} x_j^t) \sim N_{m_2}(0, \sigma_{ij}^2 B_{ij}^2)$. The benchmark data x^0 are used to calculate:

$$B_{ij} = \left\{ \begin{array}{cc} \frac{r_{ij}^2}{(1+r_{ij}^2) r_{ij}} x_i + \frac{1}{(1+r_{ij}^2)} x_j & \text{if both } x_i \text{ and } x_j \text{ are available} \\ \frac{x_i}{rij} & \text{if only } x_i \text{ is available} \\ x_j & \text{if only } x_j \text{ is available} \end{array} \right\}$$

- Since $m = n$, Lemma A1 of Theorem 1 applies. The full system is characterized by $(D_1, d_1, \Sigma_1, A, h, H)$, where

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$d_1 = \begin{bmatrix} 107.12 \\ 93.64 \\ 32.14 \\ 74.98 \\ 190 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 41.3 & 0 & 0 & 0 & 0 \\ 0 & 7.9 & 0 & 0 & 0 \\ 0 & 0 & 14.9 & 0 & 0 \\ 0 & 0 & 0 & 5.1 & 0 \\ 0 & 0 & 0 & 0 & 64.5 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.16 \\ -0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.83 & 0 & 0 & 0 \\ -0.7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$h = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; H = \begin{bmatrix} 1.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 29.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 35.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 17.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4.1.2 Mathematica program for the Bayesian estimation

Initialization

Clear[X, pp, mm, ii, kk, xx, cc, yy, rr, ss, bb, zz, Xbm, data, ratio, identity, n, p, m1, m2, m, d1, D1, h1, A1, h2, A2, system, systemL, dri, DRI, dr, DR, ri, RI, da, Da, r, R, id, Id, RankCondition, methods, dependentQ, row, col, zero, orthogonalV, selectionV, reducedV, rankV, MGS, refV, equ, linearF, det, nsV, denklem, test, sonuc];

X = {pp, mm, ii, kk, xx, cc, yy, rr, ss, bb, zz};

Xbm = {100, 90, 40, 30, 70, 50, 60, 60, 10, 20, 190};

data = {pp + 107.12 == 0, mm + 93.64 == 0, kk + 32.14 == 0, xx + 74.98 == 0, zz + 200.76 == 0};

ratio = {ii - 0.4 pp == 0, kk - 0.16 zz == 0, mm - 0.9 pp == 0, cc - 0.83 rr == 0, xx - 0.7 pp == 0};

identity = {yy - pp + ii == 0, bb - mm + xx == 0, ss - rr + cc == 0, yy - rr == 0, kk - ss - bb == 0, pp + mm - ii - cc - kk - xx == 0, zz - pp - mm == 0};

n = Length[X];

p = Length[data];

m1 = Length[ratio];

m2 = Length[identity];

m = m1 + m2;

system = Flatten[{data, ratio, identity}];

systemL =NestList[RotateLeft, system, m - 1];

Rank condition

```
rank[data_, ratio_, identity_, vecs_] := Module[
{rkDRI, rkDR, rkRI, rkDa, rkR, rkId},
{dri, DRI} = CoefficientArrays[Flatten[{data, ratio, identity}], vecs];
{dr, DR} = CoefficientArrays[Flatten[{data, ratio}], vecs];
{ri, RI} = CoefficientArrays[Flatten[{ratio, identity}], vecs];
{da, Da} = CoefficientArrays[Flatten[{data}], vecs];
{r, R} = CoefficientArrays[ratio, vecs];
{id, Id} = CoefficientArrays[identity, vecs];
RankCondition = {"rank(data,ratio,identity)=n", "rank(data,ratio) =(p+m1)", "rank(ratio,identity)
=(m1+m2)", "rank(data) =p", "rank(ratio) =m1", "rank(identity) =m2"};
rkDRI = If[MatrixRank[DRI] == Length[vecs], "satisfied", "NOT satisfied"];
rkDR = If[MatrixRank[DR] == Length[DR], "satisfied", "NOT satisfied"];
rkRI = If[MatrixRank[RI] == Length[RI], "satisfied", "NOT satisfied"];
rkDa = If[MatrixRank[Da] == Length[Da], "satisfied", "NOT satisfied"];
rkR = If[MatrixRank[R] == Length[R], "satisfied", "NOT satisfied"];
rkId = If[MatrixRank[Id] == Length[Id], "satisfied", "NOT satisfied"];
Print[TableForm[{{"=p+m1+m2" Length[DRI], MatrixRank[DRI], RankCondition[[1]], rkDRI},
{"=p+m1" Length[DR], MatrixRank[DR], RankCondition[[2]], rkDR}, {Length[RI] "=m1+m2",
MatrixRank[RI], RankCondition[[3]], rkRI}, {"=p" Length[Da], MatrixRank[Da], RankCondition[[4]],
rkDa}, {"=m1" Length[R], MatrixRank[R], RankCondition[[5]], rkR}, {"=m2" Length[Id], Ma-
trixRank[Id], RankCondition[[6]], rkId}}, TableHeadings -> {{ "Full System", "Data-Ratio", "Ratio-
Identity", "Data", "Ratio", "Identity"}, {"# of Eqns", "Rank", "Rank condition tested", "Test
result"}}] ];
];
rank[data, ratio, identity, X];
```

Identification of dependent equations

```
methods = {"GramSchmidt", "ModifiedGramSchmidt", "Householder", "Reorthogonalization"};
dependentQ[eqns_, vecs_, method_] := Module[
 {h, V, row, col, zero, orthogonalV, selectionV},
{h, V} = CoefficientArrays[eqns, vecs];
{row, col} = Dimensions[V];
zero = ConstantArray[0, col];
orthogonalV = Orthogonalize[V, Method -> method] // Chop;
selectionV = Map[If[# != zero, "keep", "discard"] &, orthogonalV];
reducedV = {};
reducedh = {};
Do[
If[orthogonalV[[i]] != zero, AppendTo[reducedV, V[[i]]]];
If[orthogonalV[[i]] != zero, AppendTo[reducedh, h[[i]]], {i, 1, row}];
Print[
TableForm[Table[{equ[i], selectionV[[i]]}, {i, 1, row}], TableHeadings -> {{}, {"Equations",
"ModifiedGramSchmidt"}}],
V // Normal // MatrixForm, reducedV // Normal // MatrixForm, reducedh // Normal //
MatrixForm
```

```
            ];
    ];
    systemRI = Flatten[{ratio, identity}];
    dependentQ[systemRI, X, methods[[2]]] ;
```

Derivation of a function of dependent equations

```
    linearF[eqns_, vecs_] := Module[
     {h, V, row, col, rankV, nsV, zero, decision, det, eq, denklem, sonuc},
     {h, V} = CoefficientArrays[eqns, vecs];
    {row, col} = Dimensions[V];
     rankV = MatrixRank[V];
     nsV = NullSpace[Transpose[V]] // Chop;
     zero = ConstantArray[0, Length[nsV]];
    decision = If[rankV == row, "Empty: therefore, linear function(s) do not exist.", "Non-empty:
therefore, linear function(s) exist."];
    det = Minors[V, rankV] // Chop;
    eq = Array[Subscript[e, #] &, row];
    denklem = Thread[nsV.eq == zero];
    sonuc = Reduce[denklem, eq] // Chop // Flatten;
    Print[TableForm[{{decision}, {sonuc}}, TableHeadings -> {{ "Null-Space", "Linear Function(s)"},
{"Outcome"}}] ];
    ];
    linearF[systemRI, X];
```

Variance-covariance matrix

```
    VarCovariance[Rdata_, data_, Rratio_, ratio_]:= Module[
    {p, m1, m2, na, Xi, Xj, rMat1, rMat, Bmat1, B1, Bmat2, B2, Bmat3, B3, H2},
    na = 0;
    Xi = Xbm;
    Xj = Xbm;
    rMat1 = ConstantArray[na, {Length[Xbm], Length[Xbm]}];
    rMat =Table[If[expR[[i + 1, j + 1]] != na,
    rMat1[[i, j]] = expR[[i + 1, j + 1]], na], {i, 1, Length[Xbm]}, {j, 1, Length[Xbm]}];
    Bmat1 = ConstantArray[na, {Length[Xbm], Length[Xbm]}];
    B1 = Table[ If[Xi[[i]] ≠ na && Xj[[j]] ≠ na && rMat[[i, j]] ≠ na, Bmat1[[i,j]] = (rMat[[i,
j]]/(1 + rMat[[i, j]]^2))*Xi[[i]] + (1/(1 + rMat[[i, j]]^2))*Xj[[j]], na], {i, 1, Length[Xbm]}, {j, 1,
Length[Xbm]}];
    Bmat2 = ConstantArray[na, {Length[Xbm], Length[Xbm]}];
    B2 = Table[If[Xi[[i]] ≠ na && Xj[[j]] == na && rMat[[i, j]] ≠ na, Bmat2[[i, j]] = (Xi[[i]]/rMat[[i,
j]]), na], {i, 1, Length[Xbm]}, {j, 1, Length[Xbm]}];
    Bmat3 = ConstantArray[na, {Length[Xbm], Length[Xbm]}];
    B3 = Table[If[Xi[[i]] == na && Xj[[j]] ≠ na && rMat[[i, j]] ≠ na, Bmat3[[i, j]] = Xj[[j]], na], {i,
1, Length[Xbm]}, {j, 1, Length[Xbm]}];
    Do[se[p] = Rdata[[p + 1, 3]]*Rdata[[p + 1, 4]], {p, 1, Length[data]}];
    Do[seR[m1] = Rratio[[m1 + 1, 7]]*Rratio[[m1 + 1, 8]], {m1, 1, Length[ratio]}];
```

```
S1 = DiagonalMatrix[Table[se[p]^2, {p, 1, Length[data]}]];
H1 = DiagonalMatrix[Table[seR[m1]^2*B1[[Rratio[[m1 + 1, 3]], Rratio[[m1 + 1, 5]]]]^2, {m1,
1, Length[ratio]}]];
S = ArrayFlatten[{{S1, 0}, {0, H1}}];
dependentQ[identity, X, methods[[2]]];
H2 = DiagonalMatrix[ConstantArray[0, Length[reducedh]]];
H = ArrayFlatten[{{H1, 0}, {0, H2}}];
Print[S1 // MatrixForm, H1 // MatrixForm, S // MatrixForm, H // MatrixForm];
];
VarCovariance[Rdata, data, Rratio, ratio]
```

Bayesian estimation method

```
bayes[data_, ratio_, identity_, vecs_, methods_]:= Module[
{d1, D1, hh, AA, row, col, zero, orthogonalAA, selectionAA, A, h, Ap, S0, CC, L, K, M, V,
postDist},
{d1, D1} = CoefficientArrays[Flatten[{data}], vecs];
{hh, AA} = CoefficientArrays[Flatten[{ratio, identity}], vecs];
{row, col} = Dimensions[AA];
zero = ConstantArray[0, col];
orthogonalAA = Orthogonalize[AA, Method -> methods] // Chop;
selectionAA = Map[If[# != zero, "keep", "discard"] &, orthogonalAA];
A = {};
h = {};
Do[ If[orthogonalAA[[i]] != zero, AppendTo[A, AA[[i]]]];
If[orthogonalAA[[i]] != zero, AppendTo[h, hh[[i]]], {i, 1, row} ];
Ap = Inverse[Transpose[A].A ].Transpose[A];
S0 = S1 + D1.Ap.H.Transpose[Ap].Transpose[D1];
M = Ap.h - (Ap.H.Transpose[Ap]).Transpose[D1].Inverse[S0].(D1.Ap.h - d1);
V = Ap.H.Transpose[Ap] - Ap.H.Transpose[Ap].Transpose[D1].Inverse[S0].D1.Ap.H.Transpose[Ap];
Do[ postDist = Table[{X[[i]], M[[i]], Sqrt[V[[i, i]]]}, {i, 1, col}]];
Print[ TableForm[postDist, TableHeadings -> {{}, {"Variable", "Post-mean", "Post-se"}}]];
];
bayes[data, ratio, identity, X, methods[[2]]]
```

4.2 Restricted least-squares estimation

For estimations in large systems, the least-squares method works better compared to the Bayesian estimation method (Theorem 1). The Bayesian problem above can be equivalently formulated as a restricted least-squares problem:

$$\underset{x}{\text{Minimize}}(d - Dx)'\Sigma^{-1}(d - Dx) \quad \text{subject to} \quad A_2 x = h_2$$

$$
\begin{aligned}
\text{where } d &= Dx + \epsilon \\
d &\mid x \sim N_{p+m_1}(Dx, \Sigma) \\
\epsilon &\sim N_{p+m_1}(0, \Sigma) \text{ and} \\
d &= \begin{pmatrix} d_1 \\ h_1 \end{pmatrix}; \quad D = \begin{pmatrix} D_1 \\ A_1 \end{pmatrix}; \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & H_1 \end{pmatrix}.
\end{aligned}
$$

From Theorem 36 in Magnus and Neudecker (1999) (p.233), the general solution, x, to this minimization problem is:

$$
\begin{aligned}
x &= x_0 + N^+ A_2^{'}(A_2 N^+ A_2^{'})^+(h_2 - A_2 x_0) + (I_n - NN^+)q \\
\text{where } x_0 &= N^+ D^{'} \Sigma^{-1} d \\
N &= D^{'} \Sigma^{-1} D + A_2^{'} A_2 \\
q &= \text{an arbirary vector}
\end{aligned}
$$

Compute x by using:

$$
d = \begin{bmatrix} 107.12 \\ 93.64 \\ 32.14 \\ 74.98 \\ 190 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -0.4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.16 \\ -0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.8 & 0 & 0 & 0 \\ -0.7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
\Sigma = \begin{bmatrix} 41.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 64.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 29.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 35.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17.6 \end{bmatrix}
$$

$$
A_2 = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
$$

4.2.1 Mathematica program for restricted least-squares estimation

```
LeastSquares1[data_, ratio_, identity_, methods_] := Module[
    {D1, d1, row, col, zero, orthogonalId, selectionId, A2, h2, N1, piN1, X0, q1, estX, postDist},
    D1 = Join[Da, R];
    d1 = Join[da, r];
    {row, col} = Dimensions[Id];
    zero = ConstantArray[0, col];
    orthogonalId = Orthogonalize[Id, Method -> methods] // Chop;
    selectionId = Map[If[# != zero, "keep", "discard"] &, orthogonalId];
    A2 = {};
    h2 = {};
    Do[ If[orthogonalId[[i]] != zero, AppendTo[A2, Id[[i]]]];
    If[orthogonalId[[i]] != zero, AppendTo[h2, id[[i]]]], {i, 1, row}];
    N1 = Transpose[D1].Inverse[S].D1 + Transpose[ A2]. A2;
    piN1 = PseudoInverse[N1];
    X0 = piN1.Transpose[D1].Inverse[S].d1;
    q1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; (* an arbitrary vector of "Lagrange Multipliers" *)
    estX = X0 + piN1.Transpose[A2].PseudoInverse[A2.piN1.Transpose[A2]].(h2 - A2.X0) + (Iden-
tityMatrix[Length[X]] - piN1.N1).q1;
    postDist = Table[{X[[i]], estX[[i]]}, {i, 1, Length[X]}];
    Print[TableForm[postDist, TableHeadings -> {{}, {"Variable", "Least-squares est."}} ]];
    ];
    LeastSquares1[data, ratio, identity, methods[[2]]]
```

4.3 Unrestricted least-squares estimation

Alternatively, the solution of the following unrestricted minimization problem is also identical to that of the Bayesian estimation of x :

$$\text{Minimize} \quad \begin{pmatrix} d - Dx \\ h_2 - A_2 x \end{pmatrix}' \begin{pmatrix} \Sigma + DD' & DA_2' \\ A_2 D' & A_2 A_2' \end{pmatrix} \begin{pmatrix} d - Dx \\ h_2 - A_2 x \end{pmatrix}$$

$$\text{where } d = \begin{pmatrix} d_1 \\ h_1 \end{pmatrix}, \quad D = \begin{pmatrix} D_1 \\ A_1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & H_1 \end{pmatrix}$$

4.3.1 Mathematica program for unrestricted least-squares estimation

```
LeastSquares2[data_, ratio_, identity_, methods_] := Module[
    {d1, D1, row, col, zero, orthogonalId, selectionId, A2, h2, m1, a11, a12, a21, a22, m2, objF,
sonuc},
    d1 = Join[da, r];
    D1 = Join[Da, R];
    {row, col} = Dimensions[Id];
    zero = ConstantArray[0, col];
```

```
orthogonalId = Orthogonalize[Id, Method -> methods] // Chop;
selectionId = Map[If[# != zero, "keep", "discard"] &, orthogonalId];
A2 = {};
h2 = {};
Do[
If[orthogonalId[[i]] != zero, AppendTo[A2, Id[[i]]]];
If[orthogonalId[[i]] != zero, AppendTo[h2, id[[i]]], {i, 1, row}];
m1 = Join[Transpose[{d1 - D1.X}], Transpose[{ h2 - A2.X}]];
a11 = S + D1.Transpose[D1];
a12 = D1.Transpose[A2];
a21 = A2.Transpose[D1];
a22 = A2.Transpose[A2];
m2 = ArrayFlatten[{{a11, a12}, {a21, a22}}];
objF = Transpose[m1].Inverse[m2].m1;
sonuc = Last[Minimize[Flatten[objF], {pp, mm, ii, kk, xx, cc, yy, rr, ss, bb, zz}]];
Print[ TableForm[{{sonuc}}, TableHeadings -> {{}, {"Least-squares"}}] ];
];
LeastSquares2[data, ratio, identity, methods[[2]]]
```

4.4 Best linear unbiased estimation

Best linear unbiased estimation (BLUE) is an alternative method that leads to the same results as restricted least-squares method. Consider the regression model:

$$\text{Minimize } (d - Dx)^{'} \Sigma^{-1} (d - Dx) \text{ subject to: } A_2 x = h_2$$

where x is a vector of parameters to be estimated. The BLUE estimator of x is given by:

$$\begin{aligned} x &= G^{-1} D^{'} \Sigma^{-1} d + G^{-1} A_2^{'} (A_2 G^{-1} A_2^{'})^{-1} (h_2 - A_2 G^{-1} D^{'} \Sigma^{-1} d) \\ \text{with variance } V &= G^{-1} - G^{-1} A_2^{'} (A_2 G^{-1} A_2^{'})^{-1} A_2 G^{-1} \\ \text{where } G &= D^{'} \Sigma^{-1} D + A_2^{'} A_2 \\ d &= \begin{pmatrix} d_1 \\ h_1 \end{pmatrix}, \quad D = \begin{pmatrix} D_1 \\ A_1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & H_1 \end{pmatrix} \end{aligned}$$

4.4.1 Mathematica program for best linear unbiased estimation

```
blue[data_, ratio_, identity_, vecs_, methods_] :=Module[
    {d, D, a, A, row, col, zero, orthogonalA, selectionA, A2, h2, G, M, V, postDist},
    {d, D} = CoefficientArrays[Flatten[{data, ratio}], vecs];
    {a, A} = CoefficientArrays[Flatten[{identity}], vecs];
    {row, col} = Dimensions[A];
    zero = ConstantArray[0, col];
    orthogonalA = Orthogonalize[A, Method -> methods] // Chop;
    selectionA = Map[If[# != zero, "keep", "discard"] &, orthogonalA];
```

```
A2 = {};
h2 = {};
Do[ If[orthogonalA[[i]] != zero, AppendTo[A2, A[[i]]]];
If[orthogonalA[[i]] != zero, AppendTo[h2, a[[i]]]], {i, 1, row} ];
G = Transpose[D].Inverse[S].D + Transpose[A2].A2;
M = Inverse[G].Transpose[D].Inverse[S].d + Inverse[G].Transpose[A2].Inverse[A2.Inverse[G].
Transpose[A2]].(h2 - A2.Inverse[G].Transpose[D].Inverse[S].d);
V = Inverse[G] - Inverse[G].Transpose[A2].Inverse[A2.Inverse[G].Transpose[A2]].A2.Inverse[G];
postDist = Table[{X[[i]], M[[i]], Sqrt[V[[i, i]]]}, {i, 1, col}];
Print[ TableForm[postDist, TableHeadings -> {{}, {"Variable", "Post-mean", "Post-se"}}] ];
]
blue[data, ratio, identity, X, methods[[2]]]
```

4.5 Linear programming

Linear programming (LP) problems involve the optimization of a linear objective function subject to linear equality and inequality constraints. LP aims to obtain the best outcome, for example, maximum gross domesticl product or maximum profit etc, under some linear constraints on, for example, investment or savings etc. using a linear mathematical model. The example linear system described in Section 3 can be put into a linear programming format as:

$$\begin{array}{rrcl} & & & \text{Maximize } Y \\ \text{subject to} & Wx & = & w \quad \text{where } x \geqslant 0 \\ & W & = & \begin{pmatrix} D_1 \\ A_1 \\ A_2 \end{pmatrix} \text{ and } w = \begin{pmatrix} d_1 \\ h_1 \\ h_2 \end{pmatrix} \end{array}$$

There are many ways to specify an objective function. Suppose that we maximize gross domestic product, denoted by Y, and that x_Y maximizes Y. Alternatively, we could also specify the objective function as maximizing savings, denoted by S, and solve the LP problem under the same constraints. Suppose that x_S maximizes savings. Obviously, these two LP problems are most likely to yield $x_Y \neq x_S$, unless the set of contraints includes a condition imposing $Y = S$. This implies that the outcomes of an LP problem cannot be compared with those of the other 4 methods described above.

However, LP method can be used to derive a range for each variable. Consider the variable Y for example. An LP problem with the objective function of maximizing Y would yield the maximum Y, $Y_{\max}$, while another LP problem with the objective function of minimizing Y would yield the minimum Y, $Y_{\min}$. This results in the range: $Y_{\min} \leq Y \leq Y_{\max}$. The same procedure can be applied to derive ranges for all other variables in the vector x. With such an approach, we can produce reasonable ranges for the variables in the linear system at hand. These ranges would allow us to assess the feasibility of the estimations made by other 4 methods. An estimated x is said to be feasible if $x_{\min} \leq x \leq x_{\max}$.

4.5.1 Mathematica program for linear programming

```
linearP[objF_, constraints_, constants_] := Module[
    {solutionL, solutionU},
    solutionL = LinearProgramming[objF, constraints, constants, {{0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}}];
    solutionU = LinearProgramming[objF, constraints, constants, {{0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}, {0, 250}}];
    Print[ TableForm[ Table[{X[[i]], solutionL[[i]], solutionU[[i]]}, {i, 1, n}], TableHeadings -> {{}, {"Variable", "Lower-bound", "Upper-bound"}}]];
    ]
    linearP[Id[[6]], DRI, dri]
```

5 Conclusion

Four data compilation methods have been presented with their illustrations using an example system of linear equations. For each method, a generic *Mathematica* program has been developed. For the example system we used, Mathematica works fine. However, the power of *Mathematica* remains to be tested with applications of large systems. Currently, a medium size linear system with 32 constraints (including indicator ratios and identities) is tested using the same programs. At the final stage, the programs will be used in data compilation in the Angola project.

References

[1] Magnus, J. R., and Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. New York, USA: John Wiley & Sons Ltd.

[2] Magnus, J. R., van Tongeren, J., and Vos. (2000).

[3] Danilov, D., and Magnus, J. R. (2006). Some equivalences in linear estimation.

[4] Danilov, D., and Magnus, J. R. (2007). On the estimation of a large sparse Bayesian system: the SNAER program.

[5] Danilov, D., and Magnus, J. R. (2005). Least-squares estimation in large sparse systems.

[6] Huang, C. J., and Crooke, P.S. (1997). *Mathematics and* ***Mathematica*** *for economists*. Oxford, UK: Blackwell Publishers.

[7] Mathematica version 6 manual

[8] Liu's paper on elimination of redundant equations

[9] Tridiagonalization paper

[10] other papers I read (4 or more)