BALANCING A SOCIAL ACCOUNTING MATRIX: Theory and application

Ismaël Fofana[1]
André Lemelin
John Cockburn

Centre Interuniversitaire sur le Risque les Politiques Economiques et L'Emploi (CIRPEE), Université Laval, August 2005

Summary

Computable general equilibrium modeling requires a consistent and coherent benchmark data set, which are generally organized in the form of a Social Accounting Matrix (SAM). These data generally come from quite diverse sources and correspond to different periods of time. As a result, they often present inconsistencies. We present a simple computer program that makes it possible to reconcile this information in order to balance a SAM. The program minimizes the changes to the base data using various optimizing techniques, including cross-entropy and least squares. The program is an attractive and easy alternative to the arbitrary and time-consuming manual and other methods generally used to balance SAMs.

Introduction

Computable General Equilibrium (CGE) modeling and Social Accounting Matrix (SAM) based research requires the use of the most recent economic data available in a coherent framework. However, these data generally come from quite diverse sources - e.g. input-output tables, national accounting data, household surveys, firm surveys, labor market surveys, government accounts, international trade accounts, etc. - and correspond to different periods of time. For example, input-output data are generally prepared every five years or more, whereas national data on income, production, trade, etc. are generated annually. In some cases, it is possible to start with an existing SAM, which is then updated using new data. In other cases, the SAM must be constructed in its entirety. "The problem in estimating a disaggregated SAM for a recent year is to find an efficient (and cost-effective) ways to incorporate and reconcile information from a variety of sources, including data from the prior years" (Robinson and al. - 2000).

In the next section, we present a literature review on some commonly used approaches for balancing a SAM. The second section of the paper presents a program that implements the most desirable of these approaches.

1. The RAS Method

RAS is a widely used methodology to balance SAMs. It is used when new information on the matrix row and column sums becomes available and we want to update an existing matrix. The problem is to generate a new nXn matrix A^1 from an existing matrix A^0 of the same dimension while respecting new given row and column totals, by applying row and column multipliers, r and s respectively. The $(2n-1)$ unknown multipliers are determined by the $(2n-1)$ independent row and column restrictions using an iterative adjustment procedure.

Define T as a matrix of SAM transactions, where t_{ij} is a cell value that satisfies the condition:

[1]Corresponding author : ifofana@ecn.ulaval.ca

$$T_{.j} = \sum_i t_{ij}$$

A SAM coefficient matrix A is constructed from T by dividing the cells in each column of T by the column sums:

$$a_{ij} = t_{ij} \big/ t_{.j} \ .$$

A classic approach to solve this problem is to generate a new matrix A^1, from the old matrix A^0 by means of "biproportional" row and column operations:

$$a_{ij}^1 = r_i \ a_{ij}^0 s_j$$

In matrix notation:

$$A^1 = \tilde{R} \ A^0 \tilde{S}$$

where the ($\tilde{}$) indicate a diagonal matrix of elements r_i and s_j[2]. RAS method is an iterative algorithm of biproportional adjustment.

In what follows, subscripts 0,1,2… refer to iteration steps and (^) is used for new column or row values.

- Step 1:

$$a_i^1 = \frac{\hat{x}_{i.}}{\sum_j x_{ij}^0} \quad \Rightarrow \quad x_{ij}^1 = a_i^1 x_{ij}^0 \quad \Rightarrow \quad b_j^1 = \frac{\hat{x}_{.j}}{\sum_i x_{ij}^1} \quad \Rightarrow \quad x_{ij}^2 = b_i^1 x_{ij}^1$$

- Step 2:

$$a_i^2 = \frac{\hat{x}_{i.}}{\sum_j x_{ij}^2} \quad \Rightarrow \quad x_{ij}^3 = a_i^2 x_{ij}^2 \quad \Rightarrow \quad b_j^2 = \frac{\hat{x}_{.j}}{\sum_i x_{ij}^3} \quad \Rightarrow \quad x_{ij}^4 = b_i^2 x_{ij}^3$$

…

- Step t:

$$a_i^t = \frac{\hat{x}_{i.}}{\sum_j x_{ij}^{2t-2}} \quad \Rightarrow \quad x_{ij}^{2t-1} = a_i^t x_{ij}^{2t-2} \quad \Rightarrow \quad b_j^t = \frac{\hat{x}_{.j}}{\sum_i x_{ij}^{2t-1}} \quad \Rightarrow \quad x_{ij}^{2t} = b_i^t x_{ij}^{2t-1}$$

[2] Bacharach shows that this « RAS » method works in that a unique set of positive multipliers (normalized) exists that satisfies the biproportionality condition and that the elements of R^* and S^* can be found by a simple iterative procedure. For the method to work, the matrix must be "connected", which is a generalization of the notion of "indecomposable" (Bacharach, 1970, p. 47). For example, this method fails when a column or row of zero exists because it cannot be proportionately adjusted to sum to a non-zero number. Note also that the matrix need not be square. The method can be applied to any matrix with known row and column sums.

This process is continued until the iterations converge. We can summarize these steps by:

$$x_{ij}^{2t-1} = \left(\prod_{h=1}^{t-1} b_j^h\right)\left(\prod_{k=1}^{t} a_i^k\right) x_{ij}^0 \text{ ; for odd rank values, } x_{ij}^1, x_{ij}^3, x_{ij}^5, \ldots$$

$$x_{ij}^{2t} = \left(\prod_{h=1}^{t} b_j^h\right)\left(\prod_{k=1}^{t} a_i^k\right) x_{ij}^0 \text{ ; for even rank values, } x_{ij}^2, x_{ij}^4, x_{ij}^6, \ldots$$

With $A_i^t = \left(\prod_{k=1}^{t} a_i^k\right)$ and $B_j^t = \left(\prod_{h=1}^{t} b_j^h\right)$

$$\Rightarrow \quad x_{ij}^{2t-1} = A_i^t B_j^{t-1} x_{ij}^0 \text{ ; for odd rank values, } x_{ij}^1, x_{ij}^3, x_{ij}^5, \ldots$$

$$\Rightarrow \quad x_{ij}^{2t} = A_i^t B_j^t x_{ij}^0 \text{ ; for even rank values, } x_{ij}^2, x_{ij}^4, x_{ij}^6, \ldots$$

When there is a solution, the RAS method has the advantage of being simple to apply. But, this simplicity has many disadvantages: (1) a lack of economic foundations, (2) inability to accommodate other sources of data than those on row and column totals, for example we cannot fix new cell values that we suppose are accurately measured. Because of these disadvantages, many researchers prefer use the more flexible cross-entropy method.

2. The Cross-Entropy Method

The cross-entropy approach was first applied to SAM balancing by Sherman Robinson and colleagues at IFPRI (Robinson, Cattaneo and El-Said, 1998; Robillard and Robinson, 1999; Robinson, Cattaneo and El-Said, 2000; Robinson and El-Said, 2000). It is based on information theory developed by Shannon (1948) and applied to the problem of estimation and statistical inference by Jaynes (1957). Theils (1967) brought the approach to economics.[3] The estimation procedure is to minimize the Kullback-Leibler (1951) cross-entropy measure of the distance between the new and the prior estimated probabilities. That is to minimize the "additional" set of information brought into X^1 comparatively to the prior set X^0. Applied to the procedure of updating a SAM, the problem is to find a new SAM X^1, close to an existing SAM X^0, by minimizing the cross-entropy distance between them, respecting all constraints. Therefore, for this simple analysis, we start with prior information t_{ij}^0 on SAM cells and assume that we have exact information on current column sums, X and any other information set as constant[4].

This can be written in term of probabilities as :

$$\min_{\{t^1\}} H = \sum_i \sum_j t_{ij}^1 \ln \frac{t_{ij}^1}{t_{ij}^0} = \sum_i \sum_j t_{ij}^1 \ln t_{ij}^1 - \sum_i \sum_j t_{ij}^1 \ln t_{ij}^0$$

[3] "The cross-entropy method uses all available information, including prior parameter estimates, and supports estimation even in a "data sparse" environment. The use of the cross-entropy measure in the estimation criterion has been justified on the basis of axiomatic arguments concerning its desirability both as a measure of "information" and as a criterion for inference. There are close links between the minimum cross-entropy criterion and the maximum likelihood estimators, but the cross-entropy criterion requires fewer statistical assumptions in that its application does not require specification of an explicit likelihood function. In our case, this sparseness in assumptions is desirable since we have no knowledge about the form of any underlying probability distributions" (Robillard and Robinson 2001).

[4] For more complex models with error components see Robillard and Robinson (1999), Robinson and al. (2000)

Subject to:

$$\sum_j t_{ij}^1 X_j = X_i$$
$$\sum_j t_{ij}^1 = 1$$

t_{ij}^1 is a new value of cell ij and $0 \le t_{ij}^1 \le 1$. This can be directly formulated in SAM transaction flows instead of the proportions as demonstrated by Lemelin (2002, translation by the authors):

"Given the sum of Kullback-Leibler's measures of n distributions, one by column,

$$H = \sum_j \sum_i t_{ij}^1 \ln\left(\frac{t_{ij}^1}{t_{ij}^0}\right)$$

minimizing *H* is equivalent to minimizing the analogue function by substituting proportions t_{ij} by transaction flows x_{ij}.

$$H = \sum_j \sum_i \frac{x_{ij}}{x_{\bullet\bullet}} \ln\left[\frac{\left(x_{ij} \middle/ x_{\bullet\bullet}\right)}{\left(x_{ij}^0 \middle/ x_{\bullet\bullet}^0\right)}\right]$$

with $x_{\bullet\bullet} = \sum_i \sum_j x_{ij}$ and $x_{\bullet\bullet}^0 = \sum_i \sum_j x_{ij}^0$

$$H = \frac{1}{x_{\bullet\bullet}} \sum_j \sum_i x_{ij} \left[\ln\left(\frac{x_{ij}}{x_{ij}^0}\right) - \ln\left(\frac{x_{\bullet\bullet}}{x_{\bullet\bullet}^0}\right)\right]$$

$$H = \frac{1}{x_{\bullet\bullet}} \sum_j \sum_i x_{ij} \ln\left(\frac{x_{ij}}{x_{ij}^0}\right) - \frac{1}{x_{\bullet\bullet}} \sum_j \sum_i x_{ij} \ln\left(\frac{x_{\bullet\bullet}}{x_{\bullet\bullet}^0}\right)$$

$$H = \frac{1}{x_{\bullet\bullet}} \sum_j \sum_i x_{ij} \ln\left(\frac{x_{ij}}{x_{ij}^0}\right) - \ln\left(\frac{x_{\bullet\bullet}}{x_{\bullet\bullet}^0}\right)$$

where the last term is a constant.

However, $H \ge 0$ and $\sum_j \sum_i x_{ij} \ln\left(\frac{x_{ij}}{x_{ij}^0}\right) \ge x_{\bullet\bullet} \ln\left(\frac{x_{\bullet\bullet}}{x_{\bullet\bullet}^0}\right)$

Note that if $x_{\bullet\bullet} < x_{\bullet\bullet}^0$, the right hand side is not necessarily non negative."

3. Comparing Cross-Entropy and RAS Methods

Günlük-Senesen and Bates (1988) and others have shown that RAS method is very close to a problem involving the minimization of $H = \sum t_{ij}^1 \ln\left(t_{ij}^1 / t_{ij}^0 \right)$, subject to known row and column sum constraints. The RAS method has been extended to accommodate uncertainty in the row and column totals, as well as negative cell values.

In essence, the cross-entropy method is formally similar to the generalized RAS method, which we saw earlier uses an entropy-based minimand and a constraint set appropriate to a "type 1[5]" balancing problem. However there are some significant differences and additional complexities.

Let us rewrite the Kullback-Leibler measure of cross-entropy of the distance between the new and the prior estimated probabilities

$$\min_{p_{ij}} \sum_i \sum_j p_{ij} \ln\left(\frac{p_{ij}}{q_{ij}} \right)$$

subject to:

$$\sum_i p_{ij} = \overline{p_{\bullet j}}$$
$$\sum_j p_{ij} = \overline{p_{i \bullet}}$$

where $\{p_{ij}\}$ are the frequencies of the adjusted matrix and $\{q_{ij}\}$ the frequencies of the initial matrix.

$$p_{ij} = \frac{x_{ij}}{\sum_g \sum_h x_{gh}}$$

$$q_{ij} = \frac{x_{ij}^0}{\sum_g \sum_h x_{gh}^0}$$

---

[5] Jeffery I Round, (2001):

Problem 1: If $X = \left[x_{ij} \right]$ is an m x n nonnegative matrix and u and v are positive vectors of orders m and n respectively then determine an m x n matrix X* 'close to' X such that $\sum_j x_{ij}^* = u_i$ and $\sum_i x_{ij}^* = v_j$; and $x_{ij}^* > 0$ if and only if $x_{ij} > 0$ (all i, j).

Problem 2: If $X = \left[x_{ij} \right]$ is an (n x n) nonnegative matrix then determine an (n x n) matrix X* 'close to' X such that $\sum_j x_{ij}^* = \sum_j x_{ji}^*$ (all i) and $x_{ij}^* > 0$ if and only if $x_{ij} > 0$ (all i, j)

The solution is obtained by defining and solving the Lagrangian for the above problem:

$$\Lambda = \sum_i \sum_j p_{ij} \ln\left(\frac{p_{ij}}{q_{ij}}\right) + \sum_i \alpha_i \left(\sum_j p_{ij} - \overline{p_{i\bullet}}\right) + \sum_j \beta_j \left(\sum_i p_{ij} - \overline{p_{\bullet j}}\right)$$

$$\Lambda = \sum_i \sum_j p_{ij} \left(\ln p_{ij} - \ln q_{ij}\right) + \sum_i \alpha_i \left(\sum_j p_{ij} - \overline{p_{i\bullet}}\right) + \sum_j \beta_j \left(\sum_i p_{ij} - \overline{p_{\bullet j}}\right)$$

The first-order conditions are:

$$\frac{\partial \Lambda}{\partial \alpha_i} = \sum_j p_{ij} - \overline{p_{i\bullet}} \qquad (1)$$

$$\frac{\partial \Lambda}{\partial \beta_j} = \sum_i p_{ij} - \overline{p_{\bullet j}} \qquad (2)$$

$$\frac{\partial \Lambda}{\partial p_{ij}} = 1 + \ln p_{ij} - \ln q_{ij} + \alpha_i + \beta_j = 0 \qquad (3)$$

The outcome combines the information from the data and the prior:

$$p_{ij} = q_{ij} e^{-\left(1+\alpha_i+\beta_j\right)} \qquad (3')$$

where α_i and β_j are Lagrangian multipliers associated with the information on row and column sums. The outcome is determined so as to respect the constraints (1) and (2). The strict convexity of the objective function ensures the uniqueness of the solution when it exists. As demonstrated in section 1, with $A_i^t = \left(\prod_{k=1}^{t} a_i^k\right)$ and $B_j^t = \left(\prod_{h=1}^{t} b_j^h\right)$, the RAS algorithm can be written as :

$x_{ij}^{2t-1} = A_i^t B_j^{t-1} x_{ij}^0$, with

$$\frac{B_j^t}{B_j^{t-1}} = b_j^t = \overline{x_{\bullet j}} \left(\sum_i A_i^t B_j^{t-1} x_{ij}^0\right)^{-1} \text{ that is } B_j^t = \overline{x_{\bullet j}} \left(\sum_i A_i^t x_{ij}^0\right)^{-1}$$

for the odd rank of values $x_{ij}^1, x_{ij}^3, x_{ij}^5, \ldots$, and $x_{ij}^{2t} = A_i^t B_j^t x_{ij}^0$, with

$$\frac{A_i^t}{A_i^{t-1}} = a_i^t = \overline{x_{i\bullet}} \left(\sum_j A_i^{t-1} B_j^{t-1} x_{ij}^0\right)^{-1}$$

That is:

$$A_i^t = \overline{x_{i\bullet}} \left(\sum_j B_j^{t-1} x_{ij}^0 \right)^{-1}$$ for even rank of values $x_{ij}^2, x_{ij}^4, x_{ij}^6, \ldots$

If the iteration procedure converges, this means that:

$$A_i = \lim_{t \to \infty} \left(\prod_{k=1}^{t} a_i^k \right)$$

$$B_j = \lim_{t \to \infty} \left(\prod_{h=1}^{t} b_j^h \right)$$

is a solution of the above problem.

And if we write:

$A_i = e^{-(1+\alpha_i)}$, with $\alpha_i = -(1 + \ln A_i)$

$B_j = e^{-\beta_j}$, with $\beta_j = -\ln B_j$

we see that the outcome from the RAS method respects the first-order conditions of the cross-entropy problem. The uniqueness of the results still holds. If the RAS method converges, it converges with a unique outcome that is also the outcome of the cross-entropy minimization problem.

Macgill (1977) defined the necessary and sufficient conditions to have a solution in the RAS method:

$$\sum_i \overline{x_{i\bullet}} = \sum_j \overline{x_{\bullet j}}$$

For all $x_{gh}^0 = 0$ in the initial matrix, $\overline{x_{g\bullet}} \leq \sum_{j \neq h} \overline{x_{\bullet j}}$

For all $x_{gh}^0 = 0$ in the initial matrix, $\overline{x_{\bullet h}} \leq \sum_{i \neq g} \overline{x_{i\bullet}}$

In summary, the cross-entropy method has the advantage of allowing the user to fix cell values in addition to row and column totals. Also, it is more obvious to find a solution to problem using CE method. (JOHN: What do you mean by more "obvious"?)

4. Other Methods

4.1 Ordinary Least Squares

Just as the cross-entropy method is similar to the maximum likelihood approach in econometrics, an approach similar to Ordinary Least Squares (OLS) estimation can be used. The method is based on the Hildreth and Houck (1968) restricted model. The problem is to find a new matrix A^1 from an existing matrix A^0, by minimizing the sum of square of deviations between the new values and the prior values in absolute or percentage terms.

$$\min_{a_{ij}^1} \sum_i \sum_j \left(a_{ij}^1 - a_{ij}^0\right)^2$$; or

$$\min_{a_{ij}^1} \sum_i \sum_j \left(\frac{a_{ij}^1}{a_{ij}^0} - 1\right)^2$$

subject to:

$$\sum_i a_{ij}^1 = \sum_j a_{ij}^1$$

With OLS, no explicit assumption needs be made with respect to the distribution of the error term, but this approach assumes that there is a linear relationship between the dependent and the explanatory variables (JOHN: What are the dependent and explanatory variables in this context?).

4.2 Stone-Byron Method

Jeffery Round (2001, p.zzz) notes that: "Another method, which is analogous to the method of restricted least squares, was first discussed in a SAM context by Stone (1977), although it was suggested many years previously by Stone, Champernowne and Meade (1942) for adjusting more general sets of social accounting estimates". However, only with modern computing capacity has the method become a practical proposition (Byron, 1978). The method has since been utilized in compiling several SAMs. (JOHN: Why don't you present this method?)

4.3 Other Methods

According to Jeffery Round (2001), "other methods have been used to balance SAMs in practice, using alternative choices of the minimand". He goes on to indicate that "two are particularly worthy of mention: the first alternative is another 'quadratic' minimand; a second alternative is suggested by the similarity of the CE method (problem 2) to RAS (problem 1). A simple hybrid follows, if a cross-entropy minimand based on transactions (i.e. flows) rather than coefficients is combined with 'type 2' problem constraints and possibly additional (linear and nonlinear) constraints. Note that under particular circumstances, the entropy function is easily approximated by the 'chi-square' function".

Ludo & Surry (2002), estimating random input-output coefficients using a maximum entropy approach, state that "The generalized maximum entropy (GME) method is relatively easy to implement in practice, in contrast with other methods, such as maximum likelihood, which is computationally cumbersome and mostly requires ad hoc corrections for the initial negative variance estimates… The GME approach appears to provide a significant improvement compared to OLS and other 'classical' methods… As a result the GME method is an estimation tool that is highly recommendable to practitioners".

5. Is There a Preferred Method for Balancing a SAM?

On this issue, we cite Round (2001, p.zzz):

"Robinson, et al (2001) carry out a range of Monte Carlo experiments which suggest the superiority of the CE method over RAS in those circumstances (under problem 1 conditions) where comparisons are valid. Günlük-Senesen and Bates (1988) also conduct experiments with several balancing methods under similar problem 1 conditions and observe more mixed

outcomes. One problem in carrying out experiments is that the criteria for assessing success (the measures of closeness of an adjusted matrix to a 'true' matrix) are intimately related to the choice of minimand. Therefore there is an inherent bias built into any experimentation, which makes objectivity difficult. This is precisely why it is fruitful to discuss the choice of a preferred method by starting with a clarification of objectives. For example, do you want to get "good" coefficients, or "good" flow estimates? Are all coefficients equally "important"? What do you consider to be the nature of your prior information? Is it the overall structure of transactions (flows), or sectoral structure (coefficient)? The relatively close analytical relationships between the most frequently-used alternative methods for balancing SAMs suggest that if the required adjustments are relatively small then the differences between the methods are likely also to be small. Schneider and Zenios (1990) applied five methods to the unbalanced and highly aggregated SAM (n = 5) used by Stone (1977) to demonstrate an application of the Stone-Byron method. The differences between the resulting balanced SAMs were of a very small order of magnitude. Of course, for a higher dimensional SAM or where the required adjustments are large then the differences might be greater. In spite of the apparent preference for the cross-entropy method by many compilers of SAMs, the Stone-Byron method (possibly extended to include additional constraints) does seem to have some advantages over alternative methods. In particular it allows us to incorporate judgment on the relative reliability of data sources and is therefore closer to the spirit of the problem at hand. In fact, Stone-Byron was used to balance the Ghana SAM (Powell and Round, 1996) although, as the SUT table had been balanced prior to the rest of the SAM, the dimension of the unbalanced matrix was considerably reduced"

Round continues: "Data reconciliation methods were not as arbitrary as it might at first seem. There were essentially three steps involved in the judgment approach. First, the initial data were set alongside each other in the accounting framework to take initial stock of the problem. Secondly, a qualitative judgment was taken on the relative reliability of the alternative estimates, relying on expert local advice. Thirdly, after choosing the most reliable estimates, further scaling and adjustments were made manually to achieve consistency".

Our personal opinion is that all practitioners should maintain a strong element of the judgment approach and not see any SAM-balancing program as a magic solution. In taking stock of the initial data inconsistencies, care should be taken to ensure that all possible sources of error in the initial data have been explored and eliminated. One needs to think long and hard about the possible causes of these inconsistencies and the appropriate adjustments. For example, it is well-known that household surveys tend to understate income. Before applying a program that will modify all the cell values of a SAM to ensure consistency, one may consider adjusting the household income estimate obtained from the household survey. Also, in applying a SAM-balancing program, the program presented below allows the user to incorporate judgment concerning the relative reliability of specific cell values by fixing these values in the program.

6. A GAMS Program for Balancing a SAM[6]

We now present a GAMS code to balance an unbalanced SAM in ten steps. Only steps 1, 2 and 10 require modifications by the user, although modifications are also possible at step 8. To illustrate, we introduce new data into a previously balanced social account matrix. We also show how to add new accounts into a balanced matrix.

[6] General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical programming and optimization. For more information: www.gams.com

6.1 Define matrix accounts

The user must first declare the SAM accounts. The GAMS command “SET” is used. The account set is named I and contains all accounts including the account “TOT” for total. Instead of actual account names, it is possible to assign numbers to the accounts when the dimensions are large, e.g. A1, A2, A3… A4000. To further lighten the program, the following notation can be adopted in GAMS: A1*A4000. Here the “TOT” account will be the last: e.g. A4000. We then declare another sub-set INT of I representing all SAM accounts except the total. In this section, you can add as many other sets or sub-sets as you need (e.g. for later definition of constraints).

```
$TITLE              SAMBAL
$STITLE        SAM BALANCING PROGRAM

*1. Input names of matrix accounts

SETS I MATRIX ACCOUNTS           / LAB     labor
                                   CAP     capital
                                   HS      wage households
                                   HK      capitalistic households
                                   ENT     firms
                                   GOV     government
                                   AAGR    agricultural branch
                                   AIND    industrial branch
                                   ASERM   market services branch
                                   ASERNM  non-market services branch
                                   PAGR    agricultural products
                                   PIND    industrial products
                                   PSERM   market services products
                                   PSERNM  non-market services products
                                   SAV     savings
                                   TOT     total          /

*** Alternative code
*** I MATRIX ACCOUNTS  /     A1*A16        /

INT(I) All accounts except total
ALIAS(I,J);
ALIAS(INT,JNT);
INT(I) = YES;
INT("TOT") = NO;
*** INT("A16") = NO;
```

6.2 Input Data

In this section, the user must input the data from the unbalanced matrix “SAM0”. This can be done in many ways. Data can be imported directly from an Excel spreadsheet by using the xlimport command[7] from XLLINK program. In this case, matrix position, name and range should be given and must exclude the row and column totals. Alternatively, the original matrix can be saved in ASCII format ("prn" format in Excel) and then directly imported into the GAMS code with the "$include" command. Finally, the data can be manually copied as a "Table".

[7] xlimport and xlexport are GAMS sub-programs not available in the standard package of GAMS software. You must previously download and install the sub-program. The XLLINK program is available free of charge at: http://www.mpsge.org/xllink/xllink.htm. Note that care must be taken to ensure that GAMS is not installed in the "Program Files" directory for XLLINK to function, as it does not recognize spaces in file and folder names.

```
*2. Input initial matrix (range should exclude the row and column totals)

PARAMETER SAM0(I,J);
$libinclude xlimport SAM0 c:\My_programs_Excel\samo.xls A1:p16

*Another possibility to include SAM data
*$include SAM0.prn

$ontext
*Another possibility is to include directly SAM data in the program

TABLE SAM0(I,J)

        LAB         CAP          HS          HK         ENT      GOV
HS    28860.000                                     140.000
HK              11100.000                          1900.000
ENT              7400.000
GOV                           580.000     650.000   1480.000
PAGR                         4263.000     618.000
PIND                        11368.000    3705.000
PSERM                        9947.000    5557.000
PSERNM                                                      8700.000
SAV               2842.000    2470.000    4020.000            -2590.000
TOT   28860.000 18500.000   29000.000   13000.000   7400.000  6250.000

+             AAGR        AIND       ASERM      ASERNM        PAGR        PIND
LAB       5760.000    7560.000    8580.000    6960.000
CAP       1440.000   11340.000    5720.000
GOV                                                        180.000    2700.000
AAGR                                                      9000.000
AIND                                                                 54400.000
PAGR       120.000    3229.300     157.150     118.350
PIND      1544.000   27599.700    5479.450    1336.050
PSERM      136.000    4671.000    2063.400     285.600
TOT       9000.000   54400.000   22000.000    8700.000    9180.000   57100.000

+            PSERM      PSERNM          SAV         TOT
LAB                                          28860.000
CAP                                          18500.000
HS                                           29000.000
HK                                           13000.000
ENT                                           7400.000
GOV        660.000                            6250.000
AAGR                                          9000.000
AIND                                         54400.000
ASERM    22000.000                           22000.000
ASERNM                8700.000                8700.000
PAGR                                674.200    9180.000
PIND                               6067.800   57100.000
PSERM                                        22660.000
PSERNM                                        8700.000
SAV                                           6742.000
TOT      22660.000    8700.000    6742.000
;
$offtext
```

6.3 Normalize Initial Matrix Cell Values

The program defines a total of cell values for the unbalanced matrix, “TOTO”, and calculates its value. All matrix transaction flows are then normalized by dividing by “TOTO”. This step does not require any modifications by the user.

```
*3.Initial matrix values transformed in proportion
PARAMETER
TOTO  total of initial matrix ;
TOTO  = SUM((INT,JNT), SAM0(INT,JNT));
SAM0(INT,JNT) = SAM0(INT,JNT)/TOTO;
```

6.4 Treatment of Negative Values

The entropy method does not allow negative values. To solve this problem, we simply transpose these values to their counterpart cell before balancing the SAM. Indeed, as the SAM represents flows from one account to another, a negative flow from account A to account B is equivalent to an equal positive flow from account B to account A. For example, if SAMO(6,5) is negative, an equal amount is **added** to cell SAMO(5,6) and the cell SAMO(6,5) is set equal to zero. To recall where these negative values belong, the program first defines a new matrix, NEG. The positions of all negative values are indicated in the new matrix by a "1". This step does not require any modifications by the user.

```
*Position of negative values in unbalanced matrix are noted
PARAMETER  NEG(I,J)      Matrix of negative values;
NEG(INT,JNT)$(SAM0(INT,JNT) LT 0)=1;
*Negative matrix values are transposed
SAM0(INT,JNT)$(SAM0(JNT,INT) LT 0)=SAM0(INT,JNT)-SAM0(JNT,INT);
SAM0(JNT,INT)$(SAM0(JNT,INT) LT 0)=0;
```

6.5 Treatment of Zero Values

To avoid having to take the log of zero in the entropy method, the program adds a small amount (delta) to each cell value. This parameter must be as small as possible to avoid influencing the results. By default, it is equal to .0000000000001. This step does not require any modifications by the user.

```
*Non-zero log parameter
SCALARS
delta      Non-zero log parameter;
delta      =.0000000000001;
```

6.6 Definition of Variables Used in Optimization Process

A new matrix “NSAM” and variable “OPT” are declared. OPT is the objective function variable which has to be minimized. The number of variables is I cells of matrix plus the optimum variable. This step does not require any modifications by the user.

```
*Definition of variables
VARIABLES
NSAM(I,J)     New SAM with transposed negative values
OPT           Distance variable;
```

6.7 Definition and Initialization of Model Equations

Three sets of equations are used to balance the matrix: an optimization equation “OPTIMIZE”, defined differently for the OLS and CE approaches, and two sets of constraint equations, the

first equalizing each row total to its corresponding column total, the second setting the sum of all proportions to one.

```
*Program equations
EQUATIONS
OPTIMIZE      Optimization criterion
CONSTRAINT(I) Equality between matrix and row sums
CONSTRAINT1   proportion sum set to one
;
*Least squares (in percentage form) optimization criterion

*OPTIMIZE..    OPT=E=SUM((INT,JNT)$SAM0(INT,JNT),(NSAM(INT,JNT)/SAM0(INT,JNT)-
1)**2);

*Entropy optimization criterion

OPTIMIZE..     OPT =E= SUM((INT,JNT)$(SAM0(INT,JNT) NE 0),(SAM(INT,JNT))
                 *(LOG(SAM(INT,JNT)+delta)-LOG(SAM0(INT,JNT)+delta)));

*Equality between row and column sums

CONSTRAINT(INT).. SUM(JNT,SAM(INT,JNT))=E=SUM(JNT,SAM(JNT,INT));

*Proportions sum equal one

CONSTRAINT1..     SUM((INT,JNT),SAM(INT,JNT))=E=1 ;
```

6.8 Initialization of Variables

All variables are initialized in this section. Also, matrix cell values are limited between 0 and infinity and empty cells remain empty. No user modifications are necessary, but the user can fix any cell values (or combination of cell values), as desired. Be careful not to fix too many values, as the program may not find an optimal solution.

```
*Cell values between 0 and infinity and empty cells remain empty
SAM.L(I,J) = SAM0(I,J);
SAM.LO(INT,JNT)                       = 0 ;
SAM.UP(INT,JNT)                       = +INF ;
SAM.FX(INT,JNT)$(NOT SAM0(INT,JNT)) = 0 ;
OPT.L       = 0;

*Fix any values as desired
*SAM.FX("RP",INT)=SAM0("RP",INT);
*SAM.FX("UP",INT)=SAM0("UP",INT);
```

6.9 Model Solving

All equations are used in the model solving statement. The solver is defined in OPTION statement. The SOLVE command minimizes the variable OPT using a non linear algorithm. This step does not require any modifications by the user.

```
*Model
MODEL SAMBAL / ALL /;
```

```
*Choose solver
OPTION NLP             = MINOS;
*OPTION NLP             = MINOS5;
*OPTION NLP             = CONOPT;

SOLVE SAMBAL USING NLP MINIMIZING OPT;
```

6.10 Results Copied to a New Matrix

Results are copied into a new matrix NSAM previously defined.

```
PARAMETER  NSAM(I,J)      New (balanced) matrix;
NSAM(INT,JNT)=SAM.L(INT,JNT);
```

Negative values are re-transposed to their original position to the new matrix.

```
NSAM(INT,JNT)$(NEG(INT,JNT)=1)=-NSAM(JNT,INT);
NSAM(JNT,INT)$(NEG(INT,JNT)=1)=0;
```

The new SAM is transformed into transaction flows.

```
*Transformation in SAM transaction flow
NSAM(INT,JNT) = NSAM(INT,JNT)*TOTO;
```

The user must provide the name and range of the Excel file for exporting the new matrix. A file with this name must previously exist. It will remain unchanged except for the defined range of the SAM.

```
$libinclude xlexport NSAM c:\Diane\nsam.xls A1:P16

DISPLAY NSAM;
```

References

Bacharach, M. (1970) "Biproportional matrices and input-output change", Cambridge University Press, London.

Byron, R P (1978) 'The Estimation of Large Social Account Matrices', Journal of the Royal Statistical Society, Series A, 141 (3); 359-367.

Channing, A., Robinson S. and Tarp F. (1999) "Parameter Estimation for a Computable General Equilibrium Model: A Maximum Entropy Approach", Purdue University, IFPRI and University of Copenhagen. Discussion paper No 40.

Günlük-Senesen, G. and J. M. Bates (1988) 'Some Experiments with Methods of Adjusting Unbalanced Data Matrices', Journal of the Royal Statistical Society, 141; 473-490.

Jaynes, E. T. (1957) «Information theory and statistical mechanics», Physical Review, vol. 106, p. 620-630 et vol. 108, p. 171-190.

Hildreth, C. and Houck, J.P. (1968) " Some estimators for a linear model with random coefficients" Journal of American Statistical Association, 63, 583-594

Kullback, S. and R. A. Leibler (1951), "On Information and Sufficiency" Ann. Math. Stat. 4, 99-111

Lemelin, A. (1998a) «Le modèle de choix spatiaux discrets de Kanaroglou et Ferguson», INRS-Urbanisation, juillet, 26 p.

Lemelin, A. (1998b) «Un modèle logit multinomial avec agrégation des possibilités», INRS-Urbanisation, mai, 22 p.

Lemelin, A. (1998c) «Le modèle logit multinomial. Aide-mémoire technique», INRS-Urbanisation,Coll.Inédits, mai,18p.

Lemelin, A. (2002) «Aide-mémoire sur l'entropie», INRS-Urbanisation, Coll. Inédits, janvier, 105 p.

McDougall, R. A. (1999), "Entropy Theory and RAS are Friends".

Macgill, S. M. (1977) «Theoretical properties of biproportional matrix adjustments», Environment and Planning A, 9: 687-701.

Macgill, S. M., and A. G. Wilson (1979) «Equivalences and similarities between some alternative urban and regional models», Sistemi Urbani, 1: 9-40.

Miller, Ronald E., et Blair Peter (1985) "Input-output analysis : foundations and extensions", Prentice-Hall Inc., Englewood Cliffs, N. J.

Peeters, L. and Surry Y., 2002, "Maximum entropy estimation of random input-output coefficients : An application to Saskatchewan farm accounting data" Limburg University Center – INRA. Presented at the 10th Congress of the European Association of Agricultural.

Robillard, A. S., and Robinson S., 1999, "Reconciling Household Surveys and National Accounts Data Using a Cross Entropy Estimation Method", IFPRI. Discussion Paper No 50.

Robinson, S., Cattaneo A. and El-Said M. (2000), "Updating and Estimating a Social Accounting Matrix Using Cross Entropy Methods", IFPRI. Discussion Paper No 58.

Robinson, S., and El-Said M. (2000), "GAMS Code for Estimating a Social Accounting Matrix (SAM) Using Cross Entropy (CE) Methods", IFPRI. Discussion Paper No 64.

Robinson, S., Cattaneo A. and El-Said M. (1998) "Estimating a Social Accounting Matrix Using Cross Entropy Methods", IFPRI. Discussion Paper No 33.

Round, J. I., (2001), "Constructing SAMs for Development Policy Analysis Department of Economics", University of Warwick. Prepared for a seminar "Data Analysis for Poverty and Income Distribution Analysis", The Hague, Amsterdam.

Shannon, C. E. (1948), A Mathematical theory of communication", Bell System Technical Journal 27, 379-423.

Stone, R (1977) 'Forward' to G Pyatt and A Roe et al, Social Accounting for Development Planning, Cambridge University Press, Cambridge; xvi-xxxi.

Stone, J. R. N., D. Champernowne and J. E. Meade (1942) 'The Precision of National Income Estimates', Review of Economic Studies, Vol IX, No 2; 111-125.

Theil, H. (1967), Economics and information theory, Rand McNally & Company, Chicago, Studies in mathematical and managerial economics, 7, 488 pages.