XIII International Conference on

Input-Output Techniques

University of Macerata, Italy August 21-25th, 2000.

Vittorio Nicolardi

(University of Bari – Faculty of Economics)

Balancing large accounting systems: an application to the 1992 Italian I-O Table.

Summary

The balancing of large accounting systems has become a fundamental issue in the study of National Accounts, because their economic data is becoming increasingly disaggregated. The solution to this problem involves considerable data management. Starting from Stone's methodology integrated with the conjugate gradient method, an algorithm for resolving the problem of balancing large accounting systems has been defined. By dividing the accounting matrix in blocks, this algorithm is able to manage accounting structures of tens of thousands of equations (accounting constraints) of magnitude, even on systems with relatively little computational power, thus overcoming the limits of previous methodologies proposed in the literature. A meta-language for defining the blocked accounting equation system and directly used by the defined algorithm is also presented.

Keywords: Matrix accounting structures; Block solution for balancing problems; *EULERO* meta-language.

1 Introduction[1].

One of the major problems one is faced with when compiling I-O Tables is balancing them.

As part of an ISTAT (Italian National Statistical Institute) project for compiling the Input-Output (I-O) Table for 1992 a balancing procedure has been developed based on the method first introduced by R. Stone (1960). He described his method in a series of articles on problems relating to the precision of estimates of the national income and which had already been investigated in a collaborative study carried out with D.G. Champernowne and J.E. Meade (1942). The method has since been frequently used and perfected by the author for other accounting schemes, whatever their type (Stone 1961, 1968, 1970, 1975), and for compiling the Social Accounting Matrices (SAM; Stone 1976)

The merit of Stone's method lies in its great flexibility. It has already been used by the ISTAT for many years for balancing Italy's annual accounts.

However, despite its undeniable advantages in terms of flexibility, the method poses huge problems in terms of use when the quantity of data to be manipulated exceeds a specific critical threshold and requires the use of sophisticated mathematical and data management techniques.

From this point of view, the complexity of the I-O accounting scheme and the large quantity of data required to compile it, as well as the complex structure of the balancing equations, make balancing the I-O Table one of the most difficult procedures to undertake when dealing with national accounts.

The complexity of the balancing scheme has been made even more formidable by the need to apply the new norms specified in the ESA (European System of Economic Accounts) for 1995; these require that the I-O Table should be compiled in a sequence of tables calculated at different price levels (ranging from basic prices right up to market prices) and at different levels of disaggregation (total flows, domestic production flows, import flows).

The aim of this study was to find a procedure that would make it possible to balance all the accounting items in the I-O Table simultaneously, so as to avoid the cascade balancing system hitherto adopted for national accounts, and which would, at the same time, be highly flexible when defining the structure of the balancing equations.

The final result of the research was in line with the initial project, since a procedure was formulated that can be applied to any accounting equation structure, and in which accounting items can be represented by means of matrices and vectors (block structures).

Furthermore, in order to make the balancing method as flexible and direct as possible, a special computing meta-language, *EULERO*, was developed, so that the equations can be set out in account blocks. *EULERO* meta-language has an extremely simple syntax; the balancing equations can be written in implicit form, keeping as close as possible to traditional mathematical notation(2).

The structuring of the balancing procedure and the adoption of the *EULERO* programming language facilitate the manipulation of accounting items by drastically reducing the amount and computational complexity of the whole procedure. Moreover its use with data processing systems that are not particularly powerful is also made feasible.

A brief description of Stone's balancing method will be given. Then the conjugate gradient method underlying the present study will be described, after which the block system balancing method and the *EULERO* meta-language will be examined.

2 Stone's method.

---

1 I thank Ms Kate Hunt for her help in translating this paper. However, any possible error and/or omission is my own responsibility.

2 With *EULERO* the structure of the equations can be defined using a simple file in text format. It carries out a series of simple basic operations which can be applied to the different data structures and makes it easy to write and adjust the equation system.

The problem that Stone tried to solve was to create a method for making it possible to balance a set of economic items, simultaneously allowing for differences in their reliability and significance. Each estimate of an economic item is usually affected by a different level of error depending on how the item has been obtained, on the different sources of data available and on the nature of the item itself.

The classic methods used for balancing accounting systems had been based essentially on mathematical methods which apportioned the residuals of the accounting system among the different aggregates almost exclusively in proportion to the weighting of each item vis-à-vis the total value of the aggregates.

Although balancing methods were developed whereby the weighting system was able to apportion the residuals among the different items in a ‘reasoned’ way, the final results were never completely satisfactory from the economic point of view.

Stone (1960), however, treated the problem as one of constrained estimates, whereby the final balanced estimates of the accounting system were obtained by ‘adjusting’ the initial estimates of the different accounting items. This came about by using a system of ‘weights’ which would express the level of reliability of each initial estimate whereby the constraints would be those expressed by the system of equations and accounting identities: the residuals could be apportioned within the system in a ‘reasoned’ way, and the less reliable estimates would be adjusted to a greater extent than those with greater reliability, and the known estimates would not be changed at all.

To explain Stone’s method, let us consider a set of items for economic aggregates of a given country organized into a coherent and complete accounting scheme.

The items can be organized in a matrix *X* to express the system of accounting constraints as follows:

$$\begin{aligned} &\sum_i x_{ij} = x_{.j} \\ &\sum_j x_{ij} = x_{i.} \\ &x_{.j} = x_{i.} \quad i, j = 1,.., \mathrm{n} \end{aligned} \qquad [1]$$

where the x_{ij} represent the *ij-nth* element of the matrix *X*, $x_{i.}$ and $x_{.j}$ the vectors of the totals of rows and columns respectively of the matrix itself and *n* the size of *X*, generally coinciding with the number of equations of the accounting scheme. In [1] the vectors $x_{i.}$ and $x_{.j}$ coincide, so that each corresponding row and column in *X* together form an accounting identity. Moreover, since the accounting system forms a complete structure the matrix *X* has the order *(n-1)*.

For defining the system of accounting equations, alternatives to the schemes suggested can be used, although the balancing restrictions of the aggregates can basically be set out as in [1].

As already mentioned, when compiling national accounts the different aggregates are not coherent with each other, because of the different methods and procedures for obtaining them and because of errors; thus the accounting constraints of *[1]* are not respected, and the account systems in *X* are not balanced, and we get:

$$x_{i.} \neq x_{.j}$$

or

$$x_{i.} - x_{.j} \neq 0$$

The imbalance of the accounting system can be formulated more generally as follows:

$$x_{i.} = x_{.j} + h_i \qquad [2]$$

whereby one or more accounting equations may be assumed by definition to have a non-zero residual: the structure of these residuals is contained in the vector *h*.

Through his method, Stone aimed to obtain estimates of the accounting aggregates of the non-balanced system to satisfy conditions *[1]*, with generalization *[2]* and which are at the same time statistically and economically meaningful.

According to this method the accounting structure *[1]* is expressed by means of a vectorization x of X, to which the accounting constraints are applied by a matrix G of the order $(k \times s)$, where k is the number of accounting equations expressed in *[1]* and s the number of elements in X.

The matrix G contains values *0*, *1* and *–1* depending on whether or not each element of x is written in a defined equation of *[1]*, expressed by the corresponding row of G, and with which sign.

Starting from an initial vector of estimates $\hat{x}$ of x, a vector $\tilde{x}$ has to be obtained which respects the accounting constraints of *[1]*-*[2]*, since in the initial non-balanced system we have:

$$G\hat{x} \neq 0 \quad \text{or} \quad G\hat{x} - h = 0 \qquad [3]$$

depending on whether scheme *[1]* is used with or without generalization *[2]*.

As already mentioned, Stone started from the consideration that the level of significance of each estimate in $\hat{x}$ is generally different, hence it is better that the balancing process affects the estimates with a lower level of significance less than those with greater significance.

For this purpose an a priori matrix of weights V is used, defined as a matrix of variances and covariances; it provides the balancing system with "information" on the reliability of the estimates of the accounting values in $\hat{x}$ and hence on those items that have to be adjusted and by how much, so as to reach an accounting equilibrium of the system by distributing the amount of the residuals $G\hat{x}$ among the different accounting aggregates: the values in $\hat{x}$ to which greater weights correspond in V undergo greater adjustment during the balancing process and vice versa, while the zero-weighted items will not be changed.

It is important to observe that, given the structure of the matrix V, which will be discussed at greater length further on, the residuals are not apportioned within the accounting system on the basis of the absolute values of the weights but rather on the basis of the relative values within the overall system of weights: it can be deduced therefore if V is multiplied by a scalar the result of the balancing process does not change.

Following the reasoning hitherto described, the balanced estimates of the accounting system can be obtained through the restricted estimator:

$$\tilde{x} = \hat{x} - VG'(GVG')^{-1}(G\hat{x} - h) \qquad [4]$$

In practice, *[4]* is an application of the constrained generalized least square method, so that, if V is considered as the matrix of the variances and covariances of the estimates in $\hat{x}$, albeit a priori, the estimate of the matrix of the variances and covariances of $\tilde{x}$ is given by:

$$\Sigma_{\tilde{x}\tilde{x}} = V - VG'(GVG')^{-1}GV \qquad [5]$$

Theil (1961) has shown that estimator *[4]* has the properties of *BLUE (Best Linear Unbiased Estimator)* estimators.

Taking into account that the matrix V can be considered as the matrix of the variances and covariances of $\hat{x}$, then it must be considered as containing values in the same unit of measure as the accounting values.

Accordingly, a relatively simple way to determine V is to give the values in $\hat{x}$ a weighting from *0* and *1* which expresses the degree of reliability of each single item and to multiply each element of $\hat{x}$ by that weight: so, if we give *0*-weighting to the reliable data and *1* to the unreliable data and intermediate values to the remaining data according to the relative reliability of the different aggregates, the balancing system will not change the values which have been given *0*-weighting; it will, however, vary the other elements in $\hat{x}$ according to the value of their weighting, and calculate the *1*-weighted values "residually".

It is in any case necessary to remember that, given the importance of the relative weightings in the balancing process with regard to the absolute values, each element in $\hat{x}$ will not be readjusted in direct proportion to its weighting, but rather according to its relative value within all the equations containing the element and to the relative value of the accounting residuals, in both absolute and relative terms.

Finally, it is also necessary to bear in mind that the matrix *V* is a matrix of the order ($s \times s$), if *s* are the elements in $\hat{x}$, so that the variances of the elements of $\hat{x}$ are contained in its main diagonal, while the elements outside the main diagonal contain the covariances of the said estimates: accordingly, it is necessary to find a method for defining an estimate of the covariances of the initial estimates in $\hat{x}$ as well, which is not easy since the weightings of the different items are "a priori" values.

However, the matrix *V* is generally used in the diagonal, since the presence of non-zero elements outside the diagonal would assume there was a correlation between the estimates of different aggregates, but this is in fact infrequent when compiling national accounts.

Even when there is assumed to be a correlation among the estimates, the diagonal of *V* would be used, because of the above-mentioned practical difficulty in defining the value of the elements of V outside the main diagonal and the practical impossibility of finding sample distributions of the estimates in $\hat{x}$, given that these are basically punctual and not repeatable.

From the strictly methodological point of view considering *V* as a diagonal introduces a factor of inefficiency of the estimator $\tilde{x}$: however, in practice, the level of error encountered is negligible in most cases, and is in any case lower than an error that could ensue from not correctly specifying the correlation between the estimates.

3 The conjugate gradient method.

The major problem currently posed when compiling and balancing the I-O Table relates to the ever greater disaggregation of the table's structure; on the one hand this means having to manage larger and larger quantities of items and on the other having to use increasingly complex accounting equation schemes, with immense computational problems as a consequence.

In particular, in searching for the solution to *[4]*, the problem of the inversion of the matrix *(GVG')* is important as is the handling of the matrices G and V, which, for complex accounting structures, reach huge magnitudes, not only because of the number of accounting equations in the balancing scheme of the table but also because of the number of accounting items used in compiling the table and their underlying information structure[3].

In fact, in a system made up of *k* equations and *s* elements (the number of the accounting items in $\hat{x}$), the matrix G will be of the order $(k \times s)$, V of the order $(s \times s)$, and *(GVG')* of the order $(k \times k)$. In the specific case of the balancing scheme created for the 1992 I-O Table for market – producers' prices (including invoiced VAT) alone was made up of 93,629 equations (*k*) and 148,789 elements (*s*).

A further computational problem is the fact that the matrix *(GVG')* is a sparse matrix, which means that it has a very low ratio between non-zero elements and the overall total of elements. This requires ad hoc numerical procedures to reduce calculation times and errors in determining the inverse matrix of *(GVG')*. The errors are due to problems connected with the computational complexity of the inversion method and to the propagation of errors relating to the numerical stability of the solutions of the inversion procedure itself.

As an alternative to directly applying the method of estimating the constrained quadratic minimums an interesting approach for reaching a solution of *[4]* is to redefine the problems in terms of quadratic loss of the information content of the restricted estimates in $\hat{x}$, a loss which is due to the balancing process itself (Byron 1978). According to this approach, the new balanced estimates $\tilde{x}$ are obtained by minimizing this loss function, whereby the balancing restrictions of the system

[3] In the case of balancing, this information relates to the level of reliability of the estimates and is necessary for defining the matrix *V* in [4].

are expressed as a penalty by means of Lagrange multipliers: this approach leads to a definition of an estimator which is basically equal to the one Stone obtained, but an equation structure can be defined more directly and which are suited to the application of computational methods with a greater stability of solutions; it also highlights how the balancing process, in modifying the value of the accounting items, also modifies the information content of the initial non-balanced item, besides its amount.

According to this approach, the quadratic loss function can be defined as:

$$Z = \tfrac{1}{2}(\tilde{x} - \hat{x})' V^{-1}(\tilde{x} - \hat{x}) + \lambda'(G\tilde{x} - h) \qquad [6]$$

where the term λ is the vector of Lagrange multipliers: this relation requires that the new estimates in $\tilde{x}$ be as close as possible to the estimates in $\hat{x}$ in terms of quadratic loss, but such as to satisfy the restriction $G\tilde{x} = h$.

The first-order conditions on *[6]* are given by:

$$\begin{aligned} \frac{\partial Z}{\partial \hat{x}} &= V^{-1}(\tilde{x} - \hat{x}) + \lambda' G = 0 \\ \frac{\partial Z}{\partial \lambda} &= G\tilde{x} - h = 0 \end{aligned} \qquad [7]$$

whose solution is given by:

$$\begin{aligned} \tilde{\lambda} &= (GVG')^{-1}(G\hat{x} - h) \\ \tilde{x} &= \hat{x} - VG'\tilde{\lambda} \end{aligned} \qquad [8]$$

which are equivalent to *[4]*.

The solution of *[6]* can be obtained in different ways and initially Stone himself used a procedure which envisaged the partitioning of vector *x* into blocks and the partitioned inversion of the matrix *(GVG')*. However, this solution requires a great number of calculations and has a basically high computational cost even for current data processing systems.

Byron (1977) subsequently proposed a more efficient method for solving the problem of the inversion of the matrix *(GVG')* using the conjugate gradient method: this method is widely used for the solution of optimization systems and can easily be adapted to the solution of linear equations such as those underlying the accounts of the I-O Table.

It can be seen from the system of equations already described that the first of *[8]* is a system of linear equations; and since the matrix *(GVG')* is symmetric positive definite, the same relation can be rewritten as the product of three sparse matrices thus:

$$(GVG')\lambda = q \quad o \quad A\lambda = q \qquad [9]$$

and the conjugate gradient method can be applied.

The mathematical solution of *[9]* can be obtained by using an iteration scheme:

$$\begin{aligned} \pi_0 &= \rho_0 = q - A\lambda_0 \\ \alpha_i &= \rho_i' \rho_i / \pi_i' A \pi_i \\ \lambda_{i+1} &= \lambda_i + \alpha_i \pi_i \\ \rho_{i+1} &= \rho_i - \alpha_i A \pi_i \\ \beta_i &= \rho_{i+1}' \rho_{i+1} / \rho_i' \rho_i \\ \pi_{i+1} &= \rho_{i+1} + \beta_i \pi_i \end{aligned} \qquad [10]$$

where π_0 and ρ_0 are gradient-based direction vectors, λ_0 is a vector of initial values for λ and *i* and *i+1* refer to the iteration count.

The iterative process is stopped when all the elements in the vector $(\theta - A\lambda_i)$ have an absolute value lower than a given fixed value ε.

The solution of *[4]* will therefore be given by the second of *[8]*, where $\tilde{\lambda}$ is the vector λ_i resulting from the iterative procedure *[10]*.

The system is seen to converge in a maximum of *k* iterations[4], if *k* is the dimension of λ, that is the number of accounting equations, or less than *k* if the system of weights has been appropriately defined and the values of the matrix have been represented in a suitable scale.

The convergence process can be greatly accelerated by appropriately transforming the matrix *(GVG')* by means of a diagonal matrix *W* whose elements are given by:

$$w_{i,i} = (\sum_j g_{i,j} V_{j,j} g'_{i,j})^{-1} \quad [11]$$

so that *[9]* can be rewritten as:

$$W\,(GVG')W\lambda^* = Wq \quad [12]$$

where $\lambda^* = W^{-1}\lambda$.

The transformation of *(GVG')* is a normalization of the initial matrix, whereby a positive definite matrix can be obtained with unit elements on the main diagonal, while all the other elements are less than *1* in absolute value.

In an article on the estimation of large scale *SAMs* Byron (1978) estimated that for the correct application of the conjugate gradient method, besides memorizing the matrix *(GVG')*, the number of vectors that had to be memorized was $4\times(3n\text{-}3)$, with *n* equal to the dimension of *X;* thus, although there is a reduction in computational costs compared to Stone's original proposal, it is however still very high for large accounting systems.

4 A block calculating system for balancing large accounting systems with a diagonal variance matrix.

The problems connected to the high computational cost of Stone's method for the solution of very large accounting systems can be overcome using appropriate variations of the methods so far presented.

In particular two specific aspects relating to balancing methods will be examined below:

- the reformulation of the first of [8] so as to drastically reduce the number of vectors of items which need to be memorized;
- the reformulation of the accounting structure *[1]* according to a block system, so that the matrix *G* of the restrictions can be expressed in implicit form and not be included in finding the solution of *[8]*.

Henceforth, approach *[12]* will be used, since it has greater convergence velocity than the iterative procedure *[10]*.

4.1 A reduced calculating system for balancing large accounting systems with a diagonal variance matrix

In applying Stone's method in both form *[4]* and form *[8]*, the main problem when using the conjugate gradient method relates to handling the matrix *(GVG')*.

If scheme *[10]* is taken into account it can be seen that the management of the *(GVG')* matrix is affected in the second and fourth relations. In both relations the product can be isolated:

4 This property is valid for the solution of linear equations using symmetric positive definite matrices and for the maximization of quadratic functions (see Comincioli, 1995 for example). Accordingly it is on average more efficient mainly for large accounting systems than the *RAS* method, which is a bi-proportional balancing structure with "borderline" convergence. For a more detailed comparison between the RAS method and the conjugate gradient method see Van Der Ploeg's article (1982).

$$\mathrm{A}_{(k\times k)}\pi_{(k)} = (G_{(k\times s)}V_{(s\times s)}G'_{(s\times k)})\pi_{(k)} \qquad [13]$$

where the dimensions of the different vectors and matrices are given in the subscripted brackets, with *k* as the number of equations and *s* the number of elements of the accounting system.

Taking account of transformation *[12]*, *[13]* will become:

$$\mathrm{A}_{(k\times k)}\pi_{(k)} = W_{(k\times k)}(G_{(k\times s)}V_{(s\times s)}G'_{(s\times k)})\pi_{(k)} \qquad [14]$$

It will be seen that the product at the second member of *[14]* is a vector of magnitude *k* and that if this product is obtained by starting with the last pair of elements and proceeding to the first, the subsequent results are still formed of vectors:

$$\begin{aligned} G'_{(s\times k)}\pi_{(k)} &= \Gamma_{(s)} \\ V_{(s\times s)}\Gamma_{(s)} &= \Gamma'_{(s)} \\ G_{(k\times s)}\Gamma'_{(s)} &= \Gamma''_{(k)} \\ W_{(k\times k)}\Gamma''_{(k)} &= \Gamma'''_{(k)} \end{aligned} \qquad [15]$$

If *V* is taken to be a diagonal matrix, procedure *[15]* requires that only 3 vectors be memorized relating to *V*, *W* and π; remembering that *W* is also a diagonal matrix, as well as temporary vectors for the intermediate products.

4.2 A block scheme for defining large accounting systems.

The accounting scheme *[1]* can be further simplified if instead of considering the single values within the matrix *X*, appropriate sets of them (blocks) are considered in both vectorial and matricial forms, where the system of equations can be redefined as sub matrices of the complete accounting system *X*.

In the case of the I-O Table this approach turns out to be particularly efficient, since the accounting structure underlying it can easily be formulated as a system of relations between accounting matrices. For example, considering a simplified scheme of the I-O Table at market prices and considering only the subdivision between total flows (${}^{pm}X^{T}$), import flows (${}^{pm}X^{M}$) and domestic production flows (${}^{pm}X^{I}$), the system of accounting constraints between the different sections can be schematized as follows:

$$
{}^{pm}x^{T}_{ij} = {}^{pm}x^{M}_{ij} + {}^{pm}x^{I}_{ij} \quad \forall i, j
$$

$$
\begin{cases}
\sum_{j} {}^{pm}x^{T}_{ij} = {}^{pm}x^{T}_{i.} & \forall i \\
\sum_{i} {}^{pm}x^{T}_{ij} = {}^{pm}x^{T}_{.j} & \forall j \\
\sum_{i,j} {}^{pm}x^{T}_{ij} = {}^{pm}x^{T}_{..}
\end{cases}
$$

$$
\begin{cases}
\sum_{j} {}^{pm}x^{M}_{ij} = {}^{pm}x^{M}_{i.} & \forall i \\
\sum_{i} {}^{pm}x^{M}_{ij} = {}^{pm}x^{M}_{.j} & \forall j \\
\sum_{i,j} {}^{pm}x^{M}_{ij} = {}^{pm}x^{M}_{..}
\end{cases}
$$

$$
\begin{cases}
\sum_{j} {}^{pm}x^{I}_{ij} = {}^{pm}x^{I}_{i.} & \forall i \\
\sum_{i} {}^{pm}x^{I}_{ij} = {}^{pm}x^{I}_{.j} & \forall j \\
\sum_{i,j} {}^{pm}x^{I}_{ij} = {}^{pm}x^{I}_{..}
\end{cases}
\qquad [16]
$$

where x_{ij} represents the *ij-nth* element of the different matrices, $x_{i.}$ and $x_{.j}$ the respective vectors of the row and column totals of the respective matrices and $x_{..}$ the relative totals, with $i=1,\ldots,n$ e $j=1,\ldots,m$. In this scheme the restriction conditions on the sum of the vectors of the row and column totals are not indicated nor those on the total sums of the matrices, thus:

$$
\begin{aligned}
&{}^{pm}x^{T}_{i.} = {}^{pm}x^{M}_{i.} + {}^{pm}x^{I}_{i.} \quad \forall i \\
&{}^{pm}x^{T}_{.j} = {}^{pm}x^{M}_{.j} + {}^{pm}x^{I}_{.j} \quad \forall j \\
&{}^{pm}x^{T}_{..} = {}^{pm}x^{M}_{..} + {}^{pm}x^{I}_{..}
\end{aligned}
\qquad [17]
$$

since they are implicit in the remaining equations.

From *[16]*, it will be seen that this can be reorganized according to a block scheme:

$$
\begin{aligned}
&X^{T} - X^{M} - X^{I} = 0 \\
&{}_{sr}X^{T} - {}_{r}x^{T} = 0 \\
&{}_{sr}X^{M} - {}_{r}x^{M} = 0 \\
&{}_{sr}X^{I} - {}_{r}x^{I} = 0 \\
&{}_{sc}X^{T} - {}_{c}x^{T} = 0 \\
&{}_{sc}X^{M} - {}_{c}x^{M} = 0 \\
&{}_{sc}X^{I} - {}_{c}x^{I} = 0 \\
&{}_{st}X^{T} - {}_{t}x^{T} = 0 \\
&{}_{st}X^{M} - {}_{t}x^{M} = 0 \\
&{}_{st}X^{I} - {}_{t}x^{I} = 0
\end{aligned}
\qquad [18]
$$

where matrices X are the same as in scheme *[16]*, omitting the suffixes *pm* for convenience, and taking the subscripts *sr*, *sc* and *st* to represent respectively the sums for row, column and totals of the matrices themselves, while vectors ${}_{r}x$ and ${}_{c}x$ are the restriction vectors of row and column respectively and ${}_{t}x$ the restriction for the sum of the different matrices.

Each i-nth row of *[18]* forms a macro equation which compactly defines a system of r_i equations, where r_i is the number of elements of each of the blocks (matrices or vectors) forming the macro equation.

Note that all the relations of *[18]* have been rendered implicitly and that in the scheme it is important that the row restrictions be represented by column vectors, and the column restrictions by row vectors. Moreover, of the vectors and matrices of scheme *[18]* only *q* are formed of the initial structures of items supplied to the system (12 in our case, namely the *X* matrices of the items and the restriction vectors and scalars), while the remaining ones are defined by algebraic operations on these matrices: for example ${}_{sr}X^T = X^T u$, where *u* is the unit vector.

By arranging the accounting system into the block scheme *[18]*, product *[14]* can be calculated using scheme *[15]* by an appropriate block multiplication procedure without making the matrix *G* explicit: this possibility turns out to be of fundamental importance, since problems connected to the management of the said matrix can be avoided as well as the time required for its definition, this operation being extremely complex in some cases. In particular the second property simplifies and accelerates the redefinition of the accounting system[5].

Observing *[15]* carefully, it will be noted that in actual fact two procedures are required for effecting the product in blocks, since the intermediate solutions of the scheme refer to two different vectors of *s* and *k* dimensions respectively, relating to the first and the second of *[15]*. As for the second and fourth of *[15]*, since *V* and *W* are two diagonal matrices and since it is possible to memorise them in vectorial form, these are reduced to the product element by element of the vectors that they form.

In the first procedure *(A)*, it is first of all necessary to decompose π into *k* blocks sequentially, equal to the number of macro equations of scheme *[18]*, since they have the same dimensions as the blocks forming each macro equation; it is also necessary to arrange a list (Ω) according to the order in which the initial *q* structures of the items (vectors or matrices) are inserted row by row into the balancing scheme. The output of the procedure will be *q* blocks (matrices or vectors) resulting from the linear combination of *k* blocks of π according to the general solution:

$$B_j = \left(\sum_i \mathrm{Z}_i \right)^{el} \qquad j = 1, \ldots, q; \;\; i \in \Lambda \qquad [19]$$

where B_j are the *q* blocks of the output of procedure *(A)*, Z_i the general *k* blocks into which π is decomposed and finally Λ is the set of indices indicating in which of the *k* macro equations each matrix corresponding to the *j-nth* element in Ω is present, both as original matrix and as transformed matrix: each of the Z_i blocks will be inserted in the linear combination with the sign that the corresponding matrix in Ω_j will have in the *i-nth* equation[6]. The vectorization of the *q* blocks thus obtained will give the Γ vector.
The superscripted symbol *el* in *[19]* indicates that the sums have to be taken element by element, as explained in the M*ethodological Appendix*.

In the second procedure *(B)* Γ' has to be decomposed into *q* blocks equal to the number of elements in Ω and be of the same magnitudes as the respective item structures. The output of the procedure will be *k* blocks (matrices or vectors) resulting from the linear combination of the *q* blocks of Γ' according to the general solution:

$$B_j = \left(\sum_i \mathrm{Z}_i \right)^{el} \qquad j = 1, \ldots, k; \;\; i \in \Lambda \qquad [20]$$

where B_j are the *k* blocks of the output of procedure *(B)*, Z_i the general *q* blocks into which Γ' is

5 In actual fact the balancing routine uses the syntax of the *EULERO* language to effect the series of products [15] without the need to define the matrix *G* explicitly.

6 As an example, with reference to scheme [18], block B_1 will be given by the sum of blocks 1, 2, 5, 8, since *XT* will be the first of the initial independent matrices of the data of vector Ω, and it will be present in the above equations both as an original matrix (*XT*), and as a transformed matrix (${}_{sr}XT$, ${}_{sc}XT$, ${}_{st}XT$).

decomposed and finally Λ is the set of indices indicating the position that the matrices of each *j-nth* equation occupy in Ω : each of the Z_i blocks will be inserted into the linear combination with the sign that the corresponding matrix in Ω_j will have in the *i-nth* equation[7]. The vectorization of the q blocks thus obtained will give the Γ vector.

The superscripted *el* symbol in *[20]* has the same meaning as in *[19]*.

It is interesting to note that if the values of the main diagonal of *V* are organized into blocks as described in scheme *(B)*, the second of *[15]* can be obtained as product element by element of the blocks of *V* for the blocks resulting from the application of scheme *(A)*, thus eliminating the sequential vectorization operation for determining Γ and its successive re-retransformation into blocks.

The block scheme described above was tested for different accounting schemes by developing an *ad hoc* programme of symbolic resolution of products *[15]*, in order to test its actual generalizability[8].

From the point of view of the computational cost, the block calculating procedure is on average more efficient than the original conjugate gradient procedure, in direct proportion to the dimension of the data structure.

The block procedure has the advantage of significantly reducing the set of items required for its implementation, thus allowing the whole procedure to be carried out in the central memory of the computer; on the other hand it does have the disadvantage of having to carry out procedures *(A)* and *(B)* at each iteration of the conjugate gradient method. However, the block procedure avoids having to define the *G* matrix, whose complexity we have already mentioned.

The classic procedure requires the matrix *(GVG')* to be calculated just once, but at the same time poses problems of memorization of the said matrix, which, for complex accounting schemes will have to be suitably partitioned and stored in bulk memory.

In the tests carried out the extra computational load due to the repetitive execution of procedures *(A)* and *(B)* turned out to be lower than the extra computational load of managing the *(GVG')* in bulk memory[9]; thus the block procedure turns out to be faster than the original procedure. Furthermore it also must be added the savings in time due to the non-definition of *G*.

An examination of the application of the conjugate gradient scheme shows how the block solution can also be applied for the second of *[8]* and for checking the stop condition of the iterative procedure *[10]*:

$$\theta - \mathrm{A}\lambda_i < \varepsilon \qquad [21]$$

5 The *EULERO* meta-language for defining the block balancing procedure.

For the operative implementation of procedures *(A)* and *(B)* a special meta-language, *EULERO*, has been defined, whereby the accounting equations of the system to be balanced can be expressed as closely as possible to the block structure of scheme *[18]*; thus it creates a calculation routine that can be applied to accounting structures, however they are defined, which makes the system extremely flexible.

From the operational point of view the meta-language is made up of a reduced set of key words (*commands*) whose purpose is to structurally define the equation system (for the implicit definition of *G*), as well as the operations that can be carried out on the various matrices.

The commands are:

7 As an example, referring to scheme [18], block B_1 will be given by the sum of blocks 1, 2, 3 since in these positions the matrices of items included in the first of [18] are in Ω.

8 The programme of symbolic analysis was carried out in the *Speakeasy* language, the same language as the one in which the whole 1992 I-O balancing procedure was carried out.

9 This extra load is mainly due to the times for accessing magnetic bulk memories which are slower by at least 6 orders of magnitude than the central memory; these are reduced to 3 if electronic bulk memories are used.

- ***MM*** : each element of a defined matrix is inserted into a different equation of the system;
- ***VR*** : each element of a defined row vector is inserted into a different equation of the system;
- ***VC*** : each element of defined column vector is inserted into a different equation of the system;
- ***SR*** : each element of the column vector given by the sum per row of the elements of a defined matrix is inserted into a different equation of the system;
- ***SC*** : each element of the row vector given by the sum per column of the elements of a defined matrix is inserted into a different equation of the system;
- ***SM*** : the element given by the sum of the elements of a defined matrix is inserted into a different equation of the system.

By using these commands the balancing scheme of any accounting structure can be easily schematized by means of a simple command sequence *(script)*.

The balancing procedure implemented is formed of two sub-procedures independent of each other: in the first (*interpreter*), the script is read and interpreted, analyzing the commands of the meta-language and the signs with which the single matrices or vectors are written into the balancing system, generating as output the pointers required for the development of the product in blocks (Λ sets); in the second, the actual balancing is carried out using the output structures generated by the first sub-procedure.

6 Applying the block balancing procedure and the *EULERO* language to the 1992 Italian I-O Table.

An application of the balancing method using the block solution method described in the preceding paragraphs was carried out within the framework of the ISTAT project for compiling the 1992 I-O Table

Both the interpreter and the balancing procedure, according to the specific requirements of the ISTAT project, were written into the *Speakeasy* language: the PC version of *Speakeasy* was used for the prototype, subsequently adapting the programmes to the same language in UNIX.

The tables for market and producers' prices (including invoiced VAT), subdivided into domestic production flows, import flows and total flows were balanced simultaneously.

6.1 Defining the balancing scheme.

In order to balance the I-O Table, its structure was divided into a table of intermediate flows, table of final uses and table of primary sectors, and into domestic production flows, import flows and total flows.

Taking into account that the scheme for simultaneously balancing market and producers' prices (including invoiced VAT) also requires matrices of trade and transport margins, and that these are subdivided into intermediate flow margins and final flow margins, for domestic production, imports, and totals, the overall number of basic accounting matrices required is 26.

Furthermore three vectors for IVA on domestic flows, imports and totals must be added.

Taking also into account the restrictions for row, column and total expressed for the different matrices then the number of accounting structures used rises to 86.

Bearing in mind that the ESA for 1995 I-O requires that the I-O Table be compiled for over 101 productive industries, the number of equations of the balancing system will be 93.629, with 148.789 accounting elements.

Using the classic conjugate gradient method solution it would have been necessary to define a matrix G of magnitude ($93{,}629 \times 148{,}789$) and handle a matrix *(GVG')*, corresponding to the matrix A of the conjugate gradient method, of magnitude ($93{,}629 \times 93{,}629$), which is not manageable in the central memory of an average processor.

6.2 *Defining the scheme in the EULERO meta-language.*

Appendix A gives the script in the *EULERO* meta-language used for balancing the 1992 I-O Table at market and producers' prices (including invoiced VAT).

The names of the accounting structures used in the procedure were codified according to a 7-letter code for two fundamental reasons.

Firstly, *Speakeasy* allows for a maximum of 8 letters for the names of the variables; thus it was necessary to reduce the names of the aggregates of items to a maximum length of 8 letters.

Moreover, to make the balancing procedure as simple as possible a convention was adopted in defining the names whereby their maximum length was reduced to 7 letters: if "DATA" is the name of one of the accounting structures, the system expects to find the variances (*weights*) relative to the values contained in it in a variable with the name "VDATA", and returns the balanced values of the same structure in a variable of the name "QDATA".

Naturally, this scheme can be adjusted by changing the interpreter of the *EULERO* commands.

In the ISTAT application the 7-letter code used was the following:

- the first letter is ***X*** if the data structure is a matrix, ***R*** if it is a row vector, ***C*** if it is a column vector, ***T*** if it is a total value (the sum of the elements of a vector or matrix);
- the second letter is ***I*** if it refers to the intermediate sectors, ***F*** to the final sectors, ***P*** to the primary sectors, ***U*** to the vector of the uses per industry within the framework of the entire economy (uses or resources);
- the third letter is ***P*** if the flows are for domestic production, ***M*** for imports, ***T*** for totals;
- the fourth letter is ***C*** for the flows from *EU* countries, ***A*** from non-*EU* countries, ***T*** for the total;
- the fifth and sixth letters are ***CF*** for *CIF* values, ***DU*** for *producers' price values (including invoiced VAT)*, ***PM*** for market price values, ***DD*** for customs duties, ***IV*** for VAT, ***MC*** for trade margins and ***MT*** for transport margins.

Thus, for example, the first row of the *script* of *Appendix A* reads*:* the total flow matrix for intermediate use at producers' prices (including invoiced VAT) is equal to the sum (element by element) of the domestic production flow matrix for intermediate use at producers' prices (including invoiced VAT) and of the import flow matrix for intermediate use at producers' prices (including invoiced VAT).

When reading the *script* the meaning of the *EULERO* commands and signs that precede the commands obviously have to be taken into account too.

Thus, in the first row the command *MM* common to the three aggregates indicates that equality is valid for each element of the matrices, so that the macro equation *1* actually defines *10.201* micro-equations, if there are *101* ESA industries: hence with the *EULERO* language even a complex system of equations can be written very simply and compactly and its structure changed very rapidly.

In the third row, however, for the row of elements of the domestic production flow matrix for intermediate use at producers' prices (including invoiced VAT) the sum is said to be equal to the column vector of industry production for intermediate use: macro-equation *3* therefore defines 101 micro-equations.

A fixed record format for the *script* was selected, so that each *sign–command–name of variable* combination must always occupy *12* characters, *2* for the sign, *3* for the command and 7 for the name of the variable (including the empty spaces).

6.3 *The convergence process.*

In experiments, the iterative procedure used converges very rapidly in a significantly much lower number of steps than the maximum limit of *k* (number of equations of the system). Convergence times can, however, be variable, for two main reasons; the first relates to the definition of the equation structure, and the second to the structure of the system of equations and variances.

As regards the first aspect, if the system is expressed with numerous implicit relations, then there is an increase in convergence time, which can even be significant, even if the definition of these relations turns out to be redundant from the formal point of view.

As for the second aspect, when increasing the "zero restrictions" of the system – that is, when increasing elements of the initial matrices of the accounting items that are not to be adjusted in the balancing process and which have been made to correspond to zero variance - care must be taken to avoid having a large number of elements of one or more equations of the system being restricted to zero where there are relatively high residuals. Clearly, the system does not converge when all the elements of one or more non-balanced equations – that is, with a non zero value in the vector of residuals $(G\hat{x} - h)$ – are all restricted to zero: accordingly, it is also necessary to bear in mind that each single element of the matrix of items is normally written into several equations, so that the effects of each "zero restriction" on the whole balancing structure, and not just on the single equations, must be considered.

Experimentally, no problems connected to the presence of a relatively high number of zeros in the matrix *V* were encountered, unless, of course, the case previously mentioned occurred.

The balancing procedure proved to be particularly efficient [10], and took fewer than *200* iterations and less than *7* minutes[11] to balance the system of equations. Moreover, the entire procedure, evidently optimised, requires little more than *8 Mbytes* of central memory.

Furthermore, from different simulations carried out on different structures, it emerged that the computational complexity of the proposed method, and hence the time taken for balancing, varies in slightly higher proportion to the square of the number of elements of the system, and is, therefore, at the lower variation *range* limit for this type of method.

7 Conclusions.

From experiments carried out it emerges that the block balancing procedure described, in combination with the *EULERO* meta-language, is a flexible method easy to implement.

Moreover, systems of large accounting equations can be balanced even on medium powered processing systems.

Improvements are currently being made from the implementational point of view, mainly by improving the efficiency and compactness of the *EULERO* commands interpreter.

However, from the point of view of defining the accounting system, the balancing system will be further expanded by making it possible to simultaneously balance accounting structures at current prices and constant prices; the method also needs to be generalized by introducing linear operators (sum of the matrices) as well as non-linear operators (product of the matrices).

---

10 The data refer to application on a PC.

[11] Actual CPU time

Methodological Appendix.

Let *u* be the unit vector *u'=[1,1,1,…,1]*, and ignoring the definitions of the sum matrix element by element of two matrices *A* and *B* and of two vectors *a* and *b* since they are obvious, define:

matrix sum element by element of a matrix *A* and a column vector *b* the matrix:

$$C = A + bu'$$

- matrix sum element by element of a matrix *A* and a row vector *b* the matrix:

$$C = A + u'b$$

- matrix sum element by element of a matrix *A* and a scalar *b*, the matrix:

$$C = A + b(uu')$$

- vector sum element by element of a vector *a* and a scalar *b* the vector:

$$c = a + bu$$

Appendix A: Configuration script of the system of equations for balancing market prices - producers' prices (including invoiced VAT) for the 1992 Italian I-O Table.

```
+ MM XIPTDU  + MM XIMTDU  - MM XITTDU
+ MM XFPTDU  + MM XFMTDU  - MM XFTTDU
+ SR XIPTDU  - VC RIPTDU
+ SR XIMTDU  - VC RIMTDU
+ SR XITTDU  - VC RITTDU
+ SR XFPTDU  - VC RFPTDU
+ SR XFMTDU  - VC RFMTDU
+ SR XFTTDU  - VC RFTTDU
+ SC XIPTDU  - VR CIPTDU
+ SC XIMTDU  - VR CIMTDU
+ SC XITTDU  - SC CITTDU
+ SC XFPTDU  - VR CFPTDU
+ SC XFMTDU  - VR CFMTDU
+ SC XFTTDU  - VR CFTTDU
+ VC RIPTDU  + VC RFPTDU  - VC RUPTDU
+ VC RIMTDU  + VC RFMTDU  - VC RUMTDU
+ VC RITTDU  + VC RFTTDU  - VC RUTTDU
+ SM RUTTDU  - MM TUTTDU
+ SM RUMTDU  - MM TUMTDU
+ SM RUPTDU  - MM TUPTDU
+ SR XPTTDU  - VC RPTTDU
+ SR CITTDU  + VC RPTTDU  + VC RUPTIV  + VC RUMTDU  - VC RUTTDU
+ VC RUPTIV  + VC RUMTIV  - VC RUTTIV
+ SM RUTTIV  - MM TUTTIV
+ SC XPTTDU  - VR CPTTDU
+ MM XITTDU  + MM XITTMC  + MM XITTMT  - MM XITTPM
+ MM XIPTDU  + MM XIPTMC  + MM XIPTMT  - MM XIPTPM
+ MM XIMTDU  + MM XIMTMC  + MM XIMTMT  - MM XIMTPM
+ MM XFTTDU  + MM XFTTMC  + MM XFTTMT  - MM XFTTPM
+ MM XFPTDU  + MM XFPTMC  + MM XFPTMT  - MM XFPTPM
+ MM XFMTDU  + MM XFMTMC  + MM XFMTMT  - MM XFMTPM
+ MM XIPTMC  + MM XIMTMC  - MM XITTMC
+ MM XFPTMC  + MM XFMTMC  - MM XFTTMC
+ MM XIPTMT  + MM XIMTMT  - MM XITTMT
+ MM XFPTMT  + MM XFMTMT  - MM XFTTMT
+ SR XITTMC  - VC RITTMC
+ SR XIPTMC  - VC RIPTMC
+ SR XIMTMC  - VC RIMTMC
+ SR XFTTMC  - VC RFTTMC
+ SR XFPTMC  - VC RFPTMC
+ SR XFMTMC  - VC RFMTMC
+ SC XITTMC  - VR CITTMC
+ SC XIPTMC  - VR CIPTMC
+ SC XIMTMC  + VR CIPTMC
+ SC XFTTMC  - VR CFTTMC
+ SC XFPTMC  - VR CFPTMC
+ SC XFMTMC  + VR CFPTMC
+ VC RITTMC  + VC RFTTMC  - VC RUTTMC
+ SR XITTMT  - VC RITTMT
+ SR XIPTMT  - VC RIPTMT
+ SR XIMTMT  - VC RIMTMT
+ SR XFTTMT  - VC RFTTMT
+ SR XFPTMT  - VC RFPTMT
+ SR XFMTMT  - VC RFMTMT
+ SC XITTMT  - VR CITTMT
+ SC XIPTMT  - VR CIPTMT
+ SC XIMTMT  + VR CIPTMT
+ SC XFTTMT  - VR CFTTMT
+ SC XFPTMT  - VR CFPTMT
+ SC XFMTMT  + VR CFPTMT
+ VC RITTMT  + VC RFTTMT  - VC RUTTMT
+ MM XIMTPM  + MM XIPTPM  - MM XITTPM
+ MM XFPTPM  + MM XFMTPM  - MM XFTTPM
+ SR XITTPM  - VC RITTPM
+ SR XIPTPM  - VC RIPTPM
```

```
+ SR XIMTPM  - VC RIMTPM
+ SR XFTTPM  - VC RFTTPM
+ SR XFPTPM  - VC RFPTPM
+ SR XFMTPM  - VC RFMTPM
+ SC XITTPM  - SC CITTDU
+ SC XIPTPM  - VR CIPTPM
+ SC XIMTPM  - VR CIMTPM
+ SC XFTTPM  - VR CFTTDU
+ SC XFPTPM  - VR CFPTPM
+ SC XFMTPM  - VR CFMTPM
+ VC RITTPM  + VC RFTTPM  - VC RUTTPM
+ VC RIMTPM  + VC RFMTPM  - VC RUMTPM
+ VC RIPTPM  + VC RFPTPM  - VC RUPTPM
+ SM RUTTPM  - MM TUTTDU
+ SM RUMTPM  - MM TUMTPM
+ SM RUPTPM  - MM TUPTPM
+ MM CITTDU  - MM CITTDU
+ MM XPTTDU  - MM XPTTDU
+ SR CITTDU  + SR XPTTDU  + VC RUPTIV  + VC RUMTDU  + VC RUTTMC  + VC RUTTMT  - VC RUTTPM
```

References.

BECKMAN F.S. *- The solution of linear equations by the conjugate gradient method, in «Mathematical Methods for Digital Computer», A. Ralston and H. S. Wilf eds., pp. 62 72, New York: Wiley, 1960.*

BYRON R. P*. - Estimation and inference for large econometric systems, in «Econometrica», 45, pp. 1499 1516, 1977.*

BYRON R. P. *- The Estimation of Large Social Account Matrices, in «Journal of the Royal Statistical Society», 141, Part 3, pp. 359 367, 1978.*

COMINCIOLI V. *– Analisi numerica. Metodi, modelli, applicazioni, McGraw-Hill, Milano, 1995.*

NICOLARDI V. *- Lo studio e la proiezione delle strutture demografiche a fini economici: una proposta di integrazione in schemi SAM, in "Scritti di Statistica Economica - A cura di Claudio Quintano", Rocco Curto Editore, Napoli, 1994.*

NICOLARDI V. *- Lo studio e la ricostruzione dei flussi demografici tra due rilevazioni censuarie: una applicazione ai dati 1981-1991, ricerca svolta nell'ambito della stesura di "Lavoro e Politiche della Occupazione in Italia - Rapporto '93-'94", Ministero del Lavoro e della previdenza Sociale - Fondazione Giacomo Brodolini, Istituto Poligrafico e Zecca dello Stato, Roma, 1994.*

NICOLARDI V. *- Un sistema di bilanciamento per matrici contabili di grandi dimensioni, in "Quaderni di ricerca" n.4, ISTAT, Roma, 1998.*

NICOLARDI V. *- Una metodologia integrata per lo studio e la proiezione delle strutture demografiche, in «Economia & Lavoro», anno XXVII, n. 4, Ott. Dic. 1993, Marsilio editori Venezia.*

STONE R. *– Direct and indirect constraints in the adjustment of observation, Mimeographed. D.A.E. Cambridge, 1975.*

STONE R. *- Input Output and National Accounts, Paris O.E.E.C., 1961*

STONE R. *– Input-output projection: consistent prices and quantity structures, in «L'industria», No. 2, 212-224, 1968.*

STONE R. *- Mathematical Models of the Economy and Other Essays, London, Chapman and Hall eds., 1970.*

STONE R. *- Social accounts at the regional level: a survey, in «Regional Economic Planning, Techniques of Analysis», W. Isard and J. H. Cumberland eds., pp. 263 296, Paris: O.E.E.C., 1961*

STONE R. *- The development of economic data systems, in «Social accounting for development Planning with Special Reference to Sri Lanka, G Pyatt and al. eds., Cambridge: Cambridge University Press, 1976.*

STONE R., MEADE J., CHAMPERNOWNE D.C. *- The precision of national income estimates, in «Review of Economic Studies», 9 [2], 111 125, 1942.*

THEIL H. *- Economic Forecasts and Policy, Amsterdam: North Holland, 1961.*

VAN DER PLOEG F. *– Reliability and the Adjustment of Sequences of Large Economic Accounting Matrices, in «Journal of the Royal Statistical Society», 145, Part 2, pp. 169-194, 1982.*