A NEW SIMULTANEOUS APPROACH TO THE STEADY-STATE SIMULATION OF COMPLEX TECHNOLOGICAL SYSTEMS

S.RAGHEB TEWFIK* and J.VRBA

Institute of Chemical Process Fundamentals, 165 02 Prague 6 - Suchdol

Received February 10th, 1977

A new simultaneous method, called the Simultaneous General Simulation Technique, has been proposed. Two algorithms have been developed: the Direct Balace Algorithm and the Iterative Balance Algorithm and a computer program has been prepared and tested to the solution of typical systems. The method may be recommended to the solution of simulation problems for both generalised flowsheeting programs and special purpose programs at both performance and design modes.

In recent years significant progress has been made in every area of analysis and treatment of complex systems as a whole. A fundamental step in the analysis of complex systems is simulation which is defined as the solution of the mathematical model representing the interconnected system. Several techniques of simulation are encountered in both literature and practice such as process[1], dynamic[2], stochastic[3], discrete and balancing of overdetermined systems[4] but the simulation technique discussed in this article is meant to be the continuous process simulation at steady-state.

Simulation of complex systems has become of widespread interest since the end of the 1950's. Considerable attention has been directed towards the development of general-purpose process simulation programs which calculate the heat and material balances. The choice of an efficient method for performing the simulation calculations can significantly reduce the solution time for the design problem considered. There are essentially two approaches for solution: sequential and simultaneous. Though both approaches are valid each has its advantages and drawbacks. In this article a new simultaneous method for performing the simulation calculations called the Simultaneous General Simulation Technique is proposed. The method surmounts the common difficulties encountered by other simultaneous methods which are mainly the requirements of a new formulation and a new computer program for every specific problem. In addition, it can be incorporated in both generalised and special purpose programs.

* Present address: National Research Centre, Pilot Plant Units, Dokki, Cairo, Egypt.

The Simultaneous General Simulation Technique SGST

On analysing the mathematical models of typical complex chemical technological systems it may be noticed that about 80% of the total number of equations expressing the model are balancing relations and are of the linear form. Motivated by this fact the presented method has been developed. The linear equations may be eliminated to simplify the system considerably thus providing a reduced set of nonlinear equations that can be solved simultaneously.

According to this basic idea two algorithms have been developed: *1.* A Direct Balance Algorithm which is applicable at a preliminary design stage when the simulation problem is that of solving a large set of equations simultaneously. *2.* An Iterative Balance Algorithm that solves model comprising units that require iterative technique for their solution (*e.g.* distillation column) and is applicable at both design and processing stages.

The Direct Balance Algorithm

The model of the complex system may be represented by

$$F_i(x_1, x_2, \ldots, x_n) = 0 \; ; \quad i = 1, 2, \ldots, n \, . \tag{1}$$

These equations are material and energy balance relationships, *PVT* data, equilibrium relationships *etc.* while the (x_j) represent stream flows, mole fractions, temperatures, pressures or equipment parameters.

Of these n equations, m are linear and may be represented by the matrix equation

$$\bar{A}\bar{X} = \bar{C} \, , \tag{2}$$

where $\bar{A}$ is a rectangular matrix (m, n) in which rows are the linear equations (m) and columns are the variable (n), $\bar{X}$ is the column (n) of variables x_j, $\bar{C}$ is the column (m) of contants of the linear equations.

This linear set can be used to express explicitly some of the variables (m) (which are to be called 'eliminated') in terms of the other variables $(n - m)$ (which are called 'remaining'). The choice of the eliminated variables is not arbitrary but is has to be ensured that each of these variables is assigned to only one of the linear equations and that each equation has a single output variable. An algorithm has been developed for this purpose[7,8].

Once the eliminated variables have been chosen and assigned to the linear equations the next step is to explicitly express these eliminated variables linearly in terms of the remaining ones.

The rectangular matrix $\bar{A}(m, n)$ of equation (2) is split into $\bar{A}1(m, n)$: a square matrix with equations as rows and the eliminated variables as columns, $\bar{A}2(m, n - m)$: a rectangular matrix with equations as rows and the remaining variables as columns. Similarly $\bar{X}(n)$ of equation (2) is split into $\bar{X}1(m)$: the vector of eliminated variables, $\bar{X}2(n - m)$: the vector of remaining variables.

Thus equation (2) may be rewritten as

$$\bar{A}1 \,.\, \bar{X}1 + \bar{A}2 \,.\, \bar{X}2 = \bar{C}\,, \tag{3}$$

then

$$\bar{X}1 = \bar{A}1^{-1}(\bar{C} - \bar{A}2 \,.\, \bar{X}2)\,. \tag{4}$$

Gauss–Jordan method is used for the inversion of matrix $\bar{A}1$. Relations (4) express the explicit dependence of the eliminated variables on the remaining ones. These relations may be derived analytically but, being an intermediate step of the algorithm programmed for digital computers, they are formed numerically.

On substituting the eliminated variables $\bar{X}1(m)$ in terms of the remaining ones $\bar{X}2(n - m)$ in the nonlinear part of equations (4) a reduced fully determined set of nonlinear equations is obtained

$$F_i'(X2(n - m)) = 0 \quad i = 1, 2, \ldots, n - m\,. \tag{5}$$

This reduced set may be solved by any suitable method for the solution of a set of nonlinear equations.

The final step of the algorithm is to substitute the values of the remaining variables calculated from the reduced nonlinear set in the linear dependency relations to recover the values of the eliminated variables. Thus the values of all the variables of the system have been calculated and the simulation problem is solved.

The Iterative Balance Algorithm

The Direct Balance Algorithm is suitable for the simulation of system when the problem is the solution of a set equations simultaneously. The situation may be complicated by the presence of separation operations (distillation columns, absorbers, *etc.*) as such units require iterative models for their solution. To overcome this difficulty an Iterative Balance Algorithm which is an extension of the Direct Balance Algorithm to handle iterative models, has been developed. For illustrative purposes the occurrence of a multicomponent distillation column in the complex is discussed. For such a unit two models are developed: a rigorous iterative model and a simple representative model in which the variables of the unit are related in terms of additional coefficients which play a role similar to the split fractions of Nagiev's Technique[5,6]. This representative model may be of the form

$$d_i = F(f_i, b_i, c_i) \quad i = 1, 2, \ldots, k \tag{6}$$

$$b_i = f_i - d_i \qquad i = 1, 2, \ldots, k\,, \tag{7}$$

where b_i – the bottoms rate for component i, d_i – the distillate rate, f_i – the feed rate, c_i – the additional coefficient.

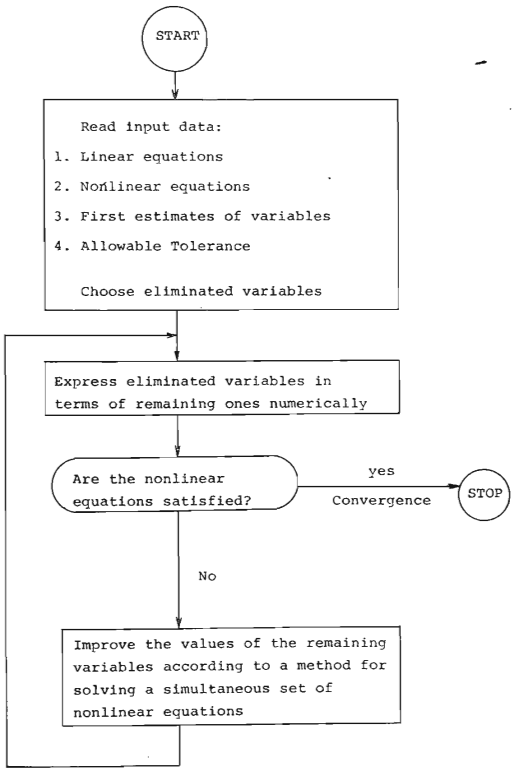

FIG. 1
Logic Diagram of Program SGST

The algorithm proceeds as follows: *a*) The set of equations expressing the representative model together with the other equations of the complex system constitute a simulation problem that can be solved by the Direct Balance Algorithm after providing first estimates for the additional coefficients. *b*) When the Direct Balance Algorithm is completed, values are obtained for all the variables of the system including the inputs and outputs of the distillation column (f_i, d_i, b_i), *c*) The calculated feed rates and any other necessary specifications such as distillate rate, rate of a specific component *etc.*, are used to solve the distillation column by the iterative rigorous model and new values are obtained for the outputs of the column (d_i's and b_i's). *d*) These values may be substituted in equations (*6*) to calculate new estimates for the additional coefficients. *e*) Steps *a*) to *d*) are repeated iteratively except that on solving the simulation problem by the Direct Balance Algorithm (step *a*)) the treatment of the linear set is performed only in the first iteration white for succeeding iterations only the solution of the reduced nonlinear set is considered. *f*) The algorithm is terminated when the relative changes of the additional coefficients between two successive iterations is smaller than a specified tolerance criterion.

The Iterative Balance Algorithm has been demonstrated by considering a complex system in which only one distillation column is present but the same logic is applicable to systems containing any number of auch separating units.

According to this technique an overall computer program (SGST) which performs all steps of the algorithm automatically has been developed. This program when provided with the set of equations modelling the complex system and first estimates of the variables computes the values of all the variables of the system. A logic diagram of the program in shown in Fig. 1.

The method is ilustrated by solving the simulation problem of a hypothetical complex system. Two versions of the system are considered: *a*) System I which is a simplified form of the system (77 equations and variables, 3 recycles). This system is simulated by the Direct Balance algorithm. *b*) System II which is the complete process (83 equations and variables, 4 recycles), and is simulated by the Iterative Balance Algorithm. The computer program (CGST) has been used to simulate the system and the computation has been carried out on IBM 370/135.

For comparison these systems have been simulated by Nagiev's technique[5,6] which is the most popular and most widely used simultaneous method. The convergence characteristic are shown in the following Tables I and II.

The advantages of the Simultaneous General Simultation Technique are apparent from these results and may be sumarised as follows: *1*. High accuracy and low computer time. *2*. The method does not require the reformulation of the problem but deals with the system' s modelin its initial form. *3*. Generality and applicability to both general purpose and special purpose programs at both performance and design modes.

TABLE I

Convergence Characteristics of System I

Characteristics	The Direct balance algorithm	Nagiev's technique		
		allowable tolerance criterion		
		0·0005	0·0001	0·00001
Sum of squares of discrepancies of equations	0·2020 + 00	0·4185 + 02	0·1681 + 02	0·1510 + 01
No of trial calculations	19	15	17	141
CPU time	2′42″	1′32″	1′42″	4′19″

TABLE II

Convergence Characteristics of System II

Characteristics	The iterative algorithm	Nagiev's technique	
		allowable 0·0005	tolerance criterion 0·0001
Sum of squares of discrepancies of equations	0·1448−01	0·2558 + 04	0·9221 + 02
No of trial calculations	35	100	246
CPU time	6′54″	3′43″	9′19″

REFERENCES

1. Crowe C. M., Hamielec A. E., Hoffman T. W., Johnson A. I., Woods D. R., Shannon P. T.: *Chemical Plant Simulation*. Prentice-Hall, New York 1971.
2. Perry R. H., Chilton C. H.: *Chemical Engineers Handbook*, 5th Ed. Mc Graw-Hill, New York 1973.
3. Gaddy J. L., Culberson O. L.: AIChE J. *19*, 1239 (1973).
4. Václavek V.: Chem. Eng. Sci. *24*, 947 (1969).
5. Nagiev M. F.: Chem. Eng. Progr. *53*, 297 (1957).
6. Rosen E. M.: Chem. Eng. Progr. *58*, 69 (1962).
7. Vrba J.: Chem. Prům. *21*, 85 (1971).
8. Ragheb Tewfik S.: *Thesis*. Czechoslovak Academy of Sciences, Prague 1976.