A STUDY OF THE PROMOTION OF CONVERGENCE IN THE CALCULATION OF COMPLEX CHEMICAL PLANTS

A STUDY OF THE PROMOTION OF CONVERGENCE IN THE CALCULATION OF COMPLEX CHEMICAL PLANTS

BY

ODED ORBACH

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

September 1969

DOCTOR OF PHILOSOPHY (1969)
(Chemical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario.

TITLE : A Study of the Promotion of Convergence in the Calculation of Complex Chemical Plants.

AUTHOR : Oded Orbach

B.Sc. (Technion, Israel Institute of Technology)
M.Sc. (Technion, Israel Institute of Technology)

SUPERVISOR : Professor C.M. Crowe

NUMBER OF PAGES : x, 191

SCOPE AND CONTENTS:

Techniques for accelerating the convergence of multi-variable iterative calculations were studied. The study is mainly concerned with iterations which result from the calculation of recycle processes of complex chemical plants.

The conditions for convergence and the existence of the solution of linear and non-linear iterations were obtained. Special attention was given to the asymptotic behaviour of the iterations since, in most cases, they approach geometric progression.

The convergence of an Alkylation unit simulation was studied by implementation of the "full matrix" technique for accelerating the rate of convergence. The geometric extrapolation technique was described and applied successfully for accelerating the convergence of the calculation of multi-component distillation columns.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my research director Dr. C.M. Crowe for his guidance and assistance in carrying out this study.

Special thanks are also due to Dr. A.I. Johnson for supervising me at the early stages of this study and above all for introducing me to the field of simulation.

I am indebted to Dr. W.H. Fleming and Dr. D.J. Kenworthy of the Applied Mathematics Department for their advice and assistance, and to D.R. Prowse for providing the information related to the distillation column case study.

I am grateful to McMaster University for financial support.

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

Page

I. INTRODUCTION

The computational complexity of chemical processes has been reduced by the development of generalized process simulators for use on a digital computer. Excellent reviews describing the process simulators available have been published lately (C4, E1, J1, L1, R1). One of these executive programs is Shannon's PACER (C4, M2, S2) which has been widely utilized at McMaster University for simulation of chemical plants such as a sulphuric acid plant, an alkylation process unit, and the Bayer process for alumina extraction.

The underlying principle of most of these simulator programs is that of modularity. A real chemical plant is made up of processing units such as distillation columns, reactors, heat exchangers, pumps, compressors, and so on. Each unit (equipment) has a set of fundamental laws of chemistry or physics which can be used to model the unit mathematically or by empirical equations. The assembly of such a set of mathematical equations and of the mathematical techniques for solving them, so that the unit is simulated, is defined as a unit computation. Each unit computation computes all the output streams from the unit, given all the input streams and the relevant parameters which describe the unit.

The process simulator controls and directs the calculation of the unit computation of the chemical plant. Thus, the unit computations, together with the information from the process flow diagram, can be utilized by the simulator to set up a mathematical model of the complete process.

The essence of a simulator which uses the modular approach is that it provides an easy means of describing the entire process itself, and the relevant data and specifications, so that modifications can be readily made and the calculations carried out without the need for any special programming by the user, other than of the necessary unit computations. The simulation of the complete process can be used as many times as required to evaluate the performance, either for different input conditions or for different values of the design or operating parameters.

If the process flow diagram contains no recycle streams, it is possible to work systematically through the process, unit by unit, using computed output streams as the inputs to succeeding units. This is not possible if a recycle stream is present, since this will always be required as an input before it has been computed as an output. All the simulator programs deal with this situation by breaking a sufficient number of streams, called assumed or recycle streams, by treating the downstream half of each as an input to the process and the other half as an output, and by using a suitable iterative procedure to match the two halves. The user must provide the initial estimates for these "inputs", although it has been claimed by Kesler and Griffiths (KI) that in some cases convergence is assured from an initial estimate of zero. Typical recycle process is given in Figure 2.1.

Determining the sequence in which the units are computed, the computational sequence, and hence the identification of the recycle streams, is the first step essential in the calculation of recycle processes. Extensive research has been done (C2, L2, S3, F2, N1) to determine the minimum number of recycle streams (assumed streams), by analyzing the topological description of the flowsheet, although the computational sequence which is obtained from these streams will not necessarily yield the most rapid convergence.

The method of successive substitutions is probably the most commonly used (E1) for computing the recycle processes. The major advantage of this method is its simplicity, but unfortunately its rate of convergence can be intolerably slow and sometimes it may not converge at all.

For this reason, various kinds of convergence accelerators are frequently used in process simulation work (C3, K1, K2, R3, R5). Cavett (C3) presented a comprehensive review of techniques for obtaining solutions for steady state process simulations. In particular, Cavett discussed the application of convergence acceleration such as Wegstein's method (W1) and Newton's method. Naphtali (N3), Ravicz and Norman (R5) have also discussed the application of Newton's method for obtaining the steady state solution of recycle processes.

Nagiev, (N2) who has not applied the modular approach to solve recycle problems, formulated the simulation problem as a system of linear equations. Linearization was achieved by linearly characterizing the operation of the processing units in terms of split fractions and the solution was obtained simultaneously by inverting the matrix of constant coefficients. Rosen (R2) extended Nagiev's formulation technique to allow for varying of the split fraction. Rosen used this method for promoting the convergence of heat and material balance calculation of recycle processes.

Despite extensive studies and wide attention given to the convergence problem in the past, the need still exists to develop more efficient convergence promotion techniques for large and complex chemical processes. The recognition of this need initiated this study.

The objective was to evaluate methods for promoting the convergence of complex chemical plants and thus, for reducing the number of iterations and overall computation time.

The formulation of the iterative calculation technique for recycle process is discussed in Section 2. In addition, the existence of the solution and the convergence condition of linear and non-linear iterations are studied. The asymptotic behaviour of the iteration is thoroughly examined.

In Section 3, the convergence promotion methods are discussed. The full matrix method and the geometric extrapolation technique are demonstrated in two case studies (Section 5). Section 4 brings out the residual method of formulation of recycle processes. A few solution techniques for solving sets of non-linear algebraic equations are discussed such as, Newton's method, a modified form of Newton's method and the Quasi-Newton method. Two methods for solving multi-component distillation columns are studied in Section 6.

2. STEADY-STATE PROCESS SIMULATION

The widespread use of computers for simulation and design of large chemical processes has led to the development of executive computer programs (C4, E1, L1). These master programs coordinate the computation of the unit process modules and perform simulation or optimal design of complex chemical plants. The computational complexity of chemical processes has been reduced by the development of these generalized process simulators.

The successive substitution iteration is the most common technique to solve recycle problems. The formulation and conditions for convergence of the iteration will be discussed in this section. In addition, the asymptotical behaviour of the iteration will be studied.

2.1 Formulation

In order to reduce the total computation time, the first step in calculation of complex recycle processes is to partition the plant equipment into blocks, such that all the recycles are within the blocks, and then to order the blocks, such that completion of the computation in one block assures that all the streams, needed to perform the computation of the next block in order, would be known. The block contains one or more equipments (nodes) where the connections between the nodes are such that it is possible to reach from any node all the others in the same block, by a finite number of steps in the direction of the arrows. A block, which can be represented by direct graph, sometimes is called a maximal cyclical net (C1).

Example of partitioning recycle process into two blocks is given in Figure 2.3. Nodes (equipments) 1 and 2 are in the first block and nodes 3, 4, and 5 are in the second block. The computation in the second block would be made only after completing the computation of the first one.

Norman (N1) and Himmelblau (H1) detect blocks by taking successively higher powers of the "Association Matrix" of the process. Sargent and Westerberg (S3) performed partition directly with a list-type representation of the "flow diagram" of the process. A similar technique was also used by Christensen (C2).

Any block that contains more than one equipment (node), has one or more recycles and thus may be solved iteratively. Before starting the sequential calculation within a block having recycle, a set of streams must be chosen, the values of their variables assumed and the sequence of calculation of the equipment implied by this set of assumed streams should be found.

2.1.1 Assumed Streams and Sequence of Calculation

Norman (N1), Himmelblau (H1) and Chartrand (C1) used the "Association Matrix" for identification of the cycles within the blocks. Lee and Rudd (L2) start from this point, assuming that some method is available to trace out all the recycles, and use the "cycle matrix" to find minimum number of assumed streams. In PACER by testing all the feasible stream combinations and increasing successively the number of assumed streams, the minimum number of assumed streams can be found. But the number of combinations to be examined is excessive when the block is of moderate size, or when the number of streams that need to be assumed, for solving this block, is bigger than three.

Sargent and Westerberg (S3) reduce the number of combinations by merging edges and nodes (streams and equipments). Christensen (C2) reduced the combinatorial problem by eliminating "ineligible" edges (streams).

In all these works (C2, L2, S2, F2), assumptions were made that the sequence of calculation which results from a minimum set of assumed streams is that sequence of calculation of the equipment within the blocks which would minimize the calculation time. However, there is neither a numerical nor a theoretical proof of this assumption. Truly, Sargent and Westerberg (S3) were concerned about the problem that the minimum number of assumed streams might not lead to minimum over-all computation time.

Furthermore, it can be shown that the same sequence can result from another set of assumed streams, larger than or equal to the minimum set, by simply starting the calculation from another equipment in the sequence, and therefore the rate of convergence will be the same.

To explain the statement above we may examine a case that was given by Lee and Rudd (L2) (See Figure 2.2). For this recycle flow-sheet they have given four available sets of assumed streams which are sufficient to render the recycle problem acyclic (Table 2.1).

Although case B has the minimum number of assumed streams and was selected as the best by Lee and Rudd (L2), all four cases give the same computational sequence, differing only in the starting point.

TABLE 2.1

Assumed Streams and Sequence of Calculation for Recycle Problem as given by Lee & Rudd (L2) (Figure 2.2)

Case	Assumed Streams	Sequence of Calculation
A	S9, S10, S11	1, 2, 3, 4, 5, 6, 7
B	S3, S7	4, 5, 6, 7, 1, 2, 3
C	S4, S8, S9	5, 6, 7, 1, 2, 3, 4
D	S6, S8, S9, S10	7, 1, 2, 3, 4, 5, 6

Therefore, in order to get the best computational sequence it is not necessary to locate the minimum set of assumed streams. Furthermore, it appears that it is not always sufficient to locate the minimum set of assumed streams, in order to get the best computational sequence, since we may have more than one minimum set.

However, for solving recycle problems some set of streams must be assumed and this set ought to be feasible in order to get some suitable computational sequence, where a feasible set is defined as follows:

"A set of assumed streams of size m is a feasible set if and only if this set is sufficient to render the recycle problem acyclic but no proper sub-set of this set is sufficient to do so"

2.1.2 Calculation Techniques

By breaking each assumed stream and treating one half as an "input" to process and other half as an "output" a suitable iterative procedure may be used for matching the two halves.

If the assumed streams and their related sequence of calculation have been correctly chosen, the "output" half of the assumed streams depends only on the "input" half (if all the design and operating variables of the process remain constant). These relations can be represented by the following set of functions:

$$y_i = F_i\ (x_1,\ x_2,\ \ldots\ x_m) \qquad i = 1,\ 2,\ \ldots\ m \tag{2.1}$$

or in vector form:

$$\bar{Y} = \bar{F}\ (\bar{X}) \tag{2.2}$$

where:

$\bar{X}$ - is the "input" vector

$\bar{Y}$ - is the "output" vector

$\bar{F}$ - is the function vector

m - is the number of elements in vectors $\bar{X}$, $\bar{Y}$ and $\bar{F}$.

Vectors $\bar{X}$ and $\bar{Y}$ represent the same set of assumed streams and the elements of these vectors contain all the componentsof these streams.

Therefore, the number of elements, m, in vectors $\bar{X}$ or $\bar{Y}$ are equal to the number of assumed streams multiplied by the number of components in those streams.

The aim of any calculation technique is to find vector $\bar{X}$ such that $\bar{X} = \bar{Y}$ or to satisfy the following equation

$$\bar{X} = \bar{F}(\bar{X}) \qquad (2.3)$$

Here, $\bar{X}$ is the variable vector and $\bar{X}_s$ is the solution of the recycle problem which satisfies equation (2.3), such that

$$\bar{X}_s = \bar{F}(\bar{X}_s) \qquad (2.4)$$

After evaluating $\bar{X}_s$, that is the values in the assumed streams, the remaining streams of the recycle process may be calculated directly.

Most of the executive programs as PACER (S2) use successive substitution for solving recycle problems. This technique may be described by the following two equations:

$$\bar{Y}_n = \bar{F}(\bar{X}_n) \qquad (2.5)$$

$$\bar{X}_{n+1} = \bar{Y}_n \qquad (2.6)$$

$$n = 0, 1, 2, 3, \ldots$$

where n is the number of the iteration and vector $\bar{X}_o$ is the initial point.

If the initial value $\bar{X}_o$ of the assumed streams is close enough to the solution $\bar{X}_s$ the iterative procedure (Equations (2.5) and (2.6)) hopefully will converge within a specified tolerance after a sufficient number of iterations.

Kesler and Griffiths (K1) have also applied this successive substitution method and they compared their results with the method of simultaneous solution described by Nagiev (N2) and Rosen (R2). The initial composition of the assumed streams $\bar{X}_o$ is assumed to be zero in their study.

Cavett (C3) and Kliesch (K2) discussed the application of convergence promotion routines to replace the successive substitution (Equation 2.6). Cavett (C3) presented a comprehensive review of iteration techniques used to obtain solutions of recycle processes. Direct iteration (successive substitution), Wegstein's method, hyperbolic extrapolation and Newton Raphson were reviewed and used to solve example problems. Cavett found the hyperbolic method to be very effective and useful for accelerating the convergence of recycle processes containing only one recycle stream.

Kliesch (K2) claimed success in accelerating the convergence using the bounded Wegstein (W1) method. Wegstein (W1) used his method, which is in fact a modification of Aitken's (A3, A4) method, for accelerating the convergence of mono-variable iterative processes. Hence, it is expected that this method, as Cavett's hyperbolic extrapolation method (C3), will be useful only for promoting the convergence of a mono-recycle process where the interaction between the components are weak.

The sum of the equations (2.5) and (2.6) gives the general iterative equation

$$\bar{X}_{n+1} = \bar{F}(\bar{X}_n) \qquad (2.7)$$

$$n = 0, 1, 2, 3 \ldots$$

This equation (2.7) is the basic iterative equation that describes calculation of recycle processes or calculation of heat and material balances which are solved iteratively.

In case of recycle processes the functions $F_i(\bar{X})$ represent in fact the numerical procedure which is carried out as the calculation progresses from unit to unit. It is possible to find the value of the set of functions for any point $\bar{X}$ but the first derivative can be taken only numerically (by a finite difference).

Although it was said that it is possible to evaluate $\bar{F}$ for any point $\bar{X}$ it should be noted that in practice only when $\bar{X}$ is located within a desired distance from the solution can the set of functions be evaluated. It mainly happens if some modules within the recycle process have a constraint on their feed composition.

Now, if the solution $\bar{X}_s$ exists in the desired interval and satisfies equation (2.4) and the initial point $\bar{X}_o$ is close enough to the solution, for material balance calculations the iteration (2.7) usually converges. Non-convergent iteration sometimes appears in the calculation of highly interactive heat and material balances. The problem of how to obtain a good initial point $\bar{X}_o$ that will be close enough to solution $\bar{X}_s$ is unresolved in general. Frequently, a good estimate of the solution is known to the problem formulator, i.e. the process engineer.

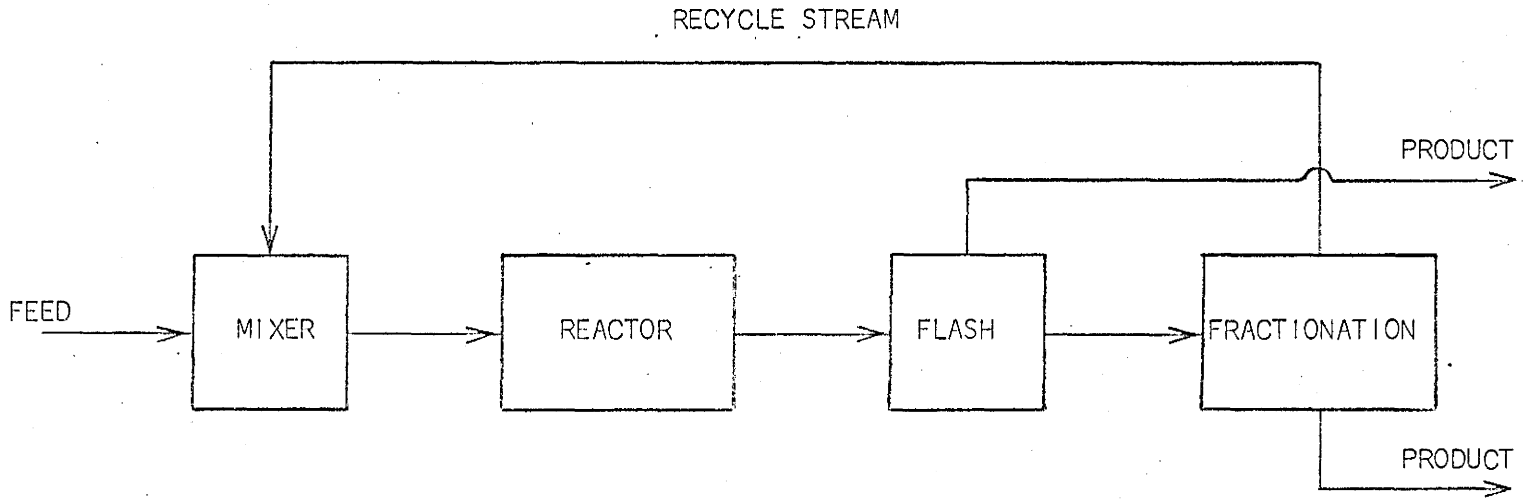

FIGURE 2.1 TYPICAL RECYCLE PROCESS

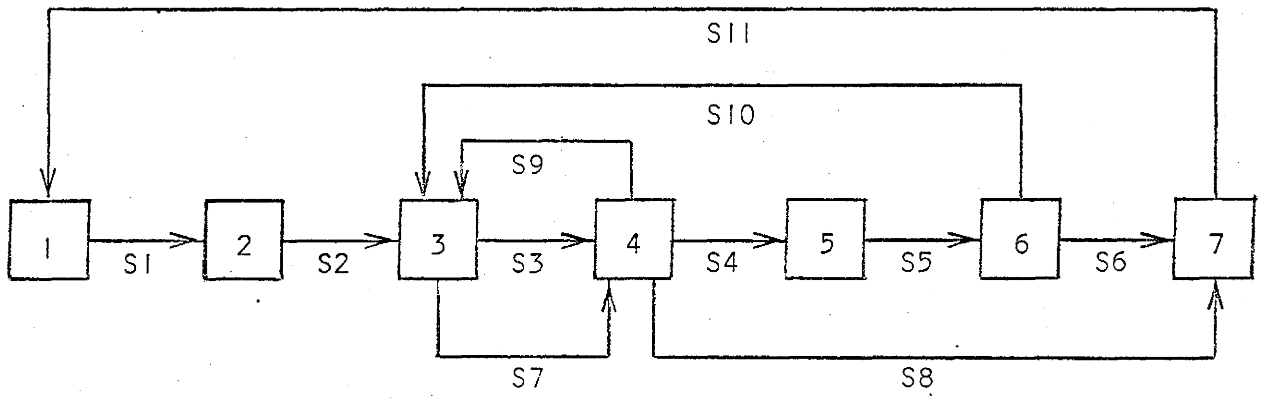

FIGURE 2.2 THE EXAMPLE OF LEE & RUDD (L2)

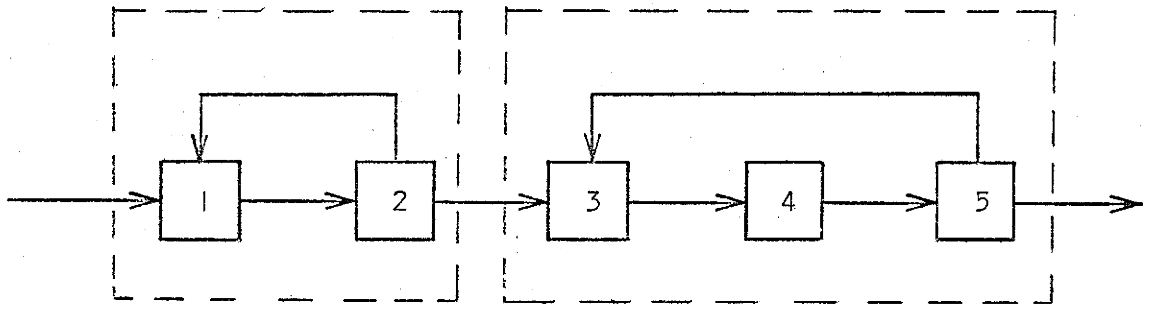

FIGURE 2.3 FLOW DIAGRAM OF RECYCLE PROCESS

2.1.3 Recycle of Heat

In many chemical processes the output flows from a unit exchange heat with the input flows, which constitutes a recycle of heat. Considering the case which is given in Figure 2.4, E1 is a heat exchanger and E2, which could represent an evaporator, distillation column etc., is a computational unit which consists of heat and material balance calculations.

It can easily be shown that it is preferable to select stream number 2 as the assumed stream instead of stream number 4 (See Figure 2.4). If stream number 2 is selected then the temperature of this stream becomes the only variable of the recycle calculation. All the components which flow in stream 2 are identical to those in stream 1, the feed stream. If stream number 4 is chosen as assumed stream, the variables of the recycle calculation are the temperature and all the components of stream 4.

Hence the computation of recycle of heat can be reduced to a one dimensional problem which can be solved by successive substitution or any other one dimensional search technique where the variable is the temperature of stream 2.

If the recycle of heat (Figure 2.4) is part of another main recycle, such that streams 1 and 5 are included in it, it may be better to converge the recycle of heat separately, every time the computational sequence reaches this section of the flowsheet.

Sometimes, the temperature of stream number 2 is controlled (See Figure 2.5). In this case, the computation of the recycle of heat can be solved directly. The calculation procedure can be as follows:

1. Transfer the components flow from stream 1 to stream 2 and fix the temperature of stream 2 as desired.
2. Calculate equipment E2, E1, and E3 in sequence.

Extracting all the recycles of heat in the simulated processes and treating them in the way discussed above can reduce the number of recycles and simplify the calculation. An example is given in Section 5.1.

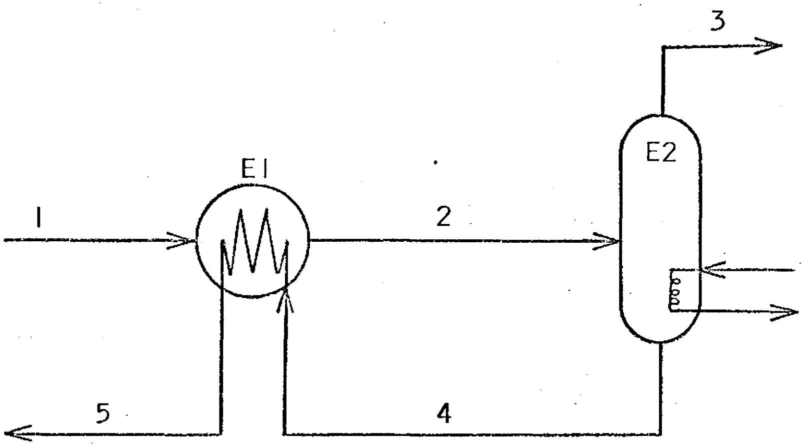

FIGURE 2.4 RECYCLE OF HEAT

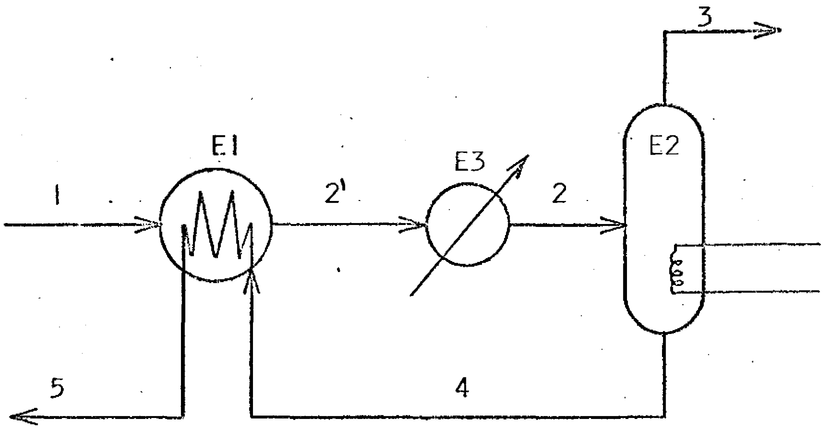

FIGURE 2.5 RECYCLE OF HEAT
WITH TEMPERATURE CONTROL

2.2 Iteration and Fixed Point

Considering again the general equation of iteration (2.7),

$$\bar{X}_{n+1} = \bar{F}(\bar{X}_n) \qquad (2.8)$$

$$n = 0, 1, 2, \ldots$$

Isaacson and Keller (11) showed that for any set of functions $\bar{F}$ and some initial approximation $\bar{X}_o$, the convergence of this iteration process is assured if the mapping $\bar{F}(\bar{X})$ carries a closed and bounded set $S \subset R^m$ into itself and if the mapping is contracting, i.e., if

$$||\bar{F}(\bar{X}) - \bar{F}(\bar{Y})|| \leq M \, || \bar{X} - \bar{Y}|| \qquad (2.9)$$

for some norm, the "Lipschitz" constant $M < 1$ and all $\bar{X}$, $\bar{Y} \varepsilon S$. It can be shown under these conditions that $\bar{F}(\bar{X})$ has a unique fixed point $\bar{X}_s$ in S satisfying equation (2.8), as

$$\bar{X}_s = \bar{F}(\bar{X}_s)$$

The vector norm may be chosen as any one of the following

$$||\bar{X}||_\infty = \max_i |X_i| \qquad (2.10)$$

$$||\bar{X}||_1 = \sum_{i=1}^{m} |X_i| \qquad (2.11)$$

$$||\bar{X}||_2 = \{ \sum_{i=1}^{m} X_i^2 \}^{\frac{1}{2}} \qquad (2.12)$$

2.2.1 Convergence Conditions

The following two theorems stated by Isaacson and Keller (II) give sufficient conditions for the existence of a solution $\bar{X}_s$ and for convergence.

Theorem I

Let $\bar{F}(\bar{X})$ satisfy

$$||\bar{F}(\bar{X}) - \bar{F}(\bar{Y})|| \leq M\,||\bar{X}-\bar{Y}|| \tag{2.13}$$

for all vectors, $\bar{X}$, $\bar{Y}$ such that

$$||\bar{X} - \bar{X}_o|| \leq r$$

and

$$||\bar{Y} - \bar{X}_o|| \leq r$$

with the Lipschitz constant, M, satisfying

$$0 \leq M < 1 \tag{2.14}$$

and let the initial, $\bar{X}_o$, satisfy

$$||\bar{F}(\bar{X}_o) - \bar{X}_o|| \leq (1-M)r \tag{2.15}$$

Then, i) all iterates X_n (2.8) satisfy

$$||\bar{X}_n - \bar{X}_o|| \leq r$$

ii) the iterates converge to some vector say,

$$\lim_{n\to\infty} \bar{X}_n \to \bar{X}_s$$

iii) $\bar{X}_s$ is the only root of (2.8) in the interval,

$$||\bar{X} - \bar{X}_o|| \leq r$$

(The proof is given in Appendix C).

The iterate, $\bar{X}_n$, can be shown as follows to satisfy

$$||\bar{X}_{n+1} - \bar{X}_n|| \leq M^n.(1-M).r$$

as from (2.8)

$$||\bar{X}_{n+1} - \bar{X}_n|| = ||\bar{F}(\bar{X}_n) - \bar{F}(\bar{X}_{n-1})||$$

Thus, by (2.13), the Lipschitz condition yields,

$$\begin{aligned} ||\bar{X}_{n+1} - \bar{X}_n|| &\leq M \ \ ||\bar{X}_n - \bar{X}_{n-1}|| \\ &\leq M^2 \ ||\bar{X}_{n-1} - \bar{X}_{n-2}|| \\ &\cdot \\ &\cdot \\ &\cdot \\ &\leq M^n \ ||\bar{X}_1 - \bar{X}_o|| = M^n||\bar{F}(\bar{X}_o)-\bar{X}_o|| \end{aligned}$$

and applied condition (2.15), then

$$||\bar{X}_{n+1} - \bar{X}_n|| \leq M^n.(1-M).r \tag{2.17}$$

As a consequence of this result (2.17) it is seen that the iterates converge geometrically, and at least as fast as $M^n \rightarrow 0$.

It also can be shown from (2.16) and (2.8) that,

$$||\bar{F}(\bar{X}_n) - \bar{X}_n|| \leq M.||\bar{F}(\bar{X}_{n-1}) - \bar{X}_{n-1}|| \tag{2.18}$$

A similar theorem, "Subcontracting Mapping Theorem", is given by Goldstein (G1). In his theorem the initial point $\bar{X}_o$ satisfies the following condition

$$||\bar{F}(\bar{X}_1) - \bar{F}(\bar{X}_o)|| \leq ||\bar{F}(\bar{X}_o) - \bar{X}_o||$$

or with (2.8)

$$||\bar{F}(\bar{X}_1) - \bar{X}_1|| \leq ||\bar{F}(\bar{X}_o) - \bar{X}_o||$$

which replaces the initial condition (2.15).

A more useful result can be obtained if we are willing to place more restrictions on $\bar{F}(\bar{X})$ and assume the existence of a solution $\bar{X}_s$.

Theorem 2

Let the iteration (2.8) have a root X_s and let the functions $F_i(\bar{X})$ have continuous first order partial derivatives,

$$F_{ij}(\bar{X}) = \frac{\partial F_i(\bar{X})}{\partial x_j} \tag{2.19}$$

and satisfy

$$\max_i \sum_{j=1}^{m} |F_{ij}(\bar{X})| \leq M < 1 \tag{2.20}$$

for all $\bar{X}$ in

$$||\bar{X} - \bar{X}_s||_\infty \leq r \tag{2.21}$$

(for the norm see equation (2.10))

Then, i) for any $\bar{X}_o$ satisfying (2.21) all the iterates $\bar{X}_n$ of (2.8) also satisfy (2.21)

ii) for any $\bar{X}_o$ satisfying (2.21) the iterates (2.8) converge to the solution $\bar{X}_s$ which is unique in (2.21)

(See Appendix C)

The iteration can be shown as follows to satisfy

$$||\bar{X}_n - \bar{X}_s||_\infty < M^n.r$$

From (2.8) and (2.10)

$$||\bar{X}_{n+1} - \bar{X}_s||_\infty = ||\bar{F}(\bar{X}_n) - \bar{F}(\bar{X}_s)||_\infty$$

Using the mean value theorem

$$F_i(\bar{X}_n) - F_i(\bar{X}_s) = \sum_{j=1}^{m} F_{ij}(\bar{\xi}_i).(x_{n,j} - x_{s,j})$$

or
$$||\bar{F}(\bar{X}_n) - \bar{F}(\bar{X}_s)||_\infty = \max_i \{ \sum_{j=1}^{m} F_{ij}(\bar{\xi}_i).(x_{n,j} - x_{s,j})\}$$

where $\bar{\xi}_i$ is a point on the open line segment joining $\bar{X}_n$ and $\bar{X}_s$. Thus with (2.10) and (2.20) yields

$$\begin{aligned}||\bar{X}_{n+1} - \bar{X}_s||_\infty &= \max_i \{ \sum_{j=1}^{m} F_{ij}(\bar{\xi}_i).(x_{n,j} - x_{s,j})\} \\ &\leq \max_i \{ \sum_{j=1}^{m} F_{ij}(\bar{\xi}_i)\}.||\bar{X}_n - \bar{X}_s||_\infty \\ &< M.||\bar{X}_n - \bar{X}_s||_\infty \\ &< M^2.||\bar{X}_{n-1} - \bar{X}_s||_\infty \\ &\vdots \\ &< M^{n+1}.||\bar{X}_o - \bar{X}_s||_\infty \\ &< M^{n+1} . r\end{aligned}$$

and
$$||\bar{X}_n - \bar{X}_s|| < M^n.r \tag{2.22}$$

Here again, from (2.22), it is seen that the iterates approach the solution geometrically, and at least as fast as $M^n \to 0$.

As theorem 2 is only a sufficient condition for convergence, the iterate might converge to solution without $\bar{X}_o$ satisfying (2.20) and (2.21). Therefore, it may be that if equation (2.20) is satisfied for $\bar{X}$ within a sphere of radius r around the solution $\bar{X}_s$, then the iteration may converge to $\bar{X}_s$ if $\bar{X}_o$ is close enough to solution X_s, but not necessarily within that sphere, as required by (2.21).

For example, we may examine one dimension iterative calculation as

$$x_{n+1} = g(x_n) \qquad n = 0, 1, 2, 3 \ldots \tag{2.23}$$

Then (2.20) and (2.21) become

$$|g'(x)| \leq M < 1 \tag{2.24}$$

for all x in

$$|x - x_s| \leq r \tag{2.25}$$

Thus in Figure 2.6, although x_o is outside of the interval of (2.25) and x_o does not satisfy (2.24) since

$$|g'(x_o)| > 1$$

the iteration still converges.

2.2.2 Convergence Test

If the iterative calculation is a convergent process, the iterate $\bar{X}_n$ would approach the solution $\bar{X}_s$ asymptotically, that is, that $\bar{X}_n$ would approach $\bar{X}_s$ arbitrarily closely as n approaches infinity. In a practical case it may be sufficient to iterate until reaching some point $\bar{X}_c$, close enough to the solution $\bar{X}_s$, and to consider this point as the solution.

However, the test of convergence used in iterative calculations commonly is to examine the fractional change of every element in vector $\bar{X}$, between two successive iterations. Hence, the test vector may be

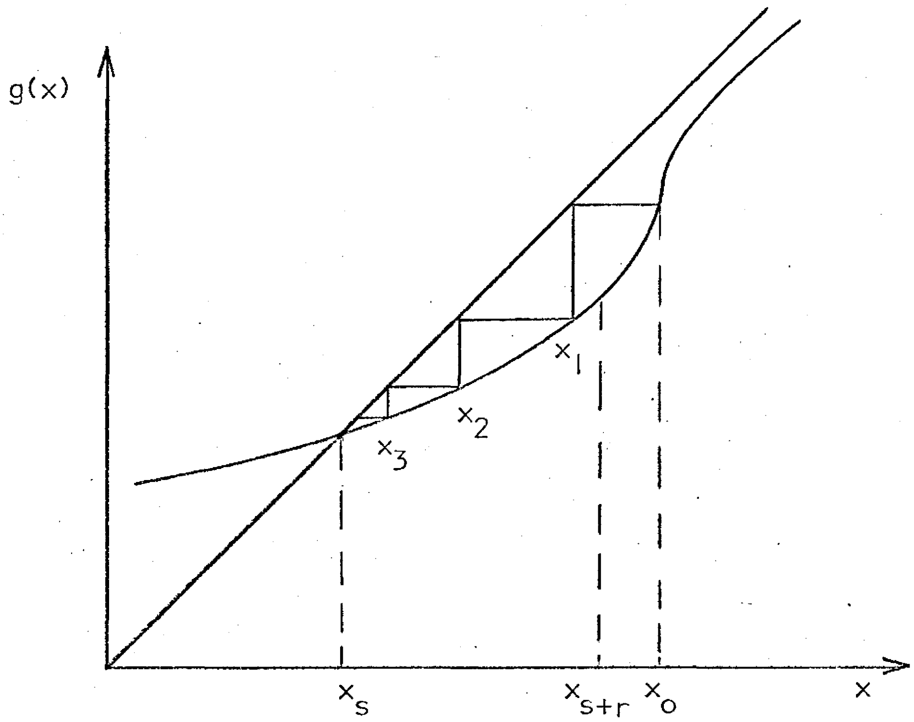

FIGURE 2.6 SINGLE EQUATION ITERATIVE CALCULATION

written as,

$$\bar{\theta}_n = [I\bar{X}_n]^{-1}(\bar{X}_{n+1} - \bar{X}_n) \tag{2.26}$$

where $[I\bar{X}]$ is to be interpreted as a diagonal matrix with elements x_i throughout this thesis. The iteration is considered to have converged when every element in vector $\bar{\theta}_n$ is less in modulus than some small tolerance. This kind of convergence test gives an indication of the fractional change of every element in the vector $\bar{X}$, but does not show how far $\bar{X}_n$ is from the true solution $\bar{X}_s$. In addition, in order to obtain the same deviation from the true solution for two different cases, the tolerance of the test vector (2.26) ought to be chosen according to the rate of convergence. It is well known that the slower the iteration converges, the smaller the tolerance must be for the same absolute accuracy. Later in this work a technique for choosing the right tolerance will be discussed.

Another indicator for convergence is the deviation vector which is defined as follows

$$\bar{D}_n = [I\ \bar{X}_s]^{-1}\ (\bar{X}_s - \bar{X}_n) \tag{2.27}$$

The elements in the deviation vector give the fractional difference between $\bar{X}_n$ and the solution $\bar{X}_s$. The deviation vector is hardly ever used as the convergence indicator because $\bar{X}_s$ is unknown. However, if a relation between $\bar{D}_n$ and $\bar{\theta}_n$ can be found, $\bar{D}_n$ may be used in choosing the right tolerance for the test vector $\bar{\theta}_n$ such that $\bar{X}_n$ is sufficiently close to $\bar{X}_s$ as given by a specified $\bar{D}_n$.

The test of convergence can also be done on the process itself (2.5) separately from the computation method. In other words, the test would examine the fractional difference between vector $\bar{X}_n$ and $\bar{Y}_n$ regardless of the kind of convergence technique used for evaluating $\bar{X}_{n+1}$, such as successive substitution (2.6) or some accelerating method.

This kind of test may be called fixed-point test vector and can be written as,

$$\bar{T}_n = [I\ \bar{X}_n]^{-1}\ (\bar{Y}_n - \bar{X}_n) \tag{2.28}$$

Since the fixed-point test vector is independent of the convergence technique, it was used for comparison of the various accelerating methods. It is worthwhile mentioning that for successive-substitution iteration (2.6) the test vector $\bar{\Theta}_n$ (2.26) and the fixed point test $\bar{T}_n$ (2.28) are the same.

$$\bar{T}_n \equiv \bar{\Theta}_n \tag{2.29}$$

if $\bar{X}_{n+1} = \bar{Y}_n$.

2.3 Linear Iteration

It is well established that modern linear system theory has been highly successful when applied to practical cases (S5), in spite of the need to use approximations. That is, that equation (2.5) may be approximated by linear equation as

$$\bar{F}(\bar{X}) = \bar{\bar{A}}\ \bar{X} + \bar{b} \tag{2.30}$$

or

$$\bar{Y}_n = \bar{\bar{A}}\ \bar{X}_n + \bar{b} \tag{2.31}$$

where $\bar{\bar{A}}$ is an $m \times m$ matrix of the linear coefficients

$\bar{b}$ is a vector constants

and successive substitution as in equation (2.6) applied to linear system (2.31) gives,

$$\bar{X}_{n+1} = \bar{\bar{A}}\,\bar{X}_n + \bar{b} \tag{2.32}$$

for some given X_o.

The linear iterative system (2.32) is useful because the mathematical knowledge of this system is well established (A2, F1). Iterative calculations of chemical plants are often linear or very close to linear, and convergence promotion techniques that are suitable for linear systems may be used for practical non-linear cases as an approximation.

2.3.1 Convergence Condition of Linear Iteration

The two theorems (1 and 2) that were applied for general iterative process are obviously true for linear systems. Applying (2.19), the partial derivatives of the set of linear functions (2.30) gives,

$$F_{ij} = \frac{\partial F_i(\bar{X})}{\partial x_j} = a_{ij} \tag{2.33}$$

where a_{ij} is the element (i, j) of the matrix A. Therefore the maximum absolute row sum (2.20), for a linear system is

$$\max_i \sum_{j=1}^{m} |a_{ij}| \leq M < 1 \tag{2.34}$$

This maximum absolute row sum is usually known as the matrix norm $||A||_\infty$

$$||A||_\infty = \max_i \sum_{j=1}^{m} |a_{ij}| \leq M < 1 \tag{2.35}$$

As the elements of the matrix A are constant, (2.35) is independent of $\bar{X}$, therefore if (2.35) is satisfied the iteration will converge (Theorem 2).

Faddeeva (FI) shows that in order that the process of linear iteration converge, it is sufficient that any norm of the matrix A be less than unity. That is, that

$$|| A || < 1 \qquad (2.36)$$

Faddeeva also has given necessary and sufficient conditions for convergence of linear systems as follows:

Theorem 3

For convergence of the process of linear iteration (2.32) with any initial vector X_o and with any value of the vector b, it is necessary and sufficient that all the eigenvalues of the matrix A be less than unity in modulus. Thus,

$$| \lambda_j | < 1 \qquad (2.37)$$

$$j = 1, 2, 3 \ldots m$$

2.3.2 The Solution of the Linear Iteration

The linear iteration equation (2.32), which is a linear difference equation, may be solved generally for some initial value X_o, (See Appendix A) to give,

$$X_n = \sum_{j=1}^{m} \frac{W_j^T[X_o - X_s]}{W_j^T Z_j} \cdot Z_j \cdot \lambda_j^n + X_s \qquad (2.38)$$

$$X_s = (I - A)^{-1} b \qquad (2.39)$$

where X_s - is the steady state solution of (2.32)

λ_j - the eigenvalue of matrix A

Z_j - the eigenvector of matrix A

W_j - the eigenrow of matrix A

If

$$C_j = \frac{W_j^T[X_o - X_s]}{W_j^T Z_j} \qquad j = 1, 2, \ldots m. \tag{2.40}$$

then the solution becomes

$$\bar{X}_n = \sum_{j=1}^{m} C_j \bar{Z}_j \lambda_j^n + \bar{X}_s \tag{2.41}$$

Examining the solution of the difference equation, it is clear that if the eigenvalues of matrix A are less than unity in modulus the iteration will converge, (Theorem 3), as

$$\lim_{n\to\infty} \lambda_j^n \to 0 \qquad j = 1, 2, 3 \ldots m.$$

and

$$\lim_{n\to\infty} \bar{X}_n = \lim_{n\to\infty} \{ \sum_{j=1}^{m} C_j \bar{Z}_j \lambda_j^n + \bar{X}_s \} = \bar{X}_s$$

Furthermore, the rate of convergence depends on the absolute value of the eigenvalues (2.41), such that, the smaller the eigenvalues in modulus, the faster the iterate will converge. Also, the rate of convergence depends on the coefficients C_j (2.40) which are linear functions of the distance between the initial point X_o and the solution X_s. Obviously, the number of iterations needed for converging would be less if the initial point X_o is closer to solution X_s.

2.3.3 Dominant Eigenvalue

The solution of the linear difference equation (2.41), that is the linear iterative calculation, is a linear combination of the eigenvectors $\bar{Z}_j$ with the scalar coefficients $C_j \lambda_j^n$. Every single coefficient j declines geometrically as n increases. The bigger the eigenvalue in modulus the slower the coefficient decreases. Therefore, if λ_k is the biggest eigenvalue in modulus of the matrix A, then after sufficient number of iterations, only one term of the summation remains since all the other coefficients have become very small. The solution (2.41) then becomes

$$\bar{X}_n = C_k \bar{Z}_k \lambda_k^n + \bar{X}_s \qquad (2.42)$$

Thus, after a sufficient number of iterations the biggest eigenvalue in modulus controls the rate of convergence of the iterative process and the iterate converges geometrically to $\bar{X}_s$ (2.42).

This geometric behaviour of the iteration may be utilized for determining the tolerance for the test vector (2.26) and for promoting the convergence.

2.3.4 The Convergence Tolerance for the Test Vector (Linear Iteration)

The tolerance of the deviation vector (2.27) can be chosen independently of the iterative process. The problem formulator may choose it according to the necessary accuracy.

For determining the tolerance of the test vector (2.26) from the selected tolerance of the deviation vector, equation (2.42) may be used. This equation (2.42) can be assumed valid as the iteration approaches convergence.

Substituting equation (2.42) into equation (2.27) the deviation vector becomes

$$\bar{D}_n = -[I \; \bar{X}_s]^{-1} \; C_k \bar{Z}_k \lambda_k^n \qquad (2.44)$$

and substituting equation (2.42) into the numerator of equation (2.26) the test vector becomes

$$\bar{\Theta}_n = [I\ \bar{X}_n]^{1}\ C_k \bar{Z}_k \lambda_k^n (\lambda_k - 1) \tag{2.45}$$

Assume first that $[I\ \bar{X}_n]^{1} \cong [I\ \bar{X}_s]^{-1}$, then from equations (2.44) and (2.45) the following relation between $\bar{\Theta}$ and $\bar{D}$ can be found

$$\bar{\Theta}_n = \bar{D}_n \cdot \phi_k \tag{2.46}$$

where $\phi_k = (1-\lambda_k)$.

A more general relation between $\bar{\Theta}_n$ and $\bar{D}_n$ can be found by substituting equation (2.42) into the denominator of equation (2.45) and then with equation (2.44) we get

$$\bar{\Theta}_n = [\ I - I\bar{D}_n\ ]^{-1} \cdot \bar{D}_n \cdot \phi_k \tag{2.47}$$

Now, if $\bar{X}_n$ is very close to the solution $\bar{X}_s$, that is, if every element in vector $\bar{D}_n$ is small, then the first part of equation (2.47) is almost the unit matrix I and (2.47) becomes equivalent to equation (2.46).

Equation (2.46) or (2.47) can be used for determining the tolerance for the test vector if the tolerance for the deviation vector had been chosen and the biggest eigenvalue is known. If β is the tolerance for the deviation vector, then

$$||\bar{D}_n||_\infty \leq \beta \tag{2.48}$$

and the iteration converges if the vector $\bar{D}_n$ satisfies (2.48). Now, using (2.46)

$$||\bar{\Theta}_n||_\infty = ||\bar{D}_n||_\infty \cdot |\phi_k|$$

and substituting (2.48) we get

$$||\bar{\Theta}_n||_\infty \leq \beta \cdot |\phi_k| \qquad (2.49)$$

Therefore the iteration satisfies (2.48) if the vector test $\bar{\Theta}_n$ satisfies (2.49).

As an example of the tolerance calculation, suppose

$$\beta = 0.001$$

$$\lambda_k = 0.9$$

then

$$\phi_k = 0.1$$

$$||\bar{\Theta}_n||_\infty \leq 0.0001$$

The tolerance for the test vector is ten times smaller than the tolerance for the deviation vector, when the biggest eigenvalue is 0.9. Table 2.2 gives the correction factor ϕ_k as a function of λ_k.

TABLE 2.2

The Correction Factor ϕ_k

λ_k	ϕ_k	$1/\phi_k$
0.5	0.5	2.0
0.6	0.4	2.5
0.7	0.3	3.33
0.8	0.2	5.00
0.85	0.15	6.67
0.90	0.10	10.
0.95	0.05	20.
0.97	0.03	33.3
0.98	0.02	50.
0.99	0.01	100.
0.999	0.001	1000.

The biggest eigenvalue in modulus of matrix A (2.30) can be obtained from the iteration by equation (2.42) using three successive values of vector $\bar{X}$ (X_{n-2}, X_{n-1}, X_n). The method for general iteration will be discussed in detail later in this work. The biggest eigenvalue in modulus can be also calculated by the power (L4) method.

2.4 The General Iteration

Geometric convergence was observed in iterative calculation of recycle processes (AI) or heat and material balance of distillation columns (P2). Thus, not only linear iterations (2.32) approach the form of equation (2.42) but also non-linear iteration (2.7) asymptotically approach the geometric progression. Although it is not a general statement for any non-linear iteration, in practical calculation of chemical engineering processes when the rate of convergence is very slow or the iterates close to solution, the iteration can be approximated by a geometric progression in form of equation (2.42) as,

$$\bar{X}_n = \bar{U}\,\mu^n + \bar{X}'_s \qquad (2.50)$$

where $\bar{U}$ is vector of coefficients

$\bar{X}'_s$ is the approximation to the solution $\bar{X}_s$

μ is the geometric coefficient

When the iteration (2.7) approaches a geometric progression, equation (2.50) can be used to obtain the geometric coefficient μ which may be used for determining the tolerance for the test vector (2.26) and for determining the rate of convergence. Furthermore, if the iteration has approached geometric progression in a small number of iterations, equation (2.50) may be utilized, as a convergence promotion method, for approximating the solution.

2.4.1 Evaluation of the Geometric Coefficient

The value in modulus of the geometric coefficient can be calculated directly from the iterate vectors ($\bar{X}_{n-1}$, $\bar{X}_n$, ...) if the number of iterations is sufficient for the iteration (2.7) to have approached geometric progression (2.50). Therefore the iterate vectors may be examined first in order to detect when (2.50) becomes valid and then to calculate the geometric coefficient.

We may define an error vector $\bar{E}_n$, which will be used in the derivation and later will be replaced by the test vector (2.26), as

$$\bar{E}_n = \bar{X}_{n+1} - \bar{X}_n \tag{2.52}$$

and we may define μ_n as the ratio between the norms of two successive error vectors, as

$$\mu_n = \frac{||\bar{E}_n||}{||\bar{E}_{n-1}||} = \frac{||\bar{X}_{n+1} - \bar{X}_n||}{||\bar{X}_n - \bar{X}_{n-1}||} \tag{2.53}$$

with any norm (2.10), (2.11) or (2.12). Now, if (2.50) is valid the error vector (2.52) becomes

$$\bar{E}_n = \bar{X}_{n+1} - \bar{X}_n = \bar{U}\,\mu^n(\mu - 1) \tag{2.54}$$

and (2.53) becomes

$$\mu_n = \frac{||\bar{E}_n||}{||\bar{E}_{n-1}||} = |\mu| \tag{2.55}$$

From (2.55) we can see that as the iterate approaches the geometric progression, μ_n approaches a constant value equal to the absolute value of the geometric coefficient μ.

Now, if Λ_n is the fractional change of μ_n in two successive iterations,

$$\Lambda_n = (\mu_n - \mu_{n-1}) / \mu_{n-1} \tag{2.56}$$

then if Λ_n is less than some small tolerance γ, (2.50) is assumed to be valid and the geometric coefficient in modulus is μ_n (equation (2.55). That is, if

$$| \Lambda_n | \leq \gamma \tag{2.57}$$

then

$$| \mu | = \mu_n$$

Practically, $\bar{E}_n$, μ_n and Λ_n can be calculated every iteration, and if Λ_n is satisfied (2.57) then $|\mu| = \mu_n$. The sign of the geometric coefficient can be calculated by using one of the elements of the error vector $\bar{E}_n$. Suppose $E_{n,j}$ is the largest element in the error vector, then the sign of the geometric coefficient μ is the same as the sign of $\mu_{n,j}$ that is obtained by the following ratio

$$\mu_{n,j} = \frac{E_{n,j}}{E_{n-1,j}} = \frac{x_{n+1,j} - x_{n,j}}{x_{n,j} - x_{n-1,j}}$$

Actually, the test vector (2.26) can be used instead of the error vector (2.52) for obtaining the geometric coefficient. Equation (2.55) becomes

$$\mu_n = \frac{||\bar{\theta}_n||}{||\bar{\theta}_{n-1}||} \tag{2.58}$$

If the iterate approaches geometric progression (2.50), then by substitution of (2.50) into (2.26) the test vector becomes

$$\bar{\Theta}_n = [I\ \bar{X}_n]^{-1}\ \bar{U}\ \mu^n(\mu-1) \tag{2.59}$$

and equation (2.58) becomes

$$\mu_n = \frac{||\bar{\Theta}_n||}{||\bar{\Theta}_{n-1}||} \simeq |\mu| \tag{2.60}$$

since $[I\bar{X}_n]^{-1} \cong [I\bar{X}_{n-1}]^{-1}$. Practically,. it is preferable to use the test vector $\bar{\Theta}_n$ as it gives fractional change between the two iterations.

For the case of linear iteration (2.32), the biggest eigenvalue in modulus can be calculated by the same procedure as above. If the number of iterations is sufficient and the linear iteration has approached a geometric progression which is dominated by the biggest eigenvalue in modulus, equation (2.42) becomes valid and by substituting equation (2.45) into equation (2.58) we get,

$$\mu_n = \frac{||\bar{\Theta}_n||}{||\bar{\Theta}_{n-1}||} \simeq |\lambda_k| \tag{2.61}$$

since $[I\bar{X}_n]^{-1} \simeq [I\bar{X}_{n-1}]^{-1}$. The sign of the biggest eigenvalue in modulus can be obtained in the same way as the sign of the geometric coefficient μ was obtained.

Finally, if the fixed point test vector (2.28) is obtained in every iteration instead of the test vector (2.26) we may define μ'_n as the ratio between the norms of two successive fixed point test vectors, as

$$\mu'_n = \frac{||\bar{T}_n||}{||\bar{T}_{n-1}||} \tag{2.62}$$

Now, if the computational method, which was used instead of the successive substitution equation (2.6) in order to accelerate the convergence, has the following linear form as

$$\bar{X}_{n+1} = \bar{X}_n + \bar{\bar{\Gamma}}(\bar{Y}_n - \bar{X}_n) \tag{2.63}$$

where $\bar{\bar{\Gamma}}$ is a non-singular matrix, and if the iteration approaches geometric progression so that equation (2.50) becomes valid, then μ'_n approaches a constant value $|\mu'|$, as

$$\mu'_n = \frac{||\bar{T}_n||}{||\bar{T}_{n-1}||} \approx |\mu'| \tag{2.64}$$

and it will be shown in the following page that

$$|\mu'| \equiv |\mu|$$

where μ is defined by equation (2.60).

Thus, assuming that the iteration has approached a geometric progression and equation (2.50) is valid and also assuming that $[I\bar{X}_n]^{-1} \approx [I\bar{X}_{n-1}]^{-1}$ then from (2.50) we get

$$\bar{X}_n = \bar{U}\,\mu^n + \bar{X}'_s$$

and

$$\bar{X}_{n+1} - \bar{X}_n = \bar{U}\,\mu^n\,(\mu-1) \tag{2.65}$$

and from (2.63) we get

$$\bar{X}_{n+1} - \bar{X}_n = \bar{\bar{\Gamma}}(\bar{Y}_n - \bar{X}_n) \tag{2.66}$$

From (2.65) and (2.66) we get

$$\bar{Y}_n - \bar{X}_n = (\bar{\bar{\Gamma}})^{-1}\,\bar{U}\,\mu^n\,(\mu-1)$$

or

$$||\bar{Y}_n - \bar{X}_n|| = ||(\bar{\bar{\Gamma}})^{-1}.\bar{U}||.|\mu^n(\mu-1)| \tag{2.67}$$

and

$$\frac{||\bar{Y}_n - \bar{X}_n||}{||\bar{Y}_{n-1} - \bar{X}_{n-1}||} = |\mu| \tag{2.68}$$

and from (2.64) and (2.68), we get

$$|\mu| = \frac{||\bar{T}_n||}{||\bar{T}_{n-1}||} = |\mu'| \tag{2.69}$$

since $[I\bar{X}_n]^{-1} \simeq [I\bar{X}_{n-1}]^{-1}$.

The convergence promotion equation in form of equation (2.63) will be discussed in detail in Section 3.

2.4.2 The Convergence Tolerance

The mathematical relation between the test vector (2.26) and the deviation vector (2.27) can be obtained for the general iteration in a way similar to that for linear iteration (Section 2.3.4). As the iterate $\bar{X}_n$ approaches the solution $\bar{X}_s$, geometric progression usually dominates the iteration and equation (2.50) can be assumed to be valid.

Substituting equation (2.50) into equation (2.27), the deviation vector becomes

$$\bar{D}_n = -[I\bar{X}_s]^{-1}\,\bar{U}\,\mu^n \tag{2.72}$$

and substituting (2.50) into the numerator of equation (2.26) the test vector becomes

$$\bar{\Theta}_n = [I\bar{X}_n]^{-1}\,\bar{U}\,\mu^n(\mu-1) \tag{2.73}$$

Now, assuming that $[I\bar{X}_n]^{-1} \simeq [I\bar{X}_s]^{-1}$, from equations (2.72) and (2.73) we get

$$\bar{\Theta}_n = \bar{D}_n \cdot \phi \tag{2.74}$$

where

$$\phi = (1-\mu)$$

which is the same result as equation (2.46). Equation (2.74) can be used for obtaining the tolerance for the test vector Θ_n when the geometric coefficient μ is known and the tolerance for the deviation vector has been chosen prior to iteration (See the example in Section 2.3.4). This equation (2.74), may be used for calculating the deviation vector or its norm from the test vector, Θ_n, and the geometric coefficient as the iteration proceeds.

2.4.3 Rate of Convergence

Let it be required to reduce the amplitude of the error by a factor of at least 10^{-P}. From equation (2.22), we see that, the norm of the error $||\bar{X}_n - \bar{X}_s||_\infty$ is reduced by at least a factor of M^n. The number of iterations required is the least value of n for which

$$M^n < 10^{-P}$$

if $0 \leq M < 1$, by taking logs we obtain

$$n \geq \frac{P}{-\log M} = \frac{P}{R}. \tag{2.75}$$

Thus, the number of iterations required to reduce the initial error by the factor 10^{-P} is inversely proportional to R, the rate of convergence.

Now, for iterative calculations (2.7) which asymptotically approach geometric convergence, that is, when the iterate satisfies (2.50), the factor M can be replaced by the geometric coefficient μ. If k is the number of iterations necessary for the iteration to reach geometric behaviour then (2.50) becomes

$$\bar{X}_{k+\nu} = \bar{U}\,\mu^{k+\nu} + \bar{X}'_s$$

and with

$$\bar{X}_k = \bar{U}\,\mu^{k} + \bar{X}'_s$$

we get

$$(\bar{X}_{k+\nu} - \bar{X}'_s) \cong \mu^{\nu}(\bar{X}_k - \bar{X}'_s) \qquad (2.76)$$

The quantity μ is frequently called the asymptotic convergence factor and the rate of convergence is

$$R = -\log|\mu| \qquad (2.77)$$

The rate of convergence R as a function of the asymptotic convergence factor μ (the geometric coefficient or the biggest eigenvalue for linear iteration) is given in Table 2.3. The value 1/R, also given there, is the number of iterations necessary to reduce the norm of the error by a factor of one-tenth (1/10).

TABLE 2.3

The Rate of Convergence R as a Function of the Asymptotic Convergence Factor μ*

μ	R	1/R
0.50	.301	3.322
0.55	.260	3.852
0.60	.222	4.508
0.65	.187	5.345
0.70	.155	6.456
0.75	.125	8.004
0.80	.097	10.319
0.85	.0706	14.168
0.86	.0655	15.267
0.87	.0605	16.534
0.88	.0555	18.012
0.89	.0506	19.759
0.90	.0458	21.854
0.91	.0410	24.415
0.92	.0362	27.615
0.93	.03152	31.729
0.94	.02687	37.213
0.95	.02228	44.891
0.96	.01773	56.406
0.97	.01323	75.596
0.98	.008774	113.974
0.99	.004365	229.105

* The geometric coefficient or the biggest eigenvalue λ_k.

3. CONVERGENCE PROMOTION TECHNIQUES

In this section various acceleration techniques for promoting the rate of convergence will be discussed. A few of these techniques can be included in the form of a general equation of convergence promotion which replaces the successive substitution equation (2.6). This general equation of convergence promotion is applied every iteration and becomes a part of the iteration procedure. The convergence promotion coefficients of the general equation are constant and are chosen prior to the calculation.

Another method to be discussed is the geometric extrapolation method. This method can be applied only when the iteration approaches geometric behaviour, that is when equation (2.50) becomes valid. Hence the geometric extrapolation method does not constitute a part of the iteration procedure and it is applied when the variables of the iteration reach geometric progression. This method is very useful for cases of slow convergence when the geometric progression is achieved in small number of iterations, (suppose less than ten).

The third method that will be discussed is the multi-dimensional extrapolation. This method can be useful for cases which have a small number of variables ($m \leq 10$).

The last one, the partition recycle method belongs to the class of problems which deal with the structure of the recycle calculation.

3.1 The General Equation

The successive substitution equation (2.6) can be replaced by the general equation of convergence promotion which is

$$\bar{X}_{n+1} = \bar{X}_n + t.\bar{\bar{G}}.(\bar{Y}_n - \bar{X}_n) \qquad (3.1)$$

$$n = 0, 1, 2, 3, \ldots$$

and by substituting (2.5) into (3.1) the iteration with convergence promotion becomes

$$\bar{X}_{n+1} = \bar{X}_n + t.\bar{\bar{G}}.(\bar{F}(\bar{X}_n) - \bar{X}_n) \qquad (3.2)$$

where $\bar{\bar{G}}$ - matrix of convergence promotion coefficients

t - relaxation factor (scalar).

The successive substitution equation (2.6) is a particular case of equation (3.1) when $t = 1$ and G is the unit matrix I.

The various techniques which are represented by the general equation (3.1) are first distinguished by the structure of matrix G, which may be divided into two main sets:

1. full matrices of convergence coefficients.
2. diagonal matrices of convergence coefficients.

The first one, the full matrix G, is more comprehensive, demands more computer storage and manipulation, but can overcome most of the recycle calculation problems. The second one, the diagonal matrix, is a simpler technique, it saves computer storage and calculation time, but can be used only for a limited number of cases.

The problem of slow convergence appears usually in some neighbourhood close to the solution $\bar{X}_s$, where the process can be approximated by a linear equation as (2.30). The general process can be linearized in some neighbourhood of the solution and the linear equation can be used to obtain the convergence promotion coefficient. The following mathematical development is mainly concerned with linear iteration, however, we should bear in mind that the general iteration can be linearized and represented by a linear equation as an approximation. The linearization procedures are given in Appendix B.

By substituting the linear equation (2.30) into (3.2) we get the linear iterative calculation with convergence promotion as

$$\bar{X}_{n+1} = \bar{X}_n + t.\bar{\bar{G}}.(\bar{\bar{A}}\ \bar{X}_n + \bar{b} - \bar{X}_n)$$

or

$$\bar{X}_{n+1} = \{I + t.\bar{\bar{G}}.(\bar{\bar{A}}-I)\}.\bar{X}_n + t.\bar{\bar{G}}.\bar{b} \qquad (3.3)$$

let

$$\bar{\bar{B}} = I + t.\bar{\bar{G}}.(\bar{\bar{A}} - I) \qquad (3.4)$$

then equation (3.3) becomes

$$\bar{X}_{n+1} = \bar{\bar{B}}.\bar{X}_n + t.\bar{\bar{G}}.\bar{b} \qquad (3.5)$$

where (3.5) is a new iterative process.

The rate of convergence of (3.5) depends upon the eigenvalues of matrix B. Therefore, we wish to find the convergence promotion matrix G such that the biggest eigenvalue in modulus of matrix B will be as small as possible.

3.1.1 Full Matrices of Convergence Coefficients

If $\bar{\bar{G}}$ is to be a full matrix of coefficients, the matrix should be chosen as

$$\bar{\bar{G}} = (I - \bar{\bar{A}})^{-1} \qquad (3.6)$$

and t = 1, so that $\bar{\bar{B}}$ (3.4) is the zero matrix, with all eigenvalues also zero. That is, that linear iterative process (3.3) would be converged in one iteration if matrix G were chosen as (3.6) and t = 1.

For a general case (2.5), the process may be linearized in order to get matrix A^* (See Appendix B) and equation (3.1) will be used in the convergence promotion routine, as

$$\bar{X}_{n+1} = \bar{X}_n + t.(I - \bar{\bar{A}}*)^{-1}.(\bar{Y}_n - \bar{X}_n) \qquad (3.7)$$

where A* is the linearization matrix and t is a relaxation factor. This relaxation factor is there to prevent oscillation of vector $\bar{X}_{n+1}$ around the solution.

This technique, with a full matrix of coefficients G, is similar to the modified form of Newton's method for solving a set of algebraic equations (L3) where the Jacobian matrix has been evaluated only once at the point $\bar{X}_o$ and used unchanged through all the iterations.

The linearization can be made around the initial point $\bar{X}_o$ or any other point of the iteration $\bar{X}_n$. Practically it may be better to iterate (2.7) until the rate of convergence becomes slow and then, since $\bar{X}_n$ is closer to the solution $\bar{X}_s$, to linearize the process for obtaining A* to use in equation (3.7).

3.1.2 Diagonal Matrices of Convergence Coefficients

Two kinds of diagonal matrices will be discussed in this section.

(a)
$$\bar{\bar{G}} = \alpha I \qquad (t = 1) \tag{3.8}$$

where α is a scalar.

Substituting (3.8) into (3.4), then

$$\bar{\bar{B}} = I + \alpha I(\bar{\bar{A}} - I)$$

or
$$\bar{\bar{B}} = (1-\alpha)I + \alpha\bar{\bar{A}} \tag{3.9}$$

Now, as before, we wish to find a coefficient α such that the eigenvalues of matrix B will be as small as possible in modulus. Hence, since $\bar{\bar{B}}$ is a matrix polynomial of A (3.9), then each eigenvalue of B is the same scalar polynomial of the respective eigenvalue of A, therefore

$$\lambda_B = (1-\alpha) + \alpha\lambda_A \tag{3.10}$$

Now, if $\lambda_{A,b}$ and $\lambda_{B,b}$ are the biggest eigenvalues of matrix A and B respectively, and $\lambda_{A,s}$ and $\lambda_{B,s}$ are the smallest eigenvalues of A and B respectively, then for $\alpha > 0$

$$\lambda_{B,b} = (1 - \alpha) + \alpha\lambda_{A,b}$$

$$\lambda_{B,s} = (1 - \alpha) + \alpha\lambda_{A,s} \tag{3.11}$$

All the other eigenvalues of the two matrices are located between the extreme values as

$$\lambda_{B,s} < \lambda_B < \lambda_{B,b}$$

$$\lambda_{A,s} < \lambda_A < \lambda_{A,b}$$

Figure 3.1 illustrates the behaviour of equation (3.11), where the upper line is $\lambda_{B,b}$ and the lower line is $\lambda_{B,s}$, and both of them vary with α. Obviously, when $\alpha = 1$, the eigenvalues of matrix B are equal to the eigenvalues of matrix A, as the two matrices are identical (3.9).

For accelerating the convergence the best α that may be chosen is the one that will satisfy (3.12), (the dashed line in Figure 3.1).

$$-\lambda_{B,s} = \lambda_{B,b} \tag{3.12}$$

and from (3.12) and (3.11) α is

$$\alpha = \frac{2}{2 - \lambda_{A,s} - \lambda_{A,b}} \tag{3.13}$$

This method of a diagonal matrix with one coefficient α (3.8) is very useful as an accelerating method when all the eigenvalues of matrix A are squeezed together as shown in Figure (3.1a). For example suppose

$$\lambda_{A,b} = 0.95 \quad \text{and} \quad \lambda_{A,s} = 0.65$$

Since the biggest eigenvalue is 0.95, the iterative process converges very slowly(See Table 2.3). From equation (3.13)

$$\alpha = \frac{2}{2 - 0.75 - 0.95} = 5$$

the new eigenvalues of matrix B are

$$\lambda_{B,b} = -\lambda_{B,s} = (1-5) + 5 \times 0.95 = 0.75$$

Reducing the biggest eigenvalue from 0.95 to 0.75 increases the rate of convergence significantly. From Table (2.3) it can be seen that the rate of convergence for eigenvalue 0.95 is 0.0223 and for eigenvalue 0.75 is 0.125 which means that the rate of convergence increases by factor 5.6 . However, if the eigenvalues of matrix A are not squeezed together, this method improves the rate of convergence very little, as shown in Figure (3.1b). For example, suppose

$$\lambda_{A,b} = 0.95 \quad \text{and} \quad \lambda_{A,s} = -0.45 \quad .$$

then from (3.13) α is

$$\alpha = \frac{2}{2 + 0.45 - 0.95} = \frac{4}{3} = 1.3333$$

The new eigenvalues

$$\lambda_{B,b} = -\lambda_{B,s} = (1 - \frac{4}{3}) + \frac{4}{3} \cdot 0.95 = 0.9333$$

reduce the biggest eigenvalue from 0.95 to 0.93333, a very small improvement.

This method of diagonal matrix (3.8) can also be used as a relaxation method for non-convergent iterative processes as well as a convergence promotion technique.

Three examples of non-convergent iterative processes are given in Figure (3.2). In the first one (a) where $\lambda_{A,s} < -1$ and $\lambda_{A,b} < 1$, the iterate oscillates and diverges. Here α as a damping factor, has the range of

$$0 < \alpha < 1$$

and it can be calculated by (3.13) (See the dashed line Figure 3.2a). In the second case (b), $\lambda_{A,s} > 1$ and $\lambda_{A,b} > 1$, so that the iterate diverges (blowup). The coefficient α, which in this case is a negative factor, can be calculated also by (3.13).

In the last example (c), when $\lambda_{A,b} > 1$ and $\lambda_{A,s} < 1$, there is no value of α which can transfer the non-convergent iterative process to a convergent one, as seen in Figure (3.2c).

Consider again equation (3.12); although this condition gives the smallest eigenvalue in modulus, the negative eigenvalue $\lambda_{B,s}$ may cause undesirable oscillation. Therefore it may be better that $\lambda_{B,s}$ would be smaller than $\lambda_{B,b}$ in modulus, that is, that

$$|\lambda_{B,s}| < \lambda_{B,b}$$

and condition (3.12) may be replaced with

$$-\lambda_{B,s} = \theta . \lambda_{B,b} \tag{3.14}$$

where

$$0 \leq \theta \leq 1$$

From this condition (3.14), and equation (3.11) α becomes

$$\alpha = \frac{1 + \theta}{1 + \theta - \lambda_{A,s} - \theta . \lambda_{A,b}} \tag{3.15}$$

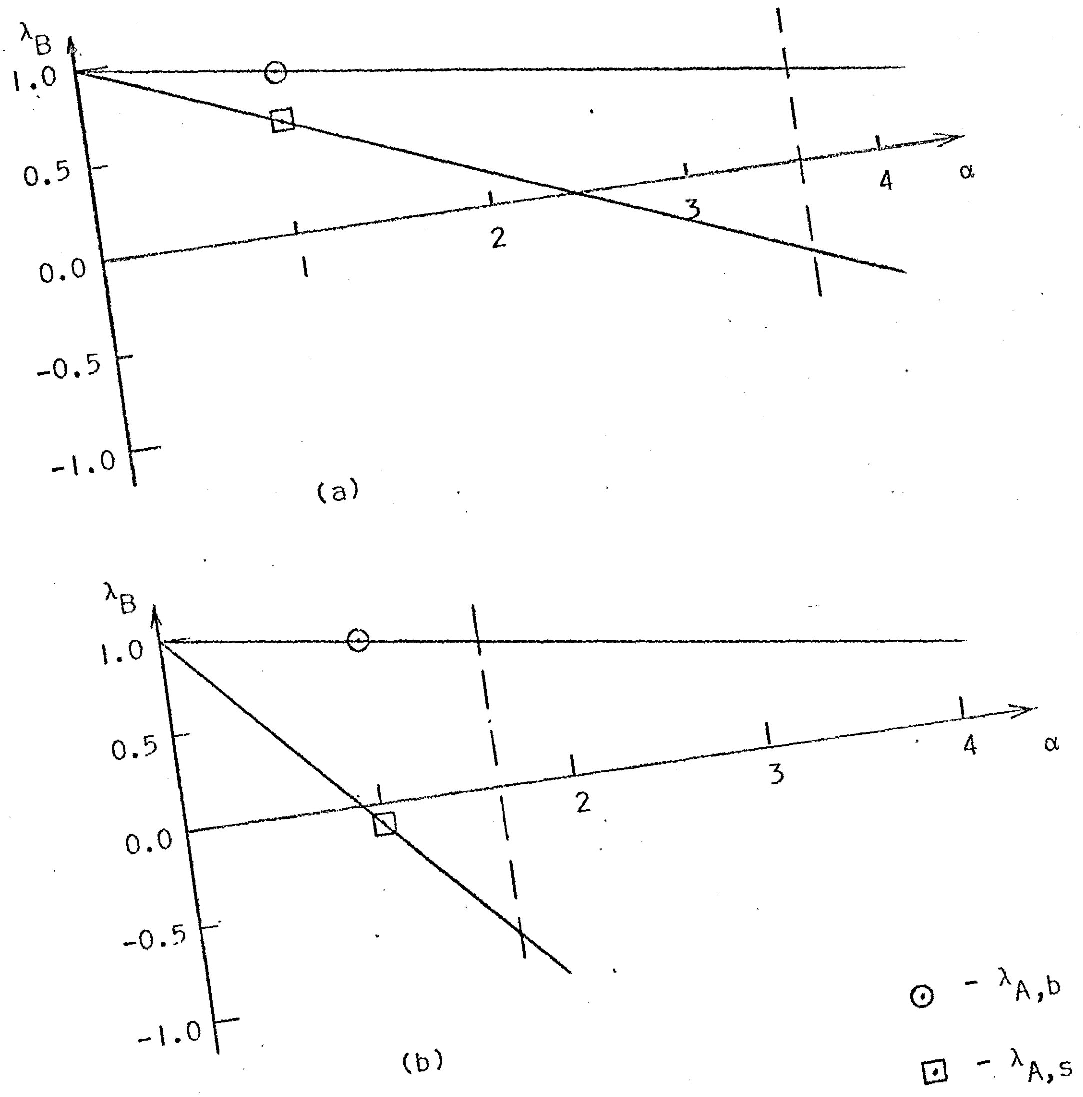

FIGURE 3.1 Variation of the Biggest and Smallest Eigenvalue With the Coefficient α.

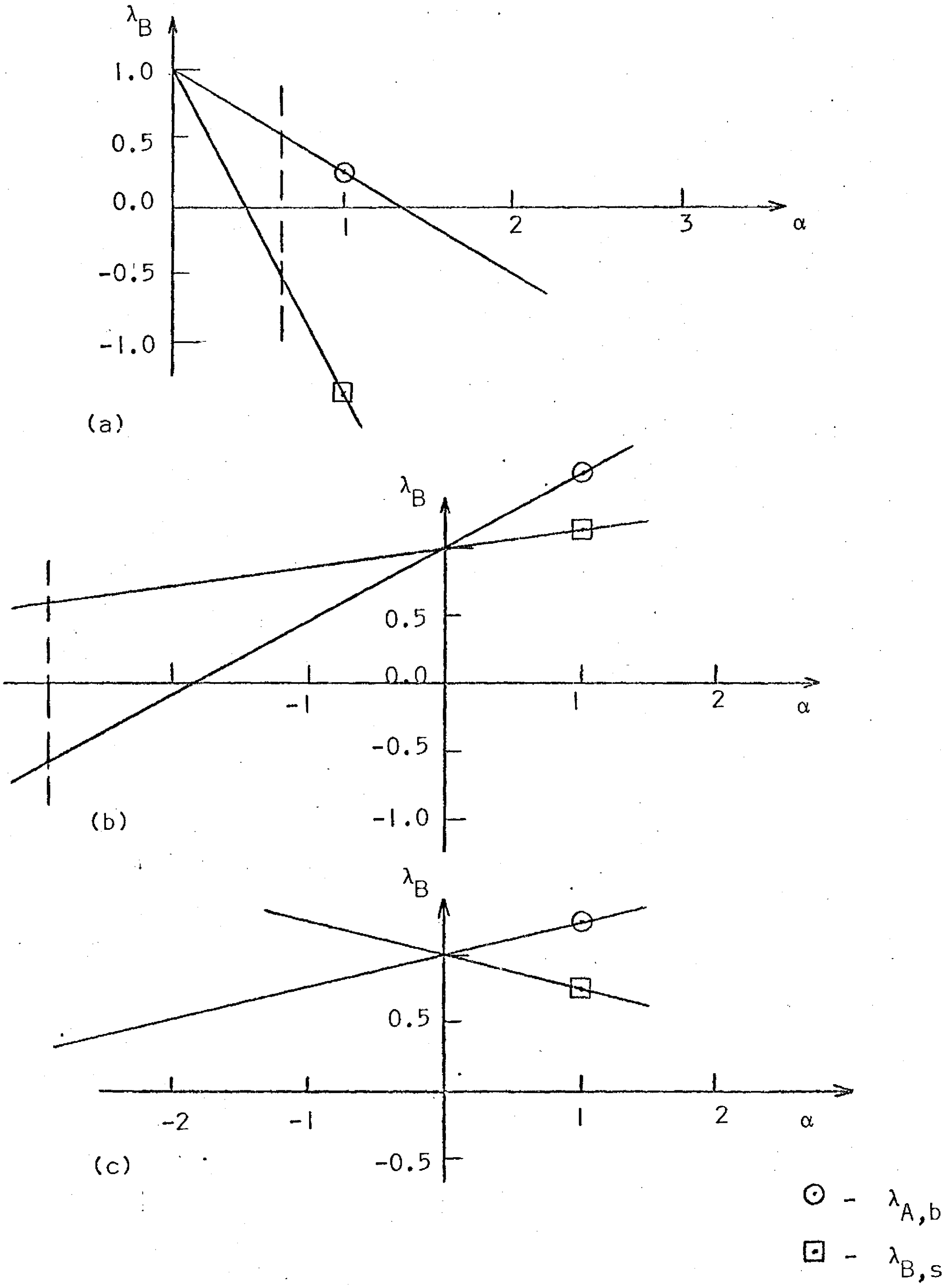

FIGURE 3.2 Variation of the Biggest and the Smallest Eigenvalue With the Coefficient α

Finally, it is worthwhile mentioning that the relaxation factor t, equation (3.1), and the coefficient α play the same role, and it is possible to use the relaxation factor t instead of α in the development above, where matrix G is considered as the unit matrix I. In addition it is important to emphasize that this diagonal method may be used as a supplemental method to the full matrix of coefficient techniques where the factor t is usually chosen smaller than unity for relaxation.

(b)
$$\bar{\bar{G}} = [I \; \bar{\alpha}] \qquad (3.16)$$
$$(t = 1)$$

where $\bar{\alpha}^T = (\alpha_1, \alpha_2, \alpha_3 \ldots \alpha_m)$

Substitute (3.16) to (3.4) to get

$$\bar{\bar{B}} = I + [I\bar{\alpha}](\bar{\bar{A}} - I) \qquad (3.17)$$

Here again, the objective is to find vector $\bar{\alpha}$ such that the biggest eigenvalue in modulus of matrix B will be as small as possible, that is

$$\min_{\bar{\alpha}} \{ \max_{j} | \lambda_j | \} \qquad (3.18)$$

where λ_j are the eigenvalues of matrix B.

This multi-variable minimization procedure (3.18) will become too expensive in computer time when the number of variables m is bigger than about three. Practically it is not a useful method for evaluating the set of coefficients $\bar{\alpha}$. However, under particular conditions the set of coefficients $\bar{\alpha}$ can be found, when the objective is to minimize the norm of matrix B (2.35) instead of using (3.18), that is

$$\min_{\bar{\alpha}} \{||\bar{\bar{B}}||_\infty\} = \min_{\bar{\alpha}} \{ \max_{i} \sum_{j=1}^{m} |b_{ij}|\} \qquad (3.19)$$

where b_{ij} is the element of matrix $\bar{\bar{B}}$. The rate of convergence may be increased as the norm of B is decreased (2.35 and 2.22).

The particular condition under which set $\bar{\alpha}$ can be found is that matrix A should satisfy the following condition,

$$|1 - a_{ii}| > \sum_{j\neq i} |a_{ij}| \qquad (3.20)$$

$$i = 1, 2, 3, \ldots m$$

Thus, the absolute row sum of matrix B is dependent on the coefficient α (3.17), and becomes

$$R_i(\alpha_i) = \sum_{j=1}^{m} |b_{ij}| = |1-\alpha_i(1-a_{ii})|+|\alpha_i| \sum_{j\neq i} |a_{ij}| \qquad (3.21)$$

$$i = 1, 2, 3, \ldots m.$$

The first term of equation (3.21) has a minimum value of zero at

$$\alpha_i^* = \frac{1}{1 - a_{ii}} \qquad (a_{ii} \neq 1) \qquad 3.22)$$

Condition (3.20) guarantees that any decrease of $|\alpha_i|$ from $|\alpha_i^*|$ will increase the first term faster than the decrease of the second term. Clearly any increase of $|\alpha_i|$ increases both terms. The minimum value of $R_i(\alpha_i)$ is given by

$$R_i(\alpha_i^*) = \frac{1}{|1 - a_{ii}|} \sum_{j=i} |a_{ij}| \qquad (3.23)$$

$$i = 1, 2, 3, \ldots m.$$

Finally, although the conditions (3.20) seem severe they are frequently satisfied in practice.

3.2 Geometric Extrapolation

The solution of iterative calculation (2.7) can be well approximated by extrapolation if the iterate has approached a geometric progression. This technique is particularly suited to cases where the geometric progression is obtained after a relatively small number of iterations.

Let it be assumed that the number of iteration n is sufficiently large so that the iterate has approached a geometric progression and satisfied (2.57). Therefore the iterate can be expressed by equation (2.50), which is

$$\bar{X}_n = \bar{U}\,\mu^n + \bar{X}'_s$$

This equation (2.50) can be written for any component i as

$$x_{n,i} = U_i\,\mu^n + x'_{s,i} \qquad (3.25)$$

$$i = 1, 2, 3, \ldots m.$$

The approximate solution of component i, $x'_{s,i}$, can be evaluated from the last three iteration points (x_{n-2}, x_{n-1}, x_n), by eliminating U_i and μ to get

$$x'_{s,i} = \left\{\frac{x_n . x_{n-2} - x_{n-1}^2}{x_n - 2.x_{n-1} + x_{n-2}}\right\}_i \qquad (3.26)$$

$$i = 1, 2, 3, \ldots m.$$

Proceeding from the evaluation of the new point $\bar{X}'_s$, the iteration is reapplied with the point $\bar{X}'_s$ as initial value. The iteration will proceed until the iterate again approaches a geometric progression.

Any time that the iterate approaches a geometric progression the solution can be approximated by (3.26) and thereby accelerate the convergence. Thus, the difference between this method and other convergence promotion techniques is that equation (3.26) is not a part of iteration procedure, but should only be used repeatedly as an extrapolation method when the iterate approaches a geometric progression.

Furthermore, the geometric extrapolation method (3.26) can be used as a supplementary acceleration method to the convergence promotion technique (3.1) with a fixed matrix G.

Aitkins δ^2 process (A3, A4), which was developed specifically for solving characteristic value problems iteratively, yields the same expression as equation (3.26).

Equation (3.26) may be transferred to the following form as

$$x'_{s,i} = x_{n-1,i} + \alpha_i (x_{n,i} - x_{n-1,i}) \qquad (3.27)$$

$$i = 1,\ 2,\ 3,\ \ldots\ m.$$

where

$$\alpha_i = \left(\frac{x_{n-2} - x_{n-1}}{x_n - 2.x_{n-1} + x_{n-2}} \right)_i \qquad (3.28)$$

In order to prevent oscillation, a relaxation factor t may be added to equation (3.27) as

$$x'_{s,i} = x_{n-1,i} + t.\alpha_i.(x_{n,i} - x_{n-1,i}) \qquad (3.29)$$

$$i = 1,\ 2,\ 3,\ \ldots\ m.$$

Equation (3.29) may be written in vector notation as

$$\bar{X}'_s = \bar{X}_{n-1} + t.[I\bar{\alpha}](\bar{X}_n - \bar{X}_{n-1}) \qquad (3.30)$$

where

$$\bar{\alpha}^T = (\alpha_1, \alpha_2, \dots \alpha_m)$$

Equation (3.30) is in the form of equation (3.1) with matrix G as (3.16) but as it has been said before equation (3.30) is not part of the iteration and should only be applied when the iterate approaches a geometric progression.

Now, examining the coefficient α_i (3.28) for any element i

$$\alpha = \frac{x_{n-2} - x_{n-1}}{x_n - 2x_{n-1} + x_{n-2}} = \frac{1}{1 - \dfrac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}}$$

Using equation (3.25) it can easily be shown that

$$\frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}} = \mu \tag{3.31}$$

and

$$\alpha = \frac{1}{1 - \mu} \tag{3.32}$$

that is, that all the coefficients α_i are equal,

$$\alpha_1 = \alpha_2 = \alpha_3 \dots = \alpha_m$$

and if $-1 < \mu < 1$

then $$0 < \frac{1}{\alpha} = 1 - \mu < 2$$

and $$\frac{1}{2} < \alpha \tag{3.33}$$

Now, combining (3.32) and (3.30) the solution approximating $\bar{X}'_s$ becomes

$$\bar{X}'_s = \bar{X}_{n-1} + \frac{t}{1-\mu}(\bar{X}_n - \bar{X}_{n-1}) \qquad (3.34)$$

Liusternik's (F1) technique for improving the convergence of the iterative process for solving systems of linear equations yields a similar expression to (3.34).

Summarizing the geometric extrapolation technique, we may say that if the iterate satisfies (2.57), μ may be evaluated by (2.57) and equation (3.34) can be used for calculating the approximate solution $\bar{X}'_s$.

However, when the geometric extrapolation technique is applied for practical cases of chemical recycle problems, it is apparent that it may be preferable to use equation (3.30) for calculating the approximate solution $\bar{X}'_s$ where α_i is calculated by (3.28) for every component.

In practice, we may have a recycle problem where the components of vector $\bar{X}$ do not have full interaction among themselves or the components form a few groups, each of which has a different geometric coefficient μ. Truly, if the components of vector $\bar{X}$ can be divided into groups where there is no interaction between them, then the problem can be solved separately for every group of components. When working with an executive program like PACER it is not feasible to separate the calculation and it may be cheaper to solve all the components together. Therefore, α_i ought to be calculated for every component by (3.28).

We may conclude that the above technique seems to be very useful and simple to operate but it should be noted that from the point of view of the total calculation time it performs best in cases where the geometric progressive stage is achieved in small number of iterations.

3.3 Multidimensional Extrapolation

The convergence of iterative calculation (2.7) may be promoted by fitting a set of linear equations (2.32) to the last $m + 2$ point vectors $\bar{X}$, and then approximating the solution by (2.39). Consider $m + 1$ iterations of equation (2.7)

$$\bar{X}_{\nu+1} = \bar{F}(\bar{X}_{\nu}) \tag{3.36}$$

$$\nu = 1,\ 2,\ 3,\ \ldots\ m+1.$$

where $\bar{X}_1$ is some initial point. The (m+2) vectors of $\bar{X}$ that result from this iteration can be used to calculate the linear coefficients of the linear iteration (2.32) as

$$\bar{X}_{\nu+1} = \bar{\bar{A}}\,\bar{X}_{\nu} + \bar{b} \tag{3.37}$$

$$\nu = 1,\ 2,\ 3,\ \ldots\ m+1.$$

By arranging the $m+2$ vectors of $\bar{X}$ in partitioned matrices, equation (3.37) may be expanded to the following form,

$$\left[\begin{array}{c|c|c|c} \bar{X}_2 & \bar{X}_3 & \cdots & \bar{X}_{m+2} \end{array}\right] = \left[\begin{array}{c|c} \bar{\bar{A}} & \bar{b} \end{array}\right] \cdot \left[\begin{array}{c|c|c|c} \bar{X}_1 & \bar{X}_2 & \cdots & \bar{X}_{m+1} \\ \hline 1 & 1 & & 1 \end{array}\right] \tag{3.38}$$

or

$$\bar{\bar{U}} = \bar{\bar{C}} \cdot \bar{\bar{V}} \tag{3.39}$$

Postmultiplying of (3.39) by $\bar{\bar{V}}^{-1}$ we get

$$\bar{\bar{C}} = \bar{\bar{U}}\,\bar{\bar{V}}^{-1} \,. \tag{3.40}$$

(for more details see Appendix B, the second method of linearization).

After $\bar{\bar{C}}$ has been calculated, the solution can be approximated by (2.39) using matrix $\bar{\bar{A}}$ and vector $\bar{b}$,

$$\bar{X}'_s = (\bar{\bar{I}} - \bar{\bar{A}})^{-1} \bar{b} \qquad (3.41)$$

The approximate solution (3.41) may be used as an initial point $\bar{X}_1$ for a new set of m+1 iterations.

This technique is well suitable for cases where the number of elements, m, in vector $\bar{X}$, is relatively small, as m+1 iterations are necessary for collecting data for matrices $\bar{\bar{U}}$ and $\bar{\bar{V}}$. In addition, the computation time for the inversion of matrix $\bar{\bar{V}}$, needs to be taken into consideration if m is large since the size of matrix $\bar{\bar{V}}$ is (m+1) x (m+1).

3.4 Partition Recycle

In an executive program such as PACER, all the recycles in a block are solved simultaneously. Thus, the iteration takes place on all the assumed streams at once. This kind of iteration is represented by equation (2.7) and for linear processes, by equation (2.32).

Now, it may be worthwhile to perform a partition of the recycle process which has more than one recycle, into two groups: the main group and the secondary group of recycles, such that, for every iteration of the main recycle, the secondary one may be iterated a few times or may be even converged.

For illustration we may examine Figure 3.3 which is a second order recycle problem. To solve this recycle problem as one group, we may choose streams S7 and S4 as assumed streams and the related sequence of unit calculations will be

$$4, 1, 2, 3 \ldots..$$

This recycle problem (Figure 3.3) may also be solved by partitioning the problem into two groups; the recycle around the nodes 1 and 2 is the secondary iteration and the nodes 4 and 3, together with the secondary iterations are the main group. The sequence will be as follows

$$4, (1, 2)_k, 3 \ldots.$$

That is, that the secondary recycle iterates k times for every iteration of the main group. The number of iterations, k, of the secondary recycle can be chosen as a fixed number or the secondary recycle may be iterated until convergence is obtained. The secondary iteration can also be chosen as the recycle around nodes 3 and 4. Thus, any recycle in the process can be chosen as secondary, but obviously we wish the secondary recycle to have a small number of equipments (nodes) or, more precisely, the calculation time of the secondary needs to be small compared to that of the main iteration.

A method for analyzing multi-recycle problems, that can be represented by linear equation (2.32), is given in Appendix D. This method can be used to evaluate the eigenvalues of the process which includes secondary iteration. The number of iterations k of the secondary recycle can be a finite number or if the secondary iteration is converged, k can be considered as $k \to \infty$. It appears that the calculation of the eigenvalues for the case that $k \to \infty$ is relatively simple, and by calculating the biggest eigenvalue in modulus for this case it is possible to evaluate if there is any benefit from the subiteration of the secondary recycle.

Any recycle in the process can be chosen as secondary one and the biggest eigenvalue in modulus may be calculated. Obviously, the secondary recycle which reduces the biggest eigenvalue in modulus the most, is the best one to choose. But, for gaining a reduction in the overall calculation time it is also important that the calculation time of the secondary recycle will be small compared to the overall calculation time.

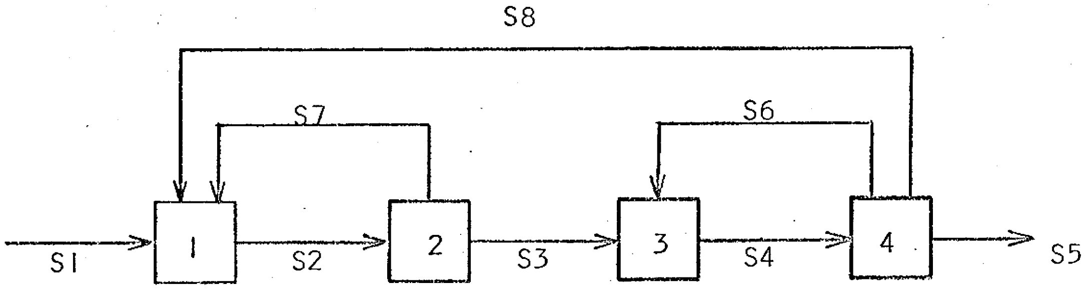

FIGURE 3.3 Two Recycle Problems

4. RESIDUAL APPROACH

4.1 Introduction

Thus far, we have been concerned with the convergence of iterative calculations in the form of equations (2.5) and (2.6) which have been represented by equation (2.7): Convergence promotion techniques have been applied in order to accelerate the convergence of the iterative calculation. The general form of the convergence promotion is given by equation (3.1) which replaced equation (2.6). Equations (2.5) and (3.1) represent the iteration with convergence promotion where matrix G, the matrix of convergence promotion coefficients, may be chosen beforehand as a diagonal matrix (3.8) or (3.16) or as a full matrix (3.6). Undoubtedly the full matrix technique (3.6), when applied, gives a more powerful method which can cover a wider range of cases. The full matrix technique, compared to diagonal matrix in a case of a truly linear iterative process, gives the solution in one step. We may say that the more non-linear the process is, the more sophisticated ought to be the methods applied. Logically the next step is to deal with methods which would recalculate matrix G every iteration. Although these techniques would be more powerful in solving recycle processes, and may reduce the number of iterations they would consume more computer time and storage.

Actually we discussed previously two methods which do not use a fixed value of matrix G. The first one was the geometric extrapolation technique where the convergence promotion coefficient α_i (3.28) was recalculated every time the iterate approached a geometric progression. The second one, the multi-dimensional extrapolation technique, approximates the solution every m+l iterations by linearizing the process repeatedly.

The Quasi-Newton method (R4, ZI) has been found very attractive for redetermining the matrix G. This method has been developed primarily for solving sets of non-linear algebraic equations.

Before discussing the Quasi-Newton method we may reconstruct the problem to represent the recycle process, or any iterative calculation, as a set of algebraic equations. This form will enable us to apply the Quasi-Newton method more smoothly and might give us a new outlook on the problem.

4.2 Formulation

Rubin (R3) has considered the error vector (2.52) as a set of residual equations as

$$\bar{f}(\bar{X}) = \bar{Y} - \bar{X} = \bar{F}(\bar{X}) - \bar{X} = \bar{0} \qquad (4.1)$$

Rubin has used the "generalized false position" method, for solving this set of equations (4.1). As the components in vector $\bar{X}$ or $\bar{Y}$ may not be of the same magnitude, working with the absolute difference, as in equation (4.1), can create difficulties in the solution. Hence, it will be more practical to normalize the set of equations to give the fixed-point test vector (2.28). Therefore the iterative calculation can be represented in form of set of algebraic equations as follows:

$$f_1(x_1, x_2, \ldots x_m) = (y_1 - x_1)/x_1 = 0$$

$$f_2(x_1, x_2, \ldots x_m) = (y_2 - x_2)/x_2 = 0 \qquad (4.2)$$

$$\vdots$$

$$f_m(x_1, x_2, \ldots x_m) = (y_m - x_m)/x_m = 0$$

where from equation (2.1)

$$y_i = F_i(x_1, x_2, \ldots x_m)$$

$$i = 1, 2, 3 \ldots m$$

And in vector notation the set of equations gives,

$$\bar{f}(\bar{X}) = \bar{T} = [I\ \bar{X}]^{-1}(\bar{Y} - \bar{X}) = \bar{0} \qquad (4.3)$$

and by substituting equation (2.2) into (4.3) we get

$$\bar{f}(\bar{X}) = \bar{T} = [I\ \bar{X}]^{-1}(\bar{F}(\bar{X}) - \bar{X}) = \bar{0} \qquad (4.4)$$

The aim of any calculation technique is to find vector $\bar{X}$ which satisfies equation (4.4). It may be assumed that at least one real solution exists and that the functions (4.4) are continuous and possess continuous first derivatives. Now, the functions themselves are often long and expensive to evaluate, since any evaluation of the functions is equivalent to one iteration around the recycles, and the value of the functional derivatives can only be obtained by finite difference approximation. Therefore, solution methods which keep the number of functional evaluations to a minimum become very attractive.

Practically, instead of satisfying the set of equation (4.4) exactly, it may be sufficient to find a vector $\bar{X}$ which is close enough to solution and reduces the norm of the functions below a certain tolerance. The norm may be chosen as (2.10), thus

$$\Phi = ||\bar{f}(\bar{X})||_{\infty} \qquad (4.5)$$

or one of the norms (2.11) or (2.12).

4.3 Solution Techniques

The most widely used method for solving the set of functions (4.4) is the Newton-Raphson technique [11]. The method can be defined by two equations, as

$$\bar{X}_{n+1} = \bar{X}_n + \bar{P}_n \qquad (4.6)$$

where $$\bar{P}_n = -(\bar{\bar{J}}_n)^{-1}\bar{f}_n \qquad (4.7)$$

for $$\det(J_n) \neq 0$$

The Jacobian $\bar{\bar{J}}_n$ is the matrix of partial derivatives of $\bar{f}(\bar{X})$ evaluated at point $\bar{X}_n$, $\bar{f}_n$ is the vector of functions evaluated at $\bar{X}_n$ ($\bar{f}_n = \bar{f}(\bar{X}_n)$), and $\bar{P}_n$ is the correction to be applied to $\bar{X}_n$ such that the value of $\bar{X}_{n+1}$ will be closer to the solution of the equations (4.4). The disadvantage of the Newton-Raphson method is that for every iteration the Jacobian ought to be evaluated and its inverse ought to be calculated. Evaluating the Jacobian numerically by finite difference means evaluating the functions (4.4) m+1 times.

Before continuing to the next method for solving the set of functions (4.4) it is worthwhile to show that the successive substitution technique (2.5) and (2.6) can be achieved by choosing

$$\bar{P}_n = [I\ \bar{X}_n]\ \bar{f}_n \tag{4.8}$$

then

$$\bar{P}_n = [I\ \bar{X}_n][I\ \bar{X}_n]^{-1}(\bar{Y}_n - \bar{X}_n) = \bar{Y}_n - \bar{X}_n$$

and with equation (4.6)

$$\bar{X}_{n+1} = \bar{X}_n + \bar{Y}_n - \bar{X}_n = \bar{Y}_n$$

as equation (2.6).

In order to reduce the number of function-evaluations a modified form of Newton's method has been introduced. Lohr and Rall (L3) used Kantorovich theorems to compare the overall efficiency of Newton's method and its modified form. They have given criteria for choosing the method of achieving a solution in minimum time.

The modified form of Newton's method can be described by the following two equations,

$$\bar{X}_{n+1} = \bar{X}_n + t.\bar{P}_n \tag{4.9}$$

$$\bar{P}_n = -(\bar{\bar{J}}_o)^{-1}\ \bar{f}_n \tag{4.10}$$

for $\det(\bar{\bar{J}}_o) \neq 0$

The Jacobian matrix $\bar{\bar{J}}_o$ is evaluated only once in point $\bar{X}_o$ and used unchanged throughout all the calculation. As $\bar{P}_n$ would be a poor estimate of the true correction, relaxation factor t may be used in order to prevent oscillation, where $0 \leq t \leq 1$.

In a case where the initial point $\bar{X}_o$ is far away from the solution $\bar{X}_s$, the Jacobian may be evaluated a few times as the iterate (4.9) progress toward the solution. The sum of equations (4.9) and (4.10) gives an iterative equation in form of equation (2.7); hence, if the convergence is slow, the geometric extrapolation technique for accelerating the convergence may be applied.

Actually, if the first estimate of the solution $\bar{X}_o$ is sufficiently close to $\bar{X}_s$ the modified form of Newton's method gives good convergence in a calculation of heat and material balance. The technique will be demonstrated in detail in Section 6.4.

4.4 Quasi-Newton Method

Recently the Quasi-Newton iterative schemes (R4, B1, B2, Z1) have been proposed. Unlike the Newton's method they do not require the Jacobian, but do use an approximate Jacobian (or its inverse). The method can be defined by the following equations

$$\bar{X}_{n+1} = \bar{X}_n + t_n.\bar{P}_n \tag{4.11}$$

$$\bar{P}_n = \bar{\bar{H}}_n \bar{f}_n \tag{4.12}$$

where $\bar{\bar{H}}_n$ is an approximation to the Jacobian inverse. At each stage of the calculation the approximate Jacobian inverse is recalculated until, at convergence, it hopefully becomes the Jacobian inverse corresponding to the solution of the system of equations.

The approximation to the Jacobian inverse is improved by the following procedure:

$$\bar{\bar{H}}_{n+1} = \bar{\bar{H}}_n - \frac{(t_n\bar{P}_n + \bar{\bar{H}}_n\Delta\bar{f}_n)\ \bar{Z}_n^T\ \bar{\bar{H}}_n}{\bar{Z}_n^T\ \bar{\bar{H}}_n\ \Delta\bar{f}_n} \qquad (4.13)$$

where $\Delta\bar{f}_n = \bar{f}_{n+1} - \bar{f}_n$

The vector $\bar{Z}$ has been selected differently in the various articles mentioned. Barnes (B1) selects $\bar{Z}$ as the direction which is orthogonal to the previous m-1 directions of the correction vector $\bar{P}$. Thus,

$$\bar{Z}_n^T\ \bar{P}_k = 0 \qquad (4.14)$$

for k = n-1, n-2, n-3, ..., n-(m-1).

The reason for this choice is that if the full step $\bar{P}_n$ is taken at each iteration (t_n=1) then for a linear system of equations the process will converge after m+2 functional evaluations. Broyden (B2) has selected $\bar{Z}$ equal to the correction vector $\bar{P}_n$,

$$\bar{Z} = \bar{P}_n \qquad (4.15)$$

Rosen (R4) has applied the last two methods of selecting $\bar{Z}$, Barnes's (4.14) and Broyden's (4.15) and in addition he has calculated $\bar{Z}$ by averaging, component by component, the $\bar{Z}$ calculated by both methods. Rosen claimed that the latter procedure gave better results than either procedure alone. It is worthwhile mentioning that equation (4.14) means solving a set of m-1 linear equations at every iteration.

The scalar t_n in equation (4.11) is not only a damping factor as in the modified Newton's method (4.9), but can be used as a one-dimensional search parameter. After the direction $\bar{P}_n$ is determined, a search along this direction may be carried out to find a point which has a smaller value of some norm of the functions f_i, say Φ, where

$$\Phi = ||\bar{f}_{n+1}||_2 \quad .$$

or

$$\Phi = ||\bar{f}_{n+1}||_1$$

and

$$\min_{t_n} \Phi = \min_{t_n} \{||\bar{f}(\bar{X}_n + t_n.\bar{P})||_2\} \qquad (4.16)$$

This one-dimensional search is crucial to the performance of the algorithm and may be carried out in a number of ways. Whether or not to attempt to minimize the norm or simply find a point which reduces the norm, is to be decided by the user. The main aim is to find the solution with a minimum total number of functional evaluations. Rosen $^{(R4)}$ suggested a continuing study of this problem and recommended applying norm minimization in the initial stages of the calculation followed by norm reduction later as the calculation proceeds.

The initial value of the matrix $\bar{\bar{H}}_o$ has also been selected differently in the various articles mentioned. The initial matrix $\bar{\bar{H}}_o$ can be chosen as the unit matrix I or as the Jacobian inverse evaluated at the initial point $\bar{X}_o$,

$$\bar{\bar{H}}_o = -(\bar{\bar{J}}_o)^{-1} \qquad (4.17)$$

It is strongly recommended to select the matrix $\bar{\bar{H}}_o$ as per (4.17) when heat and material balance calculations are involved. However, m+1 function evaluations are necessary.

Further study of the Quasi-Newton method is recommended. Possibly a technique could be formulated whereby the Quasi-Newton method is used in conjunction with the modified form of Newton's method. That is, the matrix $\bar{\bar{H}}$ would not be recalculated (by 4.13) for each iteration. This would reduce computation time, and may increase numerical stability.

5. PROCESSES STUDIED

Two case studies of process simulation are presented in this section. First, in the simulation of the alkylation unit the convergence was accelerated by the full matrix technique (Section 3.1.1). In the second example, the hydrate wash section of the Bayer process for alumina extraction, the convergence was promoted by the geometric extrapolation technique (Section 3.2).

5.1 Alkylation Plant Simulation

During the 1967/68 academic session a simulation of an alkylation unit was carried out as fourth-year student project. The 4th year students were divided into groups, each group supervised by a faculty member assisted by graduate students. Each group was designated to study a part of the process and prepare mathematical modules for the equipment in their section. Finally all the modules were combined and the simulation of the whole plant was done. As the rate of convergence of the iteration was very slow, convergence promotion was applied.

5.1.1 General Description of the Alkylation Unit (A5, M1, S4)

A general flow diagram of the alkylation unit is shown as Figure 5.1. The alkylation unit is part of the Shell Oakville Refinery which was designed by C.F. Braun and started in 1963. The alkylation unit was originally designed for 1200 bbl/day of alkylate, however, the production of alkylate increased to 1800 bbl/day. The olefins for the alkylation process are produced by catalytic cracking.

The alkylation reaction is carried out in the Stratco Reactor (equipment E-3). The sulphuric acid, which acts as a catalyst, circulates between the acid settler (E-2) and Stratco reactor with the withdrawal of spent acid (stream 9) and the addition of make-up acid taking place between the settler and the reactor (E-1). The Stratco reactor is a horizontal continuous stirred tank, the sulphuric acid to organic volume ratio in the inlet is generally maintained at unity and the organic phase is dispersed in the acid at the inlet by an impeller driven by a 400 Hp motor.

The alkylates are produced from the reaction of olefin with excess of isobutane in the presence of sulphuric acid. The olefins used are mainly propylene and butylene. The isobutane to olefin ratio in the organic feed is usually greater than 4. The temperature of the reacting dispersion is maintained at about 50^{o}F with the heat of reaction being removed by flashing the organic product from the emulsion settler (E-2) in a coil immersed in the dispersion in the reactor (E-5).

The acid emulsion settler (E-2) is a horizontal vessel approximately 50 ft. long and 12 ft in diameter. It operates at about 70 psig and is used to separate the organic dispersion in acid which leaves the reactor as product. The organic layer which leaves the emulsion settler passes through a reducing valve which drops the pressure from 70 psig to approximately 27 psia and then flashes through coils in the Stratco reactor (E-5). The gas-liquid mixture flows to the product separator (E-4) where C_3 and C_4 hydrocarbons are separated as a gas phase and are sent to the depropanizer (E-9). The liquid phase, which contains alkylate, n-butane and isobutane, is sent to the deisobutanizer (E-19). The product separator has an internal heating coil through which the bottoms from the depropanizer pass.

The vapour stream from the product separator passes to a three-stage compressor unit (E-8) with a 1000 Hp motor. The vapour, mainly propane and butane, is compressed finally to 60 psig and 155°F.

Both the vapour and the liquid streams from the product separator may contain entrained and dissolved sulphuric acid. Therefore, wash units (E-12, E-21), water settlers (E-13, E-22) and coalescers (E-14, E-23) are used for washing and drying the vapour and liquid streams.

After washing, the overhead stream from the product separator is separated in the depropanizer (E-9), into an overhead propane product which is sold as LPG by Shell, and a C_4 bottoms stream which is recycled to the Stratco reactor. The depropanizer has 40 Koch flexitrays.

After being washed the bottom stream from the product separator enters the deisobutanizer (E-19), which has 80 Koch flexitrays. This column separates isobutane and n-butane from alkylate with the isobutane coming off as overhead and n-butane taken off as a side draw. It is important that both the n-butane content in the overhead and the isobutane in the side draw be minimized.

The overhead stream from the deisobutanizer(E-19) is recycled to the Stratco reactor. The side draw is used for gasoline blending, and the bottoms stream is the desired alkylate stream for enhancing the

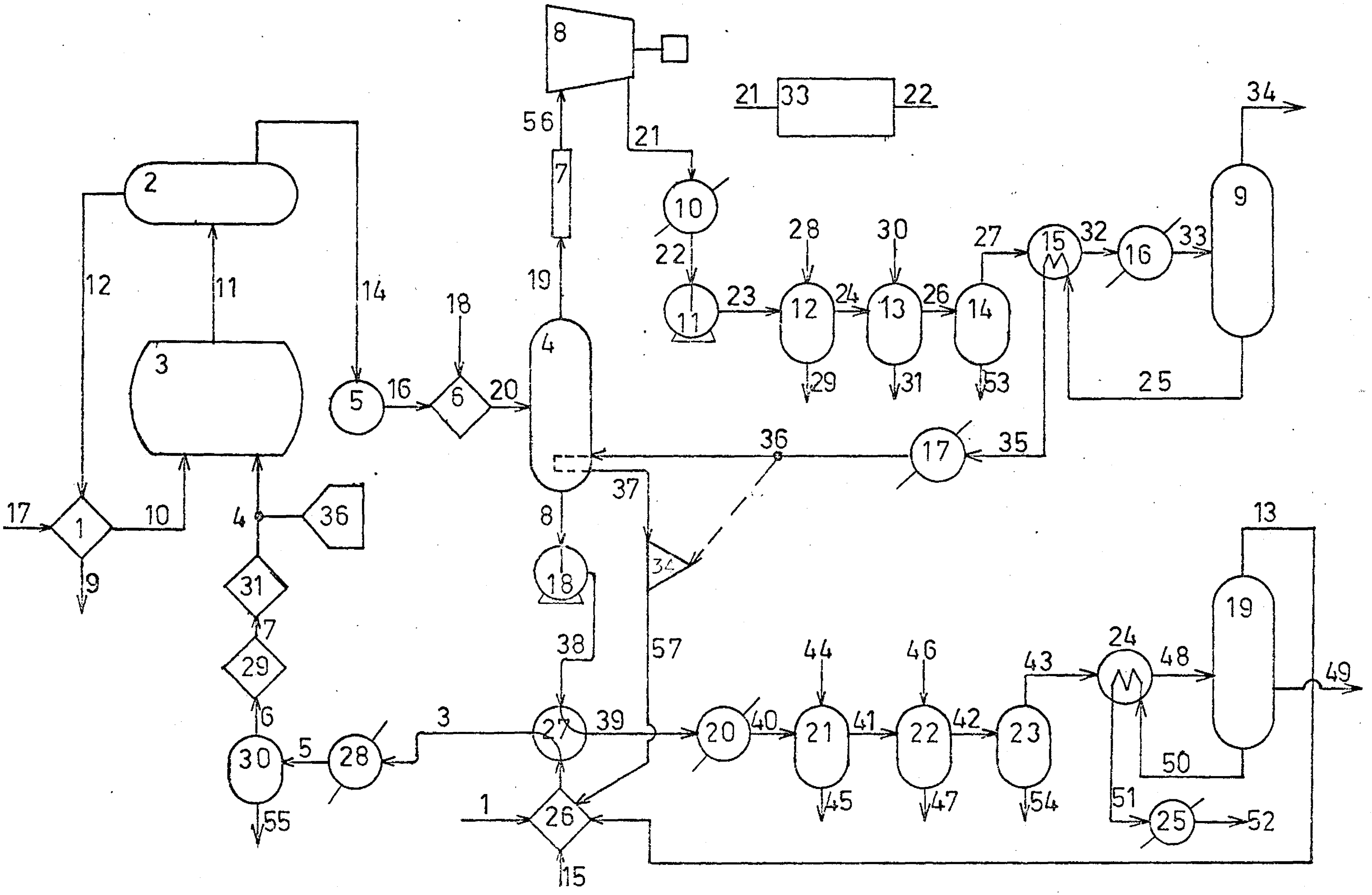

FIGURE 5.1 COMPREHENSIVE PLANT MODEL, SHELL ALKYLATION UNIT

octane number of gasoline.

In this work we will concentrate only on the iteration problem and the convergence promotion technique that was used. More information on the alkylation unit and details of the various equipment modules are given by Shaw [(S4)] and in the final report of the simulation of the alkylation plant [(MI)].

5.1.2 The Assumed Stream and Sequence of Calculation

Before starting the calculation of the alkylation plant the assumed streams have been selected and the related computational sequence of the equipment modules has been found.

Examination of the information flow diagram, Figure 5.1, indicates that it is essentially first order with respect to the flows of components. However if one examines the heat exchange loops in the process, around the product separator and depropanizer (E-4, E-9), and exchanger (E-27), it is apparent that the recycle process has a considerably higher "order" for both heat and mass flow. Since most of the heat exchange loops are controlled it is possible to calculate them directly (See Section 2.1.3). The loop around the depropanizer (E-9) has temperature control (E-16), and the temperatures of the flows from E-27 are also contolled by (E-20) and (E-28).

The only recycle of heat which may interact with the flows of the components is the return flow from the bottom of the depropanizer (E-9) which exchanges heat in the product separator (E-4). This recycle was studied and it was found that the rate of convergence of this recycle was relatively fast. Further, the separation in the product separator (E-4) was not very sensitive to changes in the component flows of stream 36, (the temperature of stream 36 is controlled by (E-17)). Therefore it was decided not to consider it as a "recycle" when the assumed streams were selected, although a good initial value is necessary for each component in stream 36.

The feed stream to the reactor, stream number 4, has been selected as the assumed stream. It should be noted that the acid make up module (E-1) fixes the flow of acid to the reactor (Stream 10) and calculates the acid consumption (Stream 17) and acid withdrawal (Stream 9). Therefore it is not necessary to consider the acid recycle in the iteration and the assumed stream was chosen as stream number 4 instead of stream 11 as might be expected.

Starting the calculation from stream number 4, the assumed stream, the computational sequence of the equipment modules has been chosen as: E-3, E-2, E-1, E-5, E-6, E-4, E-7, E-8, E-10, E-33, E-11, E-12, E-13, E-14, E-15, E-16, E-9, E-15, E-17, E-18, E-27, E-20, E-21, E-22, E-23, E-24, E-19, E-34, E-26, E-27, E-28, E-30, E-29, E-31, E-36.

The heat exchangers (E-15, E-27) are calculated twice for every iteration to ensure the flows of information around all the process in one iteration.

In order to transfer the flow from stream 36 to stream 37, one might attempt to recalculate the product separator (E-4) which in fact ties together the compressor (E-8) and the special control module (E-33). The special control module (E-33) balances the pressure and the flow between the product separator and the compressor. However, it is undesirable to repeat the lengthy calculation of the product separator only for transferring the information from stream 36 to stream 37, where in fact, only the temperature is adjusted. As a result module E-34 was written to pick up flows from stream 36 and the temperature from stream 37, passing the combined information into stream 57. Equipment 36 is the convergence promotion module.

5.1.3 Accelerating the Convergence

In Table 5.1, all the components and their location in the stream vectors (SN) are listed. Only sixteen components from the list of twenty-six are involved in the recycle. These sixteen components from stream 4, the assumed stream, are the variables of the iteration and have been located in vector $\bar{X}$. The initial value $\bar{X}_o$ and the steady state $\bar{X}_s$ for these sixteen components are given in Table 5.2.

Convergence promotion with a full matrix $\bar{\bar{G}}$ (See Section 3.1.1) was used. The process was first linearized around the initial value $\bar{X}_o$ (See Appendix B the 1st method) to give $\bar{\bar{A}}*$ and then the convergence promotion matrix $\bar{\bar{G}}$ was calculated by.equation (3.6) as

$$\bar{\bar{G}} = (I - \bar{\bar{A}}*)^{-1} \tag{5.1}$$

The convergence promotion equation (3.1) has been applied to promote the convergence as

$$\bar{X}_{n+1} = \bar{X}_n + t.\bar{\bar{G}}.(\bar{Y}_n - \bar{X}_n) \tag{5.2}$$

Seventeen iterations have been used for evaluation of matrix $\bar{\bar{A}}*$. It was done by a separate computer run prior to the iteration. The matrix A* was punched as output on computer cards, listed row by row.

TABLE 5.1

Component List for the Alkylation Unit

Component Number	Location in Stream Vector	The Component Flow lb mole/hr
1	6	Water
2	7	Sulphuric Acid
3	8	Sodium Hydroxide
4	9	Sodium Sulphate
5	10	Propyl Sulphate
6	11	C_2 and Lighter
7	12	Propane
8	13	Propene
9	14	n-Butane
10	15	Iso-Butane
11	16	n-Butene
12	17	Iso-Butene
13	18	n-Pentane
14	19	Iso-Pentane
15	20	Iso-Pentene
16	21	C_6
17	22	2,3 - Dimethyl-Butane
18	23	C_7
19	24	2,4 - Dimethyl-Pentane
20	25	2,3 - Dimethyl-Pentane
21	26	C_8
22	27	Iso-Octane
23	28	2,4 - Dimethyl-Hexane
		2,5 - Dimethyl-Hexane
		2,2,3 - Trimethyl-Pentane (Group 1)
24	29	2,3,4 - Trimethyl-Pentane
		2,3,3 - Trimethyl-Pentane
		2,3 - Dimethyl-Hexane (Group 2)
25	30	C_9
26	31	C_{10} and Greater

TABLE 5.2

Initial and Steady-State Values

of Recycle Components in Stream No. 4

No.	Component Number and Name		Initial Value $\bar{X}_s$ lb mole/hr	Steady-State $\bar{X}_s$ lb mole/hr
1	1	Water	0.4282	.3839
2	7	Propane	199.0555	297.6203
3	8	Propene	65.00	65.
4	9	n-Butane	652.3950	612.8264
5	10	Iso-Butane	900.0348	881.5224
6	12	Iso-Butene	120.5.	120.5
7	13	n-Pentane	2.3724	2.3116
8	14	Iso-Pentane	12.4159	11.2492
9	17	2,3 Dimethyl-Butane	1.4286	1.2058
10	19	2,4 Dimethyl-Pentane	0.2664	0.1989
11	20	2,3 Dimethyl-Pentane	0.1843	0.1218
12	22	Iso-Octane	0.0402	0.0297
13	23	Group 1	0.0514	0.0402
14	24	Group 2	0.0169	0.0101
15	25	C_9	0.0155	0.0150
16	26	C_{9+}	0.0187	0.0224

A small computer sub-program was used to calculate matrix $\bar{\bar{G}}$ (5.1) which was also punched on cards. Before starting the iterative calculation of the alkylation unit, matrix $\bar{\bar{G}}$ was read in and stored, row by row, on file.

The results of the iteration, with relaxation factor $t = 0.7$, are given in Tables5.3 and 5.4 and Figures(5.2) and (5.3). To demonstrate the success of this scheme, two sample runs have been made. Table 5.3 shows the flows of propane in stream 4, with and without convergence promotion. These results have been plotted in Figure 5.2 to emphasize the considerable difference in the rates of convergence. Table 5.4 shows the value of the second norm (2.12) of the fixed point test vector (2.28) which is

$$\bar{T}_n = [I \; \bar{X}_n]^{-1} \; (\bar{Y}_n - \bar{X}_n) \qquad (5.3)$$

and the geometric coefficient μ_n which has been calculated as

$$\mu_n = \frac{||\bar{T}_n||_2}{||\bar{T}_{n-1}||_2} \qquad (5.4)$$

for the iteration with and without convergence promotion (2.62) and (2.69). The second norm of the fixed point test vector versus the number of iterations has been plotted in Figure 5.3.

Comparing the rate of convergence of the two sample runs, where the relaxation factor was chosen as $t = 0.7$, using equation (2.77) as

$$R = -\log(\mu_n)$$

then for the run without convergence promotion, $\mu_n \simeq 0.94$ and $R \simeq 0.02687$,

TABLE 5.3

Two Comparison Runs

Without and With Convergence

Promotion

	Propane lb-mole/hr	
Loop	Without Convergence Promotion	With Convergence Promotion (t=0.7)
1	199.05	199.05
2	204.79	242.31
3	210.03	253.09
4	214.83	266.22
5	219.24	273.88
6	223.34	288.01
7	227.17	284.32
8	230.75	287.60
9	234.09	290.04
10	237.22	291.90
11	240.16	293.32
12	242.91	294.41
13	245.50	295.25
14	247.93	295.89
15	250.22	296.38
16	252.38	296.77
17	254.41	297.07
18	256.33	297.30
19	258.14	297.48
20	259.85	297.62 steady-state value
21	261.46	
22	262.99	
23	264.44	
24	265.81	
25	267.10	

TABLE 5.4

Results from the Iteration of the Alkylation Unit

	Iteration Without Convergence Promotion		Iteration With Convergence Promotion (t=0.7)	
Number of Iteration	Norm of the Fixed Point Test Vector $\|\|\bar{T}_n\|\|_2$	Geometric Coefficient μ_n	Norm of the Fixed Point Test Vector $\|\|\bar{T}_n\|\|_2$	Geometric Coefficient μ_n
3	.0359	-	.10107	-
4	.0329	-	.08335	-
5	.0315	.957	.01694	-
6	.0281	.8921	.02579	-
7	.0265	.9431	.01348	.523
8	.0250	.9434	.01131	.839
9	.0235	.940	.00770	.681
10	.0219	.932	.00584	.7584
11	.02056	.9388	.00430	.7363
12	.01927	.9373	.003253	.7565
13	.01807	.9377	.002463	.757
14	.01696	.9386	.001879	.7629
15	.01593	.9393	.001440	.7664
16	.01498	.940	.001106	.7681
17	.01410	.9413	.000853	.7712
18	.01328	.942	.000657	.7702
19	.01253	.9435	.000509	.7747
20	.01182	.9433	.000393	.7720
21	.01117	.945		
22	.01056	.945		
23	.00998	.945		
24	.00945	.947		
25	.00895	.947		

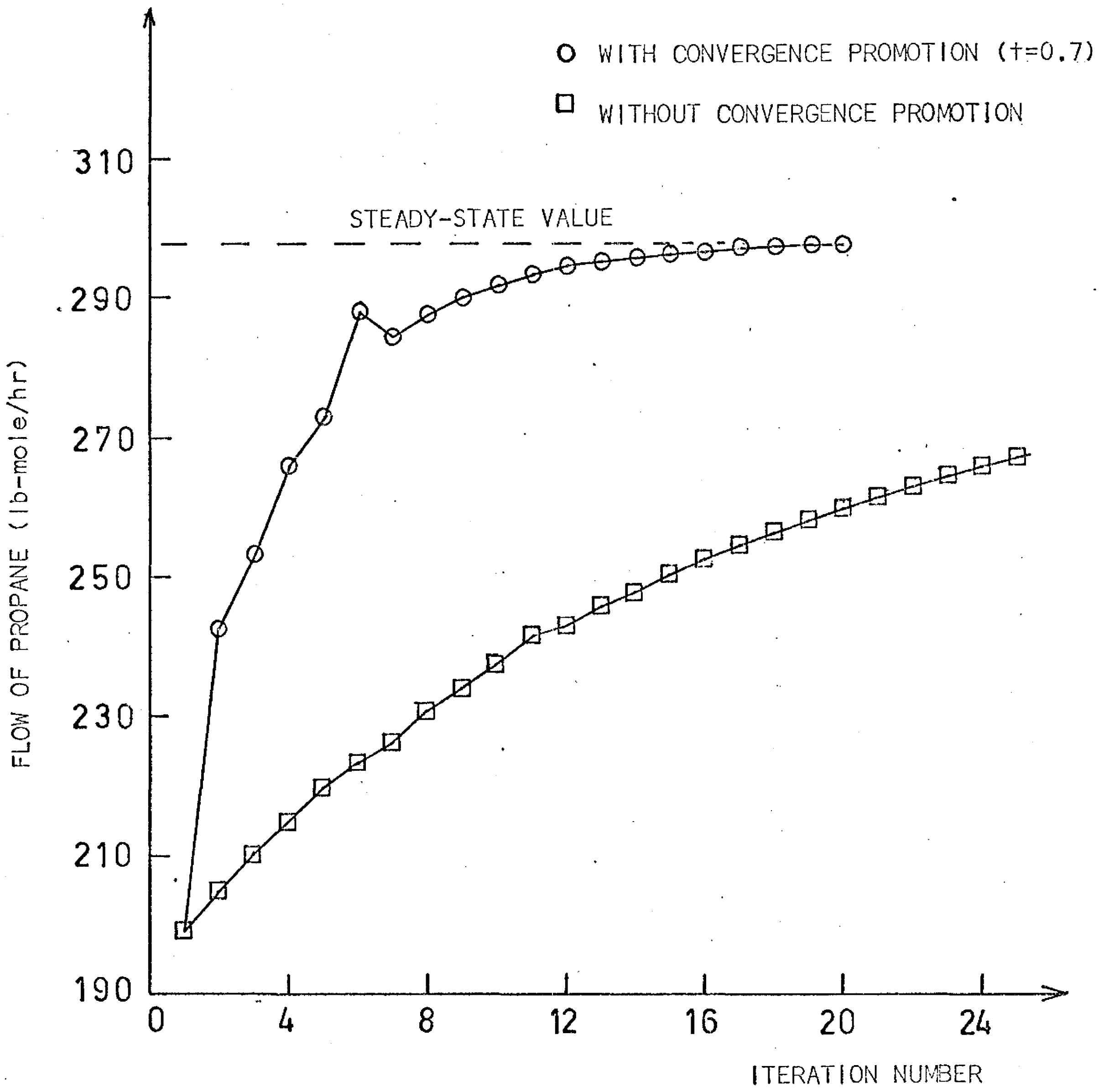

FIGURE 5.2 COMPARISON RUNS - CONVERGENCE OF THE FLOW OF PROPANE

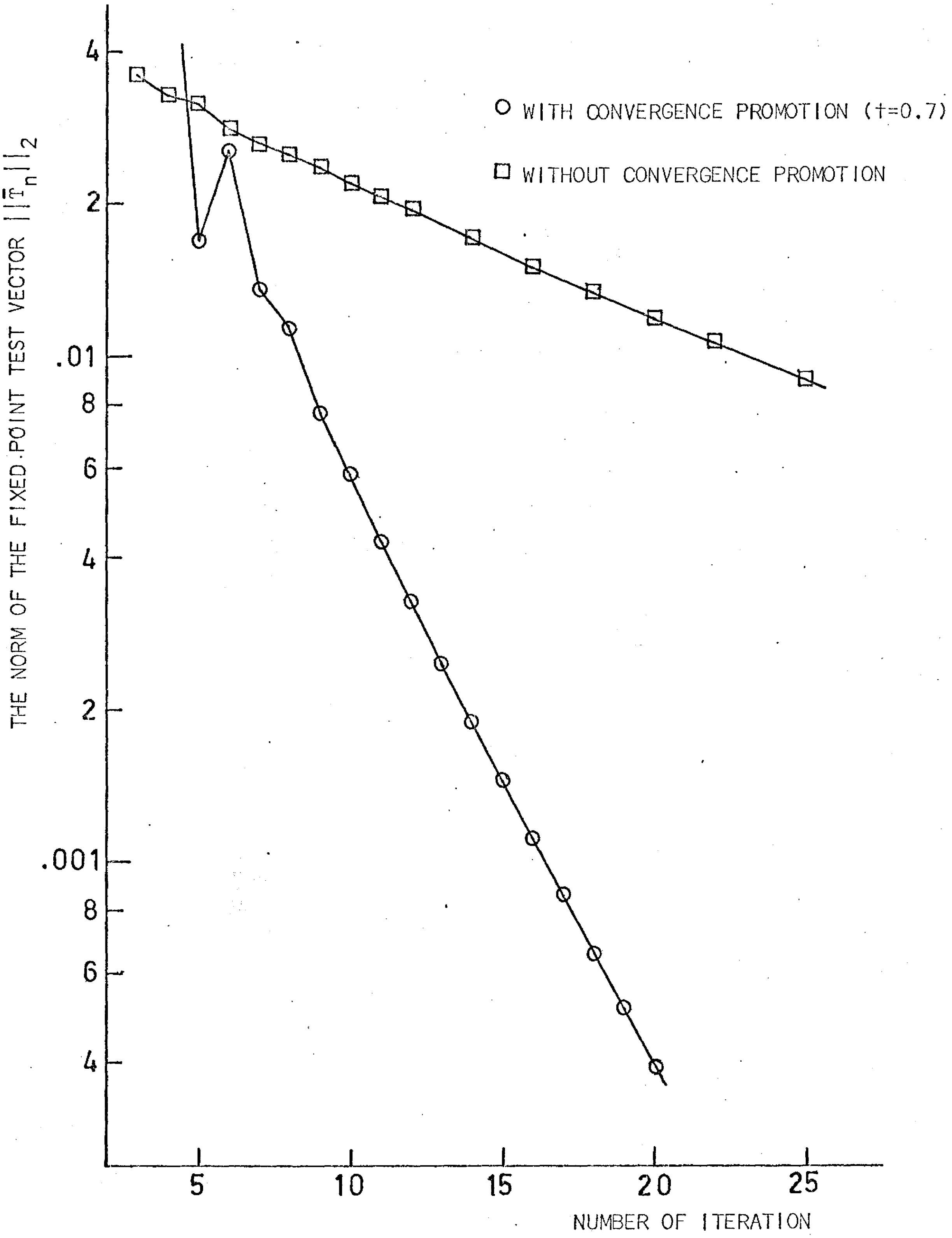

FIGURE 5.3 RESULTS FROM THE ITERATION OF THE ALKYLATION UNIT

and for the run with convergence promotion $\mu_n \simeq 0.77$ and $R \simeq 0.11351$. The rate of convergence when the convergence promotion was applied is approximately 4.25 times faster than the run without convergence promotion $(0.11351/0.02687 \simeq 4.22)$.

5.1.4 The Relaxation Factor t

In the previous section (Section 5.1.3) the relaxation factor t was chosen, prior to the iteration, as 0.7 $(t = 0.7)$. As it was said in Section 3.1.1 the relaxation factor t is applied in order to prevent undesired oscillation. In order to demonstrate the relation between the relaxation factor t and the iteration, four comparison runs were made with different values of relaxation factor as follows: $t = 0.7$, 0.8, 0.9 and 1.0. The results of these four comparison runs are given in Table 5.6 and Figure 5.7.

It has been shown before (Section 5.1.3), where the relaxation factor was 0.7, that the geometric coefficient μ approaches a constant and positive value $(\mu \simeq 0.77)$. Therefore the iteration has approached the solution without oscillation (Figure 5.2).

Now, for the case where the relaxation factor is unity $(t = 1.0)$ the geometric coefficient μ approaches constant negative value as $\mu = -0.81$. Therefore the iterate approaches the solution $\bar{X}_s$ with oscillation (2.50). The flow of propane and its fixed point test in stream number 4, the assumed stream, for the cases where the relaxation factors were chosen as $t = 0.7$ and $t = 1.0$ are given in Table 5.5 and Figures 5.4 and 5.5.

For the purpose of clarifying the case, let us assume for a moment that the above case is a linear iteration and the geometric coefficient can be assumed to be the biggest eigenvalue in modulus (2.61). Using the analysis method described in Section 3.1.2a (Figure 3.1a), a similar graph may be drawn where the upper line used the point ($t = 0.7$, $\lambda_b = \mu = 0.77$) and lower line used the point ($t = 1.0$, $\lambda_s = \mu = -0.81$) as intersection (See Figure 5.6). This graph demonstrates how the biggest and the smallest eigenvalues depend upon the value of t and it can help us to select the desired value of the relaxation factor. First, using Figure 5.6, we may find the biggest and smallest eigenvalues for the following four values of the relaxation factor, $t = 0.7$, 0.8, 0.9 and 1.0

The Relaxation Factor t	The Biggest Eigenvalue λ_b	The Smallest Eigenvalue λ_s
0.7	+ 0.77	- 0.27
0.8	+ 0.736	- 0.45
0.9	+ 0.7	- 0.63
1.0	+ 0.67	- 0.81

Second, we may calculate the value of the relaxation factor which causes the biggest and the smallest eigenvalues to be equal in modulus by using equation (3.13) as

$$t = \frac{2}{2 - 0.67 + 0.81} = 0.9346$$

and the eigenvalues are

$$\lambda_b = -\lambda_s = (1 - 0.9346) + 0.9346 \, . \, 0.67 = 0.6916$$

TABLE 5.5

Two Comparison Runs Using Different
Relaxation Factors t (Eq.5.2)

Number of Iteration	t = 0.7		t = 1.0	
	Propane lb-mole/hr	Fixed Point Test of Propane	Propane lb-mole/hr	Fixed Point Test of Propane
3	253.09	.005186	259.58	-.001682
4	266.22	.005865	283.54	+.007533
5	273.88	.003012	279.19	-.001459
6	288.01	.002395	292.55	+.003895
7	284.32	.001659	288.25	-.001241
8	287.60	.001266	296.18	+.002248
9	290.04	.000947	292.78	-.000945
10	291.90	.000723	297.64	+.001356
11	293.32	.000552	295.14	-.000685
12	294.41	.000424	298.18	+.000840
13	295.25	.000327	296.41	-.000482
14	295.89	.000252	298.34	+.000530
15	296.38	.000195	297.12	-.000333
16	296.77	.000151	298.35	+.000338
17	297.07	.000117	297.51	-.000228
18	297.30	.000090	298.31	+.000218
19	297.48	.000070	297.75	-.000154
20	297.62	.000054	298.26	+.000141
21			297.88	-.000104
22			298.22	+.000092
23			297.96	-.000069
24			298.18	+.000060
25			298.01	-.000046

TABLE 5.6

Results of the Iterations Using

Four Different Relaxation Factors

Number of Iteration	The Norm of the Fixed Point Test $\|\|\bar{T}_n\|\|_2$			
	t = 0.7	t = 0.8	t = 0.9	t = 1.0
2	.36974	.36974	.36974	.36974
3	.10107	.19029	.29114	.40330
4	.08335	.14026	.22163	.32167
5	.01694	.04147	.13620	.32072
6	.02579	.04253	.10100	.22719
7	.01348	.008778	.05682	.21305
8	.01131	.013678	.04384	.15602
9	.007696	.003756	.02338	.13964
10	.005844	.005262	.01882	.10569
11	.004299	.002391	.00952	.09144
12	.003253	.002407	.00806	.07092
13	.002463	.001446	.003839	.05998
14	.001879	.001225	.003469	.04730
15	.001440	.000835	.001533	.03942
16	.001106	.000659	.001502	.03143
17	.0008525	.000475	.000606	.02594
18	.0006573	.000363		.02084
19	.0005086			.01709
20	.0003931			.01379
21				.01127
22				.00912
23				.00744
24				.00603
25				.00491

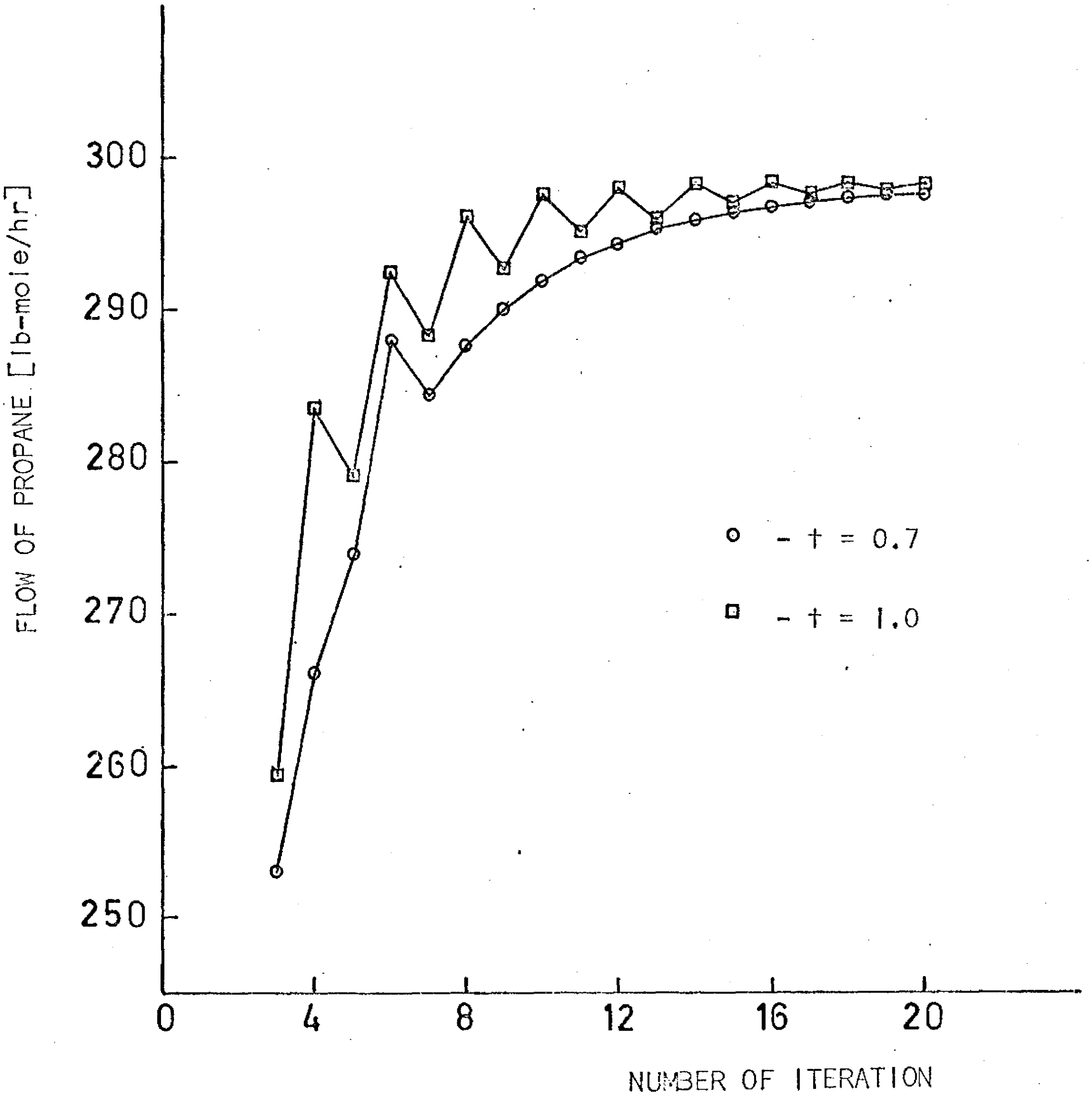

FIGURE 5.4 COMPARISON RUNS USING DIFFERENT RELAXATION FACTOR †

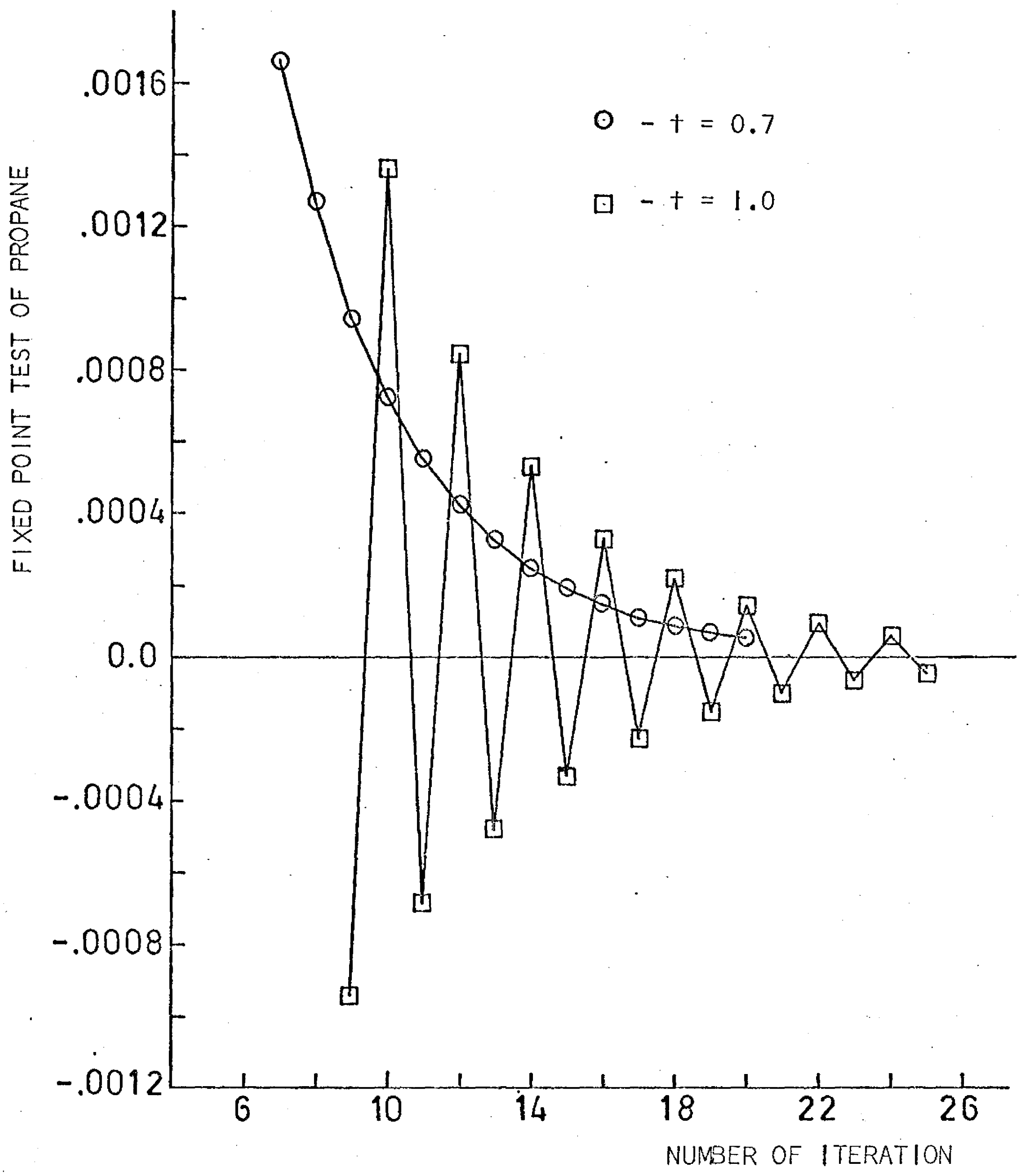

FIGURE 5.5 COMPARISON RUNS USING DIFFERENT RELAXATION FACTOR t

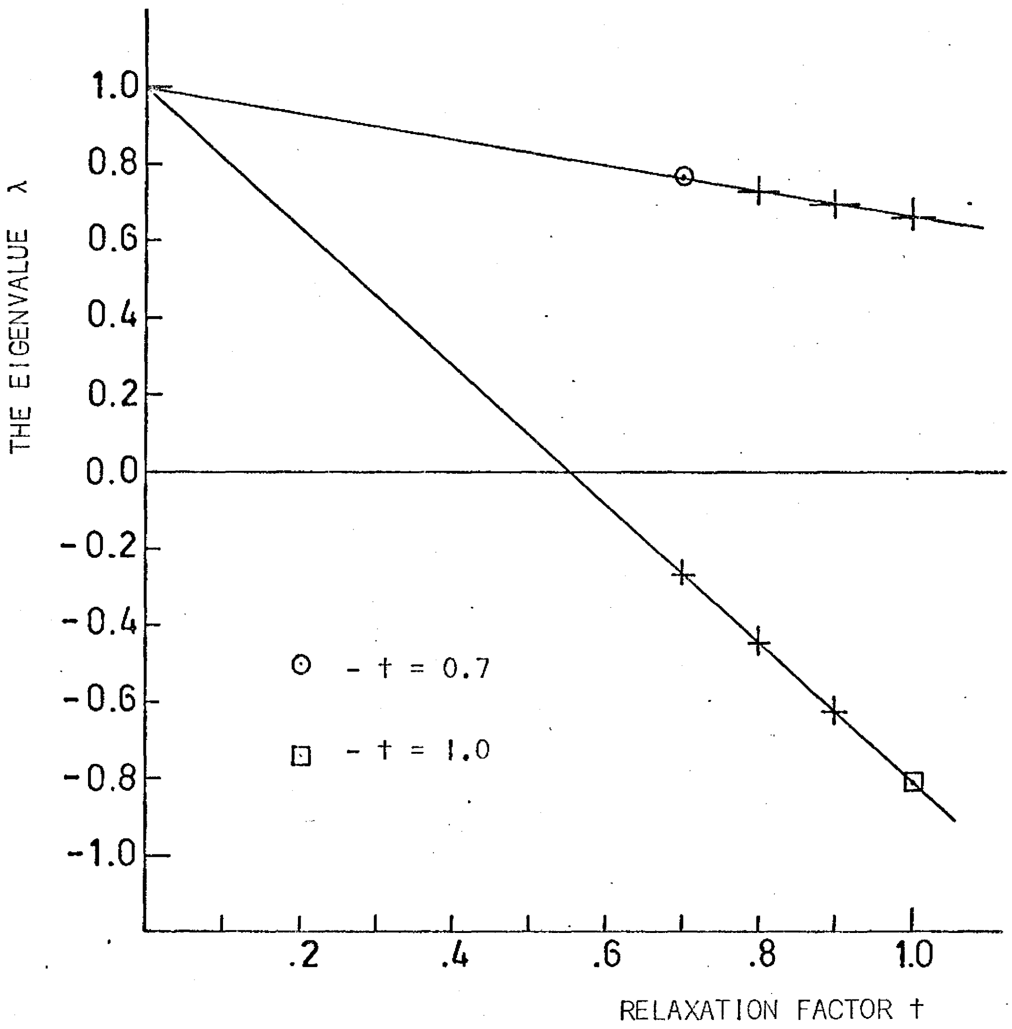

FIGURE 5.6 VARIATION OF THE BIGGEST AND THE SMALLEST EIGENVALUE WITH THE RELAXATION FACTOR t

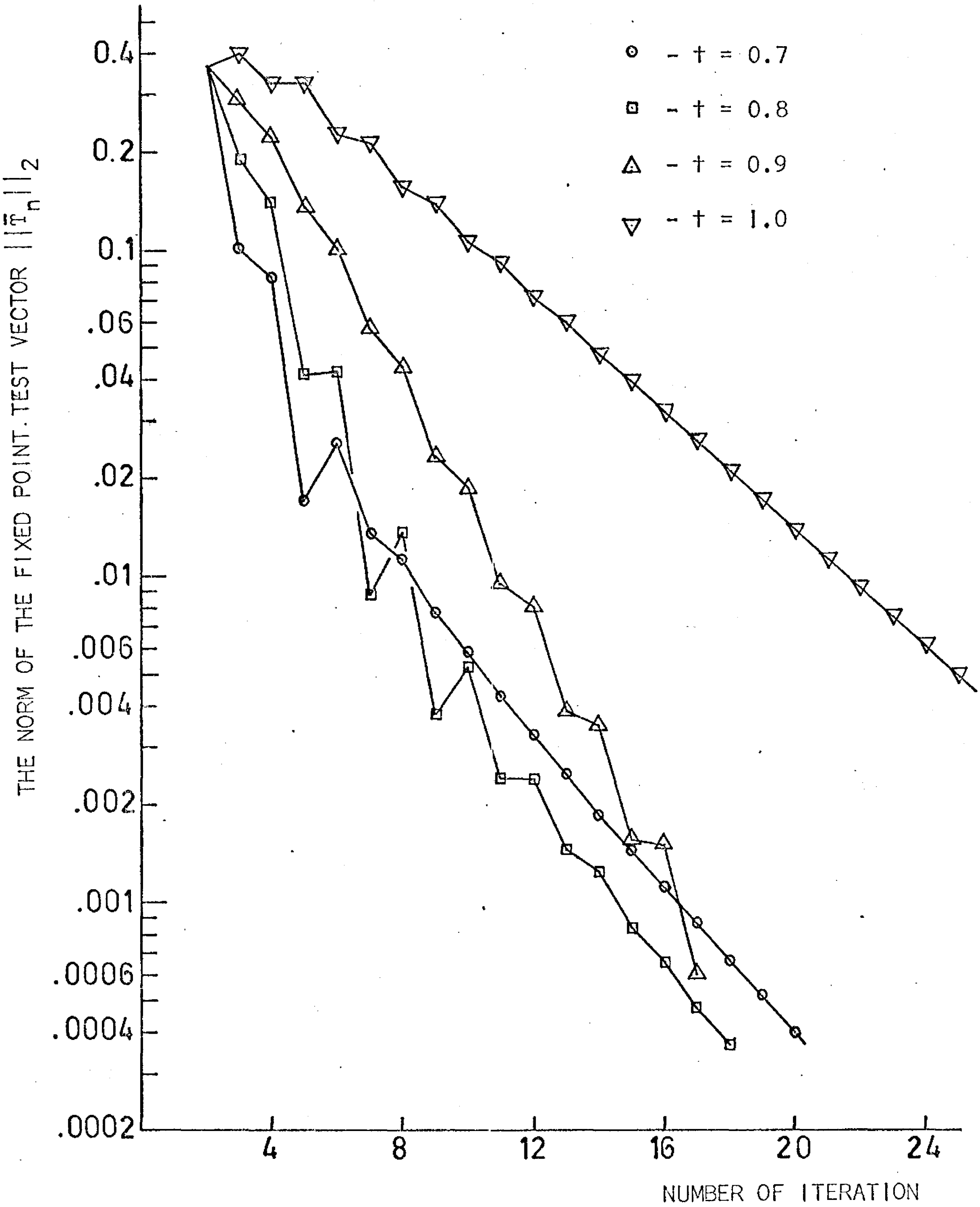

FIGURE 5.7 RESULTS OF THE ITERATION USING FOUR DIFFERENT RELAXATION FACTORS

Thus, if the relaxation factor will be chosen as 0.9346, the two eigenvalues, the biggest and smallest, will dominate the iteration which will never approach a geometric progression in the form of equation 2.50. Truly, for the case of $t = 0.9$, the biggest and the smallest eigenvalues are close in absolute value, and as it is given in Figure 5.7, the norm of the fixed point test approaches the solution not as straight line.

Now, in this case study which is a non-linear iteration, the biggest and smallest eigenvalues of the first order partial derivatives matrix (2.19) and (2.33) may slightly change as the iteration proceeds. However, it is clear that for a relaxation factor of 0.7, the positive geometric coefficient dominates the iteration and for $t = 1.0$ the negative geometric coefficient dominates the iteration. If the relaxation factor is between these two values, for example $t = 0.8$ or $t = 0.9$, the biggest and the smallest eigenvalues are very close in modulus and the iterationshardly approach geometric progression (See Figure 5.7).

5.1.5 Conclusions and Remarks

The full matrix technique for accelerating the convergence is demonstrated (Table 5.4 and Figure 5.3). The rate of convergence has been increased by factor of about 4.25. Although seventeen iterations were necessary for evaluating the matrix of the convergence promotion coefficients, this matrix was utilized unchanged for series of runs for evaluating the performance of the alkylation unit under different operation conditions.

It is important to point out that when the iteration approaches a geometric progression after a small number of iterations (See Table 5.4), the geometric extrapolation technique (Section 3.2) can also be applied to promote the convergence.

5.2 Simulation of the Hydrate Wash Section of the Bayer Process for Alumina Extraction

The convergence of the hydrate wash section was studied during the simulation project of the Bayer process for alumina extraction, which was carried out as a 4th year student project during the academic session 1968/69 (M3). This section, the hydrate wash, consists of four recycles and the convergence appeared to be slow (See Figure 5.8).

The hydrate wash section washes soluble impurities.from the product alumina hydrate, and then filters the excess moisture from the hydrate before it proceeds to the kiln for drying.

5.2.1 General Description of the Hydrate Wash Section

Alumina and other components from the precipitators enter the hydrate washing for processing. This feed enters the primary classifier (Equipment 18, Figure 5.8) where a split in solids takes place. The solids in the underflow are passed through a washing circuit, consisting of three washing stages in series (Equipments 23, 24, 25) then are filtered (Equipment 26) to increase the solids concentration, washed (Equipments 58 and 29) and then dehydrated and calcined (Stream 34) in a rotary kiln to give the final product of aluminum oxide ($A\ell_2O_3$).

The overflow from the primary classifier (Stream 42) is again classified into fine and coarse solids in the secondary classifier.

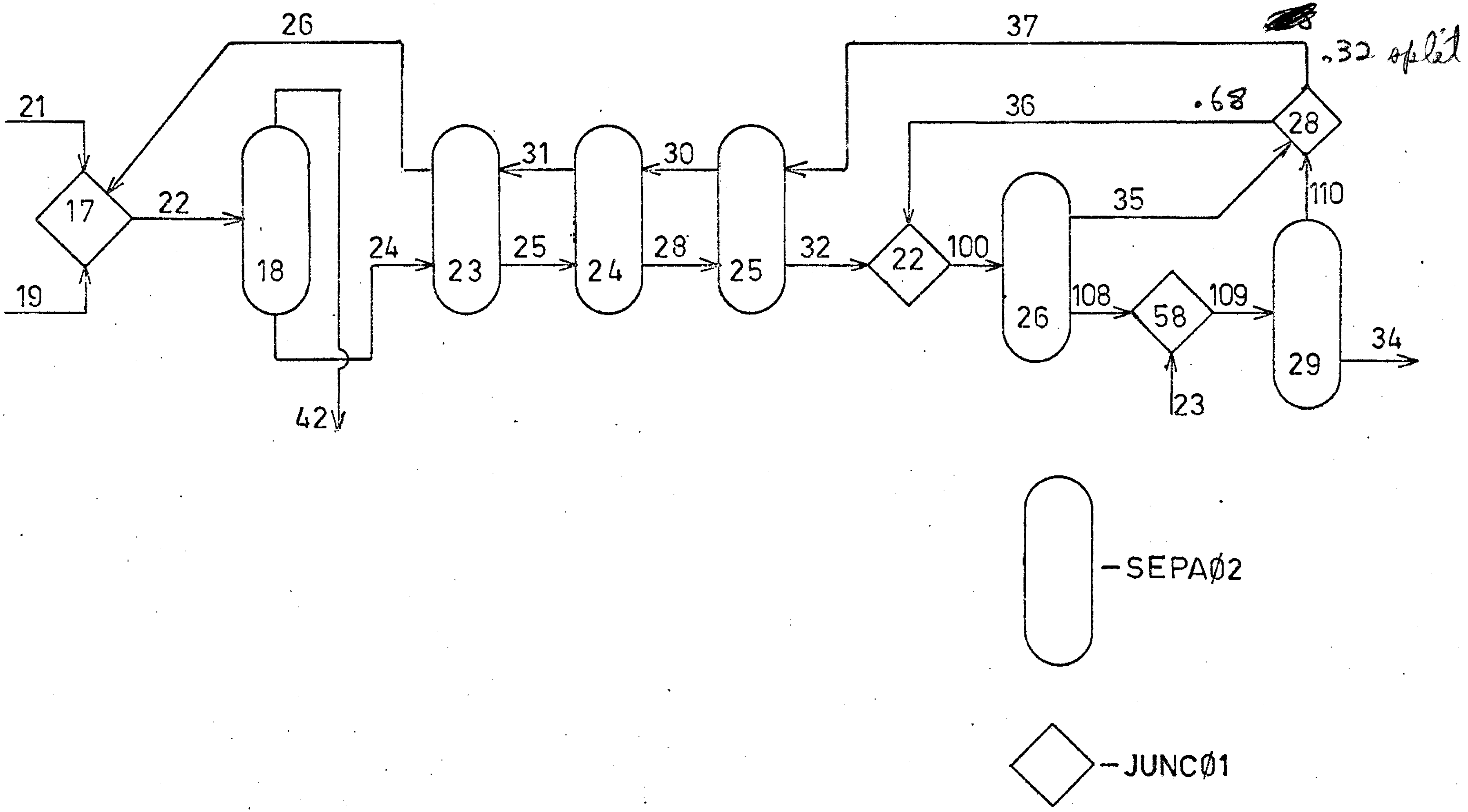

FIGURE 5.8 THE HYDRATE WASH SECTION OF THE BAYER PROCESS FOR ALUMINA EXTRACTION

Washing is carried out using a 3-stage countercurrent system of wash tanks (Equipments 23, 24, 25). The operation allows for a few percent of solids to be carried in the overflow stream.

The hydrate is filtered on a rotary drum filter (Equipment 26) (to reduce moisture to ∿15%) and the liquor is returned as wash liquor to the countercurrent wash.

The solid is washed again (Equipment 58) with fresh water (Stream 23) and the hydrate is filtered on a rotary drum filter (Equipment 29). The liquor is also returned as wash liquor to the countercurrent wash section.

5.2.2 The Equipment Modules

The classifier, the wash, and the filter are represented by the same module SEPAØ2 which calculates the split of one or more feeds according to the fraction of inlet solids passing overflow and the weight fraction of solids in the underflow stream. These parameters are stored in the equipment parameter matrix.

The program calculates the total weight of feed (lb/hr) from the known input components multiplied by their respective molecular weight (Table 5.7). From the specified weight fraction of feed as solids out the top, that total solids flow in the overflow (lb moles/hr) is calculated. The underflow of solids (lb moles/hr) is determined by difference. The weight of solids (lb/hr) in the underflow can now be calculated from the molecular weights.

The total weight of liquid components (lb/hr) entering the modules is calculated. Then the weight of liquid (lb/hr) in the underflow stream is calculated knowing the weight of solids and weight fraction of solids in the bottom stream.

A split of liquid is calculated by the ratio of the weight of liquid in the bottom stream to the total liquid entering the module. Knowing the split, the amounts of liquid (lb moles/hr) in the overflow and underflow are calculated. Equipments 18, 23, 24, 25, 26, and 29 are represented by the SEPAØ2 module.

Equipments 17, 22, 28, 58 are simple mixer-splitter module, JUNCØ1 (J1). Program listings are given in Appendix E.

5.2.3 The Computational Method

The hydrate wash section has four recycles, therefore four streams ought to be assumed in order to carry out the calculation. Streams 22, 30, 31 and 100 have been chosen as assumed streams and the computational sequence of the equipment is then as {26, 58, 29, 28, 18, 23, 17, 24, 25, 22}. The number of components in every stream is 22; the list of the components and their molecular weights is given in Table 5.7.

The two coefficients that are necessary for every "SEPAØ2" model and which are located in the Equipment Parameter Matrix, are given in Table 5.8. The feed streams to the wash section, streams 19, 21, 23 are given in Table 5.9. In addition the initial and the steady state values of the components in every assumed stream ($\bar{X}_o$ and $\bar{X}_s$) are presented in Tables 5.10, 5.11, 5.12 and 5.13. Initial values of the components were taken from plant data.

First, the successive substitution technique (2.5) and (2.6) was applied to solve this recycle problem but the convergence was slow. Then, the geometric extrapolation technique has been applied as a supplement to successive substitution in order to promote the convergence (See Section 3.2).

After every iteration the test vector (2.26) and its second norm (2.12) were calculated. The ratio between two successive norms of the test vector (2.58) were also calculated as

$$\mu_n = \frac{||\bar{\theta}_n||_2}{||\bar{\theta}_{n-1}||_2}$$

and the fractional change of μ_n (2.56) was also obtained

$$\Lambda_n = (\mu_n - \mu_{n-1}) / \mu_{n-1}$$

In addition, a test was made every iteration to detect if the iterate approaches a geometric progression, thus if Λ_n satisfies (2.57) where the tolerance was chosen in this case study as $\gamma = 0.005$.

5.2.4 The Computational Results

The norm of the test vector $||\bar{\theta}_n||_2$, and the geometric coefficient μ_n for every iteration are represented in Table 5.14. The norm of the test vector is also plotted in Figure 5.9.

Considering first the iteration without convergence promotion and examining the change of the norm of the test vector with the number of iterations it appears that the iterate has two regimes. The first one, from iterate 5 to 10, the rate of convergence is relatively rapid and the geometric coefficient is around 0.57 (See Table 5.14, Figure 5.9). The second regime, from iterate 15 to 60, the rate of convergence is much slower, and the geometric coefficient is around 0.88.

This unique behaviour is due to the fact that the components in the streams can be divided into two groups. The solid components which control the first regime and the liquid components which control the second one.

The solid components are independent of the liquid and they converge relatively quickly. The liquid components are strongly dependent on the solid components and in addition they converge slowly. Therefore after the solid components converge (∿ 10 iterations) the liquidstake control of the iteration process, which converges relatively slowly. The reason for the dependence of the liquid components on the solid flow is that, in the module "SEPAØ2", the liquid underflow is only a function of the solid underflow.

The geometric extrapolation technique for promoting the convergence was applied to the second regime. Three times the iterate approachedgeometric progression, in iteration 21, 30 and 39 (See Table 5.14). Each time, the last three iterations were used to calculate α_i (3.28) and equation (3.30) was applied for approximating the solution $\bar{X}_s'$. The relaxation factor t, in equation (3.30), was chosen as $t = 0.7$. The number of iterations due to the convergence promotion were reduced from 60 to 40.

TABLE 5.7

List of Components and their Molecular Weight

Component Number	Location in Stream Vector	The Component Flow lb mole/hr		Molecular Weight
1	6	WATER		18.020
2	7	Na_2CO_3		105.993
3	8	NaOH (as Free Caustic)		40.001
4	9	$3(Na_2O.A\ell_2O_3.2SiO_2.0.6Na_2O.$ $.A\ell_2O_3.0.22Na_2SO_4.0.23Na_2CO_3$		1006.3602
5	10	$A\ell_2O_3.3H_2O$	(SOLID)	156.020
6	11	$A\ell_2O_3.2SiO_2.2H_2O$	(SOLID)	258.180
7	12	$Fe_2O_3.H_2O$	(SOLID)	177.720
8	13	P_2O_5	(SOLID)	141.950
9	14	CaO	(SOLID)	56.080
10	15	TiO_2	(SOLID)	79.900
11	16	$CaCO_3$	(SOLID)	100.091
12	17	Carbon compounds	(SOLID)	44.011
13	18	Unknown inert solid	(SOLID)	119.980
14	19	$NaA\ell_2$	(DISSOLVED)	81.971
15	20	$A\ell_2O_3.2SiO_2.2H_2O$	(DISSOLVED)	258.180
16	21	Fe_2O_3	(DISSOLVED)	159.700
17	22	P_2O_5	(DISSOLVED)	141.950
18	23	$(3CaO.A\ell_2O_3).SiO_2$	(DISSOLVED)	600.490
19	24	TiO_2	(DISSOLVED)	79.900
20	25	Na_2SO_4	(DISSOLVED)	142.048
21	26	Carbon compounds	(DISSOLVED)	44.011
22	27	$Na_2H_2C_2O_2$	(DISSOLVED)	104.024

TABLE 5.8

Equipment Parameters Matrix

Data for SEPAØ2 Modules

Equipment Number	Wt. Fraction of Feed as Solids in the Top Stream	Wt. Fraction of Solids in the Bottom Stream.
18	0.6	0.6
23	0.036	0.6
24	0.045	0.6
25	0.0025	0.6
26	0.00	0.855
29	0.00	0.855

TABLE 5.9

Feed Streams to the Hydrate
Wash Section

Component Number	Location in Stream Vector	Stream No. 19 [lb.mole/hr]	Stream No. 21 [lb/mole/hr]	Stream No. 23 [lb.mole/hr]
1	6	87989.7969	84277.1416	6125.0
2	7	862.7023	887.8482	
3	8	3516.6652	3500.5645	
4	9	.0043	0.0036	
5	10	1166.6548	2846.943	
6	11	0.2870	0.287	
7	12	.3011	.2994	
8	13	.0001	0.0	
9	14	.0004	0.0003	
10	15	.0117	0.0099	
11	16	.0017	0.0014	
12	17	.0019	.0016	
13	18	.0022	.0018	
14	19	2065.8810	2100.5155	
15	20	5.7058	5.1675	
16	21	0.0	0.0	
17	22	3.9451	3.5907	
18	23	0.0	0.0	
19	24	60.3192	41.0175	
20	25	35.4092	27.8576	
21	26	1580.4762	1438.4926	
22	27	0.0	0.0	

TABLE 5.10

The Initial Value and Steady-State Solution in the Assumed Stream 22

Component Number	Location in SN	Initial Value [lb.mole/hr]	Steady-State Value [lb.mole/hr]
1	6	165000.0	183348.8105
2	7	1639.0	1821.1220
3	8	6570.0	7300.1219
4	9	0.007	.008
5	10	3600.0	4074.9445
6	11	0.54	0.5826
7	12	0.54	.6097
8	13	0.0001	.0001
9	14	0.0006	.0007
10	15	0.0180	.0219
11	16	0.0027	.0031
12	17	0.0033	.0036
13	18	0.0036	.0041
14	19	3800.	4334.3598
15	20	9.90	11.3116
16	21	0.0	0.0
17	22	7.1	7.8396
18	23	0.0	0.0
19	24	90.0	105.4220
20	25	59.4	65.8173
21	26	2700.0	3140.6749
22	27	0.0	0.0

TABLE 5.11

The Initial Value and Steady-State

Solution in the Assumed Stream 30

Component Number	Location in SN	Initial value [lb.mole-hr]	Steady-State Value [lb.mole/hr]
1	6	10710.0	11910.2351
2	7	28.0	32.5206
3	8	117.	130.3625
4	9	0.0	0.0
5	10	3.6	3.9
6	11	0.0005	0.0006
7	12	0.0005	0.0006
8	13	0.0	0.0
9	14	0.0	0.0
10	15	0.0	0.0
11	16	0.0	0.0
12	17	0.0	0.0
13	18	0.0	0.0
14	19	69.0	77.4004
15	20	0.18	0.202
16	21	0.0	0.0
17	22	0.13	0.1400
18	23	0.0	0.0
19	24	1.7	1.8826
20	25	0.99	1.1753
21	26	50.0	56.0839
22	27	0.0	0.0

TABLE 5.12

The Initial Value and Steady-State

Solution in the Assumed Stream 31

Component Number	Location in SN	Initial Value [lb.mole/hr]	Steady-State Value [lb.mole/hr]
1	6	10800.0	11800.9294
2	7	45.0	49.3985
3	8	180.0	198.0187
4	9	0.0001	0.0001
5	10	66.0	74.0998
6	11	0.009	0.0106
7	12	0.009	0.0111
8	13	0.0	0.0
9	14	0.0	0.0
10	15	0.0004	0.0004
11	16	0.0001	0.0001
12	17	0.0001	0.0001
13	18	0.0001	0.0001
14	19	108.0	117.5704
15	20	0.27	0.3068
16	21	0.0	0.0
17	22	0.18	0.2127
18	23	0.0	0.0
19	24	2.6	2.8596
20	25	1.6	1.7853
21	26	76.0	85.1911
22	27	0.0	0.0

TABLE 5.13

The Initial Value and Steady-State

Solution in the Assumed Stream 100

Component Number	Location in SN	Initial Value [lb.mole/hr]	Steady-State Value [lb.mole/hr]
1	6	30600.0	34327.7861
2	7	63.0	68.8231
3	8	265.0	275.8855
4	9	0.0029	0.0031
5	10	1440.0	1568.6310
6	11	0.18	0.2243
7	12	0.18	0.2347
8	13	0.0	0.0
9	14	0.0003	0.0003
10	15	0.0070	0.0084
11	16	0.0010	0.0012
12	17	0.0012	0.0014
13	18	0.0014	0.0016
14	19	144.0	163.8013
15	20	0.36	0.4275
16	21	0.0	0.0
17	22	0.27	0.2963
18	23	0.0	0.0
19	24	3.6	3.9841
20	25	2.3	2.4873
21	26	95.0	118.6893
22	27	0.0	0.0

TABLE 5.14

Results from the Iteration of the Hydrate Wash Section

Iteration Number	Iterate without Convergence Promotion		Iterate with Convergence Promotion	
	Norm of the Test Vector	Geometric Coefficient	Norm of the Test Vector	Geometric Coefficient
	$\|\|\bar{\theta}_n\|\|_2$	μ_n	$\|\|\bar{\theta}_n\|\|_2$	μ_n
3	.1898 E+0	0.2465		
4	.1378 E+0	0.7260		
5	.7840 E-1	0.5688		
6	.4465 E-1	0.5696		
7	.2572 E-1	0.5760		
8	.1502 E-1	0.5840		
9	.9462 E-2	0.6299		
10	.7264 E-2	0.7676		
11	.6755 E-2	0.9266		
12	.6687 E-2	0.9899		
13	.6572 E-2	0.9829		
14	.6319 E-2	0.9614		
15	.5951 E-2	0.9418		
16	.5513 E-2	0.9264		
17	.5043 E-2	0.9148		
18	.4570 E-2	0.9061	.4570 E-2	0.9061
19	.4111 E-2	0.8996	.4111 E-2	0.8996
20	.3678 E-2	0.8947	.3678 E-2	0.8947
21	.3277 E-2	0.8909	.3277 E-2	0.8909*
22	.2910 E-2	0.8881		
23	.2578 E-2	0.8859		
24	.2280 E-2	0.8842	.5622 E-3	0.8805

* Applying convergence promotion

TABLE 5.14 Cont'ed

	Iterate without Convergence Promotion		Iterate with Convergence Promotion	
Iteration Number	Norm of the Test Vector	Geometric Coefficient	Norm of the Test Vector	Geometric Coefficient
	$\|\|\bar{\theta}_n\|\|_2$	μ_n	$\|\|\bar{\theta}_n\|\|_2$	μ_n
25	.2013 E-2	0.8829	.5196 E-3	0.9241
26	.1775 E-2	0.8819	.4742 E-3	0.9127
27	.1564 E-2	0.8811	.4286 E-3	0.9038
28	.1377 E-2	0.8805	.3846 E-3	0.8974
29	.1212 E-2	0.8800	.3434 E-3	0.8928
30	.1066 E-2	0.8797	.3054 E-3	0.8894*
31	.9374 E-3	0.8794		
32	.8241 E-3	0.8791		
33	.7243 E-3	0.8789	.5469 E-4	0.8839
34	.6365 E-3	0.8788	.5030 E-4	0.9198
35	.5593 E-3	0.8787	.4573 E-4	0.9091
36	.4914 E-3	0.8787	.4121 E-4	0.9011
37	.4317 E-3	0.8785	.3693 E-4	0.8962
38	.3792 E-3	0.8785	.3293 E-4	0.8915
39	.3332 E-3	0.8786	.2925 E-4	0.8883*
40	.2927 E-3	0.8783		
42	.2258 E-3	0.8784		
44	.1741 E-3	0.8781		
46	.1344 E-3	0.8786		
48	.1037 E-3	0.8785		
50	.7994 E-4	0.8779		
52	.6165 E-4	0.8779		
54	.4751 E-4	0.8776		
56	.3665 E-4	0.8772		
58	.2830 E-4	0.8779		
60	.2180 E-4	0.8773		

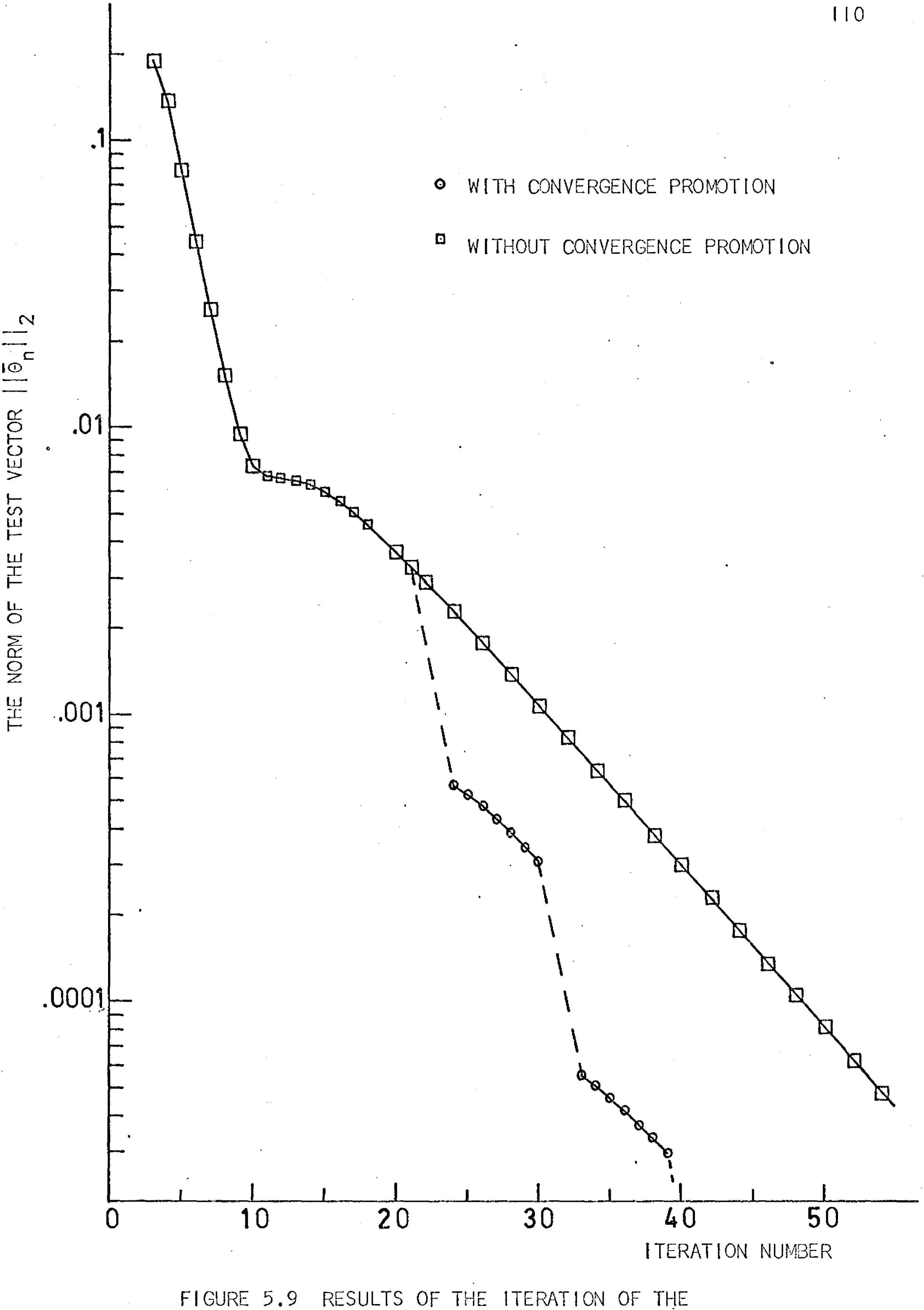

FIGURE 5.9 RESULTS OF THE ITERATION OF THE HYDRATE WASH SECTION

6. MULTI-COMPONENT DISTILLATION COLUMNS

6.1 Introduction

The solution of multi-stage separation problem requires mass balances, heat balances, and equilibrium conditions to be obeyed over each stage of the process. The basic equations are often simple but the counter-current operation interconnects all stages. Amundson and Pontinen (A6) describe the problem as the solution of series of non-linear simultaneous equations.

Many methods have been proposed for the computer solution of these equations (H2, R6, P1, P2, W2, T1). The Θ-method (H2, P1) and the relaxation method (R6, P1) are most widely used although these techniques converge very slowly for the calculation cases where pure distillate is required. The direct iteration (successive substitution) method (W2) is also proposed to solve these equations.

In this present work two methods have been applied for calculating the multi-component distillation column where, in each of which a tridiagonal matrix algorithm for the solution of the component material balances has been used.

In the first method, successive substitution iteration (W2) was employed. The profiles of the temperature and liquid flow in the column have been assumed in order to start the calculation and they have been redetermined in every iteration. The geometric extrapolation technique (Section 3.2) has been applied to accelerate the convergence of the iteration.

The second method solves the distillation columns as a set of algebraic equations where the profiles of the temperature and the liquid flow are used as the variables. The modified form of Newton's method (L3) has been employed to solve the set of equations. The calculation has been accelerated by the geometric extrapolation method.

6.2 The Mathematical Model

The basic equations for a distillation column are derived by making material and heat balances around the "jth" tray of the model (Figure 6.1). The standard assumptions of constant column pressure and theoretical equilibrium trays are made. Heat and mass balances plus equilibrium relationships comprise the five sets of equations to be solved, namely

a) Material balance

$$L_{j-1} + V_{j+1} - L_j - V_j = 0 \tag{6.1}$$

b) Component material balance

$$L_{j-1}\, x_{i,j-1} + V_{j+1}\, y_{i,j+1} - L_j\, x_{i,j} - V_j\, y_{i,j} = 0 \tag{6.2}$$

c) Heat balance

$$L_{j-1}\, h_{j-1} + V_{j+1}\, H_{j+1} - L_j\, h_j - V_j\, H_j = 0 \tag{6.3}$$

d) Vapour-liquid equilibrium

$$y_{i,j} - K_{i,j}\, x_{i,j} = 0 \tag{6.4}$$

e) The summation condition

$$\sum_i x_{i,j} - 1 = 0 \qquad (6.5a)$$

or

$$\sum_i y_{i,j} - 1 = 0 \qquad (6.5b)$$

where

L_j = liquid flow from "jth" tray [lb mole/hr.]

V_j = vapour flow from "jth" tray [lb mole/hr.]

$x_{i,j}$ = mole fraction of i component in liquid phase of tray j

$y_{i,j}$ = mole fraction i component in vapour phase at tray j

h_j = enthalpies of the liquid stream L_j, [B.T.U/lb mole]

H_j = enthalpies of the vapour stream V_j, [B.T.U/lb mole]

$K_{i,j}$ = equilibrium ratio of component i at stage j

$$K_{i,j} = y_{i,j} / x_{i,j}$$

Suppose that the number of components is Nc, then for every tray we have the (2.Nc + 3) equations shown above and (2.Nc + 3) variables which are L_j, V_j, T_j, $x_{i,j}$, $y_{i,j}$, $i = 1, 2, 3, \ldots$ Nc, where T_j = temperature at the "jth" tray.

Substitute equation (6.4) into equation (6.2) to eliminate $y_{i,j}$ and then reduce the component material balance equations (6.2) to a tridiagonal matrix form, as

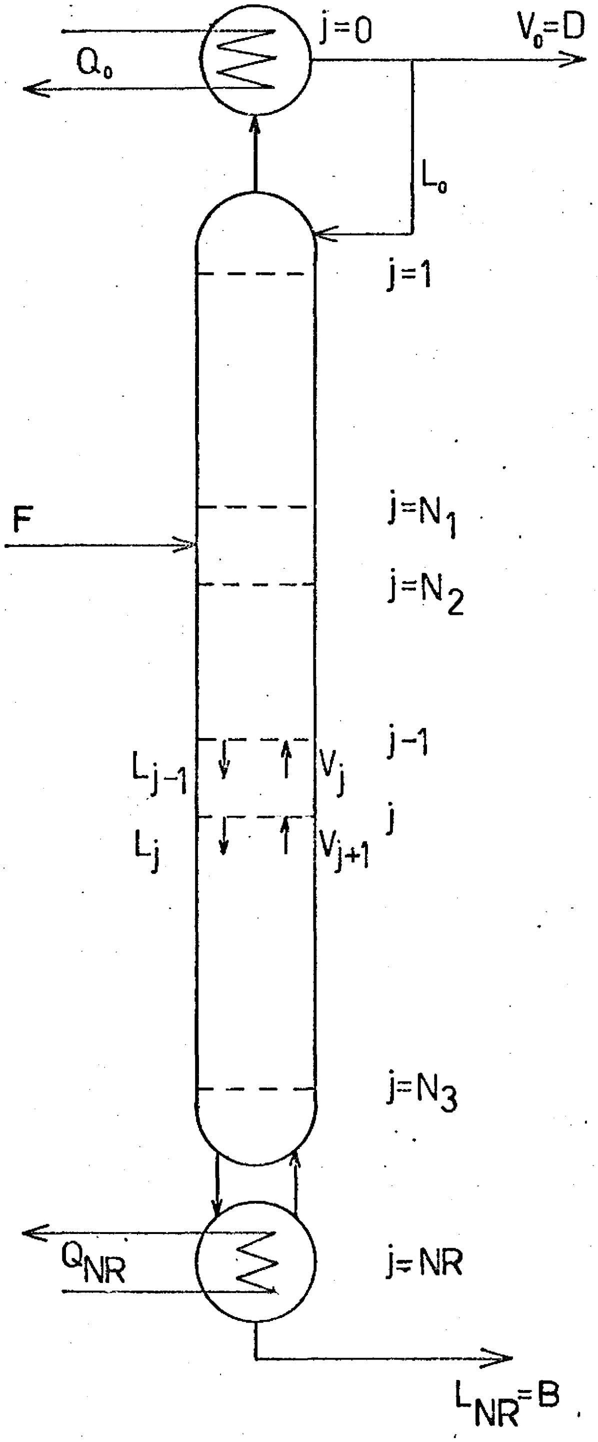

FIGURE 6.1 STANDARD DISTILLATION COLUMN CONFIGURATION

$$B_o\, x_{i,o} + C_o\, x_{i,1} = Z_o \tag{6.6}$$

$$A_j\, x_{i,j-1} + B_j\, x_{i,j} + C_j\, x_{i,j+1} = Z_j \qquad 1 \le j \le NR-1 \tag{6.7}$$

$$A_{NR}\, x_{i,NR-1} + B_{NR}\, x_{i,NR} = Z_{NR} \tag{6.8}$$

or in matrix form

$$\begin{bmatrix} B_o & C_o & & & & \\ A_1 & B_1 & C_1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & A_j & B_j & C_j & \\ & & & \cdot & \cdot & \cdot \\ & & & & A_{NR} & B_{NR} \end{bmatrix} \cdot \begin{bmatrix} x_{i,o} \\ x_{i,1} \\ \cdot \\ x_{i,j} \\ \cdot \\ x_{i,NR} \end{bmatrix} = \begin{bmatrix} Z_o \\ Z_1 \\ \cdot \\ Z_j \\ \cdot \\ Z_{NR} \end{bmatrix} \tag{6.9}$$

$1 \le i \le Nc$

or simply

$$[A_{Bc}] \cdot \{x_{i,j}\} = \{Z_j\} \qquad 1 \le i \le Nc \tag{6.9a}$$

where

$A_o = 0$

$B_o = -V_o - L_o = -D(1+R)$ (total condenser)

$C_o = V_1\, K_{i,1}$

$A_j = L_{j-1} \qquad 1 \le j \le NR$

$B_j = -L_j - V_j\, k_{i,j} \qquad 1 \le j \le NR$

$$C_j = V_{j+1} K_{i,j+1} \qquad 1 \leq j \leq NR-1$$

$$C_{NR} = 0$$

$$Z_j = 0 \qquad j \neq N1 \ \& \ j \neq N2$$

$$Z_{N1} = -V_f \, y_{i,f}$$

$$Z_{N2} = -L_f \, x_{i,f}$$

The material and heat balance equations (6.1) and (6.3) can be expressed as an overall balance over all stages from the condenser through the "jth" stage. Now if the feed flow F, the distillate flow D and reflux ratio R are given, the material balance becomes,

$$V_{j+1} = L_j + W_j \tag{6.10}$$

where

$$W_j = D \quad \text{for} \quad 1 \leq j \leq N1-1$$

$$W_j = D - V_f \quad \text{for} \quad j = N1$$

$$W_j = D - F \quad \text{for} \quad N2 \leq j \leq N3$$

and

$$V_o = D$$

$$(L_o = D.R)$$

$$V_1 = V_o + L_o = D\ (1+R)$$

$$(L_{NR} = B = F - D)$$

V_f is the vapour fraction of the feed, $F = L_f + V_f$.

The heat balance (6.3) with material balance (6.1) around the upper part of the column become

$$L_j\ (h_j - H_{j+1}) - D\ (H_{j+1} - H_D) - U_j = 0 \tag{6.11}$$

where

$$U_j = -Q_o \qquad \text{for} \quad 1 \le j \le N1-1$$

$$U_j = -Q_o + V_f(H_{v,f} - H_{j+1}) \qquad \text{for} \quad j = N1$$

$$U_j = -Q_o + F(\bar{H}_f - H_{j+1}) \qquad \text{for} \quad N2 \le j \le N3$$

where

$H_{v,f}$ = enthalpy of the vapour part of the feed

$\bar{H}_f$ = enthalpy of the feed

($\bar{H}_f = H_{L,f} + H_{v,f}$)

Finally, substitute equation (6.4) into (6.5a) to get

$$\sum_{i=1}^{Nc} K_{i,j}.x_{i,j} - 1 = 0 \qquad (6.12)$$

$$j = 1, 2, 3, \ldots NR$$

Thus, after eliminating the vapour compositions, $y_{i,j}$, the four sets of equations, which remain to be solved, are (6.9), (6.10), (6.11) and (6.12). If the reflux ratio R and distillate flow D are given then, for the case of a total condenser, the number of equations which we have to solve is NR.(Nc + 3) - 2, where Nc is the number of components and NR is the number of trays including the reboiler.

The two methods which were employed for solving these sets of equations will be discussed in the following sections. Petryschuk's multicomponent distillation program (P1, P2) has been used as a basis for this work with the necessary changes. Thus, instead of Petryschuk's "block relaxation method", equation (6.9) has been applied. The equilibrium ratio $K_{i,j}$ and the enthalpy routine remains the same as in Petryschuk's program where the equilibrium ratios are expressed as polynomial functions of temperature at the column pressure, and the enthalpy for both liquid and vapour streams is calculated by using the enthalpy correlation of Yen and Alexander (Y1).

6.3 The Successive Substitution Method of Iteration

The successive substitution method of iteration was reported by Wang and Henke (W2). Although they indicated good convergence of the iteration in their case studies it appears that the iteration converges very slowly where pure distillate is required. Thus, the iteration converges slowly when the reflux ratio is high or the distillate flow D is relatively small. Very good results were obtained where geometric extrapolation was used for promoting the convergence.

6.3.1 The Computational Procedure

When the flow rate and compositions of the feed stream are given and the amounts of the product streams and reflux ratio are specified, F, D, B and R are constants. If an initial set of $\{L_j\}_o$ and $\{T_j\}_o$ is assumed, the profile of the vapour flow $\{V_j\}$ can be calculated by equation (6.10), matrix $[A_{Bc}]$ and $\{Z_j\}$ (6.9) are also constants, provided that the equilibrium ratio $K_{i,j}$ can be expressed as function of T_j. Then equation (6.9) is a linear system and by inverting the tridiagonal matrix $[A_{Bc}]$ the solution of $\{x_{i,j}\}$ can be easily obtained as

$$\{x_{i,j}\} = [A_{Bc}]^{-1} \{Z_j\} \qquad (6.13)$$

$$i = 1, 2, 3, \ldots Nc$$

After applying the tridiagonal matrix calculation for all the components (6.13), the set of equations (6.12) is used to calculate a new temperature profile, as

$$S_j \, (T_j, \, x_{i,j}) = \sum_{i=1}^{Nc} K_{i,j} \, . \, x_{i,j} - 1 = 0 \qquad (6.14)$$

$$j = 1, \, 2, \, 3, \, \ldots \, NR.$$

This is readily recognized as the bubble-point determination of the temperature T_j. Newton's iteration technique is utilized to seek the temperature that satisfies equation (6.14), where the equilibrium ratios $K_{i,j}$ are expressed as polynomial functions of the temperature T_j. It is worthwhile mentioning that Wang and Henke (W2) used Muller's method for seeking the root of equation (6.14) and they have claimed a better convergence than with Newton's method.

After calculating the temperature profile $\{T_j\}$ the heat balance equation (6.11) can be used to calculate the new liquid flow profile $\{L_j\}$, as

$$L_j = \frac{D(H_{j+1} - H_D) + U_j}{h_j - H_{j+1}} \qquad (6.15)$$

$$j = 1, \, 2, \, 3 \, \ldots \, NR-1.$$

where U_j is given in equation (6.11). The enthalpy is calculated from Yen and Alexander's correlation (Y1) as it is programmed by Petryschuk (P2).

The new temperature and liquid flow profiles of the column can be transferred to become initial values for a new iteration.

These two profiles can be arranged in vector $\bar{X}$ and $\bar{Y}$ and the iteration can be expressed in general by equations in the form of (2.5) and (3.1) as

$$\bar{Y}_n = \bar{F}(\bar{X}_n) \tag{6.16}$$

$$\bar{X}_{n+1} = \bar{X}_n + t_r(\bar{Y}_n - \bar{X}_n) \tag{6.17}$$

where $\bar{X}_n^T = [\{T_j\}_n, \{L_j\}_n]^T$

and $\bar{Y}_n^T = [\{T_j\}_{new}, \{L_j\}_{new}]^T$

and t_r is relaxation factor that is used in case the iteration oscillates or is unstable. Vector $\bar{X}_n$ is the "input" and $\bar{Y}_n$ is the "output". (See Figure 6.2). The set of functions $\bar{F}$ (6.16) represents the calculation procedure which can be summarized by the following steps:

1) Assume an initial temperature profile $\{T_j\}_o$ and liquid flow profile $\{L_j\}_o$ where the two of them are combined in vector $\bar{X}_o$.
2) Calculate the vapour profile $\{V_j\}$ by equation (6.10)
3) Calculate the elements of matrix $[A_{Bc}]$ (6.9) and solve the matrix equations for every component to get $\{x_{i,j}\}$
4) Solve equations (6.14) by Newton's method to get new temperature profile $\{T_j\}_{new}$
5) Calculate the enthalpies of the internal vapour and liquid streams by using Yen and Alexander (YI) correlation and then calculate the new liquid profile $\{L_j\}_{new}$ by equation (6.15)

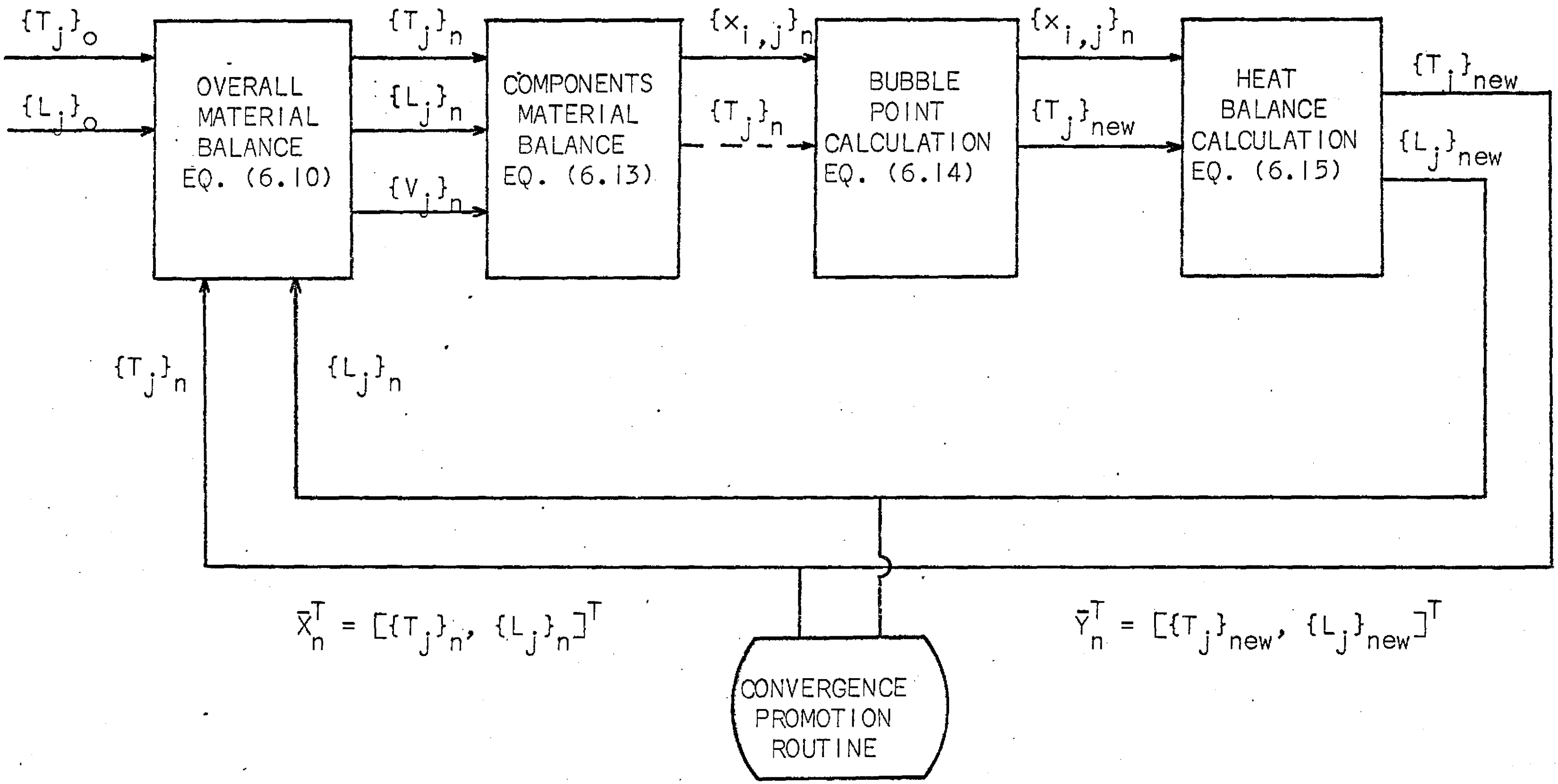
$\{T_j\}_o$
$\{L_j\}_o$
OVERALL MATERIAL BALANCE EQ. (6.10)
$\{T_j\}_n$
$\{L_j\}_n$
$\{V_j\}_n$
COMPONENTS MATERIAL BALANCE EQ. (6.13)
$\{x_{i,j}\}_n$
$\{T_j\}_n$
BUBBLE POINT CALCULATION EQ. (6.14)
$\{x_{i,j}\}_n$
$\{T_j\}_{new}$
HEAT BALANCE CALCULATION EQ. (6.15)
$\{T_j\}_{new}$
$\{L_j\}_{new}$
$\{T_j\}_n$
$\{L_j\}_n$
$\bar{X}_n^T = [\{T_j\}_n, \{L_j\}_n]^T$
$\bar{Y}_n^T = [\{T_j\}_{new}, \{L_j\}_{new}]^T$
CONVERGENCE PROMOTION ROUTINE

FIGURE 6.2 INFORMATION FLOW DIAGRAM

MULTI-COMPONENT DISTILLATION COLUMNS SOLVED BY SUCCESSIVE SUBSTITUTION

6) Arrange the two new profiles, $\{T_j\}_{new}$ and $\{L_j\}_{new}$, in vector $\bar{Y}_n$ and then equation (6.17) may be used to calculate $\bar{X}_{n+1}$.

7) Repeat steps (2) through (6) until convergence is achieved.

The fixed-point test vector (2.28) and its norm are calculated in the end of every iteration as

$$\bar{T}_n = [I\ \bar{X}_n]^{-1}\ (\bar{Y}_n - \bar{X}_n) \qquad (6.18)$$

The test for convergence has been made with the second norm (2.12)

$$||\bar{T}_n||_2 \leq \varepsilon_2 \qquad (6.19)$$

and the iteration has achieved the convergence when the fixed point test satisfies (6.19).

6.3.2 Accelerating the Convergence

The geometric extrapolation technique has been employed for promoting the convergence of the above iteration. Thus, at the end of each iteration the fixed-point test vector (6.18) and its second norm have been calculated. In addition, the geometric coefficient μ_n (2.62) and (2.69) and its fractional change Λ_n (2.56) have been obtained as

$$\mu_n = \frac{||\bar{T}_n||_2}{||T_{n-1}||_2} \qquad (6.20)$$

and

$$\Lambda_n = (\mu_n - \mu_{n-1}) / \mu_{n-1} \qquad (6.21)$$

respectively.

Now, if the number of iterations n is sufficiently large such that Λ_n satisfies equation (2.57), where γ is the specified tolerance, it could be assumed that the iteration has approached a geometric progression and can be expressed in form of equation (2.50). Therefore, the solution can be approximated by equation (3.30) which, in this case study, has been written in different form as

$$\bar{X}'_s = \bar{X}_{n-1} + t_c \cdot \bar{B} \qquad (6.22)$$

where vector $\bar{B}$ can be calculated in a few different ways as

a) $$\bar{B} = [I \; \bar{\alpha}'] \, (\bar{Y}_{n-1} - \bar{X}_{n-1}) \qquad (6.23)$$

where α'_i is defined as

$$\alpha'_i = \left\{ \frac{x_{n-2} - x_{n-1}}{y_{n-1} - y_{n-2} - x_{n-1} + x_{n-2}} \right\}_i \qquad (6.24)$$

$i = 1, 2, 3, \ldots m.$

b) $$B_i = \left\{ \frac{(x_{n-2} - x_{n-1}) \, T_{n-1}}{(T_{n-1} - T_{n-2})} \right\}_i \qquad (6.25)$$

$i = 1, 2, 3, \ldots m.$

where $T_{n,i}$ is the "ith" element of vector $\bar{T}_n$

c) $$B_i = \left\{ \frac{(x_{n-2} - x_{n-1}) \, \mu}{\mu - 1} \right\}_i \qquad (6.26)$$

In this case study, equation (6.22) has been employed, where vector $\bar{B}$ was calculated by equation (6.25). The scalar factor t_c has been used as relaxation factor. The following coefficients have been used in the convergence promotion routine,

$$t_r = 1.0 \qquad \text{(equation 6.17)}$$

$$t_c = 0.7 \qquad \text{(equation 6.22)}$$

$$\gamma = 0.01 \qquad \text{(equation 2.57)}$$

$$\varepsilon_2 = 4.5 \times 10^{-7} \qquad \text{(equation 6.19)}$$

6.3.3 The Results of the Case Study

A conventional fractionator is considered (Figure 6.1). The column has thirty one equilibrium stages, including a total condenser and a reboiler, thus NR = 30. Two hundred moles of feed have the composition; 0.25 propane, 0.25 iso-butane, 0.25 n-butane and 0.25 pentanes. The feed is entered at the thirteenth tray. The column is operated at 290 psia. Products of 50 moles of distillate and 150 moles of bottoms are to be obtained. The feed temperature is 195.0°F where the bubble point of the feed is 215.4°F. The reflux ratio R is chosen as 6.0. Equilibrium coefficients and physical property data are given in Tables 6.1 and 6.2.

The initial and final values of the temperature and liquid flow profiles are given in Table 6.3 and Figures 6.3 and 6.4. Two comparison runs, with and without convergence promotion have been made. The norm of the fixed-point test vector versus the number of iterations is represented in Table 6.4 and Figure 6.5. The geometric coefficients are also given in Table 6.4. The composition profiles in the column at the last iteration are given in Table 6.5 and Figure 6.6.

Considering first the iteration without convergence promotion (Table 6.4 and Figure 6.5). The rate of convergence appears to be very slow. The iteration has approached geometric progression after less than ten iterations. As the system is non-linear, the geometric coefficient gradually changes as the iteration proceeds.

The rate of convergence during the iteration can be calculated by equation (2.77), as

$$R = -\log (\mu_n)$$

and for example the rate of convergence for the following three points was calculated as

Iteration Number	Geometric Coefficient	Rate of Convergence
10	0.910	0.04096
20	0.948	0.02319
120	0.979	0.00922

Thus, the rate of convergence has been substantially decreased as the iteration has proceeded toward the solution.

In the last part of the iteration, after hundred and twenty iterations, the rate of convergence is very slow (R = 0.00922), and almost hundred and nine iterations are required to reduce the norm of the fixed-point test by the factor of one tenth (1/10), as

$$\frac{1}{R} = \frac{1}{0.00922} = 108.46 \; .$$

The geometric coefficient can be also used for calculating the norm of the deviation vector (2.27). The relation between the fixed point test vector (2.28) and the deviation vector (2.27), in the case of using successive substitution (2.6) can be obtained by using (2.29) and (2.74), as

$$\bar{D}_n = \bar{T}_n / \phi$$

where

$$\phi = 1 - \mu_n$$

or

$$||D_n||_2 = ||\bar{T}_n||_2 / \phi$$

As the iterate has approached the solution, $\mu_n = 0.979$, $\phi = 0.021$, $1/\phi = 47.62$. Thus, the deviation vector is almost forty eight times bigger than the fixed point test vector. Calculating the norm of the deviation vector for the last iteration, $||\bar{T}_n||_2 = 4.45.10^{-7}$ and

$$||\bar{D}_n||_2 = 4.45 \ . \ 10^{-7} \ . \ 47.62 = 2.12 \ . \ 10^{-5}$$

This case study is ideal for using geometric extrapolation type of convergence promotion as a small number of iterations is necessary for achieving geometric progression. Five times the iterate approaches geometric progression, in iteration 7, 13, 20, 25 and 33 (See Table 6.5). Each time, equations (6.22) and (6.25) were used to approximate the solution $\bar{X}'_s$. The number of iterations that are required to achieve the solution has been reduced from two hundred and sixty-three to thirty-nine. (See Figure 6.5). Thus, the number of iterations has been reduced by factor of about 6.74, whereas the computation time was reduced by a factor of about four, from 80.3 seconds to 19.2 seconds, using the CDC 6400 computer.

TABLE 6.1

Equilibrium Coefficients

$$K_{i,j} = a + b\,T_j + c\,T_j^2 + d\,T_j^3$$

P = 290 psia

(T_j is in degrees Fahrenheit)

		Coefficients			
No.	Component	a	b × 10^2	c × 10^4	d
1	Propane	0.1995	0.510	0.0745	0.0
2	Iso-butane	0.0923	0.2215	0.0985	0.0
3	n-butane	0.200	0.0	0.1420	0.0
4	Pentanes	0.0447	0.0	0.0919	0.0

TABLE 6.2

Physical Properties of the

Components for the Enthalpy Correlations

No.	Component	Molecular Weight	Critical Pressure P_c [Atm]	Critical Temperature T_c [^oF]	Critical Compressibility Z_c [-]
1	Propane	44.1	42.0	666.3	.277
2	Iso-butane	58.1	37.4	766.0	.274
3	n-butane	58.1	37.4	766.0	.274
4	Pentanes	72.1	32.6	846.5	.269

TABLE 6.3

The Initial and Final Values of the

Temperature and Liquid Flow Profiles

Tray j	Temperature Profile $\{T_j\}$	$[F^O]$	Liquid Flow Profile $\{L_j\}$[lb-mole/hr]	
	Initial	Final	Initial	Final
0	75.0	75.0	300.0	300.0
1	138.1	132.3	416.9	417.2
2	144.5	132.8	416.9	416.3
3	151.0	133.5	416.9	414.9
4	157.4	134.8	416.9	412.7
5	163.9	136.7	416.9	409.5
6	170.3	139.6	416.9	405.0
7	176.7	143.8	416.9	399.1
8	183.2	149.7	416.9	392.2
9	189.6	157.3	416.9	384.8
10	196.1	166.2	416.9	377.6
11	202.5	176.1	416.9	370.2
12	209.0	187.1	416.9	361.6
13*	215.4	199.8	605.8	598.9
14	217.5	204.1	605.8	600.4
15	219.7	208.4	605.8	602.3
16	221.8	212.5	605.8	604.6
17	223.9	216.2	605.8	607.0
18	226.1	219.4	605.8	609.4
19	228.2	222.1	605.8	611.6
20	230.3	224.4	605.8	613.5
21	232.5	226.3	605.8	615.1
22	234.6	227.8	605.8	616.5
23	236.7	229.1	605.8	617.4
24	238.9	230.3	605.8	618.1
25	241.0	231.4	605.8	618.3
26	243.1	232.7	605.8	617.9
27	245.3	234.5	605.8	616.3
28	247.4	237.5	605.8	612.8
29	249.5	242.6	605.8	606.9
30	251.7	251.5	150.0	150.0

* Feed tray

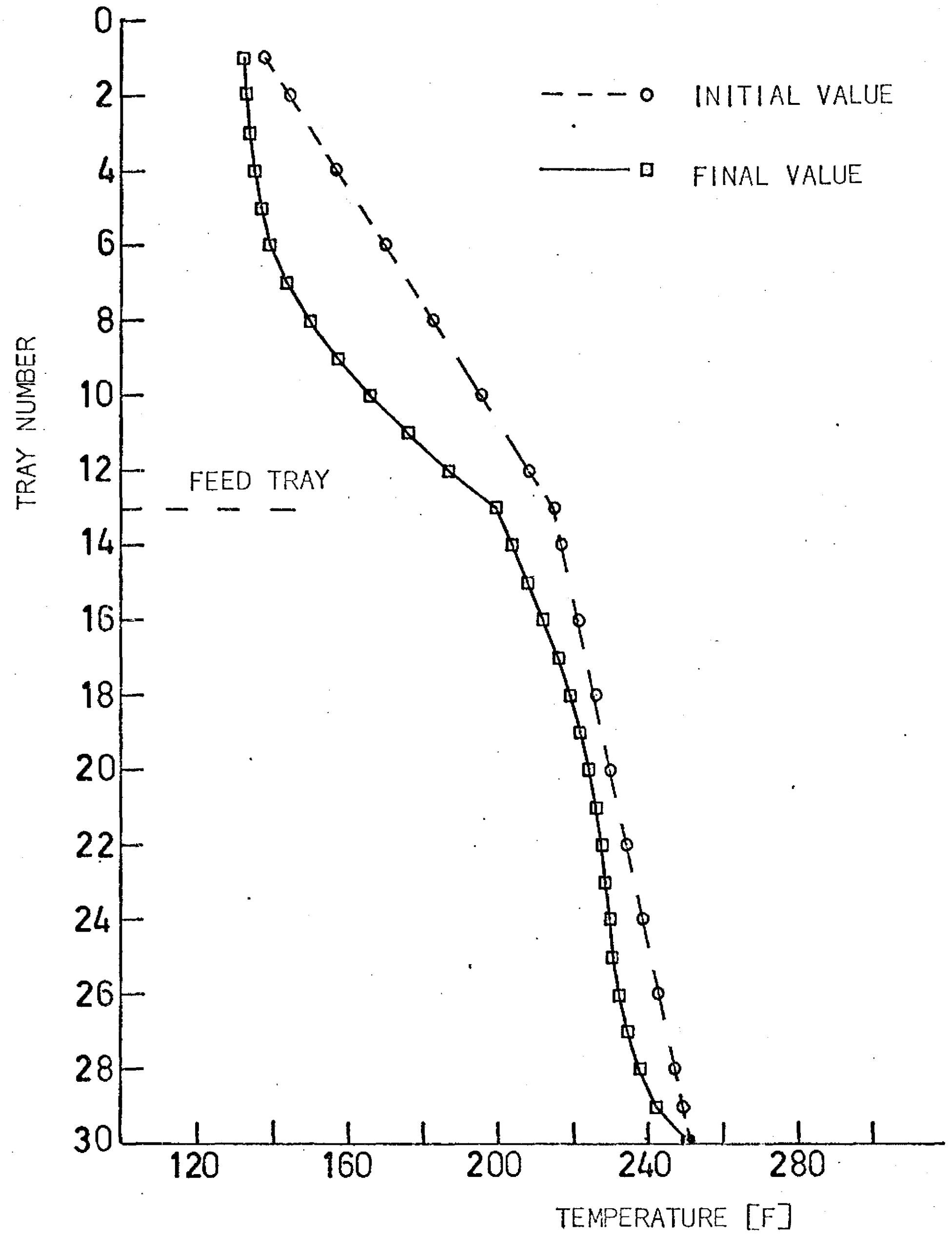

FIGURE 6.3 TEMPERATURE PROFILES

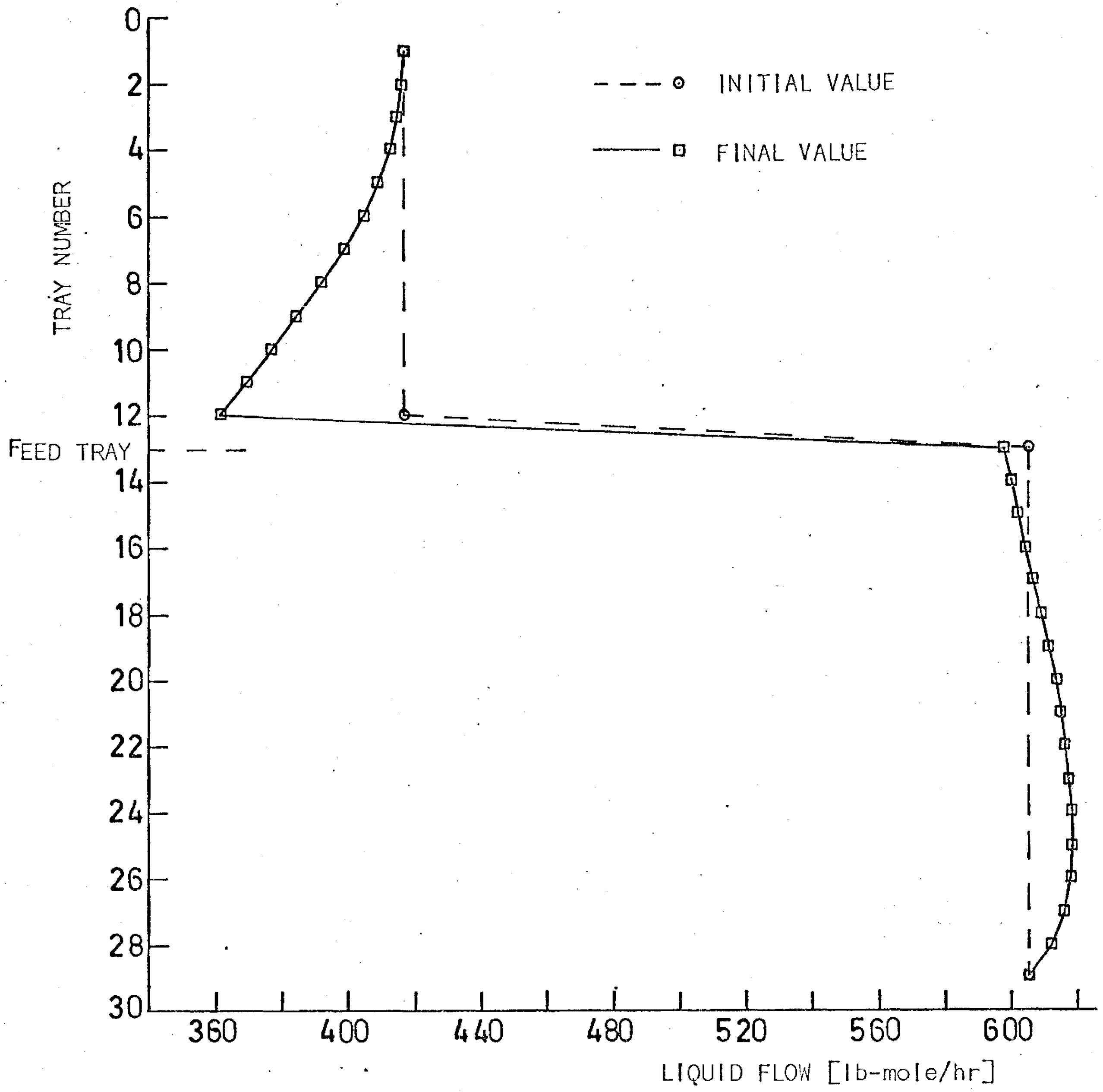

FIGURE 6.4 LIQUID FLOW PROFILES

TABLE 6.4

Results from the Iterations

Applying Geometric Extrapolation for Promoting the Convergence

	Iterate Without Convergence Promotion		Iterate With Convergence Promotion	
Iteration Number	Norm of the Fixed Point Test Vector	Geometric Coefficient	Norm of the Fixed Point Test Vector	Geometric Coefficient
	$\|\|\bar{T}_n\|\|_2$	μ_n	$\|\|\bar{T}_n\|\|_2$	μ_n
1	4.838 E-3	-	4.838 E-3	-
2	4.609 E-4	.095	4.609 E-4	.095
3	5.546 E-4	1.203	5.546 E-4	1.203
4	5.483 E-4	.989	5.483 E-4	.989
5	4.881 E-4	.890	4.881 E-4	.890
6	4.280 E-4	.877	4.280 E-4	.877
7	3.783 E-4	.884	3.783 E-4	.884*
8	3.382 E-4	.894	1.689 E-3	-
9	3.053 E-4	.903	1.693 E-4	.100
10	2.779 E-4	.910	1.286 E-4	.760
11	2.548 E-4	.917	1.199 E-4	.932
12	2.350 E-4	.922	1.142 E-4	.952
13	2.179 E-4	.927	1.091 E-4	.956*
14	2.029 E-4	.931	6.598 E-4	-
15	1.896 E-4	.935	9.497 E-5	.144
16	1.779 E-4	.938	3.954 E-5	.416
17	1.674 E-4	.941	3.362 E-5	.850
18	1.579 E-4	.944	3.221 E-5	.958
19	1.494 E-4	.946	3.132 E-5	.973
20	1.416 E-4	.948	3.053 E-5	.975*
21	1.345 E-4	.950	2.416 E-4	-
22	1.280 E-4	.952	1.969 E-5	.082
23	1.220 E-4	.953	9.517 E-6	.483
24	1.165 E-4	.955	8.654 E-6	.909
25	1.114 E-4	.956	8.400 E-6	.971
26	1.066 E-4	.957	8.211 E-6	.978*
27	1.022 E-4	.958	1.047 E-4	-
28	9.802 E-5	.959	1.359 E-5	.130
29	9.414 E-5	.960	2.723 E-6	.200
30	9.051 E-5	.961	1.892 E-6	.695
31	8.709 E-5	.962	1.797 E-6	.950
32	8.387 E-5	.963	1.754 E-6	.976
33	8.084 E-5	.964	1.717 E-6	.979*
34	7.797 E-5	.965	7.219 E-6	-
35	7.526 E-5	.965	1.217 E-6	.169

* applying convergence promotion

TABLE 6.4 Cont'ed

	Iterate Without Convergence Promotion		Iterate With Convergence Promotion	
Iteration Number	Norm of the Fixed Point Test Vector	Geometric Coefficient	Norm of the Fixed Point Test Vector	Geometric Coefficient
	$\|\|\bar{T}_n\|\|_2$	μ_n	$\|\|\bar{T}_n\|\|_2$	μ_n
36	7.268 E-5	.966	5.221 E-7	.429
37	7.024 E-5	.966	4.704 E-7	.901
38	6.792 E-5	.967	4.569 E-7	.971
39	6.572 E-5	.967	4.471 E-7	.978
40	6.361 E-5	.968		
45	5.443 E-5	.970		
50	4.702 E-5	.972		
60	3.581 E-5	.974		
70	2.780 E-5	.976		
80	2.185 E-5	.977		
90	1.733 E-5	.977		
100	1.383 E-5	.978		
110	1.108 E-5	.978		
120	8.915 E-6	.979		
130	7.187 E-6	.979		
140	5.806 E-6	.979		
150	4.696 E-6	.979		
160	3.803 E-6	.979		
170	3.082 E-6	.979		
180	2.499 E-6	.979		
190	2.028 E-6	.979		
200	1.646 E-6	.979		
210	1.337 E-6	.979		
220	1.086 E-6	.979		
230	8.822 E-7	.979		
240	7.169 E-7	.979		
250	5.826 E-7	.979		
260	4.735 E-7	.979		
263	4.450 E-7	.979		

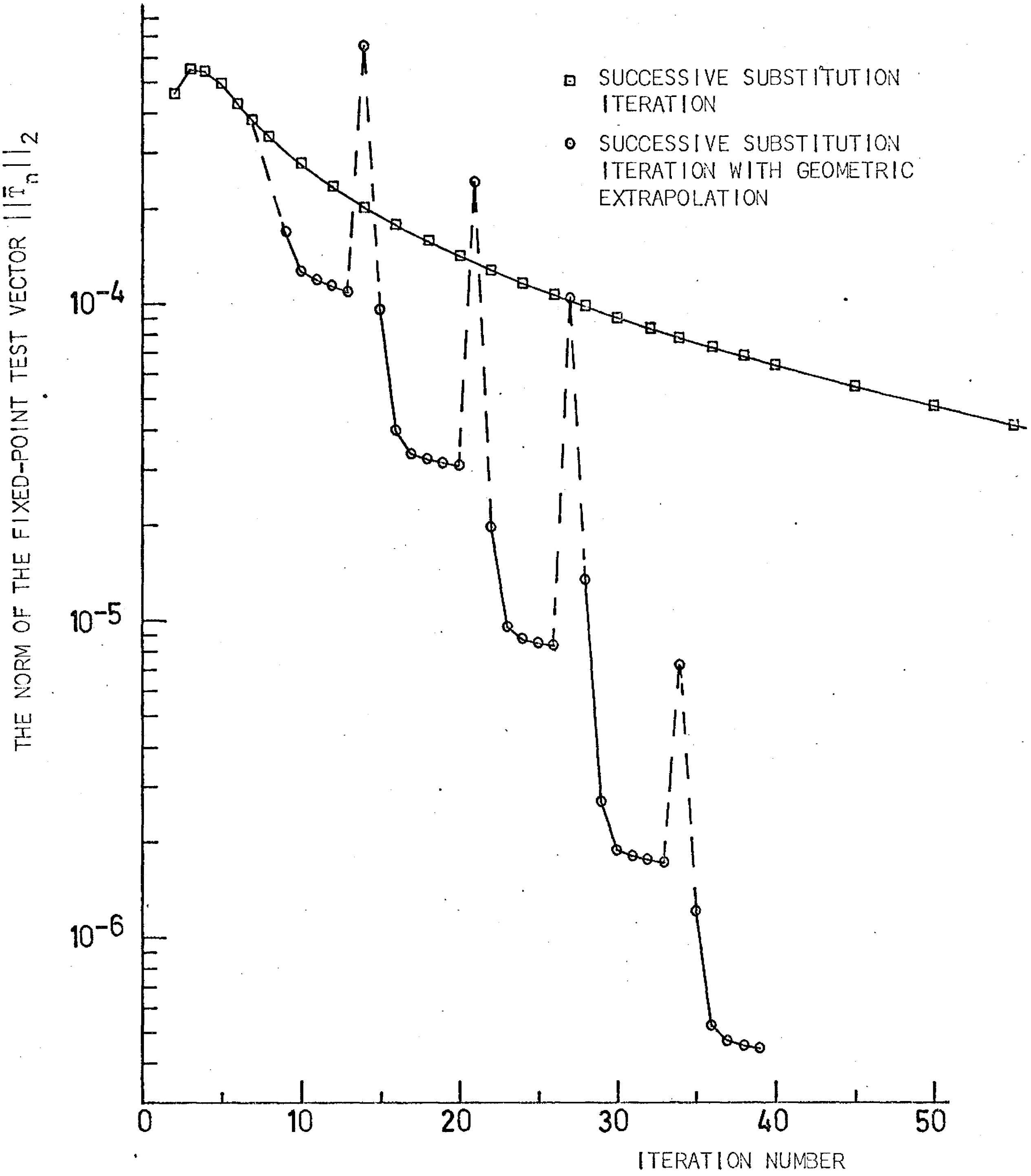

FIGURE 6.5 RESULTS FROM THE ITERATIONS (6.17) APPLYING GEOMETRIC EXTRAPOLATION FOR PROMOTING THE CONVERGENCE

TABLE 6.5

The Composition Profiles at the Last Iteration

Composition Profile $\{x_{i,j}\}$
[mole fraction]

Tray	Propane	Iso-butane	n-butane	pentanes
j	C_3	iC_4	$N\ C_4$	C_5
0	.9944	.0054	.0002	.0000
1	.9899	.0097	.0005	.0000
2	.9825	.0165	.0010	.0000
3	.9708	.0271	.0021	.0000
4	.9524	.0435	.0041	.0000
5	.9240	.0681	.0079	.0000
6	.8818	.1033	.0149	.0001
7	.8224	.1505	.0268	.0003
8	.7447	.2083	.0460	.0009
9	.6518	.2710	.0741	.0030
10	.5518	.3283	.1108	.0090
11	.4542	.3688	.1528	.0241
12	.3665	.3822	.1932	.0580
13*	.2922	.3626	.2214	.1238
14	.2510	.3907	.2325	.1258
15	.2106	.4181	.2436	.1276
16	.1728	.4435	.2543	.1293
17	.1389	.4659	.2644	.1308
18	.1097	.4846	.2737	.1320
19	.0852	.4995	.2822	.1331
20	.0653	.5104	.2903	.1340
21	.0495	.5176	.2981	.1348
22	.0371	.5212	.3060	.1357
23	.0275	.5213	.3144	.1367
24	.0202	.5177	.3235	.1385
25	.0147	.5100	.3337	.1417
26	.0105	.4970	.3445	.1480
27	.0073	.4768	.3548	.1610
28	.0050	.4460	.3615	.1875
29	.0032	.3995	.3579	.2395
30	.0019	.3315	.3333	.3333

* Feed tray

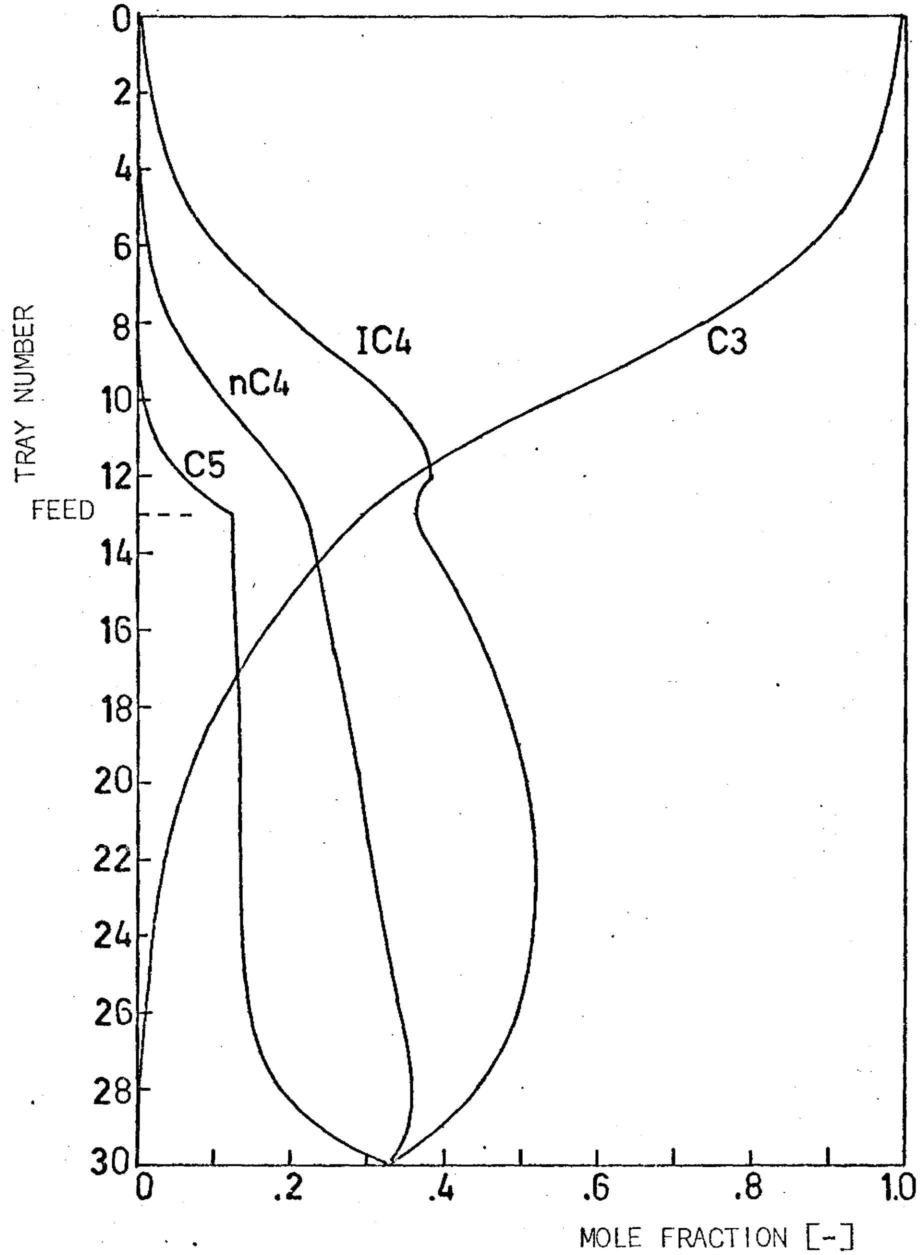

FIGURE 6.6 THE COMPOSITION PROFILES

AT THE TEST ITERATION

6.4 Solving Distillation Columns as a Reduced Set of Functions

The successive substitution method of iteration can be replaced by a method which would find the temperature and the liquid flow profiles, $\{T_j\}$ and $\{L_j\}$, such that equations (6.11) and (6.12) are satisfied. Thus, instead of using equations (6.11) and (6.12) for calculating new temperature and liquid flow profiles, the residual values of these equations are taken as a set of functions which ought to be solved.

The modified form of Newton's method was utilized successfully to solve this set of functions. In addition, geometric extrapolation was applied for convergence promotion.

An attempt was made to solve this set of equations by the Quasi-Newton method (See Section 4.4) but the result was unsatisfactory. Thus, if pure distillate is required as in the case study discussed in Section 6.3.3, the Quasi-Newton method appears to be unstable although the modified form of Newton's method easily overcame this difficulty.

A similar technique for solving multicomponent distillation columns was presented by Tomich (T1) where the set of functions which he proposed was solved by Broyden's method (B2) (Quasi-Newton). Tomich did not give much detail on his case studies and results.

6.4.1 The Computational Procedure for the Function Evaluation

The computational procedure for the function evaluation is very similar to the previous method. For any set of $\{T_j\}$ and $\{L_j\}$, it is possible to calculate the vapour flow profile $\{V_j\}$ by using equation (6.10), then evaluating matrix $[A_{Bc}]$ (6.9) and solving equation (6.13) for each component in order to get $\{x_{i,j}\}$ (See Figure 6.7).

Now the temperature profile $\{T_j\}$, the liquid and vapour flow profiles, $\{L_j\}$ and $\{V_j\}$ and the component concentration profiles $\{x_{i,j}\}$ can be utilized for evaluating the residual values of equations (6.11) and (6.12). Hence, these equations depend only on the temperature and liquid flow profiles and may be expressed as

$$S_j(\{T_\ell\}, \{L_k\}) = \sum_{i=1}^{Nc} K_{ij}\, x_{ij} - 1 \qquad (6.28)$$

$$j = 1, 2, 3, \ldots, NR$$

and

$$\Pi_j(\{T_\ell\}, \{L_k\}) = L_j(h_j - H_{j+1}) - D(H_{j+1} - H_D) - U_j \qquad (6.29)$$

$$j = 1, 2, 3, \ldots, (NR-1).$$

where

$$U_j = -Q_o \qquad \text{for} \quad 1 \leq j \leq N1-1$$

$$U_j = -Q_o + V_f(H_{v,f} - H_{j+1}) \qquad \text{for} \quad j = N1$$

$$U_j = -Q_o + F(\bar{H}_f - H_{j+1}) \qquad \text{for} \quad N2 \leq j \leq NR-1$$

The two sets of residual functions (6.28) and (6.29) are not of the same magnitude. The value of the first set $\{S_j\}$ is about unity whereas the second set $\{\Pi_j\}$ has a magnitude of millions of BTU's. Therefore it is desirable to normalize the two sets. The first one, $\{S_j\}$ may be divided by unity and the second set, $\{\Pi_j\}$ by the amount of heat which flows to the stage in the vapour phase, $(V_{j+1}\ H_{j+1})$. Thus, equation (6.29) becomes

$$\Pi_j(\{T_\ell\},\ \{L_k\}) = [L_j(h_j - H_{j+1}) - D(H_{j+1} - H_D) - U_j]/(V_{j+1}\ H_{j+1}) \qquad (6.30)$$

$$j = 1, 2, 3, \ldots, (NR-1)$$

In addition, as equation (6.28) was derived from equation (6.5b), it is worthwhile mentioning that equation (6.5a) can also be used to obtain the residual functions as

$$S_j^! \ (\{T_\ell\},\ \{L_k\}) = \sum_{i=1}^{Nc} x_{i,j} - 1 \qquad (6.31)$$

$$j = 1, 2, 3, \ldots, NR$$

Now, locating the sets of the residual functions (6.28) and (6.30) in a function vector $\bar{f}$, as

$$\bar{f}^T = [\{S_\ell\},\ \{\Pi_k\}]^T \qquad (6.32)$$

$$\ell = 1, 2, 3, \ldots, NR$$

$$k = 1, 2, 3, \ldots, NR-1$$

and arranging the temperature and liquid flow profiles in vector $\bar{X}$ as

$$\bar{X}^T = [\ \{T_\ell\},\ \{L_k\}\ ]^T \qquad (6.33)$$

$$\ell = 1, 2, 3, \ldots, NR$$

$$k = 1, 2, 3, \ldots, NR-1$$

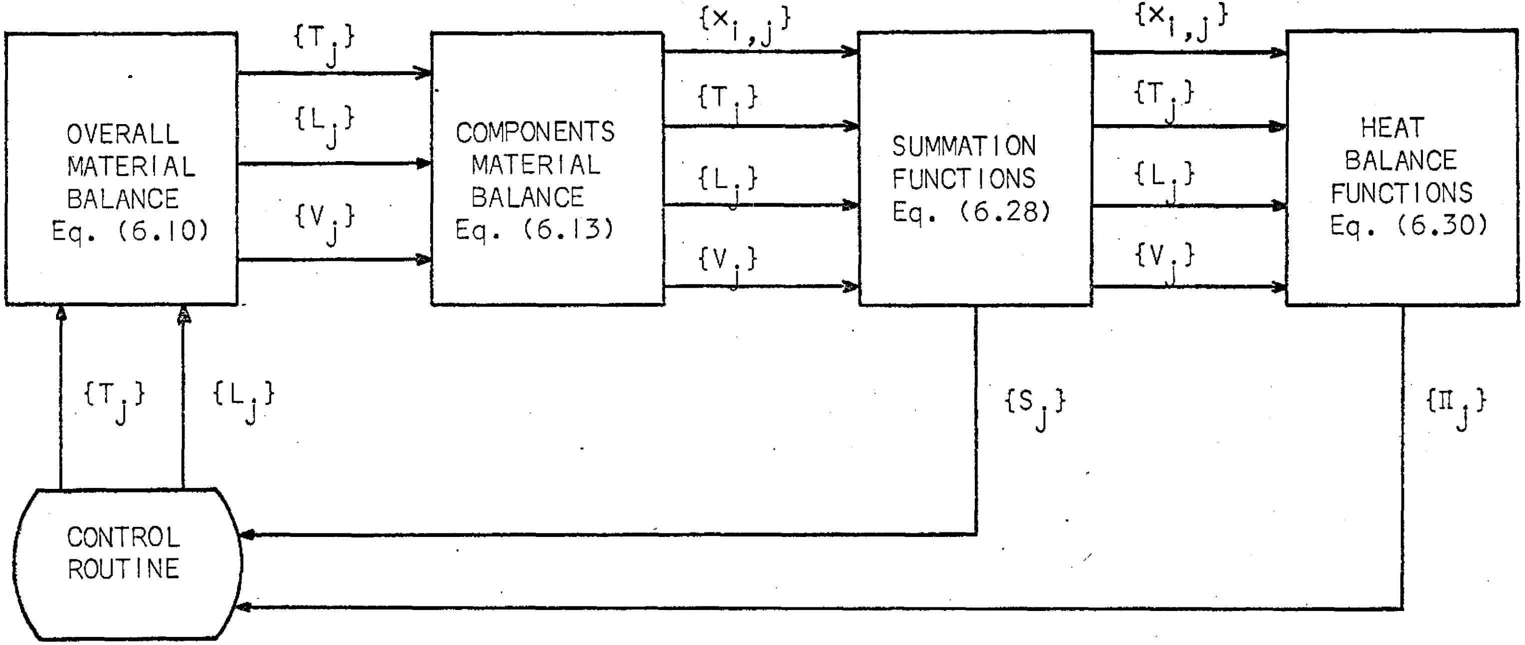
OVERALL MATERIAL BALANCE Eq. (6.10)
{T_j}
{L_j}
{V_j}
COMPONENTS MATERIAL BALANCE Eq. (6.13)
{x_i,j}
{T_j}
{L_j}
{V_j}
SUMMATION FUNCTIONS Eq. (6.28)
{x_i,j}
{T_j}
{L_j}
{V_j}
HEAT BALANCE FUNCTIONS Eq. (6.30)
{T_j}
{L_j}
{S_j}
{Π_j}
CONTROL ROUTINE

FIGURE 6.7 INFORMATION FLOW DIAGRAM

THE FUNCTIONS EVALUATION PROCEDURE

the distillation column problems have been reduced to a problem of solving a set of non-linear algebraic equations in form of

$$f_i\ (\bar{X}) = \bar{0} \qquad (6.34)$$

$$i = 1, 2, 3, \ldots, m.$$

where $m = 2.NR-1$

Solution techniques for set of non-linear algebraic equations have been discussed in Section 4.

6.4.2 The Solution Procedure

The modified form of Newton's method has been applied to solve the set of functions (6.34) for the same case study described in Section 6.3.3. In addition, the geometric extrapolation technique has been employed for promoting the convergence.

The modified form of Newton's method can be written as follows

$$\bar{X}_{n+1} = \bar{X}_n + t.(-\bar{\bar{J}}_o)^{-1}\ \bar{f}(\bar{X}_n) \qquad (6.35)$$

The Jacobian matrix $\bar{\bar{J}}_o$, of partial derivatives of the function vector,

$$(J_{ij})_o = (\frac{\partial f_i}{\partial x_j})_o$$

has been evaluated only once around the point $\bar{X}_o$ and used unchanged throughout the calculation. The Jacobian matrix has been calculated numerically by finite difference approximation for which, m+1 functional evaluations are necessary, where m is the number of elements in the vectors $\bar{f}$ and $\bar{X}$, ($m = 2*NR-1$, for column with total condenser). The "jth" column of the Jacobian matrix may be obtained by the differences between the function vector at points $(\bar{X}_o + \bar{e}_j\delta)$ and $(\bar{X}_o)$ as

$$\left(\frac{\partial f_i}{\partial x_j} \right)_o \simeq \frac{f_i(\bar{X}_o + \bar{e}_j\delta) - f_i(\bar{X}_o)}{\delta} . \qquad i = 1, 2, 3, \ldots, m. \tag{6.36}$$

where δ is a small deviation from $(x_j)_o$ which in the following case study was chosen as $\delta = 0.01.(x_j)_o$. $\bar{e}_j$ is a vector with unity in the "jth" place and zero elsewhere.

At the end of each functional evaluation, the norm of the function vector Φ_n and the ratio between two successive norms, Φ_n and Φ_{n-1}, are obtained, as

$$\Phi_n = ||\bar{f}(\bar{X}_n)||_2 \tag{6.37}$$

and

$$\mu_{fn} = \frac{\Phi_n}{\Phi_{n-1}} \tag{6.38}$$

respectively. The iteration (6.35) would continue until the norm (6.37) is less than a specified tolerance ε_f, as

$$\Phi_n = ||\bar{f}(\bar{X}_n)||_2 \leq \varepsilon_f \tag{6.39}$$

Thus, the indicator for convergence in this case is not the fractional change of the iteration $\bar{X}_n$, but the value of the functions themselves. Therefore, it is much more reliable an indicator for convergence. In addition, the iteration (6.35) may be accelerated by the geometric extrapolation method if it approaches geometric progression. Now if the μ_{fn} approach a constant value, it can be shown that the iterate

$\bar{X}_n$ approaches a geometric progression in the form of equation (2.50). It also can be shown that if the μ_{fn} become constant value then

$$\mu_{fn} = \frac{||\bar{f}_n||}{||\bar{f}_{n-1}||} = \frac{||\bar{X}_{n+1} - \bar{X}_n||}{||\bar{X}_n - \bar{X}_{n-1}||} \simeq \frac{||\bar{\theta}_n||}{||\bar{\theta}_{n-1}||} = \mu_n$$

(See Section 2.4.1)

The geometric extrapolation can be used in the same form as in Section 6.3.2 with minor changes. Instead of using equation (6.25) B_i may be calculated as

$$B_i = \left\{ \frac{(x_{n-2} - x_{n-1})\, f_{n-1}}{(f_{n-1} - f_{n-2})} \right\}_i \qquad (6.40)$$

where $f_{n,i}$ is the i^{th} element of vector $\bar{f}(\bar{X}_n)$.

6.4.3 The Results

For the same case study described in Section 6.3.3 the modified form of Newton's method was utilized for solving the equations. The column has 30 trays, including the reboiler, and the number of equations which we have to solve is fifty-nine (m = 59). Sixty functional evaluations have been used to obtain the Jacobian matrix (6.36) and then, after calculating the Jacobian inverse, the iteration (6.35) has been proceeded until the norm of the functions satisfies (6.39). The relaxation factor t (6.35) has been chosen, in this case study, as t = 0.9.

The initial and the final values of the temperature and liquid flow profiles are given in Table 6.3 and Figures 6.3 and 6.4. The norm of the function vector versus the number of iterations is presented in

Table 6.6 and Figure 6.7. The geometric coefficients μ_{fn} are also given in Table 6.6 where it can easily be seen that the geometric coefficient has approached an almost constant value after eight iterations.

The rate of convergence R has been calculated by equation (2.77) as

$$R = -\log \mu_f \simeq -\log 0.844 = 0.07366$$

and $\frac{1}{R}$ = 13.58 . Thus, almost fourteen iterations are required to reduce the norm of the functions (6.37) by one tenth (1/10).

As the geometric coefficient has approached constant value in a small number of iterations, geometric extrapolation was applied to achieve faster convergence. The results of the iteration with convergence promotion are given in Table 6.6 and Figure 6.7. The number of iterations was reduced from 45 to 27 due to the convergence promotion, an improvement of forty percent. The tolerance γ, equation (2.57), was chosen as 0.005.

As it was discussed previously, equation (6.31) can be used instead of equation (6.28). Thus, for demonstration, the calculation was repeated using equation (6.31) and the results, which are very similar to those in the first case, are given in Table 6.7 and Figure 6.8.

TABLE 6.6

Results from the Iteration (6.35)
Solving Equations (6.28) and (6.30)

Iteration Number	Functional Evaluations-Number	Iteration Without Convergence Promotion		Iteration With Convergence Promotion	
		$\|\|f_n\|\|_2$	μ_{fn}	$\|\|f_n\|\|_2$	μ_{fn}
				1.049 E-3	.192
1	61	1.049 E-3	.192	6.722 E-4	.641
2	62	6.722 E-4	.641	5.675 E-4	.844
3	63	5.675 E-4	.844	4.608 E-4	.812
4	64	4.608 E-4	.812	3.805 E-4	.826
5	65	3.805 E-4	.826	3.166 E-4	.832
6	66	3.166 E-4	.832	2.654 E-4	.838
7	67	2.654 E-4	.838	2.234 E-4	.842
8	68	2.234 E-4	.842	1.887 E-4	.845*
9	69	1.887 E-4	.845	6.208 E-4	-
10	70	1.597 E-4	.846	9.899 E-5	.160
11	71	1.353 E-4	.847	3.791 E-5	.383
12	72	1.146 E-4	.847	3.195 E-5	.843
13	73	9.714 E-5	.848	2.764 E-5	.865
14	74	8.231 E-5	.847	2.358 E-5	.853
15	75	6.973 E-5	.847	1.998 E-5	.847
16	76	5.905 E-5	.847	1.689 E-5	.846*
17	77	4.999 E-5	.847	9.556 E-6	-
18	78	4.230 E-5	.846	4.234 E-6	.443
19	79	3.578 E-5	.846	3.510 E-6	.829
20	80	3.026 E-5	.846	2.996 E-6	.854
21	81	2.558 E-5	.845	2.540 E-6	.848
22	82	2.162 E-5	.845	2.146 E-6	.845*
23	83	1.827 E-5	.845	8.302 E-7	-
24	84	1.543 E-5	.845	5.317 E-7	.640
25	85	1.303 E-5	.845	4.497 E-7	.846
26	86	1.100 E-5	.844	3.823 E-7	.850
27	87	9.290 E-6	.844		
28	88	7.843 E-6	.844		
29	89	6.620 E-6	.844		
30	89	5.587 E-6	.844		
32	92	3.979 E-6	.844		
34	94	2.833 E-6	.844		
36	96	2.017 E-6	.844		
38	98	1.436 E-6	.844		
40	100	1.022 E-6	.844		
42	102	7.274 E-7	.844		
44	104	5.177 E-7	.844		
45	105	4.367 E-7	.844		

* applying geometric extrapolation

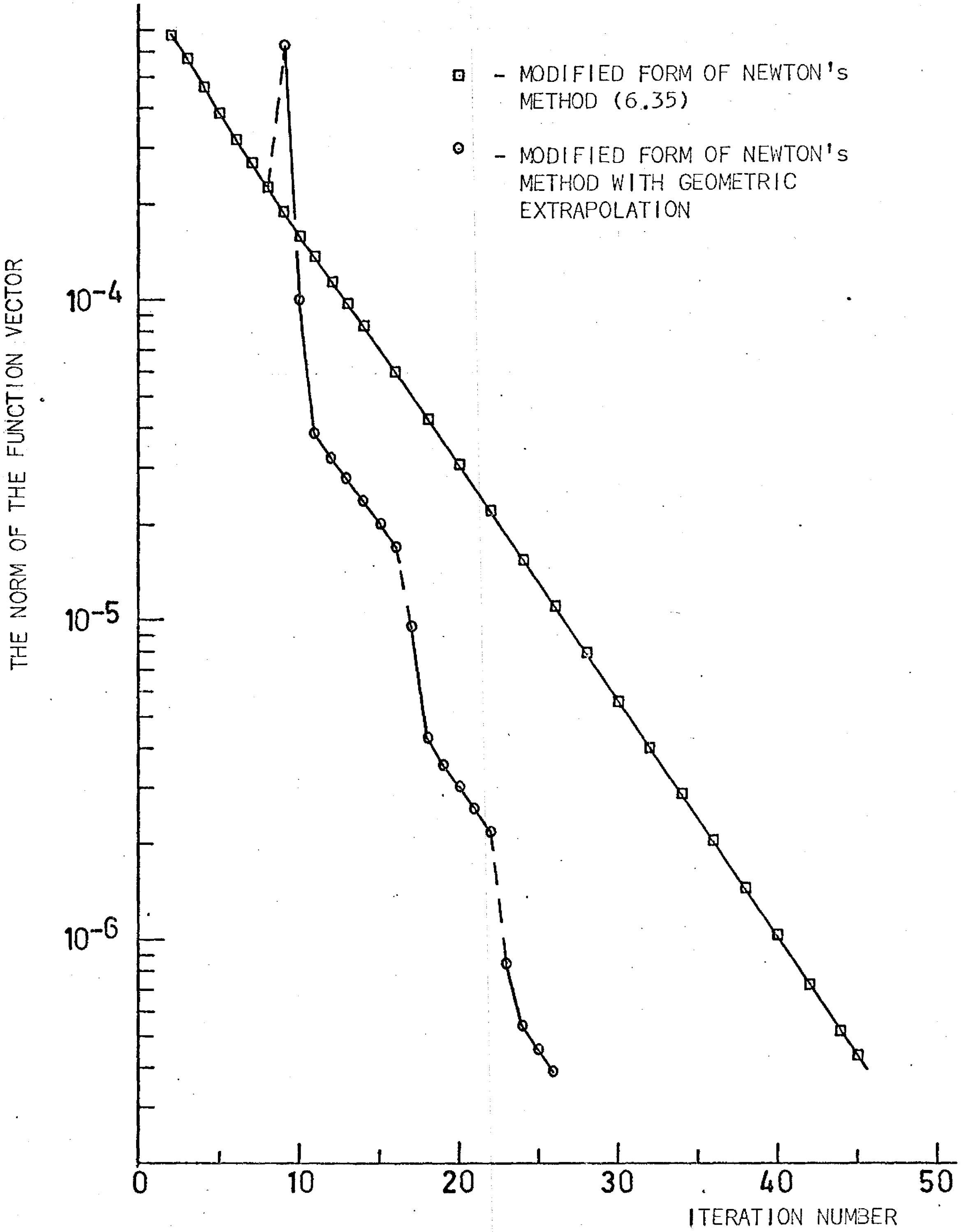

FIGURE 6.8 RESULTS FROM THE ITERATION (6.35)
SOLVING EQUATIONS (6.28) and (6.30)

TABLE 6.7

Results from the Iteration (6.35)

Solving Equations (6.31) and (6.30)

Iteration Number	Functional Evaluations-Number	Iteration Without Convergence Promotion		Iteration With Convergence Promotion	
		$\|\|f_n\|\|_2$	μ_{fn}	$\|\|f_n\|\|_2$	μ_{fn}
1	61	2.972 E-3	.407	2.972 E-3	.407
2	62	2.544 E-3	.856	2.544 E-3	.856
3	63	2.159 E-3	.848	2.159 E-3	.848
4	64	1.813 E-3	.840	1.813 E-3	.840
5	65	1.531 E-3	.845	1.531 E-3	.845
6	66	1.297 E-3	.847	1.297 E-3	.847
7	67	1.101 E-3	.849	1.101 E-3	.849
8	68	9.353 E-4	.850	9.353 E-4	.850*
9	69	7.948 E-4	.850	4.289 E-3	-
10	70	6.751 E-4	.849	1.234 E-3	.289
11	71	5.731 E-4	.849	4.570 E-4	.370
12	72	4.861 E-4	.848	3.090 E-4	.676
13	73	4.119 E-4	.847	2.441 E-4	.790
14	74	3.487 E-4	.847	2.002 E-4	.820
15	75	2.949 E-4	.846	1.661 E-4	.830
16	76	2.492 E-4	.845	1.385 E-4	.834
17	77	2.104 E-4	.844	1.158 E-4	.836
18	78	1.775 E-4	.844	9.687 E-5	.837*
19	79	1.497 E-4	.843	3.480 E-5	-
20	80	1.262 E-4	.843	2.735 E-5	.786
21	81	1.063 E-4	.842	2.262 E-5	.827
22	82	8.945 E-5	.842	1.892 E-5	.837
23	83	7.528 E-5	.842	1.586 E-5	.838
24	84	6.333 E-5	.841	1.331 E-5	.839*
25	85	5.326 E-5	.841	4.908 E-6	-
26	86	4.478 E-5	.841	3.669 E-6	.748
27	87	3.765 E-5	.841	2.990 E-6	.815
28	88	3.164 E-5	.841	2.484 E-6	.831
29	89	2.659 E-5	.840	2.077 E-6	.836
30	90	2.234 E-5	.840	1.741 E-6	.838
31	91	1.877 E-5	.840		
32	92	1.577 E-5	.840		
33	93	1.325 E-5	.840		
34	94	1.113 E-5	.840		
36	96	7.850 E-6	.840		
38	98	5.537 E-6	.840		
40	100	3.905 E-6	.840		
42	102	2.754 E-6	.840		
44	104	1.942 E-6	.840		
46	106	1.369 E-6	.840		

* applying geometric extrapolation

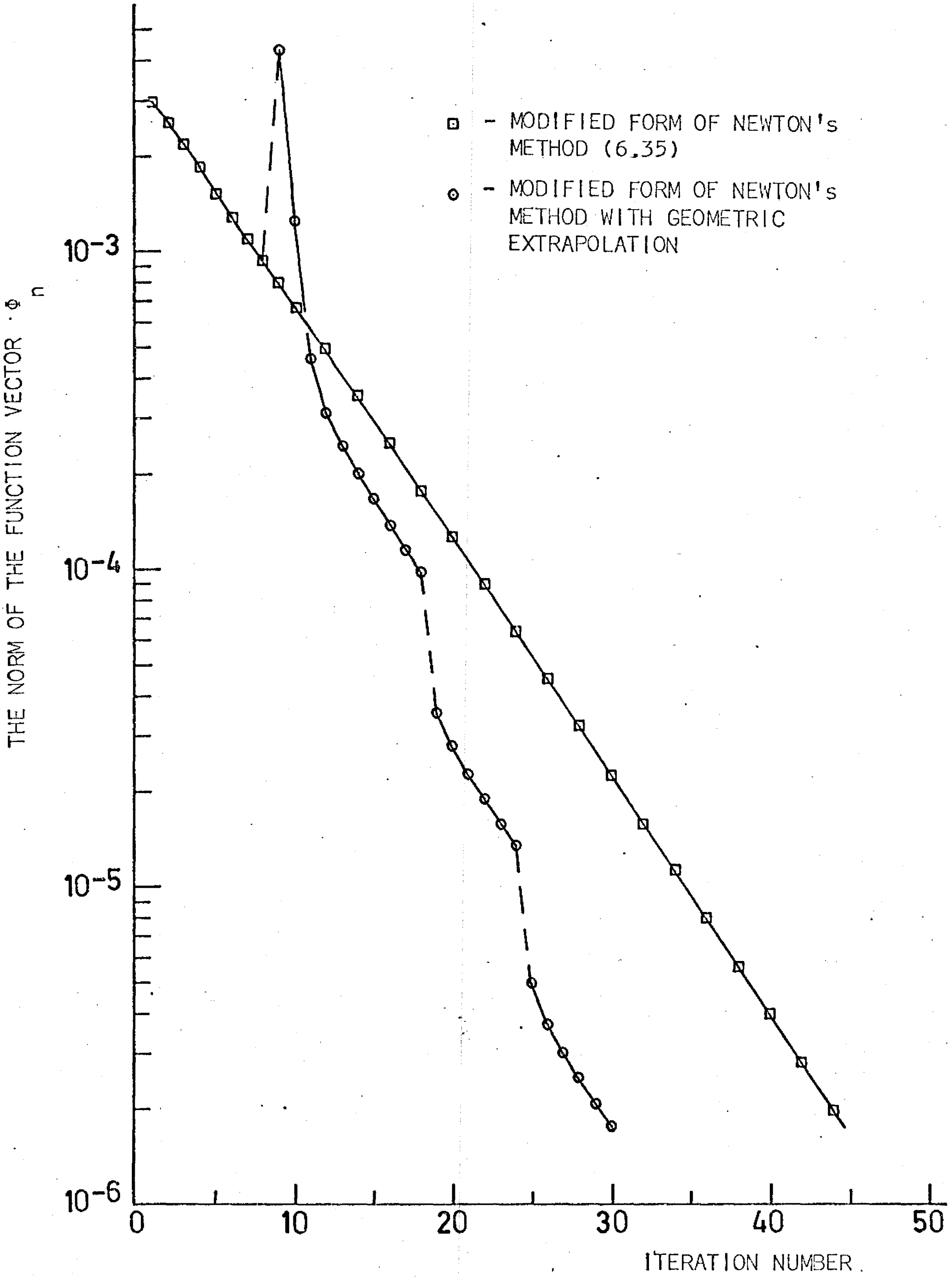

FIGURE 6.9 RESULTS FROM THE ITERATION (6.35)

SOLVING EQUATIONS (6.31) and (6.30)

6.4.4 Conclusions and Remarks

The technique for solving distillation columns as a reduced set of equations has been demonstrated.

The modified form of Newton's method has been utilized to seek the solution. The iteration (6.35) appears to be very stable with a rate of convergence almost eight times faster than the results obtained by successive substitution (Section 6.3).

The disadvantage of this method is the necessity of evaluating the Jacobian matrix $\bar{\bar{J}}_o$ prior to the iteration (6.35). However, the Jacobian matrix may be used more than once, under various operating conditions of the distillation column. Hence, if the performance of the distillation column is studied for different feed composition, distillate rate and reflux ratio, and if the variation of these parameters is sufficiently small, the Jacobian matrix which was obtained for the base case condition can be used unchanged.

In addition, we may point out, that the equilibrium ratio (K_{ij}) is evaluated fewer times for every iteration than in successive substitution (Section 6.3). This is due to the fact that in successive substitution the set of equations (6.14) is solved for every iteration. Thus, the present technique will be very suitable when more comprehensive methods for evaluating equilibrium ratio (P3) are implemented.

7. SUMMARY AND CONCLUSIONS

This thesis has presented the mathematical analysis of iterations which result mainly from the calculation of recycle processes of chemical plants.

The formulation of the iteration, the existence of a solution and the convergence conditions have been discussed. In addition, linear iteration and its asymptotic behaviour were studied. It was shown that linear iteration usually approaches a geometric progression whereby the geometric coefficient is the largest eigenvalue in modulus. It was also demonstrated, in the case studies, that the iteration generally approached a geometric progression. This unique behaviour of the iteration was utilized for determining the relation between the test and the deviation vectors and, more important, for approximating the solution.

The geometric extrapolation technique for accelerating the convergence has been developed and demonstrated. This technique appears to be very powerful for a case of slow convergence, particularly when the absolute value of the geometric coefficient is greater than 0.9. However, it is a suitable technique only when the geometric progression is achieved in a small number of iterations.

The application of the "full matrix" technique for accelerating the convergence was demonstrated in the simulation of the Alkylation unit. Very stable and relatively fast convergence was achieved. However, (m+l) iterations were necessary for evaluating the matrix of the convergence promotion coefficients, where m is the number of variables.

Two methods for solving the multi-component distillation column were given. In the first method, successive substitution was utilized to seek the solution, and since the convergence was slow, the geometric extrapolation technique was applied to promote the convergence. Thus, the number of iterations was reduced by a factor of about six.

In the second method, the distillation column was represented by a set of algebraic equations and the modified form of Newton's method was employed to seek the solution.

The modified form of Newton's method, which is iterative in form, appears to be very stable. The results obtained showed rates of convergence which were almost eight times faster than those obtained by successive substitution. Unsuccessful attempts were made to solve this set of equations by the Quasi-Newton method. A further examination of this method is recommended.

NOMENCLATURE

a_{ij}	element of matrix $\bar{\bar{A}}$.
$\bar{\bar{A}}$	m x m matrix of linear coefficients (2.30)
$\bar{b}$	m x 1 vector of linear coefficients (2.30)
$\bar{\bar{B}}$	matrix defined by equation (3.4)
C_j	coefficient defined by equation (2.40)
$\bar{D}$	deviation vector (2.27)
$\bar{E}$	error vector (2.52)
$\bar{f}$	function vector (4.1)
$\bar{F}$	function vector (2.2)
F_i	element i of the function vector (2.1)
F_{ij}	partial derivatives of the function vector (2.19)
F	feed flow rate
g	single iterative function (2.23)
$\bar{\bar{G}}$	matrix of convergence promotion coefficients
h_j	enthalpies of the liquid stream L_j
H_j	enthalpies of the vapour stream V_j
$\bar{\bar{H}}$	approximation to the Jacobian inverse
I	unit matrix
$\bar{\bar{J}}$	Jacobian matrix
K_{ij}	equilibrium ratio of component i at tray j
L_j	liquid flow from "jth" tray
L_f	liquid fraction of the feed (6.10)

m	number of variables
M	Lipschitz constant (2.9)
n	iteration number
NR	number of trays including the reboiler
$\bar{P}$	correction vector (4.7)
Q_o, Q_{NR}	condenser and reboiler cooling and heating loads respectively
r	interval
R	rate of convergence (2.77)
R	reflux ratio
T_j	temperature at the "jth" tray
$\bar{U}$	vector of coefficients (2.50)
V_f	vapour fraction of the feed (6.10)
V_j	vapour flow from "jth" tray
$\bar{W}$	eigenrow of matrix $\bar{\bar{A}}$
$\bar{X}$	variable vector ("input" vector)
$\bar{X}_s$	solution point
x_i	element i of vector $\bar{X}$
x_{ij}	mole fraction of i component in liquid phase at tray j
$\bar{Y}$	variable vector ("output" vector)
y_i	element i of vector $\bar{Y}$
y_{ij}	mole fraction i component in vapour phase at tray j
$\bar{Z}$	eigenvector of matrix $\bar{\bar{A}}$

Greek Symbols

α	convergence promotion coefficient
$\bar{\alpha}$	vector of convergence promotion coefficients
β	tolerance for the deviation vector
γ	tolerance for the fixed point test vector
$\bar{\Theta}$	test vector (2.26)
λ	eigenvalue
λ_k	biggest eigenvalue
Λ	fractional change of the geometric coefficient (2.56)
μ	geometric coefficient (2.50)
$\bar{\Upsilon}$	fixed point test vector (2.28)
ϕ	correction factor (2.74)
ϕ_k	correction factor (2.46)

BIBLIOGRAPHY

A1 "A Study of an Alkylation Plant Using MACSIM",
The Final Year Project, Dept. of Chemical Engineering,
McMaster University, Hamilton, Ontario, (1967).

A2 Amundson, N.R.,
"Mathematical Methods in Chemical Engineering",
Prentice-Hall, Inc., (1966).

A3 Aitken, A.C.,
"On Bernoulli's Numerical Solution of Algebraic Equations",
Proceedings of the Royal Society of Edinburgh (1925-1926).

A4 Aitken, A.C.,
"Studies in Practical Mathematics", Part III
Proceedings of the Royal Society of Edinburgh (1936-1937).

A5 Albright, L.F.,
"Alkylation: Chemical and Engineering Factors for Reactor Design",
Chem. Eng., 73, 14, (1966).

A6 Amundson, N.R. and A.J. Pontinen,
"Multicomponent Distillation Calculations on a Large Digital Computer",
I.E.C., 50, 5 (1958).

B1 Barnes, J.,
"An Algorithm for Solving Non-Linear Equations Based on Secant Method",
Computer J., 8, 1965).

B2 Broyden, C.G.,
"A Class of Methods for Solving Non-linear Simultaneous Equations",
Math. Comput., 19, (1965).

C1 Chartrand, G.,
Graduate Students Project, Chemical Plant Simulation
Course, McMaster University, Hamilton (1967)..

C2 Christensen, J.H.,
"The Use of Structure in Process Design Computation",
Ph.D. Thesis, University of Wisconsin (1967).

C3 Cavett, R.H.,
"Application of Numerical Methods to the Convergence
of Simulated Processes Involving Recycle Loops",
Proc. Am. Petrol. Inst., 43, III, 57-76, (1963).

C4 Crowe, C.M., A.E. Hamielec, T.W. Hoffman, A.I. Johnson,
P.T. Shannon, D.R. Woods,
"Chemical Plant Simulation", McMaster University (1969).

E1 Evans, L.B., D.G. Steward, and C.R. Sprague,
"Computer-Aided Chemical Process Design",
Chem. Eng. Progr., 64, No. 4, (1968).

F1 Fadeeva, V.N.,
"Computational Methods of Linear Algebra",
Dover Publications, Inc., (1959).

F2 Forder, G.J. and H.P. Hutchison,
"The Analysis of Chemical Plant Flowsheets",
Chem. Eng. Science, 24, (1969).

G1 Goldstein, A.A.,
"Constructive Real Analysis",
A Harper International Edition (1967).

H1 Himmelblau, D.M.,
"Decomposition of Large Scale Systems",
Chem. Eng. Sci., 21, 425-438, (1966).

H2 Holland, C.D.,
"Multi-Component Distillation",
Prentice-Hall Inc., (1963).

I1 Isaacson, E., and H.B. Keller,
"Analysis of Numerical Methods",
John Wiley & Sons, Inc., (1966).

J1 Johnson, A.I. and T. Toong,
"GEMCS - Information Handling Program",
Dept. of Chemical Engineering, McMaster University (1968).

K1 Kesler, M.G. and P.R. Griffiths,
"A Computer System for Process Simulation",
Proc. Am. Petrol. Inst., 43, III, 49-56, (1963).

K2 Kliesch, H.C.,
"An Analysis of Steady-State Process Simulation:
Formulation and Convergence",
Ph.D. Thesis, Tulane University, New Orleans, Louisiana (1967).

L1 Lederman, P.B.,
"Flowsheet Simulation and Beyond",
Chem. Eng., 75, No. 26, pp. 127-132, (1968).

L2 Lee, W., and D.F. Rudd,
"On the Ordering of Recycle Calculations",
AIChE J., 12, 1184 (1966).

L3 Lohr, L.R., and L.B. Rall,
"Efficient Use of Newton's Method",
I.C.C. Bull., 6, (1967).

L4 Lapidus, L.,
"Digital Computation for Chemical Engineers",
McGraw-Hill, Inc., (1962).

M1 The Final Year Project, 1966-1967 and 1967-1968
"Comprehensive Report on A Digital Computer Simulation
of an Alkylation Unit",
Dept. of Chemical Engineering, McMaster University (1968).

M2 Mosler, H.A.,
"PACER - A Digital Computer Executive Routine for
Process Simulation and Design",
M.S. Thesis, Purdue University (January 1964).

M3 The Final Year Project, 1968-1969
"A Digital Computer Simulation of the Bayer Process
for Alumina Extraction",
Dept. of Chemical Engineering, McMaster University (1969).

N1 Norman, R.L.,
"A Matrix Method for Location of Cycles in a Directed Graph",
AIChE J., 11, 3, pp 450-452 (1965).

N2 Nagiev, M.F.,
"The Theory of Recycle Processes in Chemical Engineering",
Macmillan, New York (1964).

N3 Naphtali, L.M.,
"Process Heat and Material Balance",
C.E.P., 60, 9, (1964).

P1 Petryschuk, W.F., and A.I. Johnson,
"A Simulation and Parametric Study of Four Existing
Multi-Component Distillation Columns",
Can. J. Chem. Eng., 44, 241 (1966).

P2 Petryschuk, W.F.,
"The Mathematical Representation and Analysis of a
Light Hydrocarbon Refining Network",
Ph.D. Thesis, Chemical Engineering Department,
McMaster University (1967).

P3 Prausnitz, J.N. et. al.,
"Computer Calculations for Multicomponent Vapour-Liquid
Equilibrium",
Prentice-Hall, Inc., (1967).

R1 Rinard, I.H., and D.L. Ripps,
"The Steady-State Simulation of Continuous Chemical
Processes",
Chem. Eng. Progr., Symposium Ser., No. 55, 61 (1965)

R2 Rosen, E.M.,
"A Machine Computation Method for Performing Material Balance",
Chem. Eng. Progr., 58, No. 10, (1962).

R3 Rubin, D.I.,
"Generalized Material Balance",
Chem. Eng. Progr., Symposium Series, 58, No. 37 (1962).

R4 Rosen, E.M.,
"A Review of Quasi-Newton Methods in Nonlinear Equation
Solving and Unconstrained Optimization",
Proceedings - A.C.M. National Meeting (1966).

R5 Ravicz, A.E., and R.L. Norman,
"Heat and Mass Balancing on a Digital Computer",
C.E.P., 60, 5, (1964).

R6 Rose, A., Sweeney, R.F., and Schrodt, V.N.,
"Continuous Distillation Calculations by Relaxation Method",
I.E.C., 50, No. 5 (1958).

S1 Saaty, T.L., and Bram, J.,
"Nonlinear Mathematics",
McGraw-Hill, Inc., (1964).

S2 Shannon, P.T., A.I. Johnson, C.M. Crowe, T.W. Hoffman,
A.E. Hamielec, and D.R. Woods,
"Computer Simulation of Sulfuric Acid Plant",
Chem. Eng. Progr., 62, No. 6, (1966).

S3 Sargent, R.W.H., and A.W. Westerberg,
"Speed-Up" in Chemical Engineering Design,
Trans. I Chem. E., 42, T190-97 (1964).

S4 Shaw, I.D.,
"An Appraisal of the MACSIM Simulation Routine in its
Application to an Alkylation Plant",
M.S. Thesis, McMaster University (1969).

S5 Schultz, D.G., and J.L. Melsa,
"State Functions and Linear Control Systems",
McGraw-Hill, Inc., (1967).

T1 Tomich, J.F.,
"A New Simulation Method for Equilibrium Stage Processes",
"Second Joint AIChE - IIQPR Meeting" (May 1968).

W1 Wegstein, J.H.,
"Accelerating Convergence of Iterative Processes",
Comm. of A.C.M., 1, 9, (1958).

W2 Wang, J.C., and G.E. Henke,
"Tridiagonal Matrix for Distillation",
Hydrocarbon Processing, 45, 155 (1966).

Y1 Yen, L.C., and R.E. Alexander,
"Estimation of Vapor and Liquid Enthalpies",
53rd National Meeting, AIChE, (May 1964).

Z1 Zeleznik, F.J.,
"Quasi-Newton Methods for Non-linear Equations",
J. ACM, 15, No. 2, (1968).

APPENDIX A

THE SOLUTION OF MATRIX DIFFERENCE EQUATION (A2)

Consider the matrix difference equation

$$\bar{X}_{n+1} = \bar{\bar{A}}\,\bar{X}_n + \bar{b} \qquad \text{(A-1)}$$

for initial value $\bar{X}_o$. For solving (A-1) we may consider first the homogeneous equation

$$\bar{X}_{n+1} = \bar{\bar{A}}\,\bar{X}_n \qquad \text{(A-2)}$$

and assume it has a solution of the form

$$\bar{X}_n = \bar{Z}\,\lambda^n \qquad \text{(A-3)}$$

where $\bar{Z}$ is a vector of constants and λ is an unknown parameter. Equation (A-3) has to satisfy equation (A-2). Therefore,

$$\bar{Z}\,\lambda^{n+1} = \bar{\bar{A}}\,\bar{Z}\,\lambda^n$$

and if $\lambda \neq 0$

$$\bar{Z}\,\lambda = \bar{\bar{A}}\,\bar{Z}$$

or

$$(\bar{\bar{A}} - \lambda\,\bar{\bar{I}})\,\bar{Z} = 0 \qquad \text{(A-4)}$$

and because we desire vectors $\bar{Z}$ which are not the zero vectors (not the trivial solution) the condition that equation (A-3) be a solution is that λ be an eigenvalue of $\bar{\bar{A}}$, and $\bar{Z}$ be an eigenvector belonging to the eigenvalue. Therefore let λ_j be an eigenvalue which satisfies

$$\det (\bar{\bar{A}} - \lambda_j \bar{\bar{I}}) = 0$$

and $\bar{Z}_j$ be a non-zero column of

$$\text{adj } (A - \lambda_j I)$$

then there are m solutions in the form of equation (A-3)

$$\bar{X}_n = \bar{Z}_j \lambda_j^n \tag{A-5}$$

$$j = 1, 2, 3, \ldots m.$$

A particular solution of the nonhomogeneous equation (A-1) can be found easily. Consider constant vector $\bar{X}_s$ which is a particular solution, then

$$\bar{X}_s = \bar{\bar{A}} \bar{X}_s + \bar{b}$$

and

$$\bar{X}_s = (\bar{\bar{I}} - \bar{\bar{A}})^{-1} \bar{b} \tag{A-6}$$

Therefore the general solution that contains m linear independent solutions is

$$\bar{X}_n = \sum_{j=1}^{m} C_j \bar{Z}_j \lambda_j^n + \bar{X}_s \tag{A-7}$$

If (A-7) satisfies the initial value $\bar{X}_o$ then

$$\bar{X}_o = \sum_{j=1}^{m} C_j \bar{Z}_j + \bar{X}_s \tag{A-8}$$

and C_j may be determined from (A-8). Now, if $\bar{W}_j$ is the associated eigenrow of A, then

$$\bar{W}_k^T \bar{Z}_j = 0 \qquad k \neq j$$

and (A-8) becomes

$$\bar{W}_k^T (\bar{X}_o - \bar{X}_s) = \sum_{j=1}^{m} C_j \bar{W}_k^T \bar{Z}_j$$

$$= C_k \bar{W}_k^T \bar{Z}_k$$

Therefore

$$C_j = \frac{\bar{W}_j^T [\bar{X}_o - \bar{X}_s]}{\bar{W}_j^T \bar{Z}_j} \tag{A-9}$$

and the desired solution is

$$\bar{X}_n = \sum_{j=1}^{m} \frac{\bar{W}_j^T [\bar{X}_o - \bar{X}_s]}{\bar{W}_j^T \bar{Z}_j} \cdot \bar{Z}_j \lambda_j^n + \bar{X}_s \tag{A-10}$$

APPENDIX B

THE LINEARIZATION PROCEDURE

The recycle process can be represented by a set of functions that gives the relation between the two halves of the assumed streams, as

$$y_i = F_i(\bar{X}) \qquad (B-1)$$

$$i = 1, 2, 3, \ldots m$$

or in vector notation,

$$\bar{Y} = \bar{F}(\bar{X}) \qquad (B-2)$$

where $\bar{F}$ - set of functions

$\bar{X}$ - the "input" part of the assumed streams

$\bar{Y}$ - the "output" part of the assumed streams

Two methods for linearization of iterative processes are given here. The first one linearizes the process around certain point $\bar{X}_L$ by series of m perturbations in all the main directions of the vector space. The second method linearizes by way of fitting set of linear equations to m+1 points of vectors $\bar{X}$ and $\bar{Y}$, which have been obtained by m+1 evaluations of the process.

A. The process functions (B-2) can be linearized around any point $\bar{X}_L$ to give linear set of equations in form of

$$\bar{Y} = \bar{\bar{A}}\,\bar{X} + \bar{b} \qquad (B-3)$$

where $\bar{\bar{A}}$ is the matrix of partial derivatives of $\bar{F}$ as

$$a_{ij} = \frac{\partial F_i(\bar{X}_L)}{\partial x_j} \tag{B-4}$$

where $\bar{X}_L$ is the linearization point and a_{ij} is an element of matrix $\bar{\bar{A}}$. The derivative (B-4) can only be found numerically as the set of functions (B-1) have no analytical derivatives.

For calculating matrix $\bar{\bar{A}}$, m+1 evaluations of the set of functions are necessary. That is, the recycle process has to be calculated for m+1 different "inputs" $\bar{X}$, to evaluate m+1 "outputs" $\bar{Y}$. The varying "inputs" $\bar{X}$ are as follows:

$$\bar{X}_L \rightarrow \bar{Y}_L \tag{B-5}$$

and

$$(\bar{X}_L + \delta\bar{e}_j) \rightarrow \bar{Y}_j \tag{B-6}$$

$$j = 1, 2, 3, \ldots m$$

where $\bar{e}_j$ is the vector with unity in the jth position and zeroes elsewhere. δ is a small perturbation. Thus $\bar{X}$ varies in each of the main directions in the m dimension space. The perturbation value δ is usually taken as a fraction of $x_{j,L}$, the jth element in vector $\bar{X}_L$. Now using (B-3) and (B-5)

$$\bar{Y}_L = \bar{\bar{A}}\,\bar{X}_L + \bar{b} \tag{B-7}$$

and from (B-3) and (B-6) we get

$$\bar{Y}_j = \bar{\bar{A}}(\bar{X}_L + \delta\bar{e}_j) + \bar{b}$$

$$\bar{Y}_j = \bar{\bar{A}}\,\bar{X}_L + \bar{b} + \delta\bar{\bar{A}}\,\bar{e}_j$$

$$\bar{Y}_j = \bar{\bar{A}}\,\bar{X}_L + \bar{b} + \delta \bar{a}_j \tag{B-8}$$

where $\bar{a}_j$ is the jth column in matrix A. And finally from (B-7) and (B-8)

$$\bar{a}_j = \frac{1}{\delta}(\bar{Y}_j - \bar{Y}_L) \tag{B-9}$$

$$j = 1, 2, 3, \ldots m$$

After m+1 evaluations of the process, all the columns of matrix $\bar{\bar{A}}$ are calculated (B-9), and the matrix of the linear coefficients may be used for further analysis of the process. The matrix $\bar{\bar{A}}$ can be also used to evaluate convergence promotion data.

B. By this method it is possible to obtain the matrix linear equation by any m+1 points $\bar{X}_j$ and the related points $\bar{Y}_j$. In other words, by m+1 evaluations of the process with some random "input" $\bar{X}_j$ to obtain $\bar{Y}_j$, it is possible to fit the matrix linear equation to these m+1 points $\bar{X}_j$ and $\bar{Y}_j$ as

$$\bar{Y}_j = \bar{\bar{A}}\,\bar{X}_j + \bar{b} \tag{B-10}$$

$$j = 1, 2, 3, \ldots m+1$$

These m+1 matrix equations (B-10) can be rearranged in partitioned matrices as follows:

$$\left[\begin{array}{c|c|c|c} \bar{Y}_1 & \bar{Y}_2 & \cdots & \bar{Y}_{n+1} \\ & & & \end{array}\right] = \left[\begin{array}{c|c} \bar{\bar{A}} & \bar{b} \\ & \end{array}\right] \left[\begin{array}{c|c|c|c} \bar{X}_1 & \bar{X}_2 & \cdots & \bar{X}_{n+1} \\ \hline 1 & 1 & & 1 \end{array}\right] \tag{B-11}$$

or

$$\bar{\bar{U}} = \bar{\bar{C}} \cdot \bar{\bar{V}} \tag{B-12}$$

The matrix $\bar{\bar{U}}$ has m rows and m+1 columns where the columns $\bar{U}_j$ are

$$\bar{U}_j = \bar{Y}_j \qquad \text{(B-13)}$$

$$j = 1, 2, 3, \dots m+1$$

The matrix $\bar{\bar{V}}$ has m+1 rows and m+1 columns where the columns $\bar{V}_j$ are

$$\bar{V}_j = \begin{bmatrix} \bar{X}_j \\ \\ 1 \end{bmatrix} = \begin{bmatrix} x_{1,j} \\ x_{2,j} \\ x_{3,j} \\ \cdot \\ \cdot \\ \cdot \\ x_{m,j} \\ 1 \end{bmatrix} \qquad \text{(B-14)}$$

$$j = 1, 2, 3, \dots m+1$$

where $x_{i,j}$ is the "ith" element in vector $\bar{X}_j$.

The matrix $\bar{\bar{C}}$ has m rows and m+1 columns where the columns $\bar{C}_j$ are

$$\bar{C}_j = \bar{a}_j \qquad \text{(B-15)}$$

$$j = 1, 2, 3, \dots m$$

and

$$\bar{C}_{m+1} = \bar{b}$$

Now, postmultiplying of (B-12) by $\bar{\bar{V}}^{-1}$ gives

$$\bar{\bar{C}} = \bar{\bar{U}} \,.\, \bar{\bar{V}}^{-1} \qquad \text{(B-16)}$$

where

$$\bar{\bar{C}} = \left[\begin{array}{c|c} \bar{\bar{A}} & \bar{b} \end{array} \right]$$

The matrix linear equation (B-10) with $\bar{\bar{A}}$ and $\bar{b}$ which has been obtained by (B-16), represents linearization among the m+1 points of $\bar{X}_j$.

These m+l points of $\bar{X}_j$ can be chosen independently by some random way or by the following two methods:

1. Applying the same technique as in case A, using (B-5) and (B-6) to obtain $\bar{X}_j$ around certain point $\bar{X}_L$.

2. Using successive substitution, where

$$\bar{X}_{j+1} = \bar{Y}_j$$

That is, during m+l iterations we can collect m+2 points of $\bar{X}_j$ as

$$\bar{X}_{j+1} = \bar{F}(\bar{X}_j)$$
$$j = 1, 2, 3, \ldots m+1$$

and these m+2 points can be used to calculate matrix $\bar{\bar{C}}$ (B-16). The only change is in the columns of matrix $\bar{\bar{U}}$, which were $\bar{Y}_j$, would be replaced by $\bar{X}_{j+1}$. Thus, equation (B-13) have to be replaced by

$$\bar{U}_j = X_{j+1} \qquad \text{(B-17)}$$
$$j = 1, 2, \ldots m+1$$

APPENDIX C

ITERATIVE CALCULATION OF SYSTEMS OF EQUATIONS

(Proof for Theorems I and 2 which are given in Section 2.2.I)

Consider the iterative calculation

$$\bar{X}_{n+1} = \bar{F}(\bar{X}_n) \qquad \text{(C-I)}$$

$$n = 0, 1, 2 \ldots$$

where $\bar{X}$ is an m-dimensional column vector with components $x_1, x_2, \ldots x_m$, and $\bar{F}(\bar{X})$ is an m-dimensional vector valued functions, i.e. a column vector with components $F_1(\bar{X}), F_2(\bar{X}), \ldots F_m(\bar{X})$. The solution of the iteration is some vector, say $\bar{X}_s$, which is some point in the m-dimensional space. And the initial point, say $\bar{X}_o$, is some initial estimate of the solution. Considering the iterate (C-I) with some initial point $\bar{X}_o$, Isaacson and Keller (II) give an important result concerning the convergence of this procedure and prove the existence of a unique solution.

Theorem I

Let $\bar{F}(\bar{X})$ satisfy

$$||\bar{F}(\bar{X}) - \bar{F}(\bar{Y})|| \leq M\,||\bar{X} - \bar{Y}|| \qquad \text{(C-2)}$$

for all vectors $\bar{X}$, $\bar{Y}$ such that

$$||\bar{X} - \bar{X}_o|| \leq r$$

$$||\bar{Y} - \bar{X}_o|| \leq r$$

with the Lipschitz constant M satisfying

$$0 \leq M < 1 \tag{C-3}$$

and let the initial iterate, $\bar{X}_o$, satisfy

$$||\bar{F}(\bar{X}_o) - \bar{X}_o|| \leq (1 - M).r \tag{C-4}$$

Then

i) all iterates (C-1), satisfy

$$||\bar{X}_n - \bar{X}_o|| \leq r \tag{C-5}$$

ii) the iterates converge to some vector, say

$$\lim_{n\to\infty} \bar{X}_n \to \bar{X}_s$$

iii) $\bar{X}_s$ is the only root of (C-1) in the interval,

$$||\bar{X} - \bar{X}_o|| \leq r$$

where the norm may be chosen as any one of the following

$$\begin{aligned} ||\bar{X}||_\infty &\equiv \max_i |x_i| \\ ||\bar{X}||_1 &\equiv \sum_{i=1}^{m} |x_i| \\ ||\bar{X}||_2 &\equiv \{ \sum_{i=1}^{m} x_i^2 \}^{\frac{1}{2}} \end{aligned} \tag{C-6}$$

Proof: We prove (i) by induction. Since $\bar{X}_1 = F(\bar{X}_o)$, we have by (C-3) and (C-4)

$$||\bar{X}_1 - \bar{X}_o|| \leq (1-M)\ r \leq r \tag{C-7}$$

and hence $\bar{X}_1$ is in the interval (C-5). Assume the above to be true for the iterates $\bar{X}_1$, $\bar{X}_2$, ... $\bar{X}_n$. Then from (C-1)

$$||\bar{X}_{n+1} - \bar{X}_n|| = ||\bar{F}(\bar{X}_n) - \bar{F}(\bar{X}_{n-1})||$$

and by the inductive assumption, $\bar{X}_n$ and $\bar{X}_{n-1}$ are in the interval (C-5). Thus, by (C-2), the Lipschitz condition yields

$$\begin{aligned} ||\bar{X}_{n+1} - \bar{X}_n|| &\leq M\ ||\bar{X}_n - \bar{X}_{n-1}|| \\ &\leq M^2||\bar{X}_{n-1} - \bar{X}_{n-2}|| \\ &\cdot \\ &\cdot \\ &\leq M^n||\bar{X}_1 - \bar{X}_o|| \\ &\leq M^n(1 - M).r \end{aligned} \tag{C-8}$$

Here we have used (C-1) and (C-2) recursively and then applied (C-7). However,

$$\begin{aligned} ||\bar{X}_{n+1} - \bar{X}_o|| &= ||(\bar{X}_{n+1} - \bar{X}_n) + (\bar{X}_n - \bar{X}_{n-1}) + \dots +(\bar{X}_1 - \bar{X}_o)|| \\ &\leq ||\bar{X}_{n+1} - \bar{X}_n|| + ||\bar{X}_n - X_{n-1}||+ \dots +||\bar{X}_1 - \bar{X}_o|| \\ &\leq (M^n + M^{n-1} + \dots + 1)(1-M)r = (1-M^{n+1})r \\ &\leq r \end{aligned}$$

which completes the proof of (i).

To prove part (ii), from (C-8) we can see that the iterate converges, and it is possible to show that the sequence $\{\bar{X}_n\}$ is a Cauchy sequence.

Suppose the limit is $\bar{X}_s$, then using (C-1) and (C-2)

$$\begin{aligned}||\bar{X}_n - \bar{X}_s|| &= ||\bar{F}(\bar{X}_{n-1}) - \bar{F}(\bar{X}_s)|| \\ &\leq M\,||\bar{X}_{n-1} - \bar{X}_s|| \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ &\leq M^n||\bar{X}_o - \bar{X}_s|| \\ &\leq M^n \,.\, r\end{aligned}$$

and if $n \to \infty$, then $||\bar{X}_n - \bar{X}_s|| = \bar{0}$ since $M < 1$.

For part (iii), the uniqueness, let $\bar{X}_r$ be another root in the interval (C-5). Then, since $\bar{X}_s$ and $\bar{X}_r$ are both in this interval, (C-2) and (C-3) hold and we have, if $||\bar{X}_s - \bar{X}_r|| \neq \bar{0}$

$$\begin{aligned}||\bar{X}_s - \bar{X}_r|| &= ||\bar{F}(\bar{X}_s) - \bar{F}(\bar{X}_r)|| \\ &\leq M||\bar{X}_s - \bar{X}_r|| \\ &< ||\bar{X}_s - \bar{X}_r||\end{aligned}$$

This contradiction implies that $\bar{X}_s = \bar{X}_r$, and the proof of the theorem is concluded.

Theorem 2

Let (C-1) have a root $\bar{X}_s$ and let the function $F_i(\bar{X})$ have continuous first order partial derivatives,

$$F_{ij}(\bar{X}) = \frac{\partial F_i(\bar{X})}{\partial x_j} \qquad \text{(C-9)}$$

and satisfy

$$\max_{i} \sum_{j=1}^{m} |F_{ij}(\bar{X})| < M \leq 1 \qquad (C-10)$$

for all $\bar{X}$ in

$$||\bar{X} - \bar{X}_s||_\infty \leq r \qquad (C-11)$$

(See (C-6))

Then

i) for any $\bar{X}_o$ satisfying (C-11) all the iterates $\bar{X}_n$ of (C-1) also satisfy (C-11)

ii) for any $\bar{X}_o$ satisfying (C-11) the iterates (C-1) converge to the solution $\bar{X}_s$ which is unique in (C-11).

Proof: for any two points $\bar{X}$, $\bar{Y}$ in (C-11) we have by Taylor's theorem:

$$F_i(\bar{X}) - F_i(\bar{Y}) = \sum_{j=1}^{m} F_{ij}(\bar{\xi}^{(i)}).(x_j - y_j) \qquad (C-12)$$

where $\bar{\xi}^{(i)}$ is a point on the open line segment joining $\bar{X}$ and $\bar{Y}$. Thus, $\bar{\xi}^{(i)}$ is in (C-11) and using (C-6) and (C-10) yields

$$\begin{aligned} |F_i(\bar{X}) - F_i(\bar{Y})| &\leq \sum_{j=1}^{m} |F_{ij}(\bar{\xi}^{(i)})|.|x_j - y_j| \\ &< ||\bar{X} - \bar{Y}||_\infty . \sum_{j=1}^{m} |F_{ij}(\bar{\xi}^{(i)})| \\ &< M.||\bar{X} - \bar{Y}||_\infty \end{aligned}$$

Since the inequality holds for each i, we have

$$||\bar{F}(\bar{X}) - \bar{F}(\bar{Y})||_\infty \leq M.||\bar{X} - \bar{Y}||_\infty \qquad (C-13)$$

and it has been proven that $\bar{F}(\bar{X})$ is Lipschitz continuous in the domain

(C-11), with respect to the indicated norm. Now, for any $\bar{X}_o$ in (C-11),

$$\begin{aligned}||\bar{X}_1 - \bar{X}_s||_\infty &= ||\bar{F}(\bar{X}_o) - \bar{F}(X_s)||_\infty \\ &\leq M.||\bar{X}_o - \bar{X}_s|| \\ &\leq M . r\end{aligned}$$

So $\bar{X}_1$ is also in (C-11). By induction we have then

$$\begin{aligned}||\bar{X}_n - \bar{X}_s||_\infty &= ||\bar{F}(\bar{X}_{n-1}) - \bar{F}(\bar{X}_s)||_\infty \\ &\leq M.||\bar{X}_{n-1} - \bar{X}_s||_\infty \\ &\vdots \\ &\leq M^n||\bar{X}_o - \bar{X}_s|| \\ &\leq M^n . r \qquad \text{(C-14)}\end{aligned}$$

and hence all $\bar{X}_n$ lie in (C-11). The convergence immediately follows from (C-14) since $M < 1$. The uniqueness follows as before.

APPENDIX D

PARTITION ITERATIVE MATRIX

The question whether to converge all the recycles simultaneously or to converge some of them as the calculation proceeds in the main recycle is raised frequently while performing simulation of complex recycle processes. Thus, if the process has more than one recycle, that is, that more than one stream ought to be assumed in order to render the recycle process acyclic, then the recycles may be divided in two groups; the main and the secondary recycles. For any iteration of the main recycle the secondary may be iterated few times or even be iterated until convergence is obtained.

If the general iterative calculation was linearized to give

$$\bar{X}_n = \bar{\bar{A}}\,\bar{X}_{n-1} + \bar{b} \qquad \text{(D-1)}$$

then by analyzing matrix $\bar{\bar{A}}$ it is possible to indicate how to divide the iterative process.

Suppose the secondary recycles have p variables and it is iterated k times for any iteration of the main group that contains ℓ variables.

Equation (D-1) can be written in partitioned form, as

$$\begin{bmatrix} \bar{X}_1 \\ \hline \bar{X}_2 \end{bmatrix}^{(n)} = \left[\begin{array}{c|c} \bar{\bar{A}}_1 & \bar{\bar{A}}_2 \\ \hline \bar{\bar{A}}_3 & \bar{\bar{A}}_4 \end{array}\right] \cdot \begin{bmatrix} \bar{X}_1 \\ \hline \bar{X}_2 \end{bmatrix}^{(n-1)} + \begin{bmatrix} \bar{b}_1 \\ \hline \bar{b}_2 \end{bmatrix} \qquad \text{(D-2)}$$

where $\bar{X}_1$ is the secondary recycle variables vector of $p \times 1$ elements.

$\bar{X}_2$ is the main recycle variable vector of $\ell \times 1$ elements.

$\bar{\bar{A}}_1$ is $p \times p$ matrix

$\bar{\bar{A}}_2$ is $p \times \ell$ matrix

$\bar{\bar{A}}_3$ is $\ell \times p$ matrix

$\bar{\bar{A}}_4$ is $\ell \times \ell$ matrix

$\bar{b}_1$ is $p \times 1$ vector

$\bar{b}_2$ is $\ell \times 1$ vector

and $m = p + \ell$ where m is the total number of variables in the assumed streams.

Equation (D-2) can be written as two matrix equations:

$$X_1^{(n)} = A_1 X_1^{(n-1)} + A_2 X_2^{(n-1)} + b_1 \qquad \text{(D-3a)}$$

$$X_2^{(n)} = A_3 X_1^{(n-1)} + A_4 X_2^{(n-1)} + b_2 \qquad \text{(D-3b)}$$

Now, introducing the k sub-iterations of the secondary group of variable X_1, such that (D-3a) gives

$$X_1^{(n-1,i)} = A_1 X_1^{(n-1,i-1)} + A_2 X_2^{(n-1)} + b_1 \qquad \text{(D-4a)}$$

$$i = 1, 2, 3, \ldots k$$

where $X_1^{(n-1,o)} \equiv X_1^{(n-1)}$

$X_1^{(n-1,k)} \equiv X_1^{(n,o)}$

Then after k iterations of the secondary recycles equations (D-3a) and (D-3b) we get

$$X_1^{(n,o)} = X_1^{(n-1,k)} = A_1 X_1^{(n-1,k-1)} + A_2 X_2^{(n-1)} + b_1 \qquad \text{(D-5a)}$$

$$X_2^{(n)} = A_3 X_1^{(n-1,k-1)} + A_4 X_2^{(n-1)} + b_2 \qquad \text{(D-5b)}$$

It can easily be shown that for any linear iterative calculation as

$$\bar{Y}^{(k)} = \bar{\bar{B}} \bar{Y}^{(k-1)} + \bar{d}$$

where $Y^{(o)}$ is the initial value, $\bar{Y}^{(k)}$ can be expressed as

$$\bar{Y}^{(k)} = \bar{\bar{B}}^k Y^{(o)} + (I + B + B^2 + --- +B^{k-1}).\bar{d} \qquad \text{(D-6)}$$

Using (D-6) for the iterate i in (D-4a) gives

$$X_1^{(n-1,i)} = A_1^i X_1^{(n-1,o)} + (I+A_1+A_1^2 + --- A_1^{i-1})[A_2X_2^{(n-1)} + b_1] \qquad \text{(D-7)}$$

Now, using equation (D-7) with $i=k$ instead of equation (D-5a) and substituting (D-7) with $i=k-1$ to equation (D-5b) we get:

$$X_1^{(n,o)} = A_1^k X_1^{(n-1,o)} + (I+A_1+A_1^2 + --- +A_1^{k-1})A_2X_2^{(n-1)} +$$

$$+ (I+A_1+A_1^2 + --- A_1^{k-1})\, b_1 \qquad \text{(D-8a)}$$

$$X_2^{(n)} = A_3A_1^{k-1}\, X_1^{(n-1,o)} + [A_3(I+A_1+A_1^2+---+A_1^{k-2})A_2+A_4]\, X_2^{n-1}$$

$$+ A_3(I+A_1+A_1^2+---+A_1^{k-2})b_1 + b_2 \qquad \text{(D-8b)}$$

This new iterative procedure can be compressed in a partitioned matrix form as

$$\bar{X}^{(n)} = \bar{\bar{B}}_k \, \bar{X}^{(n-1)} + b_k \qquad \text{(D-9)}$$

where

$$\bar{\bar{B}}_k = \left[\begin{array}{c|c} A_1^k & (I+A_1+A_1^2 + --- A_1^{k-1}) \, A_2 \\ \hline A_3 \, A_1^{k-1} & A_3(I + A_1+A_1^2 + --- + A_1^{k-2}) \, A_2 + A_4 \end{array}\right] \qquad \text{(D-10)}$$

and

$$\bar{b}_k = \left[\begin{array}{c} (I + A_1 + --- + A_1^{k-1}) \, b_1 \\ \hline A_3(I + A_1 + --- + A_1^{k-2}) \, b_1 + b_2 \end{array}\right] \qquad \text{(D-11)}$$

for $k > 1$

Now, by comparing the biggest eigenvalue in modulus of matrices $\bar{\bar{A}}$ and $\bar{\bar{B}}_k$ it can be evaluated if the sub-iteration of the secondary recycle reduces the rate of convergence of the main recycle. We may study first the extreme case were $k \to \infty$, that is, that the sub-iteration of the secondary recycle is converged for every iteration of the main recycle. If $k \to \infty$ and all the eigenvalues of the partition matrix A_1 are less than unity in modulus then

$$\lim_{k\to\infty} {A_1}^k = \bar{\bar{0}}$$

and

$$\lim_{k\to\infty}(I + A_1 + A_1^2 + --- + A_1^{k-2}) = (I-A_1)^{-1}$$

then matrix $\bar{\bar{B}}_k$ (D-10) becomes

$$\bar{\bar{B}}_{k\to\infty} = \left[\begin{array}{c|c} \bar{\bar{0}} & (I - A_1)^{-1} A_2 \\ \hline \bar{\bar{0}} & A_3(I - A_1)^{-1} A_2 + A_4 \end{array}\right] \qquad \text{(D-12)}$$

The eigenvalues of matrix $\bar{\bar{B}}_k$ where $k \to \infty$ (D-12) are

$$\lambda_i = 0$$

$$i = 1, 2, 3, \ldots p$$

and the remaining ℓ eigenvalues are the eigenvalues of the partitioned matrix

$$[A_3(I - A_1)^{-1} A_2 + A_4] \qquad \text{(D-13)}$$

Comparing the eigenvalues of matrix (D-13) with the eigenvalues of the original matrix A, it is possible to evaluate if there is any benefit from the sub-iteration of the secondary recycle. If the biggest eigenvalue is reduced in modulus, we may consider iterating around the secondary iteration a few times or even to converge it. But this iteration will be worthwhile only if the calculation time of the secondary recycle is small in comparison to that of the main iteration.

NUMERICAL EXAMPLE

Second Order Recycle Problem with One Component
(See Figure D-1)

2 assumed streams: S3, S7

Sequence of calculation: 1, 2, 6, 4, 5, 3

After elimination we get

$$\begin{bmatrix} S7 \\ S3 \end{bmatrix}^{(n)} = \begin{bmatrix} 0.36 & 0.072 \\ 0.36 & 0.792 \end{bmatrix} \cdot \begin{bmatrix} S7 \\ S3 \end{bmatrix}^{(n-1)} + \begin{bmatrix} 264.8 \\ 912.8 \end{bmatrix}$$

that is, that

$$\bar{\bar{A}} = \begin{bmatrix} 0.36 & 0.072 \\ 0.36 & 0.792 \end{bmatrix} \qquad b = \begin{bmatrix} 264.8 \\ 912.8 \end{bmatrix}$$

and the eigenvalues of matrix $\bar{\bar{A}}$ are

$$\lambda_1 = 0.8454$$

$$\lambda_2 = 0.3066$$

The rate of convergence calculated by equation (2.77) is

$$R = -\log \lambda_1 = -\log(0.8454) = 0.07294$$

or $1/R = 13.71$

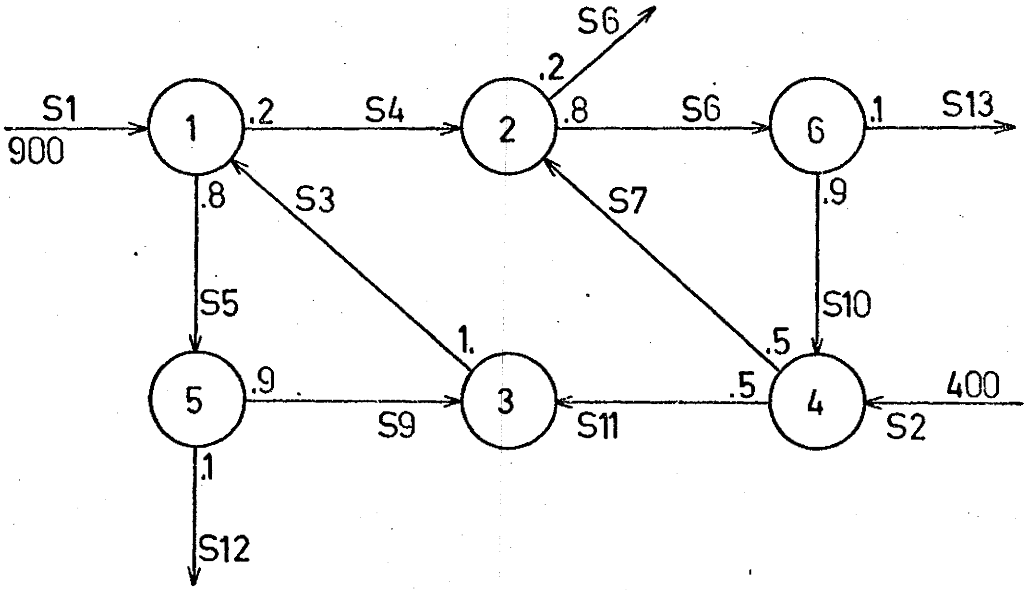

FIGURE D-1 SECOND ORDER RECYCLE PROBLEM

CASE 1

S3 is the main iteration and S7 is the secondary one. Iterate k times around S7 where the sequence is, $1,(2,6,4)_k,5,3,1,(2,6,4)_k,5,3,1,..$ and if $k \rightarrow \infty$ then

$$\bar{\bar{B}}_{k\rightarrow\infty} = \left[\begin{array}{c|c} 0 & (1-0.36)^{-1}\ 0.072 \\ \hline 0 & 0.36(1-0.36)^{-1}\ 0.072 + 0.792 \end{array}\right] = \left[\begin{array}{c|c} 0 & .1125 \\ \hline 0 & .8325 \end{array}\right].$$

and the new eigenvalues are

$$\lambda_1 = 0.0$$
$$\lambda_2 = 0.8325$$

A very small decrease of the biggest eigenvalue, from 0.8454 to 0.8325, is seen.

CASE 2

S7 is the main iteration and S3 is the secondary, iterate k times around S3 where the sequence is, $1,2,6,4,(5,3,1)_k,2,6,4,(5,3,1)_k$ .... and if $k \rightarrow \infty$. then

$$\bar{\bar{B}}_{k\rightarrow\infty} = \left[\begin{array}{c|c} 0.072(1-0.792)^{-1}\ 0.36 + 0.36 & 0 \\ \hline (1-0.792)^{-1}\ 0.36 & 0 \end{array}\right] = \left[\begin{array}{c|c} 0.4846 & 0 \\ \hline 1.7308 & 0 \end{array}\right]$$

The new eigenvalues are

$$\lambda_1 = 0.4846$$
$$\lambda_2 = 0.0$$

The biggest eigenvalue in modulus decreases significantly from 0.8454 to 0.4846.

As case 2 gives a better result we may examine the eigenvalues for this case, where k is equal to 2 and 3.

CASE 2a   k = 2

$$\bar{\bar{B}}_{k=2} = \left[\begin{array}{c|c} 0.36 + 0.072\ (1)\ 0.36 & 0.072\ (0.792) \\ \hline (1 + 0.792)\ 0.36 & 0.792^2 \end{array}\right] =$$

$$\bar{\bar{B}}_{k=2} = \left[\begin{array}{c|c} 0.38592 & 0.057024 \\ \hline 0.64512 & 0.627264 \end{array}\right]$$

and the eigenvalues are

$$\lambda_1 = 0.7334$$

$$\lambda_2 = 0.27984$$

and the rate of convergence is

$$R_{k=2} = -\log(0.7334) = 0.13466$$

or

$$\frac{1}{R_{k=2}} = 7.426$$

CASE 2b   k = 3

$$\bar{\bar{B}}_{k=3} = \left[\begin{array}{c|c} 0.36 + 0.072\ (1.792)\ 0.36 & 0.072\ (0.792)^2 \\ \hline (1 + 0.792 + 0.792^2)\ 0.36 & 0.792^3 \end{array}\right] =$$

$$\bar{\bar{B}}_{k=3} = \left[\begin{array}{c|c} 0.40645 & 0.045163 \\ \hline 0.870935 & 0.4968 \end{array}\right]$$

and the eigenvalues are

$$\lambda_1 = 0.655$$

$$\lambda_2 = 0.2482$$

and the rate of convergence is

$$R_{k=3} = -\log(0.655) = 0.18376$$

or

$$\frac{1}{R_{k=3}} = 5.44$$

The extra time for the secondary iteration need to be included in order to achieve a true comparison. Therefore if τ_1 and τ_2 are the calculation time of the main and secondary recycles respectively then the normalized rate of convergence is

$$\bar{R}_k = \frac{R_k}{1 + (k-1).\tau_2/(\tau_1 + \tau_2)} \qquad \text{(D-14)}$$

Thus, if we assume that all the computation units in this example have the same calculation time then

$$\tau_2/(\tau_1 + \tau_2) = 0.5$$

and

$$\bar{R}_{k=1} = 0.07294$$

$$\bar{R}_{k=2} = \frac{0.13466}{1 + 0.5} = 0.08977$$

$$\bar{R}_{k=3} = \frac{0.18376}{1 + 2(0.5)} = 0.09188$$

APPENDIX E

PROGRAM LISTINGS FOR SECTION 5.2

```
$IBFTC JUNC01
C
C
C     SUBROUTINE JUNC01  (MIXER - MASS BALANCE)
C     REVISED BY T. TOONG    FEB.20,1968
C
C
C     *MAXIMUM VALUE OF NOCOMP, NUMBER OF FLOW COMPONENTS, IS 26
C     *        ,,          NIN,     ,,      INPUT STREAMS,     5
C     *        ,,          NOUT,    ,,      OUTPUT STREAMS,    5
C
C
C     EN, EQUIPMENT VECTOR -
C     1. EQUIPMENT NUMBER
C     2. TEMPERATURE OF OUTPUT STREAMS, DEG F, IF 0., OUTPUT TEMPERATURE
C        ARE SET EQUAL TO TEMPERATURE OF 1ST INPUT STREAM
C     3. PRESSURE OF OUTPUT STREAMS, PSIA, IF 0., OUTPUT PRESSURE
C        ARE SET EQUAL TO PRESSURE OF 1ST INPUT STREAM
C     4. (BLANK)
C     5. (BLANK)
C     6. FRACTION OF TOTAL INPUT IN 1ST OUTPUT STREAM
C     7.                 ,,           2ND       ,,
C     8.                 ,,           3RD       ,,
C     9.                 ,,           4TH       ,,
C     10.                ,,           5TH       ,,
C
C
C     ENC, EQUIPMENT CONTROL VECTOR (OPTIONAL) -
C     1. EQUIPMENT NUMBER
C     2. FLAG
C     3. LENGTH OF EN LIST (FOR PRINTING)
C
C
C     SN, STREAM VECTOR -
C     1. STREAM NUMBER
C     2. STREAM FLAG
C     3. TOTAL FLOW IN LBMOLE/HR
C     4. TEMPERATURE IN DEGREE F
C     5. PRESSURE IN PSIA
C     6. FLOW OF COMPONENT 1 IN LBMOLE/HR
C     7.             ,,       2      ,,
C     8.             ,,       3      ,,
C     9.             ETC.
C
C
C     SNC, STREAM CONTROL VECTOR - NOT REQUIRED
C
C
C
```

```
      SUBROUTINE JUNC01
C     ********MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,KSETS,NIN,NOUT,NE,STRMI(5,30),STRMO(5,30),EN(80,20)
C
C TO CHECK IF INPUT STREAMS ARE NOT MORE THAN 5
      IF(NIN.LE.5) GOTO 10
      WRITE(6,90) NE
      NIN= 5
   10 JJ= NOCOMP+5
C
C
C MASS BALANCE
C (INPUT)
      DO 1 J=6,JJ
      STRMO(NOUT,J)=0.
      DO 1 I=1,NIN
    1 STRMO(NOUT,J)= STRMO(NOUT,J)+STRMI(I,J)
C (OUTPUT)
      DO 2 I=1,NOUT
      STRMO(I,4)= EN(NE,2)
      IF(EN(NE,2).EQ.0.) STRMO(I,4)=STRMI(1,4)
      STRMO(I,5)= EN(NE,3)
      IF(EN(NE,3).EQ.0.) STRMO(I,5)=STRMI(1,5)
      STRMO(I,3)=0.
      DO 2 J=6,JJ
      STRMO(I,J)= STRMO(NOUT,J)*EN(NE,I+5)
    2 STRMO(I,3)= STRMO(I,3)+STRMO(I,J)
C
C
C FORMATS
   90 FORMAT (////1X, 25H**ERROR**IN JUNC01 MODULE, I3,
     173H, NIN EXCEEDS 5, CALCULATION CONTINUES USING THE FIRST FIVE INP
     2UT STREAMS////)
C
      RETURN
      END
```

```
$IBFTC SEPA02
      SUBROUTINE SEPA02
C     THIS MODULE CALCULATES SPLIT OF FEED ACCORDING TO FRACTION OFINLET
C     SOLIDS THAT LEAVES OUT THE TOP AND FRACTION BY WEIGHT OF SOLIDS IN
C     THE BOTTOM STREAM
C
C     *ONE INPUT STREAM AND TWO OUTPUT STREAMS
C     STRMO(1,J) = TOP STREAM
C     STRMO(2,J) = BOTTOM STREAM
C
C     EQUIPMENT VECTOR -
C     1. EQUIPMENT NUMBER
C     2. WT. FRACTION OF FEED AS SOLIDS OUT TOP
C     3. WT. FRACTION OF SOLIDS IN BOTTOM STREAM
C     4. (BLANK)
C     5. (BLANK)
C
C     NO AEN VECTOR
C
C     ********MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,KSETS,NIN,NOUT,NE,STRMI(5,30),STRMO(5,30),EN(80,20)
C     ******************************************************************
      DIMENSION WTSOL(30),AA(30)
      COMMON/BLK1/WTMOL(22)
      JJ=NOCOMP+5
      DO 15 J = 6,JJ
      AA(J) = 0.0
      DO 15 I = 1,NIN
      AA(J) = AA(J) + STRMI(I,J)
   15 CONTINUE
      DO 10 I = 6,JJ
   10 WTSOL(I) = AA(I)*WTMOL(I-5)
      DO 11 I = 9,18
      STRMO(1,I) = AA(I)*EN(NE,2)
   11 STRMO(2,I) = AA(I)-STRMO(1,I)
      WSB=0.0
      DO 12 I=9,18
   12 WSB=WSB+STRMO(2,I)*WTMOL(I-5)
      WL=0.0
      DO 50 I=6,8
   50 WL=WL+WTSOL(I)
      DO 51 I=19,27
   51 WL=WL+WTSOL(I)
      WLB=WSB*(1.0/EN(NE,3)-1.0)
      SPLIQ=WLB/WL
      DO 60 I=6,8
      STRMO(2,I) = AA(I)*SPLIQ
   60 STRMO(1,I) = AA(I)-STRMO(2,I)
      DO 70 I=19,27
      STRMO(2,I) = AA(I)*SPLIQ
   70 STRMO(1,I) = AA(I)-STRMO(2,I)
C     SET TEMP. AND PRESSURES
      DO 80 I=4,5
      STRMO(1,I) = STRMI(1,I)
   80 STRMO(2,I) = STRMI(1,I)
C     CALCULATE TOTAL OUTPUT FLOWS
      STRMO(1,3) = 0.0
      STRMO(2,3) = 0.0
      DO 90 I = 6,JJ
      STRMO(1,3) = STRMO(1,3) + STRMO(1,I)
   90 STRMO(2,3) = STRMO(2,3) + STRMO(2,I)
      RETURN
      END

$IBFTC DATA
      BLOCK DATA
      COMMON / BLK1 / WTMOL(22)
      DATA WTMOL / 18.020, 105.993, 40.001, 1006.36015, 156.020,
     1             258.180, 177.720, 141.950, 56.080, 79.900, 100.091,
     2             44.011, 119.980, 81.971, 258.180, 159.700,141.950,
     3             600.490, 79.900, 142.048, 44.011, 104.024 /
      END
```

```
$IBFTC CP2
      SUBROUTINE CONV40
C
C     GEOMETRIC EXTRAPOLATION TECHNIQUE
C     WRITTEN BY O. ORBACH  MCMASTER UNIV.  DEC.  23  1968
C
C
C     EN, EQUIPMENT VECTOR -
C     1. EQUIPMENT NUNBER
C     2. NUMBER OF ITERATIONS BEFORE EXTRAPOLATION IS
C        APPLIED IN THE FIRST TIME.
C     3. RELAXATION FACTOR.
C     4. TOLERANCE OF THE FRACTIONAL CHANGE OF THE
C        GEOMETRIC COEFFICIENT (EG).
C     5. NUMBER OF ITERATION BEFORE EXTRAPOLATION
C        IS APPLIED .
C     *******MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,KSETS,NIN,NOUT,NE,STRMI(5,30),STRMO(5,30),EN(80,20)
      COMMON  STRMCI(5,10),STRMCO(5,10),ENC(80,10),AEN(30,30)
      COMMON SN(120,30),AAA(3655),LOOP
C
      DIMENSION X(180),F(180),X0(180),F0(180)
C
C     THE ASSUMED STREAMS.
      DIMENSION NS(8)
      NN=4
      DATA NS/22,30,31,100,0,0,0,0/
C
      LP=EN(NE,2)
      TC=EN(NE,3)
      EGT=EN(NE,4)
      LK=EN(NE,5)
C
      NNN=NOCOMP*NN
      IF(LOOP.GT.1) GOTO 200
      EN(NE,6)=1.0
      DO 100 K=1,NNN
      X0(K)=0.0
      F0(K)=0.0
      X(K)=0.0
      F(K)=0.0
  100 CONTINUE
  200 CONTINUE
      KOUNT=EN(NE,6)
C
      IF(KOUNT-2) 50,60,70
C
   50 CONTINUE
      DO 51  I=1,NN
      II=NS(I)
      DO 51  J=1,NOCOMP
      K=(I-1)*NOCOMP+J
      X0(K)=SN(II,J+5)
   51 CONTINUE
      GOTO 79
C
C
```

```
C
   60 CONTINUE
      SF0=0.0
      DO 61 I=1,NN
      II=NS(I)
      DO 61 J=1,NOCOMP
      K=(I-1)*NOCOMP+J
      X(K)=SN(II,J+5)
      F0(K)=0.0
      IF(ABS(X0(K)).LT.(1.0E-10)) GOTO 61
      F0(K)=(X(K)-X0(K))/X0(K)
      SF0=SF0+F0(K)*F0(K)
   61 CONTINUE
      SF0=SQRT(SF0)
      GOTO 79
C
C
   70 CONTINUE
      SF=0.0
      DO71 I=1,NN
      II=NS(I)
      DO 71 J=1,NOCOMP
      K=(I-1)*NOCOMP+J
      X1=SN(II,J+5)
      F(K)=0.0
      IF(ABS(X(K)).LT.(1.0E-10))  GOTO 71
      F(K)=(X1-X(K))/X(K)
      SF=SF+F(K)*F(K)
   71 CONTINUE
      SF=SQRT(SF)
      FG=SF/SF0
      SS=1.-FG
C
      WRITE(6,500) LOOP,SF,SS,FG
  500 FORMAT(//20X,6H LOOP=,I5,3X.3HSF=,F12.5.3X.3HSS=,F12.4,3X,3HFG=,
     1F9.4)
C
      IF(LOOP.GE.LP.AND.KOUNT.GT.LK) GOTO 80
C
   90 CONTINUE
      SF0=SF
      FG0=FG
      DO 75 I=1,NN
      DO 75 J=1, NOCOMP
      II=NS(I)
      K=(I-1)*NOCOMP+J
      X0(K)=X(K)
      X(K)=SN(II,J+5)
   75 F0(K)=F(K)
   79 FN(NE,6)=FN(NE,6)+1.0
      RETURN
```

```
C
   80 CONTINUE
      CEG=ABS((EG-EG0)/EG0)
      IF(CEG.GT.EGT ) GOTO 90
   84 WRITE(6,506) LOOP,KOUNT
  506 FORMAT(/40X,27HAPPLY CONVERGENCE PROMOTION,3X,5HLOOP=,I4,3X,
     16HKOUNT=,I4/)
C
      DO 81 I=1,NN
      II=NS(I)
      DO 81 J=1, NOCOMP
      K=(I-1)*NOCOMP+J
      B=X(K)*F(K)
      BB=F0(K)-F(K)
C
C
      BC=BB/F0(K)
      IF(BC.LT.(1.E-5)) GOTO 82
      IF(BC.GE.2.0) GOTO 82
C
C
      B=F(K)*(X(K)-X0(K))/BB
   82 CONTINUE
      X0(K)=X(K)
      F0(K)=F(K)
      SN(II,J+5)=X(K)+TC*B
   81 CONTINUE
      SN(NE,6)=1.0
      RETURN
```