

AN ABSTRACT OF THE THESIS OF

DANIEL E. BARCHERS for the degree DOCTOR OF PHILOSOPHY
(Name) (Degree)

in CHEMICAL ENGINEERING presented on January 3, 1975
(Major Department) (Date)

Title: OPTIMAL CONVERGENCE OF COMPLEX RECYCLE
PROCESS SYSTEMS

Abstract approved: Redacted for privacy
Eugene Elzy

The asymptotic convergence acceleration method used in steady state process simulation is examined in detail, and a criterion is developed to decide when it should be applied. The criterion is related to the rate of convergence of each variable and does not require any prejudgment of its tolerance level by the user.

A method of developing model systems of equations with any desired characteristics is presented and used to study the behavior of iterative systems. The new asymptotic technique is compared to previous variations of the asymptotic method and the direct substitution method using 11 equation sets and process systems. The new method resulted in more successful convergence promotion than the other methods and reduced the number of iterations to converge the example systems by an average of 50% when compared to direct substitution.

Many of the alternative cut sets for a given problem are shown to be nearly equivalent and will ultimately converge at the same rate by the method of successive substitutions. Cut sets in the equivalent group calculate the units of each individual recycle loop in the order in which the units appear in the loop. This is equivalent to the requirement that no single loop should be cut more than once. Mathematical analysis of a single variable, multiple loop problem showed that any cut set not in the equivalent group will converge at a slower rate than those in the equivalent group. Several process examples are used to demonstrate that cut sets in the non-equivalent group will also converge slower in the more general multiple variable, multiple loop case. These process examples also demonstrate that the equivalent cut sets will be more amenable to convergence promotion by the asymptotic method.

Optimal Convergence of Complex Recycle
Process Systems

by

Daniel E. Barchers

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

June 1975

APPROVED:

Redacted for privacy

Associate Professor of Chemical Engineering
in charge of major

Redacted for privacy

Head of Department of Chemical Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented

January 3, 1975

Typed by Clover Redfern for

Daniel E. Barchers

ACKNOWLEDGMENT

The author wishes to extend his sincere appreciation to:

Dr. Eugene Elzy, who inspired my interest in process simulation, for his guidance, encouragement, and advice.

The Department of Chemical Engineering, Dr. Charles Wicks, Head, for its financial support and graduate program.

Most of all, my wife, Suzanne, whose understanding, support, and belief in me was invaluable during the course of this work.

TABLE OF CONTENTS

<u>Chapter</u>	<u>Page</u>
1. INTRODUCTION	1
2. LITERATURE SURVEY AND MATHEMATICAL FOUNDATIONS	4
Cut Set Selection	4
Convergence Promotion	8
Mathematics of the Asymptotic Method	16
Convergence Tests	19
3. ASYMPTOTIC CONVERGENCE PROMOTION	24
Streams Tested for Convergence	24
Behavior of Iterative Systems	31
Estimation of the Number of Iterations to Converge	47
Extrapolation to the Solution	48
Useful Limits on Apparent eigenvalues	49
The Acceleration Criterion	55
Applications and Comparisons	58
4. CUT SET SELECTION	72
Initial Guesses	72
Sensitivity Matrices	75
Single Recycle Loops	78
Multiple Recycle Loops	80
Partitioned Sensitivity Matrices	80
Properties of Partitioned Sensitivity Matrices	95
Equivalent Cut Sets	98
Non-Equivalent Cut Sets	102
Process System Examples	108
5. CONCLUSIONS AND RECOMMENDATIONS	116
BIBLIOGRAPHY	118
APPENDICES	121
A. Descriptions of Example Systems	121
B. Subroutine Listings	138
C. Nomenclature	144

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Simple recycle loop.	25
2. Apparent eigenvalues of model system 1.	38
3. Apparent eigenvalues of model system 2.	39
4. Apparent eigenvalues of model system 5.	40
5. Apparent eigenvalues of model system 6.	41
6. Test vector versus iteration, system 1.	42
7. Test vector versus iteration, system 2.	43
8. Test vector versus iteration, system 5.	44
9. Test vector versus iteration, system 6.	45
10. Critical eigenvalues.	54
11. Logic flowchart, subroutine ASYM.	60
12. Single recycle loop system.	76
13. Multiple recycle loop system.	81
14. Types of process units.	88
15. Recycle system for cut set comparison.	104
16. Test vector for salt-water mixing example, cut set {3, 7}.	112
17. Test vector for salt-water mixing example, cut set {7, 9, 13}.	113

<u>Appendix Figure</u>	<u>Page</u>
A-1. Hypothetical ethylene dichloride process.	124
A-2. Single node scheme for the solution of a set of simultaneous equations.	124
A-3. Multi-node scheme for solution of photochemical reaction system.	125
A-4. Salt-water mixing system.	128
A-5. Cavett's hydrocarbon separation process.	129
A-6. Hypothetical reactor system.	130
A-7. Bayer hydrate wash section.	132
A-8. Phosphoric acid plant, Case I.	133
A-9. Phosphoric acid plant, Case II.	134
A-10. Phosphoric acid plant, Case III.	135
A-11. Reichhold urea process.	137

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Correction factor relating test and deviation vectors.	22
2. Apparent convergence of Cavett problem by successive substitutions for different cut sets.	28
3. Constants defining variable interaction.	36
4. Eigenvalues for model systems.	36
5. Iterations to converge simple example systems.	67
6. Iterations to converge complex example systems.	68
7. Iterations to converge urea system.	70
8. Apparent system eigenvalues for urea system.	71
9. Computational order of Figure 13 for various cut sets.	100
10. Convergence and dominant eigenvalues with various cut sets for salt-water mixing example.	109
11. Convergence and dominant eigenvalues with various cut sets for the hydrocarbon separation system.	110
12. Convergence and dominant eigenvalues with various cut sets for phosphoric acid plant.	110
13. Convergence of Cavett problem by ASYM for equivalent cut sets.	115

OPTIMAL CONVERGENCE OF COMPLEX RECYCLE PROCESS SYSTEMS

1. INTRODUCTION

The steady state simulation of chemical processes has become an important tool for use in process development, evaluation of alternative plant designs, and improvement of existing processes. Thus the development of improved simulation techniques has become a vigorously pursued branch of chemical engineering. The results of such work have made possible the evaluation of detailed and complex mathematical models and have greatly expanded the ability of the engineer to analyze problems.

One of the key elements in the development of simulation techniques is the executive system concept for modular simulation. In the modular approach each process unit is described mathematically by a set of equations which calculate output variables such as component flow rates, temperature, and pressure from the input variables and the equipment parameters. The unit computation is formulated in a general fashion so that it can be used for many different processes. An executive routine handles the mechanics of processing the input and output information, the flow of information between units, and the iteration and convergence of any recycle nets in the process.

In acyclic systems unit computations proceed from the known feedstocks to the end of the process. When recycle is present, however, the calculations become iterative, and often sophisticated techniques are employed in order to speed the convergence of the computations. In highly complex systems just the identification of the recycle nets and the subsequent defining of the order of calculation become difficult. When alternative possibilities exist for the order of calculation, the one which will result in the fastest computation of the solution is the most desirable one to choose. A basis for making this choice is needed.

The objectives of this work are first to examine in detail the asymptotic convergence acceleration techniques and to determine the optimal way in which they should be applied by developing a criterion for their application. Secondly, the problem of cut set selection for the most rapid convergence is to be considered and a method of selecting the best possible cut sets developed.

A number of observations regarding the asymptotic method are made in this work. The type of behavior necessary for successful application of the method is noted. Detection of system convergence and selection of streams for testing convergence are examined. A technique for utilizing the intermediate iterative variables to predict the number of iterations necessary to converge a system is presented. Limits on the promotion coefficients are derived and used to develop

a criterion for when to apply the acceleration equation. This criterion is related to the rate of convergence for each variable and does not require the user to preset any tolerance levels.

The asymptotic method utilizing this criterion is demonstrated with a number of examples and compared to other variations of the asymptotic method. The basis for comparison is the number of system evaluations required to reach a given solution accuracy.

The problem of selection of initial values in order to make a fair comparison of cut sets is examined, followed by an evaluation of single recycle loops and the effects of selecting different cut streams on convergence. The multiple recycle loop problem is shown to be analogous to the single loop problem. It is shown that many of the alternative cut sets are essentially equivalent and will ultimately converge at the same rate. The cut sets in the equivalent group always calculate the units of each particular loop in the sequence obtained by merely following the loop. This leads to the requirement that no loop be cut more than once. An analytical analysis of a single variable, multiple loop problem yields the result that any cut set not in the equivalent group will converge at a slower rate than the sets in the equivalent group. Process examples are used to demonstrate that non-equivalent cut sets will also converge slower in the more general multiple variable, multiple recycle loop problem. The examples demonstrate that the equivalent cut sets are also more amenable to convergence promotion by the asymptotic method than the non-equivalent cut sets.

2. LITERATURE SURVEY AND MATHEMATICAL FOUNDATIONS

Cut Set Selection

The chemical process flowsheet can be represented as a directed graph (digraph) which consists of nodes connected by directed edges. The nodes represent mathematical models of the unit processes, and the edges represent the information pertaining to the physical streams entering and leaving the units. For systems with no recycle streams, the flow of information is in the forward direction only, and the order of calculation of the nodes is easily determined. When recycle is present, however, the digraph of a system is cyclic and the calculation of the system becomes more involved.

The calculation of systems with cyclic nets can be handled in two ways as pointed out by Kehat and Shacham (10). The first method is simultaneous solution of all of the nodes. This technique is effective for systems which can be linearized and was used successfully in certain applications by Rosen (25) and Nagiev (19, 20). Cavett (2) noted that this method is difficult to apply in general simulation programs.

The alternative method of calculation is that of tearing or cutting the cyclic nets. This is accomplished by guessing values for the variables in a stream, calculating nodes along the net to the torn stream, and iteration. Frequently a set consisting of more than one

stream will have to be cut. In order to implement this technique it is necessary to identify the cyclic nets (partition the digraph) and then find an optimal tearing scheme so that the node calculations can be ordered. Generally the two parts of the problem have been attacked together, although they can be considered separately.

The "optimal" tearing scheme has been defined usually as the one which results in the fewest number of iterative variables. The reasoning behind this is that the computational work will be less for fewer iterative variables. The use of this criterion of optimality is generally accompanied by a recognition that this does not always result in cut sets which converge a system in the minimum number of iterations.

Sargent and Westerberg (28) state that a preferred criterion would be to select the cut set which minimizes the total calculation time. They indicate that weighting the different variables in proportion to the difficulty they introduce when occurring as iterative variables and then minimizing the sum of the weights of the cut streams would be advantageous. The idea of using weights was also discussed by Christensen and Rudd (3) and Upadhye and Grens (31), but no method of assigning weights other than the number of variables in a stream was presented. In spite of this lack of an acceptable criterion for defining an optimal cut set, a number of schemes were set forth for various less than optimal criteria.

The simplest method of finding a minimal cut set is to compare all the possibilities. While this is too laborious for large systems, it is the method used by PACER (5) for up to three cuts.

Rubin (27) presented the first tearing procedure. It was based on an adjacency matrix, with the matrix elements corresponding to the number of variables in the streams. He sought to re-order the matrix to minimize the total number of variables in the cut streams. A counter-example was given which showed that his method did not find the global minimum.

Sargent and Westerberg (28) presented a set of rules for combining the nodes of a digraph and for selecting the cut set which would minimize the sum of the weights of the cut streams.

The problem of determining the minimum number of recycle parameters was discussed by Lee and Rudd (17). They assumed that some method of locating the cycles in a digraph would be available and mentioned the works of Norman (21) and Steward (30) as possible techniques to use. A cycle/stream matrix was used to order the calculations and to tear the system by identifying cycles contained within one another and nodes with only one input.

Forder and Hutchison (6) recognized that the objective of cut set selection was to find a sequence of calculation which would minimize the time taken to compute the associated iterations. They reasoned that this would be a function of the total number of cut variables and so

sought to minimize that total. They first ordered the flowsheet into blocks containing the cyclic nets and generated a Boolean cycle matrix which was reduced in the manner of Lee and Rudd (17). An added feature of their work allowed user specification of preferred streams to be included in the cut set if at all possible.

The algorithm of Sargent and Westerberg (28) was extended by Christensen and Rudd (3). Their algorithm consisted of three parts. The first part orders calculation as far as possible and identifies cycles. The second systematically merges edges and nodes to simplify the nets. This was often sufficient in the cases tried, but in others it was necessary to use the third part of the algorithm. This involved the deletion of "index nodes", all of whose inputs and outputs were eligible to be recycle streams.

Upadhye and Grens (31) stated that a minimization of computational effort is desired and that this could be approached through the minimization of the summation over the torn streams of some form of weightings. These weightings would be associated with the variables and would be functions of the variables. Their algorithm for starting with a cyclic process and ending with the desired acyclic process is based upon systematic tearing and the use of dynamic programming.

Barkely and Motard (1) introduced the idea of using signal flow graph analysis to reduce a cyclic system. This representation is the reverse of the digraph in that the nodes represent the streams and the

edges represent the process units. The decomposition procedure is accomplished by cutting nodes rather than edges. Their algorithm consists of carrying out the following steps:

1. Identification of maximal cyclic nets
2. Reduction of the graph to intervals
3. Elimination of self-loop nodes
4. Processing of nodes associated with two-way edges
5. Cutting nodes with the maximum number of output edges

This method finds the minimum number of cut elements to render a problem acyclic.

Genna and Motard (7) presented the first attempt to compare cut sets based upon information generated as iterative calculations were carried out. They reasoned that the cut set which would converge the fastest would also have the smallest eigenvalues and thus based their selection of cut set upon an estimate of these values. The method was not successful.

Convergence Promotion

Once some cut set has been selected, it is necessary to carry out the iterative calculations and hopefully to accelerate their convergence in some manner. Several related methods have been proposed and used for this task with varying degrees of success.

Solving the iterative calculations amounts to solving a set of generally nonlinear equations which can be represented in the form:

$$\bar{x} = \bar{F}(\bar{x}) \quad 2.1$$

where \bar{x} is a vector representing the variables in the cut streams and \bar{F} represents the functional relationships defined by the process. The most straightforward method of solution is that of successive or direct substitution. In this method the new guess for each iteration is simply set equal to the calculated value from the previous iteration so that 2.1 becomes:

$$\bar{x}_{i+1} = \bar{F}(\bar{x}_i) \quad 2.2$$

The advantages of this method have been pointed out by Rinard and Ripps (24) and Shacham and Motard (29). The method must converge for a physically realizable system for an initial guess which is close enough to the solution. The disadvantage is that frequently many iterations are required for convergence.

A number of methods for accelerating the convergence of recycle calculations have been developed. These are classified by Kehat and Shacham (11) as methods deriving from successive substitution (asymptotic methods) and Newton methods.

Cavett (2), Kleisch (14), Kwon (15), and Orbach (22) have compared various methods applied to chemical engineering problems.

Newton type methods, because of the large storage requirements and the large amount of computation required per iteration, have been regarded as too costly in spite of their usually rapid convergence. Asymptotic methods involve comparatively little extra computation and storage and have been used quite successfully in accelerating the convergence of a number of systems.

An asymptotic convergence technique assumes that a variable approaches its asymptotic value according to some equation form. Past values of the variable are used to evaluate the constants in the assumed equation form and to predict the asymptote (solution). Various approximations have been used in these methods such as a simple exponential (9, 13), rectangular hyperbola (2, 8), and compound exponential (13). The methods have been used on single and multiple variable problems.

Isakson (9) assumed that successive values of an iteration followed the form of a simple exponential. That is, when one value is considered the first in a series, the value obtained after I iterations would be given by:

$$x_I = x_\infty + Ae^{-aI} \quad 2.3$$

By writing Equation 2.3 for three successive iterations, the constants a and A are eliminated and the resulting promotion

equation is:

$$x_{\infty} = x_{I-2} + \frac{x_{I-1} - x_{I-2}}{1 - \frac{x_I - x_{I-1}}{x_{I-1} - x_{I-2}}} \quad 2.4$$

Isakson noted that, because of erratic behavior in the first few iterations, it was frequently necessary to wait until five or six iterations had been performed before applying the promotion equation.

The hyperbolic method (2) was based upon the assumption that successive points in a direct iteration procedure could be approximated by a rectangular hyperbola. The acceleration equation used was:

$$x_{\infty} = x_I - (x_{I-1} - x_I) \left(1 - \frac{T}{S}\right) \quad 2.5$$

where

$$T = \frac{x_{I-2} - x_{I-1}}{x_{I-1} - x_I}$$

and

$$S = 1.0 \quad \text{for } 0.5 < T < 1.5$$

$$S = T - 1.0 \quad \text{otherwise}$$

Heaslet and Lomax (8) encountered very slow rates of convergence using Equation 2.5 when a poor choice of the starting value was used.

Kelin (13) observed the following law could approximately

describe the iterative behavior of a direct substitution problem after a relatively few iterations:

$$x_{I+1} - x_I = (x_{I_0+1} - x_{I_0}) e^{-a(I-I_0)} \quad 2.6$$

or

$$(I-I_0)a = \ln \frac{x_{I_0+1} - x_{I_0}}{x_{I+1} - x_I} \quad 2.7$$

This is an alternate form of Isakson's Equation 2.3. The asymptote is found from:

$$x_\infty = x_{I_0} + \frac{x_{I_0+1} - x_{I_0}}{1 - e^{-a}} \quad 2.8$$

If I_0 is replaced by $I-2$ and I by $I-1$ in Equations 2.7 and 2.8, Equation 2.8 reduces to Equation 2.4.

For problems in which the simple exponential approach to an asymptote was not followed, Klein suggested the following equations were sometimes good approximations:

$$x_{I+1} - x_I = \{x_{I_0+1} - x_{I_0}\} \{1 + a(I-I_0)\}^{-p} \quad 2.9$$

and

$$x_{I+1} - x_I = \alpha e^{-a(I-I_0)} + \beta e^{-b(I-I_0)} \quad 2.10$$

with α , β , a , b , and p constants.

These equations were recommended for extrapolating during the beginning period of iteration to the point where the logarithmic damping takes over.

Equation 2.10 leads to the following equation for the asymptote:

$$x_{\infty} = x_{I-4} + \frac{\Delta x_0 [\Delta x_1^2 - \Delta x_2 (\Delta x_0 + \Delta x_1) + \Delta x_3 \Delta x_0] + \Delta x_1 (\Delta x_1^2 - \Delta x_0 \Delta x_3)}{[\Delta x_1^2 - \Delta x_2 (\Delta x_0 + \Delta x_1) + \Delta x_0 \Delta x_3] - [\Delta x_1 \Delta x_3 - \Delta x_2^2]} \quad 2.11$$

where

$$\begin{aligned} x_0 &= x_{I-3} - x_{I-4} \\ \Delta x_1 &= x_{I-2} - x_{I-3} \\ \Delta x_2 &= x_{I-1} - x_{I-2} \\ \Delta x_3 &= x_I - x_{I-1} \end{aligned} \quad 2.12$$

Wegstein (32) developed a convergence acceleration technique for single variable systems based upon the equation:

$$x_{\infty} = q \cdot x_I + (1-q)x_{I+1} \quad 2.13$$

where

$$q = \frac{a}{a-1} \quad 2.14$$

$$a = \frac{x_{I+1} - x_I}{x_I - x_{I-1}} \quad 2.15$$

Equation 2.13 was applied at every other step after an initial three steps by successive substitutions.

The method was developed from geometric considerations, but results in the same promotion equation as 2.4. This can be seen by eliminating q and a from Equation 2.13.

Wegstein reported convergence was always obtained in single variable cases, and that an optimum value of the q parameter fell within certain ranges depending upon the iterative behavior of the system.

Kwon (15) observed that the Wegstein method worked well for multivariable systems with little or no variable interactions, but failed otherwise.

Kleisch (14) developed a bounded Wegstein method for use in multivariable process simulation problems. Noticing that instabilities arose from the Wegstein method predicting negative flow rates, he placed an upper limit of $+1$ on the q parameter. For variables with q greater than or equal to $+1$, he set q equal to zero. Thus a direct substitution step was taken for that variable.

Kliesch applied this method to two process examples and concluded that it was the most efficient technique studied.

A different method of bounding the Wegstein method was used in the CHESS (18) executive. If q did not fall within the bounds of $-10 < q < 0$, no acceleration was taken for that variable. In addition the sign of each q was compared on succeeding iterations and no acceleration was taken when the sign changed. Under this system

only component flow rates were accelerated. Other variables were iterated by successive substitutions.

Kwon (15) revised the method of Klein (13) and applied it to multivariable recycle calculations. His revised asymptotic method (RAM) was based on the observation that each variable in a multivariable system can be approximated by an exponential approaching its asymptote as in a single variable system. The promotion equation used was the same as the Wegstein Equation 2.13 with the same method of calculation for the q parameter. The most important feature of Kwon's method was that the following bounds were employed.

$$\begin{aligned} x_{\infty} &= x_I & q > 1.0 \\ x_{\infty} &= 2x_{I-1} - x_I & q \leq -50 \end{aligned} \quad 2.16$$

Kwon then extended the approach and employed a second algorithm for RAM which uses five iterative values and Equation 2.11 to take the first acceleration step. Equation 2.13 was used thereafter. This was done to account for the fact that often the iterative behavior did not settle down to the logarithmic damping until after several iterations had been calculated.

Kwon applied the two algorithms of RAM to nine different systems and compared their convergence to successive substitutions, to modified successive substitutions, to the three variations of Wegstein's method, to the Newton-Raphson method and to Rosen's (25)

generalized false position method. His recommendation was that the second algorithm of RAM was the best to use for the wide variety of cases studied.

Mathematics of the Asymptotic Method

Orbach (22) contributed greatly to the development of asymptotic methods by establishing the mathematical basis for their use as follows:

If the general iteration of Equation 2.2 is expanded in a Taylor Series to first order terms about the solution, the result for a multi-variable problem is:

$$\bar{x}_{I+1} = A\bar{x}_I + \bar{b} \quad 2.17$$

where

$$A = \left. \frac{\partial \bar{F}}{\partial \bar{x}} \right|_{\bar{x}_s} = \text{Jacobian} = J$$

$$\bar{b} = \bar{F}(\bar{x}_s) - A\bar{x}_s$$

Note that any convergent process will eventually approach this behavior. The particular solution to 2.17 is:

$$\bar{x}_I - \bar{x}_s = \sum_{i=1}^n c_i \bar{z}_i \lambda_i^I \quad 2.18$$

where

i = subscript for each variable

n = number of variables

I = iteration number

λ_i = eigenvalues of A

\bar{z}_i = eigenvectors of A for λ_i

$c_i = \bar{W}_j^T (\bar{x}_0 - \bar{x}_s) / \bar{W}_j^T \bar{z}_i$

\bar{W}_i = eigenrows of A for λ_i

The necessary and sufficient condition for 2.18 to converge is:

$$|\lambda_1| < 1.0 \quad 2.19$$

where λ_1 is the largest or dominant eigenvalue and the eigenvalues are subscripted in descending order. If 2.19 holds then $|\lambda_j/\lambda_1|_{i=1}^n$ approaches zero as I increases and Equation 2.18 approaches:

$$\bar{x}_I - \bar{x}_s = c_i \bar{z}_i \lambda_1^I \quad 2.20$$

For a single variable

$$x_{jI} - x_{js} = c_{ij} z_{ij} \lambda_1^I \quad 2.21$$

Applying 2.21 at three successive iterations and solving for x_{js}

gives

$$x_{js} = x_{jI-1} + \frac{x_{jI} - x_{jI-1}}{1 - \lambda_1} \quad 2.22$$

where

$$\lambda_1 = \frac{\bar{x}_{jI} - \bar{x}_{jI-1}}{\bar{x}_{jI-1} - \bar{x}_{jI-2}} \quad 2.23$$

This again is the same form of promotion equation that was described earlier.

In vector form 2.22 is:

$$\bar{x}_s = \bar{x}_{I-1} + \frac{\bar{x}_I - \bar{x}_{I-1}}{1 - \lambda_1} \quad 2.24$$

Analogous to 2.23, the eigenvalue can be estimated as shown by Orbach and Crowe (23) as:

$$|\lambda_1'| = \frac{\|\bar{x}_I - \bar{x}_{I-1}\|}{\|\bar{x}_{I-1} - \bar{x}_{I-2}\|} \quad 2.25$$

The prime is used to differentiate the estimate obtained for the dominant eigenvalue from that obtained by Equation 2.23.

Orbach used a variation of 2.25 to calculate λ_1 at each iteration, and when the fractional change in its value was less than some preselected tolerance, an acceleration step utilizing Equations 2.22 and 2.23 was taken. This was the first establishment of a criteria which decided when an acceleration step should be taken.

If the assumption is made that two eigenvalues dominate the

iterative behavior Equation 2.18 reduces to:

$$\bar{x}_I - \bar{x}_s = c_1 \bar{z}_1 \lambda_1^I + c_2 \bar{z}_2 \lambda_2^I \quad 2.26$$

Application of this equation to five iterative values and elimination of the constants results in Equation 2.11, the second algorithm of Kwon's RAM (15).

Convergence Tests

If the iterative Equation 2.2 is convergent, the values of \bar{x}_I will eventually approach the solution value \bar{x}_s to within some arbitrarily small difference as I goes to infinity. In practice, the iteration is carried out until some point near the solution is reached, and that point is taken as the solution.

Ideally, it is desired to iterate until the deviation from the solution is less than some specified tolerance. The fractional deviation from the solution is defined by:

$$D_{iI} = \frac{x_{is} - x_{iI}}{x_{is}} \quad 2.27$$

or in vector form:

$$\bar{D}_I = [I \bar{x}_s]^{-1} [\bar{x}_s - \bar{x}_I] \quad 2.28$$

The term $[\bar{I}\bar{x}]$ will be used to indicate a diagonal matrix with elements \bar{x}_i . This vector is used as a convergence test only rarely in special studies since the solution values must be known for its application.

The usual convergence test vector is that which gives the fractional step size from one iteration to the next:

$$\bar{\theta}_I = [\bar{I}\bar{x}_I]^{-1}[\bar{x}_I - \bar{x}_{I-1}] \quad 2.29$$

The term $[\bar{I}\bar{x}_{I-1}]$ is sometimes used in the denominator.

Convergence is considered to be attained when some norm of this vector is less than a specified tolerance. This test does not actually show how far the iterative values are from the solution. When an iteration converges very slowly due to a dominant eigenvalue near one in magnitude, the tolerance must be set lower than for a rapidly converging iteration in order to obtain the same absolute accuracy. That is, an iteration may be taking quite small steps and still be insufficiently close to the solution to be considered converged to the desired accuracy. One method of averting this difficulty is simply to set the tolerance much smaller than is actually desired or needed. This is not particularly desirable as it can result in unnecessary calculations.

Orbach and Crowe (23) show the relationship between the test and deviation vectors as follows. Substitution of 2.20 for \bar{x}_I and

\bar{x}_{I-1} into 2.29 gives:

$$\begin{aligned}\bar{\theta}_I &= [\bar{I x}_I]^{-1} [c_1 \bar{z}_1 \lambda_1^I - c_1 \bar{z}_1 \lambda_1^{I-1}] \\ \bar{\theta}_I &= [\bar{I x}_I]^{-1} [c_1 \bar{z}_1 (\lambda_1^I - 1) \lambda_1^{I-1}]\end{aligned}\quad 2.30$$

Equations 2.18 and 2.28 give:

$$\bar{D}_I = [\bar{I x}_s]^{-1} [-c_1 \bar{z}_1 \lambda_1^I] \quad 2.31$$

When $[\bar{I x}_I]^{-1}$ is approximately equal to $[\bar{I x}_s]^{-1}$ these can be combined to give:

$$\bar{D}_I \approx \bar{\theta}_I \frac{\lambda_1}{1-\lambda_1} \quad 2.32$$

Or if norms are used:

$$\begin{aligned}\|\bar{D}_I\| &= \|\bar{\theta}_I \cdot \frac{\lambda_1}{1-\lambda_1}\| \\ &= \|\bar{\theta}_I\| \cdot \left| \frac{\lambda_1}{1-\lambda_1} \right|\end{aligned}$$

Since λ_1 is a scalar constant, and since $|\lambda_1| < 1$, $1-\lambda_1 > 0$

so that:

$$\|\bar{D}_I\| = \|\bar{\theta}_I\| \cdot \frac{|\lambda_1|}{1-\lambda_1} \quad 2.33$$

This relationship allows estimation of the deviation vector from the test vector when an iteration has proceeded sufficiently that it is near linear behavior and is dominated by the largest eigenvalue. The effect of the correction can be seen in Table 1.

Table 1. Correction factor relating test and deviation vectors.

λ	Correction Factor	λ	Correction Factor
0.1	0.111	-.10	0.09
0.2	0.250	-.20	0.1667
0.3	0.429	-.30	0.231
0.4	0.667	-.40	0.286
0.5	1.000	-.50	0.333
0.6	1.500	-.60	0.375
0.7	2.333	-.70	0.412
0.8	4.000	-.80	0.444
0.9	9.000	-.90	0.474
0.95	19.000	-.95	0.487
0.98	49.000	-.98	0.495

Thus as the dominant eigenvalue approaches one or zero the difference between the test and deviation vectors approaches infinity. The latter case is of little concern because it is a rapidly converging one. The case when the eigenvalue nears unity, however, demonstrates dramatically how checking the test vector can lead to apparent convergence long before the deviation vector satisfies the desired tolerance.

The use of this correction is rather restricted by the assumption of the dominant eigenvalue so that it cannot be used early in an iteration or immediately following an acceleration step which perturbs the approach to the linear form. The factor may still be used, however, by deferring its calculation until the iteration has approached the linear form and by not calculating a new value for several iterations following an acceleration step.

The recommended procedure for testing convergence is then to calculate the dominant eigenvalue at each iteration from Equation 2.25 until a nearly constant value is obtained. The correction factor, ϕ , is then calculated from:

$$\phi = \frac{|\lambda_1|}{1-\lambda_1} \quad 2.34$$

After an acceleration step has been taken retain the last estimate of ϕ until the estimate of the dominant eigenvalue again approaches a constant. The estimate of ϕ can then be updated with each iteration until another acceleration step is taken.

3. ASYMPTOTIC CONVERGENCE PROMOTION

Streams Tested for Convergence

A further consideration in the convergence test is whether or not all of the variables in a system are satisfactorily converged when the iterative variables have reached the specified tolerance level. If this is not the case then all of the variables in the system will not be obtained to the accuracy desired. In an iterative system values of all variables are available at each iteration so that convergence checks can be made on any variables or streams desired. In several instances the recommendation has been made that only the iterative streams need be checked.

Consider the simple recycle loop of Figure 1. Two streams with n variables each are contained in the loop, and either of the streams may be chosen as the iterative stream. Depending upon which one is chosen the general iterative relationships are:

$$\bar{x}_{1,I} = F_1(\bar{x}_{1,I-1}) \quad 3.1$$

$$\bar{x}_{2,I} = F_2(\bar{x}_{2,I-1}) \quad 3.2$$

As the iteration proceeds, either iterate would approach the linear form:

$$\bar{x}_{1,I} - \bar{x}_{1,S} = \sum_{j=1}^n c_j \bar{z}_j \lambda_j^I \quad 3.3$$

$$\bar{x}_{2,I} - \bar{x}_{2,S} = \sum_{j=1}^n c_j' \bar{z}_j' \lambda_j^I \quad 3.4$$

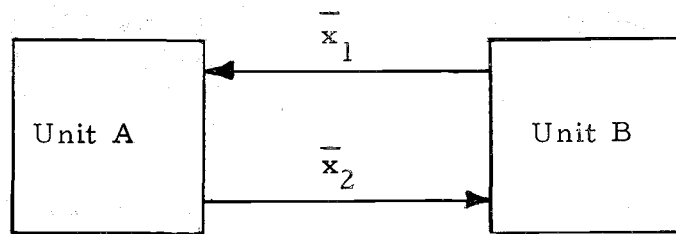


Figure 1. Simple recycle loop.

The values of the eigenvalues, λ_j , would be the same for either stream, but the eigenvectors, \bar{z}_j , and the constants c_j would not generally be the same.

Premultiplication by $[\bar{I}x_{1,S}]^{-1}$ and $[\bar{I}x_{2,S}]^{-1}$ and utilization of the definition of the deviation vector, Equation 2.28, yields:

$$\bar{D}_{1,I} = [\bar{I}x_{1,S}]^{-1} \sum_{j=1}^n c_j \bar{z}_j \lambda_j^I \quad 3.5$$

$$\bar{D}_{2,I} = [\bar{I}x_{2,S}]^{-1} \sum_{j=1}^n c_j' \bar{z}_j' \lambda_j^I \quad 3.6$$

As I becomes large and the dominant eigenvalue controls the convergence rate, these become:

$$\bar{D}_{1,I} = \{[\bar{I}x_{1,s}]^{-1} c_1 \bar{z}_1\} \lambda_1^I \quad 3.7$$

$$\bar{D}_{2,I} = \{[\bar{I}x_{2,s}]^{-1} c_2 \bar{z}_2\} \lambda_1^I \quad 3.8$$

Since the bracketed terms are constant for a given starting guess the result is:

$$\bar{D}_{1,I} = \bar{c}_1 \lambda_1^I \quad 3.9$$

$$\bar{D}_{2,I} = \bar{c}_2 \lambda_1^I \quad 3.10$$

Therefore:

$$\|\bar{D}_{1,I}\| : \|\bar{D}_{2,I}\| = \|\bar{c}_1\| : \|\bar{c}_2\| \quad 3.11$$

This result can obviously be extended to single loops with any number of streams. It can also be shown to hold for values of the test vector $\|\bar{\theta}_I\|$.

Furthermore, if the dominant eigenvalue is a multiple eigenvalue the result also holds. In this case, Equations 3.5 and 3.6 become:

$$\bar{D}_{1,I} = [\bar{I}x_{1,s}]^{-1} \sum_{j=1}^m c_j \bar{z}_j \lambda_1^I \quad 3.12$$

$$\bar{D}_{2,I} = [\bar{I}x_{2,s}]^{-1} \sum_{j=1}^m c_j^I \bar{z}_j^I \lambda_1^I \quad 3.13$$

where m is the multiplicity of the dominant eigenvalue. Equations 3.12 and 3.13 yield the same result as Equation 3.11 namely, that the deviations for the different streams approach constant ratios which depend upon the initial guess and the system characteristics.

From Equation 3.11 it is seen that when one stream "converges" it is not necessarily true that all streams have reached a point within the same tolerance.

An example of the large differences which can occur is given by the hypothetical ethylene dichloride process outlined in Appendix A. Due to the structure and characteristics of the system, three of four recycle streams have identical values for the norm of the test vector at each iteration. The other stream, however, has a value of the norm of the test vector which is approximately one-sixth of that for the other streams. Since this example has a large dominant eigenvalue (approximately 0.93), convergence is slow and once the stream with the smaller test vector has converged to some tolerance level an additional 25 iterations are required to converge the other streams to the same tolerance. If the stream with the smaller test vector is chosen as the iterative stream and the process "converged", the variables in the other streams will not be converged to the same tolerance.

In a multiple loop recycle problem, different values of the solution eigenvalues may occur for various choices of the streams making up the iterative set of streams, and a simple relationship analogous to Equation 3.11 cannot be obtained. Nonetheless, there will still be differences in the deviation and test vectors for the various possible cut sets at a given point in the iteration. For example, the hydrocarbon separation system first presented by Cavett (2) was solved by successive substitutions using zero initial guesses and the values of two norms of the test vector printed at each iteration. Examination of these values yield the "point of convergence" for various combinations of streams as cut sets. The results are shown in Table 2.

Table 2. Apparent convergence of Cavett problem by successive substitutions for different cut sets.

Cut Set Streams	tol = .01	tol = .001
2, 7	30	61
5, 7	38	69
2, 10	31	62
2, 8	27	55
3, 7	30	61
5, 9, 10	38	69
3, 9, 10	34	68
5, 9, 8	38	69
3, 9, 8	34	65
All	38	69

As can be seen, the apparent convergence is faster when some sets are checked than with others. The actual convergence, however, is not obtained until all streams in the system satisfy the criterion.

This difference in apparent convergence of different streams has not been previously emphasized. The assumption that all streams are converged when the iterative stream(s) are converged is analogous to assuming that all variables within a stream have converged when one of the variables has converged. The latter is an assumption that is never made. Both Cavett (2) and Rinard and Ripps (24) are investigators who tested for convergence only upon the iterative streams. In many cases the effect of this difference may be negligible but an awareness of the possible effects is essential. An actual simulation may involve a large number of streams so that checking convergence on all streams is quite time consuming. It is not necessary to check all of the streams. Consider a simulation used for an optimization which has only a few variables appearing in the performance function. These few variables are those for which the specified accuracy is essential. The other variables would have absolutely no effect on the optimization and their convergence would be of no importance. In any given problem convergence should be checked only for those streams which are important for the given case.

Previous work on comparison of cut sets (7) has not specified clearly what streams were tested for convergence. Checking

convergence only on the cut streams would not be suitable for comparisons because the same degree of convergence would not be obtained for various cases. In order to make a fair comparison a common basis for convergence testing must be defined. One common basis is to test all streams in a system for convergence. Another basis is to test the same selected streams in each case to be compared. The important point is that of consistency for all comparisons.

The Cavett problem was used to demonstrate this principle. The system was solved using the second algorithm of Kwon's (15) convergence method, RAM (2), for the cut sets {3, 7} and {5, 9, 10}. When convergence is tested only on the cut streams with a tolerance of 10^{-2} , the former set converges in 20 iterations and the latter in 24 iterations. Yet when all streams are tested, both runs converge in 26 iterations. With a tolerance of 10^{-3} the effect of testing all streams for convergence is seen to be even more pronounced. In this case set {3, 7} converges in 30 iterations and set {5, 9, 10} converges in 43 iterations when only cut streams are tested. When all streams are tested the convergence of the two sets is attained in 41 and 43 iterations. Thus the apparent large difference in behavior of these two sets is insignificant when a common basis of comparison is used.

In summary this section has pointed out that any stream may be used to test an iterative calculation for convergence, and that testing a selected set of streams does not guarantee that all streams will be converged to the desired tolerance. The streams which should be tested are those of greatest interest in a given problem. Studies

which compare cut sets must use a common basis for testing convergence in order to make honest and meaningful comparisons.

Behavior of Iterative Systems

The approach taken in this work was to combine the theoretical observations of systems with a detailed study of the behavior of the variables during iteration. In order to do this for a wide variety of systems, the need for the ability to adjust some of the system characteristics soon became obvious. The objective was to specify the eigenvalues of a system and the degree of interaction between the variables in order to observe the effects of the different characteristics.

In order to meet this need a method of creating a model system of equations was devised. The basis for the construction of the model system is the set of basic facts from algebra (4):

$$\begin{aligned}
 \sum_i \lambda_i &= \text{tr } J & 3.14 \\
 \sum_{i < j} \lambda_i \lambda_j &= \text{tr}_2 J \\
 \sum_{i < j < k} \lambda_i \lambda_j \lambda_k &= \text{tr}_3 J \\
 \prod_i \lambda_i &= |J|
 \end{aligned}$$

These equations give n relationships between the eigenvalues and Jacobian terms of an n -dimensional system. Specification of the n eigenvalues would leave the n^2 Jacobian terms to define. By setting the ratio of the $(n-1) \cdot n$ off diagonal terms to the n diagonal terms, it become possible in theory to determine the magnitudes of the diagonal terms and thus the entire Jacobian matrix. This is done by selecting the values of:

$$c_{ij} = j_{ij}/j_{ii} \quad 3.15$$

When all $c_{ij} \ll 1$, the system would have weakly interacting variables; when $c_{ij} > 1$ the interaction is strong.

The Jacobian matrix thus found can be used to derive a set of equations which will have the desired eigenvalues and variable interaction. Some form can be chosen for the equations in terms of constant parameters to be determined. The equation form thus gives the Jacobian in terms of the parameters. Equating this to the values calculated on a term by term basis allows calculation of the parameters. If in addition, the solution values for the system are specified, a system form involving up to $n+1$ constants can be chosen.

An example of this procedure can be seen by developing the three dimensional case.

The Jacobian is defined by:

$$J = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} \quad 3.16$$

Equations 3.14 become:

$$\lambda_1 + \lambda_2 + \lambda_3 = j_{11} + j_{22} + j_{33} = a_1 \quad 3.17$$

$$\lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3 = \begin{vmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{vmatrix} + \begin{vmatrix} j_{11} & j_{13} \\ j_{31} & j_{33} \end{vmatrix} + \begin{vmatrix} j_{22} & j_{23} \\ j_{32} & j_{33} \end{vmatrix} = a_2 \quad 3.18$$

$$\lambda_1 \lambda_2 \lambda_3 = |J| = a_3 \quad 3.19$$

The a values can be found after specification of the eigenvalues. If the constants of Equation 3.15 are specified, the off diagonal terms in 3.18 and 3.19 can be eliminated. Thus when the determinants are expanded:

$$a_1 = j_{11} + j_{22} + j_{33} \quad 3.20$$

$$a_2 = (1 - c_{12}c_{21})j_{11}j_{22} + (1 - c_{13}c_{31})j_{11}j_{33} + (1 - c_{23}c_{32})j_{22}j_{33} \quad 3.21$$

$$a_3 = \{(1 - c_{23}c_{32}) - c_{12}(c_{21} - c_{23}c_{31}) + c_{13}(c_{21}c_{32} - c_{31})\} \cdot j_{11}j_{22}j_{33} \quad 3.22$$

By defining the constants to simplify the equations:

$$a_4 = (1 - c_{12}c_{21}) \quad 3.23$$

$$a_5 = (1 - c_{13}c_{31}) \quad 3.24$$

$$a_6 = (1 - c_{23}c_{32}) \quad 3.25$$

$$a_7 = \{1 - c_{23}c_{32} - c_{12}(c_{21} - c_{23}c_{32}) + c_{13}(c_{21}c_{32} - c_{31})\} \quad 3.26$$

The resulting equations are:

$$a_1 = j_{11} + j_{22} + j_{33} \quad 3.27$$

$$a_2 = a_4 j_{11} j_{22} + a_5 j_{11} j_{33} + a_6 j_{22} j_{33} \quad 3.28$$

$$a_3 = j_{11} j_{22} j_{33} \cdot a_7 \quad 3.29$$

The resulting system is difficult to solve directly for the Jacobian terms. The equations can be rearranged to give:

$$\begin{aligned} & \left(j_{11}^3 - a_1 j_{11}^2 + \frac{a_2}{a_4} j_{11} - \frac{a_3 a_6}{a_4 a_7} \right) \cdot \left(j_{11}^3 - a_1 j_{11}^2 + \frac{a_2}{a_4} j_{11} - \frac{a_3 a_6}{a_5 a_7} \right) \\ & = -j_{11}^3 \frac{a_3}{a_7} \frac{(a_4 - a_5)^2}{a_4 a_5} \end{aligned} \quad 3.30$$

In order to simplify the calculations, the restriction was added to the system that $a_4 = a_5$ which requires the constants defining the variable interaction to be chosen so that $c_{12} \cdot c_{21} = c_{13} \cdot c_{31}$. This results in the cubic equation:

$$j_{11}^3 - a_1 j_{11}^2 + \frac{a_2}{a_4} j_{11} - \frac{a_3 a_6}{a_4 a_7} = 0 \quad 3.31$$

which can be solved directly for three values of j_{11} . The procedure in this work was to choose one of the roots and to proceed to calculate the rest of the Jacobian.

This procedure was programmed on the digital computer so that the eigenvalues and constants defining the variable interaction could be selected and the Jacobian calculated.

For the initial study a linear equation form was chosen:

$$x_i = \sum_{j=1}^n \beta_{i,j} x_j + \beta_{i,n+1} \quad 3.32$$

The Jacobian terms are thus:

$$j_{ij} = \beta_{ij} \quad 3.33$$

Then the constants were known for $j \leq n$ once the Jacobian had been calculated. The final constants were found after the solution values were specified:

$$\beta_{i,n+1} = x_{i,s} - \sum_{j=1}^n \beta_{i,j} x_{j,s} \quad 3.34$$

The equation sets thus generated were then solved iteratively so that the behavior could be followed. Jacobians and systems were generated for two sets of constants defining variable interaction, one weak and one strong. These are listed in Table 3. A number of eigenvalues were used as defined by Table 4.

Table 3. Constants defining variable interaction.

Constant	Jacobian 1 Weak Interaction	Jacobian 2 Strong Interaction
c_{12}	.002	3.0
c_{13}	-.015	-2.5
c_{21}	-.075	2.0
c_{23}	.009	-1.8
c_{31}	.01	-2.4
c_{32}	-.011	0.9

Table 4. Eigenvalues for model systems.

System #	Eigenvalues			Jacobian #
1	.9	.6	.1	1
2	.9	.6	.1	2
3	.9	.1	.05	1
4	.9	.1	.05	2
5	.9	.8	.1	1
6	.9	.8	.1	2
7	.9	.9	.1	1
8	.9	.9	.1	2
9	.9	.85	.8	1
10	.9	.85	.8	2
11	.9	.9	.9	2

The apparent eigenvalues and step sizes at each step were printed out. The study of these values demonstrated different patterns for the strongly and weakly interacting systems. For comparative purposes, Figures 2, 3, 4, and 5 show the approach of the apparent eigenvalue to the dominant eigenvalue for several systems.

Several trends are immediately apparent. The first is that the apparent eigenvalues for all of the variables of the strongly interacting systems tend to be near each other in magnitude and follow a pattern of a gradual decrease to a large negative value preceding a sharp rise and then a gradual decrease to the actual magnitude of the dominant eigenvalue. For this type of system it is seen that a large number of iterations are required before the system is behaving linearly.

The weakly interacting system shows different characteristics in that initially each variable maintains a value very near the actual eigenvalue dominating its convergence. As the variable converges, the decrease, sharp increase, and decrease to the dominant eigenvalue pattern is repeated, but the apparent eigenvalues of the different variables are much more distinct. Here it is seen that the system behaves linearly early in the iteration (as evidenced by the near constant eigenvalues), goes through a transition period as the variables fall under the influence of the dominant eigenvalue, and then again approaches linear behavior.

From Figures 6, 7, 8, and 9 it can be seen that the step size for

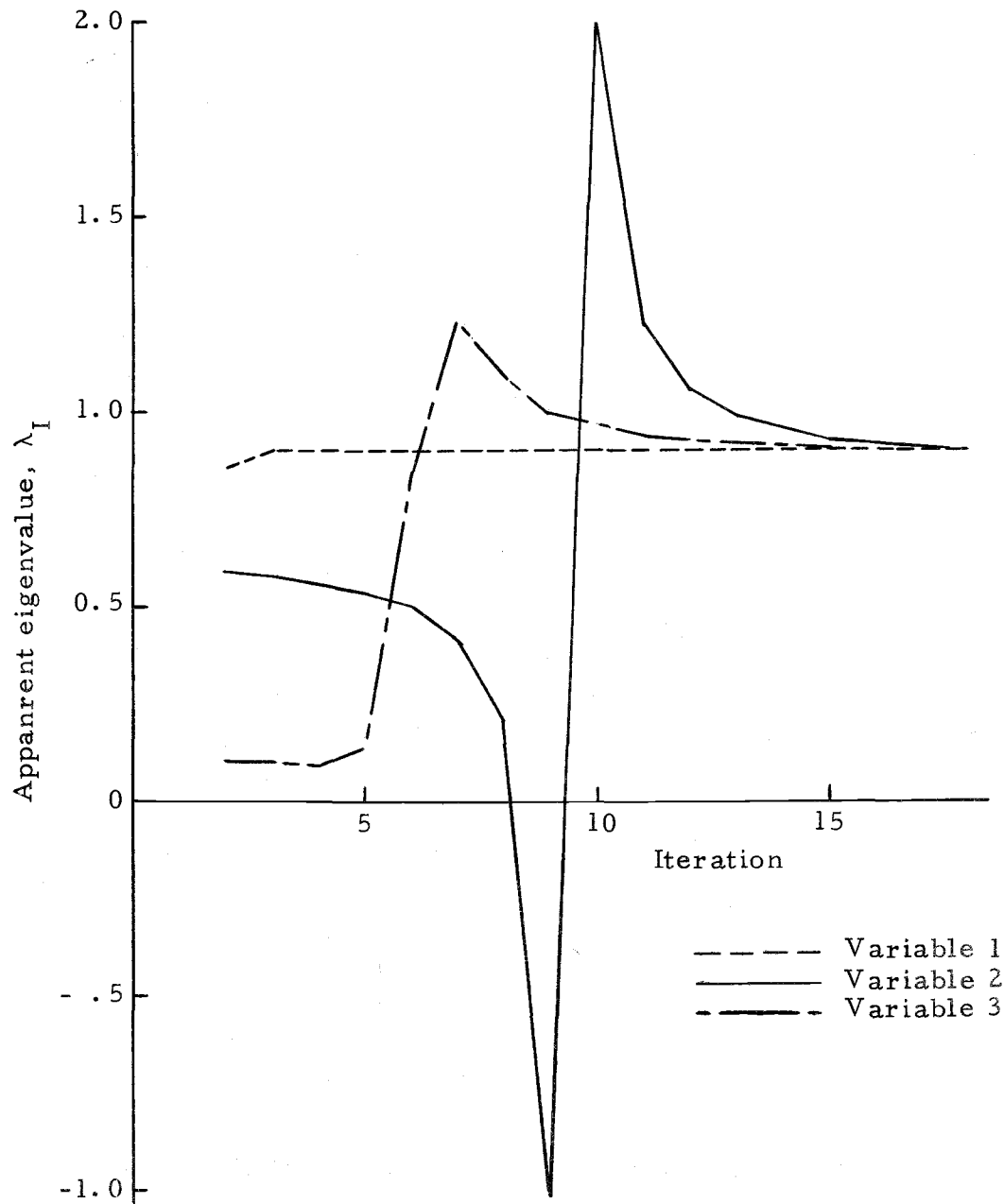


Figure 2. Apparent eigenvalues of model system 1.

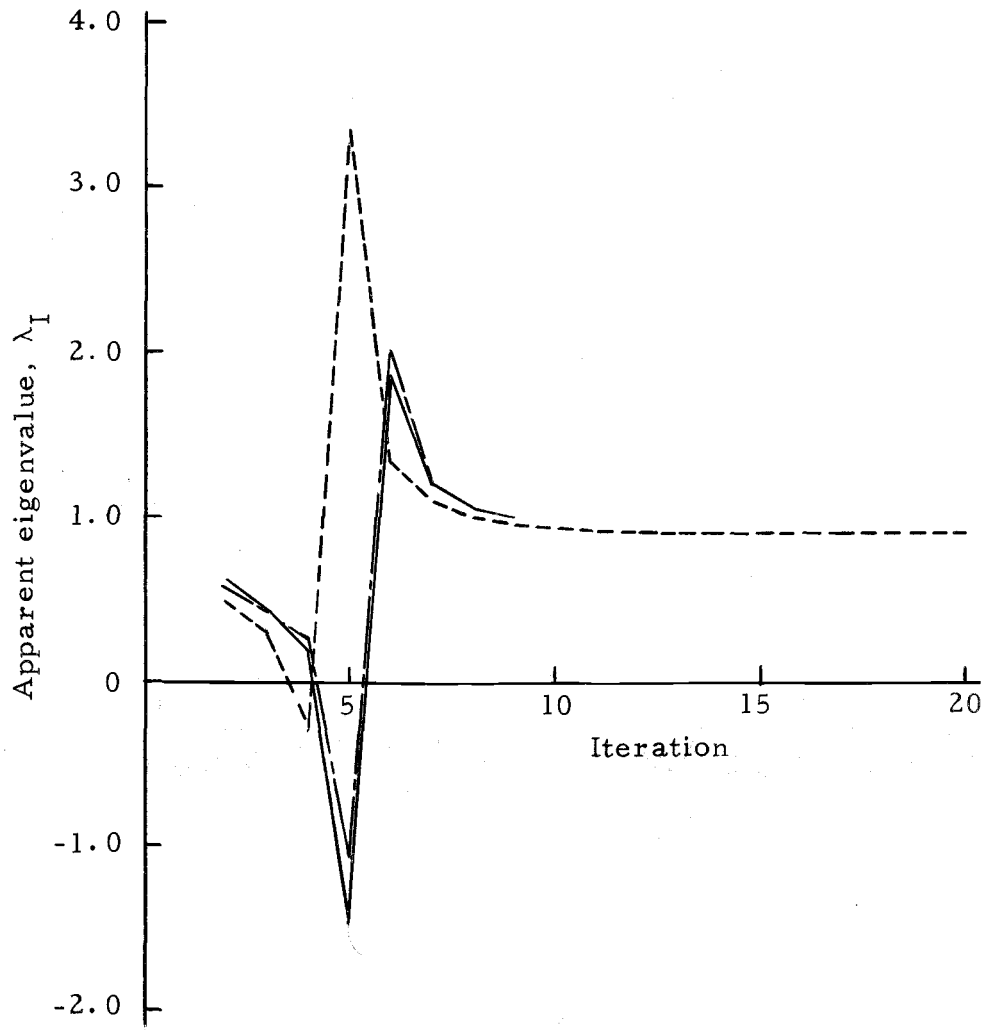


Figure 3. Apparent eigenvalues of model system 2.

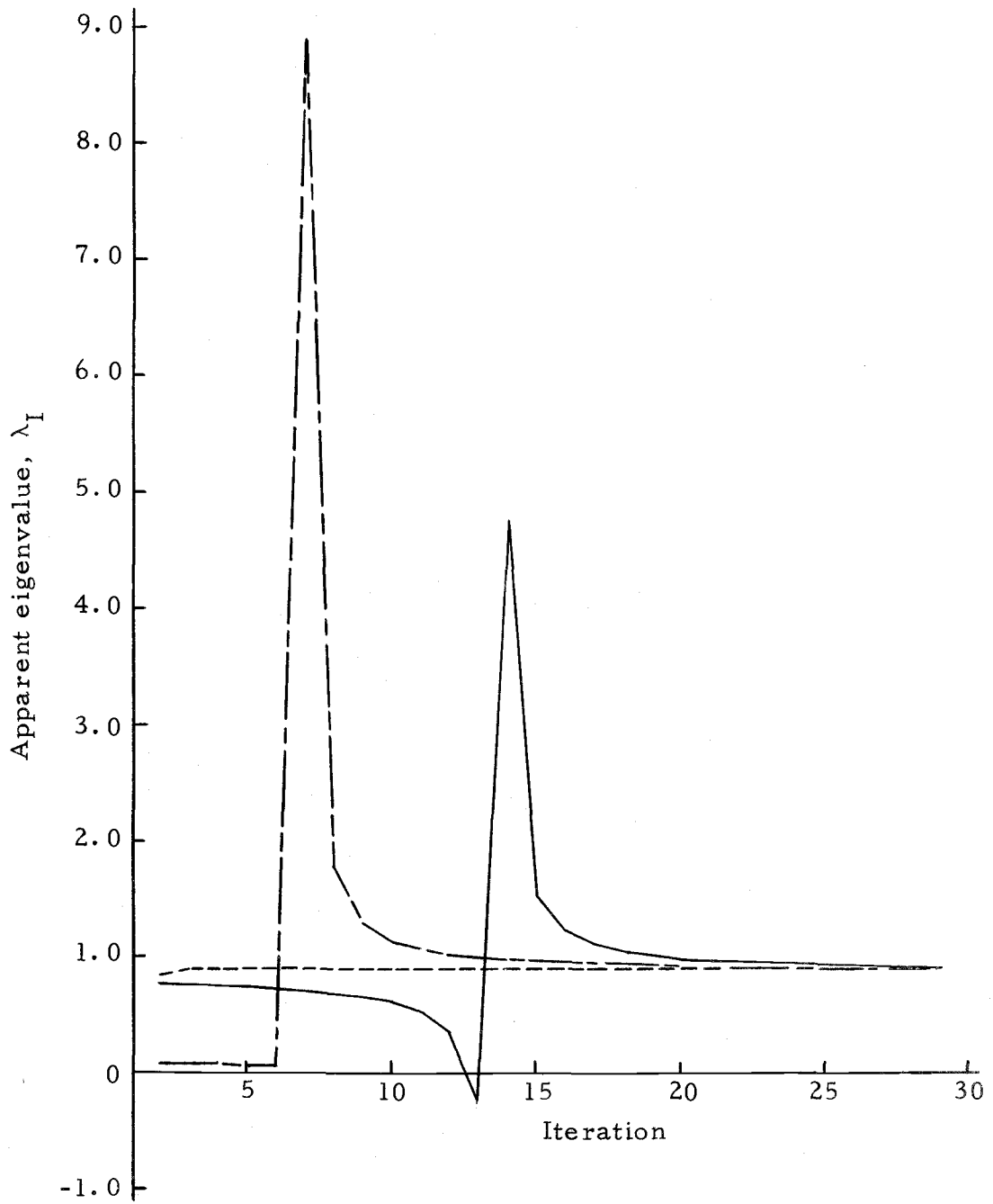


Figure 4. Apparent eigenvalues of model system 5.

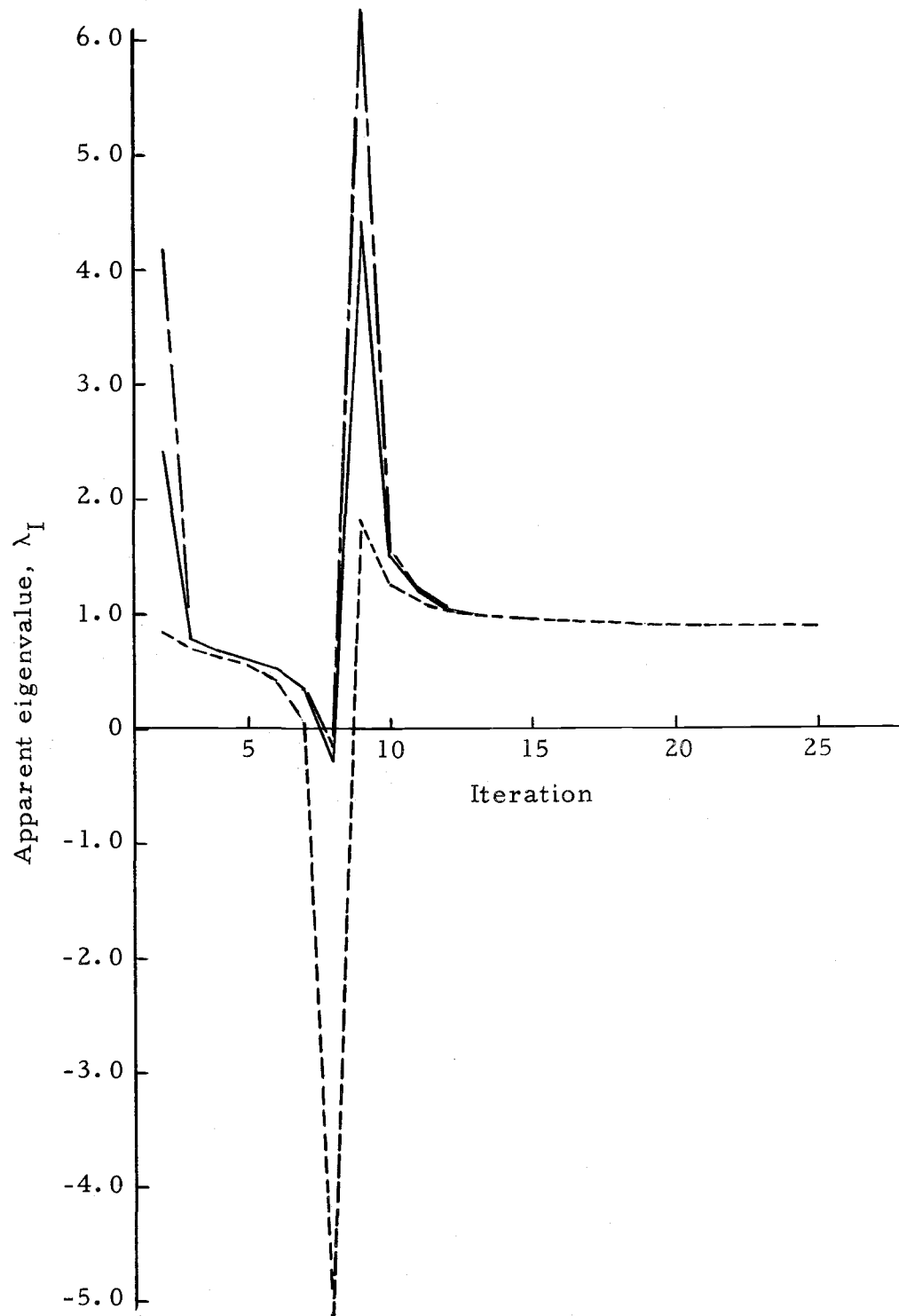


Figure 5. Apparent eigenvalues of model system 6.

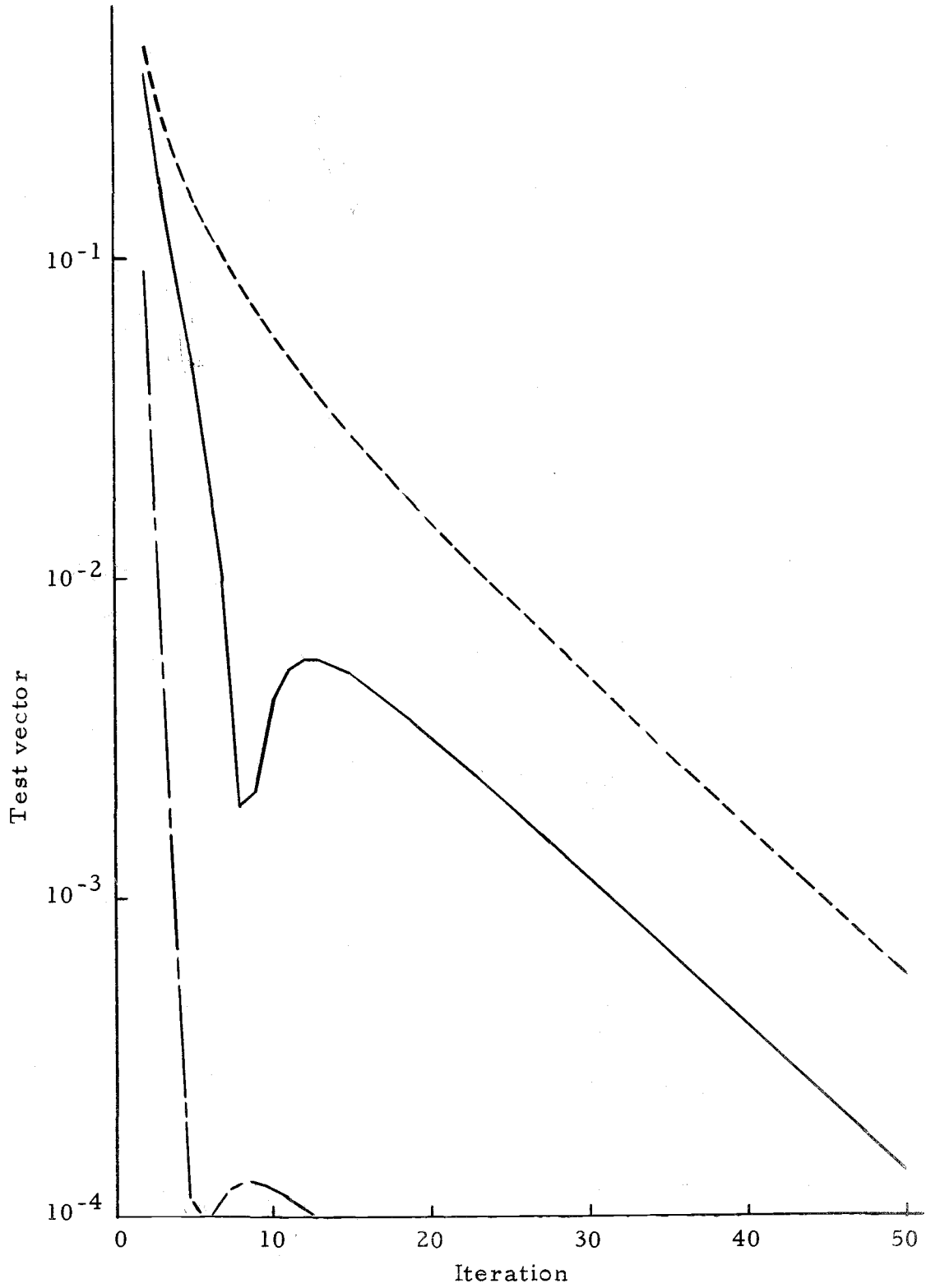


Figure 6. Test vector versus iteration, system 1.

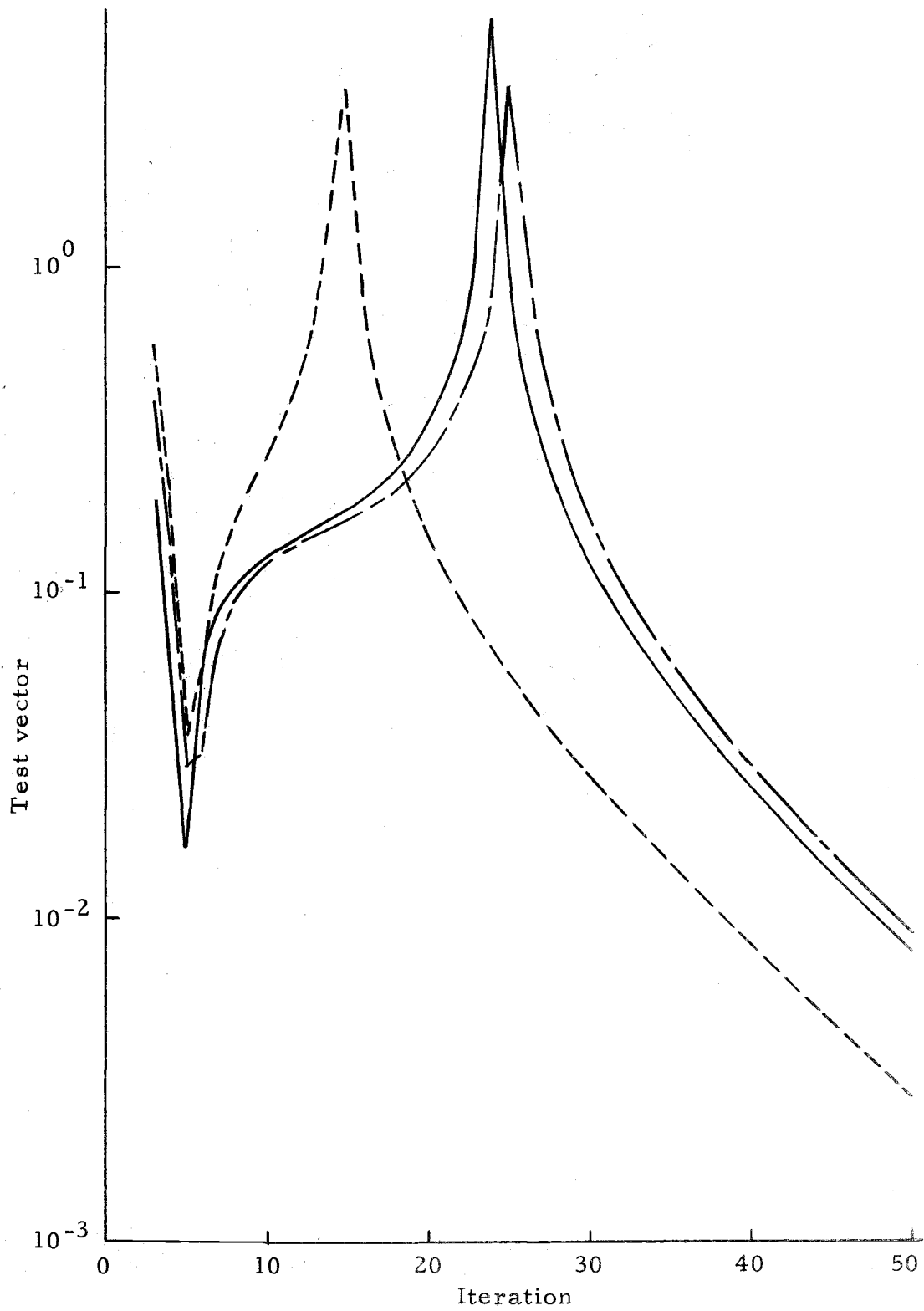


Figure 7. Test vector versus iteration, system 2.

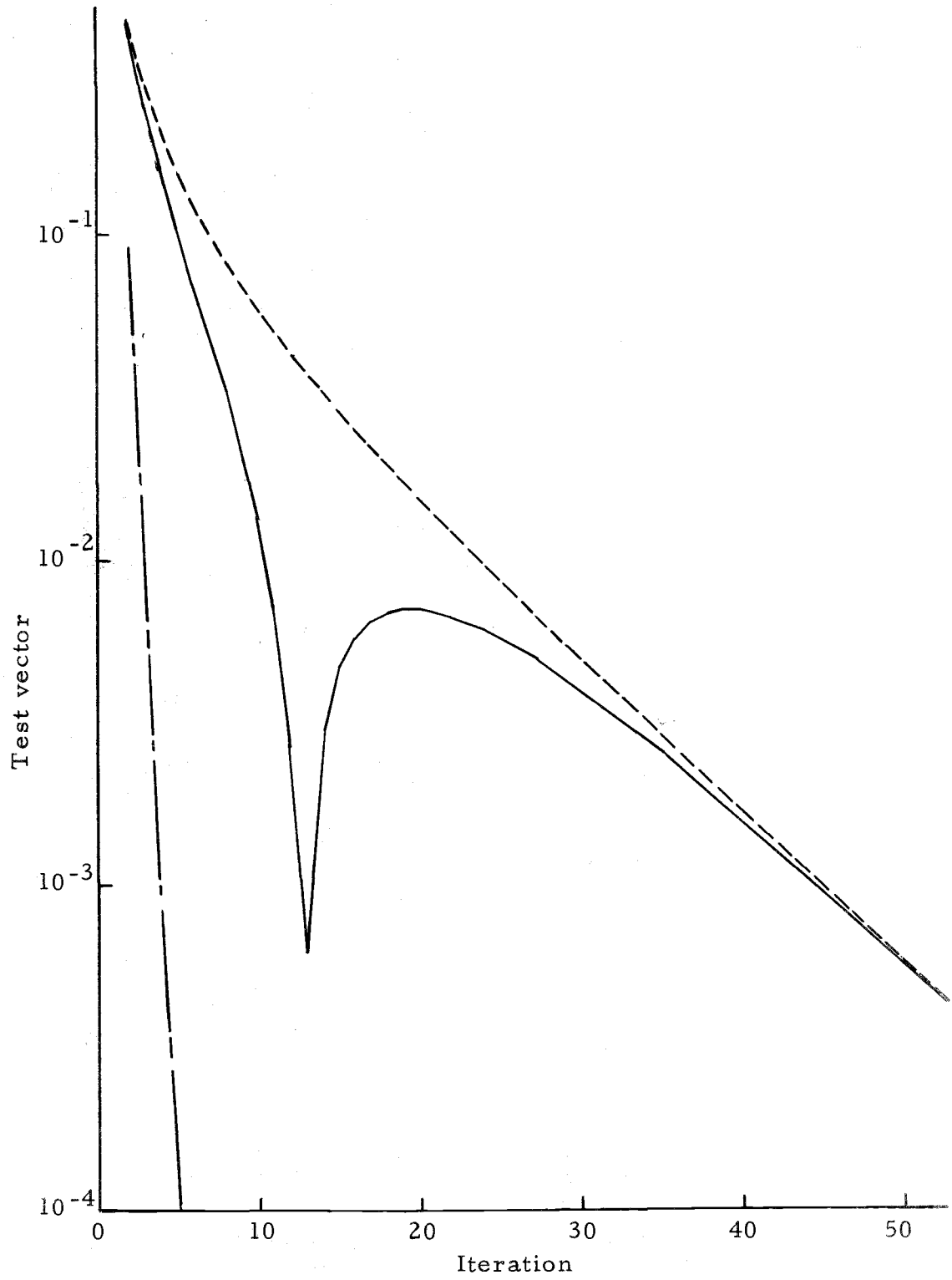


Figure 8. Test vector versus iteration, system 5.

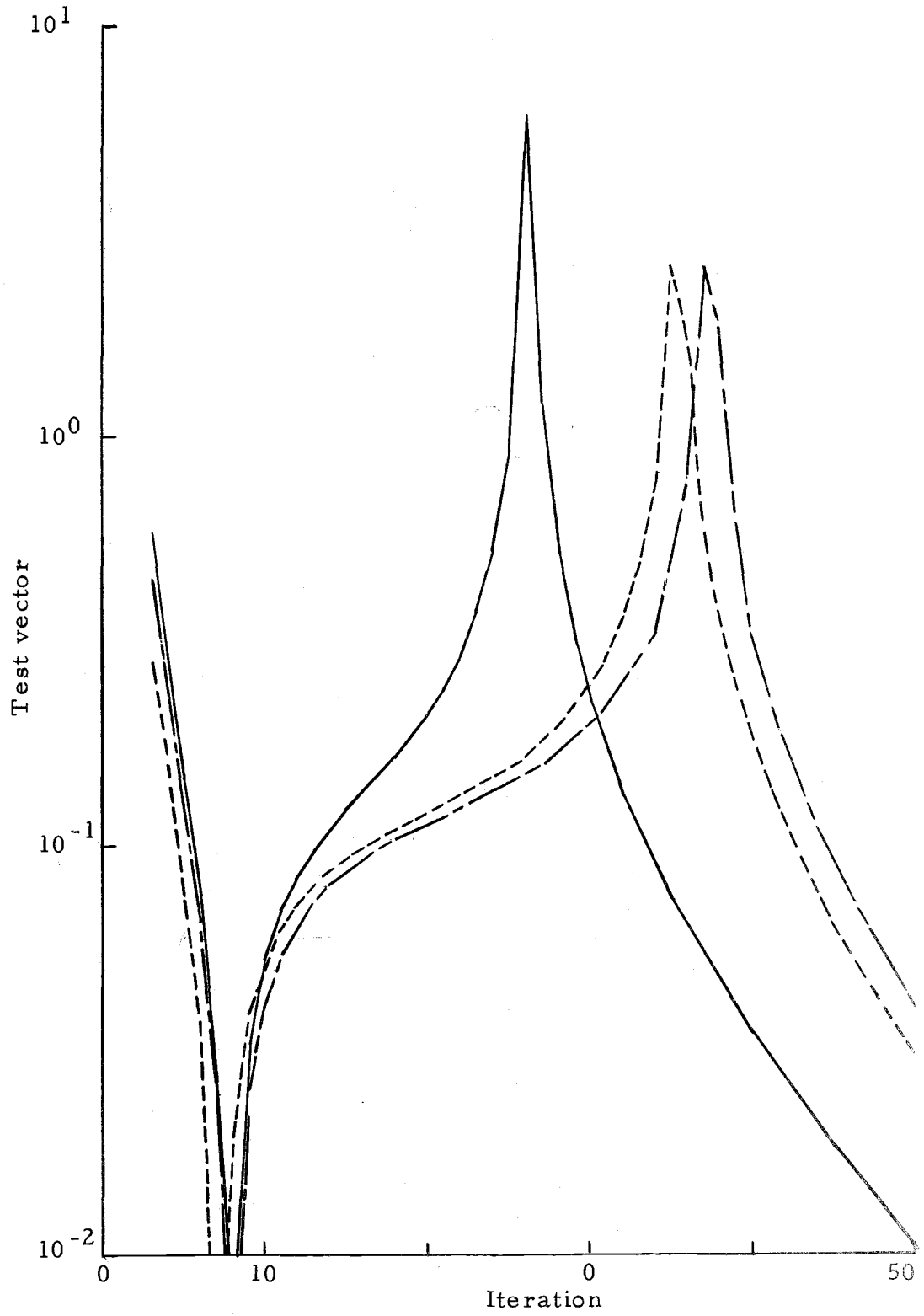


Figure 9. Test vector versus iteration, system 6.

the variables also varies for the different amounts of variable interaction. The values are of the same order of magnitude for the strongly interacting cases at any given iteration, but vary widely for the weakly interacting cases. This is because in the latter case the variables dominated by the small eigenvalues converge rapidly to the point where they fall under the influence of the dominant eigenvalue. Thus during this transition period it is not so important that these variables be behaving linearly and be amenable to acceleration since they are already converged more than the other variables.

This type of analysis is also applicable to nonlinear systems. Preliminary work was done with several equation forms, but difficulties were encountered for the strongly interacting cases with several large eigenvalues. The systems generated would have more than one set of roots and would converge to a set other than the desired one.

This analysis pointed out the requirements an acceleration technique must meet in order to be able to be successful for various types of systems. Any acceleration technique must not accelerate at regular intervals with no attention given to actual system behavior. A good technique must be able to tell from the iterative behavior when an acceleration step will be better than a step by direct substitution.

Estimation of the Number of Iterations to Converge

The dominant eigenvalue can be used to estimate the number of iterations to converge to a desired tolerance if the iteration is continued from a given point by successive substitution. Such an estimate can be used to decide if convergence acceleration is worthwhile for a given problem.

Suppose that the deviation value of the iterative variables in a problem at iteration K is given by δ_K . Then from Equations 2.27 and 2.20:

$$\delta_K = \frac{x_K - x_s}{x_s} \doteq c_1 z_1^K \lambda_1^K$$

$$\delta_K = c_1' \lambda_1^K \tag{3.35}$$

Assuming the iteration has become sufficiently linear, the deviation at iteration $I+K$ will be:

$$\delta_{K+I} = \delta_c \tilde{c}_1' \lambda_1^{K+I} \tag{3.36}$$

In order to find the number of iterations necessary to go from iteration K to a deviation of δ_c at convergence, divide Equation 3.36 by Equation 3.35:

$$\frac{\delta_c}{\delta_K} = \frac{c_1^{K+I} \lambda_1^c}{c_1^K \lambda_1^K} = \lambda_1^{I_c} \quad 3.37$$

taking logarithms gives:

$$\ln \frac{\delta_c}{\delta_K} = I_c \ln \lambda_1 \quad 3.38$$

or

$$I_c = \frac{\ln \delta_c / \delta_K}{\ln \lambda_1} \quad 3.39$$

Equation 3.39 reduces to Orbach's (22) definition of the rate of convergence if δ_c / δ_K is set equal to 0.10. He defined the rate of convergence as the reciprocal of the number of iterations required to reduce the deviation by a factor of 10, or:

$$R = \frac{1}{I_{10}} = -\ln \lambda_1 \quad 3.40$$

Equation 3.39 was derived independently by Shacham and Motard (29) by a different method of analysis.

Extrapolation to the Solution

At a given iteration there exists a value of the apparent eigenvalue, not necessarily equal to the solution value and not necessarily exactly the same for all variables which, if used in Equation 2.22, would give the solution. It can be found by solving Equation 2.22 with

$$x_{\infty} = x_s :$$

$$x_{\infty} = x_s = x_{I-1} + \frac{x_I - x_{I-1}}{1 - \lambda_{sI}} \quad 3.41$$

where

$$\lambda_{sI} = \text{value of } \lambda_I \text{ such that } x_{\infty} = x_s$$

Thus

$$x_s - x_{I-1} = \frac{(x_I - x_s) - (x_{I-1} - x_s)}{1 - \lambda_{sI}} \quad 3.42$$

$$1 - \lambda_{sI} = - \frac{x_s - x_I}{x_s - x_{I-1}} + 1 \quad 3.43$$

$$\lambda_{sI} = \frac{x_s - x_I}{x_s - x_{I-1}} = \frac{D_I}{D_{I-1}} \quad 3.44$$

This value, if known, could be used to go exactly to the solution at any given iteration. It will appear in later derivations.

Useful Limits on Apparent Eigenvalues

In order to be able to predict whether or not an acceleration step should be taken at a given iteration, it is necessary to know when it will be advantageous to accelerate. If the apparent eigenvalues for each variable do not fall in the proper ranges, an acceleration step will cause the variables to fall further away from their solutions than retaining the value obtained by successive substitution.

As noted before, previous works have bounded the eigenvalues, generally so that they are less than one in magnitude. This criterion

is not sufficiently accurate to eliminate "bad" acceleration steps.

If acceleration is to be successful, the accelerated value of each variable should be closer to the solution than the value obtained from a direct substitution step. That is:

$$|x_s - x_\infty| < |x_s - x_I| \quad 3.45$$

In order to determine what the apparent eigenvalue must be to satisfy this requirement, first assume that the approach to the solution is a monotonic increasing function so that:

$$x_{I-1} < x_I < x_s \quad 3.46$$

When the absolute value signs are removed from Equation 3.45, two inequalities are obtained:

$$x_s - x_\infty < x_s - x_I \quad 3.47$$

and

$$x_s - x_I < x_\infty - x_s \quad 3.48$$

Simplification of Equation 3.47 gives:

$$x_\infty > x_I \quad 3.49$$

from the acceleration Equation 2.22

$$x_{I-1} + \frac{x_I - x_{I-1}}{1 - \lambda_1} > x_I \quad 3.50$$

$$\frac{x_I - x_{I-1}}{1 - \lambda_1} > x_I - x_{I-1} \quad 3.51$$

Since $x_I > x_{I-1}$ by 3.46

$$\frac{1}{1 - \lambda_1} > 1 \quad 3.52$$

$$1 > 1 - \lambda_1 \quad 3.53$$

$$\lambda_1 > 0 \quad 3.54$$

Similarly, Equation 3.48 gives

$$2x_s - x_I > x_\infty$$

$$2x_s - x_I > x_{I-1} + \frac{x_I - x_{I-1}}{1 - \lambda_1} \quad 3.55$$

Subtraction of x_{I-1} from each side and rearrangement gives:

$$(x_s - x_I) + (x_s - x_{I-1}) > \frac{(x_I - x_s) - (x_{I-1} - x_s)}{1 - \lambda_1} \quad 3.56$$

Noting from 3.46 that $x_s - x_{I-1}$ is positive,

$$\frac{x_s - x_I}{x_s - x_{I-1}} + 1 > \frac{\left(\frac{x_I - x_s}{x_s - x_{I-1}}\right) + 1}{1 - \lambda_1} \quad 3.57$$

Noting that:

$$\lambda_{sI} = \frac{x_s - x_I}{x_s - x_{I-1}} \quad 3.44$$

then

$$\lambda_{sI} + 1 > \frac{1 - \lambda_{sI}}{1 - \lambda_I} \quad 3.58$$

$$\lambda_I < 1 - \left(\frac{1 - \lambda_{sI}}{1 + \lambda_{sI}} \right) \quad 3.59$$

or

$$\lambda_I < \frac{2\lambda_{sI}}{1 + \lambda_{sI}} \quad 3.60$$

If the assumption is made that the approach to the solution is a monotonic decreasing function, the same results are obtained.

When the assumption is that the approach is oscillating about the solution, or

$$x_{I-1} < x_s < x_I \quad 3.61$$

it can be shown that the apparent eigenvalue must satisfy:

$$0 > \lambda_I > \frac{2\lambda_{sI}}{1 + \lambda_{sI}} \quad 3.62$$

in order for a good acceleration step to be taken.

Thus Equations 3.54, 3.60, and 3.62 give the requirement for a worthwhile acceleration step to be taken, namely that the apparent

eigenvalue must satisfy:

$$0 < |\lambda_I| < \left| \frac{2\lambda_{sI}}{1+\lambda_{sI}} \right| \quad 3.63$$

The limit will be defined as the critical eigenvalue so that:

$$|\lambda_{\text{crit}}| \triangleq \left| \frac{2\lambda_{sI}}{1+\lambda_{sI}} \right| \quad 3.64$$

and

$$0 \leq |\lambda_I| \leq |\lambda_{\text{crit}}| \quad 3.65$$

When the solution eigenvalue is negative and approaching one in magnitude it is seen that the limit on λ_I approaches infinity. When the solution eigenvalue is positive, however, the acceptable range for the apparent eigenvalue shrinks appreciably as the magnitude nears one. This can be noted graphically in Figure 10 which shows the critical eigenvalue versus the value which extrapolates to the solution. As can be seen, when the solution eigenvalue approaches one, the value of the apparent eigenvalue must be very good for acceleration to be good.

Thus any acceleration criterion must consider the magnitude of the dominant eigenvalue as well as the change in its apparent value. The criterion must adapt to each particular problem and even to each particular variable within a problem. A set of bounds or a criteria

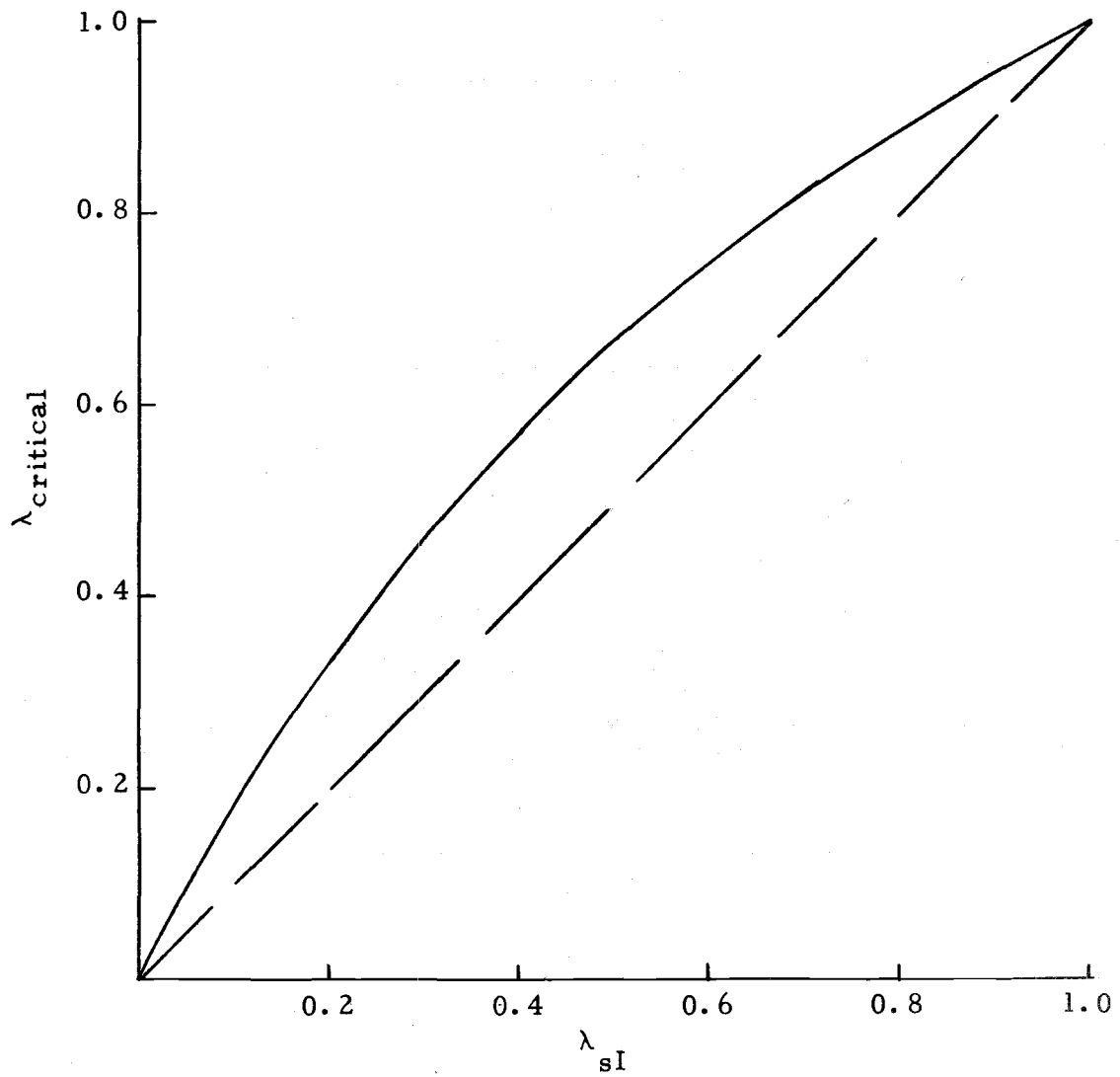


Figure 10. Critical eigenvalues.

which uses fixed values for all cases cannot adequately provide for all cases.

The Acceleration Criterion

All variations of the asymptotic method prior to DEM applied acceleration at regular intervals regardless of the actual behavior of the iteration. The development of a criterion for when to accelerate allows judgment of whether or not the iterative variables are obeying the assumed asymptotic form sufficiently closely to justify using acceleration.

As seen in the discussion of the critical eigenvalue the magnitude of the actual eigenvalue determines how close the apparent eigenvalues must be in order to insure a good acceleration step. Thus the tolerance against which a promotion criterion is tested varies from problem to problem and even from iteration to iteration. With DEM this tolerance must be specified by the user. This tolerance can affect the convergence adversely as a standard tolerance does not exist. The following approximations allow the use of a variable tolerance which adjusts to the given problem and requires no user specification of tolerance.

When the apparent eigenvalue of a variable is itself considered to be a variable approaching its true value with each iteration, its deviation from the true value can be approximated in the same manner as

the deviation of the variable, that is:

$$\Delta\lambda = \frac{\lambda_{\text{true}} - \lambda_I}{\lambda_{\text{true}}}$$

Utilization of Equation 2.32 gives:

$$\begin{aligned} \Delta\lambda &= \theta_\lambda \cdot \frac{\lambda_I}{1-\lambda_I} \\ &= \left(\frac{\lambda_I - \lambda_{I-1}}{\lambda_I} \right) \left(\frac{\lambda_I}{1-\lambda_I} \right) \\ \Delta\lambda &\approx \frac{\lambda_I - \lambda_{I-1}}{1-\lambda_I} \end{aligned} \quad 3.66$$

Or,

$$\lambda_{\text{true}} = \lambda_I (1 \pm \Delta\lambda) \quad 3.67$$

Here λ_I and λ_{I-1} are the apparent eigenvalues at iterations I and $I-1$.

If the upper value of this exceeds the critical eigenvalue, acceleration should not be taken. The critical value can be estimated from 3.64 if the assumption is made that:

$$\lambda_I \approx \lambda_{sI}$$

So

$$\lambda_{\text{crit}} \approx \frac{2\lambda_I}{1+\lambda_I}$$

The fractional difference between λ_I and $\lambda_{\text{crit}, I}$ is thus:

$$\begin{aligned}\delta\lambda &= \frac{\lambda_{\text{cI}} - \lambda_I}{\lambda_I} \approx \frac{1}{\lambda_I} \left\{ \frac{2\lambda_I}{1+\lambda_I} - \lambda_I \right\} \\ &= \frac{2 - (1+\lambda_I)}{1+\lambda_I} \\ \delta\lambda &\approx \frac{1 - \lambda_I}{1 + \lambda_I} \quad 3.68\end{aligned}$$

or

$$\lambda_{\text{crit}} = \lambda_I(1 + \delta\lambda) \quad 3.69$$

As shown in the previous section, for good acceleration 3.65 must be true:

$$0 \leq |\lambda_{\text{true}}| \leq |\lambda_{\text{crit}}| \quad 3.65$$

Substitution of 3.67 and 3.69 gives:

$$|\lambda_I(1 \pm \Delta\lambda)| \leq |\lambda_I(1 + \delta\lambda)|$$

or

$$|\Delta\lambda| \leq |\delta\lambda|$$

From 3.68 it is seen that $\delta\lambda$ is positive for any system with $|\lambda| < 1$ so that:

$$|\Delta\lambda| \leq \delta\lambda \quad 3.70$$

Equation 3.70 provides the test for whether or not an acceleration step should be taken, and it gives a tolerance which adjusts according to the given problem and the stage of the iteration.

In order to apply this criterion it is merely necessary to calculate $\Delta\lambda$ and $\delta\lambda$ at each iteration for each variable and accelerate when Equation 3.70 is satisfied.

It was recognized in the discussion on system behavior that some variables can be nearly converged many iterations sooner than others. It is not necessary to test these variables against the criterion, since it is not necessary to accelerate them. Prior to checking the criterion for each variable the test value (or estimated deviation value) can be compared to the test (deviation) value which is largest. If it is much less than the largest test value the criterion check is simply bypassed and the particular variable is not accelerated.

This practice is consistent with the use of a vector norm to make the criterion check (as in DEM). Such a technique would also minimize the effect of variables near convergence. This technique, however, isolates the particular variables and does not try to accelerate them.

Applications and Comparisons

The asymptotic method utilizing the new acceleration criterion was programmed as subroutine ASYM in a format compatible with the

DISCOSSA (16) executive system. Figure 11 is a flow chart of the basic logic of ASYM. The steps which deal only with bookkeeping tasks unique to DISCOSSA have been omitted to make the flow chart clearer.

DISCOSSA is an executive for steady state process system simulation developed at Oregon State University. The executive handles the input and output operations required in a simulation and provides routines for converging iterative calculations. One of the main features of DISCOSSA is that a simulation is executed by using a main FORTRAN calling program provided by the user. This allows great flexibility, easily provides for man-machine interaction, and is quite simple to use.

ASYM was used to solve 11 equation and process systems and compared to successive substitutions, both versions of RAM, and DEM. The systems are described in Appendix A. Different initial guesses were used to solve each system to provide additional variation.

The total computing time required for the calculations necessary for the asymptotic method will be small compared to the time required to evaluate a process system. Therefore, the total computing time is directly proportional to the number of system evaluations. The number of iterations is therefore used in this study as the basis of comparison of the various methods. Each system was solved until the

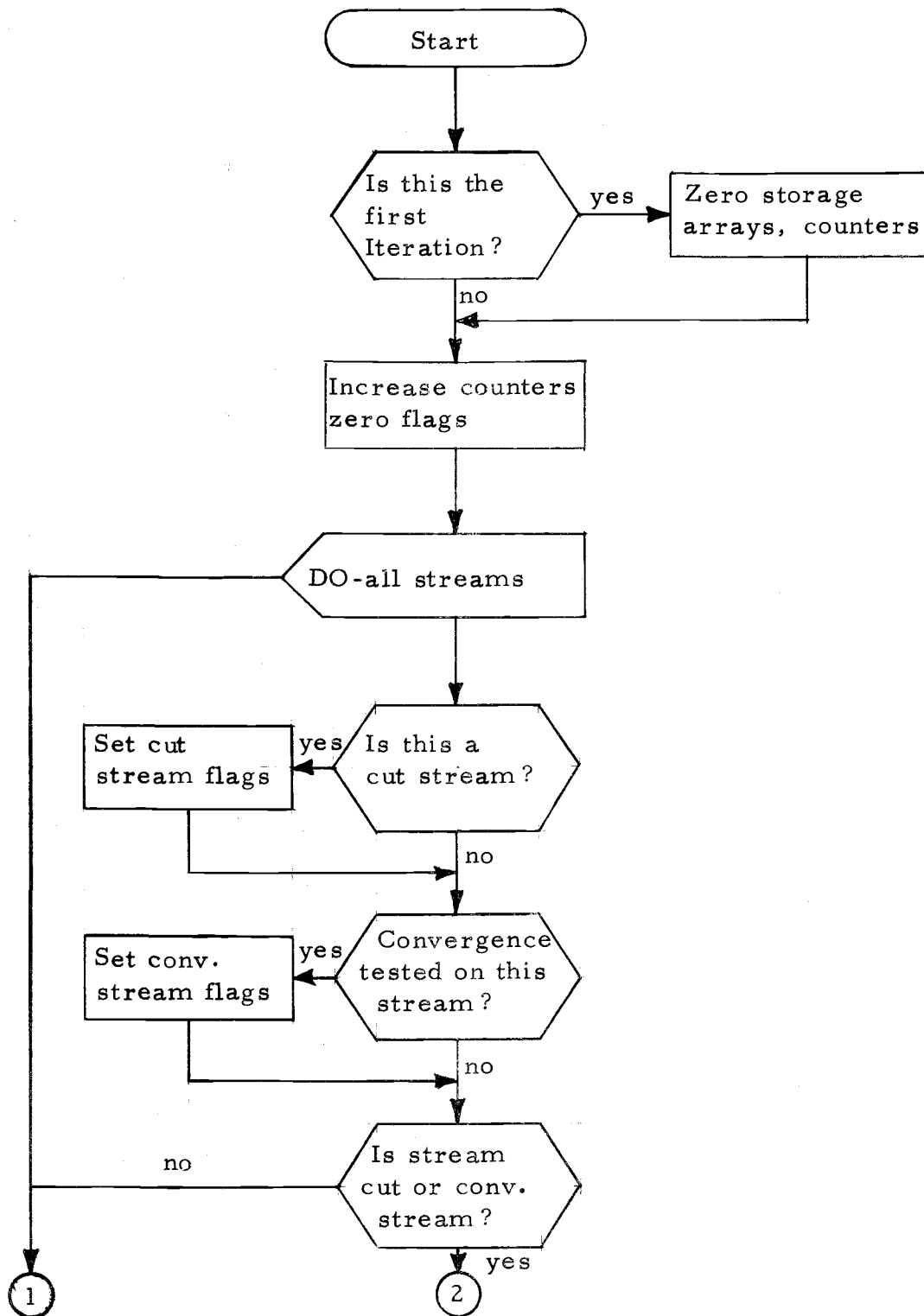


Figure 11. Logic flowchart, subroutine ASYM.

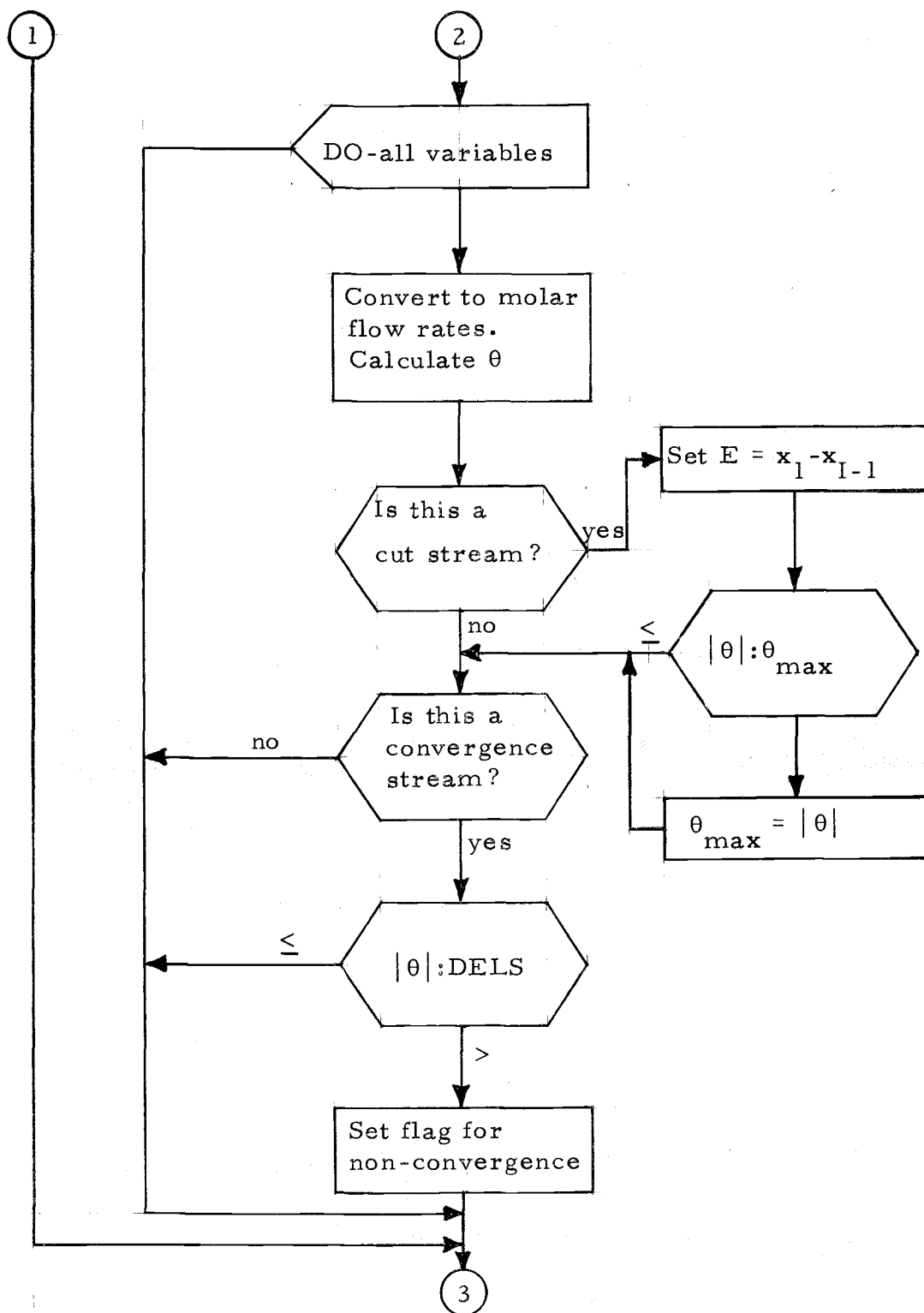


Figure 11. Continued.

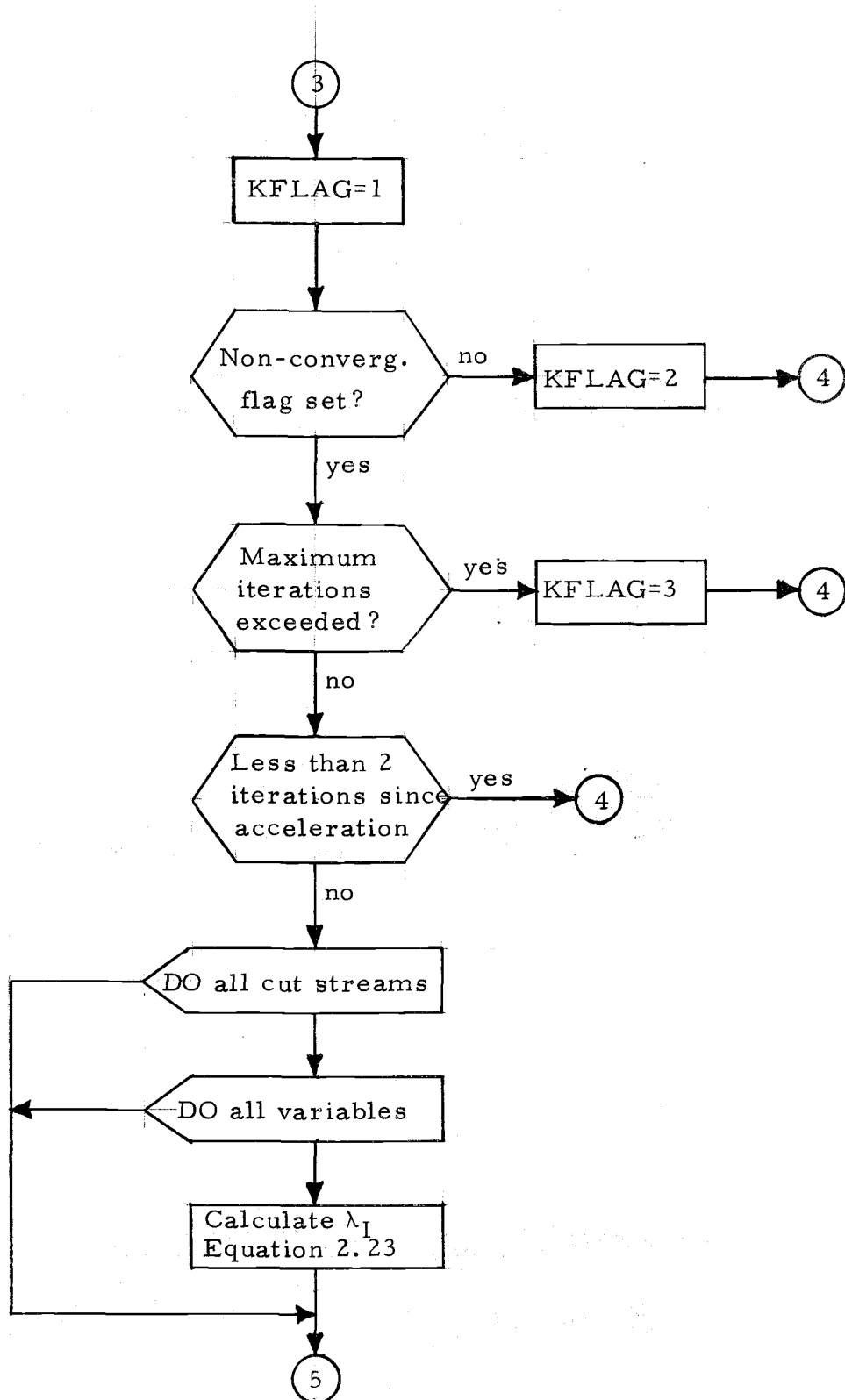


Figure 11. Continued.

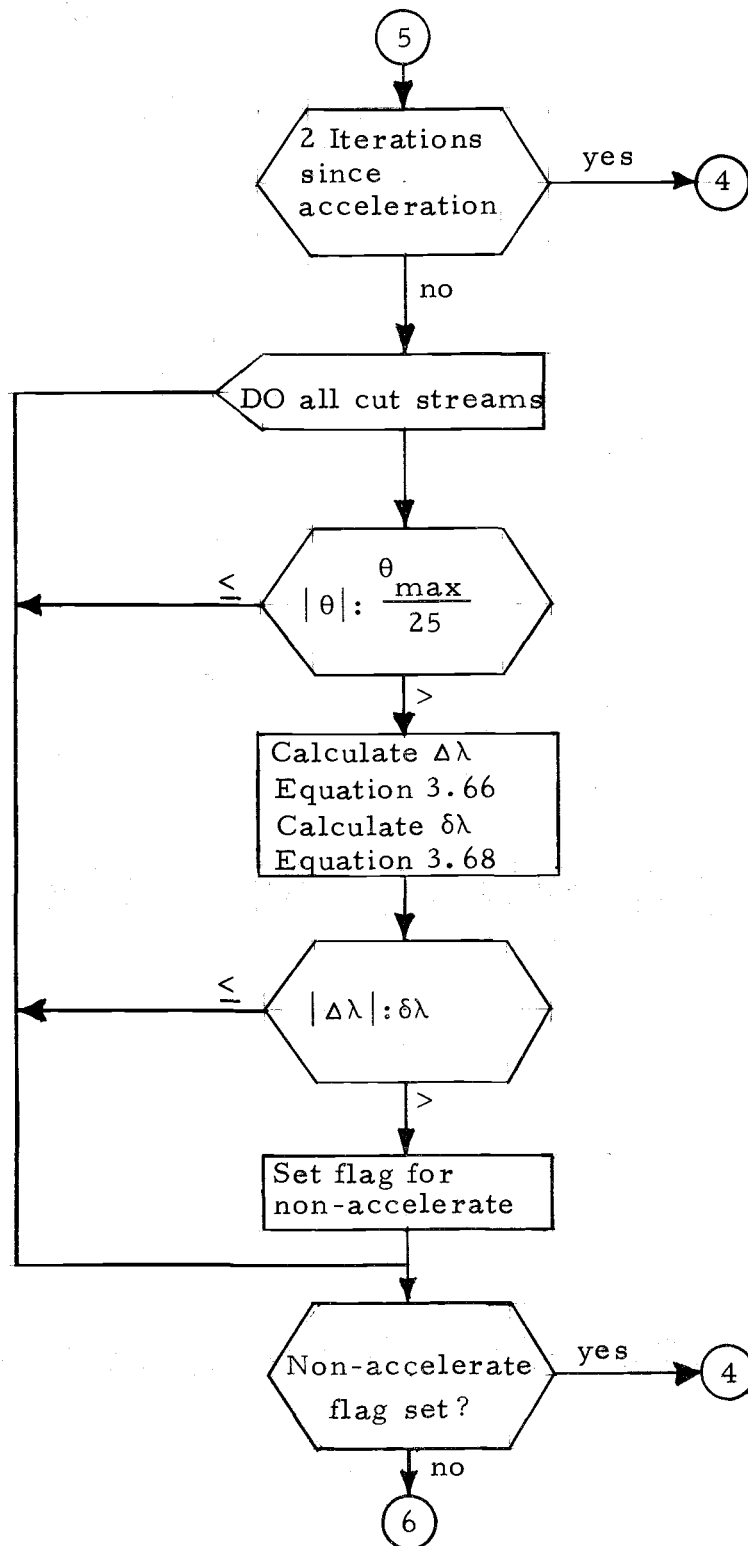


Figure 11. Continued.

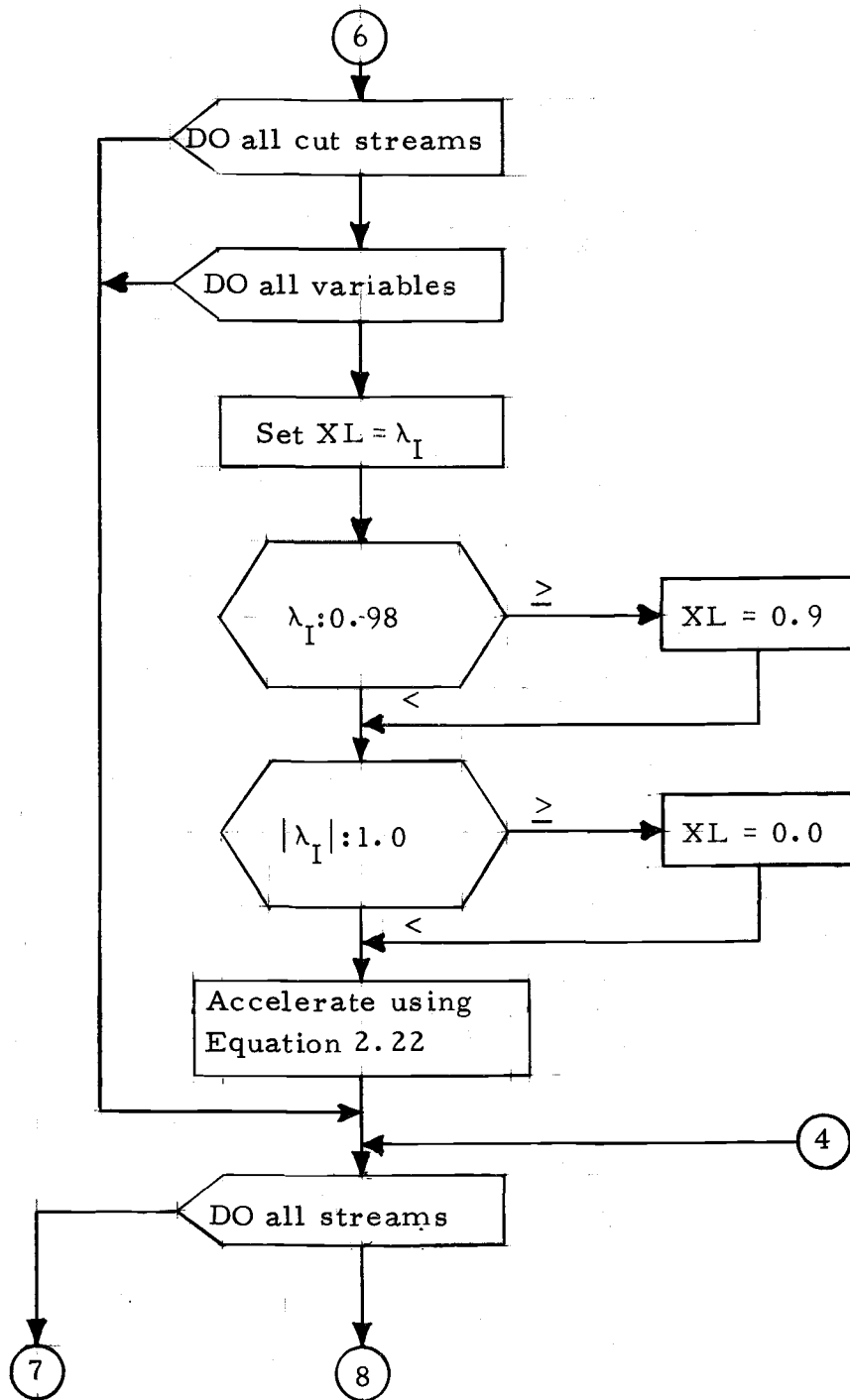


Figure 11. Continued.

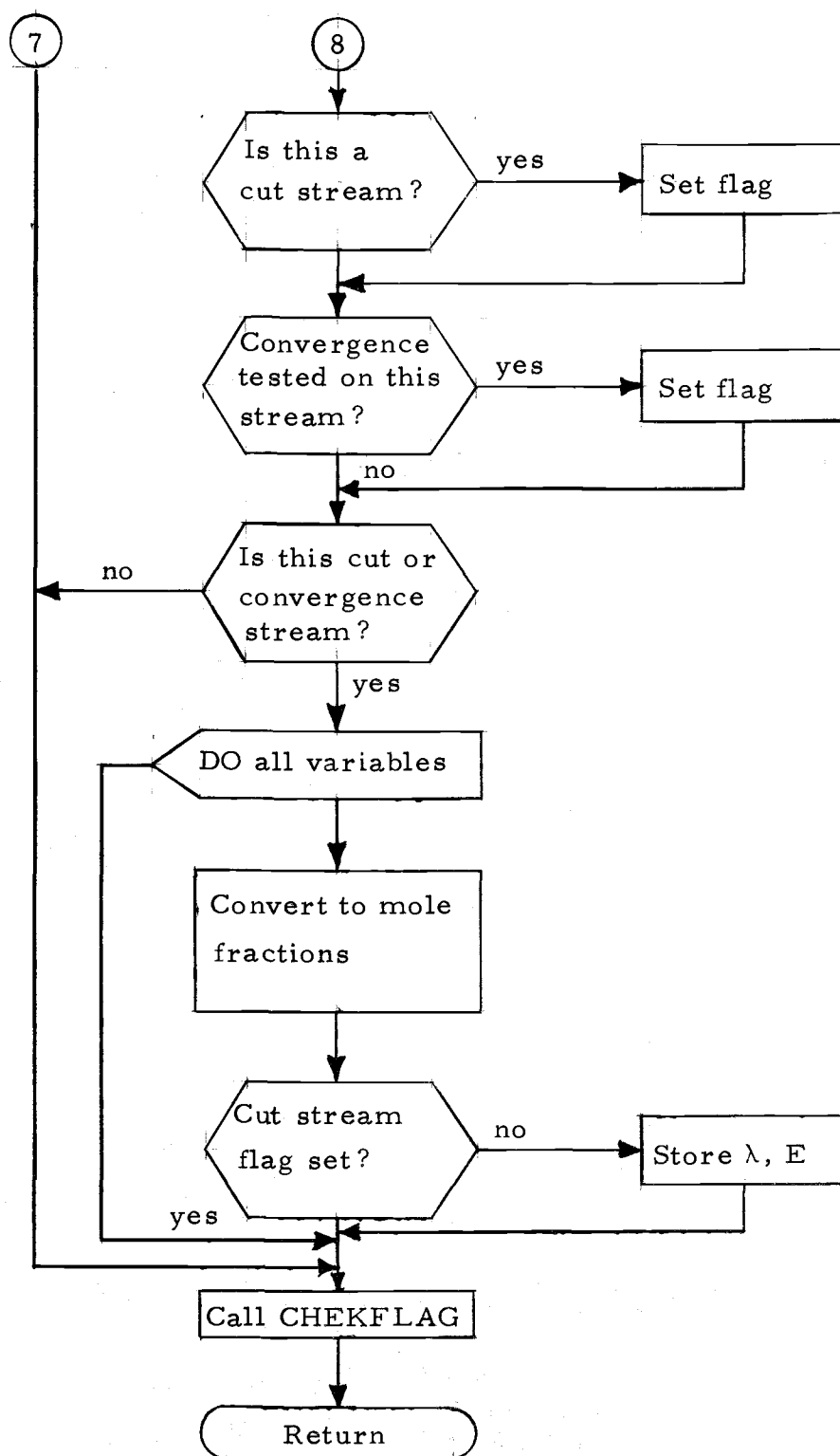


Figure 11. Continued.

test vector of each variable in the recycle (iterative) streams was less than .0001. The recycle streams only were used to test convergence in order that the evaluations be compatible with and directly comparable to the work of Kwon (15). The correction factor, ϕ , was not used in testing convergence for the same reason. In either case, there would be no significant effect on this work since the interest is on differences between methods of solution rather than the actual solutions.

The results of the runs of the first four systems are shown in Table 5. Results from Kwon (15) are also included for comparison. ASYM consistently converged the systems in fewer iterations than the other methods listed.

It is noted that RAM(2) converged the salt-water mixing example in six iterations, much better than any of the other methods. This example has only two variables, and so has only two eigenvalues. Furthermore, it is linear so that RAM(2) is exact and finds the solution on the first acceleration step. This will be the case with any initial guess when RAM(2) is used on this example.

The results of the next six examples are given in Table 6. These examples are in general more complex than the previous set.

The Bayer hydrate wash section example is one of those given by Orbach (22) in demonstration of DEM. It is interesting to note that all of the asymptotic methods tested were close in effectiveness on this

Table 5. Iterations to converge simple example systems.

System	Initial Guess	SUCCESS	WGSTEIN	WGSTEINB	WGSTEINC	RAM(1)	RAM(2)	NEWTON	GFP	DEM	ASYM
Photochemical Reaction	1	10	22	21	10	11	11	21	17	10	9
	2	9	--	14	11	12	13	--	--	9	9
Multi-Nodal	3	10	--	15	9	11	11	--	--	10	9
Salt-Water Mixing	1	46	30	27	21	14	6	11	50+	23	10
	2	42	--	20	16	18	6	--	--	17	12
	3	59	--	26	21	24	6	--	--	43	14
Photochemical Reaction	1	80	58	32	26	30	29	401+	Diverge	55	46
	2	78	--	30	29	28	35	--	--	53	25
Single Node	3	79	--	32	22	29	30	--	--	54	25
Nonlinear Equation Set	1	31	Diverge	Diverge	Diverge	16	19	Diverge	Diverge	19	17
	2	30	--	Diverge	13	20	16	17	--	16	13
	3	29	20	10	13	15	18	--	--	19	15

WGSTEIN - Wegstein's Method (32).

WGSTEINB - Bounded Wegstein (14).

WGSTEINC - Bounded Wegstein (18).

NEWTON - Newton-Raphson Method (15).

GFP - Generalized False Position (27).

Table 6. Iterations to converge complex example systems.

System	Initial Guess	SUCCESS	RAM(1)	RAM(2)	DEM	ASYM
Nagiev's Example	1	27	27	17	17	19
	2	24	24	15	21	16
Hypothetical Reaction System	1	47	27	13	24	16
	2	20	19	16	16	18
	3	51	24	27	30	19
Bayer Hydrate Wash Section	1	42	33	24	31	23
	2	56	31	24	34	26
Phosphoric Acid Plant	1	56	40	34	15	10
	2	73	40	34	15	15
	3	54	30	26	20	10
Cavett Problem 3 Cut Streams	1	88	52	51	39	35
	2	63	16	20	19	15
	3	69	36	30	34	25
Cavett Problem 2 Cut Streams	1	97	62	62	58	49
	2	54	65	58	42	51
	3	115	50	72	50	51

example, with both ASYM and RAM(2) performing better than DEM.

The final example is the Urea process simulation by Yazhari (33). This simulation was prepared for optimization studies of the Reichhold urea plant in St. Helens, Oregon. Two cases were examined, and their results are shown in Table 7. Case I was converged to a tolerance of .001. Neither ASYM nor DEM took an acceleration step in this case and so converged in the same number of iterations as successive substitutions. RAM(1), however, required twice as many iterations to converge because of poor acceleration steps early in the iteration. This shows an additional advantage of the use of a criterion for acceleration. It indicates when acceleration should definitely not be attempted. The urea simulation is quite involved and is being developed further. It requires roughly six to eight seconds of CPU time per iteration on the Oregon State University CDC 3300 computer. A savings of ten iterations is then approximately a minute of time saved, or five dollars per run.

The second case of the urea system indicated another process idiosyncrasy which can cause problems with convergence calculations. The second initial guess used appeared to get into an infinite oscillation after 12 iterations when solved by successive substitutions, ASYM, and DEM. The reason for this behavior is that several of the unit subroutines in the simulation utilized the interval halving method to solve internal iterations. This caused the system to stop behaving as

a continuous function when the calculations neared convergence. The outputs of the subroutines could thus take only discrete values, and in this case these discrete values caused the oscillation. In order to obtain a continuous type behavior it was necessary to change the tolerance on the internal iterations to a value much lower than that on the system iteration. When this was done the convergence indicated was obtained.

Table 7. Iterations to converge urea system.

Case	Initial Guess	Iterations to Converge				
		RAM(1)	RAM(2)	Success	DEM	ASYM
I	1	23	9	11	11	11
II	1	24	17	17	17	17
II	2	--	17	+	+	+
II	2#	--	15	12	12	12
II	3	--	*	19	13	11

* Bad acceleration.

+ Oscillating as explained in text.

Rerun of initial guess 2 explained in text.

This effect can be seen more clearly by examining the apparent eigenvalue used by DEM for its acceleration criteria. This is calculated from:

$$\lambda_D = \frac{\|\theta_I\|}{\|\theta_{I-1}\|} \quad 3.69$$

The values for the two runs are given in Table 8. As can be seen the apparent eigenvalue fluctuates for the original system, but smoothly

approaches a constant value for the system with the tightened criteria on the internal iterations.

Table 8. Apparent system eigenvalues of urea system.

Iteration	λ_D Defined by Orbach (22) Equation 3.69	
	Original System	System with Tight Internal Criteria
3	.156	.155
4	.325	.319
5	.812	.830
6	.633	.651
7	.647	.632
8	.553	.616
9	.769	.622
10	.508	.622
11	.646	--

The final runs of the urea process utilizing a different initial guess and the tightened criteria resulted in convergence acceleration being applied successfully by both ASYM and DEM. In this case RAM(2) took a very bad step early in the iteration and calculation was halted.

4. CUT SET SELECTION

Initial Guesses

One of the primary needs in the comparison of various cut sets is to determine how the system characteristics affect convergence for each alternative cut set. In order to isolate these effects it is necessary to start calculation with initial guesses that are equivalent so that they affect each cut set the same.

Previous work (7) has utilized the fractional deviation of the initial values from the solutions as the basis for equivalency between cut sets. In particular, the use of zero values for initial guesses were said to give equivalent starting values since all variables would start 100% away from their solutions. This is not the desired choice as can be seen from a Taylor series expansion of the general iteration Equation (2.2) about its solution.

$$\bar{x}_{1,I} = \bar{F}(\bar{x}_{1,S}) + \frac{\partial \bar{F}}{\partial \bar{x}_1} \Big|_{\bar{x}_S} (\bar{x}_{1,I-1} - \bar{x}_{1,S}) + \frac{1}{2} \frac{\partial^2 \bar{F}}{\partial x_1^2} \Big|_{\bar{x}_{1,S}} [(\bar{x}_{1,I-1} - \bar{x}_{1,S})^2] + \dots \quad 4.1$$

Noting that $\bar{F}(\bar{x}_{1,S}) = \bar{x}_{1,S}$ and expanding for a two component stream gives:

$$\begin{aligned}
\begin{bmatrix} x_{11} - x_{11s} \\ x_{12} - x_{12s} \end{bmatrix}_I &= \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}_s \begin{bmatrix} x_{11} - x_{11s} \\ x_{12} - x_{12s} \end{bmatrix}_{I-1} \\
&+ \frac{1}{2} \frac{\partial}{\partial x_1} \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}_s \begin{bmatrix} (x_{11} - x_{11s})^2 \\ (x_{12} - x_{12s})^2 \end{bmatrix}_{I-1} \\
&+ \dots
\end{aligned} \tag{4.2}$$

The Jacobian and its derivatives represent the system characteristics which will affect the convergence of the system and will differ for various cut streams. If Equation 4.2 is rewritten for a different stream in the recycle system, the following is obtained:

$$\begin{bmatrix} x_{21} - x_{21s} \\ x_{22} - x_{22s} \end{bmatrix}_I = \begin{bmatrix} J'_{11} & J'_{12} \\ J'_{21} & J'_{22} \end{bmatrix}_s \begin{bmatrix} x_{21} - x_{21s} \\ x_{22} - x_{22s} \end{bmatrix}_{I-1} + \dots \tag{4.3}$$

These equations relate the absolute deviation from the solution at a given iteration to the absolute deviation at the previous iteration via the system function's derivatives. Obviously, if initial values for streams one and two are chosen so that they have the same absolute deviations from the solution, subsequent iterative deviations will depend only upon the system characteristics.

If, on the other hand, the fractional deviations are chosen as initially equivalent:

$$[\bar{I}x_{1s}]^{-1} (\bar{x}_1 - \bar{x}_{1s})_{I=0} = [\bar{I}x_{2s}]^{-1} (\bar{x}_2 - \bar{x}_{2s})_{I=0} \tag{4.4}$$

then the iterative equation for stream two becomes:

$$\begin{bmatrix} x_{21} - x_{21s} \\ x_{22} - x_{22s} \end{bmatrix}_I = \begin{bmatrix} J'_{11} & J'_{12} \\ J'_{21} & J'_{22} \end{bmatrix}_s \begin{bmatrix} \frac{x_{21s}}{x_{11s}} & 0 \\ 0 & \frac{x_{22s}}{x_{12s}} \end{bmatrix} \begin{bmatrix} x_{11} - x_{11s} \\ x_{12} - x_{12s} \end{bmatrix}_{I-1} + \dots$$

4.5

The subsequent iterative values of stream two will then depend not only upon the system characteristics, but also upon the relative magnitudes of the variables of the streams. This difference from the use of absolute deviations can change the effect of the system characteristics so that different behavior may be observed than with the former case. It may be argued that this extra factor is in itself a system characteristic since it involves solution values. The question is then one of a definition of terminology which has never been clarified. The definition of system characteristics as put forth in this work would require the use of equivalent absolute deviations from the solution as the starting point in the comparison of cut sets.

Another important consideration, however, is how a cut set selection algorithm is to be utilized. If it is assumed that some cut set exists for starting a calculation and that the method will evaluate alternative sets during calculation, the choice of initial values for comparison purposes becomes irrelevant. The question in this case becomes: Given an initial cut set, will it be advantageous to switch to another set to promote convergence after calculations have been

initiated? This is the approach considered in this work.

When comparative calculations were made for this work, the procedure was to start calculation and perform one iteration with some initial value and cut set. The values from this iteration were then used as initial guesses for the cut sets to be compared.

Sensitivity Matrices

Given a process unit with input vector \bar{x} and output vector \bar{y} such that:

$$\bar{y} = \bar{f}(\bar{x}) \quad 4.6$$

the sensitivity matrix of the unit will be defined as the derivative of the functions \bar{f} with respect to the input vector evaluated at the solution. Thus:

$$S = \left. \frac{\partial \bar{f}}{\partial \bar{x}} \right|_{\bar{x}_s} \quad 4.7$$

For a linearized unit the sensitivity matrix is a set of constants.

The sensitivity matrices of the units in a recycle net combine to form the Jacobian for the net. Consider the recycle system of Figure 12. If \bar{x}_1 is chosen as the iterative stream the new value of \bar{x}_1 will be calculated by serially advancing through the units:

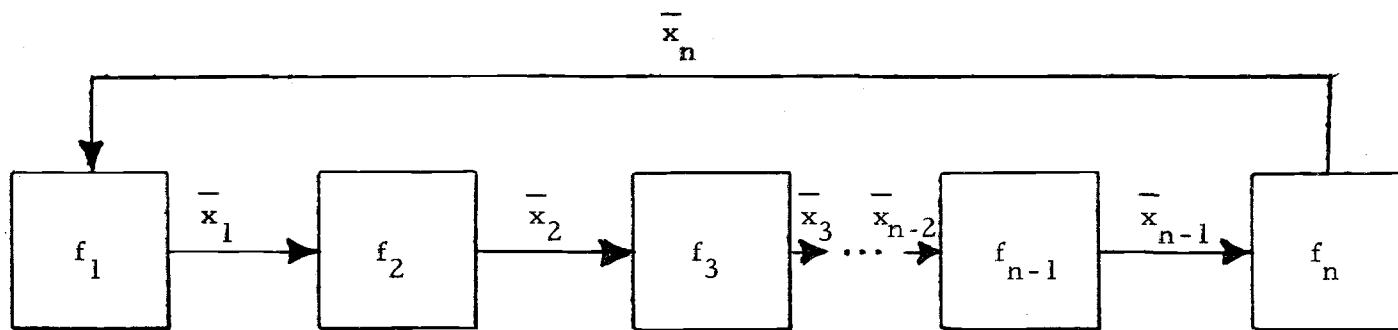


Figure 12. Single recycle loop system.

$$\begin{aligned}
\bar{x}_2 &= \bar{f}_2(\bar{x}_{1\text{guess}}) \\
\bar{x}_3 &= \bar{f}_3(\bar{x}_2) \\
&\vdots \\
\bar{x}_n &= \bar{f}_n(\bar{x}_{n-1}) \\
\bar{x}_{1\text{new}} &= \bar{f}_1(\bar{x}_n)
\end{aligned}
\tag{4.8}$$

Or:

$$\bar{x}_{1\text{new}} = \bar{f}_1(\bar{f}_n(\bar{f}_{n-1}(\dots\bar{f}_2(\bar{x}_{1\text{guess}})\dots)))
\tag{4.9}$$

$$= \bar{F}_1(\bar{x}_{1\text{guess}})
\tag{4.10}$$

Thus by the chain rule,

$$J = \frac{\partial \bar{F}_1}{\partial \bar{x}_1} = \frac{\partial \bar{f}_1}{\partial \bar{x}_n} \cdot \frac{\partial \bar{f}_n}{\partial \bar{x}_{n-1}} \dots \frac{\partial \bar{f}_3}{\partial \bar{x}_2} \cdot \frac{\partial \bar{f}_2}{\partial \bar{x}_1}
\tag{4.11}$$

$$= S_1 \prod_{i=0}^{n-2} S_{n-i}$$

Therefore, the Jacobian of the iterative net is the product of the sensitivity matrices of the individual units of the net taken in the reverse of the order of calculation.

The sensitivity matrix gives a linearized transfer matrix for a unit and can be used as a linear model of a unit. When a linear model is used, the system evaluation may be considered as a series of transformations acting on a vector. This is the condition which is approached as calculations converge to the solution. The concept is

useful in describing the behavior of multiple loop systems.

Single Recycle Loops

Previous investigators (7, 22) have assumed that the convergence of a single loop will be the same regardless of which stream in the loop is chosen as the cut stream. Genna and Motard (7) incorrectly state that the Jacobians of the one cycle process are the same for all cut streams since they are the product of the sensitivity matrices of the various process units in "arbitrary order." Since matrix multiplication is not commutative, the Jacobians of the various cut streams in a single loop are not the same.

What is true is that the eigenvalues of the Jacobians of the various cut streams in the single loop are the same. Consider again the loop of Figure 12. If stream one is cut the Jacobian will be:

$$J_1 = S_1 \cdot S_n \dots S_3 \cdot S_2 \quad 4.12$$

For stream 2 as the cut stream:

$$J_2 = S_2 \cdot S_1 \cdot S_n \dots S_4 \cdot S_3 \quad 4.13$$

Now define the partial product of sensitivity matrices:

$$P_1 = S_1 \cdot S_n \dots S_3 \quad 4.14$$

so that

$$J_1 = P_1 \cdot S_2 \quad 4.15$$

and

$$J_2 = S_2 \cdot P_1 \quad 4.16$$

The eigenvalues of J_1 and J_2 are the same by the following result from Roy (26):

"Every non-zero characteristic root of $A(p \times q) \cdot B(q \times p)$ is a non-zero characteristic root of $B(q \times p) \cdot A(p \times q)$ and vice versa."

Similarly, it can be seen for any pair of streams i and j where $i < j$ that:

$$J_i = S_i S_{i-1} \dots S_2 S_1 S_n S_{n-1} \dots S_{j+1} S_j S_{j-1} \dots S_{i+1}$$

and

$$J_j = S_j S_{j-1} \dots S_{i+1} S_i S_{i-1} \dots S_2 S_1 S_n S_{n-1} \dots S_{j+1} \quad 4.17$$

Or,

$$J_i = P \cdot Q \quad 4.18$$

$$J_j = Q \cdot P$$

$$P = \prod_{k=0}^{i-1} S_{i-k} \prod_{k=0}^{n-(j+1)} S_{n-k} \quad 4.19$$

$$Q = \prod_{k=0}^{j-(i+1)} S_{j-k}$$

Thus the eigenvalues of J_i and J_j are the same. Therefore the eigenvalues of the Jacobians of any possible cut stream in the single

loop are the same. Ultimately, the rates of convergence of all streams in the single loop will be the same since the rates will be determined by the same dominant eigenvalue. Any differences in convergence will be due to slight differences in variable interaction caused by the differences in the Jacobians. These differences will not be detectable if the iteration is started closely enough to the solution so that the system will exhibit linear behavior.

Multiple Recycle Loops

Partitioned Sensitivity Matrices

The problem of cut set selection for multiple loop systems can be considered in a manner similar to the single loop case. The general iterative equation (2.2) still holds but the iterative vector is now composed of several vectors and may be partitioned for an n loop system as:

$$\bar{x} = \begin{bmatrix} \bar{x}_1 \\ - \\ \bar{x}_2 \\ \vdots \\ - \\ \bar{x}_n \end{bmatrix}_{I+1} = \bar{F} \left\{ \begin{bmatrix} \bar{x}_1 \\ - \\ \bar{x}_2 \\ - \\ \bar{x}_n \end{bmatrix} \right\}_I \quad 4.20$$

For example, the choice of streams $\{3, 8, 9\}$ as the cut set for the system of Figure 13 would give the iterative relationship:

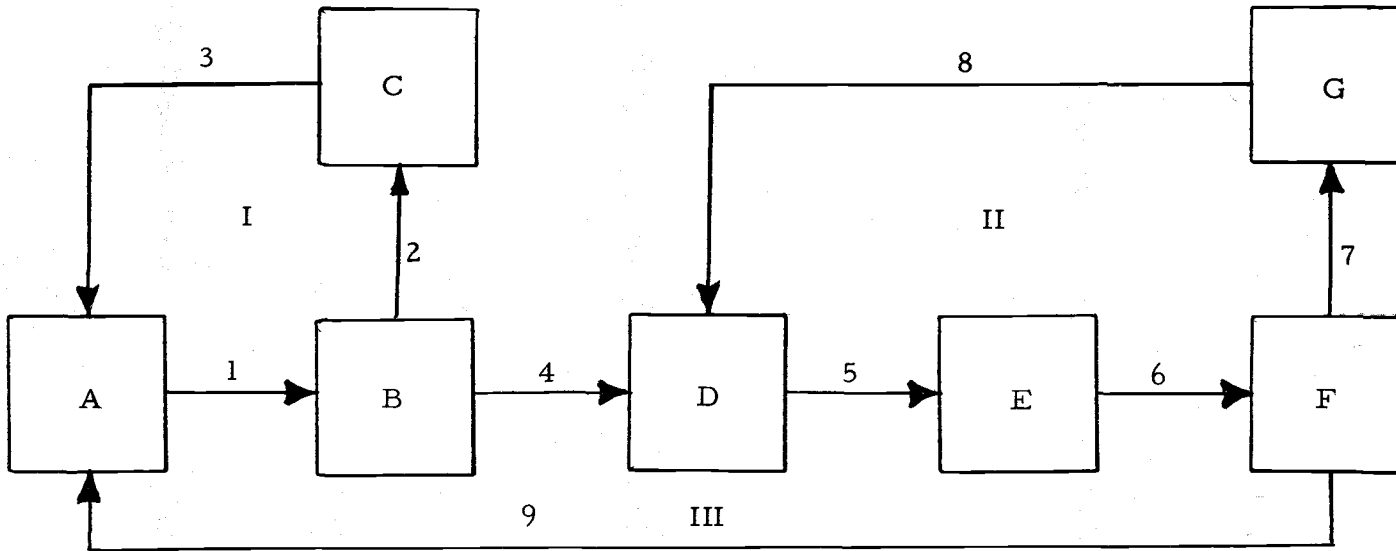


Figure 13. Multiple recycle loop system.

$$\begin{bmatrix} \bar{x}_3 \\ \bar{x}_8 \\ \bar{x}_9 \end{bmatrix}_{I+1} = \bar{F} \left\{ \begin{bmatrix} \bar{x}_3 \\ \bar{x}_8 \\ \bar{x}_9 \end{bmatrix} \right\}_I = \begin{bmatrix} \bar{F}_3 \\ \bar{F}_8 \\ \bar{F}_9 \end{bmatrix}_I \quad 4.21$$

where \bar{x}_3 cuts loop I, \bar{x}_8 cuts loop II, and \bar{x}_9 cuts loop III.

For the three loop problem, it is seen that two streams are sufficient to cut the system since some streams are members of more than one loop and can thus cut more than one loop. Another possible iterative vector for the cut set $\{3, 5\}$ is thus

$$\begin{bmatrix} \bar{x}_3 \\ \bar{x}_5 \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{x}_3 \\ \bar{x}_5 \end{bmatrix} \quad 4.22$$

here \bar{x}_3 cuts loop I and \bar{x}_5 cuts loops II and III.

Analogous to the single loop problem, the Jacobian for a particular cut set is still the product of the sensitivity matrices of the units in reverse order of calculation. Consider the cut set $\{3, 8, 9\}$ for the process in Figure 13 with the order of calculation A, B, C, D, E, F, G. The iterative equation is then:

$$\bar{x} = \begin{bmatrix} \bar{x}_3 \\ \bar{x}_8 \\ \bar{x}_9 \end{bmatrix} = \bar{F}(\bar{x}) = \bar{f}_G(\bar{f}_F(\bar{f}_E(\bar{f}_D(\bar{f}_C(\bar{f}_B(\bar{f}_A) \dots))) \quad 4.23$$

So that the Jacobian is:

$$J = \frac{\partial \bar{F}}{\partial \bar{x}} = S'_G S'_F S'_E S'_D S'_C S'_B S'_A \quad 4.24$$

where S' is used to denote the sensitivity matrices for the multiple loop problem. The expression for S' will be developed further.

Note now that Equation 4.21 can be further broken down as:

$$\frac{\partial \bar{F}(\bar{x})}{\partial \bar{x}} = \begin{bmatrix} \frac{\partial \bar{F}_3}{\partial \bar{x}_3} & \frac{\partial \bar{F}_3}{\partial \bar{x}_8} & \frac{\partial \bar{F}_3}{\partial \bar{x}_9} \\ \frac{\partial \bar{F}_8}{\partial \bar{x}_3} & \frac{\partial \bar{F}_8}{\partial \bar{x}_8} & \frac{\partial \bar{F}_8}{\partial \bar{x}_9} \\ \frac{\partial \bar{F}_9}{\partial \bar{x}_3} & \frac{\partial \bar{F}_9}{\partial \bar{x}_8} & \frac{\partial \bar{F}_9}{\partial \bar{x}_9} \end{bmatrix} \quad 4.25$$

The Jacobian is thus seen to be a partitioned matrix.

It is now necessary to determine the form of the sensitivity matrices, S' , for the multiple loop problem. Since a particular unit will still be represented by a set of functions as in Equation 4.6, the sensitivity matrix will again be:

$$S' = \left. \frac{\partial f}{\partial \bar{x}} \right|_{\bar{x}_s} \quad 4.26$$

Now, however, the output vector has been partitioned so that 4.6 becomes:

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_n \end{bmatrix} = \begin{bmatrix} \bar{f}_1(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \\ \bar{f}_2(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \\ \vdots \\ \bar{f}_n(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \end{bmatrix} \quad 4.27$$

Thus the sensitivity matrix becomes:

$$S' = \begin{bmatrix} \frac{\partial \bar{f}_1}{\partial \bar{x}_1} & \frac{\partial \bar{f}_1}{\partial \bar{x}_2} & \dots & \frac{\partial \bar{f}_1}{\partial \bar{x}_n} \\ \frac{\partial \bar{f}_2}{\partial \bar{x}_1} & \frac{\partial \bar{f}_2}{\partial \bar{x}_2} & \dots & \frac{\partial \bar{f}_2}{\partial \bar{x}_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \bar{f}_n}{\partial \bar{x}_1} & \frac{\partial \bar{f}_n}{\partial \bar{x}_2} & \dots & \frac{\partial \bar{f}_n}{\partial \bar{x}_n} \end{bmatrix} \quad 4.28$$

Before examining the individual coefficient matrices it is necessary to recognize two concepts which are not immediately obvious. The first of these concepts is that the streams or vectors which link the units in a given loop can be considered as a single stream or vector which undergoes transformation by the units in the loop. Thus the nomenclature \bar{x}_I can be used to refer to the iterative vector in loop I of Figure 13. At a given stage of the calculation \bar{x}_I may be represented by \bar{x}_3 , \bar{x}_1 , or \bar{x}_2 . Similarly, loops II and III may be considered to have iterative vectors \bar{x}_{II} and \bar{x}_{III} .

which can be represented by $\bar{x}_5, \bar{x}_6, \bar{x}_7,$ or \bar{x}_8 and by $\bar{x}_1, \bar{x}_4, \bar{x}_5, \bar{x}_6,$ or \bar{x}_9 respectively. Thus, Equation 4.27 can be written as:

$$\begin{bmatrix} \bar{x}_I \\ \bar{x}_{II} \\ \bar{x}_{III} \end{bmatrix} = \begin{bmatrix} \bar{f}_I(\bar{x}_I, \bar{x}_{II}, \bar{x}_{III}) \\ \bar{f}_{II}(\bar{x}_I, \bar{x}_{II}, \bar{x}_{III}) \\ \bar{f}_{III}(\bar{x}_I, \bar{x}_{II}, \bar{x}_{III}) \end{bmatrix}$$

for some unit U. Here the \bar{f} 's relate the input and output vectors for the process units. They may also be regarded as transformations acting upon the vector \bar{x} .

The second concept deals with the transformation of vectors which are seemingly unaffected by the calculation of some of the units. More specifically, a unit which operates only on one iterative vector appears to do nothing to the other iterative vectors. It is implicit that the other vectors have been transformed by the identity matrix. Therefore, the functional relationship relating the input vectors to the output vectors of a unit which performs no calculations which alter those vectors is simply multiplication by the identity matrix. Thus when unit C of Figure 13 is calculated, the iterative variable of loop I is transformed by:

$$\bar{x}_I = \bar{f}_{IC}(\bar{x}_I) \quad 4.29$$

and the iterative vectors of loops II and III are transformed by:

$$\bar{x}_{II} = I \bar{x}_{II} \quad 4.30$$

$$\bar{x}_{III} = I \bar{x}_{III} \quad 4.31$$

The expression for the partitioned sensitivity matrix for unit C can be seen to follow from Equation 4.28 as:

$$S'_C = \begin{bmatrix} \frac{\partial \bar{f}_I}{\partial \bar{x}_I} & \frac{\partial \bar{f}_I}{\partial \bar{x}_{II}} & \frac{\partial \bar{f}_I}{\partial \bar{x}_{III}} \\ \frac{\partial \bar{f}_{II}}{\partial \bar{x}_I} & \frac{\partial \bar{f}_{II}}{\partial \bar{x}_{II}} & \frac{\partial \bar{f}_{II}}{\partial \bar{x}_{III}} \\ \frac{\partial \bar{f}_{III}}{\partial \bar{x}_I} & \frac{\partial \bar{f}_{III}}{\partial \bar{x}_{II}} & \frac{\partial \bar{f}_{III}}{\partial \bar{x}_{III}} \end{bmatrix} C \quad 4.32$$

From Equation 4.29 and previous definitions, it follows that:

$$\left. \frac{\partial \bar{f}_I}{\partial \bar{x}_I} \right|_C = \frac{\partial \bar{f}_C}{\partial \bar{x}_2} = \frac{\partial \bar{x}_3}{\partial \bar{x}_2} = S_C$$

and from 4.30 and 4.31:

$$\frac{\partial \bar{f}_{II}}{\partial \bar{x}_{II}} = I \quad 4.33$$

$$\frac{\partial \bar{f}_{III}}{\partial \bar{x}_{III}} = I$$

Since all of the off diagonal terms of Equation 4.32 are equal to the zero matrix, it follows that:

$$S'_C = \begin{bmatrix} S_C & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \quad 4.34$$

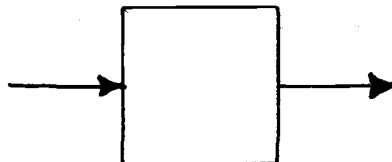
The completely general form of the partitioned sensitivity matrix for a unit in an n loop system is the extension of Equation 4.32:

$$S' = \begin{bmatrix} \frac{\partial \bar{f}_I}{\partial \bar{x}_I} & \frac{\partial \bar{f}_I}{\partial \bar{x}_{II}} & \cdots & \frac{\partial \bar{f}_I}{\partial \bar{x}_n} \\ \frac{\partial \bar{f}_{II}}{\partial \bar{x}_I} & \frac{\partial \bar{f}_{II}}{\partial \bar{x}_{II}} & \cdots & \frac{\partial \bar{f}_{II}}{\partial \bar{x}_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \bar{f}_n}{\partial \bar{x}_I} & \frac{\partial \bar{f}_n}{\partial \bar{x}_{II}} & \cdots & \frac{\partial \bar{f}_n}{\partial \bar{x}_n} \end{bmatrix} \quad 4.35$$

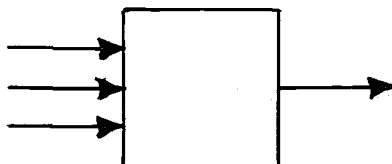
This is the same form as Equation 4.28 except for the revised concept of the iterative vector being identified with the entire loop.

The simple form of Equation 4.34 leads to the impression that the general form can be reduced to a similar simple form. This is indeed the case. It can be observed that all units in any process system can be characterized in terms of four basic transformations on iterative vectors. These are illustrated schematically in Figure 14

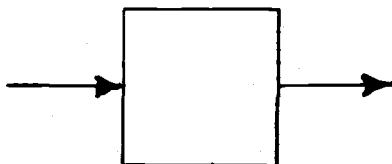
Type I. Member of one loop.



Type II. Combines two or more loops.



Type III. Member of two or more loops.



Type IV. Splits one stream into two or more.

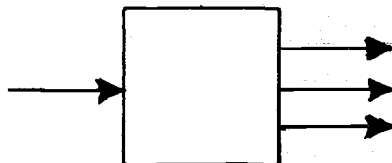


Figure 14. Types of process units.

and are shown as they occur in a system in Figure 13. The types are:

- I. A unit which performs calculations on the iterative vector of only one loop (Units C and G).
- II. A unit which combines the iterative vectors from two or more loops into a single iterative vector for these loops (Units A and D).
- III. A unit which performs calculations on one vector which represents the iterative vectors for two or more loops (Unit E).
- IV. A unit which splits a vector representing two or more loops into two or more separate vectors (Units B and F).

The type I unit has already been discussed and an example given of the form of its sensitivity matrix (Equation 4.34). In general this type of unit acts upon only one of the iterative vectors. All other iterative vectors are transformed by the identity matrix. Consider some unit, U , in loop K . Here U and K are used as general indices. The transformation for the iterative vector, \bar{x}_K , of loop K will be:

$$\bar{x}_K = \bar{f}_{KU}(\bar{x}_K) \quad 4.36$$

All other transformations will be:

$$\bar{x}_L = I \bar{x}_L \quad \begin{array}{l} L = I, II, \dots \\ L \neq K \end{array} \quad 4.37$$

Thus from Equation 4.35,

$$S'_{\text{typeI}} = \begin{bmatrix} I & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & I & & & & \cdot \\ \vdots & & \ddots & & & \cdot \\ \vdots & & & S_{KU} & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & I \end{bmatrix} \quad 4.38$$

The sensitivity matrix will be a diagonal matrix with all terms equal to the identity matrix except one. That term, S_{KU} , can be in any position on the diagonal.

The type II unit will involve calculating one iterative vector as a function of two or more vectors,

$$\bar{x}_K = \bar{f}_{KU}(\bar{x}_K, \bar{x}_L, \dots) \quad 4.39$$

The remaining vectors which are being combined, \bar{x}_L , \bar{x}_M , etc. are transformed to zero vectors by a type II unit since they will be represented by the \bar{x}_K vector. That is, the vectors being represented by the \bar{x}_K vector carry a zero value until they are again split so that they are separate from the \bar{x}_K vector. The choice of which vector will represent the others is arbitrary. As before, the iterative vectors in any loops not being combined are merely transformed by the identity matrix.

Again utilizing Equation 4.35, the general form for the type II sensitivity matrix is:

$$S'_{\text{typeII}} = \begin{bmatrix} I & 0 & & & & & 0 \\ 0 & I & & & & & \\ & & S_K & & & & \\ & & & I & & & \\ & & & & S_L & & \\ & & & & & S_M & \\ & & & & & & I \\ & & & & & & & I \\ & & & & & & & & I \\ & & & & & & & & & 0 \\ & & & & & & & & & & I \\ & & & & & & & & & & & I \\ & & & & & & & & & & & & I \\ & & & & & & & & & & & & & I \\ 0 & & & & & & & & & & & & & & 0 \end{bmatrix} \quad 4.40$$

Here there will be one row with one or more terms off the diagonal. Each of these terms will be a unit sensitivity for a single stream. The terms may appear anywhere in the row. The diagonal terms corresponding to the columns in which these single stream sensitivities appear will be zero. All other diagonal terms will be the identity matrix. All other off diagonal terms will be zero.

Unit A of the example is a type II unit. The transformations involved are seen to be:

$$\begin{aligned} \bar{x}_I &= f_{AI}(\bar{x}_I, \bar{x}_{III}) \\ \bar{x}_{II} &= I \bar{x}_{II} \\ \bar{x}_{III} &= 0 \end{aligned} \quad 4.41$$

The sensitivity matrix is then

$$S'_A = \begin{bmatrix} S_{AI} & 0 & S_{AIII} \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad 4.42$$

A type III unit will carry out a transformation on only one vector. The other vectors will again be operated on by the identity matrix. This type of unit must be preceded by either another unit of the same type or a type II unit so that some of the iterative vectors will be zero vectors. For these vectors the transformation will be simply considered to be the multiplication by a zero matrix.

There will then be three forms of transformations for the type III unit:

$$\begin{aligned} \bar{x}_K &= f_{KU}(\bar{x}_K) \\ \bar{x}_L &= I \bar{x}_L \\ \bar{x}_M &= 0 \end{aligned} \quad \begin{array}{l} L \neq K \neq M \\ L, M = I, II, III \dots \end{array} \quad 4.43$$

The general form for the sensitivity matrix is then:

$$S'_{\text{typeIII}} = \begin{bmatrix} I & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & I & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & S_K & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & I & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & I & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & I \end{bmatrix} \quad 4.44$$

Again, all off diagonal terms are zero. One of the diagonal terms will be a single loop unit sensitivity matrix. There will be one or more zeroes in other positions on the diagonal. The remainder of the diagonal terms will be the identity matrix.

In the case of unit E, the transformations are:

$$\begin{aligned}\bar{x}_I &= I \bar{x}_I \\ \bar{x}_{II} &= S_E \bar{x}_{II} \\ \bar{x}_{III} &= 0\end{aligned}\tag{4.45}$$

Or,

$$S'_E = \begin{bmatrix} I & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix}\tag{4.46}$$

The type IV unit may carry out a number of transformations but will have at least two of the form:

$$\begin{aligned}\bar{x}_K &= \bar{f}_{KU}(\bar{x}_K) \\ \bar{x}_L &= \bar{f}_{LU}(\bar{x}_K)\end{aligned}\tag{4.47}$$

Again, all other transformations will be multiplication by the identity matrix. The general form of the sensitivity matrix is thus:

$$S'_{\text{type IV}} = \begin{bmatrix} I & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & I & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & S_K & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & I & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & I & \cdot \\ \cdot & \cdot & \cdot & S_L & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & I \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & I \end{bmatrix} \quad 4.48$$

In this type of sensitivity matrix there will be one column with an S matrix on the diagonal and one or more S matrices in other positions in that column. There will be zeroes on the diagonal in rows in which these S matrices occur. All other diagonal terms will be I , and all other off diagonal terms will be zero.

Unit B of Figure 13 is seen to be a type IV unit with the transformations:

$$\begin{aligned} \bar{x}_I &= f_{BI}(\bar{x}_I) \\ \bar{x}_{II} &= I \bar{x}_{II} \\ \bar{x}_{III} &= f_{BII}(\bar{x}_I) \end{aligned} \quad 4.49$$

so that,

$$S'_B = \begin{bmatrix} S_{BI} & 0 & 0 \\ 0 & I & 0 \\ S_{BIII} & 0 & 0 \end{bmatrix} \quad 4.50$$

It should be emphasized that any unit which does not appear to fit one of these four types will be a combination of two or more types and can be broken down into pseudo units which will be one of these types. This is consistent with Rosen's (25) technique for determining sensitivity coefficients. The four types of units described here are sufficient to classify all units which will be encountered in process systems.

Properties of Partitioned Sensitivity Matrices

Two important combinatorial properties of unit sensitivity matrices can now be derived. The first of these properties is:

None of the sensitivity matrices for a single loop commute with any other sensitivity matrix in that loop.

That is, for units A and B in a single loop,

$$S'_{A B} S'_B \neq S'_B S'_A$$

Or, in terms of the example system of Figure 13:

$$\begin{aligned} & \begin{bmatrix} S_{A3} & 0 & S_{A9} \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{B2} & 0 & 0 \\ 0 & I & 0 \\ S_{B4} & 0 & 0 \end{bmatrix} = \begin{bmatrix} S_{A3}S_{B2} + S_{A9}S_{B4} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \neq & \begin{bmatrix} S_{B2} & 0 & 0 \\ 0 & I & 0 \\ S_{B4} & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{A3} & 0 & S_{A9} \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} S_{B2}S_{A3} & 0 & S_{B2}S_{A9} \\ 0 & I & 0 \\ S_{B4}S_{A3} & 0 & S_{B4}S_{A9} \end{bmatrix} \end{aligned} \quad 4.51$$

This non-commuting property can be shown to hold for any pair of types of units which are in the same loop.

For units in the same loop, say loop j , the partitioned unit sensitivity matrices will have matrices other than the identity matrix in the j th row and/or column. Thus two type I units in loop j would both have sensitivity matrices of the form:

$$S'_A = \begin{bmatrix} 1 & & & & & & & \\ & 2 & & & & & & \\ & & \dots & & & & & \\ & & & j & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & n \end{bmatrix}, \quad S'_B = \begin{bmatrix} 1 & & & & & & & \\ & 2 & & & & & & \\ & & \dots & & & & & \\ & & & j & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & n \end{bmatrix} \quad 4.52$$

Obviously the product $S'_A S'_B$ would yield a diagonal partitioned matrix with the identity matrix representing all elements except for the j th one, which would be $S'_A \cdot S'_B$. The product $S'_B S'_A$ would yield $S'_B \cdot S'_A$ in the j th position of the diagonal. Thus the two sensitivity matrices would not necessarily commute. This holds for any permutation of any of the four types of units when they are in the same loop.

The second important property of the sensitivity matrices is:
The sensitivity matrices of any two units which are not in the same loop do commute.

For the example system of Figure 13:

$$S'_C S'_E = S'_E S'_C \quad 4.53$$

or

$$\begin{aligned} & \begin{bmatrix} S_C & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} S_C & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ & = \begin{bmatrix} I & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_C & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \begin{bmatrix} S_C & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad 4.54$$

Two type I units in different loops, say loops j and k , would have sensitivity matrices with diagonal elements equal to the identity matrix except for the j th and k th positions so:

$$S'_A = \begin{bmatrix} 1 & 2 & \dots & j & \dots & n \\ I & & & & & \\ & \dots & & & & \\ & & S_A & & & \\ & & & & & \\ & & & & I & \\ & & & & & \dots \\ & & & & & & I \end{bmatrix}, \quad S'_B = \begin{bmatrix} 1 & 2 & \dots & k & \dots & n \\ I & & & & & \\ & \dots & & & & \\ & & S_B & & & \\ & & & & & \\ & & & & & \\ & & & & I & \\ & & & & & \dots \\ & & & & & & I \end{bmatrix} \quad 4.55$$

In this case the product $S'_A S'_B$ would yield a diagonal matrix with all identity elements except the j th element would be $S_A \cdot I$ and the k th element $I \cdot S_B$. The product $S'_B S'_A$ would have $I \cdot S_A$ in the j th position and $S_B \cdot I$ in the k th position. Now since the identity matrix commutes with all other square matrices, the terms of both products would be the same so that $S'_A S'_B = S'_B S'_A$.

This property can be seen to hold for any permutation of types of units in the same loop. Any two units in the same loop will have an S matrix in the same position on the diagonal of their sensitivity matrices. Thus the product of the sensitivity matrices will involve at least one product of S matrices. The S matrices will not in general commute, therefore, the S' matrices will not generally commute.

Equivalent Cut Sets

The usefulness of these two properties lies in recognizing that the sensitivity matrices which make up the Jacobian of a given cut set can be manipulated to some extent without changing the Jacobian. This will allow the comparison of the eigenvalues of the Jacobians of different cut sets.

Recall for example the Jacobian defined earlier by Equation 4.24 for cut set $\{3, 8, 9\}$ of the example in Figure 13:

$$J_{389} = S'_G S'_F S'_E S'_D S'_C S'_B S'_A \quad 4.24$$

For comparative purposes, an order of calculation for set $\{3, 5\}$ of E, F, A, G, B, D, C would result in the Jacobian:

$$J_{35} = S'_C S'_D S'_B S'_G S'_A S'_F S'_E \quad 4.56$$

At first glance Equations 4.24 and 4.56 appear to have little in common other than the fact that they are products of the same matrices in seemingly arbitrary order. By the first property of sensitivity matrices, however, 4.56 can be rewritten since units C and D are not in the same loop.

$$J_{35} = S'_D S'_C S'_B S'_G S'_A S'_F S'_E \quad 4.57$$

And since units A and G are in different loops:

$$J_{35} = S'_D S'_C S'_B S'_A S'_G S'_F S'_E \quad 4.58$$

If the partial products are defined as

$$P = S'_D S'_C S'_B S'_A \quad 4.59$$

$$Q = S'_G S'_F S'_E$$

it is easy to see that Equations 4.24 and 4.58 result in:

$$J_{389} = Q \cdot P \quad 4.60$$

$$J_{35} = P \cdot Q$$

Thus by the result of Roy (26) described earlier, the eigenvalues of the cut sets {3, 8, 9} and {3, 5} will be exactly the same. Thus the rate of convergence of the cut sets will be the same.

A close examination of the two cut sets reveals one common factor which makes it simple to identify cut sets which will have the same eigenvalues. That is that the units of each loop are calculated in the same order except for starting at different places within the loops. This can be seen in Table 9.

Table 9. Computational order of Figure 13 for various cut sets.

Cut Set	Overall Order	Order of Individual Loops		
		Loop I	Loop II	Loop III
3, 8, 9	A, B, C, D, E, F, G	A, B, C	D, E, F, G	A, B, D, E, F,
3, 5	E, F, A, G, B, D, C	A, B, C	E, F, G, D	E, F, A, B, D,
1, 7	G, B, D, E, F, C, A	B, C, A	G, D, E, F	B, D, E, F, A
1, 5	E, F, G, B, D, C, A	B, C, A,	E, F, G, D	E, F, B, D, A

The additional example of cut set $\{1, 7\}$ has been included.

The order of calculation G, B, D, E, F, C, A gives the Jacobian:

$$J_{17} = S'_A S'_C S'_F S'_E S'_D S'_B S'_G \quad 4.61$$

This can be rearranged to give:

$$J_{17} = S'_A S'_F S'_E S'_D S'_C S'_B S'_G \quad 4.62$$

In this case a bit more manipulation is required than before to compare the cut sets. Define an intermediate product:

$$T = S'_F S'_E S'_D S'_C S'_B$$

so that

$$J_{17} = S'_A T S'_G$$

By Roy's (26) result, the eigenvalues of $S'_A T S'_G$ are the same as the eigenvalues of $S'_G S'_A T$. Since units A and G are in different loops,

$$S'_G S'_A T = S'_A S'_G T$$

As before, the eigenvalues of $S'_A S'_G T$ are the same as those of $S'_G T S'_A$. This is the same as J_{389} as defined by Equation 4.24.

Therefore the eigenvalues of J_{17} and J_{389} are the same.

In contrast, a different result is obtained with cut set $\{1, 5\}$,

the calculational order E, F, G, B, D, C, A, and Jacobian

$$J_{15} = S'_A S'_C S'_D S'_B S'_G S'_F S'_E \quad 4.63$$

In order that this cut set have the same eigenvalues as those previously examined it would be necessary to be able to change the order of the sensitivity matrices so that S'_B followed S'_A . This is not possible, however, since neither S'_D and S'_B nor S'_C and S'_B commute because they are in the same loop.

When different cut sets result in the same order of calculation of the units within each individual loop, all of the sensitivity matrices which do not commute will already be in the same order in the Jacobians of the various cut sets. The sensitivity matrices which do commute can then be rearranged and Roy's (26) result used to show that the eigenvalues of the various cut sets are the same. For sets in which the order of calculation within one of the loops varies, the order of some pair of non-commuting sensitivity matrices will be reversed in the Jacobians. In these cases, the eigenvalues will not be the same.

Non-Equivalent Cut Sets

Examination of Figure 13 when considering cut set {1, 5} reveals that loop III has been cut twice. Thus it becomes possible to

calculate the units in the loop in some sequence other than the natural order obtained by merely taking the units in order around the loop. All of the cut sets which had the same eigenvalues did follow the natural order. In order to obtain a different order of calculation and thus different eigenvalues it is necessary to cut a loop at least twice. Furthermore, this is the only way to obtain a cut set with different eigenvalues.

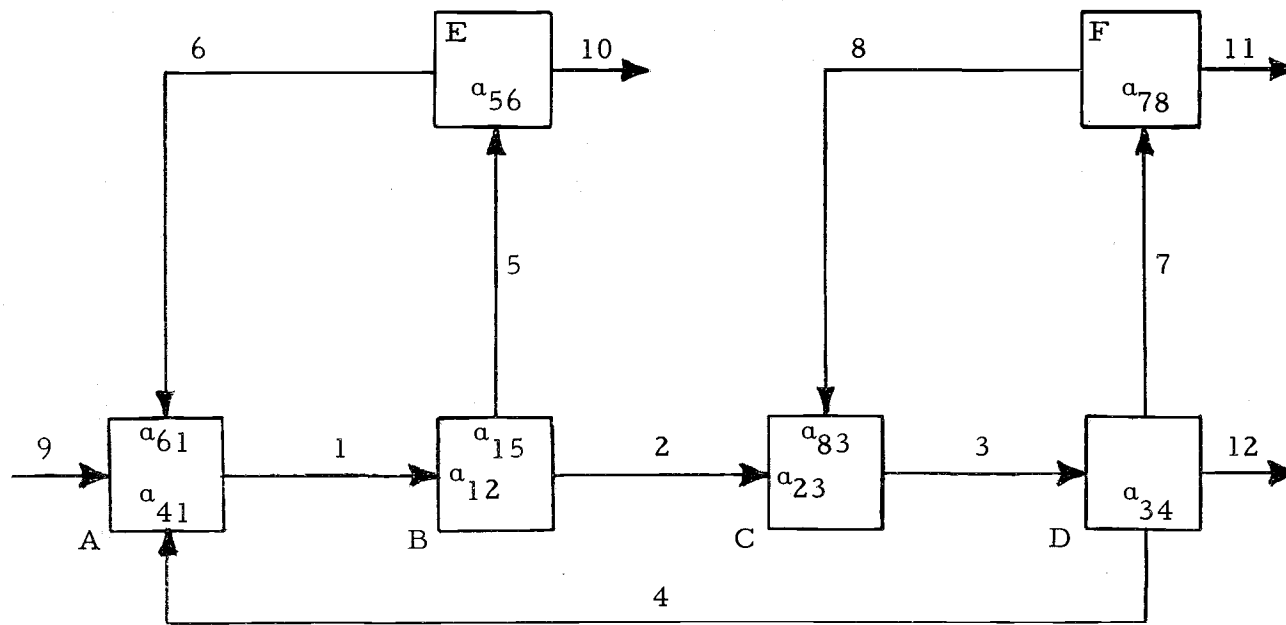
Intuitively, this seems to be an illogical choice of cut sets. Such a choice will indeed result in a greater dominant eigenvalue than the cut sets which maintain the natural order of calculation for each iterative loop.

Consider for example the three loop, single variable case of Figure 15. The a 's represent the sensitivity coefficients of the output streams with respect to the inputs. Thus,

$$a_{15} = \frac{\partial x_5}{\partial x_1}, \quad a_{83} = \frac{\partial x_3}{\partial x_8}, \quad \text{etc.} \quad 4.64$$

Since units A and C are mixing units, the values of a_{61} , a_{41} , a_{83} , and a_{23} are all equal to one.

For this simplified system it is possible to calculate the eigenvalues of various cut sets and compare their magnitudes directly. First of all a cut set in the equivalent group, $\{1, 8\}$ will be examined.



A. Mixer C. Mixer E. Flasher
 B. Flasher D. Separator F. Flasher

Figure 15. Recycle system for cut set comparison.

$$J_{18} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_8} \\ \frac{\partial F_8}{\partial x_1} & \frac{\partial F_8}{\partial x_8} \end{bmatrix} = \begin{bmatrix} a_{56} a_{15} + a_{34} a_{12} & a_{34} \\ a_{78} a_{37} a_{12} & a_{78} a_{37} \end{bmatrix} \quad 4.65$$

The eigenvalues are calculated from,

$$|J_{18} - I\lambda| = 0 \quad 4.66$$

as

$$\lambda = \frac{(a_{56} a_{15} + a_{34} a_{12} + a_{78} a_{37}) \pm \sqrt{(a_{56} a_{15} + a_{34} a_{12} + a_{78} a_{37})^2 - 4(a_{15} a_{56} a_{78} a_{37})}}{2} \quad 4.67$$

The maximal or dominant eigenvalue will be the positive root so:

$$\lambda_{18 \max} = \frac{(a_{56} a_{15} + a_{34} a_{12} + a_{78} a_{37}) + \sqrt{(a_{56} a_{15} + a_{34} a_{12} + a_{78} a_{37})^2 - 4a_{56} a_{15} a_{78} a_{37}}}{2} \quad 4.68$$

The cut set $\{1, 3\}$ is one which cuts the same loop twice. Its Jacobian is:

$$J_{13} = \begin{bmatrix} a_{56} a_{15} & a_{34} \\ a_{12} & a_{78} a_{37} \end{bmatrix} \quad 4.69$$

The maximum eigenvalue is calculated as:

$$\lambda_{13 \max} = \frac{(a_{56} a_{15} + a_{78} a_{37}) + \sqrt{(a_{56} a_{15} + a_{78} a_{37})^2 - 4(a_{56} a_{15} a_{78} a_{37} - a_{34} a_{12})}}{2}$$

4.70

Define the terms:

$$a = a_{56} a_{15} + a_{78} a_{37}$$

$$b = a_{56} a_{15} a_{78} a_{37}$$

$$c = a_{34} a_{12}$$

4.71

$$\therefore \lambda_{18 \max} = \frac{(a+c) + \sqrt{(a+c)^2 - 4b}}{2}$$

4.72

$$\lambda_{13 \max} = \frac{a + \sqrt{a^2 - 4(b-c)}}{2}$$

4.73

The goal now is to compare the two maximum eigenvalues and determine which will be greater. First assume that:

$$\lambda_{18, \max} \geq \lambda_{13, \max}$$

4.74

or

$$\frac{(a+c) + \sqrt{(a+c)^2 - 4b}}{2} \geq \frac{a + \sqrt{a^2 - 4(b-c)}}{2}$$

4.75

This reduces to:

$$c + \sqrt{(a+c)^2 - 4b} \geq \sqrt{a^2 - 4(b-c)}$$

4.76

By squaring both sides:

$$\begin{aligned}
c^2 + (a+c)^2 - 4b + 2c\sqrt{(a+c)^2 - 4b} &\geq a^2 - 4(b-c) \\
2c^2 + a^2 + 2ac - 4b + 2c\sqrt{(a+c)^2 - 4b} &\geq a^2 - 4b + 4c \\
2c(c+a + \sqrt{(a+c)^2 - 4b}) &\geq 4c \\
\therefore \frac{(a+c) + \sqrt{(a+c)^2 - 4b}}{2} &\geq 1 \tag{4.77}
\end{aligned}$$

The quantity on the left is $\lambda_{18, \max}$

$$\therefore \lambda_{18, \max} \geq 1 \tag{4.78}$$

This is the condition that Equation 4.74 is true. However, for any convergent system, Equation 4.78 is false. Therefore Equation 4.74 must be false. Thus, in general:

$$\lambda_{18, \max} < \lambda_{13, \max}$$

Therefore the dominant eigenvalue of cut set $\{1, 3\}$ is always greater than that of cut set $\{1, 8\}$ and set $\{1, 3\}$ will be the slower to converge. An identical result is obtained when comparing set $\{1, 8\}$ with sets $\{1, 6, 8\}$, $\{3, 6, 7\}$, and $\{1, 2, 3\}$.

Due to the complexity of calculating the eigenvalues for large, multiple variable systems, it may be impossible to prove in general that cut sets which cut one loop more than once and are therefore not members of the equivalent group will always have a dominant eigenvalue greater than that of sets which are in the equivalent group. A

number of process examples, however, have shown this to be the case.

Process System Examples

The Cavett hydrocarbon separation system, the salt-water mixing example and Case I of the phosphoric acid pilot plant described in Appendix A were rerun for a number of cut sets. The procedure in each case was to start calculation for some initial guess and perform one iteration. The values obtained for all streams during this iteration were then stored as data and then used to provide starting values for all of the various cut sets. This simulates the effect of starting a calculation and changing cut sets after one iteration.

Each system was solved by successive substitutions and by ASYM for eight different cut sets. Four of the cut sets were from the equivalent group, and the other four were not. The intermediate values of the norm of the test vector were obtained for the SUCCESS cases. These values were used to calculate the dominant eigenvalue for the cut set by Equation 3.69

$$\lambda = \frac{\|\theta_I\|}{\|\theta_{I-1}\|} \quad 3.69$$

Convergence was tested over the same streams for each case with a tolerance of .0001. For the hydrocarbon separation system, convergence was tested on streams 2, 3, 5, 7, 8, and 9. The

streams 3, 5, 7, 10, 13, and 14 were used for the salt-water mixing system. The phosphoric acid plant was converged on streams 6, 7, 8, 9, 14, and 28.

The results of these runs are shown in Tables 10, 11, and 12. As can be seen the equivalent sets in Table 10 all converged by SUCCESS in 45 or 46 iterations with a dominant eigenvalue of 0.845. In contrast the non-equivalent sets required from 51 to 73 iterations to converge with dominant eigenvalues ranging from 0.863 to 0.910. The two cut sets in the non-equivalent classification which had the same dominant eigenvalue were selected so that both cut the same loop twice but with different streams within that loop. This demonstrates that cut sets in this group form subgroups of sets which are equivalent to each other. This follows directly from the statement that cut sets which cause the units in each loop to be calculated in the same order are equivalent.

Table 10. Convergence and dominant eigenvalues with various cut sets for salt-water mixing example.

Cut Set	Group*	Iterations to Converge		Dominant Eigenvalue
		ASYM	SUCCESS	
3, 7	E	10	45	0.845
9, 7, 11	E	10	45	0.845
5, 14	E	10	46	0.845
8, 9	E	10	45	0.845
3, 14	NE	11	51	0.863
10, 13	NE	11	52	0.863
7, 8, 13	NE	22	56	0.875
7, 9, 13	NE	73	73	0.910

* E - Equivalent Group. N - Non-equivalent Group.

Table 11. Convergence and dominant eigenvalues with various cut sets for the hydrocarbon separation system.

Cut Set	Group*	<u>Iterations to Converge</u>		Dominant Eigenvalue
		ASYM	SUCCESS	
5, 9, 10	E	34	87	0.929
3, 7	E	36	88	0.929
2, 8	E	34	88	0.929
3, 8, 9	E	35	88	0.929
2, 7	NE	51	128	0.954
2, 8, 9	NE	52	128	0.954
2, 3, 10	NE	38	103	0.941
5, 7, 10	NE	44	104	0.942

* E - Equivalent Group. N - Non-equivalent Group.

Table 12. Convergence and dominant eigenvalues with various cut sets for phosphoric acid plant.

Cut Set	Group*	<u>Iterations to Converge</u>		Dominant Eigenvalue
		ASYM	SUCCESS	
6, 7, 8, 14, 28	E	10	56	0.922
15, 24, 29	E	10	56	0.922
9, 16, 25	E	11	56	0.922
13, 15, 28	E	10	56	0.922
9, 15, 24	NE	12	68	0.939
9, 15, 25	NE	13	60	0.929
11, 16, 24	NE	11	63	0.934
13, 15, 25, 29	NE	35	80	0.952

From Table 11 it is seen that the cut sets in the equivalent group for the Cavett problem converge in 87 and 88 iterations by SUCCESS with the dominant eigenvalue equal to 0.929. The convergence ranged from 103 to 128 iterations with dominant eigenvalues from 0.941 to 0.954 for the cut sets in the non-equivalent groups.

The runs with equivalent cut sets for the phosphoric acid plant are seen in Table 12 to converge in 56 iterations by SUCCESS with a dominant eigenvalue equal to 0.922. The non-equivalent sets yielded dominant eigenvalues ranging from 0.929 to 0.952 and converged in 60 to 80 iterations.

The use of ASYM to accelerate the convergence resulted in vastly improved results for all except one of the cases. The non-equivalent cut sets, however, still did not converge as fast as the equivalent cut sets.

The use of convergence promotion on the non-equivalent sets is complicated by the fact that in addition to having larger dominant eigenvalues, these sets have large negative eigenvalues. This is seen by the oscillating behavior of the norm of the test vector as shown in Figure 17 for the salt-water mixing system. For comparison the test vector for a cut set from the preferred group is shown in Figure 16. The effect of the oscillation is to increase the number of iterations required for the system to approach asymptotic behavior closely enough to accelerate. The results in Tables 10 and 11 indicate that this effect is increased as the dominant eigenvalue increases.

The Cavett problem was investigated further to demonstrate the effect of initial guesses on the number of iterations to converge for various equivalent cut sets. Five cut sets in the equivalent group were compared for their adaptability to convergence promotion. As

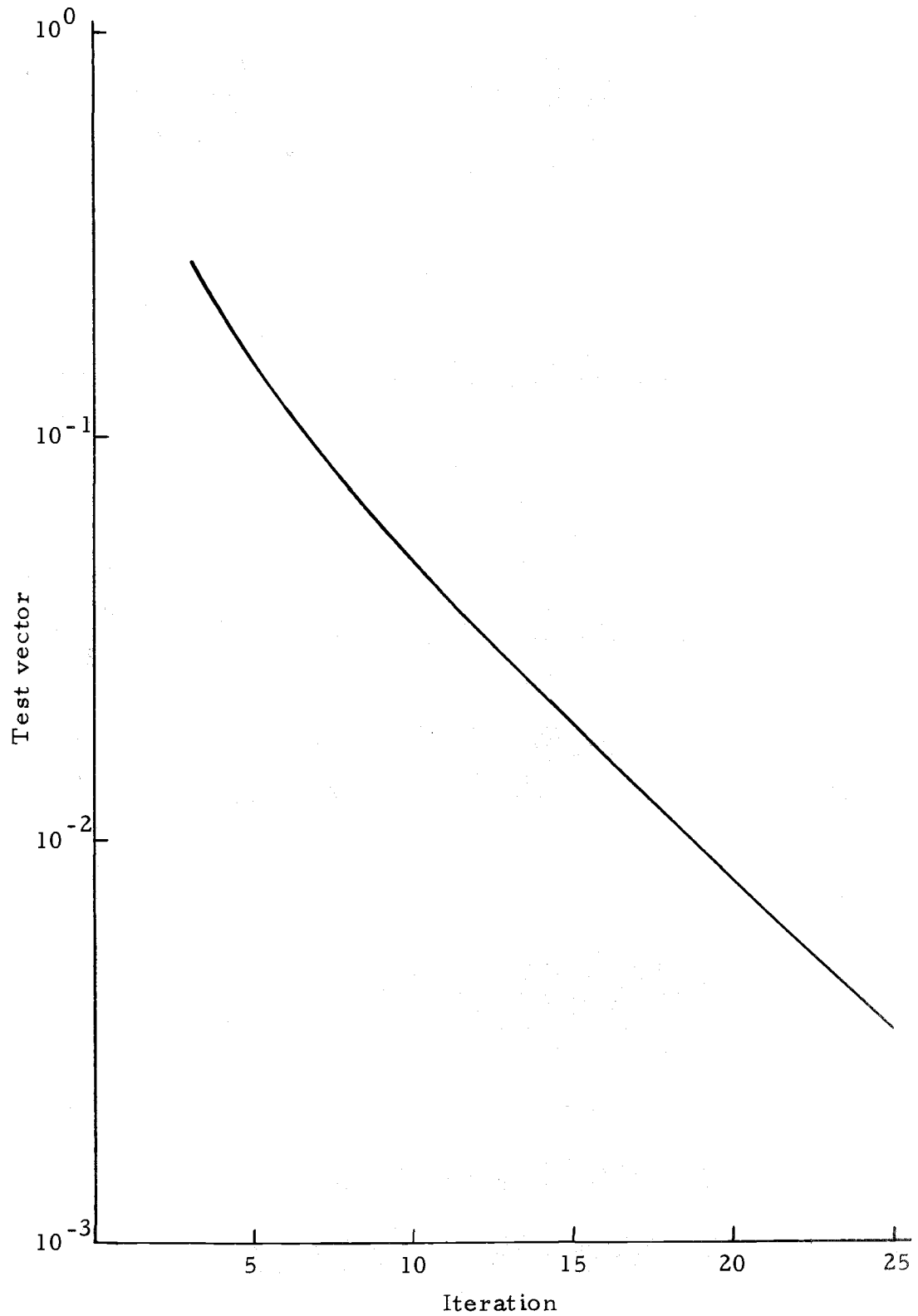


Figure 16. Test vector for salt-water mixing example, cut set {3,7}.

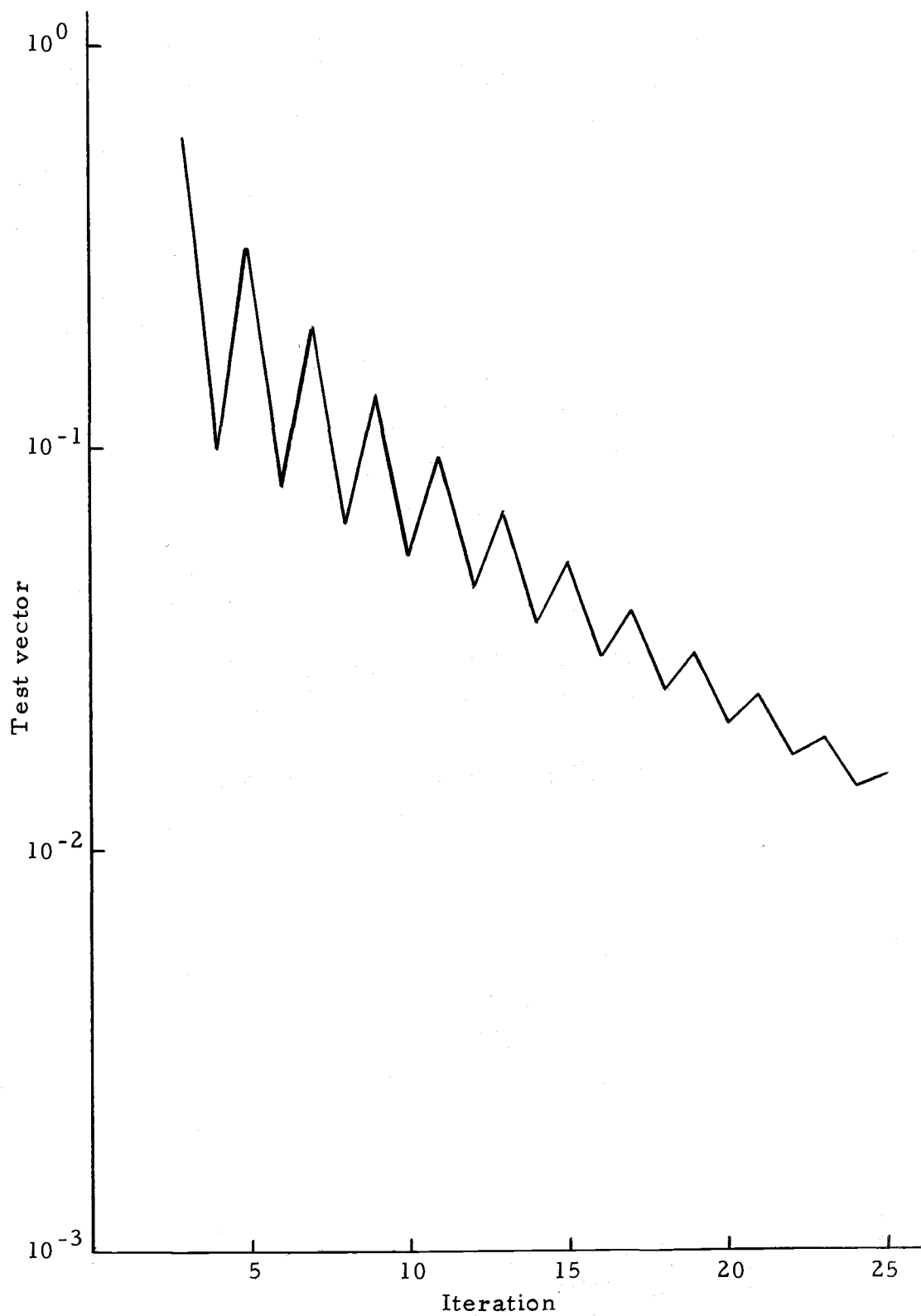


Figure 17. Test vector for salt-water mixing example, cut set {7,9,13}.

before one iteration was run to establish values for starting all of the cut sets. In this case three different sets of starting values corresponding to three initial guesses were generated. The dominant eigenvalue for these sets was 0.929. Table 13 gives the iteration numbers at which acceleration steps were taken and the number of iterations to converge by ASYM.

The differences of three to five iterations to converge are due to the variations in the Jacobians for the cut sets. The Jacobians cause each set to approach linearity uniquely and thus some sets are ready to accelerate at different points and gain more from their acceleration steps. It is also noted that the effect of the Jacobian varied for the different initial guesses. For the second and third initial guesses, cut set $\{3, 7\}$ converged the fastest with acceleration, but was slowest for the first initial guess.

These results demonstrate that even though the equivalent sets will converge at the same rate when the iterative values are sufficiently close to the solution, there will be differences for cases when the initial guesses are far from the solutions.

Table 13. Convergence of Cavett (2) problem by ASYM for equivalent cut sets.

Cut Set Streams	Initial Guess 1		Initial Guess 2		Initial Guess 3	
	Acceleration at Iterations	Iterations to Converge	Acceleration at Iterations	Iterations to Converge	Acceleration at Iterations	Iterations to Converge
2, 8	14, 31, 40	42	9, 32	34	19, 38, 44	46
3, 7	14, 29, 39	41	9, 27	29	12, 32, 41	43
5, 9, 10	14, 30, 37	39	12, 31	33	20, 39, 45	47
3, 8, 9	14, 32, 39	41	10, 29	31	19, 39	48
5, 6, 10	14, 29	42	11, 32	34	12, 31	43

5. CONCLUSIONS AND RECOMMENDATIONS

The asymptotic method developed in this work provides flexibility in solving iterative problems and adapts to each particular problem depending upon how long it takes the iteration to reach asymptotic behavior. The key to the method is the adaptive criterion for when to apply acceleration. This criterion has been shown to be more successful over a wide range of problems than the application of promotion at regular intervals or the criterion which utilizes a value preset by the user.

It has been shown that many choices of cut sets for complex recycle systems are equivalent in convergence properties because they have the same eigenvalues. Furthermore, other cut sets not in the equivalent group converge at a slower rate than the equivalent cut sets. Cut sets in the equivalent group calculate the units in each individual loop in the order in which the units appear in the loop. This is equivalent to the requirement that no loop be cut more than once.

Further studies into cut set selection should consider the early behavior of the iterative vectors. Any cut set selection technique designed to distinguish among the various cut sets in the equivalent group would have to take advantage of the differences in behavior caused by the relative magnitudes of the diagonal and off diagonal terms of the Jacobians. Off diagonal terms indicate the degree of variable interaction.

A valuable extension of this work would be the automation of the cut set selection. The current algorithms appearing in the literature for analyzing process systems and locating recycle loops should be evaluated to determine the most suitable technique to modify for locating and defining the equivalent cut sets. Provision could still be made for the user to indicate streams which might be regarded as preferable.

BIBLIOGRAPHY

1. Barkley, R.W. and R.L. Motard. Decomposition of nets. Chemical Engineering Journal 3:265-275. 1973.
2. Cavett, R.H. Application of numerical methods to the convergence of simulated processes involving recycle loops. In: Proceedings of the American Petroleum Institute. Vol 43, sec. 3. Refining. New York. 1963. p. 57-76.
3. Christensen, J.H. and D.F. Rudd. Structuring design computations. A.I.Ch.E. Journal 15:94-100. 1969.
4. CRC Standard Mathematical Tables, 20th Edition. Chemical Rubber Co. Cleveland. 1972.
5. Crowe, C.M. et al. Chemical Plant Simulation. Prentice-Hall. 1971. 368 pages.
6. Forder, G.J. and H.P. Hutchison. The analysis of chemical plant flowsheets. Chemical Engineering Science 24:771-785. 1969.
7. Genna, P.L. and R.L. Motard. Optimal decomposition of process networks. Paper presented at 66th Annual AIChE meeting. Philadelphia. November 1973. Session 28c.
8. Heaslet, M.A. and H. Lomas. Numerical predictions of radiative interchange between conducting fins with mutual irradiations. Washington, D.C., 1961. 22 p. (National Aeronautics and Space Administration. Technical Report NASA TR R-116)
9. Isakson, G. A method for accelerating the convergence of an iteration procedure. Journal of the Aeronautic Science 16:443. 1949.
10. Kehat, E. and M. Shacham. Chemical process simulation programs - 1. Process Technology Int. 18:35-37. Jan. 1973.
11. Kehat, E. and M. Shacham. Chemical process simulation programs - 2. Process Technology Int. 18:115-118. Mar. 1973.
12. Kehat, E. and M. Shacham. Chemical process simulation programs - 3. Process Technology Int. 18:181-184. Apr. 1973.

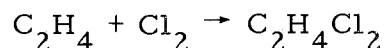
13. Klein, B. A rapid method of approximating the asymptotes to an iteration process. *International Journal of Computer Mathematics* 1:91-95. 1965.
14. Kliesch, H.C. An analysis of steady-state process simulation: formulation and convergence. Ph.D. thesis. New Orleans. Tulane University. 1966. 139 numb. leaves.
15. Kwon, Y.J. Asymptotic convergence technique and executive concept for steady state process systems analysis. Ph.D. thesis. Corvallis, Oregon. Oregon State University. 1970. 157 numb. leaves.
16. Kwon, Y.J. and E. Elzy. DISCOSSA-An executive for steady state process systems analysis. Corvallis, Oregon State University, Dept. of Chemical Engineering, 1972. Various paging. (Duplicated)
17. Lee, W. and D.F. Rudd. On the ordering of recycle calculations. *A.I. Ch.E. Journal* 12:1184-1190. 1966.
18. Motard, R.L., H.M. Lee and R.W. Barkley. Chemical engineering simulation system - user's guide. Houston, University of Houston, Dept. of Chemical-Petroleum Engineering, 1968. 172 numb. leaves.
19. Nagiev, M.F. The theory of recycle processes in chemical engineering. New York, MacMillan, 1964. 278 p.
20. _____ Material balance in complex and multistage recycle chemical processes. *Chemical Engineering Progress* 53:297-303. 1957.
21. Norman, R.L. A matrix method for location of cycles of a directed graph. *A.I. Ch.E. Journal* 11:450-452.
22. Orbach, O. A study of the promotion of convergence in the calculation of complex chemical plants. Ph.D. thesis. Hamilton, Ontario, McMaster University. 1969.
23. Orbach, O. and C.M. Crowe. Convergence promotion in the simulation of chemical processes with recycle - the Dominant Eigenvalue Method. *Canadian Journal of Chemical Engineering* 49:509-513. 1971.

24. Rinard, I.H. and D.L. Ripps. The steady-state simulation of continuous chemical processes. Chemical Engineering Progress Symposium Series 61(55):34-51. 1965.
25. Rosen, E.M. A machine computation method for performing material balances. Chemical Engineering Progress 58:69-73. 1962.
26. Roy, S.N. Some aspects of multivariate analysis. New York, Wiley. 1957.
27. Rubin, D.I. Generalized material balance. Chemical Engineering Progress Symposium Series 58(37):54-61. 1962.
28. Sargent, R.W.H. and A.W. Westerberg. Speed-up in chemical engineering design. Transactions of the Institution of Chemical Engineers (London) 42:190-197. 1964.
29. Shacham, M. and R.L. Motard. Applications of the theory of linear recycle systems. Paper presented at 78th AIChE national meeting, Salt Lake City, August 1974.
30. Steward, D.V. On an approach to techniques for the analysis of the structure of large systems of equations. SIAM Review 4:321-343. 1962.
31. Upadhye, R.S. and E. A. Grens. An efficient algorithm for optimum decomposition of recycle systems. A.I.Ch.E. Journal 18:533-539. 1972.
32. Wegstein, J.H. Accelerating convergence of iterative processes. Communications of the Association for Computing Machinery 1:9-13. 1958.
33. Yazhari, F. Simulation model of a total recycle urea process. M.S. thesis. Corvallis, Oregon State University. 1974. 91 numb. leaves.
34. Yazhari, F., W. Byers and E. Elzy. Simulation of USBM phosphoric acid pilot plant (Albany, Oregon). Unpublished report. Corvallis, Oregon State University, Dept. of Chemical Engineering. 1973.

APPENDIX A

Descriptions of Example SystemsHypothetical Ethylene Dichloride Process

A hypothetical process for the production of ethylene dichloride was described by Kwon and Elzy (16). The product was formed by the reaction:



The information flow diagram of the system is shown in Figure A-1. The reactor yields 90% conversion of C_2H_4 on each pass. The separator overhead stream, 5, consists of 98% of the Cl_2 entering the unit, 92% of the C_2H_4 entering, and 0.1% of the $\text{C}_2\text{H}_4\text{Cl}_2$ entering. Five percent of the overhead stream is purged by the splitter.

Photochemical Reaction Process

An isothermal, photochemical reaction carried out in a constant volume, well stirred reactor was described by Kwon (15). The system involves five reactions and seven components. The equations describing the system at steady state were derived by Kwon as:

$$\begin{aligned}
 w_1 &= \frac{F_1}{F + k_1 w_2 + k_4 w_6 I^{1/2}} \\
 w_2 &= \frac{F_2}{F + k_1 w_1 + 2k_2 w_3} \\
 w_3 &= \frac{F_3}{F + k_2 w_2} \\
 w_4 &= \frac{2k_1 w_1 w_2}{F + k_3 w_5} \\
 w_5 &= \frac{3k_2 w_2 w_3}{F + k_3 w_4} \\
 w_6 &= \frac{2k_3 w_4 w_5}{F + k_4 w_1 I^{1/2}} \\
 w_7 &= \frac{2k_4 w_1 w_6 I^{1/2}}{F}
 \end{aligned} \tag{A-1}$$

The w 's represent the weight fractions of the components. The k 's are reaction rate constants, the F 's feed rates, and I the light intensity. The values of these parameters are:

$$\begin{aligned}
 k_1 &= 17.6 & F_1 &= 3.0 & I &= 0.6 \\
 k_2 &= 73.0 & F_2 &= 4.75 \\
 k_3 &= 51.3 & F_3 &= 1.25 \\
 k_4 &= 23.0 & F &= 9.0
 \end{aligned}$$

The schematic of Figure A-2 shows the method of solution of the simultaneous set of equations by a single node scheme.

The set of Equations (A-1) can be broken down into groups of one or more equations which are then represented by blocks analogous to process units. The blocks are arranged so that calculated outputs from one block can be used as input to the next block. Figure A-3 represents this schematically. Unit 1 calculates w_1 and w_3 , unit 2 w_2 , unit 3 w_4 and w_7 , unit 4 w_5 , and unit 5 w_6 . The terms in parentheses represent the information contained in the various streams. The system was solved using the cut set $\{2, 11, 3, 7\}$.

Nagiev's Example

The following set of linear, simultaneous equations represents the material balance for a fuel oil processing system (19, 20):

$$\begin{aligned}
 x_1 &= 1000. + 0.4624x_1 + 0.0436x_2 \\
 x_2 &= 200. + 0.235x_1 + 0.67x_2 + 0.1667x_5 + 0.05x_7 \\
 x_3 &= 100. + 0.008x_1 + 0.0061x_2 + 0.445x_3 + 0.001x_4 \\
 x_4 &= 200. + 0.0026x_1 + 0.0022x_2 + 0.268x_4 + 0.011x_6 \\
 x_5 &= 50. + 0.0032x_1 + 0.0025x_2 + 0.213x_3 + 0.0833x_5 + 0.05x_7 \\
 x_6 &= 70. + 0.0017x_1 + 0.0014x_2 + 0.29x_4 + 0.482x_6 \\
 x_7 &= (0.75x_5 + 0.08x_7) / 0.27
 \end{aligned}
 \tag{A-2}$$

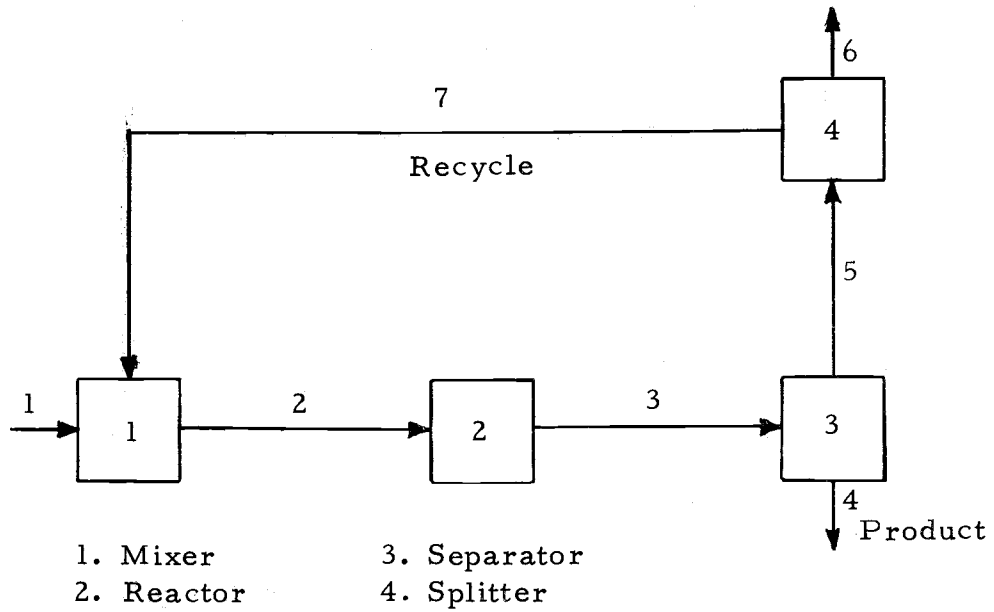


Figure A-1. Hypothetical ethylene dichloride process.

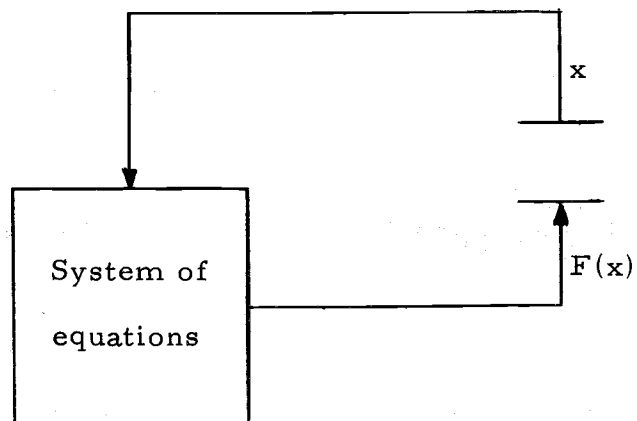


Figure A-2. Single node scheme for the solution of a set of simultaneous equations.

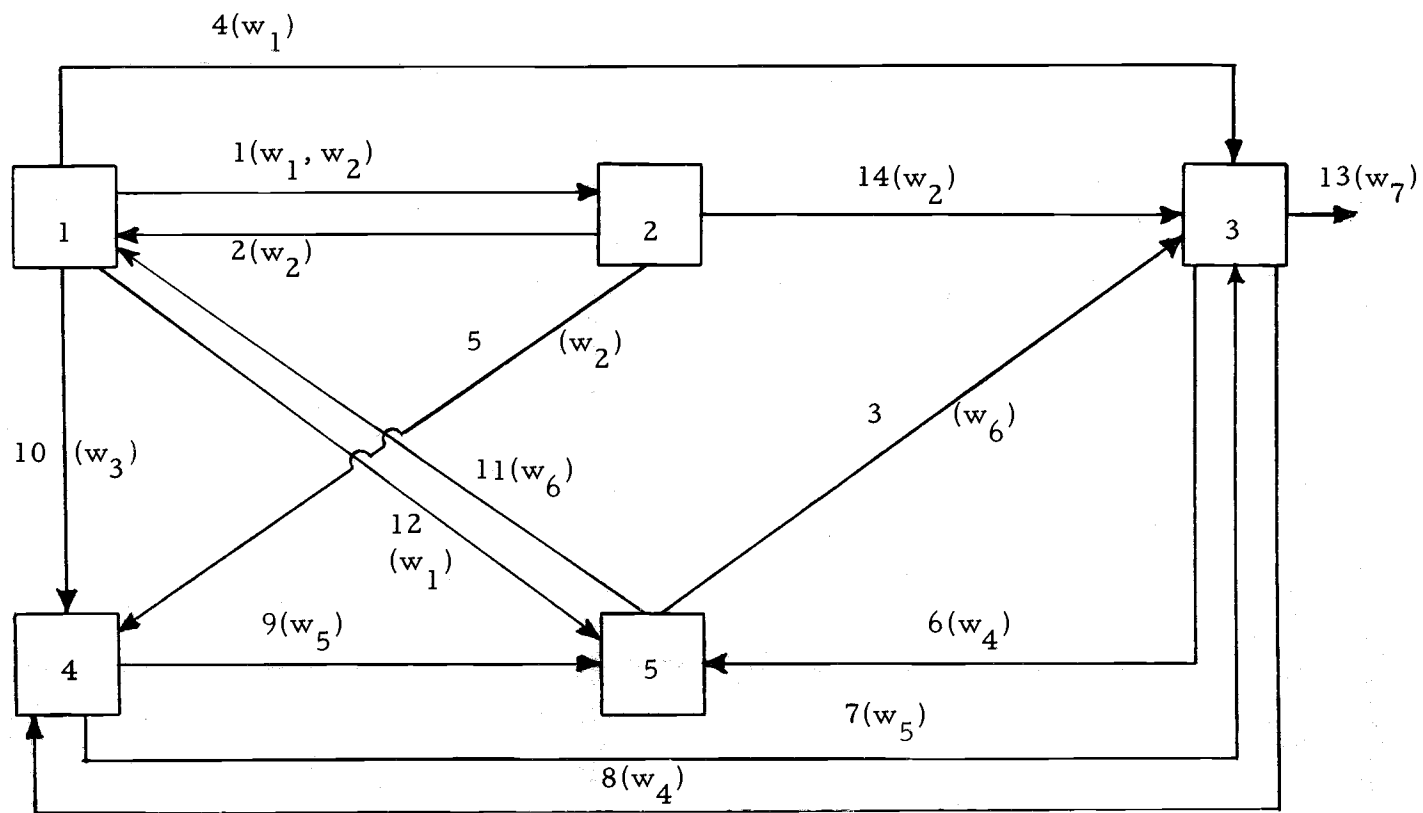


Figure A-3. Multi-node scheme for solution of photochemical reaction system.

The x 's represent the rate of charge of the various components. The system of equations was solved simultaneously using the single unit scheme of Figure A-2.

Nonlinear Set of Equations

A set of nonlinear equations was devised by Kwon (18) to study the stability of iterative procedures. Again, the single unit scheme of Figure A-2 was used to solve the system.

$$\begin{aligned}x_1 &= 100 - 0.2x_1 - 0.3x_2 e^{-1.5x_3} \\x_2 &= 50 - 2 \ln x_1 + 0.7x_2 - 0.1x_3^{1/2} \\x_3 &= 0.5x_1 e^{-0.03x_2}\end{aligned}\tag{A-3}$$

Salt-Water Mixing System

A linear, two variable schematic of a mixing system composed of mixers and splitters was presented by Kwon (15) and is shown in Figure A-4. Feed stream 1 consists of 900 lb-moles of water and 100 lb-moles of salt. Stream 2 is 400 lb-moles of water. The split fractions are specified as:

- Unit 1 20% of input exits in stream 4
- Unit 2 20% of input exits in stream 6
- Unit 4 50% of input exits in stream 7
- Unit 5 90% of input exits in stream 9
- Unit 6 90% of input exits in stream 10

The cut set $\{3, 7\}$ was used for the results presented in Chapter 3. A variety of cut sets were used in Chapter 4.

Hydrocarbon Separation Process

The hydrocarbon separation process shown in Figure A-5 was first presented by Cavett (2). The system is described in detail by Cavett (2).

This system has frequently been used as an example in studies of convergence techniques. Its behavior, particularly with the cut set $\{2, 7\}$, has made it a challenging problem and an unofficial standard for testing convergence techniques.

The system was run as a three member cut set problem with the set $\{5, 9, 10\}$ and as a two member set, $\{2, 7\}$, problem for the convergence comparisons of Chapter 3. Various cut sets were used for the results presented in Chapter 4.

Hypothetical Reactor System

Rosen (25) and Kwon (15) studied a hypothetical reaction system as diagrammed in Figure A-6. Basically the system centers on two well mixed reactors in which the series reactions:



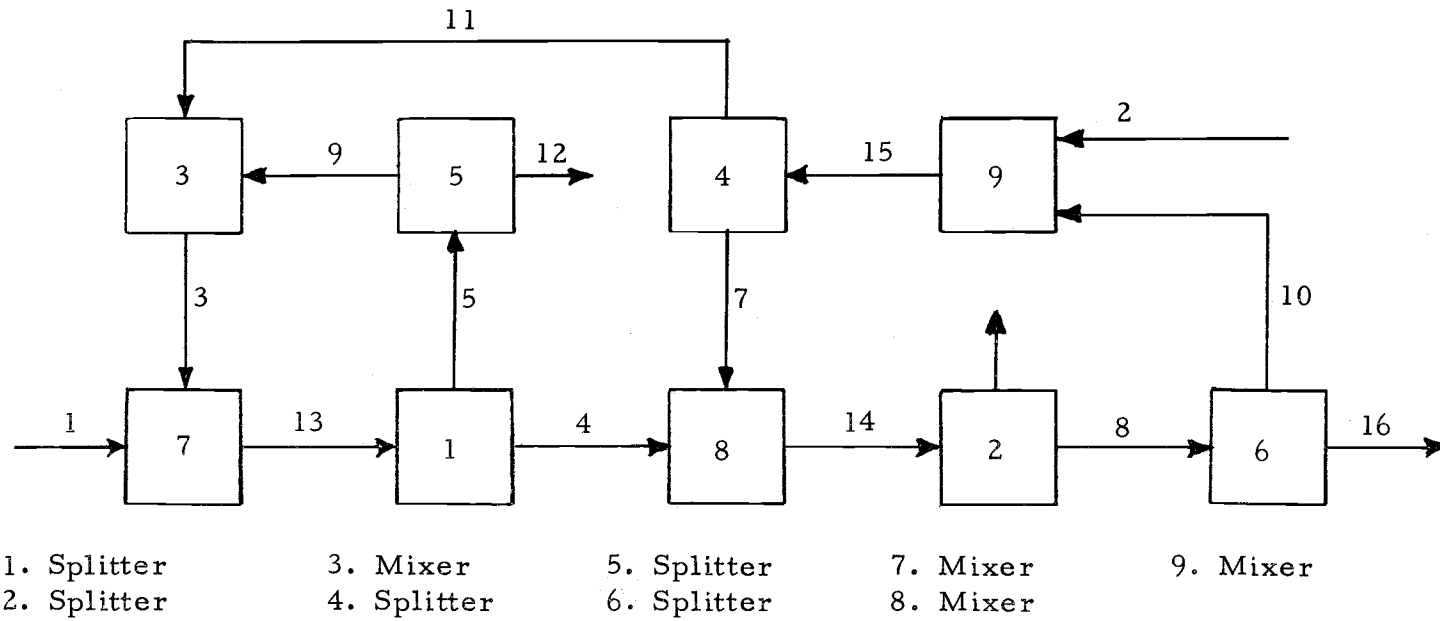


Figure A-4. Salt-water mixing system.

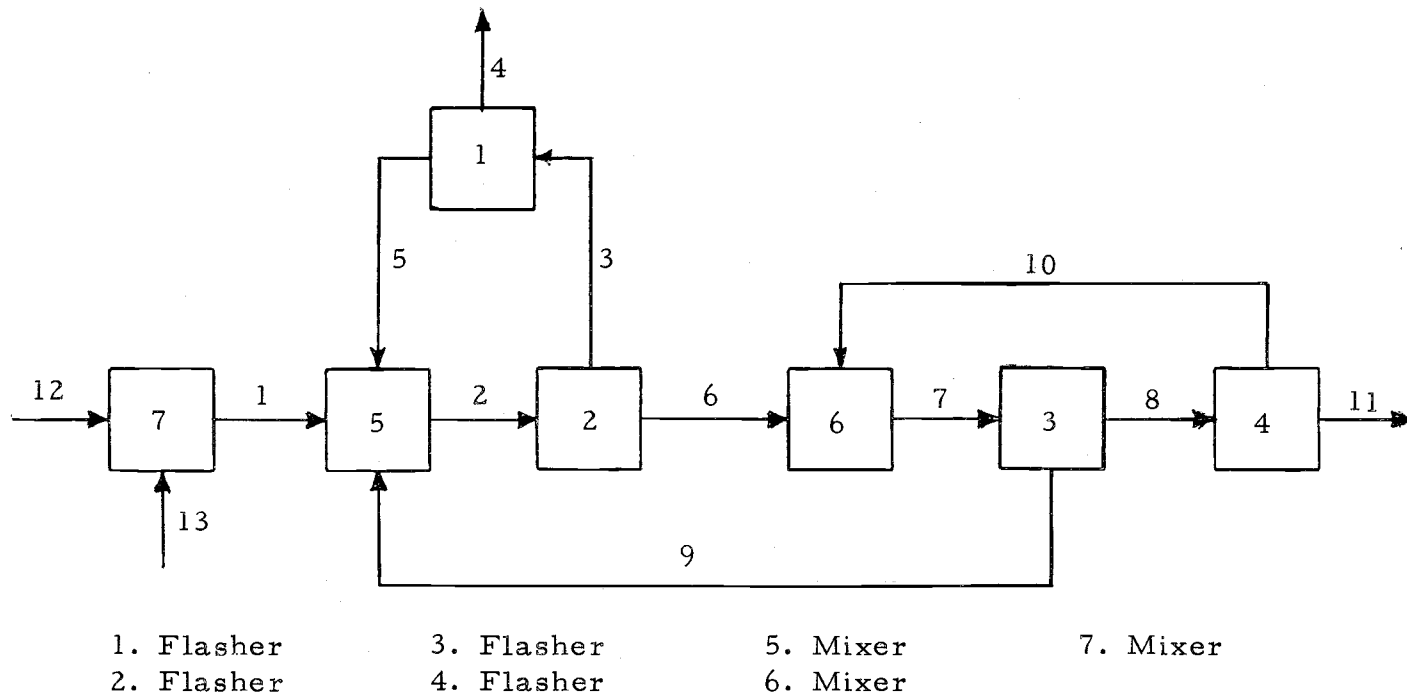


Figure A-5. Cavett's hydrocarbon separation process.

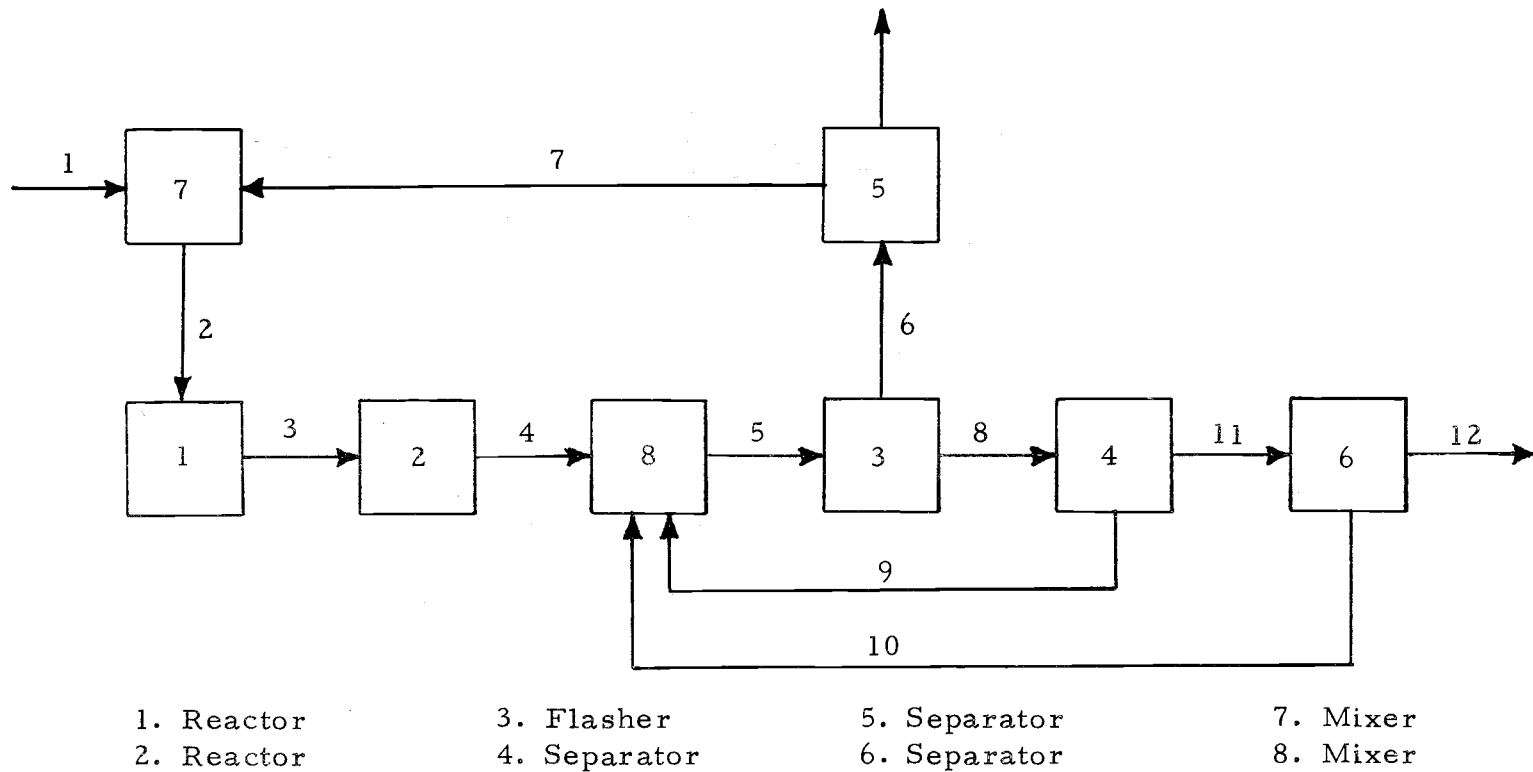


Figure A-6. Hypothetical reactor system.

take place. The k 's are reaction rate constants.

The details of the model used for the reactors and the flash units are reported elsewhere (15). The cut set $\{7, 9, 10\}$ was used to obtain the results presented in Chapter 3.

Bayer Hydrate Wash Section

The process shown in Figure A-7, the hydrate wash section of the Bayer alumina process, was presented by Orbach (22). The data and model of the separator units were also given (22). The clarifiers are liquid-solid counterflow units which wash certain components from the process streams. The only data not supplied by Orbach for the system was the split fraction for unit 11. This was back calculated from the data given and was found to be 32% for stream 12. The system was used for convergence comparison in Chapter 3 with the cut set $\{4, 8, 10, 15\}$.

Phosphoric Acid Plant

A phosphoric acid pilot plant simulation was developed (34) at Oregon State University to study processing alternatives. The details of the system and subroutines developed are available elsewhere (34).

Three different cases were used with this system. The cases involved rearrangement of some of the process units. Figures A-8, A-9, and A-10 show the process flow diagrams. The cut set

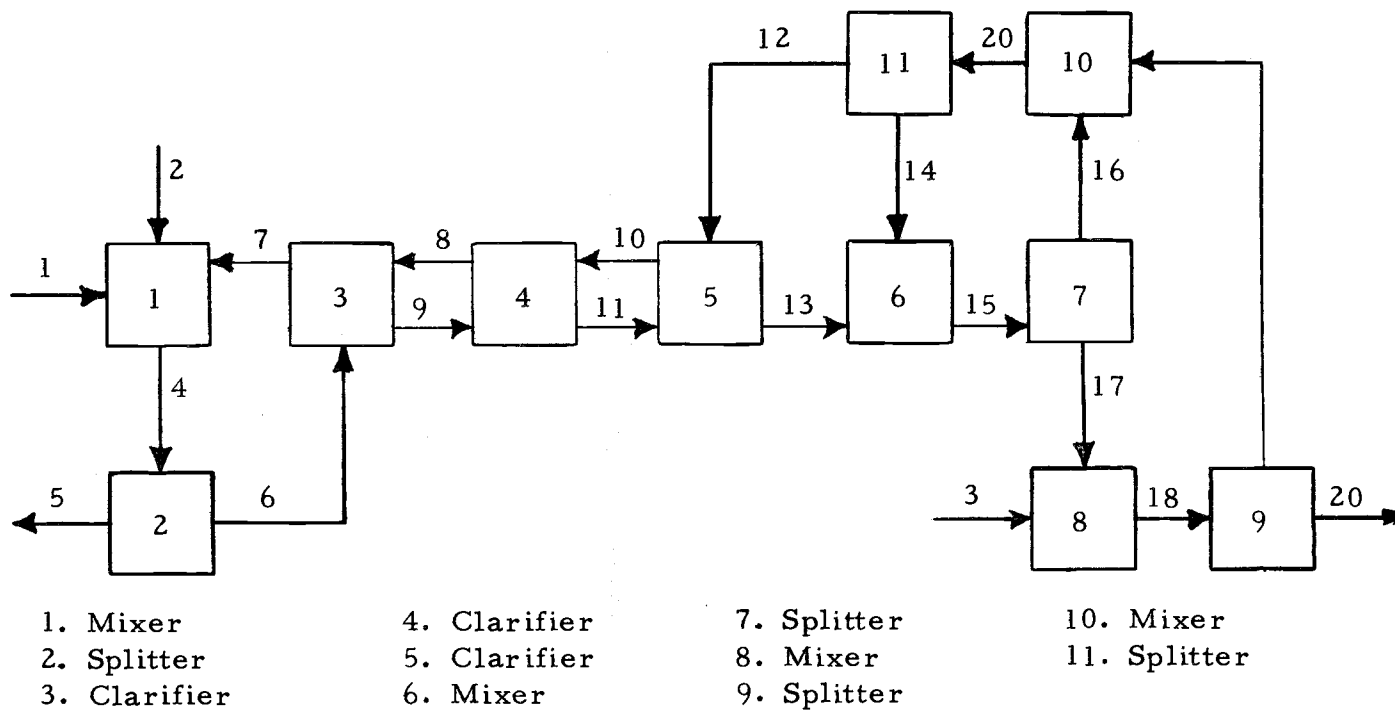
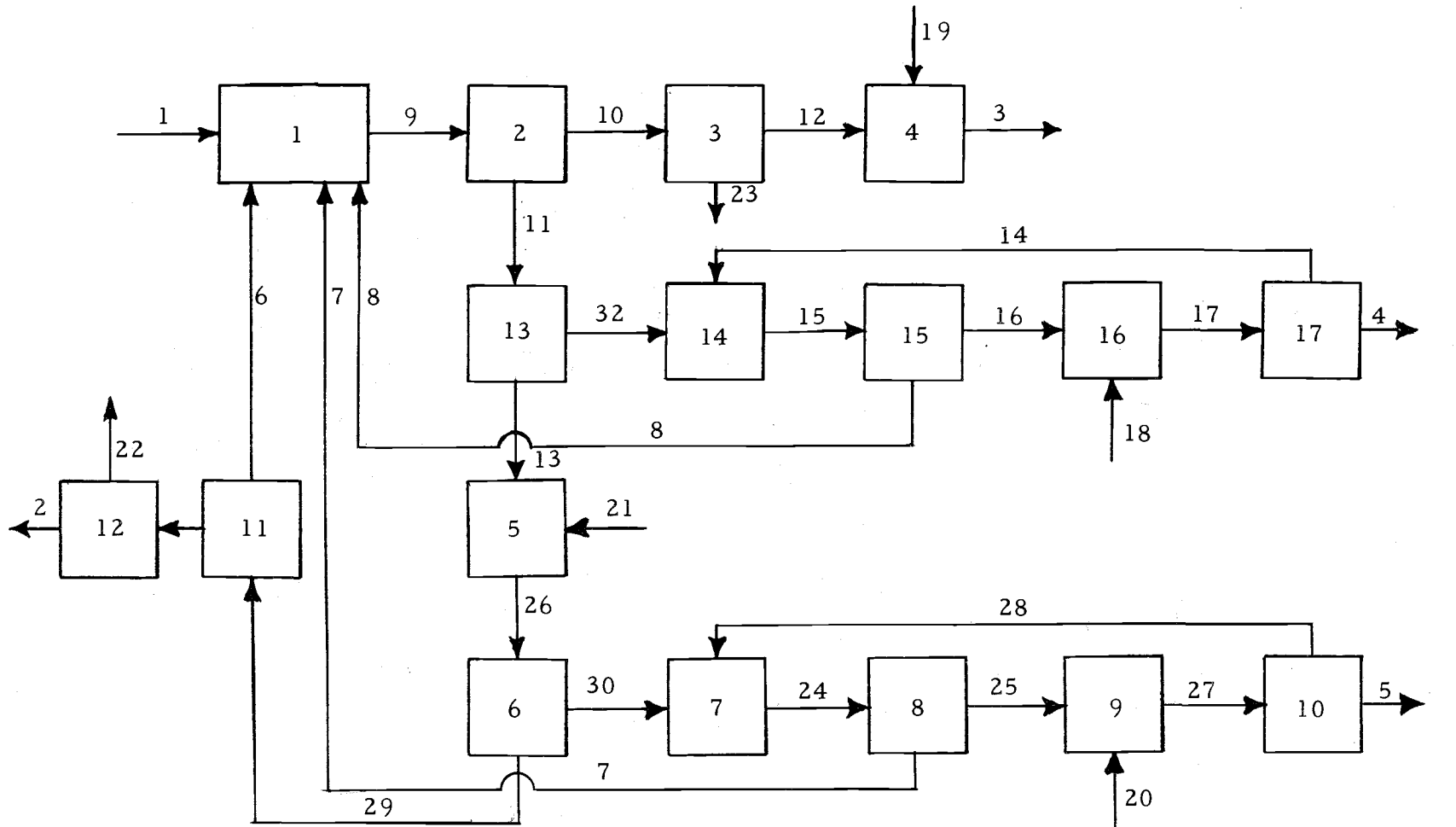


Figure A-7. Bayer hydrate wash section.



- | | | | | | |
|--------------|-----------------|--------------|----------------|---------------|---------------|
| 1. Reactor | 4. Mixer | 7. Mixer | 10. Separator | 13. Separator | 16. Mixer |
| 2. Separator | 5. Precipitator | 8. Separator | 11. Splitter | 14. Mixer | 17. Separator |
| 3. Separator | 6. Separator | 9. Mixer | 12. Evaporator | 15. Separator | |

Figure A-8. Phosphoric acid plant, Case I.

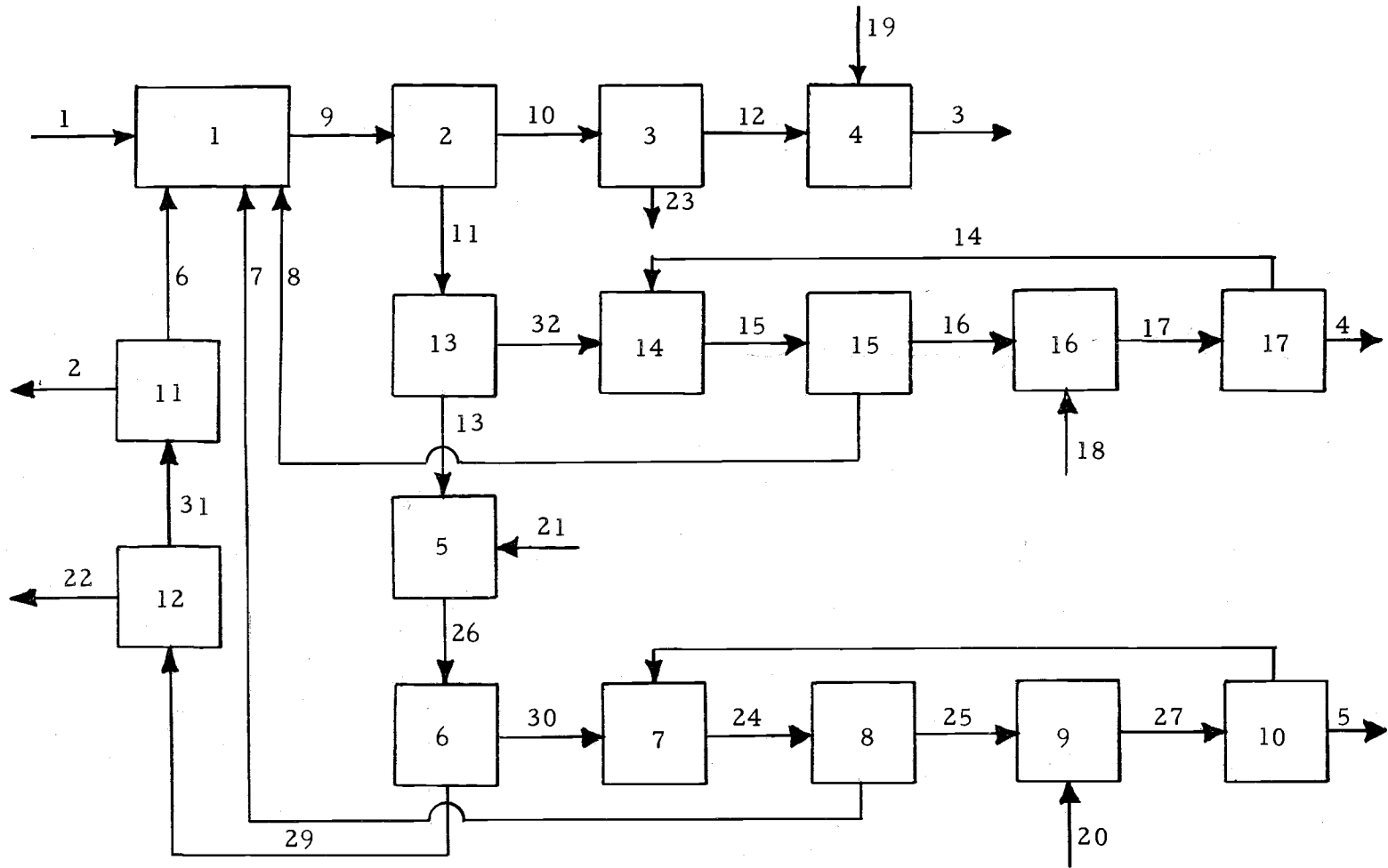


Figure A-9. Phosphoric acid plant, Case II.

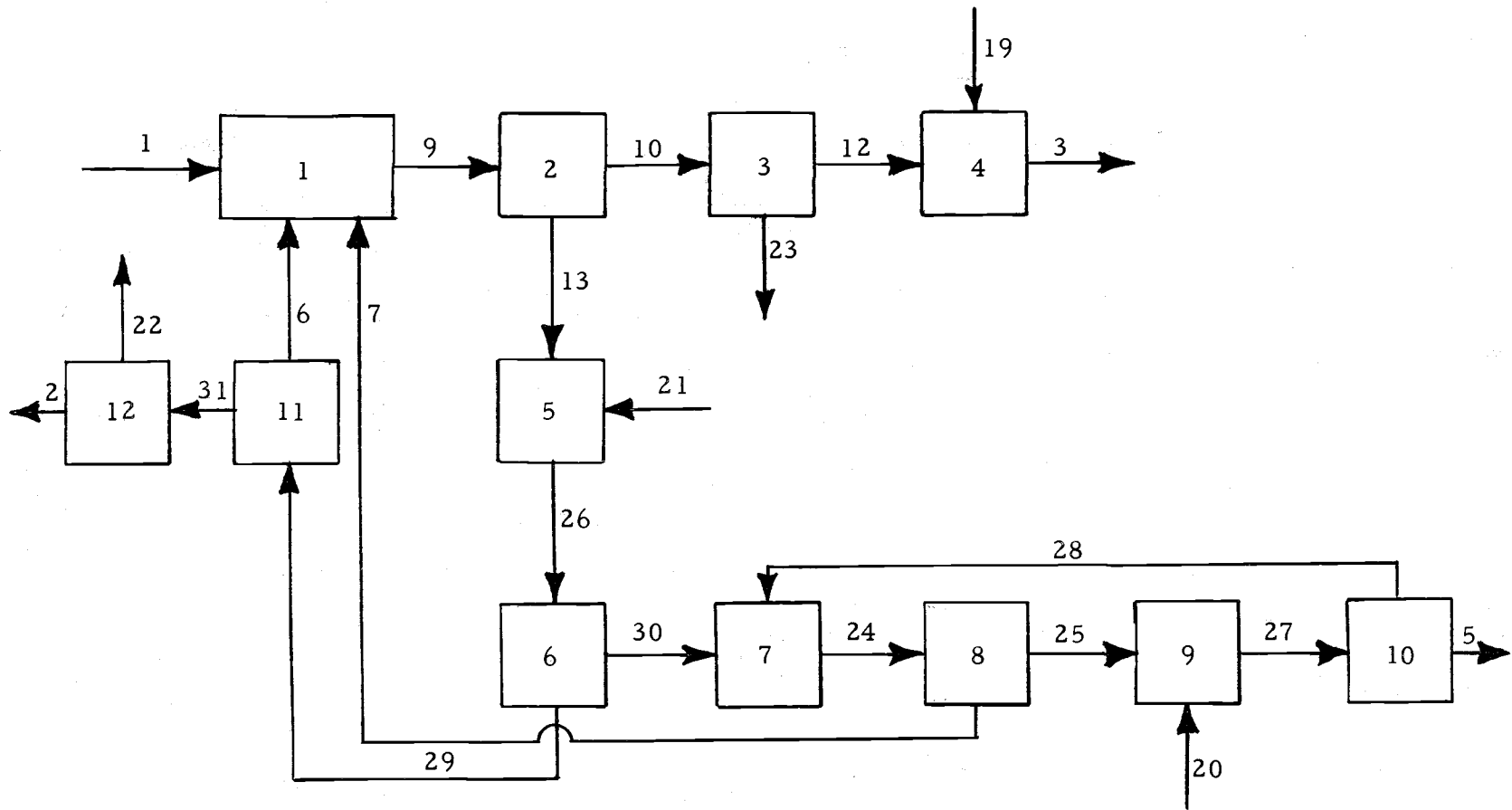


Figure A-10. Phosphoric acid plant, Case III.

{6, 7, 18, 14, 28} was used to converge the system for all three cases to obtain the results presented in Chapter 3. A number of cut sets were used for the results in Chapter 4.

Reichhold Urea Process

Yazhari (33) developed a steady-state simulation of the Chemico Urea Process owned and operated by Reichhold Chemicals, Inc., St. Helens, Oregon. The information flow diagram is shown in Figure A-11.

The simulation consists primarily of material balances with energy balances performed only for the flash separators to determine the operating temperatures. The reactor was simulated using a mathematical model for an industrial urea reactor developed by Dr. Eugene Elzy. Subroutine listing of the process units and more complete details of the simulation are given by Yazhari (33).

The cut set {2, 3, 17, 29, 36} was used to converge the system for the results of Chapter 3.

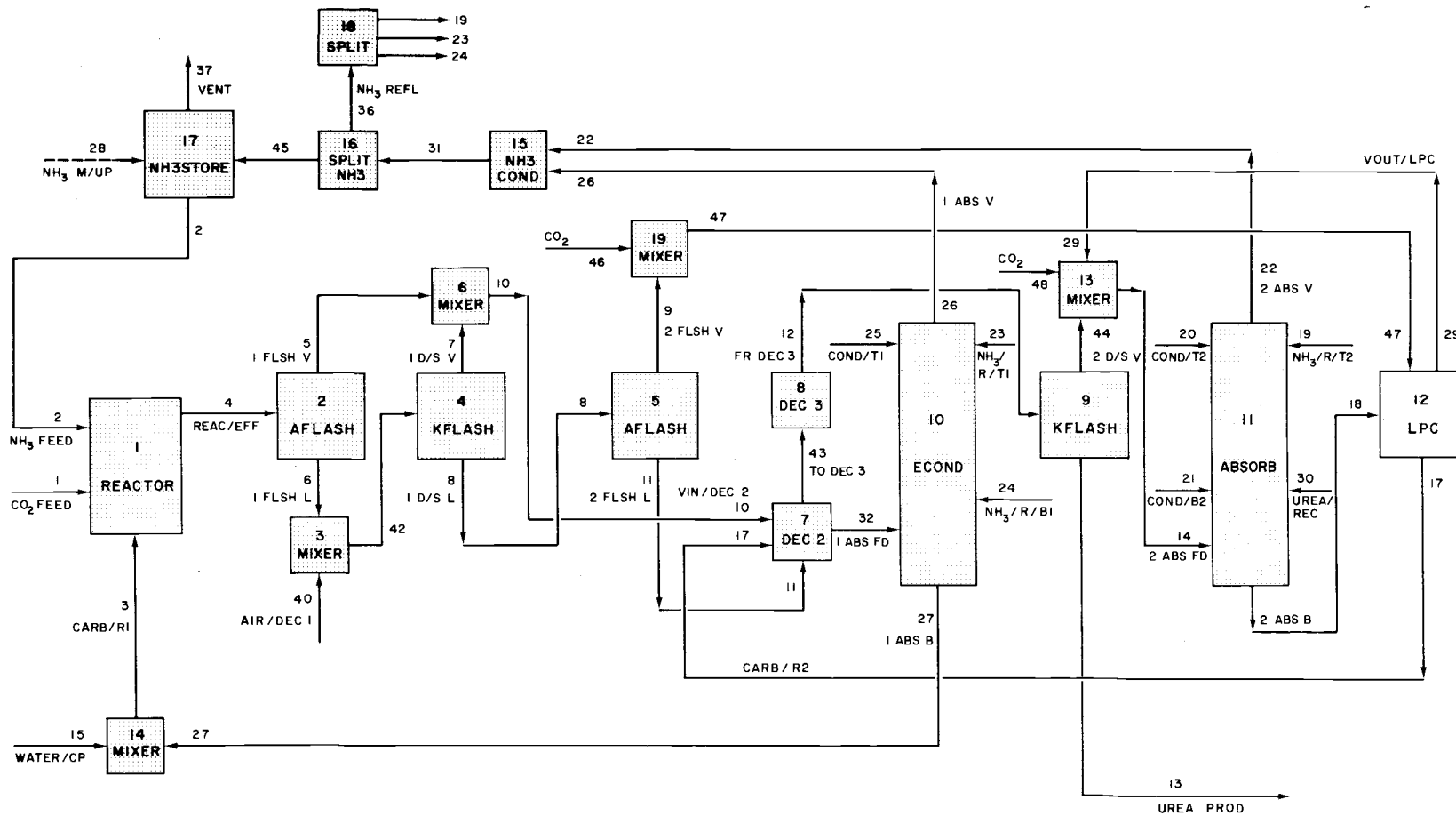


Figure A-11. Reichhold urea process.

APPENDIX B

Subroutine Listings

```

SUBROUTINE ASYM(KONCH)
  INCLUDE DCD
  DIMENSION KONCH(12), THETA(6,25), SMO(75,25), TMAX(6)
  ISM=0
  IDEX6=0
  N06=NOCOMP+6
  DO 5 I=1,6
    IF(NST(I).NE.0)GO TO 25
5  CONTINUE
    IF(IDEX5.NE.1)GO TO 10
    ISM=1
    IF(KFLAG.GE.2)GO TO 155
    CALL XASYM
    LOOP=IDEX2
    KFLAG=1
    GO TO 155
10  NETIND=IDEX4
    IDEX3=IDEX3+1
    IF(IDEX3.NE.1)GO TO 15
    KRAM1=0
    KDEX1=0
    KFLAG=0
15  KDEX1=KDEX1+1
    IDEX2=0
    IDEX6=1
    DO 20 I=1,6
      NST(I)=MNST(I)
      DO 20 J=1,NSLMAX
        XIF(I,J)=XXIF(I,J)
        XIG(I,J)=XXIG(I,J)
        XSC1(I,J)=XXSC1(I,J)
        XSC2(I,J)=XXSC2(I,J)
20  XSC3(I,J)=XXSC3(I,J)
25  IF(KFLAG.NE.0)GO TO 30
    DO 29 I=1,6
      IF(NST(I).EQ.0)GO TO 29
      KSB=NST(I)
      DO 28 J=1,N06

```

```

SMO(KSB,J)=XIG(I,J)
XIF(I,J)=0.0
XSC1(I,J)=0.0
XSC2(I,J)=0.0
28 XSC3(I,J)=0.0
29 CONTINUE
KRAM=0
IDEX2=0
KDEX=0
30 IDEX2=IDEX2+1
KDEX=KDEX+1
KONOK=0
KON=0
LOOP=IDEX2
IF(IDEX6.EQ.1)LOOP=IDEX3
LOOP=LOOP-1
CALL IOOUTPUT(LUNOUT)
LOOP=LOOP+1
DO 65 I=1,NSMAX
K CUT=0
KCON=0
DO 35 IL=1,6
IF(I.NE.NST(IL))GO TO 35
KSB=NST(IL)
TMAX(IL)=0.0
K CUT=IL
GO TO 40
35 CONTINUE
40 DO 45 IL=1,12
IF(I.NE.KONCH(IL))GO TO 45
KONC=KONCH(IL)
KCON=IL
GO TO 50
45 CONTINUE
50 IF(KCUT.EQ.0.AND.KCON.EQ.0)GO TO 65
DO 60 J=4,N06
FACTOR=1.0
IF(J.GE.7.AND.J.LE.N06)FACTOR=SM(I,3)
SM(I,J)=SM(I,J)*FACTOR
TH=SM(I,J)
IF(SM(I,J).NE.0.0)TH=1.0-SMO(I,J)/SM(I,J)
IF(KCUT.EQ.0)GO TO 55
XIF(KCUT,J)=SM(I,J)-SMO(I,J)
THETA(KCUT,J)=TH
IF(ABS(TH).GT.TMAX(KCUT))TMAX(KCUT)=ABS(TH)
IF(KCON.EQ.0)GO TO 60
55 IF(ABS(TH).LE.DELS(J))GO TO 60
KONOK=KONOK+1
60 CONTINUE
65 CONTINUE

```

```

KFLAG=1
IF(IDEX6.NE.1)GO TO 75
IDEX2=IDEX3
KDEX=KDEX1
KRAM=KRAM1
75 IF(KONOK.NE.0)GO TO 80
KFLAG=2
GO TO 115
80 IF(IDEX2.GE.LOOPS)KFLAG=3
LOOP=IDEX2
IF(KFLAG.EQ.3)GO TO 115
IF(KDEX.LT.2)GO TO 115
DO 85 I=1,6
IF(NST(I).EQ.0)GO TO 85
KSB=NST(I)
DO 84 J=4,N06
XSC2(I,J)=0.0
IF(XSC1(I,J).NE.0.0)XSC2(I,J)=XIF(I,J)/XSC1(I,J)
84 CONTINUE
85 CONTINUE
IF(KRAM.EQ.1)GO TO 100
IF(KDEX.EQ.2)GO TO 115
DO 95 I=1,6
IF(NST(I).EQ.0)GO TO 95
KSB=NST(I)
DO 94 J=4,N06
IF(ABS(THETA(I,J)).LE.TMAX(I)/25.0)GO TO 94
IF(ABS(XSC2(I,J)).GE.1.0)GO TO 90
DL=(XSC2(I,J)-XSC3(I,J))/(1.0-XSC2(I,J))
DELL=(1.0-XSC2(I,J))/(1.0+XSC2(I,J))
IF(ABS(DL).LE.DELL)GO TO 94
90 KON=KON+1
94 CONTINUE
95 CONTINUE
IF(KON.NE.0)GO TO 115
IF(IDEX2.LE.4)KRAM=1
100 DO 110 I=1,6
IF(NST(I).EQ.0)GO TO 110
KSB=NST(I)
DO 105 J=4,N06
XL=XSC2(I,J)
IF(XSC2(I,J).GE.0.98)XL=0.9
IF(ABS(XSC2(I,J)).GE.1.0)XL=0.0
105 SM(KSB,J)=SM0(KSB,J)+(SM(KSB,J)-SM0(KSB,J))/(1.0-XL)
110 CONTINUE
KDEX=0
WRITE(61,200)IDEX2
200 FORMAT(1H0'*** FROM SUBROUTINE ASYM, ACCELERATION AT '
&' ITERATION'14' ***)
IF(IDEX6.NE.1)GO TO 115

```

```

      KRAM1=KRAM
      KDEX1=KDEX
115  DO 150 I=1,NSMAX
      KSB=0
      KCUT=0
      DO 120 IL=1,6
      IF(I.NE.NST(IL))GO TO 120
      KSB=NST(IL)
      KCUT=IL
      GO TO 125
120  CONTINUE
125  DO 130 IL=1,12
      IF(I.NE.KONCH(IL))GO TO 130
      KSB=KONCH(IL)
      GO TO 135
130  CONTINUE
135  IF(KSB.EQ.0)GO TO 150
      SM(KSB,3)=0.0
      DO 140 J=4,N06
      SMO(KSB,J)=SM(KSB,J)
      IF(J.GE.7)SM(KSB,3)=SM(KSB,3)+SM(KSB,J)
      IF(KCUT.EQ.0)GO TO 140
      XSC1(KCUT,J)=XIF(KCUT,J)
      XSC3(KCUT,J)=XSC2(KCUT,J)
140  CONTINUE
      IF(ABS(SM(KSB,3)).LT.1.E-9)GO TO 150
      DO 145 J=4,N06
      FACTOR=1.0
      IF(J.GE.7)FACTOR=SM(KSB,3)
145  SM(KSB,J)=SM(KSB,J)/FACTOR
150  CONTINUE
      IF(IDEX5.EQ.1)GO TO 160
155  CALL CHEKFLAG
      IF(KFLAG.EQ.2.OR.IDEX6.EQ.1)IDEX2=0
160  DO 165 I=1,6
165  NST(I)=0
      RETURN
      END
      DEFINE DCD
      COMMON/DATA/SM(75,25),EM(50,20),NOCOMP,NSLMAX,IWRITE
      COMMON/DATA/SI(6,25),S0(6,25),NIN,NOUT
      COMMON/DATA/PP(30,20),PPA(30,20),NRPP,NCPP,NRPPA,NCPA
      COMMON/DATA/CPNAM(40),DELS(25),ENAM(50),IDEX1,IDEX2,
*   IDEX3, IDEX4, IDEX5, IDEX6, IDEX7, IDEX8, IDEX9, IDEX10,
*   KARDS, KCLEAN, KFLAG, KPRINT, KRUN, KSOUT(40), KW(6,25),
*   LOOP, LOOPS, LUNOUT, MNST(6), NELMAX, NEMAX, NETIND, NOGO
      COMMON/DATA/NSMAX, NST(6), SNAM(75), TITLE(9), XIF(6,25),
*   XIG(6,25), XSC1(6,25), XSC2(6,25), XSC3(6,25), XXIF(6,25),
*   XXIG(6,25), XXSC1(6,25), XXSC2(6,25), XXSC3(6,25)
      END

```

```

SUBROUTINE XASYM
INCLUDE DCD
DIMENSION TMAX(6),XSMO(6,25),THETA(6,25)
N06=NOCOMP+6
IF(IDEX2.NE.1)GO TO 4
KDEX=0
KRAM=0
DO 3 I=1,6
IF(MNST(I).EQ.0)GO TO 3
KSB=MNST(I)
DO 2 J=1,N06
XSMO(I,J)=XXIG(I,J)
XXIF(I,J)=0.0
XXSC1(I,J)=0.0
XXSC2(I,J)=0.0
2 XXSC3(I,J)=0.0
3 CONTINUE
4 KDEX=KDEX+1
KON=0
DO 25 I=1,6
IF(MNST(I).EQ.0)GO TO 25
MSB=MNST(I)
DO 10 J=4,N06
FACTOR=1.0
IF(J.GE.7)FACTOR=SM(MSB,3)
10 SM(MSB,J)=SM(MSB,J)*FACTOR
TMAX(I)=0.0
DO 20 J=4,N06
THETA(I,J)=SM(MSB,J)
IF(SM(MSB,J).NE.0.0)THETA(I,J)=1.0-XSMO(I,J)/SM(MSB,J)
XXIF(I,J)=SM(MSB,J)-XSMO(I,J)
IF(ABS(THETA(I,J)).GT.TMAX(I))TMAX(I)=ABS(THETA(I,J))
20 CONTINUE
25 CONTINUE
IF(KDEX.LT.2)GO TO 70
DO 30 I=1,6
IF(MNST(I).EQ.0)GO TO 30
MSB=MNST(I)
DO 30 J=4,N06
XXSC2(I,J)=0.0
IF(XXSC1(I,J).NE.0.0)XXSC2(I,J)=XXIF(I,J)/XXSC1(I,J)
30 CONTINUE
IF(KRAM.EQ.1)GO TO 50
IF(KDEX.EQ.2)GO TO 70

```

```

DO 45 I=1,6
  IF(MNST(I).EQ.0)GO TO 45
  MSB=MNST(I)
  DO 40 J=4,N06
    IF(ABS(THETA(I,J)).LE.TMAX(I)/25.)GO TO 40
    IF(ABS(XXSC2(I,J)).GE.1.0)GO TO 35
    DL=(XXSC2(I,J)-XXSC3(I,J))/(1.0-XXSC2(I,J))
    DLL=(1.0-XXSC2(I,J))/(1.0+XXSC2(I,J))
    IF(ABS(DL).LE.DLL)GO TO 40
35  KON=KON+1
40  CONTINUE
45  CONTINUE
    IF(KON.NE.0)GO TO 70
50  DO 60 I=1,6
    IF(MNST(I).EQ.0)GO TO 60
    MSB=MNST(I)
    DO 55 J=4,N06
      XL=XXSC2(I,J)
      IF(XXSC2(I,J).GE.0.98)XL=0.90
      IF(ABS(XXSC2(I,J)).GE.1.0)XL=0.0
55  SM(MSB,J)=XSMO(I,J)+(SM(MSB,J)-XSMO(I,J))/(1.0-XL)
60  CONTINUE
      KDEX=0
      IF(IDEX2.LE.4)KRAM=1
70  DO 95 I=1,6
    IF(MNST(I).EQ.0)GO TO 95
    MSB=MNST(I)
    DO 75 J=1,N06
      XSMO(I,J)=SM(MSB,J)
      XXSC1(I,J)=XXIF(I,J)
75  XXSC3(I,J)=XXSC2(I,J)
      SM(MSB,3)=0.0
      DO 85 J=7,N06
85  SM(MSB,3)=SM(MSB,3)+SM(MSB,J)
      IF(ABS(SM(MSB,3)).LE.1.E-9)GO TO 95
      DO 90 J=4,N06
        FACTOR=1.0
        IF(J.GE.7)FACTOR=SM(MSB,3)
90  SM(MSB,J)=SM(MSB,J)/FACTOR
95  CONTINUE
      DO 100 I=1,6
100 MNST(I)=0
      RETURN
      END

```


APPENDIX C

Nomenclature

D	Deviation of a variable from its solution.
f	The input to output functional relationship for a single process unit.
F	The functional relationship describing an iterative problem.
i	Subscript indicating a particular variable.
I	Iteration number or identity matrix.
j	The individual terms of the Jacobian matrix.
J	Jacobian matrix defined by Equation 2.17.
P	Matrix.
q	Convergence parameter.
Q	Matrix.
S	Sensitivity matrix for a unit in a single loop recycle system.
S'	Sensitivity matrix for a unit in a multiple loop recycle system.
tr	The trace of a matrix.
x_i	Recycle variable.
x_0	Initial value of a recycle variable.
x_s	Solution value of a recycle variable.
x_∞	Asymptotic or accelerated value of a recycle variable.
δ_c	Desired accuracy for deviation from solution in a recycle problem.

- δ_K Deviation from solution at iteration K.
- $\delta\lambda$ Parameter for acceleration criterion, defined by Equation 3.68.
- $\Delta\lambda$ Parameter for acceleration criterion, defined by Equation 3.66.
- $\bar{\theta}_I$ Test vector for determining convergence of a recycle problem.
- λ_1 The largest magnitude of the eigenvalues of the Jacobian matrix.
- λ_{crit} The critical eigenvalue for acceleration, defined by Equation 3.64.
- λ_{iI} The apparent eigenvalue of variable i at iteration I.
- λ_{sI} The value of the eigenvalue which would give the exact solution if used in the acceleration equation, defined by Equation 3.44.
- ϕ Correction factor for convergence testing.