IMPROVEMENT OF CONVERGENCE IN THE SIMULATION OF COMPLEX CHEMICAL PLANTS.

By

STEPHEN MICHAEL LORD, B.Sc.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

September 1970.

IMPROVEMENT OF CONVERGENCE IN THE SIMULATION OF COMPLEX CHEMICAL PLANTS

MASTER OF ENGINEERING (1970) McMASTER UNIVERSITY,

(Chemical Engineering) Hamilton, Ontario.

TITLE: Improvement of Convergence in the Simulation of Complex Chemical Plants.

AUTHOR: Stephen M. Lord,
B.Sc. (Aston University, Birmingham, England)

SUPERVISOR: Professor C.M. Crowe.

No. OF PAGES: viii,116.

SCOPE AND CONTENTS:

The various criteria and methods suggested in the literature for choosing the optimum sequence of calculation are compared. They are then analysed using the results of runs on three different simulations. Two techniques of convergence are used, and it is found that Geometric Convergence Promotion will improve the rate of convergence compared to Direct Substitution for any feasible sequence. It is also shown that the optimal sequences of calculation are different for the two techniques.

It is concluded that the Minimum Cut Streams criterion is the best to use for Direct Substitution, but that the use of the Feasible Set criterion and Convergence Promotion will yield better convergence. The iterative improvement of convergence is discussed, and it is concluded that the available time is best spent on improving the starting point and the Convergence Promotion technique rather than on changing the sequence of calculation.

ACKNOWLEDGEMENTS.

I should like to express my gratitude to my research director, Dr. C.M. Crowe, for his guidance and assistance.

I should also like to express my appreciation to all the people who have worked on the Alkylation Plant and Bayer Process Simulations, which form the basis of this thesis, and to Dr. O. Orbach for the use of his geometric convergence promotion routine and for his general assistance.

I must also thank my typist, Miss K. Thornton, who, together with O. Owsley, helped to sustain my dedication.

I am also indebted to McMaster University for their financial and technical support.

TABLE OF CONTENTS.

INDEX OF TABLES

INDEX OF FIGURES.

CHAPTER ONE.

DISCUSSION OF THE CRITERIA AND METHODS USED IN THE SELECTION OF THE SEQUENCES OF CALCULATION.

1.1 Introduction.

Chemical Plant Simulation is the use of mathematical techniques to model, or simulate, the behaviour of the chosen chemical plant. The usual purpose of the simulation is to evaluate the effects of one or more proposed plant improvements, and to provide an on-going basis for further plant improvement in an existing, usually complex, plant. It is a growing field of interest in Canada and Canadian universities, where the number of faculty engaged in this field increased from nine to fifteen in the past year.* Industry has also taken a great interest, and several industry-university joint simulations have been undertaken.** This interest has been greatly aided by the development and use of generalised process simulators (C2, EI, LJ, RI, FI).

One of these generalised process simulator programs is Shannon's PACER (C2, M2, S2) which has been used at McMaster University both in its original form, and in a modified form called MACSIM (McMaster Simulation). Plants which have been simulated using PACER or MACSIM include a sulphuric acid plant, an alkylation plant, and the Bayer Process for alumina extraction. The last two simulations form the basis for this thesis.

---

* Chemistry in Canada. June 1970. Page 22.

** The Chem. Eng. Department at McMaster University has been very active in this field with several successful simulations to its credit.

The underlying principle of PACER, MACSIM, and most other simulator programs is modularity. Modularity consists in describing the plant in terms of modules, or building blocks, which can be connected to one another so that it would be possible to simulate different plants with the same modules. An actual chemical plant has reactors, heat exchangers, pumps and distillation columns, separators, compressors, etc., connected together in some way. The performance of each piece of equipment, or unit, can be described by a set of equations based either on the fundamental laws of physics and chemistry, or on empirical data. These equations can then be solved in order to simulate the unit. The computer program to solve these equations is defined as a unit computation.

Each unit computation calculates the output streams from the unit, given all input streams and any other necessary parameters.

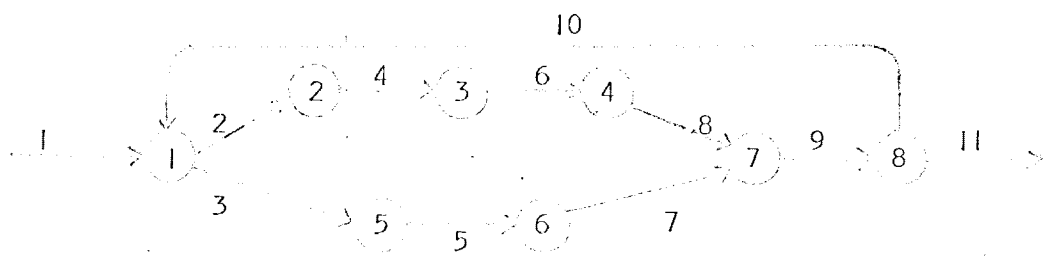
The process simulator program, sometimes calle the executive program, controls the sequence of calling the unit computations to effect the simulation of the chemical plant. In order to simulate the entire plant, every unit computation must be calculated. In a given plant, every sequence may have a unit whose input streams are the output streams of some other unit, which has still to be calculated; this is then a plant with recycle and must be calculated iteratively.

One can see that if there is no recycle, one can start at the feed streams and calculate directly every unit in the plant so that the plant can be simulated. However most chemical plants involve recycle. This means that the recycle stream is required as an input before it has been calculated as an output. This situation is usually solved by cutting, or assuming, a sufficient number of streams, called cut, recycle, or assumed

streams. By using the downstream half as an input, and the other half as an output, the two halves are matched by some iterative technique.

The values of the variables in a cut stream must be initially provided for the downstream half in order to calculate that unit which this stream enters. In addition, in order for the executive to simulate the plant, it must be provided with a sequence of calculation of the unit computations. Each sequence of calculation implies its own set of cut or assumed streams. Although each set of cut streams may imply more than one sequence of calculation, on examination they can be shown to be the same. Cases where more than one sequence is apparently possible are usually due to parts of the plant being in parallel. For example: -

Fig. 1.1 A Recycle Process with a Parallel Sequence of Calculation.



If we choose stream 10 as a cut stream then we can have the following sequences of calculation: -

1 (2, 3, 4) (5, 6) 7, 8

and

1 (5, 6) (2, 3, 4) 7, 8

These sequences are really identical because the same information is used in an exactly similar way, and the results must be the same. It is therefore the different sets of assumed streams which distinguish sequences of calculation, and this is the basis of the work done in this thesis.

1.2 Information Flow and Sequence of Calculation.

Information Flow and Calculation Sequence are closely linked. The concept of information flow was first used by Steward (S4). He dealt with the problem of solving a large sparse matrix equation, or a set of simultaneous equations, without having to invert the matrix. He defined a matrix to be sparse where: -

> " The number of variables in any equation is small compared to the total number of variables in the system."

His approach was to solve the equation in such a sequence that the variables found previously will be sufficient to solve the next equation.

He also defined an 'output set' which is so constructed that: -

(1) Each equation has exactly one output variable.

(2) Each variable appears as the output variable of exactly one equation.

From each solved equation there 'flows' a variable which is then used in solving the next equation. Steward looked on this as 'information' which 'flows' through the matrix being modified by each equation. This process he termed 'information flow.' He then illustrated this by an information flow diagram.

If one takes the following equations: -

(1) $f_1 (a, c) = 0$

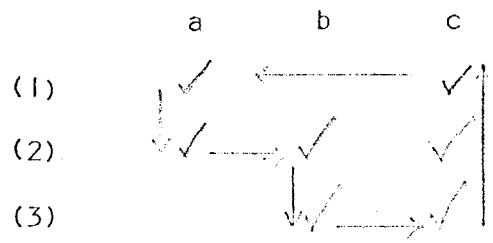
(2) $f_2 (a, b, c) = 0$

(3) $f_3 (b, c) = 0$

one can generate an information flow diagram by a matrix composed of equations and variables where the presence of a variable in an equation is marked by a tick in the row of the equation under the respective

variable. The information flow diagram derived from the above equations is

Fig. 1.2 Information Flow Diagram.



By following the arrows, one can see the 'information flow' through the system of equations.

For example, at equation (1) one could assume c; then one could solve equation (1) to give a: with a and c one could solve equation (2) to get b: with b one can solve equation (3) to get c. The sequence in which the equations were solved is the 'calculation sequence' for this set of equations. If the equations are represented by numbers, and the variable solved in the preceding equation is represented in brackets, then the sequence of calculation is 1(a) 2(b) 3(c). If the value of c found equals the assumed value of c, then the system is solved; if it does not, then another value of c must be assumed. Usually the latest value of c found is used as the new assumed value. This is termed 'Direct Substitution.' However, this is not necessary, and any value could be substituted. The use of values other than the latest one calculated is intended to decrease the number of iterations taken to converge, and is called 'Convergence Promotion.'

Usually these techniques take advantage of some progression in the values of the variable. For instance, the sequence of numbers may involve geometric progression as in

$1,\ 1\frac{1}{2},\ 1\frac{3}{4},\ 1\frac{7}{8},\ \ldots.$

This sequence contains a geometric progression of which the limit is zero and itself converges to two. If by direct substitution, a sequence results which involves a converging geometrical progression, then one can use this information to predict the limit after an infinity of iterations. This limit will then be the apparent 'true' value, and would be an improved guess for restarting the calculation. By shortening the number of iterations required, one would have promoted the convergence.

The simulation of a chemical plant involves writing a large number of equations, each with few variables relative to the total number. This obviously corresponds to the sparse matrix of Steward. Thus it is not surprising that the analogy between information flow through a system of equations and of material flow through the units of a chemical plant, has been utilised, especially in modular approaches such as PACER or MACSIM. In such modular approaches, the information produced by one unit computation module must be communicated to the correct module. This is very important since the simulation of a particular plant depends on the way its modules are connected. Each connection between the modules is termed a stream and is identified by its own particular number. Each module is also given its peculiar number, and the streams which are needed as inputs, and those which are produced as outputs, are defined. Therefore in PACER and MACSIM the information flow diagram shows the streams and the units. Because of the above analogy, the information flow diagram is usually derived from the material flowsheet. However, unlike the equations in Steward's paper (S4) which can have only one output, and need all the other variables in the equation as inputs in order to be solved, the models may have more than one output or input.

The information communicated by the streams consists of several stream variables which together form the stream vector - a further deviation from the above paper where the variable considered as the output or input is scalar.

A consequence of the derivation of the information flow diagram from the material flowsheet is that the direction of the information flow is usually taken to be the same as that of the material flow. This leads to the design of unit computations in which the information flow is in the same direction as the material flow in the corresponding unit of the chemical plant. Once the unit computations have been designed in this way, it becomes difficult to change the direction of the information flow without modifying the models. Therefore the information flow diagram will have been fixed before anyone has considered which might be the easiest information flow diagram to calculate.

That the information flow diagram so fixed may be less than the optimal one, and so will produce a worse calculation sequence, is easy to see. If the simulation could be calculated more quickly by a reversal of the assumed information flow between two units, then the information flow diagram is not optimal.

In fact, the more the information flow is determined prior to the choice of the calculation sequence, the more likely it is that the calculation sequence found will not be the optimal. This can be shown more generally as follows:-

Let S be a set of equations, and V a set of variables which taken together can be used to describe the chemical plant. Let $M = \{M1, M2, \ldots\}$ be the partition of S into subsets of S such that each subset describes a

module. The object is to find the sequences of calculation, C_s and C_m for S which will minimize some criterion such as total computation time T, (where C_s and C_m are the sequences of calculation of the unpartitioned set S and of the partition M of set S respectively, and T_s and T_m their respective total computing times). It can be seen that generally

$$T_s \leq T_m$$

because the partition M of S imposes a constraint on C_m which is not imposed on C_s. If a further constraint is imposed by fixing the direction of calculation within each module M1, M2, ..., then it follows that

$$T_m \leq T_f$$

where T_f is the total computing time for the optimal sequence when the direction of calculation within each module is fixed.

1.3 Methods of Finding the Calculation Sequence.

There are several ways of finding the calculation sequence: -

(1) Direct Method - PACER.

The method used by PACER is direct, but it is also very time-consuming. The executive scans the simulation to see if it can be calculated directly from the feed streams. If not, there are some unknown streams, and there is recycle. Then it will assume each unknown stream in turn to be known, and will see if all the models can then be calculated. If no one stream is sufficient, then it will assume all possible pairs of assumed streams in turn. If no pair is sufficient, then it will assume all possible combinations of three streams.

If this does not solve the problem, then PACER cannot find a calculation sequence. If it succeeds, it has found the minimum number of cut streams that will solve the simulation. As the number of possible assumed streams increases, the time taken to find the calculation sequence increases geometrically.

(2) Digraph and Adjacency Matrix Techniques.

One can show that an information flow diagram, such as the one used in MACSIM, can be represented by a directed graph, or digraph. (See overleaf for an example of a digraph).

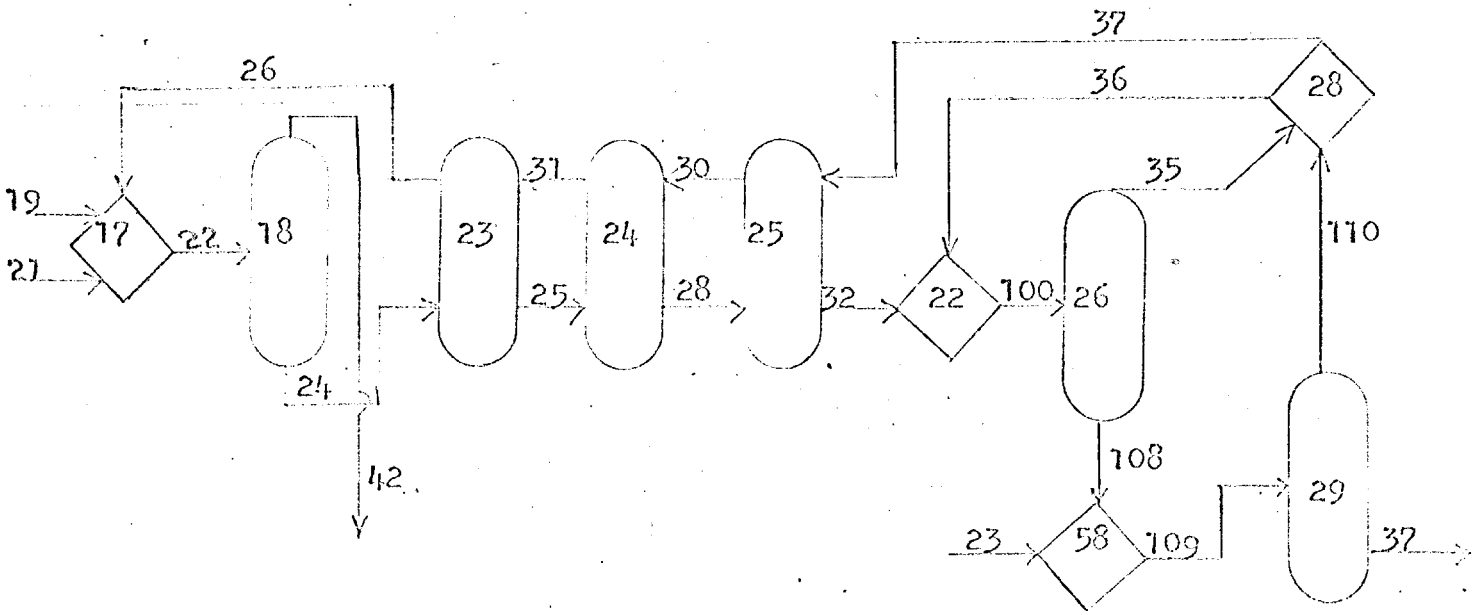
One of the properties of a digraph is the fact that it can be represented simply by an adjacency matrix.

The adjacency matrix of a labelled digraph, D, is defined by Harary, (HI): -

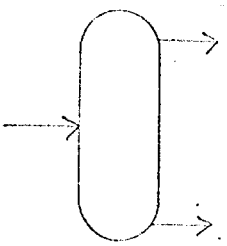
$\bar{\bar{A}} = A(D) = [a_{ij}]$ where $a_{ij} = 1$ if arc V_iV_j is in D, where V_i is a node of D, and $a_{ij} = 0$ otherwise.

i.e. $a_{ij} = 1$ if V_i is connected to V_j, and $a_{ij} = 0$ otherwise.

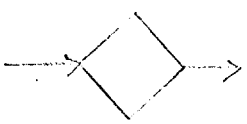
Fig. 1.3.1 Diagram and Digraph of the BAYER Hydrate Wash Section.



SEPA02 MODEL.



JUNC01 MODEL.



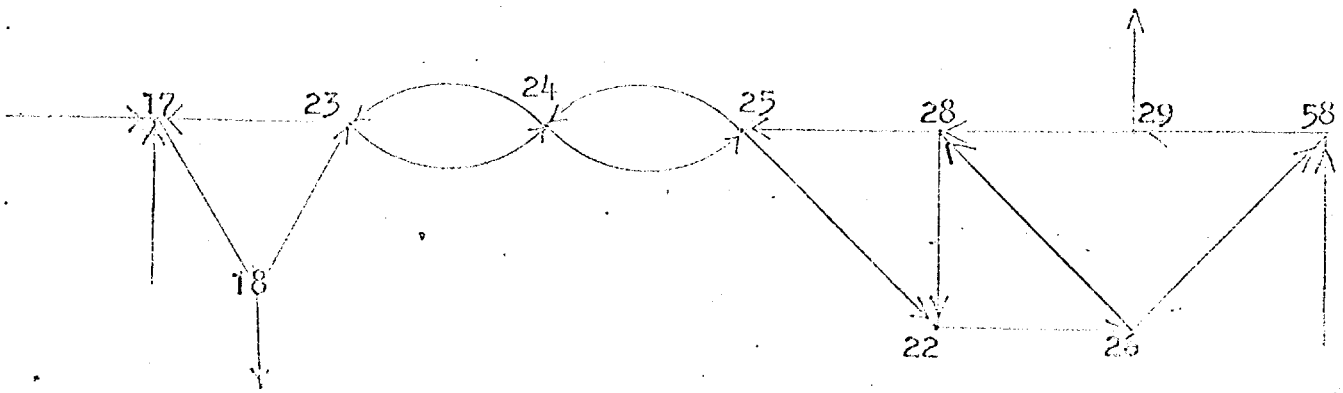
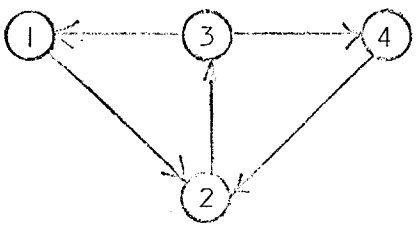


Fig. 1.3.2 Example of a Digraph and its corresponding Adjacency Matrix.

Digraph D.



Adjacency Matrix A.

	(1)	(2)	(3)	(4)
(1)	0	1	0	0
(2)	0	0	1	0
(3)	1	0	0	1
(4)	0	1	0	0

This technique is used by Steward (S4), Norman (N1), Himmelblau and Bischoff (H2), Harary (H1), and Dulmage and Mendelsohn (D1).

The advantage of the adjacency matrix lies in the use of powers of the matrix. If one raises matrix A to its n^{th} Boolean power, it will show as 1's all paths of length n. This has been used to trace the paths of length 1, 2, 3, 4, etc., in the information flow diagram by Norman. The other references do essentially the same thing but use more sophisticated techniques for generating the matrices. A very fast technique for partitioning into recycle blocks is using the powers of [I+A] where [I] is an identity matrix of the same size as A. To find all paths of length $\leq 2^n$, one raises [I+A] by continued squaring to $[I+A]^{2^n}$ and then treats it as in the above techniques.

1.3.3 List Processing.

This general heading refers to the use of list-type representation of the information flow diagram in the partitioning of the simulation. The approach is to trace a path or 'chain' backwards, as in Sargent and Westerberg (S1), or from both ends, as in Christensen (C1), along the streams or 'links' connecting the units, until a 'loop' is

found. Sargent and Westerberg then merge the units of this loop into a single unit which they term a 'logically defined block.' This technique is then continued until all loops are partitioned into blocks.

Christensen detects and merges the cycles in a similar way to Sargent and Westerberg. However, he also deletes any 'node' with no input 'edges.' This combination yields his first algorithm, IR.

Both Christensen and Sargent and Westerberg reduce the number of combinations by merging edges and nodes. Christensen also eliminates 'ineligible edges' based on the respective number of input parameters to the various units or nodes. For example, if node A has W_{A+} input parameters, and node B has W_{B-} total output parameters, then edge (AB) is ineligible and need never be considered as a recycle stream if

$W_{AB} \geq W_{A+}$ where W_{AB} is the total number of parameters between A and B.

and

$W_{AB} \geq W_{B-}$

It should be noted that equality in fact does not disqualify edge AB and that there is then no way of distinguishing which of the equal edges is the better choice. This then, is a problem of eliminating valid choices, and it is common to other techniques too, for example the cycle and adjacency matrix techniques.

1.3.4 Use of Cycle Matrix and Spanning Tree.

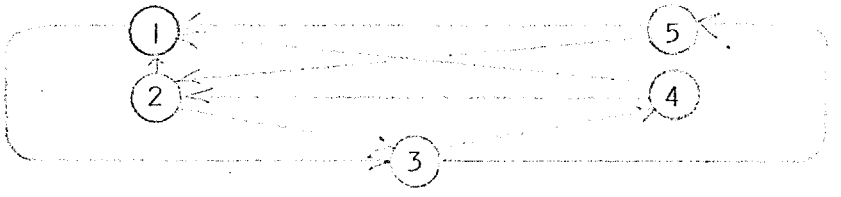
The cycle matrix is defined as the matrix $\bar{\bar{C}}$ with elements c_{ij} where : -

$c_{ij} = 1$ if stream j appears in cycle i.

$c_{ij} = 0$ otherwise.

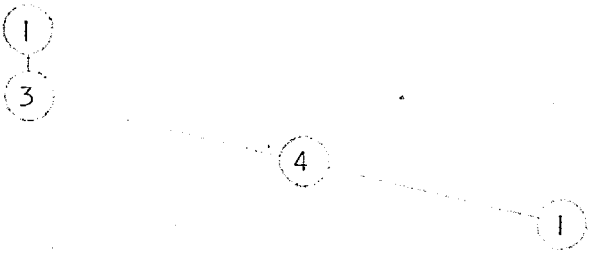
A very straightforward and convenient way of tracing the cycles is one originally proposed by Steward (S4), and used as the basis of an algorithm and computer program by Longeway (L3), which makes use of the 'Spanning Tree.' Consider Longeway's example which he takes from Rubin (R3) : -

Fig. 1.3.3 Digraph of Rubin.



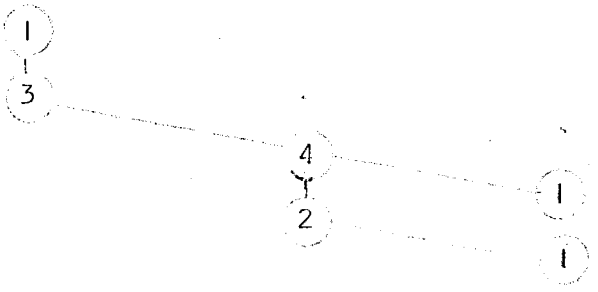
One can start, arbitrarily, at node 1 and proceed round the digraph generating the spanning tree. As soon as a node is reached which has already appeared, then a cycle has been located and this branch is terminated: -

Fig. 1.3.4 Spanning Tree of the above digraph. (1st. cycle).



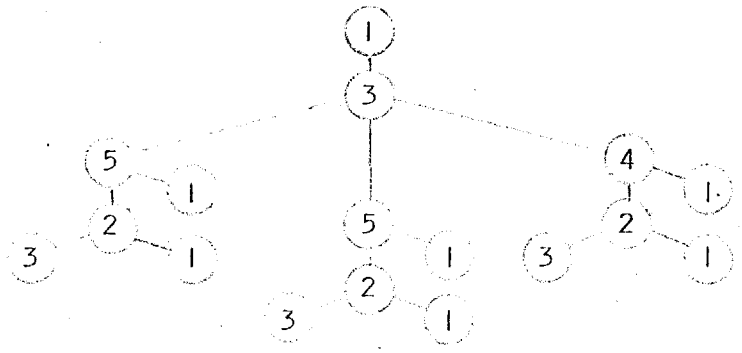
Then one returns to node 4 and looks for an output. This branch is then terminated: -

Fig. 1.3.5 Spanning Tree of the above digraph (1st. and 2nd. cycles).



This process is repeated until the full tree is generated, which is for Rubin's example: -

Fig. 1.3.6 Full Spanning Tree.



Each branch is a potential loop although there may be duplicates.

Beginning at the end of each branch, each cycle can be traced, stopping wherever a node is re-encountered. Once the nodal sequence of each cycle is found, then the corresponding streams can be found, and the cycle matrix generated.

This technique*has been used on the Bayer Wash and Crystalliser Sections in Chapter Four. It was not used on the Alkylation Plant because, although a large plant, it is relatively simple, and the required cycles could be obtained by inspection of the plant digraph. In large complex plants it would be more cumbersome to use, but this could be reduced by the following techniques. Any purely sequential units could be grouped together, and the tree could be split up in a similar manner to the splitting up of a large computer program.

An alternative approach is to use the powers of the adjacency matrix A (Harary, HI) to generate the cycle matrix.

---

* N.B. Longeway, L3, has written a computer program to solve spanning tree problems, but it was not necessary to use it.

1.4 Sequence Within Recycle Set.

1.4.1 Criteria for Optimum Calculation Sequence.

In the previous section it was shown how the various authors have partitioned the original incidence matrix, or digraph, which describes the whole simulation. They use different terminology for the same cyclic part, Harary calls it a 'strongly connected subgraph' or strong component; Sargent and Westerberg 'a logically defined block'; Christensen detects 'loops'; and Longeway traces 'cycles.' The terminology used within MACSIM is 'recycle set', and this, therefore, will be the term used henceforth.

Once the recycle sets have been found, there still remains the problem of solving each of them. To do this, the calculation sequence within the recycle set must be found. By definition of a recycle set, some stream or streams must be assumed to calculate it.

Within any recycle set there are several possible calculation sequences and sets of assumed streams. One therefore needs some criteria for deciding which one to use. Most of the above-mentioned writers choose as their criteria minimum computing time. Christensen seeks "the sequence (U) of unit computations within each such block so as to minimise the time T_c, required to obtain convergence for the block." This time, T_c, is given as

$$T_c = N_i \left(T_u + N_p T_p \right)$$

where N_i is the total number of iterations required to reach convergence, T_u is the time required to update all the recycle parameters at each iteration, N_p is the number of times that 'once through' computation of the units in the block must be repeated at each iteration, and T_p

is the time required for one such computation.

Longeway also has the same approach: -

> " The ultimate goal in the selection of recycle streams is to select those streams such that the total computation time required to converge to the steady state is as small as possible."

Both authors, then, are faced with the problem of finding an a priori criterion. Christensen developed his argument by maintaining that only T_p is independent of the calculation sequence, (U), and that N_i, though affected by (U), cannot be predicted a priori. This leaves him with the factors T_u and N_p, and he continues: -

> "for any iterative method, the factors T_u and N_p will be non-decreasing functions of a single parameter of U: - the number, k, of recycle parameters."

Christensen notes the constraint that convergence must be attainable, but the difficulty in his argument is that N_i is dependent on (U). Longeway comes to the conclusion that the weighted sum of parameters for the recycle streams must be minimised, and recommends that the weighting parameter, p_j, of each stream j, should depend on the convergence technique. This is in contrast to Christensen's above statement, and, it should also be noted, no proof of this is given by Longeway. The assertion was only tested on cases with a single recycle strem. The recommended parameters are: -

(a) Direct Substitution.

Each parameter, p_j, should represent the number of cycles in which that stream is involved.

(b) Wegstein's Method.

Each parameter, p_j, equals unity.

(c) Newton-Raphson's Method.

Each parameter, p_j, should equal the number of variables in the corresponding stream.

The third case is identical to Christensen's choice of k, the number of recycle parameters, although Christensen does not suggest using a Newton-Raphson technique.

Sargent and Westerberg (S1) allow for minimising an arbitrary weighting of the streams, and choose as their weighting factor the number of parameters in each stream.

1.4.2 Methods of Finding the Calculation Sequence within a Recycle Set.

The direct approach of PACER, or of Koppel et al. (K1) which is very similar to that of PACER, will find the calculation sequence within a recycle set, but this takes a long time if there are many choices of assumed streams. Lee and Rudd (L2), assume the cycle matrix already exists, and then use two distinct algorithms, one for minimising the number of cut streams, and one for minimising the number of cut parameters.

That these are both sub-problems of a larger problem is pointed out by Longeway (L3). Their algorithms effectively try all possible combinations. The number of possible combinations is reduced, however, by the use of two concepts.

These same concepts are also used by Longeway (L3), and are defined as: -

(a) Dependent Streams.

> "If stream α is a member of all cycles in which stream β appears, and if $p_\alpha \leq p_\beta$, then stream β is said to be dependent upon stream α.
> Two streams will be said to be independent if they are not dependent."

It must be pointed out that there are, in fact, three possibilities: -

(1) $p_\alpha < p_\beta$ which should be termed strictly dependent.

(2) $p_\alpha = p_\beta$ which should be termed equivalent.

(3) All other cases, which should be termed independent.

(b) Dependent Cycles.

> "If for every method of tearing cycle α, cycle β will thereby also be torn, then cycle β is said to be dependent upon cycle α.
> Two cycles will be said to be independent if they are not dependent."

Similarly there should also be three cases here: -

(1) If, for every method of tearing cycle α, cycle β will thereby also be torn, and the reverse is not true, then cycle β is strictly dependent on cycle α.

(2) If the reverse is true, then the cycles α and β are equivalent.

(3) If neither strictly dependent, nor equivalent, then they are independent.

Again this is the problem met earlier in discussing List Processing (1.3.3) of eliminating valid choices.

Longeway uses a zero-one linear programming technique to change the original cycle matrix into one in which every cycle contains only one stream. However, the use of the two above definitions greatly simplifies the procedure. The dependent streams and cycles are deleted in sequence until there are none remaining. The resulting, greatly simplified, matrix is then treated by the linear programming technique until a set of cut streams with minimum parameters is derived. If valid choices are not eliminated then, of course, he would probably derive more

than one set of cut streams with minimum parameters.

Sargent and Westerberg use dynamic programming to solve the calculation sequence within the recycle set, or 'logically defined set.' Unfortunately they do not show any examples. Besides which, dynamic programming usually uses a large amount of computer storage, yet they reject matrix representational techniques on the grounds of excessive storage.

Christensen uses his algorithm, IIR, for edge and line merging with the concept 'node indexing' to solve several problems. An index node is defined as: -

> "one [node] all of whose inputs or outputs or both, are eligible to be recycle streams."

He therefore selects an index node, and then deletes the streams connected to it. The resultant sub-graph is treated by algorithm IIR. He claims optimal or near optimal results, but as the number of possible index nodes increases, the number of combinations of nodes increases geometrically.

He therefore describes algorithms IT and IIT, which 'tear' selected variables. In this approach, though, he goes away from the modular situation and back to considering individual equations. He uses substitution and elimination as an aid, then uses his algorithms to select the 'torn' variables.

This approach requires all but one of the variables of an equation as input, and only one as output. Thus it is not directly relevent to the modular situation.

1.5 Discussion of Minimum Cut Streams as a Criterion for Optimal Calculation Sequence.

All the papers which have dealt with finding the calculation sequence (CI, L3, SI, S2, C2, FI) have made the assumtion that that calculation sequence which results in a minimum set of weighted parameters of the assumed streams, is the calculation sequence of the equipment within the block which minimises the calculation time. Several authors have, in fact, noted that this approach might not lead to minimum computing time. As Orbach (OI) states: -

> "However, there is neither a numerical nor a theoretical proof of this assumption."

Christensen (CI) believes that minimum cut streams is a criterion related only to the time taken for one iteration, and not to the number of iterations needed for convergence. Sargent and Westerberg (SI) look to the possibility of some method of improving the initial calculation sequence, thereby implying that the initial is not optimal.

Forder (FI) states that the process designer may prefer certain streams to be cut, and even if this leads to an increase in the number of cut streams, it may still lead to faster convergence. Orbach (OI), also shows: -

> "that the same sequence can result from another set of assumed streams, larger than, or equal to, the minimum set, by simply starting the calculation from another equipment in the sequence, and therefore the rate of convergence is the same."

This can be illustrated by a case given by Lee and Rudd (L2): -

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

Case B is the one selected by Lee and Rudd because it has the minimum set of assumed streams, but, in fact, all cases have the same calculation sequence with different starting points.

Is the criterion of Minimum Cut Streams necessary and sufficient?

From the previous argument it has been shown that the criterion is not necessary because one can obtain the best calculation sequence without obtaining the minimum set of streams.

It is also possible to show that it is not sufficient by illustrating a case which has more than one minimum set. The example shown below is from a selection of the BAYER Process Simulation.*

Case	Assumed Streams	Sequence of Calculation
A	100, 22, 25, 30	26, 58, 29, 28, 18, 24, 25, 23, 17, 22.
B	100, 26, 30, 31	26, 58, 29, 28, 17, 18, 23, 24, 25, 22.
C	100, 28, 25, 23	26, 58, 29, 28, 25, 24, 23, 17, 18, 22.
D	100, 22, 30, 31	26, 58, 29, 28, 18, 23, 24, 25, 17, 22.

There are, in fact, twelve possible sets of four assumed streams which may be represented as below: -

---

* A diagram and digraph of this section are shown on p. 10.

$$\begin{matrix} & & (31, 30) & & \\ (26) & & (\quad) & & \\ (\quad) & & (25, 28) & & \\ (22) & + & (\quad) & + & (100) \\ (\quad) & & (25, 30) & & \\ (24) & & (\quad) & & \\ & & (31, 28) & & \end{matrix}$$

The sets may be constructed by selecting any stream from the first bracket, plus any pair from the second, together with stream 100 in the last bracket.

Is there an Empirical Justification for the Minimum Cut Streams Criterion?

Although there is not a numerical, nor a theoretical, justification for the above criterion, the possibility exists of an empirical justification of the assumption.

The number of papers, algorithms, and computer programs on this area of research has pointed to the importance of finding a satisfactory calculation order. Since the minimum cut streams criterion has been the basis of all the approaches up to the present day, it seems important to determine whether or not it is justified.

All the approaches are essentially search techniques, and as a result, as the size of simulations increases, the number of necessary searches will be proportional to the number of possible permutations which increases geometrically with size. This reasoning also includes matrix techniques: a 20 x 20 matrix is four times the size of a 10 x 10 matrix.

Therefore, if the minimum cut streans criterion is not justified, it would be desirable to select another criterion which would be easier to apply.

Orbach's definition of a feasible set could point to an appropriate substitute: -

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

The adoption of this feasible set criterion would greatly shorten computation time and decrease storage space. Still more important, as the algorithm would not require searches, the time taken by the computer would only increase linearly with the size of the problem, and not geometrically.

CHAPTER TWO.

DESCRIPTION OF THE SIMULATIONS AND CONVERGENCE TECHNIQUES USED.

2.1 Aims of Work.

The aims are to test the suggested a priori criteria* for obtaining the optimum sequence of calculation by comparing the times taken to converge the same simulation from the same starting point for different sequences of calculation. Three different simulations will be used, and three different starting points will be used on one of the simulations.**

In addition, the effect of the use of convergence promotion by geometric extrapolation as derived by Orbach (01) will be investigated. The aims in this area are to see if the optimum sequence of calculation is the same for both Direct Substitution and Convergence Promotion, and to compare the relative improvement in computer time achieved by finding the optimal sequence compared to that obtained by using a Convergence Promotion technique.

2.2 The Simulations Used.

It was only possible to undertake this work because several chemical plant simulations had already been developed at McMaster University. All of these simulations have been based on large amounts of data collected from the plants being simulated, and have been extensively verified. The

---

*Pp. 16 and 22 give the four criteria

** The BAYER Wash Section.

results for the 'base case' simulation matched the plant data very well, so that these 'base case' simulations have been used for most of the work.

The programs were written using the MACSIM executive program, the models themselves are written in FORTRAN II, and the whole system used the McMaster I.B.M. 7040. The programs are being transferred to the more-recently acquired C.D.C. 6400, but the transfer was not completed at the time of research.

2.3 Description of the Hydrate Wash Section in the BAYER Alumina Plant Simulation.

The simulation of the BAYER Alumina Extraction Plant was done as a graduate study in 1967-68, and as a 4th. year project in 1968-69. The convergence characteristics of this plant were studied in 1968-69 by O. Orbach (01). The purpose of the Hydrate Wash section is to wash soluble impurities from alumina hydrate, and then to filter off the excess moisture before the hydrate is calcined in the kilns to give alumina, the desired product.

2.3.1 Description of Plant Functions.*

The feed streams** 19 and 21 join the recycle stream 26 at Equipment 17, and enter, as stream 22, the primary classifier (Equipment 18) where a split in the solids occurs. The underflow stream (24) enters the three-stage counter-current washer (Equipments 23, 24, 25). After washing, the concentrated hydrate is filtered out (Equipment 26), washed again (Equipments 29 and 58), and then sent (as Stream 34) to a rotary kiln (not shown) where it is dehydrated and calcined to give the desired product of alumina.. The feed stream 23 is pure water only.

The overflow from the primary classifier (stream 42) passes to the Crystalliser Section (see 2.4).

---

* The diagram and digraph for this plant are on p. 10

** The composition of feed streams 19 and 21 and 23 are given in tables 2.3(a) and 2.3(b).

The three-stage counter-current washer uses a SEPA02 model for each stage; this model allows a few percent of the solids to be carried over in the overflow stream as happens in the plant itself. The filter is a rotary drum type (Equipment 26) and is also represented by a SEPA02 model, which reduces the moisture in the solid to 15% by weight, and returns the liquor (stream 35) to the counter-current wash.

The concentrated hydrate (stream 108) is washed with clean water (stream 23) in Equipment 58, then filtered on another rotary drum filter (Equipment 29), also represented by a SEPA02 model. This liquor is also returned to the counter-current wash section.

TABLE 2.3(a)

Feed Streams to the Hydrate Wash Section.

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

TABLE 2.3(b)

BAYER Process Stream List.

Component Number	Location in Stream Vector	The Component Flow lb.mole/hr		Molecular Weight
1	6	WATER		18.020
2	7	Na_2CO_3		105.993
3	8	NaOH (as Free Caustic)		40.001
4	9	$3(Na_2O.Al_2O_3.2SiO_2.0.6Na_2O.$ $.Al_2O_3.0.22Na_2SO_4.0.23Na_2CO_3)$		1006.3602
5	10	$Al_2O_3.3H_2O$	(SOLID)	156.020
6	11	$Al_2O_3.2SiO_2.2H_2O$	(SOLID)	258.180
7	12	$Fe_2O_3.H_2O$	(SOLID)	177.720
8	13	P_2O_5	(SOLID)	141.950
9	14	CaO	(SOLID)	56.080
10	15	TiO_2	(SOLID)	79.900
11	16	$CaCO_3$	(SOLID)	100.091
12	17	Carbon Compounds	(SOLID)	44.011
13	18	Unknown inert solid	(SOLID)	119.980
14	19	$NaAlO_2$	(DISSOLVED)	81.971
15	20	$Al_2O_3.2SiO_2.2H_2O$	(DISSOLVED)	258.180
16	21	Fe_2O_3	(DISSOLVED)	159.700
17	22	P_2O_5	(DISSOLVED)	141.950
18	23	$(3CaO.Al_2O_3).SiO_2$	(DISSOLVED)	600.490
19	24	TiO_2	(DISSOLVED)	79.900
20	25	Na_2SO_4	(DISSOLVED)	142.048
21	26	Carbon Compounds	(DISSOLVED)	44.011
22	27	$Na_2C_2O_4$	DISSOLVED)	134.000

2.3.2 The Equipment Modules.

The SEPA02 module calculates the split of one or more feeds according to parameters stored in the equipment parameter (EN) vector, which specify the weight fraction of the inlet solids that pass in the overflow, and the weight fraction of solids in the underflow stream. It is assumed that the particle size distribution is constant, and that the only split occurs between the solid and the liquid components, (dissolved components are considered as part of the liquid flow).

Therefore the program calculates the weight of solids (lb/hr) and the weight of liquid (lb/hr) in the feed from the known molar flowrates and respective molecular weights (Table 2.3(b)). The weight of solids in the feed is then multiplied by the above specified fraction to give the weight of solids in the overflow. The underflow of solids can then be found by subtraction. Knowing the above specified weight fraction of solids in the underflow, and the weight of the solids in the underflow, the total weight of the underflow may be found, and hence, by difference, the weight of liquid in the underflow. The weight of the liquid passing overflow can be calculated as the difference between the weight of the input liquid and the liquid passing overflow.

The other models (17, 22, 28, 58) are JUNC01 modules. This is a simple mixer-splitter module which combines all the inputs, then splits them into the output streams according to the splits specified in the equipment, EN, vector.

2.4 Secondary Crystalliser Section.

The overflow (stream 42) from the primary classifier (Equipment 18 in the Hydrate Wash Section) serves as feed for this whole section. The feed enters the secondary classifier (Equipment 45) where it is split into two streams; the underflow (stream 67) is used as coarse seed for the regular precipitators, the overflow (stream 43) is pumped to a holding tank (Equipment 46). In this tank, stream 43 is mixed with the underflow (stream 50) from the tray thickener (Equipment 49) and with some of the overflow from the fine seed tank (Equipment 51). This mixture (stream 44) goes to a number of crystallisers (Equipment 47) to give a six-to-eight hour residence time, and a concentration of about 100 gm./litre solids. The slurry from these crystallisers (stream 45) goes to a continuous tray thickener (Equipment 49) where it is concentrated to about 250 gm./litre. The underflow (stream 48) is split, some (stream 50) going back to the holding tank (Equipment 46), the rest, (stream 51) going to the fine seed holding tank (Equipment 51). The overflow (stream 72) from the tray thickeners becomes recycle spent liquor. The underflow (stream 68) from this tank is mixed with advance wash to give a 20% solids by weight, and used as seed slurry for the special precipitators. Some of the overflow (stream 49) from the fine seed tank is removed as a purge stream to be calcined to remove organic material; some (stream 54) is returned to the holding tank (Equipment 46); and some (stream 53) is sent as a bypass to the continuous tray thickeners (Equipment 49). Stream 53 is usually empty except when stream 54 is too large for the second stage crystallisers. (See decription of SETROI module below).

Fig. 2.4.1 Information Flow Diagram of the Crystalliser Section.

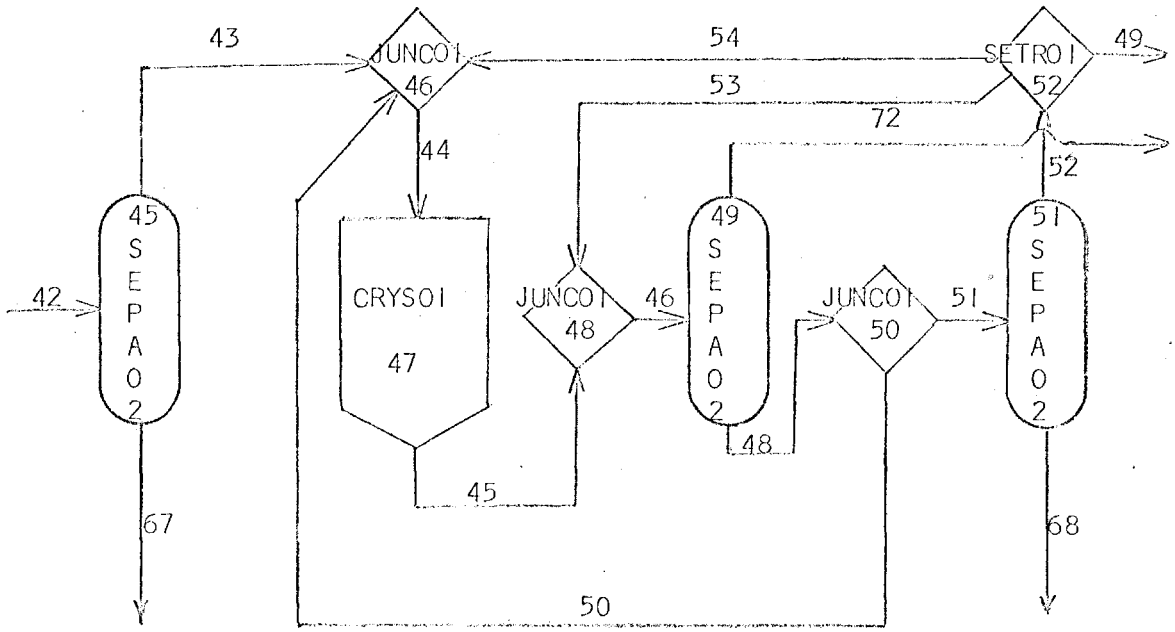
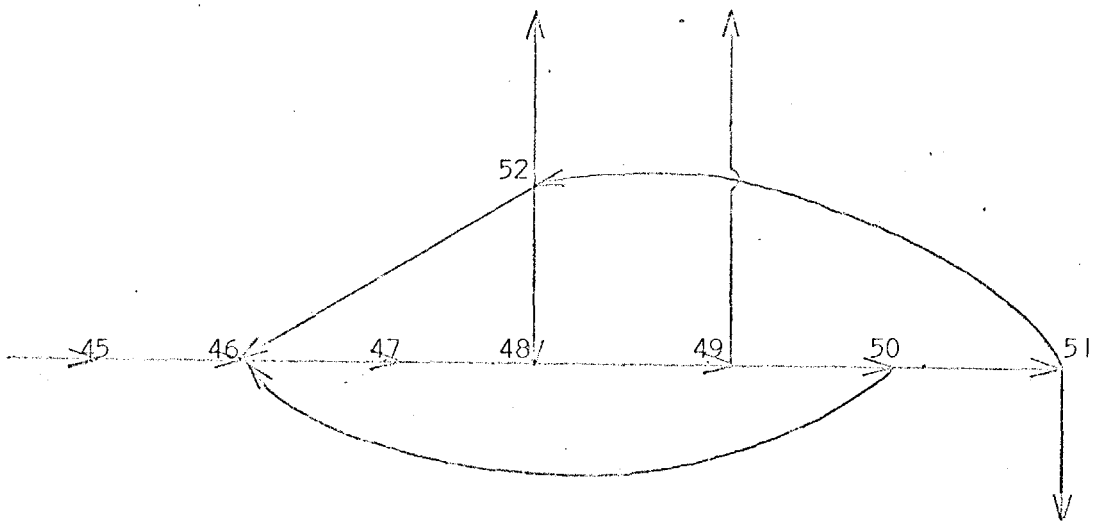


Fig. 2.4.2 Digraph of the above plant.



2.4.1 Module Description.

The SEPA02 modules (Equipments 45, 49, and 51) and the JUNC01 modules (Equipments 46, 48, and 50) are the same as described above (2.3.2).

The Second Stage Precipitator Module, CRYS01, simulates the continuous precipitation of hydrate using a rate constant, K_cA. The module first calculates the total mass flowrate (lb./hr.) from inlet stream composition (lb. moles./hr) and component molecular weights.

The density of the input stream liquor is calculated as a function of total titrable soda (TTS) and sodium aluminate (AL), and from it, the total volumetric flowrate is found: -

$$\text{Volumetric Flow} = \frac{\text{Mass Flowrate}}{\text{Density}}$$

Then the average molar concentration (AUG) is found: -

$$\text{AUG} = \frac{\text{Total Molar Flowrate}}{\text{Total Volumetric Flowrate}}$$

Next, one needs the required volumetric flowrate (TDT) to give the residence time (TOW) specified in the EN vector.

$$\text{TDT} = \frac{\text{Volume of precipitators (from EN vector)}}{\text{TOW}}$$

A required total molar flowrate (REQD) can then be calculated and stored in the EN vector for SETR01 (Equipment 52) to control the recycle in streams 53 and 54.

$$\text{REQD} = \text{TDT} \times \text{AUG}$$

The driving force for crystallisation is the supersaturation of the inlet liquor as $NaAlO_2$ compared to the equilibrium value for operating conditions. The inlet concentration of $NaAlO_2$ as Al_2O_3 is calculated (CAL) and the equilibrium value of Al_2O_3 is found from the correlation

with temperature and caustic concentration (EQUILM). The outlet concentration of Al_2O_3 is given as C: -

$$C = \frac{CAL + K_c A \times TOW \times EQUILM \times 28.313/454}{(1 + K_c A \times TOW)}$$

(where 28.313/454 is a units conversion)

The precipitation reaction is: -

$$2NaAlO_2 + 4H_2O \dashrightarrow 2NaOH + Al_2O_3.3H_2O$$

The Al_2O_3 precipitated is the volumetric flowrate multiplied by the difference in concentration between inlet and outlet, (CAL - C).

Using the stoichiometry of the above, a final mass balance is done for each component to correct for the precipitation reaction.

The Flow Controller Module, SETROI, is used to fix the total flow (stream 49) to the calciner, then to direct the remaining flow according to the desired flow, REQD, in the crystallisers. If the remaining flow is so large that it would reduce the residence time too much if it all went in stream 54 and thence to the crystallisers, then some is sent to stream 53. (Stream 53 is zero under normal plant operations).

To calculate the flow in stream 54, it is necessary to make a mass balance around the holding tank (Equipment 46). Streams 43 and 50 are known, and the desired value of stream 44 is given as REQD in the EN vector of CRYSOI. If no or negative flow is required in stream 54, then it is set to zero, and all the flow is sent to the bypass, stream 53.

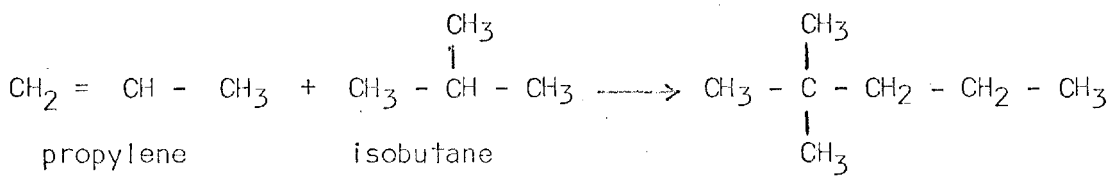
If material is needed, then it is necessary to see if the remaining flow is greater or smaller than the desired flow in stream 54. If it is greater, then the difference is sent to stream 53; if it is smaller, then all the flow is sent to stream 54.

The module sets the output temperatures and pressures equal to the input temperatures and pressures, and calculates the total, and component, flows.

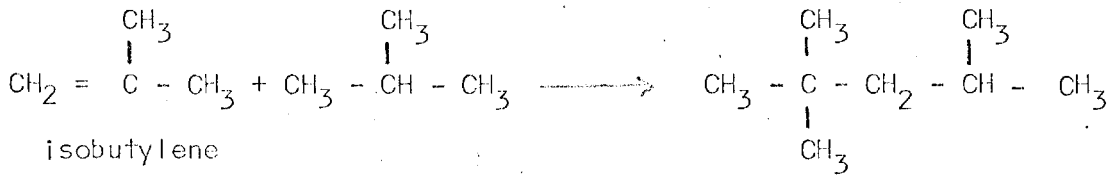
2.5 The Alkylation Plant.

The plant produces branched chain hydrocarbons for use in upgrading high octane motor fuels. The use of these compounds will become of greater importance as the ethyl-lead additions are eliminated.

The main reactions are as follows: -



propylene + isobutane → 2,2 dimethyl pentane



isobutylene + isobutane → 2,2,4 trimethyl pentane

Both these reactions are catalysed by high strength, about 95wt.% sulphuric acid.

The alkylation is carried out in the Stratco reactor (Equipment 3). The sulphuric acid in which the reaction has occurred (stream 11) goes to the acid settler (Equipment 4). There the acid settles and goes (stream 19) to the acid make-up (Equipment 11) where the spent acid is added to give the acid feed (stream 10) to the reactor.

The product stream (12) is mixed with the isobutane feed (stream 30), then the mixture is passed through a reducing valve to drop the pressure from 70 psig to about 27 psia, and then is flashed in coils immersed in the reactor liquid. The vapour-liquid mixture goes to the product separator where it is split into a vapour stream (14) composed mainly of C_3 and C_4 hydrocarbons, and a liquid stream (15) containing

alkylate and n-butane and iso-butane. The product separator has an internal heating coil through which the bottoms (stream 27) from the depropaniser (Equipment 12) pass. The flashing coils and the product separator are represented by Equipment 5.

The vapour stream (14) is compressed (Equipment 8) to 60 psig and 155°F, then washed in the caustic wash unit, (Equipment 16) and in the water settler (Equipment 17), then passed to the coalescer (Equipment 25). After washing, the stream is heated (equipment 29), then separated in the depropaniser (Equipment 12) into an overhead propane product (stream 6) and a C_4 bottoms stream (7) which is recycled to the Stratco reactor.

The liquid stream (15) is similarly washed in Equipments 19, 20, and 21, then separated in the de-isobutaniser into an overhead isobutane stream (2), a side-draw of n-butane (stream 3), and a bottoms of alkylate (stream 4). The isobutane is recycled back to the Stratco reactors. The olefin feed (stream 22) is mixed with this recycle, then the mixture (stream 23) is cooled (Equipment 30), and mixed with the C_4 recycle stream, then passed through the feed coalescer (Equipment 14) to give the hydrocarbon feed stream (9).

ALKYLATION PLANT

Fig.2.5.1 Information Flow Diagram.

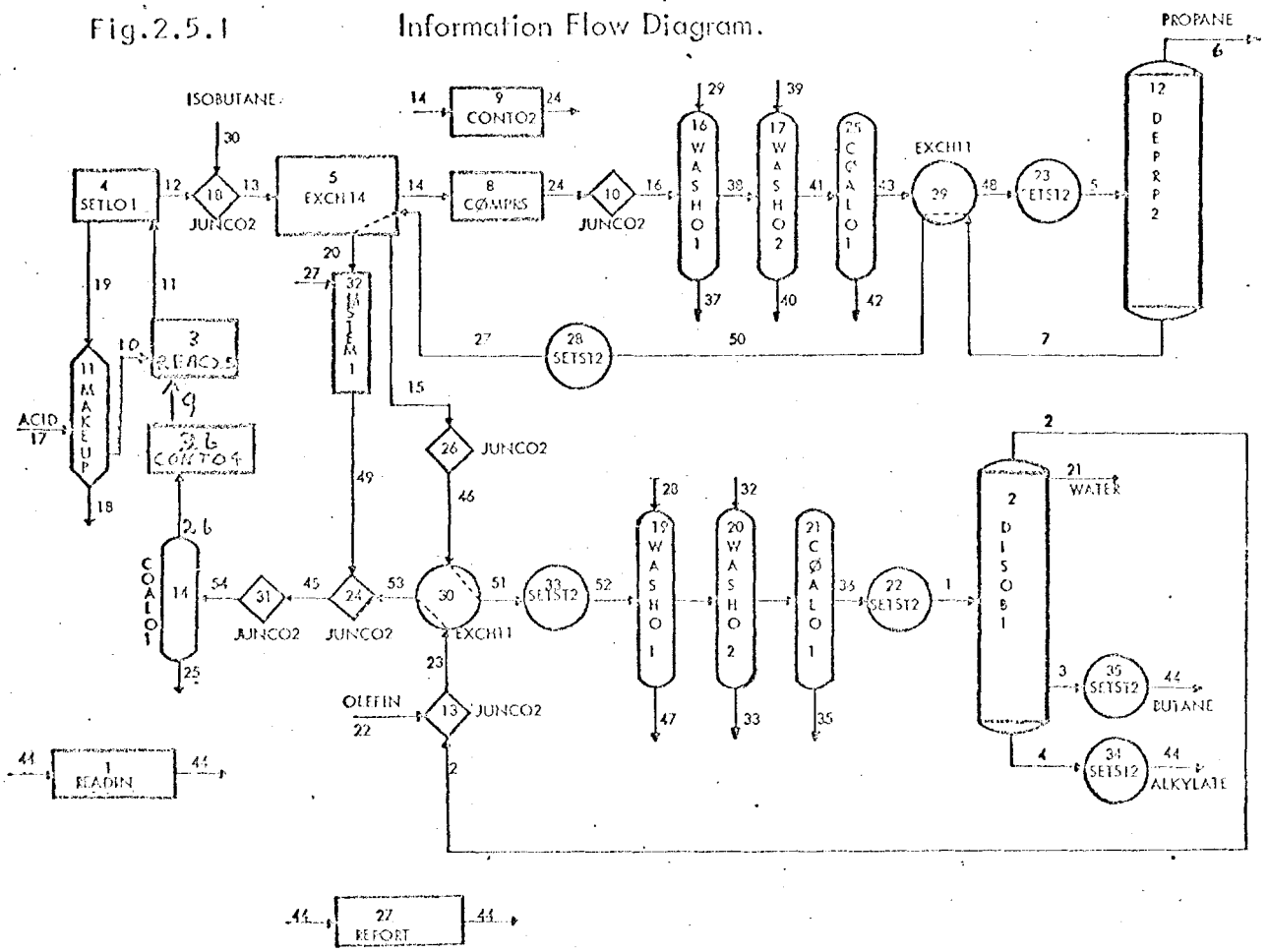
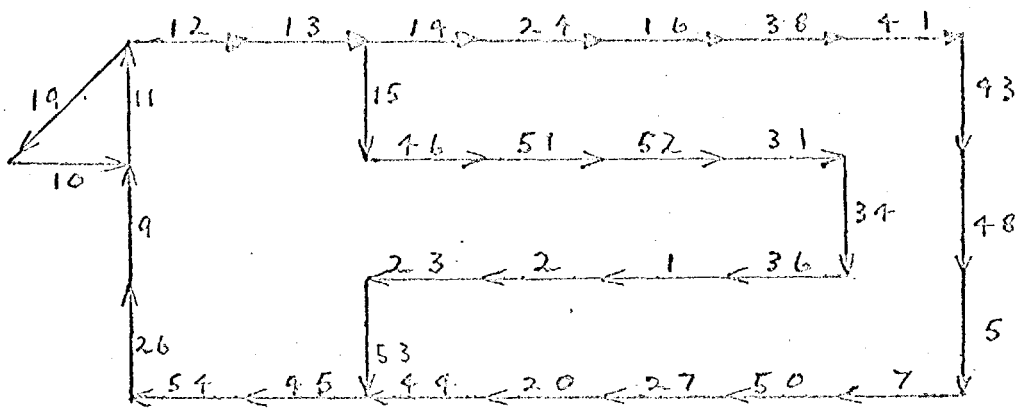


Fig.2.5.2 Digraph of the above plant



Note: The numbers represent streams.

2.5.1 The Equipment Modules.

The reactor module used is REAC 23, which is a comprehensive statistical model based on a set of simplifying assumptions* and on linear regressions on all the independent variables for each dependent variable. The assumption that no alkylate is present in the intake feed becomes important if any alkylate returns from the depropaniser bottoms, i.e. there must be no alkylate in the vapour phase at the product separator.

The acid used is obtained from the plant factor for bbl. of acid per bbl. of olefin input.

The total heat load is calculated from enthalpy change, heat of reaction, and impeller heat, and transferred to Equipment 5, the product separator.

One of the problems of using statistical modules is one of correlation limits. So a check is made of inputs to see if they are outside the limits, and a warning is printed if necessary. If the outputs are obviously wrong, for example if they are negative, they are set to a reasonable level on the assumption that in the next iteration the inputs will be within range.

The two distillation column modules, DEPROP2 and DISOB1, are quite similar. They both assume the distillate consists only of a light and heavy key. The number of minimum stages can then be found, and if the split of one of the keys is known, then the composition of distillate

---

* See p. 2.9, Section C of Comprehensive Report on a Digital Computer Simulation of an Alkylation Plant. McMaster University, 1969.

can be found, and by difference, the bottoms can also be found. The heat balance is done by finding the reflux ratio from a linearised Gilliland graph.

Three of the models, WASH01, WASH02, and COAL01, are concerned with water removal, and all use the same sub-routine, FSOL, to calculate the water solubility as a function of temperature in a hydrocarbon stream.

WASH01 is used to calculate the neutralisation of sulphuric acid by sodium hydroxide for three different cases which give: -

(a) A perfectly neutral solution with no acid or base left.

(b) A caustic solution in which the Caustic Molar Flowrate is twice the Acid Molar Flowrate.

(c) An acid solution in which the Acid Molar Flowrate is twice the Caustic Molar Flowrate.

The water that exists in the hydrocarbon stream consists of dissolved water and entrained water. The dissolved water is calculated by FSOL; the amount of entrained water is found by multiplying the total feed flowrate by an arbitrary entrainment factor which is read into the EN vector.

The water wash module, WASH02, transfers the hydrocarbons directly to the hydrocarbon output stream. It then dilutes the other components with the water input stream, and calculates the dissolved and entrained water as above. The salts and acid, or caustic, are split according to the water split, and the remaining water, salts, and acid, or caustic, exit through the sewer stream.

An energy balance is also performed by both models to find the temperatures of the exit streams.

The coalescer model, COAL01, knocks out part of the input water according to a given water removal efficiency stored in the EN vector, and sends the hydrocarbon stream overhead.

The acid make-up module, MAKEUP, represents the batch removal of spent acid, and the addition of fresh. The composition and flowrate desired as feed to the reactor, and the concentration of fresh acid, is stored in the EN vector. The incoming stream from the decanter is known, so an acid and an overall material balance will give the fresh and the spent acid flowrates. A heat balance is also carried out over the four streams.

The acid settler module, SETL01, transfers the hydrocarbons and the acid carryover specified in the EN vector, to the overhead stream, and the acid and the degradation products to the bottoms. The water, sodium sulphates, and alkyl sulphates, are split into the exit streams in the same ratio as the acid.

The product separator and the heat exchanger inside the Stratco reactor are both modelled by the EXCH14 module. The module calculates the input enthalpy by summing the enthalpy of the incoming stream, which is assumed to be liquid, the reactor heat input, which is stored in the EN vector, and the heat input from the product separator heat exchanger. The output enthalpy is the enthalpy of the liquid and vapour streams, plus the enthalpy of the output from the heat exchangers. The enthalpy of the liquid and vapour streams depends on the flashing temperature which is found by a Reguli-Falsi search for the temperature where input and output enthalpies are equal. The acid in the feed to this module is dealt with by destroying half of the input in order to simulate the acid

which has collected in the acid leg, and by assuming 10% of the remainder is entrained in the vapour stream.

The compressor model, COMPRS, is a comprehensive model which also includes the throttling valve and condensers. The product separator pressure and compressor outlet pressure are linked, and a search procedure is needed to find the right product separator pressure in order to compress the vapour stream for that particular volumetric flow. First the product separator pressure is taken from the input stream, then the pressure drop between the product separator and compressor, and the compressor compression ratio, are calculated. Now the operating pressure of the condenser must equal the pressure drop from the compressor to the condenser, plus the outlet pressure of the compressor. If it does not, the control module, CONTO2, is used to perform a Reguli-Falsi search to alter the pressure in the product separator until it does. This is done by placing the new value of the pressure in the EN vector of EXCHI4, and forcing the recalculation of this module.

MSTEMI is a module used to break the heat recycle iteration round the product separator-compressor-depropaniser section of the plant. It picks up the flows from stream 27 and the temperature from stream 20, passing the combined information into stream 49.

The EXCHII module is a comprehensive heat exchanger model with a choice of eight different types of exchanger: -

(1) Countercurrent.

(2) Parallel.

(3) Crossflow, both streams unmixed.

(4) Crossflow, both streams mixed.

(5) Crossflow, stream CMIN unmixed where CMIN = smaller of the thermal capacitances.

(6) Crossflow, stream CMAX unmixed where CMAX = larger of the thermal capacitances.

(7) Parallel - counter flow with N shell passes.

(8) Liquid or gas on one side and a condensing or evaporating fluid on the other side.

The type desired can be selected by placing the appropriate number in the third place of the EN vector. Equipment 29 was simulated by means of the countercurrent model.

The mode of calculation uses the effectiveness factor concept to give the outlet temperatures of both streams directly. The UA, the number of shell passes, and the location of the hot fluid, tube or shell side, must be specified.

The JUNCO2 modules are similar to the JUNCO1 modules except that the JUNCO2 modules have the ability to set temperatures and pressures of output streams, and thus may be used to simulate liquid pumps.

The SETST2 modules set the temperature of the outlet stream.

2.6 Experimental Techniques.

In order to satisfy any of the aims of the work (see 2.1), it is necessary to have some criteria, or tests, to determine when a solution has been obtained, or when the simulation has converged. In addition, it is necessary to define an error norm so that the progression of the simulation towards the converged solution may be studied.

2.6.1 Convergence Tests and Error Norms. (Orbach, 01)

In any iterative procedure that converges, the vector $\bar{X}_n$ after n iterations, will approach the solution vector $\bar{X}_s$ asymptotically so that when n goes to infinity, $\bar{X}_n$ will equal $\bar{X}_s$. Usually one wishes to find some converged vector, $\bar{X}_c$, which will be in the neighbourhood of $\bar{X}_s$ after a finite number of iterations.

The test of convergence used in MACSIM is to examine the fractional change of each element in every stream between two successive iterations. If the modulus of this fractional change is less than some small tolerance, then the iterate is said to have converged. This can be written as: -

$$\bar{E}_n = [\, D(\bar{X}_n)\,]^{-1} (\bar{X}_{n+1} - \bar{X}_n) \qquad \ldots\ldots\ldots 2.6.1$$

where $D(\bar{X}_n)$ is a diagonal matrix with elements x_i.

One can take the norm of this test vector, $\bar{E}_n$, : -

$$\| \bar{E}_n \|_2 = \left\{ \sum_{i=1}^{m} (\bar{E}_{ni})^2 \right\}^{\frac{1}{2}} \qquad \ldots\ldots\ldots 2.6.2$$

which will give a measure of the deviation from the true solution. The ratio of two successive error norms, u, is asymptotically the value of the dominant eigenvalve at the n^{th} iteration.

$$u_n = \frac{\| \bar{E}_n \|_2}{\| \bar{E}_{n-1} \|_2} \qquad \ldots\ldots\ldots 2.6.3$$

However, though u_n can be calculated at each iteration, it is not until it remains approximately constant for at least three iterations that it can be said to be the value of the dominant eigenvalve of the system.

2.6.2 Direct Substitution.

Since most of the work in simulation has either been done using Direct Substitution or is based on this method, all of the cases which have been used in this work have been tested in this mode.

2.6.3 Convergence Promotion by Geometric Extrapolation.*

This technique can be used whenever the iterate has approached a geometric progression, and can be expressed thus: -

$$\bar{X}_n = \bar{U}u^n + \bar{X}'_s \qquad \text{(where } \bar{X}'_s \text{ is an estimate of } \bar{X}_s\text{)} \quad \ldots.2.6.4$$

This expression can be written for any component i: -

$$x_{n,i} = U_i u^n + x'_{s,i}$$

$$i = 1, 2, 3, \ldots m. \quad \ldots.2.6.5$$

Therefore, by eliminating U_i over the last three iteration points, (x_{n-2}, x_{n-1}, x_n) one obtains: -

$$x'_{s,i} = \left\{ \frac{x_n \cdot x_{n-2} - x_{n-1}^2}{x_n - 2.x_{n-1} + x_{n-2}} \right\} i \quad \ldots.2.6.6$$

Therefore, the apparently 'true' value $x'_{s,i}$ can now be substituted as the improved estimate.

However, the equation which is actually used in the CONV40 program

---

* For convergence tests and error norms, and convergence promotion by geometric extrapolation, see pp. 24 and 55 respectively of O. Orbach's thesis, A Study of the Promotion of Convergence in the Calculation of Complex Chemical Plants. McMaster University, 1969.

is derived from the above one, and is: -

$$x'_{si} = x_{n-1,i} + \frac{x_{ni} - x_{n-1,i}}{1 - u} \qquad \ldots\ldots 2.6.7$$

The program used for geometric extrapolation is one written by Orbach, called CONV40. This was originally written for the BAYER process. There is a choice of up to eight assumed streams which must be specified within the program. The normal mode of operation is to use as assumed streams in the promotion technique, the same ones that have been specified as 'cut' streams in the sequence of calculation. However, some cases have been run in which the streams assumed by the promotion technique were different from the cut streams specified by the sequence of calculation.

Modifications had to be made to the CONV40 program to run it on the Alkylation Plant. These consisted of changing the COMMON storage, and changing certain symbols which clashed with the Alkylation COMMON storage.

2.6.4 The Timing of the Iterations.

In order to test the time taken for one iteration of various sequences of calculation, a small dummy module was put into the sequence of calculation. Every time it is called, this module prints the value of the execution time at that moment.

2.6.5 Effect of Starting Point.

Certain of the cases were run for five iterations, then the streams were printed out and used as initial values of the streams for different sequences of calculation. This technique is known as changing one's sequence in mid-calculation.

CHAPTER THREE.

RESULTS.

3.1 Introduction.

In this chapter, the results of the analysis of the three plants described in Chapter Two are presented. Each plant is divided into 'Groups' such that a stream, or pair of streams, from each group is present in every feasible set of assumed streams. Then it is shown how to generate every feasible set and hence every feasible sequence of calculation, for each plant. The number of iterations taken to converge are given for two techniques of convergence, Direct Substitution, and Geometric Convergence Promotion, for each feasible sequence which is run in each plant.

Originally it was hoped to compare directly the processing time and cost between the sequences of calculation. However it soon appeared that there were large differences in compilation time. It was therefore decided to use a library function, TIMNOW, which obtains the machine time directly from the machine clock. A model, TIMNOI, was written, which calls TIMNOW and prints the machine time. This model is usually put at the end of the sequence of calculation and prints the machine time every iteration. It is therefore possible to obtain time per iteration and the overall time. The time taken for any large amounts of printing can also be decided, and allowed for.

When the runs were made, it was apparent that for some runs the time for the runs corresponded to the number of iterations taken, i.e. that time per iteration was constant; but this was not true for other

runs, i.e. the time per iteration differed by 10% - 20% within the run itself. One run had a time per iteration of one second up to the sixty-sixth iteration which took 140 seconds. The subsequent iterations also took one second each. Therefore several runs were repeated, and variations of 10% were found. This variation can be attributed to several factors, some of them data dependent, and others operator dependent. The multiplication and division of numbers can vary by a factor of five, depending on whether the number in the accumulator is one or zero. However, with a large number of operations this should average out. The other data dependent factor is the presence of IF statements and DO loops. If the computer suddenly goes into a new part of the program not normally used, then this could be reflected in a change in time. The models used, JUNC01 and SEPA02, are very simple, and the time differences cannot be accounted for in this way, particularly since the initial starting point is the same, and the converged results are very close together indeed.

The other factors are dependent on the operator. On the I.B.M. 7040 there are two modes of operation, on spool and off spool. On spool uses disc storage between the computer and the printer, and off spool uses the computer linked directly to the printer. Off spool is slower because it is printer limited. In addition, the machine clock used by TIMNOW can be turned on and off by the operator. Because of the variability, and these random factors, it was decided that the number of iterations taken to converge would be a better measure of convergence than the time taken.

3.2 BAYER Process Wash Section.

An example shown at the end of Chapter One illustrates how the collection of all the feasible sets of fourth-order assumed streams may be generated. This collection is a subset of all the feasible choices of assumed streams, and hence of sequences of calculation. The plant is divided into three groups, each of which must contribute to the feasible set in order for the calculation to be possible. It should be emphasised that these three sections are not independent of each other, and, in fact, form one recycle. The division into three parts is, like Caesar's division of Gaul, for convenience only.

Group I		Group II		Group III	
Code	Stream No.	Code	Stream Nos.	Code	Stream Nos.
A	22	1	31, 30	0	100
B	24	2	25, 30	1	32, 36
C	26	3	31, 28	2	35, 108
		4	25, 28	3	35, 109
				4	35, 110
				5	36, 37

Therefore to generate a feasible set of streams, a stream, or pair of streams, is selected from each group. This procedure also gives the run coding for the set, e.g. A20 is coding for 22, 25, 30, 100. Once the feasible set is generated, then a sequence of calculation can be found in a similar way; -

Group I		Group II	
A	[18 ( ) 17]	1	(23, 24, 25)
B	[( ) 17, 18]	2	(24, 23, 25)
C	[17, 18 ( )]	3	(23, 25, 24)
		4	(25, 24, 23)

Group III

0	26, 58, 29, 28, [ ], 22
1	22, 26, 58, 29, 28, [ ]
2	58, 29, 28, [ ], 22, 26
3	29, 28, [ ], 22, 26, 58
4	28, [ ], 22, 26, 58, 29
5	[ ], 22, 26, 58, 29, 28

Using the coding for the particular set, e.g. A20, the square bracket from Group I corresponding to A, 18 ( ) 17 is taken and filled with the round bracket from Group II corresponding to 2, (24, 23, 25) to give A2 18 (24, 23, 24) 17. This is then placed in the appropriate place marked by the square bracket in a sequence from Group III, in this case 0, to give: -

A20 = 26, 58, 29, 28, [18 (24, 23, 25), 17], 22.*

Using this technique, it is possible to generate all 72 feasible sets of assumed streams and their associated sequences of calculation. It is usually possible to obtain an equivalent sequence of calculation which is apparently** but not essentially, different from a given one.

---

* N.B. The brackets are for illustration purposes only

** See back to page 3.

For example, in A20 one could calculate Equipment 25 before Equipment 23. However, as long as Equipment 24 is calculated before either of them, and Equipment 17 is calculated after Equipment 23, there is no difference at all in the convergence. This has been verified experimentally, in that the rate of convergence was the same in the two cases, and the final and intermediate values of the stream variables were identical for the two equivalent sequences of calculation. However, the results were not identical when two different, non-equivalent, sequences of calculation were compared, nor when the same sequence was calculated by Direct Substitution and by Convergence Promotion.

3.2.1 Direct Substitution.

The number of iterations that each sequence of calculation takes to converge is tabulated in Table 3.2.1. The minimum number of iterations is 76, and there are six different sequences of calculation which yield this result. Three of them are fourth-order, and three fifth-order; however, there are twelve fourth-order cases and 60 fifth-order, so that the fourth-order sequences are proportionately better. The choice of streams may be expressed as below: -

Group I $A \leq B \leq C$

Group II $4 < 3 = 2 < 1$

Group III $0 \leq 2 \leq 1 < 3 = 4 = 5$

Therefore A40, B40, C40, A42, B42, A41, are the optimal sequences, and C13, C14, and C15, are the worst sequences with the other sequences graded in between.

If one looks at the choices of streams from Group II necessary to obtain the optimal sequences of calculation, and then refers to the

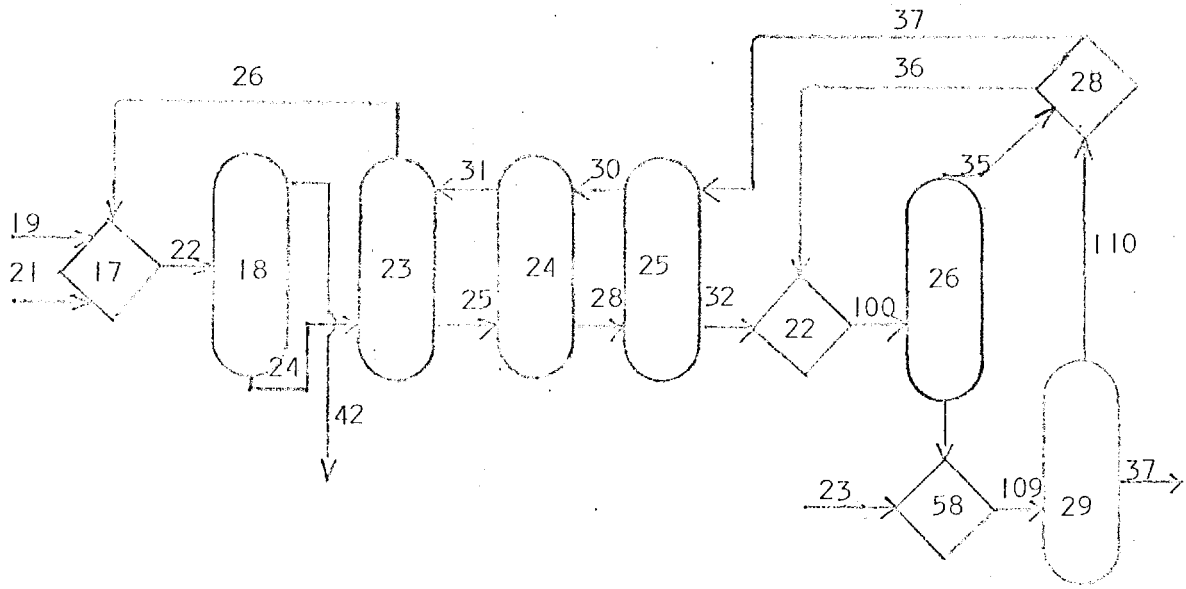
diagram on the following page, one can make the following observations: -

(1) Stream 100, which links unit 22 to unit 26, is the best choice because it produces three optimal sequences.

(2) Streams 35 and 108, which are the output streams of unit 26, are the second best choices, producing two optimal sequences.

(3) Streams 32 and 36, which are the input streams to unit 22, are the third best choices, producing one optimal sequence.

Fig. 3.2.1 Information Flow Diagram of BAYER Wash Section.



Therefore the three best choices are linked together by stream 100, which is the best choice.

Similarly for Group II one may make the following observations: -

(1) Streams 25 and 28 are the best choices.

(2) Streams 30 and 31 are the worst choices.

(3) Streams 25 and 30, and streams 31 and 28, are equivalent to each other,

and midway between the other two cases. There is only one example where this is not completely true: C30 converges in 77 iterations, whereas B20 converges in 80.

TABLE 3.2.1 Number of Iterations taken to converge when Direct Substitution is Used.

GROUP III \ GROUP II	1			2			3			4		
GROUP I	A	B	C	A	B	C	A	B	C	A	B	C
0	80	80	81	77	77	80	77	77	77	76	76	76
1	80	80	81	77	77	80	77	77	80	76	77	77
2	80	80	81	77	77	80	77	77	80	76	76	77
3	82	82	83	81	81	82	81	81	82	78	78	81
4	82	82	83	81	81	82	81	81	82	78	78	81
5	82	82	83	81	81	82	81	81	82	78	78	81

3.2.2 Convergence Promotion.

The use of Orbach's Geometric Convergence Promotion (01) gives an approximately 50% reduction in the number of iterations taken to converge. The optimal sequence is C21 which converges in 38 iterations. The two worst sequences were* A21 and B21 needing 58 and 63 iterations respectively. The reason for the poor convergence in both the above cases is that the second convergence promotion step was in the wrong direction, and approximately 18 iterations were needed to reduce the error norm to below the value at which the wrong promotion occurred. The similarity in the sequence of calculation between the best case and the two worst cases illustrates the necessary compromise between rapid convergence and accurate convergence promotion. The beneficial effect of a constraint on promotion for the two worst cases substantiates the sensitivity of these sequences. The graphs of error norm against number of iterations illustrate this point.

When Convergence Promotion is used, it is difficult to state any order of preference for choosing assumed streams, unlike the case where Direct Substitution was used. It is possible to use a game theory

---

* These cases were for a minimum KOUNT of 5, and promoted after a KOUNT of 6, when the minimum KOUNT was increased to 8 promotion occurred after a KOUNT of 11 and convergence was attained in 45 iterations for both cases. See Table 3.2.2.

The KOUNT facility in the convergence promotion routine counts the number of iterations after the previous convergence promotion jump. Normally the jump occurs when the iterate approaches a geometric progression, i.e.: -

$$|\Lambda_n| \leq \gamma \quad \text{where} \quad \Lambda_n = (u_n - u_{n-1}) / u_{n-1}$$

and γ is a small tolerance which was chosen to be 0.005.

However, it is possible to use the KOUNT facility to ensure that the jump does not occur before a fixed number of iterations have occurred.

approach and treat the choices of assumed streams as strategies. In this approach one wishes to find the best strategy, or choice of stream, within a group whatever the choice within the other groups. This can be done easily from Table 3.2.2(a) by summing the rows and columns and taking the averages. This generates directly the average time for the Group II choices, the averages obtained from summing and averaging the rows, and thus the best strategy from Group III can be picked, (Table 3.2.2(b)i). The summing of the columns gives the averages for each choice in Group I combined with each choice in Group II, and so is represented in Table 3.2.2(b)ii. Because of bad promotion in A2I and B2I, the runs were made again and are represented as A2I* and B2I*. This leads to extra strategies, A* and B* in Group I, 2* in Group II, and I* in Group III.

For Group I the order of preference is: - $C \prec A^* \prec B^* \prec A \prec B$

For Group II the order of preference is: - $4 \prec 3 \prec 2^* \prec 1 \prec 2$

For Group III the order of preference is: - $1^* \prec 2 \prec 4 \prec 3 \prec 0 \prec 5 \prec 1$

One would conclude from this that the optimal sequence should be C4 I* However this sequence although above average, is not the optimal one. Since the optimal sequence cannot be chosen in this way even when all the feasible sequences have been run, it is unlikely that a statistical technique based on a sample of the feasible sequences will choose the optimum.

It acts as a constraint on the technique, but it can improve the convergence dramatically in appropriate cases such as A2I and B2I.
N.B. u_n is defined in Chapter Two, p. 44.

TABLE 3.2.2(a) Number of Iterations taken to converge when Convergence Promotion is used.

GROUP III	GROUP II												
	1			2					3			4		
	A	B	C	A	A*	B	B*	C	A	B	C	A	B	C
0	47	48	49	44		47		48	49	48	44	47	46	44
1	47	47	46	58		63		38	43	43	44	47	51	45
1*					45		45							
2	47	48	47	46		46		47	46	46	44	44	44	47
3	47	47	45	49		50		48	46	47	45	43	43	49
4	47	46	44	48		48		47	46	48	45	43	44	49
5	48	48	48	48		50		47	48	48	48	44	44	49

TABLE 3.2.2(b)i Average Number of Iterations for Group III choices

0	1	1*	2	3	4	5
46.666	47.666	45.083	46	46.583	46.25	47.5

TABLE 3.2.2(b)ii Average Number of Iterations for Groups I and II.

Group II	Group I A	Group I B	Group I C	Average Group II
1	47.17	47.33	46.5	47
2	48.83	50.67	45.83	48.44
2*	46.67	47.67		46.72
3	46.33	46.67	45	46
4	44.67	46.67	47.17	45.76
Average Group I	46.75	47.5	46.125	
Average Group I*	46.21	46.75		

Fig.3.2.2 Graph of A21 comparing Direct Substitution ——————
and Convergence Promotion — — — — —

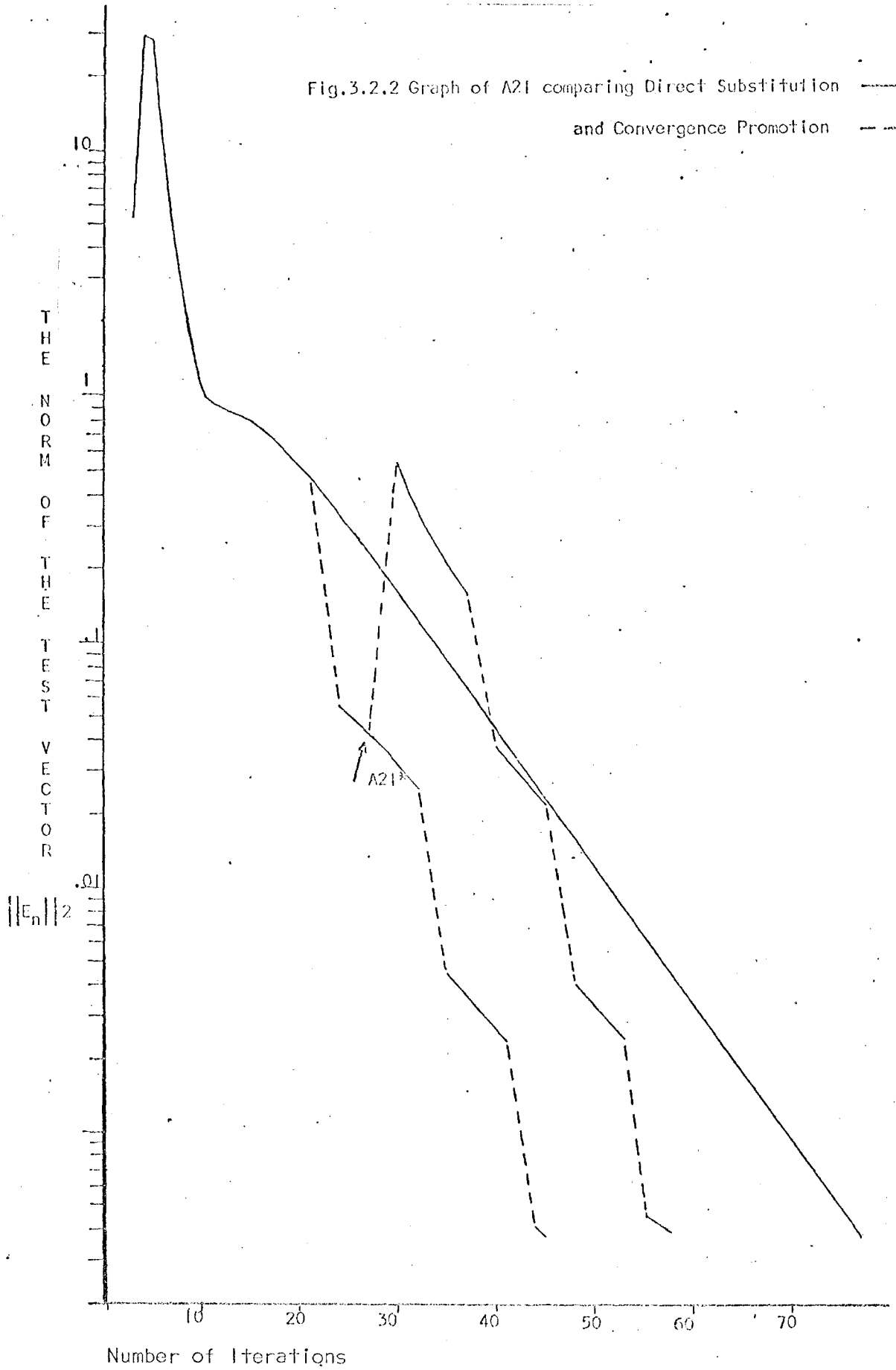


Fig.3..2.3 Graph of B21 comparing Direct Substitution————
and Convergence Promotion - - - -

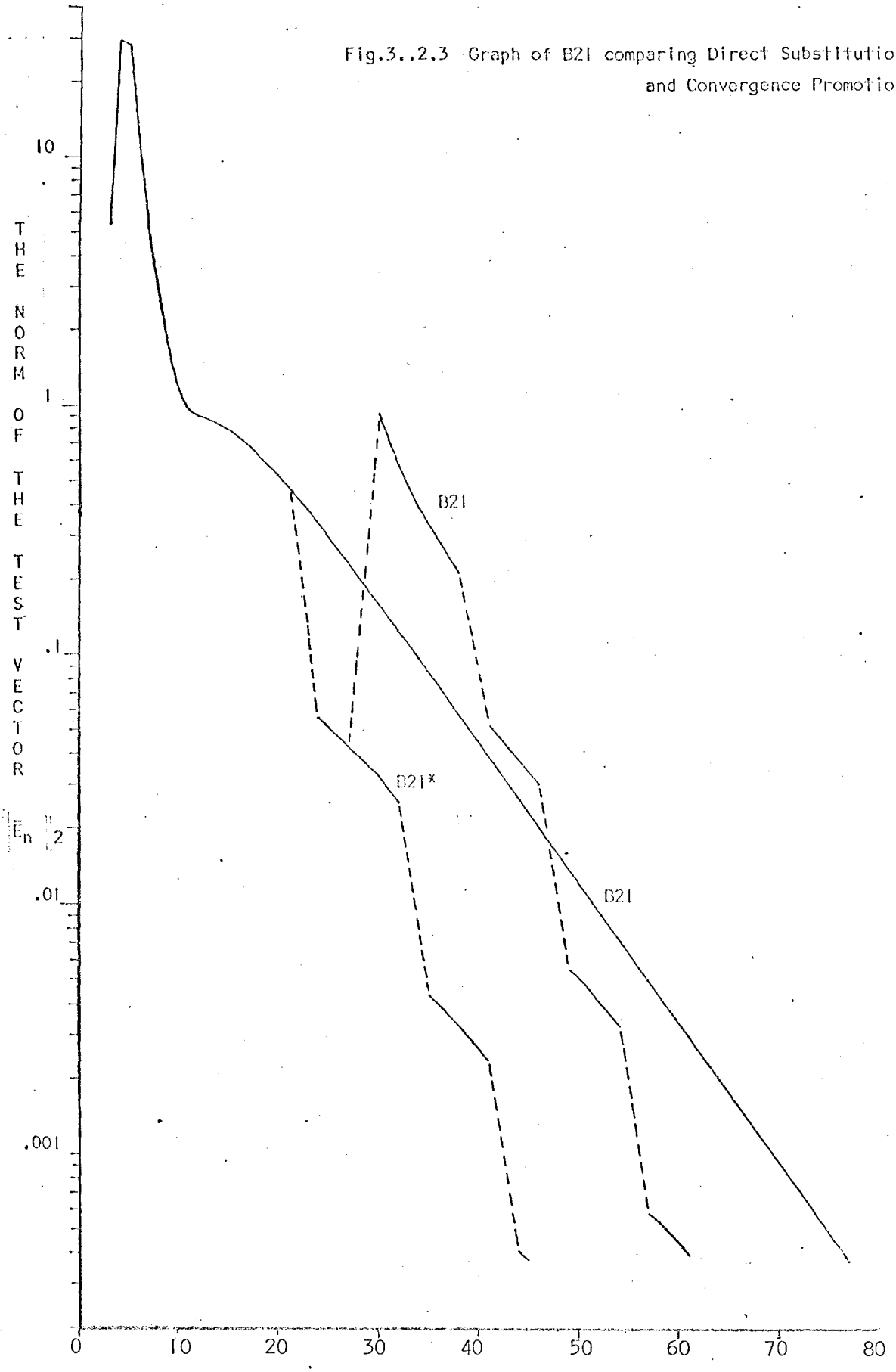
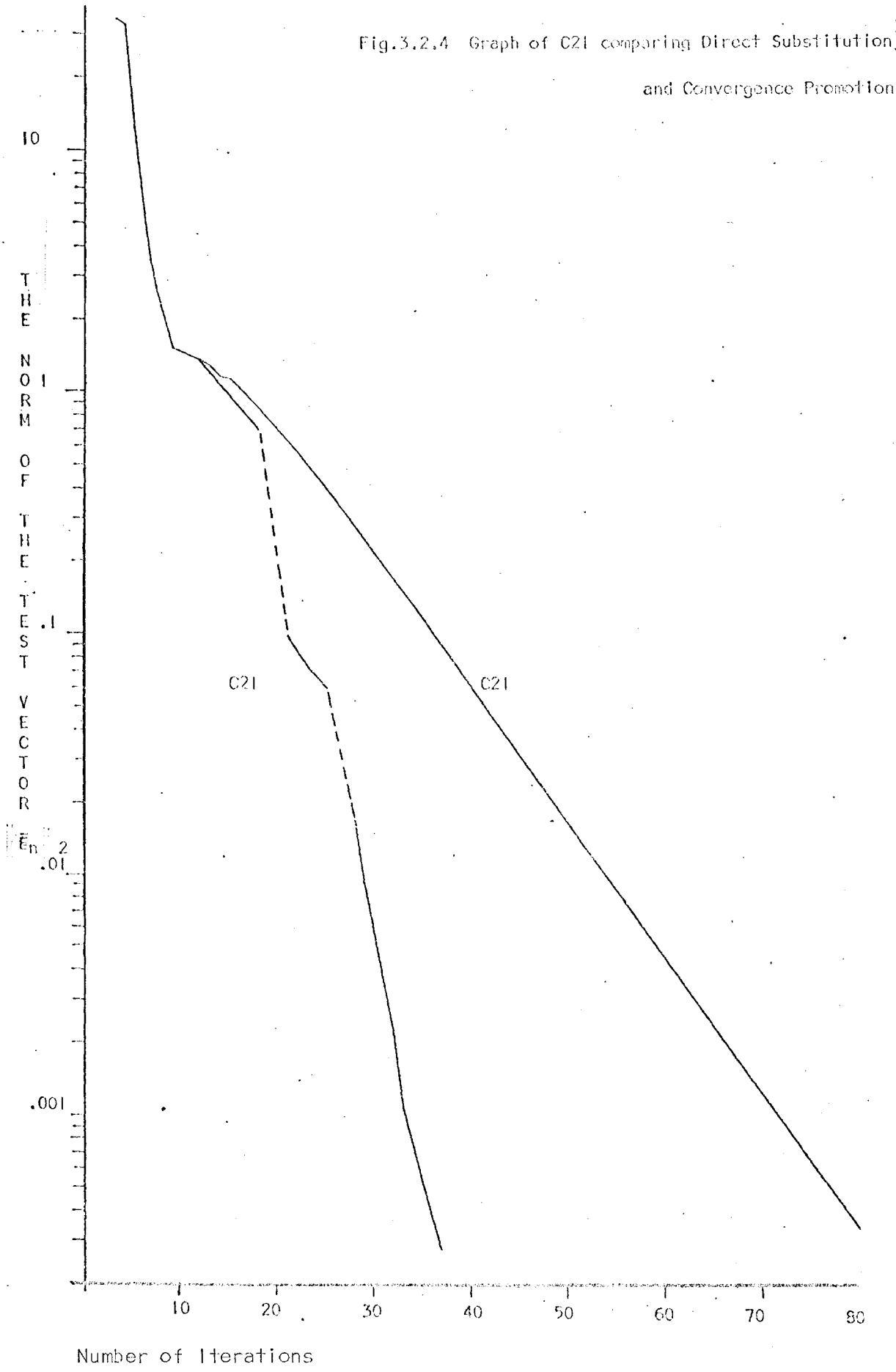


Fig.3.2.4 Graph of C21 comparing Direct Substitution

and Convergence Promotion- - - - -



3.2.3 Starting Point and its Effect on Convergence.

All the previous runs were made with all streams, except feed streams, initially zero.

In order to examine the effect of starting point on the number of iterations taken to converge, several runs were made where the values of the initial streams were obtained by running a case for five iterations, and then punching out the values of the streams.

Two runs were chosen to produce the initial streams, A20, a fourth-order case, and C15, a fifth-order case. Four runs were chosen to test the effect of the different starting points, A20, C20, C41, and C15.

TABLE 3.2.3.

	Convergence Promotion.			Direct Substitution.		
	Zero	A20	C15	Zero	A20	C15
A20	44	39	42	77	72	
C20	48	39	42	80	72	
C15	48	41	43	83	72	77
C41	44	40	62	77		

As can be seen for C20, the use of initial streams from A20 improves convergence appreciably and makes this case as good as A20. The use of initial streams from C15 improves C20 less, but it still converges as well as A20 and better than the other two. Therefore it would seem that for Group I the difference in overall convergence between A, B, and C is caused by differences in convergence in the first five iterations only. Run C15 converges better using as its initial streams those from run A20 than it does using its own values.

Run C4I does not converge better using as its initial streams those from A20 than it does using its own values, and using the values from run C15 it converges very badly due to a convergence promotion step in the wrong direction. The graph of the error norm for this run is very interesting as the convergence promotion step in the wrong direction also causes the subsequent Direct Substitution calculations to oscillate. The error norm increases, decreases, increases again, then decreases steadily, and eventually convergence is achieved. This could probably be prevented by increasing the number of iterations, KOUNT, needed before promotion is allowed in a similar way to the A2I and B2I cases earlier. Usually, therefore, the choice of initial values does not influence the rate of geometric convergence, the straight line portion after the turning point on the graph of the error norm versus the number of iterations, but does change the point at which this convergence occurs.

Fig. 3.2.5

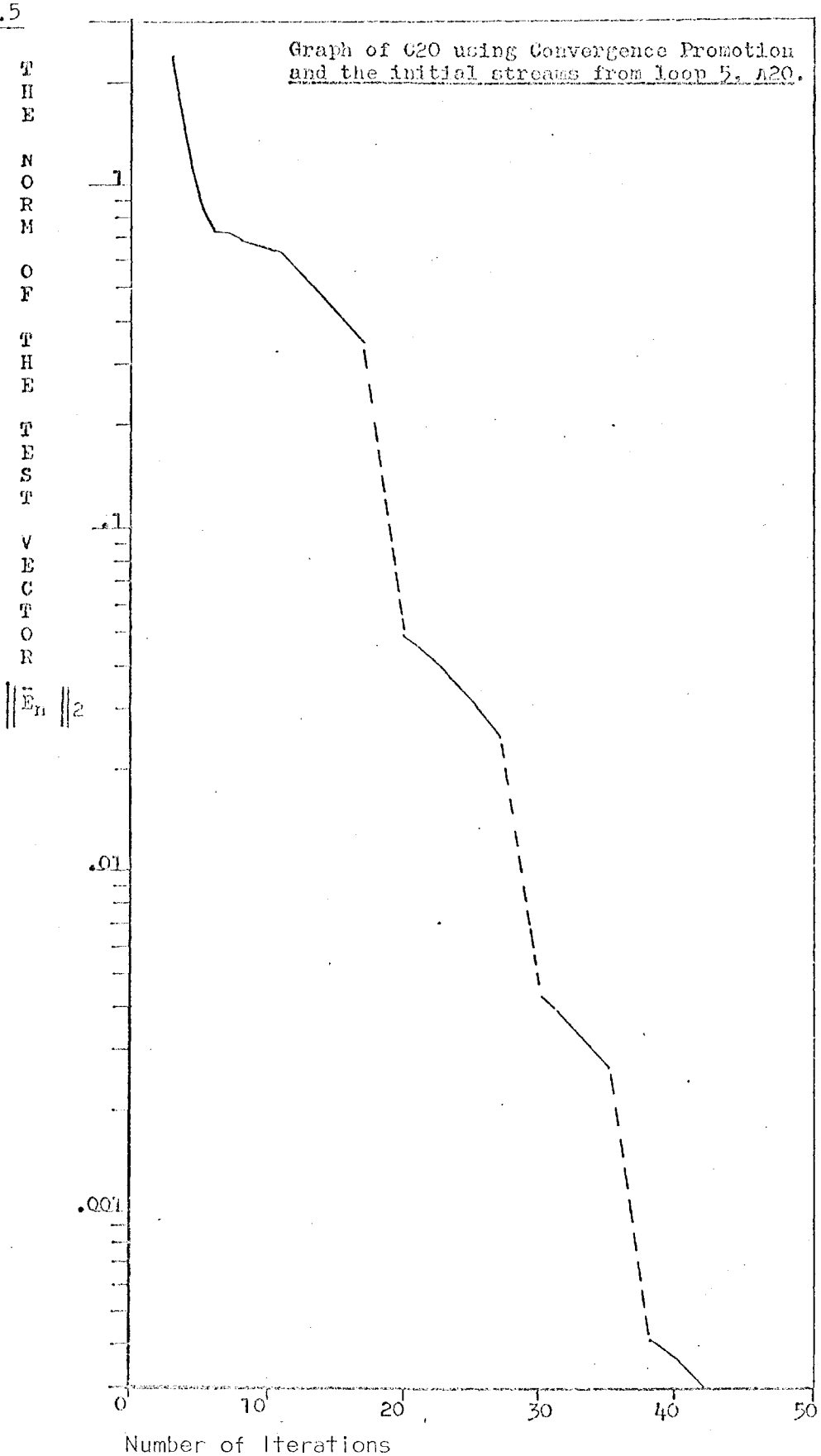

Graph of C20 using Convergence Promotion and the initial streams from loop 5, A20.
THE NORM OF THE TEST VECTOR $\|\vec{E}_n\|_2$
1
.1
.01
.001
0
10
20
30
40
50
Number of Iterations

Fig. 3.2.6

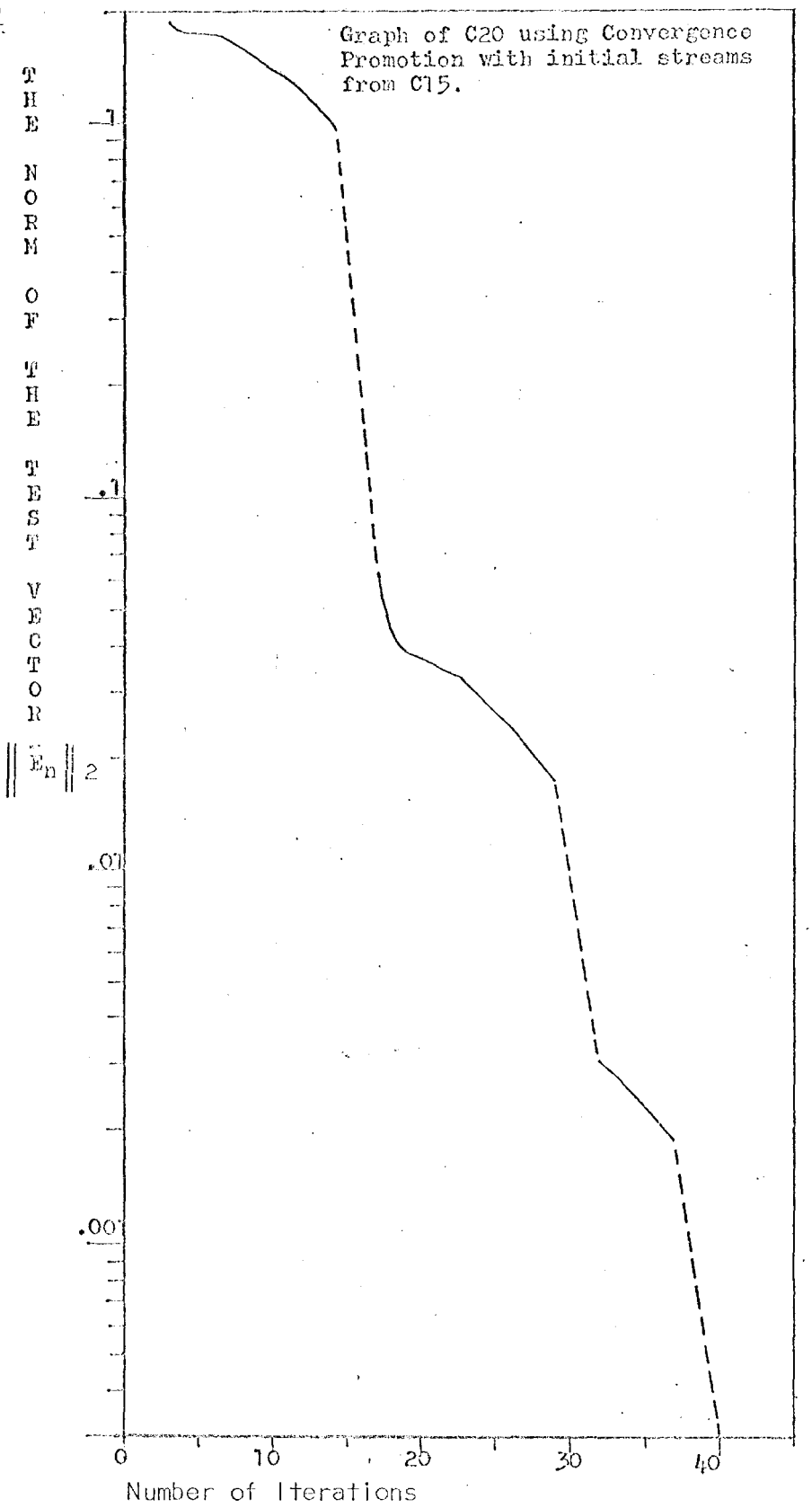
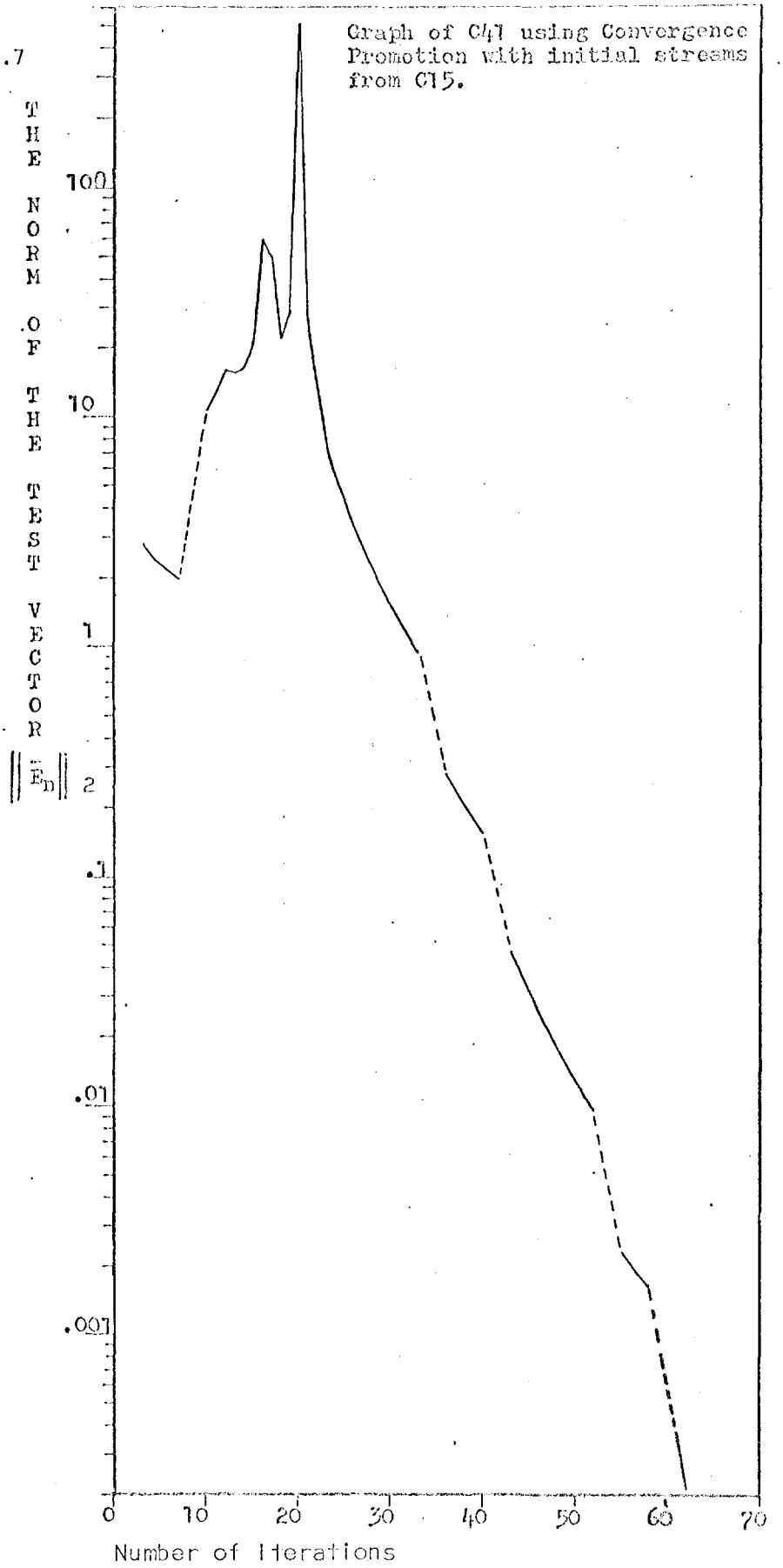

Graph of C20 using Convergence Promotion with initial streams from C15.
THE NORM OF THE TEST VECTOR $\|\bar{E}_n\|_2$
1
.1
.01
.001
0
10
20
30
40
Number of Iterations

Fig.3.2.7


Graph of C41 using Convergence Promotion with initial streams from C15.
THE NORM OF THE TEST VECTOR $\|\bar{E}_n\|_2$
100
10
1
.1
.01
.001
0
10
20
30
40
50
60
70
Number of Iterations

3.2.4 Choice of RANDOM Sequence.

The question of the correct choice of assumed streams and of calculation sequence becomes pointless if a sequence chosen at random is as good as any other sequence. Therefore three random sequences were generated by picking equipment numbers out of a well-shaken bag and noting down the sequence. The three sequences were: -

RAND. I.

Sequence of Calculation: - 28, 24, 26, 58, 22, 25, 23, 17, 18, 29

Assumed Streams: - 110, 35, 25, 30, 100, 32, 26, 24

RAND. II.

Sequence of Calculation: - 17, 18, 26, 22, 28, 25, 24, 29, 23, 58

Assumed Streams: - 26, 100, 32, 36, 110, 28, 25, 109

RAND. III.

Sequence of Calculation: - 29, 23, 28, 58, 24, 26, 17, 22, 25, 18

Assumed Streams: - 109, 24, 31, 35, 108, 30, 100, 32

None of the above sequences converged either using Convergence Promotion or Direct Substitution, within the 100 iterations for which they were run.

3.3 Crystalliser Section.*

The crystalliser section is connected together in a simpler way than the wash section, but it uses more complex models. One of the models, the SETROI, equipment number 52, has three output streams, 49, 53, and 54. However 53 is an overflow stream and is usually empty, therefore it has not been used as an assumed stream, which simplifies the problem of designating the feasible sets of assumed streams. The sequences of calculation and associated assumed streams are given below: -

	Sequence	Assumed Streams
1	46, 47, 48, 49, 50, 51, 52	50, 54
2	47, 48, 49, 50, 51, 52, 46	44
3	48, 49, 50, 51, 52, 46, 47	45
4	49, 50, 51, 52, 46, 47, 48	46
5	50, 51, 52, 46, 47, 48, 49	48
6	51, 52, 46, 47, 48, 49, 50	50, 51
7	52, 46, 47, 48, 49, 50, 51	50, 52

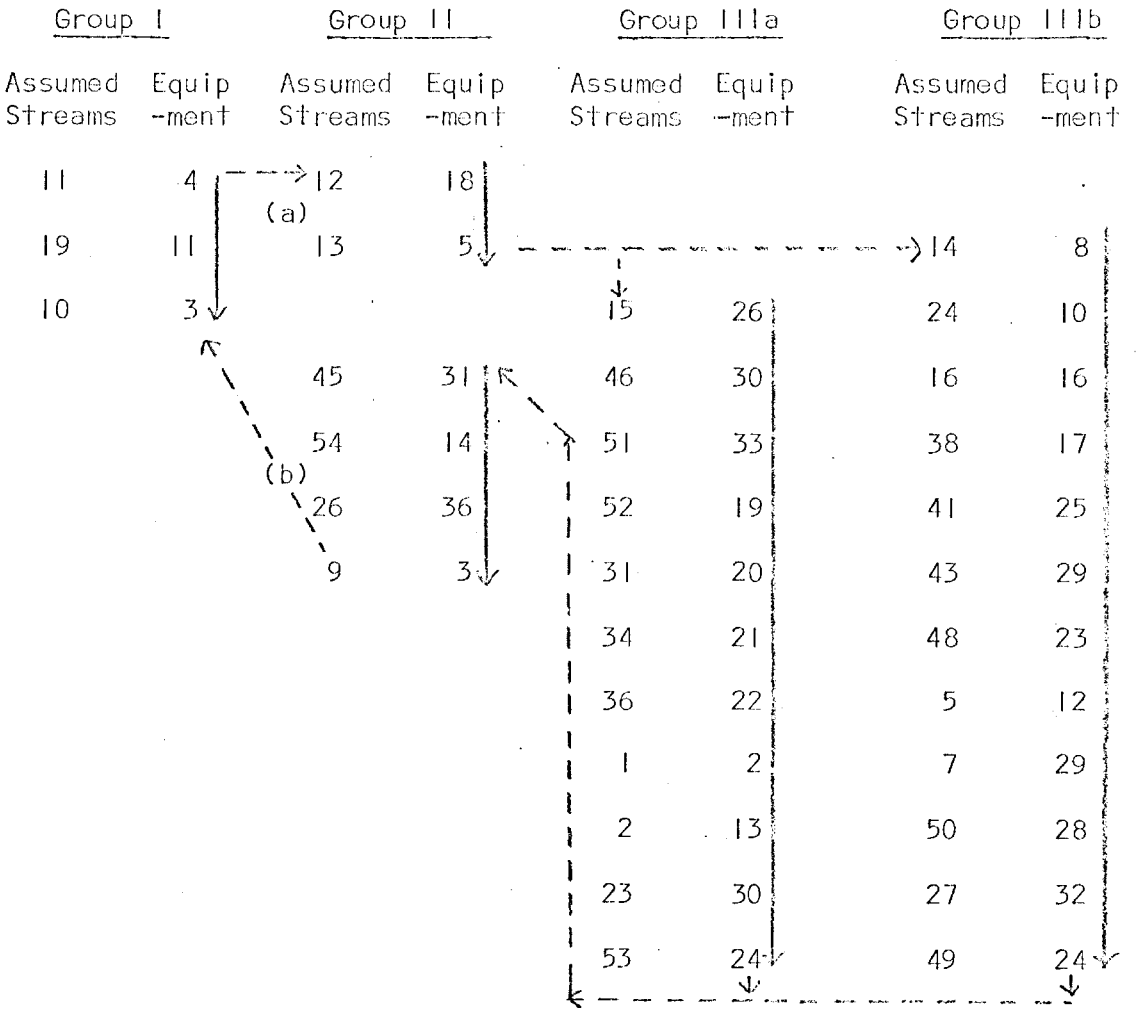
Number of Iterations to Converge

	Convergence Promotion	Direct Substitution
1	N.B. 47	N.B. 74
2	41	75
3	41	75
4	41	75
5	41	75
6	41	75
7	41	75

* See diagram and digraph on p.32.

3.4 Alkylation Process.

In this plant there are many possible sequences of calculation and feasible sets of assumed streams. From the Alkylation Plant diagram on page 38 it is possible to show the necessary structure of these feasible sets. There are two cycles, one of which contains a sub-cycle. However, the easiest way to find the structure of all the possible feasible sets of cut streams is to divide the streams into four groups: -



All the feasible sets may be constructed by using one of the following formulae, where SN refers to stream, e.g. SN11 is Stream 11.

(1) First Order: SN11

(2) Second Order: (SN19 or SN10) + (Any SN from Group II)

(3) Third Order: (SN19 or SN10) + (Any SN from IIIa) + (Any SN from IIIb)

Once a feasible set has been chosen, it is possible to find the appropriate sequence of calculation by using the above table and the digraph of the plant on page 38.

In each group the stream number level with an equipment number is the number of an input stream to that equipment. The stream number immediately below the input stream number is the number of the output stream of that equipment. Thus by following the arrows downwards, the equipments are in the correct sequence within each group. For example, in Group IIIb, Stream 14 is the input to Equipment 8, and Stream 24 is the output from Equipment 8 and the input to Equipment 10; and so it continues to Stream 44 which is one input to Equipment 24. The connections between groups are provided by equipments which have two inputs, Equipments 3 and 24, or two outputs, Equipments 4 and 5. For equipments with two inputs, both must be known before the equipment can be calculated. The dashed lines indicate the connections between groups, and the arrows indicate the direction.

A simple example is the first order case where the assumed stream is Stream 11.

Assuming Stream 11, one can calculate Equipment 4, and after calculating Equipment 4, one can then calculate Equipment 11, which gives Stream 10. However, in order to calculate Equipment 3, and thereby obtain the new value of assumed Stream 11, one also needs Stream 9. But Stream 9 cannot be calculated until all the other equipments have been calculated. Therefore it is necessary to follow the dashed line to Group IIa, then to

Groups IIIa and IIIb which are calculated in parallel, both of them terminating in the calculation of Equipment 24, then to Group IIb, finally terminating in the calculation of Equipment 3 to give Stream II.

This sequence of calculation is as follows: -

(4, 11), (18, 5), (26, 30, 33, 19, 20, 21, 22, 2, 13, 30)
Group I Group IIa Group IIIa

(8, 10, 16, 17, 25, 29, 23, 12, 29, 28, 32) 24 (31, 14, 36) 3
Group IIIb (link to IIb from IIIb) IIb (link to Group I)

The sequence of calculation for the assumed streams 9 and 10 is the same as the above, but Equipment 3 is at the beginning and not at the end.

For other sequences it is necessary to start at other points within the groups, i.e. at the equipments whose input streams have been assumed.

One of the problems of using this plant is that the time per iteration can vary appreciably because there is a search routine in one of the models (CONTO2). This model searches for the correct operating pressure of the condenser and the time taken for this can vary appreciably. However, because of the limitations on the timing device on the I.B.M.7040, it is difficult to measure this variation accurately. For this reason, and the length of time needed to run one case, only two sequences have been examined; one is first order, SN11, and one second order, SN9, 10. Several tests were done with the tolerances used in earlier work* which are rather large. Convergence was attained in all runs in 16 iterations whether or not Convergence Promotion was used.** Other runs were made with lower

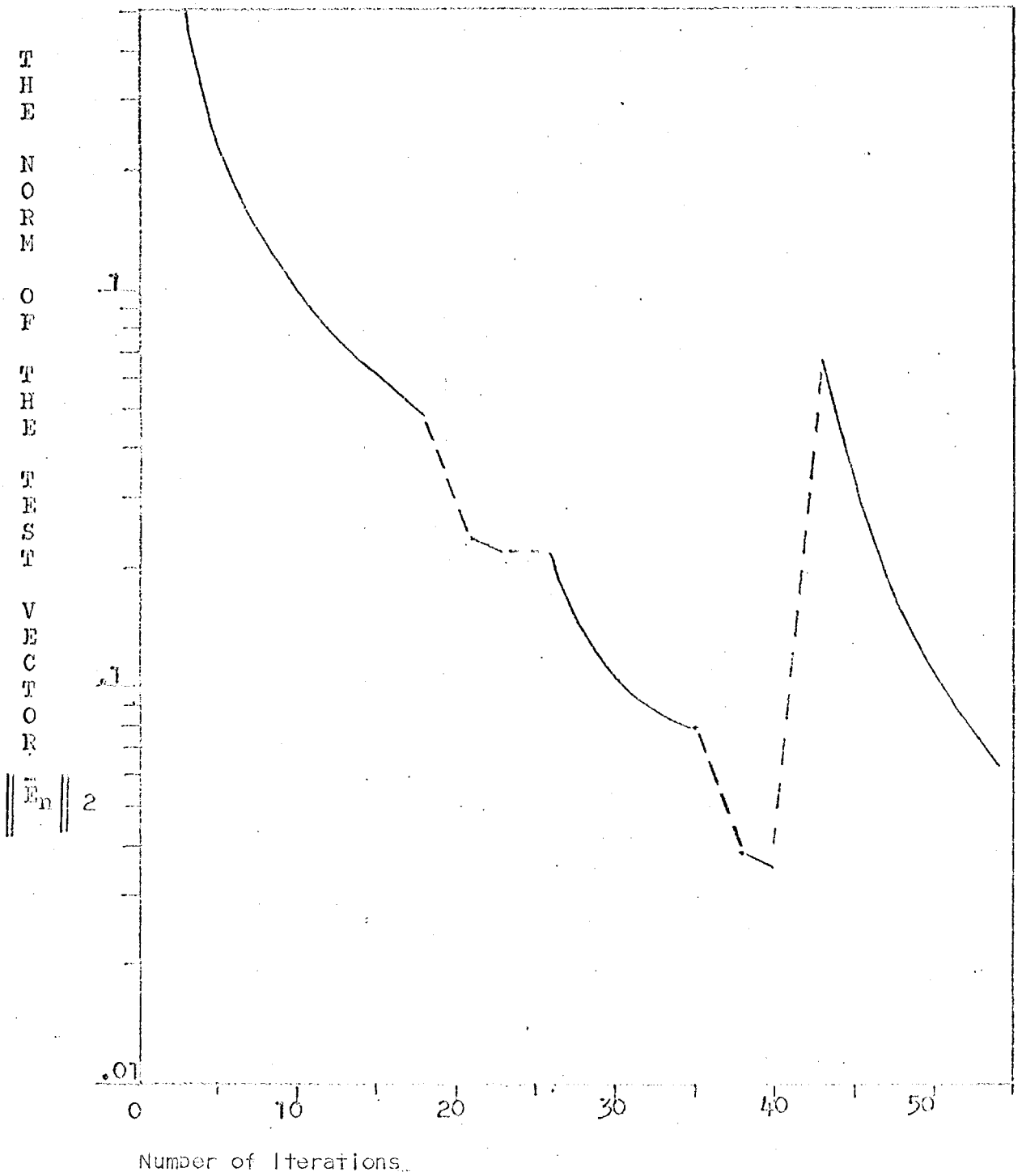
---

*I.D. Shaw (S3).

** The convergence criterion of geometric progression was never met, so a convergence promotion step was never made.

tolerances, .001, but these did not converge within the time allowed (30 mins.), with or without Convergence Promotion. From the graph shown of one of these sequences, SN11, one can see that there is bad promotion at several steps: the error norm either does not decrease, or even increases, after the promotion. The bad jumps come in both cases after the Direct Substitution part of the iterative technique has been working for only a few iterations.

Fig. 3.4 Graph of Alkylation Plant Assumed Stream II.



CHAPTER FOUR

CONCLUSIONS.

4.1 Introduction.

It is now possible to evaluate the various approaches for reducing the computation time of a given modular situation. Each modular simulation has a sequence of calculation and a convergence technique. Reductions in computation time should be possible by choosing an optimal sequence of calculation and an optimal convergence technique. However, the sequence of calculation and convergence technique may not be independent of each other. Indeed their dependence is the *sine qua non* of Longeway's thesis (L3). Therefore the various criteria for selecting an optimal sequence of calculation will be examined in conjunction with the two techniques of convergence, which will also be evaluated to find the optimal technique. Thus it can be found if the optimal sequence of calculation is dependent on the technique of convergence. In addition, since in engineering one is not so much concerned with the unattainable perfect as with the practical solution, the engineering decision whether to seek improvement by a better sequence of calculation, or by a better convergence technique, will be discussed.

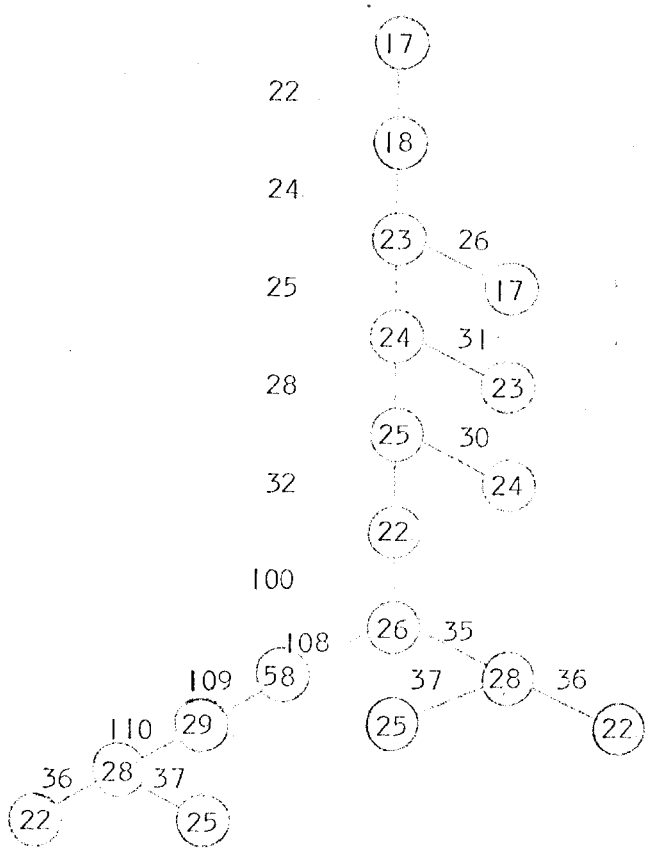
4.2 Selection of the Optimal Sequence of Calculation.

There are two possible approaches to the above problem; one is to use some property of the simulation as an *a priori* criterion, and the other is to use convergence characteristics of the simulation to improve the initial choice of sequence.

The first approach has been dealt with at length in the selected readings, and in Chapter One, and the suggested criteria comprise four different properties of the simulation. Each one of the first three has been associated by Longeway (L3) with a different means of convergence promotion, but as other authors have suggested these criteria, they have

TABLE 4.2.1 The BAYER Wash Section.

(a) Spanning Tree



(45) = Equipment 45

43 = Stream 43

(b) Table

Stream	Number of Cycles	Non-Zero Variables
22	I	21
24	I	20
26	I	18
25	I	20
28	I	20
31	I	17
30	I	12
32	II	18
100	IIII	19
108	II	19
109	II	19
110	II	10
35	II	10
36	II	10
37	II	10

all been compared for effectiveness using both convergence techniques. They all operate by seeking a set of streams such that the sum $\sum p_j$ of the parameters p_j is a minimum, where p_j has the following chosen values:

(a) Direct Substitution.

p_j = Number of cycles in which the stream participates.

(b) Wegstein's Method.

$p_j = 1$

(c) Newton Raphson.

p_j = Number of variables in the stream.

The number of cycles in which each stream takes part can be found quite easily from a 'spanning tree.'*

The number of variables in a stream of the BAYER Wash is a constant of the simulation because each simulation has a fixed Stream List size. However, a variable whose value is zero throughout the whole computation is always converged, and transmits no information to other streams. Therefore the number of non-zero variables is found for each stream from the converged values of the streams.

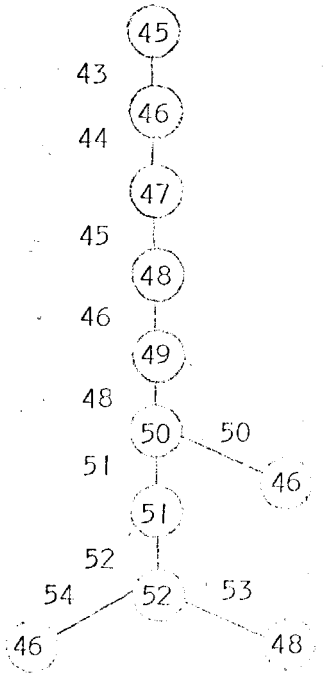
The criteria will be checked by seeing if they differentiate correctly within each group into which the considered plant is divided, and by seeing if they make the correct choice of overall optimum. The values of the criteria for the best and worst case will be checked, and the case or cases selected by the criteria as being the best will be checked to find the number of iterations taken to converge.

---

* See Tables 4.2.1 and 4.2.2.

TABLE 4.2.2 The BAYER Crystalliser Section.

(a) Spanning Tree



(b) Table

Stream	Number of Cycles	Number of Non-Zero Variables
43	-	-
44	11	20
45	11	20
46	110	20
48	110	20
50	1 0	20
51	1 0	20
52	1 0	20
53	0	1
54	1	20

* the 1 signifies the stream takes part in a cycle, the 0 signifies participation in the cycle involving Stream 53 which does not convey information.

4.2.1 Number of Cycles in which each stream occurs as the Optimum Criteria

The minimum value of $\sum p_j$ must be equal to the total number of cycles in the simulation, and this number is an invariant of the particular simulation. By definition, a feasible set must cut all cycles; therefore any set of assumed streams which meets the above criterion must also be a feasible set. The reverse is not necessarily true, but for the three plants below, every feasible set studied does meet the above criterion.

(a) BAYER Wash Section.

$\sum p_j = 7$ for all feasible sets.

(b) BAYER Crystalliser Section.

$\sum p_j = 2$ for all feasible sets if stream 53 is neglected, or

$\sum p_j = 3$ for all feasible sets if stream 53 is included as a recycle stream

(c) The Alkylation Plant.

$\sum p_j = 3$ for all feasible sets

(d) Summary.

The predicted sequences for this criterion are identical to those predicted for the feasible set criterion for these three plants. Therefore both criteria are dealt with in the discussion of the feasible set criterion, 4.2.4.

4.2.2 The Minimum Cut Streams Criterion.

In this and the following sections, the predicted optimal choices made by the respective criteria are indicated as follows: -

e.g. $A < B = C$

where A is chosen in preference to B or to C, and that no differentiation is made between B and C.

(a) BAYER Wash Section.

The preferred order within each group is as follows: -

Group I	Group II	Group III
A = B = C	1 = 2 = 3 = 4	0 < 1 = 2 = 3 = 4 = 5

Only in Group III is any differentiation made, and therefore twelve sequences are chosen out of the seventy-two feasible ones.

If one considers the use of Direct Substitution, then the chosen cases converge as follows: -

		Group II											
		1			2			3			4		
Group I		A	B	C	A	B	C	A	B	C	A	B	C
Group III	0	80	80	81	77	77	80	77	77	77	76	76	76

There is, then, quite a variation in the cases chosen. However, this variation is less than in the whole of the feasible set, and the twelve predicted choices are better than average choices. This can be illustrated by Table 3.2.1 where it can be seen that for Group III, choices 0, 1, 2, are approximately equal and better than choices 3, 4, 5. The Minimum Cut Streams Criterion does not make any distinction between choices 1, 2, 3, 4, 5 in Group III, but it does choose 0, thereby improving the chance of selecting a good criterion.

If one now considers the Convergence Promotion case, one finds the results for the predicted optimum choices from Table 3.2.2: -

		Group II											
		1			2			3			4		
Group I		A	B	C	A	B	C	A	B	C	A	B	C
Group III	0	47	48	49	44	47	48	49	48	44	47	46	44

Although appreciably better than the results for <u>any</u> Direct Substitution case, they are no better than the average case using Convergence Promotion. This can be illustrated by Tables 3.2.2(a) and 3.2.2(b), and if one looks at the empirical preferred order in Group III (from page 55): -

$$1^* \prec 2 \prec 4 \prec 3 \prec 0 \prec 5 \prec 1$$

one finds the predicted optimum choice, '0', is, in fact, worse than average.

(b) Crystalliser Section.

This plant has four sequences, 2, 3, 4, and 5, each with one assumed stream, and three sequences, 1, 6, and 7, with two assumed streams, (see Table 3.4.1). The four with one assumed stream are the ones chosen by the Minimum Cut Streams Criterion as follows: -

$$1 < 2 = 3 = 4 = 5 > 6 = 7$$

The results for the above sequences for both Direct Substitution and Convergence Promotion are in Table 3.4.2.

It can be seen that all sequences except '1' converge in the same number of iterations, taking 75 iterations using Direct Substitution, and 41 iterations using Convergence Promotion. Sequence 1 takes 74 and 47 iterations respectively. Therefore the Minimum Cut Streams Criterion did not choose the optimum for Direct Substitution, but did avoid the worst case for Convergence Promotion.

(c) Alkylation Plant.

This plant has many possible feasible sequences, but only one which meets the Minimum Cut Streams Criterion. This is the sequence which has Stream 11 as its assumed stream. A second order case with Streams 9 and 10 as assumed streams was chosen for comparison.

$$\text{SN } 11 < \text{SN } 9, 10$$

Both runs converged in 16 iterations (see page 69), therefore the criterion is of no benefit in this plant.

(d) Summary.

The Minimum Cut Streams Criterion is shown to be neither necessary nor sufficient. However, it may be advantageous to use a Minimum Cut Stream sequence when Direct Substitution is used. When

Convergence Promotion is used, there is no advantage to be gained by using this criterion to partition the set of feasible sequences. Instead it is easier to choose any feasible sequence, which, of course, includes all the Minimum Cut Stream sequences, and thus not needlessly eliminate valid choices. It is assumed that finding a feasible sequence takes less effort than finding the Minimum Cut Stream sequence or sequences.

4.2.3 The Minimization of the Number of Non-Zero Variables as the Criterion.

This criterion differentiates between streams far more than any of the others. The below predicted orders are based on the number of non-zero variables in each stream given in Tables 4.2.1 and 4.2.2.

(a) The BAYER Wash Section.

Group I	Group II	Group III
$C \lessdot B \lessdot A$	$1 \lessdot 2 \lessdot 3 \lessdot 4$	$0 \lessdot 4 = 5 \lessdot 1 \lessdot 2 = 3$

Optimal Case = C10

Worst Case = A42 or A43

In the BAYER Wash Section the predicted choices are very interesting, particularly when Direct Substitution is used. From the results it is possible to determine a best order of preference within each group (see section 3.2.1).

Group I $A \eqslantless B \eqslantless C$

Group II $4 \lessdot 3 = 2 \lessdot 1$

Group III $0 \eqslantless 2 \eqslantless 1 \lessdot 3 = 4 = 5$

Thus it can be seen that for Groups I and II, the predicted order is the reverse of the best order of preference obtained from the results.

For Group III, an interesting result emerges if one treats the fourth order case, 0, and the fifth order cases 1, 2, 3, 4, 5, separately.

For the fifth order cases, the predicted order for Group III is then: -

$$4 = 5 \lesssim 1 \lesssim 2 = 3$$

This is very nearly the reverse of the fifth order cases' best order in Group III: -

$$2 \leq 1 \lesssim 3 = 4 = 5$$

When Convergence Promotion is used, there is a best preferred order (see section 3.2.2) calculated from the results: -

Group I $\quad C \lesssim A \lesssim B$

Group II $\quad 4 \lesssim 3 \lesssim 2 \lesssim 1$

Group III $\quad 1 \lesssim 2 \lesssim 4 \lesssim 3 \lesssim 0 \lesssim 5$

The correlation with the predicted order is mixed. In Group I the optimal stream, C, is predicted correctly, but the other two streams are reversed. In Group II the predicted order is the reverse of the best order. In Group III there is no obvious correlation at all. It should be pointed out that the actual empirical optimum set using Convergence Promotion, C21, cannot be picked using these best orders. It is also true that the criterion used will not predict this optimum either.

(b) BAYER Crystalliser Section.

For this section, the predicted order is: -

$$1 = 2 = 3 = 4 = 5 = 6 = 7$$

The results are given in section 3.4, and are essentially the same. Only '1' is different from the rest, taking one iteration less for Direct Substitution, and six iterations more for Convergence Promotion.

(c) Alkylation Plant.

For this section the predicted order is: -

$$SN\ 11 < SN\ 9,\ 10$$

This is not the case as both runs converge in 16 iterations.

(d) Summary.

If one takes the set of all feasible sets of assumed streams as a base, then divides it into two groups of fourth and fifth order feasible sets, then one can predict the optimal sets in each group more accurately by maximising the number on non-zero assumed stream variables, than by any other method. For the fourth order group, the predicted set will be A40, and for the fifth order set, the predicted sets will be A42 and A43. A40 and A42 converge in 72 iterations, and are optimal; however, A43 converges in 78 iterations and is not optimal. There seems to be no theoretical reason for this technique, and unfortunately the Crystalliser Section and the Alkylation Plant do not support or contradict the technique appreciably. It does fit the results better than any other approach, and it is the only approach which correctly predicts the best choice in Group II for both Direct Substitution and Convergence Promotion.

4.2.4 The Feasible Set Criterion.

It was shown in section 4.3.1 that the minimisation of the number of cut cycles produced the same sets of assumed streams as the feasible set criterion. Therefore the following comments apply to both criteria.

All the cases tested have satisfied the Feasible Set Criterion. Because of the variation in results, the criterion cannot be considered sufficient to produce the optimal result. However, it may still be

necessary. This seems likely from a theoretical point of view as it is difficult to postulate reasons for an improvement in convergence which would result from using a non-feasible set.

If one considers two examples from the BAYER Process: -

B20		C20	
Assumed Streams:	24, 25, 30, 100	Assumed Streams:	26, 25, 30 100
Sequence of Calculation:	26, 58, 29, 28, 24, 23, 25, 17, 18, 22.	Sequence of Calculation:	26, 58, 29, 28, 17, 18, 24, 23, 25, 22.

Let us consider the case BC20, with assumed streams 24, 26, 25, 30, and 100, and the sequence of calculation being 26, 58, 29, 28, 17, 24, 25, 23, 18, 22. This is a case where only one of 24 and 26 is needed to make a feasible set.

	Convergence Promotion	Direct Substitution
B20	47	77
C20	48	80
BC20	47	77

The Convergence Promotion case for BC20 was promoted on only four streams, but these were the streams of C20, i.e. 26, 25, 30, 100. These results show that the feasible set criterion also is neither necessary nor sufficient.

4.2.5 Random Sequence of Calculation.

One way out of the difficulty of choosing a sequence of calculation could be by using a Monte Carlo, or random, approach. For the small simulation a simple raffle, or pulling the sequence out of a hat, might suffice, while for the more complex simulation, one of the many random number generators might be more appropriate.

Unfortunately none of the three random cases tested converged in less than one hundred iterations.

4.2.6 The Optimal Criterion.

None of the criteria are necessary or sufficient - i.e. none of the criteria tested can select the optimum sequence alone when either Direct Substitution or Convergence Promotion is used. However, some are less equal than others. The Random Criterion produces very bad convergence and should not be used. The Feasible Set Criterion can be used to generate all the sequences which the other crieria select, but this is not necessary and a single feasible sequence can be found very easily from the flowsheet. The Minimum Cut Streams Criterion could be used advantageously with Direct Substitution, but the extra effort is not worthwhile if Convergence Promotion is used.

Minimisation of p_j where p_j is the number of non-zero variables was shown to be of no use for either Direct Substitution or Convergence Promotion. There is a possibility of Maximising p_j within a feasible set, which might be useful, but it has no theoretical backing, and the supporting results may be mere coincidence.

Minimisation of p_j where p_j is the number of cycles in which each stream takes part, produces feasible sets, but it is more difficult to use since all the sets must be produced, whereas using the feasible criterion, only one sequence need be produced.

4.3 Sequence of Calculation or Convergence Promotion? An Engineering Decision.

For the cases investigated, the time differences between the various sequences of calculation were four or five times less than the differences between the same sequences using Direct Substitution and Convergence Promotion. The optimum case in the BAYER Wash Section for Direct Substitution took 76 iterations, while the optimum case for Convergence Promotion* took 38 iterations. Even allowing for the extra calculation involved in Convergence Promotion, this is an appreciable saving.

In addition, a case where there was bad Convergence Promotion,** A21* and B21*, could be improved appreciably and easily by a slight modification of the Convergence Promotion technique. In contrast, where there is a bad sequence of calculation, either when Direct Substitution or Convergence Promotion is used, it is harder to find a better sequence, and the possible improvement is also less.

It therefore seems that the best way to reduce computation time is by improving the Convergence Promotion technique rather than by trying to choose a better sequence of calculation.

This conclusion is supported by the graph of the Alkylation Plant on page 71. Here it should be possible to obtain quite drastic improvements in convergence by increasing the KOUNT, the minimum number of iterations before a Convergence Promotion jump, as was done for the two badly-promoted examples in the BAYER Wash Section.

---

* Not the same sequence as the optimum for Direct Substutution.

** See Chapter Three, page 54.

4.4 Iterative Improvement of Convergence.

After a simulation run has been made with the chosen sequence of calculation, several new things are known. One knows the converged* solution, the number of iterations used to provide the solution, and the change in error norm at every iteration. One can therefore use this information to try and improve the starting point, the convergence promotion technique, and the sequence of calculation.

Improving the starting point is relatively easy: one can either use the new converged values, or, if other information is available, a more complex approach such as an average, or weighted average, of several runs with similar feed streams or operating conditions.

As stated earlier,** the convergence promotion technique can be easily improved by using the graph of the error norm to detect any points of bad promotion. This graph will also show when very good promotion is obtained, thus eliminating the need to attempt improvement.

In contrast, when one tries to improve the sequence of calculation the 'new' results do not really give a pointer to the way of improvement. It is more the case that one has eliminated one of several possible sequences (depending on the criterion used). There is also the inherent possibility that the change may be for the worse, which is compounded if the changes made to the convergence promotion technique are dependent on the sequence of calculation. This is not to say that the use of different

---

* Even if the solution has not converged to the desired tolerance, it may still provide an improved starting point.

** Section 4.3.

sequences of calculation may not be beneficial, only that it is difficult to say that any given untried sequence of calculation will be an improvement. For example, it might be desirable to check a converged solution by repeating the run with a different sequence of calculation. Or, if there were only a few possible sequences and a large number of runs to be made, one might try all the sequences and then use the best for the majority of the runs.

4.5 Summary of Conclusions.

None of the criteria* tested were found to be necessary or sufficient to obtain the 'optimal' sequence of calculation. In fact the 'optimal' sequence cannot generally be found a priori, although when Direct Substitution is used as the convergence technique, the choice of a sequence which has the minimum number of cut streams will yield good convergence.

The comparison of convergence techniques shows that Convergence Promotion always improves the rate of convergence compared to that obtained by the use of Direct Substitution only. It was also found that the empirical optimal sequence varied according to the convergence technique, and that when the technique is Convergence Promotion, a sequence with the minimum number of cut streams does not, in general, give better convergence than a feasible** sequence.

The use of an a priori criterion often gives more than one predicted optimum, so one must still make a choice of sequence at random from those selected. Since it is easier to choose a feasible sequence by definition than one which has a minimum number of cut streams, then a feasible sequence is a more practical choice unless it is expected to converge at a slower rate. As Convergence Promotion is the faster convergence technique, then the practical choice is its use, together with a feasible sequence.

---

* These criteria are defined in Section 1.4.1, p.16, and below.

**The Feasible Sequence Criterion is defined in Section 1.5, p. 23.

Information gathered from previous simulations is better used in improving the starting point and the convergence promotion technique than in trying to obtain a better sequence of calculation. There may, however, be times when it is desirable to use several, or all, of the sequences which meet the chosen criterion.

BIBLIOGRAPHY.

C1 Christensen, J.H.,
"The Use of Structure in Process Design Computation",
Ph.D. Thesis, University of Wisconsin (1967).

C2 Crowe, C.M., A.E. Hamielec, T.W. Hoffman, A.I. Johnson,
P.T. Shannon, D.R. Woods,
"Chemical Plant Simulation", McMaster University (1969).

D1 Dulmage, A.L. and N.S. Mendelsohn,
"Two Algorithms for Bipartite Graphs",
J.Soc. Indust. Appl. Math., Vol.11, No.1, (March 1963).

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

F1 Forder, G.J., and H.P. Hutchinson,
"The Analysis of Chemical Plant Flowsheets",
Chem. Eng. Science, 24, (1969).

H1 Harary, F., R.Z. Norman, D. Cartwright,
"Structural Models - An Introduction to the Theory of
Directed Graphs",

H2 Himmelblau, D.M.,
"Decomposition of Large Scale Systems",
Chem. Eng. Science, 21, 425-438, (1966).

K1 Koppel, L.B., et al.,
"Generalised Computer Program for Flowsheet Calculation
and Process Data Reduction",
Argonne National Laboratory, Chem. Eng. Div.,
Report No. ANL-7197, (April 1966).

L1 Lederman, P.B.,
"Flowsheet Simulation and Beyond",
Chem. Eng., 75, No. 26, pp. 127-132, (1968).

L2 Lee, W., and D.F. Rudd,
"On the Ordering of Recycle Calculations",
AIChE J., 12, 1184, (1966).

L3 Longeway, G.D.,
"On the Group Ordering Problem in Chemical Process Simulation",
Ph.D. Thesis, Sever Instit. of Washington University, (1969).

M1 The Final Year Project, 1966-67, and 1967-68,
"Comprehensive Report on A Digital Computer Simulation of an Alkylation Unit",
Dept. of Chemical Engineering, McMaster University, (1968).

M2 Mosler, H.A.,
"PACER - A Digital Computer Executive Routine for Process Simulation and Design",
M.S. Thesis, Purdue University, (January 1964).

M3 The Final Year Project, 1968-69,
"A Digital Computer Simulation of the Bayer Process for Alumina Extraction",
Dept. of Chemical Engineering, McMaster University, (1969).

N1 Norman, R.L.,
"A Matrix Method for Location of Cycles in a Directed Graph",
AIChE J., 11, 3, pp. 450-452, (1965).

O1 Orbach, O.,
"A Study of the Promotion of Convergence in the Calculation of Complex Chemical Plants",
Ph.D. Thesis, McMaster University, (September 1969).

R1 Rubin, D.I.,
"Generalised Material Balance",
Chem. Eng. Progr., Symposium Series, 58, No. 37, (1962).

R2 Ravicz, A.E., and R.L. Norman,
"Heat and Mass Balancing on a Digital Computer",
C.E.P., 60, 5, (1964).

S1 Sargent, R.W.H., and A.W. Westerberg,
"Speed-Up", in Chemical Engineering Design,
Trans. I Chem. E., 42, T190-97, (1964).

S2 Shannon, P.T., A.I. Johnson, C.M. Crowe, T.W. Hoffman, A.E. Hamielec, and D.R. Woods,
"Computer Simulation of Sulfuric Acid Plant",
Chem. Eng. Progr., 62, No. 6, (1966).

S3 Shaw, I.D.,
"An Appraisal of the MACSIM Simulation Routine in its Application to an Alkylation Plant",
M.S. Thesis, McMaster University, (1969).

S4 Steward, D.V.,
"On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations",
SIAM Review, 4, 4, pp 321-342, (October 1962).

WI Wegstein, J.H.,
"Accelerating Convergence of Iterative Process",
Comm. of A.C.M., 1, 9, (1958).

NOMENCLATURE.

a_{ij}	element of matrix A
$\bar{\bar{A}}$	m x m matrix
c_{ij}	element of matrix C
$\bar{\bar{C}}$	m x n matrix
$\bar{E}_n$	error vector after nth iteration
I	identity matrix
p_j	parameter of stream j
Σp_j	sum of parameters p_j
u	geometric coefficient
$\bar{U}$	vector of coefficients
$\bar{X}$	variable vector
$D(\bar{X})$	diagonal matrix with elements x_i on the diagonal
x_i	element of vector $\bar{X}$
$\bar{X}_c$	converged vector $\bar{X}$
$\bar{X}_s$	solution point
$\bar{X}'_s$	estimate of $\bar{X}_s$

APPENDIX PROGRAM LISTINGS

```
$IPFTC JUNC01
C
C
C     SUBROUTINE JUNC01  (MIXER - MASS BALANCE)
C     REVISED BY T. TOONG     FEB.20,1968
C
C
C     *MAXIMUM VALUE OF NOCOMP, NUMBER OF FLOW COMPONENTS, IS 26
C     *    ,,          NIN,     ,,     INPUT STREAMS,     5
C     *    ,,          NOUT,    ,,     OUTPUT STREAMS,    5
C
C
C     EN, EQUIPMENT VECTOR -
C     1. EQUIPMENT NUMBER
C     2. TEMPERATURE OF OUTPUT STREAMS, DEG F, IF 0., OUTPUT TEMPERATURE
C        ARE SET EQUAL TO TEMPERATURE OF 1ST INPUT STREAM
C     3. PRESSURE OF OUTPUT STREAMS, PSIA, IF 0., OUTPUT PRESSURE
C        ARE SET EQUAL TO PRESSURE OF 1ST INPUT STREAM
C     4. (BLANK)
C     5. (BLANK)
C     6. FRACTION OF TOTAL INPUT IN 1ST OUTPUT STREAM
C     7.              ,,              2ND     ,,
C     8.              ,,              3RD     ,,
C     9.              ,,              4TH     ,,
C     10.             ,,              5TH     ,,
C
C
C     ENC, EQUIPMENT CONTROL VECTOR (OPTIONAL) -
C     1. EQUIPMENT NUMBER
C     2. FLAG
C     3. LENGTH OF EN LIST (FOR PRINTING)
C
C
C     SN, STREAM VECTOR --
C     1. STREAM NUMBER
C     2. STREAM FLAG
C     3. TOTAL FLOW IN LBMOLE/HR
C     4. TEMPERATURE IN DEGREE F
C     5. PRESSURE IN PSIA
C     6. FLOW OF COMPONENT 1 IN LBMOLE/HR
C     7.          ,,      2       ,,
C     8.          ,,      3       ,,
C     9.          ETC.
C
C
C     SNC, STREAM CONTROL VECTOR - NOT REQUIRED
C
C
C
```

```
      SUBROUTINE JUNC01
C     ********MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,KSETS,NIN,NOUT,NE,STRMI(5,30),STRMO(5,30),EN(80,20)
C
C TO CHECK IF INPUT STREAMS ARE NOT MORE THAN 5
      IF(NIN.LE.5) GOTO 10
      WRITE(6,90) NE
      NIN= 5
   10 JJ= NOCOMP+5
C
C
C MASS BALANCE
C (INPUT)
      DO 1 J=6,JJ
      STRMO(NOUT,J)=0.
      DO 1 I=1,NIN
    1 STRMO(NOUT,J)= STRMO(NOUT,J)+STRMI(I,J)
C (OUTPUT)
      DO 2 I=1,NOUT
      STRMO(I,4)= EN(NE,2)
      IF(EN(NE,2).EQ.0.) STRMO(I,4)=STRMI(1,4)
      STRMO(I,5)= EN(NE,3)
      IF(EN(NE,3).EQ.0.) STRMO(I,5)=STRMI(1,5)
      STRMO(I,3)=0.
      DO 2 J=6,JJ
      STRMO(I,J)= STRMO(NOUT,J)*EN(NE,I+5)
    2 STRMO(I,3)= STRMO(I,3)+STRMO(I,J)
C
C
C FORMATS
   90 FORMAT (////1X, 25H**ERROR**IN JUNC01 MODULE, I3,
     173H, NIN EXCEEDS 5, CALCULATION CONTINUES USING THE FIRST FIVE INP
     2UT STREAMS////)
C
      RETURN
      END
```

```
      SUBROUTINE SEPA02
C     THIS MODULE CALCULATES SPLIT OF FEED ACCORDING TO FRACTION OFINLET
C     SOLIDS THAT LEAVES OUT THE TOP AND FRACTION BY WEIGHT OF SOLIDS IN
C     THE BOTTOM STREAM
C
C     *ONE INPUT STREAM AND TWO OUTPUT STREAMS
C     STRMO(1,J) = TOP STREAM
C     STRMO(2,J) = BOTTOM STREAM
C
C     EQUIPMENT VECTOR -
C     1. EQUIPMENT NUMBER
C     2. WT. FRACTION OF FEED AS SOLIDS OUT TOP
C     3. WT. FRACTION OF SOLIDS IN BOTTOM STREAM
C     4. (BLANK)
C     5. (BLANK)
C
C     NO AEN VECTOR
C
C     *********MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,KSETS,NIN,NOUT,NE,STRMI(5,30),STRMO(5,30),EN(80,20)
C     ************************************************************************
      DIMENSION WTSOL(30),AA(30)
      COMMON/BLK1/WTMOL(22)
      JJ=NOCOMP+5
      DO 15 J = 6,JJ
      AA(J) = 0.0
      DO 15 I = 1,NIN
      AA(J) = AA(J) + STRMI(I,J)
   15 CONTINUE
      DO 10 I = 6,JJ
   10 WTSOL(I) = AA(I)*WTMOL(I-5)
      DO 11 I = 9,18
      STRMO(1,I) = AA(I)*EN(NE,2)
   11 STRMO(2,I) = AA(I)-STRMO(1,I)
      WSB=0.0
      DO 12 I=9,18
   12 WSB=WSB+STRMO(2,I)*WTMOL(I-5)
      WL=0.0
      DO 50 I=6,8
   50 WL=WL+WTSOL(I)
      DO 51 I=19,27
   51 WL=WL+WTSOL(I)
      WLB=WSB*(1.0/EN(NE,3)-1.0)
      SPLIQ=WLB/WL
      DO 60 I=6,8
      STRMO(2,I) = AA(I)*SPLIQ
   60 STRMO(1,I) = AA(I)-STRMO(2,I)
      DO 70 I=19,27
      STRMO(2,I) = AA(I)*SPLIQ
   70 STRMO(1,I) = AA(I)-STRMO(2,I)
C     SET TEMP. AND PRESSURES
      DO 80 I=4,5
      STRMO(1,I) = STRMI(1,I)
   80 STRMO(2,I) = STRMI(1,I)
C     CALCULATE TOTAL OUTPUT FLOWS
      STRMO(1,3) = 0.0
      STRMO(2,3) = 0.0
      DO 90 I = 6,JJ
```

```
      STRMO(1,3) = STRMO(1,3) + STRMO(1,I)
   90 STRMO(2,3) = STRMO(2,3) + STRMO(2,I)
      RETURN
      END


$IBFTC DATA
      BLOCK DATA
      COMMON / BLK1 / WTMOL(22)
      DATA WTMOL / 18.020, 105.993, 40.001, 1006.36015, 156.020,
     1             258.180, 177.720, 141.950, 56.080, 79.900, 100.091,
     2             44.011, 119.980, 81.971, 258.180, 159.700,141.950,
     3             600.490, 79.900, 142.048, 44.011, 104.024 /
      END
```

```
0 $IBFTC CP2
1       SUBRØUTINE CØNV40
  C
  C     GEØMETRIC EXTRAPØLATIØN TECHNIQUE
  C     WRITTEN BY Ø. ØRBACH  MCMASTER UNIV.  DEC.  23  1968
  C
  C
  C     EN, EQUIPMENT VECTØR -
  C     1. EQUIPMENT NUMBER
  C     2. NUMBER ØF ITERATIØNS BEFØRE EXTRAPØLATIØN IS
  C        APPLIED IN THE FIRST TIME.
  C     3. RELAXATIØN FACTØR.
  C     4. TØLERANCE ØF THE FRACTIØNAL CHANGE ØF THE
  C        GEØMETRIC CØEFFICIENT (EG).
  C     5. NUMBER ØF ITERATIØN BEFØRE EXTRAPØLATIØN
  C        IS APPLIED .
  C     ********MACSIM CØMMØN DECK, FØR BAYER PRØCESS.  T.T.  JAN.31,1968.
2       CØMMØN  NØCØMP,KSETS,NIN,NØUT,NE,STRMI(5,30),STRMØ(5,30),EN(80,20)
3       CØMMØN  STRMCI(5,10),STRMCØ(5,10),ENC(80,10),AEN(30,30)
4       CØMMØN SN(120,30),AAA(3655),LØØP
  C
5       DIMENSIØN X(180),F(180),XØ(180),FØ(180)
  C
  C     THE ASSUMED STREAMS.
6       DIMENSIØN NS(8)
7       NN=1
10      DATA NS/46,0,0,0,0,0,0,0/
  C
11      LP=EN(NE,2)
12      TC=EN(NE,3)
13      EGT=EN(NE,4)
14      LK=EN(NE,5)
  C
15      NNN=NØCØMP*NN
16      IF(LØØP.GT.1) GØTØ 200
21      WRITE(6,2000) NN,(NS(I),I=1,NN)
26   2000 FØRMAT(/10X,I4,15HASSUMED STREAMS,8I5)
27      EN(NE,6)=1.0
30      DØ 100 K=1,NNN
31      XØ(K)=0.0
32      FØ(K)=0.0
33      X(K)=0.0
34      F(K)=0.0
35    100 CØNTINUE
37    200 CØNTINUE
40      KØUNT=EN(NE,6)
  C
41      IF(KØUNT-2) 50,60,70
  C
42     50 CØNTINUE
43      DØ 51 I=1,NN
44      II=NS(I)
45      DØ 51 J=1,NØCØMP
46      K=(I-1)*NØCØMP+J
47      XØ(K)=SN(II,J+5)
```

```
 50      51 CØNTINUE
 53         GØTØ 79
    C
    C
 54      60 CØNTINUE
 55         SFO=0.0
 56         DØ 61 I=1,NN
 57         II=NS(I)
 60         DØ 61 J=1,NØCØMP
 61         K=(I-1)*NØCØMP+J
 62         X(K)=SN(II,J+5)
 63         FO(K)=0.0
 64         IF(ABS(XO(K)).LT.(1.0E-10)) GØTØ 61
 67         FO(K)=(X(K)-XO(K))/XO(K)
 70         SFO=SFO+FO(K)*FO(K)
 71      61 CØNTINUE
 74         SFO=SQRT(SFO)
 75         GØTØ 79
    C
    C
 76      70 CØNTINUE
 77         SF=0.0
100         DØ71 I=1,NN
101         II=NS(I)
102         DØ 71 J=1,NØCØMP
103         K=(I-1)*NØCØMP+J
104         X1=SN(II,J+5)
105         F(K)=0.0
106         IF(ABS(X(K)).LT.(1.0E-10))  GØTØ 71
111         F(K)=(X1-X(K))/X(K)
112         SF=SF+F(K)*F(K)
113      71 CØNTINUE
116         SF=SQRT(SF)
117         EG=SF/SFO
120         SS=1.-EG
    C
121         WRITE(6,500) LØØP,SF,SS,EG
122    500 FØRMAT(//20X,6H LØØP=,I5,3X,3HSF=,E12.5,3X,3HSS=,E12.4,3X,3HEG=,
           1F9.4)
    C
123         IF(LØØP.GE.LP.AND.KØUNT.GT.LK) GØTØ 80
    C
126      90 CØNTINUE
127         SFO=SF
130         EGO=EG
131         DØ 75 I=1,NN
132         DØ 75 J=1, NØCØMP
133         II=NS(I)
134         K=(I-1)*NØCØMP+J
135         XO(K)=X(K)
136         X(K)=SN(II,J+5)
137      75 FO(K)=F(K)
142      79 EN(NE,6)=EN(NE,6)+1.0
143         RETURN
    C
```

```
144      80 CØNTINUE
145         CEG=ABS((EG-EGO)/EGO)
146         IF(CEG.GT.EGT ) GØTØ 90
151      84 WRITE(6,506) LØØP,KØUNT
152     506 FØRMAT(/40X,27HAPPLY CØNVERGENCE PRØMØTIØN,3X,5HLØØP=,I4,3X,
           16HKØUNT=,I4/)
     C
153         DØ 81 I=1,NN
154         II=NS(I)
155         DØ 81 J=1, NØCØMP
156         K=(I-1)*NØCØMP+J
157         B=X(K)*F(K)
160         BB=FO(K)-F(K)
     C
     C
161         BC=BB/FO(K)
162         IF(BC.LT.(1.E-5)) GØTØ 82
165         IF(BC.GE.2.0) GØTØ 82
     C
     C
170         B=F(K)*(X(K)-XO(K))/BB
171      82 CØNTINUE
172         XO(K)=X(K)
173         FO(K)=F(K)
174         SN(II,J+5)=X(K)+TC*B
175      81 CØNTINUE
200         EN(NE,6)=1.0
201         RETURN
202         END
```

```
$IBFTC SETRC1
C
C      THIS MODULE FIXES MOLAR FLOWRATE TO CALCINE (STRM49) AND
C      RECYCLES REMAINING FLOW TO THE SECONDARY CRYSTALLIZER DEPENDING
C      ON ITS RESIDENCE TIME
C      EN, EQUIPMENT VECTOR -
C      1. EQUIPMENT NUMBER
C      2.EQUIPMENT NO. WHERE TOTAL FLOW IS SPECIFIED  PD. MOLES / HR.
C      3. (BLANK)
C      4. (BLANK)
C      5. TOTAL MOLAR FLOW RATE FOR STR18 = 543.0 PD MOLES / HR
C      6. (BLANK)
C      7. (BLANK)
C      8. TOTAL FLOW INTO PRECI3 ( PD. MOLES / HR.)
C      9. LOCATION OF STRM 5
C      10.LOCATION OF STRM 9
C
C      SN, STREAM VECTOR -
C      1. STREAM NUMBER
C      2. STREAM  FLAG
C      3. TOTAL FLOW IN LBMOLE/HR
C      4. TEMPERATURE IN DEGREE F
C      5. PRESSURE IN PSIA
C      6. FLOW OF COMPONENT 1 IN LBMOLE/HR
C      7.            ,,       2       ,,
C      8.            ,,       3       ,,
C      9.           ETC.
C
      SUBROUTINE SETRC1
C     ********MACSIM COMMON DECK, FOR BAYER PROCESS.  T.T.  JAN.31,1968.
      COMMON  NOCOMP,NSTS,NIN,NOUT,NR,STRMI(5,30),STRMO(5,30),EN(80,20)
      COMMON  STRMC(5,10),STRFCO(5,10),ENC(80,10),AEN(80,30)
      COMMON  SR(120,30), SRC(120,10), KPM(80,10), KUS(120), KPS(3)
C     ***********************************************************************
      KK = INT(EN(NE,2))
      KL = INT(EN(NE,9))
      KM = INT(EN(NE,10))
      STRMO(1,3) = EN(NE,5)
      STRMO(2,3) = EN(KK,8)-SR(KL,3)-SR(KM,3)
      IF(STRMO(2,3))21,21,5
    5 IF(STRMO(2,3)+STRMO(1,3)-STRMI(1,3))20,10,10
   10 STRMO(3,3) = 0.0
      STRMO(2,3) = STRMI(1,3)-STRMO(1,3)
      GO TO  25
   20 STRMO(3,3) = STRMI(1,3)-STRMO(1,3)-STRMO(2,3)
      GO TO 25
   21 STRMO(2,3) = 0.0
      STRMO(3,3) = STRMI(1,3)-STRMO(1,3)
   25 JJ = 5+NOCOMP
      DO 30 I = 1,NOUT
      DO 30 J = 6,JJ
      STRMO(I,J) = STRMO(1,3)*STRMI(1,J)/STRMI(J,3)
C     SET TEMPERATURES AND PRESSURES OF OUTPUT STREAMS
      STRMO(I,4) = STRMI(1,4)
   30 STRMO(I,5) = STRMI(1,5)
      RETURN
```

```
0 $IBFTC CRYS01
1       SUBRØUTINE CRYS01
  C     HYDRATE WASHING SECTIØN,1969
  C     EN VECTØR*****************
  C      ***************************************************************************
  C     1. NUMBER ØF EQUIPMENT
  C     2. TEMP. 145 F(4/2/69)
  C     3. RATE CØNSTANT KCA(0.3/HR.)
  C     4. VØL. ØF CRYSTALLIZERS(9*33288.=299592. CU.FT.)
  C     5. CST-EQUILM. CØNC ØF AL2Ø3.3H2Ø(GPL)
  C     6. DENSITY ØF LIQUØR(GM/CC)
  C     7. DENSITY ØF SØLID(GM/CC)
  C     8. STARTING VALUE,THEN STØRES REQD. AVG.MØLAR FLØWRATE FØR SETR01
  C     9. TØW
  C
  C     ********MACSIM CØMMØN DECK, FØR BAYER PRØCESS.  T.T.  JAN.31,1968.
2       CØMMØN  NØCØMP,KSETS,NIN,NØUT,NE,STRMI(5,30),STRMØ(5,30),EN(80,20)
  C
  C     CALCULATIØN ØF Q THE LIQUØR FLØW RATE(CU.FT./HR.)
  C     WTMØL ARE MØLECULAR WEIGHTS
3       CØMMØN/BLK1/WTMØL(22)
  C     AMAS1 IS INFLØW ØF SN 6,7,8 (LB./HR.)
4       AMAS1=0.0
5       DØ 10 J=6,8
6       M=J-5
7    10 AMAS1=AMAS1+STRMI(1,J)*WTMØL(M)
  C     AMAS2 IS INFLØW ØF SN 19-27 INCL.
11      AMAS2=0.0
12      DØ 20 J=19,27
13   20 AMAS2=AMAS2+STRMI(1,J)*WTMØL(M)
  C     AMAS IS TØTAL INFLØW ØF LIQUØR(LB./HR.)
15      AMAS=AMAS1+AMAS2
  C      ***************************************************************************
  C     DENSITY AS ØF MARCH 13,1969
  C     CAU=CØNC. CAUSTIC AS CARBØNATE
16      CAU=(STRMI(1,8)+STRMI(1,19))*52.9945
  C     AL=SØDIUM ALUMINATE AS AL2Ø3
17      AL=STRMI(1,19)*51.
20      RATIØ=AL/CAU
21      SØD=STRMI(1,7)*105.989
22      TTS=CAU+SØD
  C     RHØ1=DENSITY(GM/CC) CØRRECTED FØR CAUSTIC
23      RHØ1=1.034/(1.-(0.77*TTS+0.5*AL)/AMAS)
  C     RHØ2=DENSITY(GM./CC) CØRRECTED FØR CAUSTIC AND TEMP.
24      RHØ2=RHØ1-((STRMI(1,4)-32.)/1.8-25.)*0.00077
25      RHØ=RHØ2*62.4
26      Q=AMAS/RHØ
27      IF(EN(NE,10).GT.0.1)GØ TØ 50
  C     SPECIFIED TØW IN EN VECTØR
32      TØW=EN(NE,9)
  C     TØTAL MØLAR FLØW FØR TØW=TØT=VØLUME CRYST./TØW
33      TØT=EN(NE,4)/TØW
  C     AVG=(LB.MØLES/CU.FT.)
34      AVG=STRMI(1,3)/Q
35      REQD=AVG*TØT
```

```
      C       STØRE REQD IN EN(NE,8) FØR SETR01
   36         EN(NE,8)=REQD
      C       *  TAU  IS  THE  RESIDENCE  TIME FRØM ACTUAL  FLØW
   37         TAU=EN(NE,4)/Q
   40         TØW=TAU
      C       CAL=CØNC. ØF NAALØ2 AS AL2Ø3
   41         CAL=AL/Q
   42         E=(CAU/Q)*454./28.313
   43         F=STRMI(1,4)
   44         DRVC(E,F)=(16.07-0.552*E+0.000914*E*E-0.0282*F
             1 +0.00376*E*F)
   45         EQUILM=DRVC(E,F)
   46         CØN = EN(NE,3)*TØW
   47         C=(CAL+CØN*EQUILM*28.313/454.)/(1.+CØN)
   50         CHA=Q*(CAL-C)
      C       2NAALØ2+4H2Ø=2NAØH+AL2Ø3.3H2Ø
   51         GØ TØ 60
   52      50 RATIN=AL/CAU
   53         RATØUT=EN(NE,10)
   54         CHA=(RATIN-RATØUT)*CAU*Q
   55         IF(CHA.LE.0.) CHA=0.
      C       AL2Ø3 PPTD.(LB.MØLES/HR.)
   60      60 DEL1=CHA/102.
      C       WTAL=WT AL2Ø3.3H2Ø(LB./MIN.)
   61         WTAL=DEL1*156./60.
      C       NAALØ2 REMØVED (LB.MØLES/HR.)
   62         DEL2 = 2.0*DEL1
      C       H2Ø REMØVED(LB.MØLES/HR.)
   63         DEL3=4.0*DEL1
      C       NAØH ADDED(LB.MØLES/HR.)
   64         JJ=NØCØMP+5
   65         DØ 30 J=6,JJ
   66      30 STRMØ(1,J)=STRMI(1,J)
   70         STRMØ(1,6)=STRMØ(1,6)-DEL3
   71         STRMØ(1,8)=STRMØ(1,8)+DEL2
   72         STRMØ(1,10)=STRMØ(1,10)+DEL1
   73         STRMØ(1,19)=STRMØ(1,19)-DEL2
   74         STRMØ(1,3)=0.0
   75         DØ 11 J=6,JJ
   76      11 STRMØ(1,3)=STRMØ(1,3)+STRMØ(1,J)
  100         STRMØ(1,4)=STRMI(1,4)
  101         STRMØ(1,5)=STRMI(1,5)
  102         CAU1 = (STRMØ(1,8)+STRMØ(1,19))*53.
  103         AL1 = (STRMØ(1,19))*51.0
  104         RATIØ1 = AL1/CAU1
  105         DELRAT = RATIØ-RATIØ1
  106     103 FØRMAT(1HO,7H      Q=,E10.7)
  107     202 FØRMAT(1HO,7H    CAL=,E10.7)
  110     203 FØRMAT(1HO,7H      C=,E10.7)
  111     205 FØRMAT(1HO,7H   DEL1=,E10.7)
  112     206 FØRMAT(1HO,7HDELRAT=,E10.7)
  113     207 FØRMAT(1HO,7H   WTAL=,E10.7)
  114     209 FØRMAT(1HO,7H RATIØ=,E10.7)
  115     211 FØRMAT(1HO,7H    RHØ=,E10.7)
  116     212 FØRMAT(1HO,7HRATIØ1=,E10.7)

  117     220 FØRMAT(1HO,7HEQUILM=,E10.7)
  120         RETURN
  121         END
```

19.00000	1.00000	97288.15820	145.00000	0.00000
87989.79688	862.70230	3516.66519	0.00430	1166.65480
0.28700	0.30110	0.00010	0.00040	0.01170
0.00170	0.00190	0.00220	2065.88101	5.70580
0.00000	3.94510	0.00000	60.31920	35.40920
1580.47620	-0.00000			
21.00000	1.00000	95129.73438	145.00000	0.00000
84277.14160	887.84820	3550.56451	0.00360	2846.94299
0.28700	0.29940	0.00000	0.00030	0.00990
0.00140	0.00160	0.00180	2100.51550	5.16750
0.00000	3.59070	0.00000	41.01750	27.85760
1438.49260	0.00000			
22.00000	0.00000	198631.15625	145.00000	0.00000
177964.12305	1799.88004	7214.97131	0.00802	4074.94382
0.58277	0.60966	0.00010	0.00071	0.02193
0.00315	0.00355	0.00466	4283.80322	11.17970
0.00000	7.74815	0.00000	104.19231	65.04962
3104.04163	0.00000			
23.00000	1.00000	6125.00000	0.00000	0.00000
6125.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000			
24.00000	0.00000	9492.58044	145.00000	0.00000
7190.14508	72.89947	292.22369	0.00321	1629.97278
0.23311	0.24387	0.00004	0.00023	0.00877
0.00126	0.00142	0.00162	173.50433	0.45280
0.00000	0.31382	0.00000	4.22004	2.63467
125.72115	0.00000			
25.00000	0.00000	9747.22925	145.00000	0.00000
7505.02545	64.98287	260.46933	0.00323	1642.71069
0.23493	0.24578	0.00004	0.00029	0.00854
0.00127	0.00143	0.00164	154.66243	0.40363
0.00000	0.27974	0.00000	3.76176	2.34855
112.06832	0.00000			
26.00000	0.00000	6213.25850	145.00000	0.00000
5697.18585	49.32959	197.74165	0.00012	61.34604
0.00877	0.00918	0.00009	0.00001	0.00033
0.00003	0.00005	0.00006	117.40675	0.30640
0.00000	0.21235	0.00000	2.85561	1.78282
85.07287	0.00000			
28.00000	0.00000	9552.48694	145.00000	0.00000
7503.36127	51.68544	207.18545	0.00309	1572.22659
0.22485	0.23523	0.00004	0.00027	0.00846
0.00121	0.00137	0.00157	125.01389	0.32104
0.00000	0.22250	0.00000	2.99199	1.86797
89.13581	0.00000			
30.00000	0.00000	8085.80133	0.00000	0.00000
7774.08868	33.39002	133.84673	0.00001	3.93057
0.00056	0.00059	0.00000	0.00000	0.00002
0.00000	0.00000	0.00000	79.46990	0.20740
0.00000	0.14374	0.00000	1.93290	1.20675
57.58385	0.00000			
31.00000	0.00000	6467.90674	145.00000	0.00000
6012.06622	41.41294	166.00729	0.00015	74.08398
0.01060	0.01108	0.00000	0.00001	0.00040
0.00006	0.00006	0.00007	98.56485	0.25723
0.00000	0.17828	0.00000	2.39733	1.49671

71.42004	0.00000			
32.00000	0.00000	9893.57678	0.00000	0.00000
8007.77374	34.39371	137.87009	0.00309	1568.29602
0.22429	0.23464	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	81.85872	0.21363
0.00000	0.14806	0.00000	1.99100	1.24303
59.31479	0.00000			
34.00000	2.00000	3830.30643	0.00000	0.00000
2254.17426	1.37231	5.50101	0.00308	1563.00952
0.22353	0.23385	0.00004	0.00027	0.00841
0.00121	0.00136	0.00156	3.26616	0.00852
0.00000	0.00591	0.00000	0.07944	0.04960
2.36666	0.00000			
35.00000	0.00000	20286.88599	0.00000	0.00000
19856.91235	46.64629	186.98560	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	111.02047	0.28974
0.00000	0.20080	0.00000	2.70028	1.68585
80.44538	0.00000			
36.00000	0.00000	17907.14355	0.00000	0.00000
17591.81519	34.20887	137.12914	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	81.41879	0.21248
0.00000	0.14726	0.00000	1.98030	1.23635
58.99602	0.00000			
37.00000	0.00000	8426.69111	0.00000	0.00000
8278.50122	16.09829	64.53136	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	38.31473	0.09999
0.00000	0.06930	0.00000	0.93191	0.58161
27.76283	0.00000			
42.00000	2.00000	186854.49414	145.00000	0.00000
168647.50391	1709.88397	6854.21454	0.00481	2444.95917
0.34966	0.36581	0.00006	0.00043	0.01316
0.00189	0.00213	0.00244	4069.60815	10.62071
0.00000	7.36074	0.00000	98.98258	61.79707
2948.83600	0.00000			
100.00000	0.00000	27800.71997	0.00000	0.00000
25599.58887	68.60258	274.99923	0.00309	1568.29602
0.22429	0.23464	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	163.27751	0.42612
0.00000	0.29532	0.00000	3.97130	2.47937
118.31080	0.00000			
108.00000	0.00000	3752.45590	0.00000	0.00000
2142.57874	5.03318	20.17591	0.00308	1563.00952
0.22353	0.23385	0.00004	0.00027	0.00841
0.00121	0.00136	0.00156	11.97921	0.03126
0.00000	0.02167	0.00000	0.29136	0.18190
8.68013	0.00000			
109.00000	0.00000	9877.45520	0.00000	0.00000
8267.57874	5.03318	20.17591	0.00308	1563.00952
0.22353	0.23385	0.00004	0.00027	0.00841
0.00121	0.00136	0.00156	11.97921	0.03126
0.00000	0.02167	0.00000	0.29136	0.18190
8.68013	0.00000			
110.00000	0.00000	6047.14923	0.00000	0.00000
6013.49448	3.66087	14.67491	0.00000	0.00000

0.00000	0.00000	0.00000	8.71305	0.02274
0.00000	0.01576	0.00000	0.21192	0.13231
6.31347	0.00000			

CD TOT 0120

19.00000	1.00000	97288.15820	145.00000	0.00000
87989.79688	892.70230	3516.66519	0.00430	1166.65480
0.28700	0.30110	0.00010	0.00040	0.01170
0.00170	0.00190	0.00220	2065.58101	5.70580
0.00000	3.94510	0.00000	60.31920	35.40920
1580.47620	-0.00000			
21.00000	1.00000	95129.73438	145.00000	0.00000
84277.14160	887.84820	3500.56451	0.00360	2846.94299
0.28700	0.29940	0.00000	0.00030	0.00990
0.00140	0.00160	0.00180	2100.51550	5.16750
0.00000	3.59070	0.00000	41.01750	27.85760
1438.49260	0.00000			
22.00000	0.00000	196115.38086	145.00000	0.00000
175595.56445	1783.99961	7150.95251	0.00802	4074.94382
0.58277	0.60968	0.00010	0.00071	0.02193
0.00315	0.00355	0.00406	4245.79277	11.08051
0.00000	7.67940	0.00000	103.26781	64.47243
3076.49933	0.00000			
23.00000	1.00000	6125.00000	0.00000	0.00000
6125.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000			
24.00000	0.00000	9490.22754	145.00000	0.00000
7186.75348	73.01163	292.67330	0.00321	1629.97754
0.23311	0.24387	0.00004	0.00028	0.00877
0.00126	0.00142	0.00162	173.77128	0.42350
0.00000	0.31430	0.00000	4.22653	2.63872
125.91458	0.00000			
25.00000	0.00000	9622.10693	145.00000	0.00000
7325.59314	70.97270	284.09922	0.00323	1642.72963
0.23493	0.24578	0.00004	0.00029	0.00884
0.00127	0.00143	0.00164	168.68052	0.44022
0.00000	0.30509	0.00000	4.10271	2.56142
122.22582	0.00000			
26.00000	0.00000	6102.12671	145.00000	0.00000
5546.16174	53.65729	215.08978	0.00012	61.34675
0.00877	0.00918	0.00000	0.00001	0.00033
0.00005	0.00005	0.00006	127.79699	0.33328
0.00000	0.23098	0.00000	3.10614	1.93923
92.53642	0.00000			
28.00000	0.00000	9366.10913	145.00000	0.00000
7234.87854	60.55636	242.74528	0.00310	1572.56123
0.22490	0.23528	0.00004	0.00027	0.00846
0.00121	0.00137	0.00157	144.12712	0.37614
0.00000	0.26068	0.00000	3.50552	2.18857
104.43444	0.00000			
30.00000	0.00000	8952.76271	0.00000	0.00000
8448.52539	54.27638	217.57949	0.00001	3.93140
0.00056	0.00059	0.00000	0.00000	0.00002
0.00000	0.00000	0.00000	129.18523	0.35714
0.00000	0.23366	0.00000	3.14209	1.96168
93.60755	0.00000			
31.00000	0.00000	7962.09137	145.00000	0.00000
7322.97839	61.29376	245.70122	0.00015	74.09974
0.21960	0.21109	0.00000	0.00001	0.00040
0.00006	0.00006	0.00007	145.88217	0.38072
0.00000	0.26386	0.00000	3.54820	2.21522

105.70615	0.00000			
32.00000	0.00000	9592.36560	0.00000	0.00000
7574.68573	48.66431	195.07502	0.00309	1568.62982
0.22434	0.23469	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	115.82347	0.30227
0.00000	0.20949	0.00000	2.81710	1.75878
83.92562	0.00000			
34.00000	2.00000	3812.39435	0.00000	0.00000
2216.86047	2.86722	11.49348	0.00309	1568.62982
0.22434	0.23469	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	6.82412	0.01781
0.00000	0.01234	0.00000	0.16598	0.10362
4.94475	0.00000			
35.00000	0.00000	25447.50342	0.00000	0.00000
24262.35742	128.24667	514.08761	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	305.23336	0.79659
0.00000	0.55208	0.00000	7.42400	4.63497
221.17196	0.00000			
36.00000	0.00000	21359.09204	0.00000	0.00000
20567.46143	92.39028	370.35426	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	219.89343	0.57387
0.00000	0.39772	0.00000	5.34833	3.33909
159.33467	0.00000			
37.00000	0.00000	10051.33752	0.00000	0.00000
9650.57007	43.47778	174.28436	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	103.47926	0.27006
0.00000	0.18716	0.00000	2.51686	1.57123
74.98102	0.00000			
42.00000	2.00000	186625.15234	145.00000	0.00000
168408.81055	1710.89796	6858.27917	0.00481	2444.96628
0.34966	0.36581	0.00006	0.00043	0.01316
0.00189	0.00213	0.00244	4072.02148	10.62700
0.00000	7.36510	0.00000	99.04128	61.83371
2950.58472	0.00000			
100.00000	0.00000	29097.82373	0.00000	0.00000
26249.89233	136.73528	556.13211	0.00309	1568.62982
0.22434	0.23469	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	330.19682	0.86173
0.00000	0.59723	0.00000	8.03117	5.01404
23 .26044	0.00000			
108.00000	0.00000	3656.32025	0.00000	0.00000
1984.53481	10.48861	42.04450	0.00309	1568.62982
0.22434	0.23469	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	24.96342	0.06515
0.00000	0.04515	0.00000	0.60717	0.37907
18.08848	0.00000			
109.00000	0.00000	9775.32007	0.00000	0.00000
810 .53479	10.48861	42.04450	0.00309	1568.62982
0.22434	0.23469	0.00004	0.00027	0.00844
0.00121	0.00137	0.00156	24.96342	0.06515
0.00000	0.04515	0.00000	0.60717	0.37907
18.08848	0.00000			
110.00000	0.00000	5962.92627	0.00000	0.00000
5892.57432	7.62140	30.55102	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000

0.00000	0.00000	0.00000	18.13930	0.04734
0.00000	0.03281	0.00000	0.44119	0.27545
13.14373	0.00000			

CD TOT 0120

```
  RUN   A  ONE  THREE
 WASH    SECTION
  113     0     0  100     0     0  100     0011110110100000
   58     5    58  110    10     0   27     0     0   22
   12
   17 JUNC01              21    19   26   -22
   18 SEPA02              22   -42   -24
   22 JUNC01              32    36  -100
   23 SEPA02              24    31   -26   -25
   24 SEPA02              25    30   -31   -28
   25 SEPA02              37    28   -30   -32
   26 SEPA02             100   -35  -108
   28 JUNC01              35   110   -36   -37
   29 SEPA02             109  -110   -34
   40 CONV40               1    -1
   50 TIMN01               2    -2
   58 JUNC01             108    23  -109
   11
17.0
1.0
18.             0.6             0.6
0.
22.0
1.0
23.0            0.036           0.60
0.0
24.0            0.045           0.60
0.0
25.0            0.0325          0.60
0.0
26.             0.0             0.855
0.
28.0
0.68            0.32
29.0            0.0             0.855
0.
40.0            8.0             0.7             0.005           5.0
0.0
58.0
1.0
    0
    3
      19.00000        1.00000     97288.15820      145.00000        0.00000
   87989.79688      862.70230      3516.66519        0.00430     1166.65480
       0.28700        0.30110         0.00010        0.00040        0.01170
       0.00170        0.00190         0.00220     2065.88101        5.70580
       0.00000        3.94510         0.00000       60.31920       35.40920
    1580.47620       -0.00000
      21.00000        1.00000     95129.73438      145.00000        0.00000
   84277.14160      887.84820      3500.56451        0.00360     2846.94299
       0.28700        0.29940         0.00009        0.00030        0.00990
       0.00140        0.00160         0.00180     2190.51550        5.16750
       0.00000        3.59070         0.00000       41.01750       27.85750
    1438.49260        0.00000
23.0            1.0             6125.0          0.0             0.0
6125.0          0.0             0.0             0.0             0.0
0.0             0.0             0.0             0.0             0.0
0.0             0.0             0.0             0.0             0.0
0.0             0.0             0.0             0.0             0.0
```

```
0.0             0.0
    0
   27
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001
   0
   0
   5   12
   29    2   28    2   18   -2   23    2   24    2
   25    2   17    2   22    2   26    2   58    2
   40    2   50    3

$IBSYS
```

CD TOT 0076

```
 CRYSTALLISER  SECTION
RUN  ONE
   1     0     0  100     0     0    20     0011101101000001
  52     5    52   72    10     0    27     0     0    22
  10
  40 CONV40              1    -1
  41 TIMN01              1    -1
  45 SEPA02             42   -43   -67
  46 JUNC01             43    54    50   -44
  47 CRYS01             44   -45
  48 JUNC01             45    53   -46
  49 SEPA02             46   -72   -48
  50 JUNC01             48   -50   -51
  51 SEPA02             51   -52   -68
  52 SETR01             52   -49   -54   -53
   9
40.0            8.0             0.7             0.005           5.0
0.0
45.0            0.138           0.6
0.0
46.0
1.0
47.0            145.0           0.0380          299592.         0.3275
                                269110.9297     4.
48.0
1.0
49.0            0.0             0.29
0.0
50.0
0.65            0.35
51.0            0.65            0.44            0.              0.
0.0
52.00           47.0            0.0             0.0             540.
0.0             0.0             0.0             48.0            50.0
   0
   1
      42.05000        1.00006    194694.71680      145.00000        0.00000
  176123.83203     1749.36125      7012.46246        0.00461     2444.96671
       0.34966        0.36581         0.00006        0.00043        0.01316
       0.00189        0.00213         0.00244     4163.56604       10.86591
       0.00000        7.53068         0.00000      101.26786       63.22382
    3016.91782        0.00000
   0
  27
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001         0.00001         0.00001         0.00001
0.00001         0.00001
   0
   0
   5   10
  45    1   48    2   49    2   50    2   51    2
  52    2   46    2   47    2   40    2   41    3

$IBSYS
```

Sample Data Set

```
 SHELL ALKYLATION SIMULATION
BASE CASE EXCEPT OLEFIN FEED INCREASED BY 1.2   FEB 27  1970
    2     0     0  100     0     0    50     0 111111111 1 11
   36     7    36   60    10    20    31     0     0   26
   36
    1 READIN              44   -44
    2 DISOB1  DIB          1    -2    -3    -4   -21
    3 REAC25 REACTR        9    10   -11
    4 SETL01              11   -12   -19
    5 EXCH14 PR SEP       13    27   -14   -15   -20
    6 JUNC02 OLEFIN       22    -8
    7 CONV40              44   -44
    8 COMPRS              14   -24
    9 COMT02              14   -24
   10 JUNC02  PUMP         24   -16
   11 MAKEUP  ACID         19    17   -10   -18
   12 DEPRP3 DEPROP         5    -6    -7
   13 JUNC02                2     8   -23
   14 COAL01               54   -26   -25
   15 COSTER               44   -44
   16 WASH01               16    29   -38   -37
   17 WASH02               38    39   -41   -40
   18 JUNC02               12    30   -13
   19 WASH01               52    28   -31   -47
   20 WASH02               31    32   -34   -33
   21 COAL01               34   -36   -35
   22 SETST2               36    -1
   23 SETST2               48    -5
   24 JUNC02               53    49   -45
   25 COAL01               41   -43   -42
   26 JUNC02               15   -46
   27 REPORT               44   -44
   28 SETST2               50   -27
   29 EXCH11 E-6010         7    43   -50   -48
   30 EXCH11 E-6002        46    23   -51   -53
   31 JUNC02               45   -54
   32 MSTEM1               20    27   -49
   33 SETST2               51   -52
   34 SETST2                4   -44
   35 SETST2                3   -44
   36 CONT04 I-C4          26    -9
   33
```

```
2.              .587            .033            .70             .90
.30             .50             .45             .10
3.              .0              .0              1.              5.
1.              7.              340.            .926
4.0             3.6
0.0
5.              30300.          17.7            18.
29.                             0.0
6.0
1.2
7.0             8.0             0.7             0.005
0.0
9.0                                             0.01            2.
5.              20.7511         0.0             2.              1.
10.             80.             325.            0.              0.
1.0
11.             0.992           0.91            0.48            9.
```

13.	1.	7.	50.	
12.	30.	.891	120.	7.
15.	.985	847.		
13.0		95.0		
1.0				
14.	.80			
0.0				
15.	13.		75.	
0.38	0.38	0.282	0.0063	0.011
16.0	0.20	7.0		
0.0				
17.0	0.20	7.0		
0.0				
18.0				
1.0				
19.0	0.20	7.0		
0.0				
20.0	0.20	7.0		
0.0				
21.0	0.80			
0.0				
22.0	170.0	0.0		
0.0				
23.	196.	2.	847.	
7.				
24.0				
1.0				
25.0	0.80			
0.0				
26.		225.		
1.0				
27.	9.	.0		100.
.0				15.
28.	80.	7.	104.	
7.				
29.	44000.	1.		
7.	7.			
30.	120000.	1.		
7.	7.			
31.				
1.0				
33.	142.	1.	938.7	0.
7.				
34.	80.	7.	115.	15.
7.				
35.	90.	7.	101.	15.
7.				
36.				30.
.0				
7				
.2.			104.	15.
	6.	.9		
.0				
.0				
5.		10.	7.	6.
17.	7.	0.	0.	0.
0.0				
0.0				

```
6.0             0.0             5.              7.0             0.0
0.0
0.0
0.0
13.                             10.             7.
0.0
0.0
0.0
18.0                            5.0             7.0
0.0
0.0
0.0
24.                                             7.
0.0
0.0
0.0
31.                             5.              7.
.0
.0
.0
   9                                                        STREAM VECTORS
9.              .0              1731.           43.8644         95.
1.1549
.0              191.31          65.             490.78          854.27
58.00           61.61           2.32            6.46            .0
.0
.0
.0
10.             .0              1256.           51.832          70.
108.61          1120.                                           44.21
.0
.0
.0
.0
.0
27.             .0              690.7           80.
0.0
.0              19.8            .0              247.4           412.4
1.3             1.5             .8              6.4
.0              .10             .0              .25             .26
.0              .15             .07             .15             .015
.015
22.             1.              500.5           42.             14.7
2.
0.0             122.0           65.0            116.0           80.0
55.4            58.6            1.5
0.
0.0
0.0
28.0            1.0             40.016          50.0            180.0
39.09                           0.926
0.0
0.0
0.0
0.0
0.0
29.0            1.0             10.2275         50.0            318.0
10.0            0.0             0.2375
```

```
0.0
0.0
0.0
.0.0
0.0
30.             1.              276.0612        38.             80.
2.3572
0.0             24.9056         0.0             126.6782        111.6029
3.7395          3.7395          0.5842          2.4540          0.0
.0
0.0
0.0
32.0            1.0             208.0           50.0            225.0
208.0
0.0
0.0
0.0
0.0
0.0
39.0            1.0             30.5            50.0            325.0
30.5
0.0
0.0
0.0
0.0
0.0
    0                                                           STREAM CONTROL VECTORS
   31                                                           COMPONENT TOLERANCES
                                .1              .01                     .01
.1              .1              .1              .1                      .1
0.001           0.001           0.001           0.001                   0.001
0.001           0.001           0.001           0.001                   0.001
0.001           0.001           0.001           0.001                   0.001
0.001           0.001           0.001           0.001                   0.001
0.001
    0                                                           PREFERRED STREAMS
    2                                                           ADDITIONAL EQUIP VECTOR
15.0            45.             1.26                                    3.48
2.94            3.55            3.55            3.8                     3.8
0.0                             1.16            1.16                    3.97
4.13            3.24            3.24            3.8                     3.8
0.0                             4.13            3.6                     3.52
8.68            3.5
30.0            8.              4.0             6.0                     3.0
17.0            28.0            29.0            11.                     12.
15.             9.
36.             5.
    1   38
    1   1   6   1   3   2   4   2   11  2
   18   2   5   2   8   2   9   2   10  2
   16   2   17  2   25  2   29  2   23  2
   12   2   29  2   28  2   32  2   26  2
   30   2   33  2   19  2   20  2   21  2
   22   2   2   2   13  2   30  2   24  2
   31   2   14  2   36  2   7   2   27  3
   34   1   35  1   15  1
```