DYNSYS - A DIGITAL COMPUTER PROGRAM FOR STUDYING THE TRANSIENT BEHAVIOUR OF SYSTEMS USING A MODULAR APPROACH

DYNSYS - A DIGITAL COMPUTER PROGRAM FOR STUDYING THE TRANSIENT BEHAVIOUR OF SYSTEMS USING A MODULAR APPROACH

BY

STANLEY BOBROW

A Report

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

September, 1969

MASTER OF ENGINEERING (1969)
(Chemical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario.

TITLE : DYNSYS - A Digital Computer Program for Studying the Transient Behaviour of Systems Using a Modular Approach

AUTHOR : Stanley Bobrow
B.Sc. (Eng.) Technion-Israel Institute of Technology

SUPERVISOR : Professor A.I. Johnson

NUMBER OF PAGES : viii, 148

SCOPE AND CONTENTS:

This report deals with the transient behaviour of process systems such as those normally encountered by the chemical engineer.

Two competing approaches for the numerical solution of this problem are described. The first is an "equation oriented" approach, and the second is an "equipment oriented" or "modular" approach. The concept of modularity is discussed, and a brief presentation of its advantages is given.

A program DYNSYS-B was written to apply the modular approach to transient studies. The program employs a predictor-corrector technique to solve the system's differential equations in a sequential fashion.

The DYNSYS-B program was applied to a multicomponent liquid-liquid extraction unit with heat effects and reflux, with the object of evaluating the program and gaining some experience with its use.

Based on this study, we believe that the program is readily applied to the solutions of many problems. Some suggestions are made for improving the computation efficiency of the program.

ACKNOWLEDGEMENTS

The author is indebted to his supervisor, Professor A.I. Johnson, for his help and guidance throughout this work. Financial assistance from McMaster University is gratefully acknowledged. Special thanks are due to D.R. Prowse and A.J. Leather, who carried out the initial studies in applying the modular approach to transient studies.

The author is grateful to members of the technical staff at the Polymer Corporation, Sarnia, for suggesting the butadiene unit as an example of a system whose transient behaviour could be studied using DYNSYS, and for their interest in the project.

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
1.1 Incentives for the Study of Transient Behaviour	1
1.2 Analytical Techniques	4
1.3 Equation and Equipment Oriented Systems	6
1.3.1 Equation Oriented Systems	6
1.3.2 Equipment Oriented Systems	6
1.3.3 The Process Matrix	7
1.3.4 Advantages and Disadvantages of the Modular Approach	9
1.4 Mathematical Approaches	10
1.4.1 Simultaneous (Implicit)Method	10
1.4.2 Sequential Method	11
2. THE DYNSYS-B PROGRAM	12
2.1 Choice of Integration Method	12
2.2 A General Description of DYNSYS-B	15
2.2.1 Unit Computations	15
2.2.2 Order of Calculation	17
2.2.3 The Main Steps in the DYNSYS-B Program	18
2.2.4 Output of Calculation Results	21
2.2.5 Discussion of the Arrays and Variables Used in DYNSYS-B	21
2.3 Description of the DYNSYS-B Executive Subroutines	28
2.3.1 DYNSYS-B	28
2.3.2 Subroutine DYN1	31
2.3.3 Subroutine FIND	39
2.3.4 Subroutine DYN2	43
2.3.5 Subroutine AMØS	45
2.3.6 Subroutine OUTPUT (Version I)	49
2.3.7 Subroutine OUTPUT (Version II)	54

Page

2.3.8 Routine DYNPLOT 56

2.4 The DYNSYS-B Library 58
2.5 Illustrative Examples on the Use of DYNSYS-B in Solving Transient Problems 85

3. APPLICATION OF DYNSYS-B TO A LIQUID-LIQUID EXTRACTION UNIT WITH HEAT EFFECTS 105

3.1 Aim of the Simulation 105
3.2 Description of the System 105

3.2.1 Chemical 105
3.2.2 Physical 106
3.2.3 Operational 108

3.3 Scope of the Simulation 111
3.4 Construction of the Information Flow Diagram 112
3.5 Estimation of Physical Properties 118
3.6 Results of the Simulation 133

4. CONCLUSIONS AND RECOMMENDATIONS 137

REFERENCES 141

APPENDICES 144

INDEX OF FIGURES

		Page
FIGURE 1-1	Strip Chart of Reaction Temperature	3
FIGURE 1-2	Construction of a Process Matrix	8
FIGURE 2-1	Unit Computation Information Flow Diagram	16
FIGURE 2-2	Order of Calculation	18
FIGURE 2-3	DYNSYS-B Information Flow Diagram	22
FIGURE 2-4	Information Flow Diagram Representation of STIRØ1	85
FIGURE 2-5	Information Flow Diagram Representation of MXPLTØ2	85
FIGURE 2-6	System for Example 1	86
FIGURE 2-7	Information Flow Diagram for Blending System of Example 1	87
FIGURE 2-8	Information Flow Diagram Representation of Subroutine VALVØ1	96
FIGURE 2-9	Information Flow Diagram Representation of CNTRLØ1	96
FIGURE 2-10	Information Flow Diagram Representation of LAGØ2	97
FIGURE 2-11	System for Example 2	97
FIGURE 2-12(a)	Information Flow Diagram for Example 2	98
FIGURE 2-12(b)	Improved Information Flow Diagram for Example 2	99
FIGURE 3-1	A Typical Mixer/Settler Stage	107
FIGURE 3-2	Extraction Unit and Product Purification	109

FIGURE 3-3 Information Flow Diagram for a Mixer/Settler Stage 114

FIGURE 3-4 Information Flow Diagram for Extraction Process 115

FIGURE 3-5 Response of the Unit to a Step Change 134

1. INTRODUCTION

1.1 Incentives for the Study of Transient Behaviour

Much of the simulation work carried out in the Department of Chemical Engineering, McMaster University, has been concerned with the steady-state behaviour of chemical engineering systems. These studies have made extensive use of executive programs such as PACER [1], MACSIM [2] and GEMCS [3]. Steady-state simulation has been extremely useful in

- determining optimum operating conditions
- design aspects such as predicting the demands to be made on equipment at future production rates.

For certain processes, the results of the steady-state simulation are all that the engineer requires. Usually, these processes operate for long periods of time without being subjected to changes in production rates, or operating conditions. Maintenance shut-downs are infrequent, and there are few problems associated with starting-up or shutting down the unit. In some cases, such problems may exist but be of little economic significance in the overall operation of the plant.

However, there remain certain questions which cannot be answered by means of a study of the steady-state alone. Chemical processes are generally subject to disturbances, and indeed it may be said that no process is ever truly in steady-state. These disturbances may be intentionally introduced, such as in the case of a multi-product plant where the same equipment is used for the production of several different

and unrelated products. In other cases, disturbances are due to difficulties in plant operation - lines may plug, equipment is subject to failure etc. For these processes, the engineer may be called upon to answer some of the questions listed below:

- What are the effects of these disturbances and how are they propagated through the process?
- What corrective action should be taken to minimize economic loss due to disturbances?
- Which, amongst various control systems, will ensure the corrective action is taken in the most efficient way?
- What is the optimum way to start-up and shut-down the process?
- What will be the result of equipment failure? Are safety interlock systems required, and how should they be designed?
- Is it possible to reduce the magnitude of deviations from the optimum steady-state operating conditions and thereby improve the average yield or production capacity of existing equipment?

As an illustration, consider a hypothetical reactor whose performance is critically related to reaction temperature. The yield of desired products increases with temperature, but above the temperature TMAX indicated in Figure I-I a dangerous condition exists due to a runaway reaction (for example, nitric acid oxidation of hydrocarbons). Control over the reaction temperature is such that there may be as much as a 5% fluctuation around the set-point TSET. Consequently, to ensure

safe operation TSET is at least 5% below TMAX.

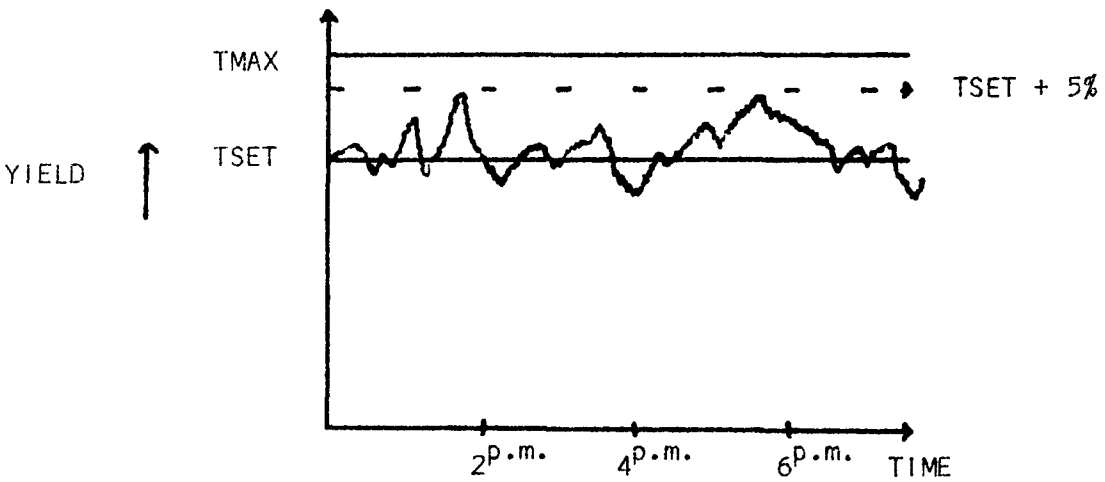


Figure I-I Strip Chart Record of Reaction Temperature

After acquiring a thorough understanding of the dynamic behaviour of the reactor, the engineer is able to improve control of the temperature such that the maximum deviation from the set-point is no greater than 3%. Therefore, the set-point may be increased by 2% without compromising safety tolerance limits. If the yield to desired products is approximately linearly related to temperature, a 2% relative improvement in yield would be achieved. In a typical process, an improvement of this magnitude may be measured in hundreds of thousand of dollars annually.

Generally speaking, the above questions can only be answered by studying the transient behaviour of the process.

1.2 Analytical Techniques

The classical approach of the electrical engineer to the solution of the differential equations he encounters in control problems has been the use of transfer functions. The differential equations describing the system must be linearized by making suitable approximations. The Laplace transform is then used to convert the differential equations to algebraic equations in the Laplace variable, commonly denoted by the letter "s". Finally, an algebraic expression, the ratio of output (dependent variable) to input (independent variable) is formed, and is known as the system's transfer function. The engineer now has several tools with which to proceed with his analysis. Questions of optimization of system parameters and system stability may be answered through the use of the root locus method, the Routh-Hurwitz criterion and the frequency response methods such as Bode diagrams and the Nyquist criterion. Usually, the inclusion of time-delays in the system will not unduly complicate analysis of the system stability. Inversion of the transfer function gives an analytical expression describing the time-response of the system. A full discussion of these techniques may be found in any standard text such as [11] or [12].

An alternate method uses matrix theory. A system of linear ordinary differential equations

$$\dot{\underline{X}} = \underline{A}\,\underline{X} + \underline{b}$$

$\underline{A}$ = matrix of constant coefficients

$\underline{b}$ = constant vector

$\underline{X}$ = vector of time dependent variables

may be solved simultaneously, the solution being expressed in terms of

eigen-values and eigen-vectors.

$$\underline{X} = \sum_j \left(\underline{W}_j^T \frac{(\underline{X}^i + \underline{A}^{-1}\underline{b}}{\underline{W}_j^T \underline{Z}_j} \right) \underline{Z}_j e^{\lambda_j t}) - \underline{A}^{-1}\underline{b}$$

where

$\underline{Z}_j$ are the eigenvectors of $\underline{A}$

$\underline{W}_j$ are the eigenrows of $\underline{A}$

λ_j are the eigenvalues of $\underline{A}$

Theory and derivations of the above may be found in texts such as [13] or [14].

Analytical methods, then, will provide exact answers to many problems. In practice, however, objections to these methods are readily found. Firstly, they are generally only applicable to linear equations. One of the main virtues, an exact solution, is defeated when the system differential equations must be approximated by linear differential equations. Secondly, the complexity of inverting a transfer function, or of calculating eigenvalues and vectors, increases rapidly as the number of equations in the system increases. Use of the transfer function approach is restricted to single variable problems, and s-plane analysis is far less meaningful to the engineer than time-domain analysis. For most practical problems involving tens of linear or non-linear differential equations, the engineer turns to numerical methods for the solution of a transient problem.

1.3 Equation and Equipment Oriented Systems

There are two basic approaches in the creation of an executive program for the numerical solution of a set of differential equations. These are described below.

1.3.1 Equation Oriented Systems

During the past several years, a number of equation oriented programs have been created. Some examples are MIMIC [4], MIDAS [5], DSL90 [6], and DYNAMO [19].

Traditionally, the engineer who wishes to solve a set of differential equations has found the analogue computer a useful tool. He has, however, found some drawbacks. The problem of scaling is often frustrating and tedious, the size of the problem that can be handled may be severely restricted, and in certain cases machine error is a problem. In an attempt to overcome some of these drawbacks, programs such as MIMIC were created as digital simulators of analogue computers.

The MIMIC user, according to special MIMIC format rules, establishes a list of the differential equations he wishes to solve. Algebraic and logical operations may also be included. The MIMIC executive automatically sorts and assembles the equations into a suitable order of calculation, and proceeds with the solution.

1.3.2 Equipment Oriented Systems

To our knowledge, there has been no publication describing an equipment oriented executive program for the solution of differential equations. However, equipment oriented programs (PACER [1], MACSIM [2], GEMCS [3], CHESS [15], and DISCOSSA [16]) do exist for the solution of systems of algebraic equations encountered in steady-state problems.

The essential feature of this approach is that the system of equations is divided into a number of subsystems, and each subsystem describes or represents a particular piece of equipment in the process. In general, each subsystem comprises a subroutine which is attached to the executive program. The subsystems are called equipment subroutines, computation modules, or unit computations. Programs using this approach are said to be modular.

For an equation oriented executive program, the user sets up an ordered assembly of equations. For an equipment oriented executive program, the user sets up a number of subroutines, each describing a unit in the process, together with some means of describing how these units are connected together. To describe this inter relationship between units is the role of the Process Matrix.

1.3.3 The Process Matrix

The physical arrangement or configuration of a number of unit computations required to describe a process, together with information flow between units, is described by means of a "Process Matrix".

Each equipment in the plant is assigned an "equipment number", unique to that equipment, and an "equipment type number", unique to that type of equipment. Thus valves in different parts of the plant would have different equipment numbers, but the same equipment type number. Furthermore, each stream in the plant is assigned a stream number - a plant stream of material flow between units corresponds to a flow of information between unit computations.

Each unit is given one row in the process matrix. The contents of that row are: the number of the particular unit, the type number

representing the unit and the input stream numbers (as positive numbers) followed by the output stream numbers (as negative numbers). For example, consider the flow diagram in Figure I-2. Equipment type numbers are also shown.

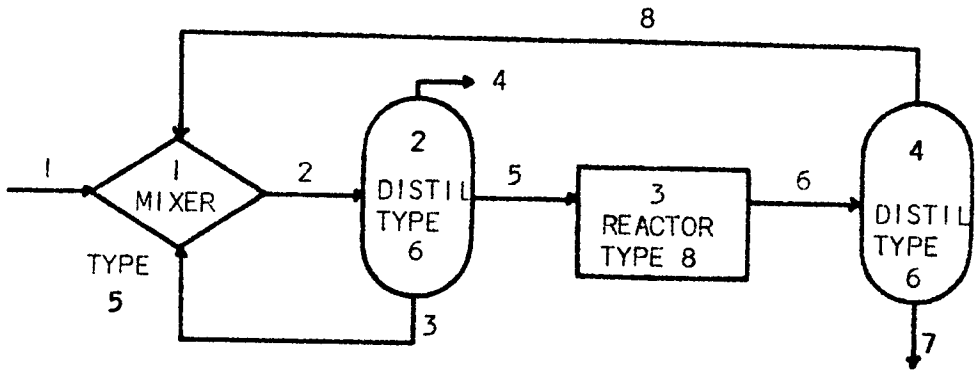


FIGURE I-2

CONSTRUCTION OF A PROCESS MATRIX

The process matrix for this system is shown in Table I-I.

Table I-I Example of a Process Matrix

Unit Number	Unit Type	Associated Streams			
1	5	+1	+8	+3	-2
2	6	+2	-4	-3	-5
3	8	+5	-6		
4	6	+6	-8	-7	

1.3.4 Advantages and Disadvantages of the Modular Approach

The modular approach has two basic advantages.

a) Transformation of the physical plant to an information flow diagram in preparation for the simulation is facilitated by the close correspondence between actual pieces of equipment and unit computations. This is particularly convenient for chemical engineers, who are accustomed to think of chemical processes as a collection of unit operations and reactors.

b) As a library of equipment subroutines becomes available, they may be readily used in the simulation of different plants, simply by setting up the required process matrix. (The modules would need to have a reasonable degree of generality). Also, addition, removal or rearrangement of equipments in the simulation of a given process is easily carried out.

On the other hand, an equation oriented program allows the engineer to utilize the particular structural form of the equations to obtain a solution in the most efficient way.

Generally speaking the modular approach will save engineer time in setting up a problem, but at the expense of computer time.

1.4 Mathematical Approaches

As a result of the preceding discussion, a major study project was initiated to apply the concept of modularity in the creation of an executive program for the digital simulation of transient behaviour.

There are two basic mathematical approaches to this problem.

1.4.1 Simultaneous (Implicit) Method

In each unit computation, the relevant differential equation is transformed to a backward time finite difference equation. Thus the differential equation

$$\dot{x} = f(x) \tag{1}$$

is written as

$$(x^{n+1} - x^n)/\Delta t = \alpha_1 . f(x^{n+1}) + \alpha_2 . f(x^n) + \alpha_3 . f(x^{n-1})$$

or

$$X^{n+1} - \{\alpha_1 . f(X^{n+1}) + \alpha_2 . f(X^n) + \alpha_3 . f(X^{n-1}) \ldots\} * \Delta t = X^n \tag{2}$$

where

x = process variable (e.g. temperature)
t = time
Δt = small increment in time
x^n = value of process variable at time t
x^{n+m} = value of process variable at time $t+m*\Delta t$
$\alpha(i)$ = coefficients

By the Mean Value Theorem, there exists some choice of $\alpha(i)$ such that equation (2) is exact.

For the entire process, a matrix of equations similar to (2) is constructed:

$$\underline{x}^{n+1} - \{\underline{\alpha}_1 . \underline{f}(\underline{X}^{n+1}) + \underline{\alpha}_2 . \underline{f}(\underline{X}^n) + \underline{\alpha}_3 . f(\underline{X}^{n-1}) \ldots\} . \Delta t = \underline{x}^n \tag{3}$$

In equation (3), $\underline{x}^n$ is a (known) vector of the values of the process variable under consideration in the various plant streams, and $\underline{x}^{n+1}$ is the vector we wish to calculate. In the case where $\underline{f}(\underline{x})$ is a linear function, we have an equation of the form

$$\underline{A}\ \underline{x}^{n+1} = \underline{B}\ \underline{x}^n + \underline{C}\ \underline{x}^{n-1} \ldots$$

which is solved by inversion of the matrix A to yield

$$\underline{x}^{n+1} = \underline{A}^{-1}(\underline{B}\ x^n + \underline{C}\ x^{n-1} + \ldots)$$

In the non-linear case, equation (3) must be solved by some iterative technique. In either case values of $\underline{x}^{n+1}$ are solved for simultaneousely throughout the process.

Using the implicit approach, work was begun by D.R. Prowse and continued by A.J. Leather and resulted in the creation of the program DYNSYS-A (Dynamic Systems Simulator). DYNSYS-A is described in detail in [21].

1.4.2 Sequential Method

This approach more closely follows the sequential technique of solving steady state problems which has been used extensively at McMaster. The equations in each unit computation are solved before proceeding to the next unit computation. A numerical integration technique (such as the Runge-Kutta or predictor/corrector equations) is employed to advance the values of the process variables in each equipment by one Δt. Calculation proceeds from equipment to equipment in a sequential fashion.

Based on this approach, the executive program DYNSYS-B was created, and is described in detail in Section 2 of this report.

2. THE DYNSYS-B PROGRAM

2.1 Choice of Integration Method

Choice of the method for numerical integration of the differential equations was based on a consideration of the following criteria:

a) Accuracy

In general, accuracy improves with the order of the finite difference approximations. This improvement is achieved at the expense of computation time for the single-step (e.g. Runge-Kutta) formulae, since the number of derivative evaluations increases. For multi-step (e.g. predictor/corrector) formulae, computer storage requirements increase.

b) Computation Time

This criterion depends on the number of derivative evaluations required for each integration step. Lapidus [7] demonstrates a scheme for predictor/corrector methods whereby two derivative evaluations are required per time step. The scheme is

- Predictor Step
- Corrector Step
- Improvement of the result of the corrector step using error analysis.

To achieve comparable accuracy with a single step method such as the 3rd order Runge-Kutta formulae, three derivative evaluations are required per step. Therefore, the predictor/corrector approach can have an appreciable advantage in computation time.

c) Numerical Stability

Distefano [8] has provided a useful comparison of the stability characteristics of various numerical techniques for the solution of differential equations.

Amongst the predictor/corrector techniques, it is demonstrated that while the predictor formula alone has poor stability characteristics, considerable improvement is achieved when the predictor/corrector pair is used. The corrector formula has good stability characteristics, typical of all implicit formulae. Predictor/corrector pairs based on Newton's backward formula (e.g. the Adams Moulton method) are markedly superior to those based on Newton's forward formula (e.g. Milne's Method) [7].

The 3rd order Adams Moulton pair appear to be somewhat superior to the 2nd and 3rd order Runge-Kutta equations.

d) Ease of Monitoring Errors

The predictor/corrector scheme outlined in (b) provides a ready and convenient method for the estimation of the integration error. For the Runge-Kutta equations, error estimates are commonly made by repeating the calculations over a given time interval but with half the original step size. Thus additional evaluations of the derivatives are required to obtain an estimate of the integration error.

e) Ease of Automatic Control of Integration Step Size

Single-step methods have long been popular since they require no previous or "history" points. Once an estimate of the integration error is obtained, the step size is readily altered. In the case of the multi-step methods, initial history points are required before they may be used. These history points are commonly calculated by means of a single step method, or, more simply, by using the lower order predictor/corrector pairs (e.g. Euler's method). Adjustment of integration step size involves either interpolation between existing history points, or "restarting" the integration with low order formulae as described above.

Much of this difficulty has been eliminated in the Adams Moulton Shell technique. This is a generalization of the Adams Moulton method to allow non-constant integration step size. The method was developed by D.L. Shell [9] and extended at the University of Florida [10]. The unique advantage of the method is that it provides automatic truncation error control of step size without requiring any special procedure for restart.

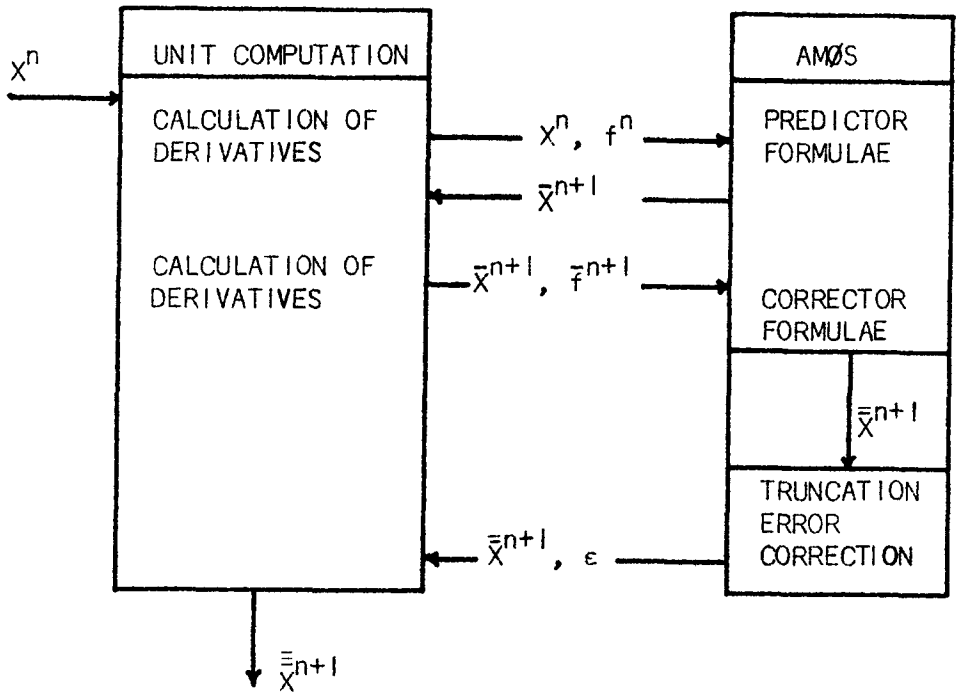
f) As has been stated above, use of higher order predictor/corrector formulae achieves increased accuracy at the expense of increased computer memory requirements. At McMaster University, it has been felt that 3rd or 4th order formulae present a reasonable compromise for many chemical engineering problems. Naturally, the choice of order would vary from problem to problem and is dependent on the available hardware. (core size and cycle time).

In consideration of the foregoing discussion, the Adams Moulton Shell approach was chosen for use in the DYNSYS-B program. For derivation of the formulae, the reader is referred to the University of Florida publication [10], or to a summary in Appendix I. A description of the method as applied in DYNSYS-B is given in Section 3-3 of this report (subroutine AMØS).

2.2 A General Description of DYNSYS-B

2.2.1 Unit Computations

For unit computations, it is assumed that we have input information at a time t^n and wish to calculate output information at an incremented time t^{n+1}. For example, in the case of a reactor, we might know chemical composition, temperature, pressure etc. at time t^n. The problem is to calculate the values of these variables of time t^{n+1}. This calculation is performed in three steps. Firstly, the unit computation uses the AMØS predictor formulae to estimate the values at t^{n+1}. Secondly, using these estimated values at t^{n+1} and the AMØS corrector formulae, an improved estimate at t^{n+1} is obtained. Finally, the truncation error is estimated and added to the results of step 2. On examining the AMØS formulae, we note that the time derivatives of the process variables must be known, and these are calculated in the unit computation, usually from mass and energy balances. The flow of information described above is illustrated schematically in Figure 2-1.



X^n	process variable at time t^n
$\bar{X}^{n+1}$	predicted value of variable at incremented time t^{n+1}
$\bar{\bar{X}}^{n+1}$	corrected value of variable at incremented time t^{n+1}
$\bar{\bar{\bar{X}}}^{n+1}$	final value after further corrections using truncation error analysis
f^n	value of time derivative at t^n (using X^n)
$\bar{f}^{n+1}$	estimated value of time derivative at t^{n+1} (using $\bar{X}^{n+1}$)
ε	estimated truncation error

Figure 2-1

Unit Computation Information Flow Diagram

2.2.2 Order of Calculation

The reader should note that the above procedure is applicable for unit computations described by differential equations. In certain cases, unit computations may contain algebraic equations alone, and cannot use the AMØS formulae. Typically, this case arises when an equipment has "zero" capacity, such as a T-junction in a pipe (subroutine MXPLTØ2). Thus, given values at t^n, we are unable to predict values at t^{n+1}. In an attempt to overcome this difficulty, the DYNSYS-B program carries out a simple linear extrapolation on all process variables to provide a rough estimate of values at t^{n+1}. It should be possible, of course, to perform higher order extrapolation techniques, but only at the expense of computer storage requirements and calculation time. Further, it is often possible to overcome this difficulty by proper attention to the sequence in which the unit computations are calculated. Before illustrating this point with an example, it is necessary to discuss one aspect of the DYNSYS-B program.

In Figure 2-1 we saw that the AMØS predictor formulae are used to estimate values at t^{n+1} (step 1), and the corrector formulae are used to improve these estimates (step 2). In fact, the process being studied will be composed of a number of unit computations. In DYNSYS-B, step 1 is carried out for all the units in the process. Subsequently, steps 2 and 3 are carried out for the whole process.

Consider the process illustrated in Figure 2-2. Unit computations numbered 1 and 2 contain algebraic equations only (such as a T-junction in a pipe line with "zero capacity"). Equipment number 3 contains differential equations.

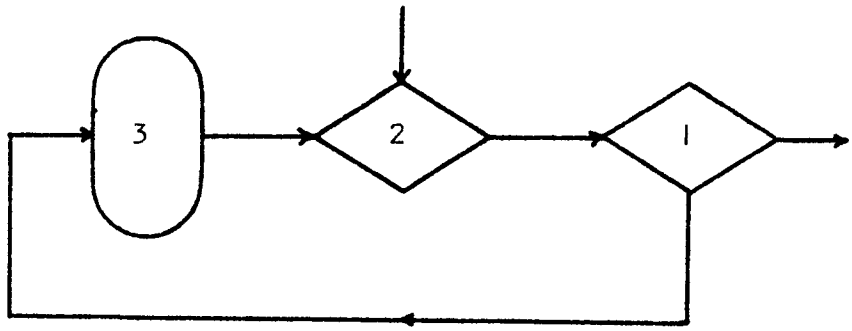


Figure 2-2 Order of Calculation

To calculate the output information for units 1, 2 at time t^{n+1}, we need to know the input information at t^{n+1}. For unit number 3, however, we can estimate the output information at t^{n+1} by knowing the input information at t^{n}, Clearly, a suitable order of calculation is 3 - 2 - 1 . An unsuitable order is 1 - 2 - 3 .

2.2.3 <u>The Main Steps in the DYNSYS-B Program</u>

As may be surmised, the **heart** of the program is in the integration formulae in subroutine AMØS. The remainder of the executive subroutines are concerned with the flow of information between unit computations, and the flow of information from the DYNSYS-B user to the computer and then back to the user. The sequence of these operations is described below:

a) READ the system parameters and initial conditions
(Subroutine DYN1)

System parameters would include the process matrix, the total number of streams, information about the unit computations (such as the size of a valve, the number of trays in a distillation column, etc.), truncation error tolerances etc. A full set of initial conditions - temperatures, concentrations

flows for all streams, vessel levels etc. - must be supplied. Normally, the initial conditions would be a set of steady-state values for the process, including one or more step changes in feed stream conditions, control points etc.

b) ARRANGE space for storage of values
(Subroutines DYN1 and FIND)

History values for the derivatives, parameters for unit computations etc. are used continuously during the calculations, and must be readily accessible to the unit computations. Special tables are defined for this purpose, and rows within these tables are assigned to the units in the process.

c) MARCH down the Process Matrix
(Subroutine DYN2)

Each equipment subroutine or unit computation is called upon in turn. Subroutine AMØS is called upon as required to predict values of process variables at time t^{n+1}.

d) MARCH down the Process Matrix
(Subroutine DYN2)

Each unit computation is again called upon, but this time the AMØS corrector formulae are used. During this step, truncation errors are calculated and used to improve the corrected values. To distinguish between steps (c) and (d), a variable "IG" is defined which takes on the value 2 for (c) and 1 for (d).

e) OUTPUT of calculated values of t^{n+1}
(Subroutine OUTPUT)

This subroutine is called upon to write out the calculation results. The user also has the option of plotting pre-selected variables.

f) UPDATING history values

Values of all stream variables are stored (remembered) for two time points, t^n and t^{n+1}. After all values at t^{n+1} have been calculated, those for t^n are discarded to leave space for values at t^{n+2}.

Updating operations also take place within the equipment subroutines, and are concerned with information "within" equipments rather than with information flowing "between" equipments. Included in the former class are history values of the derivatives (the right hand sides of differential equations).

g) STEP SIZE of integration

The truncation errors calculated in (d) are stored as a vector. The largest error is found, and is used to adjust the integration step size such that the largest error will fall between user specified error tolerance limits.

h) REPEAT

The integration time is incremented from t^{n+1} to t^{n+2}, and steps (c) - (h) are repeated. When t^{n+1} exceeds the user specified "final time", values are printed (or plotted) out and execution stops.

A simplified information flow diagram is given in Figure 2-3.

2.2.4 Output of Calculation Results

Output of results is handled by subroutine OUTPUT. The DYNSYS-B user may specify a "print" or a "plot" mode.

An additional alternative is available, and is intended for use where there are a large number of process variables we wish to plot. In this situation, it is convenient to have several plots, e.g. various process temperatures on one plot, concentrations on another. This is accomplished by replacing subroutine OUTPUT (version I) with another subroutine OUTPUT (version II) which simply writes the results out onto magnetic tape.

A calling program DYNPLOT is then used in conjunction with subroutine OUTPUT (version I) to transfer the results from magnetic tape to a number of plots.

2.2.5 Discussion of the Arrays and Variables Used in DYNSYS-B

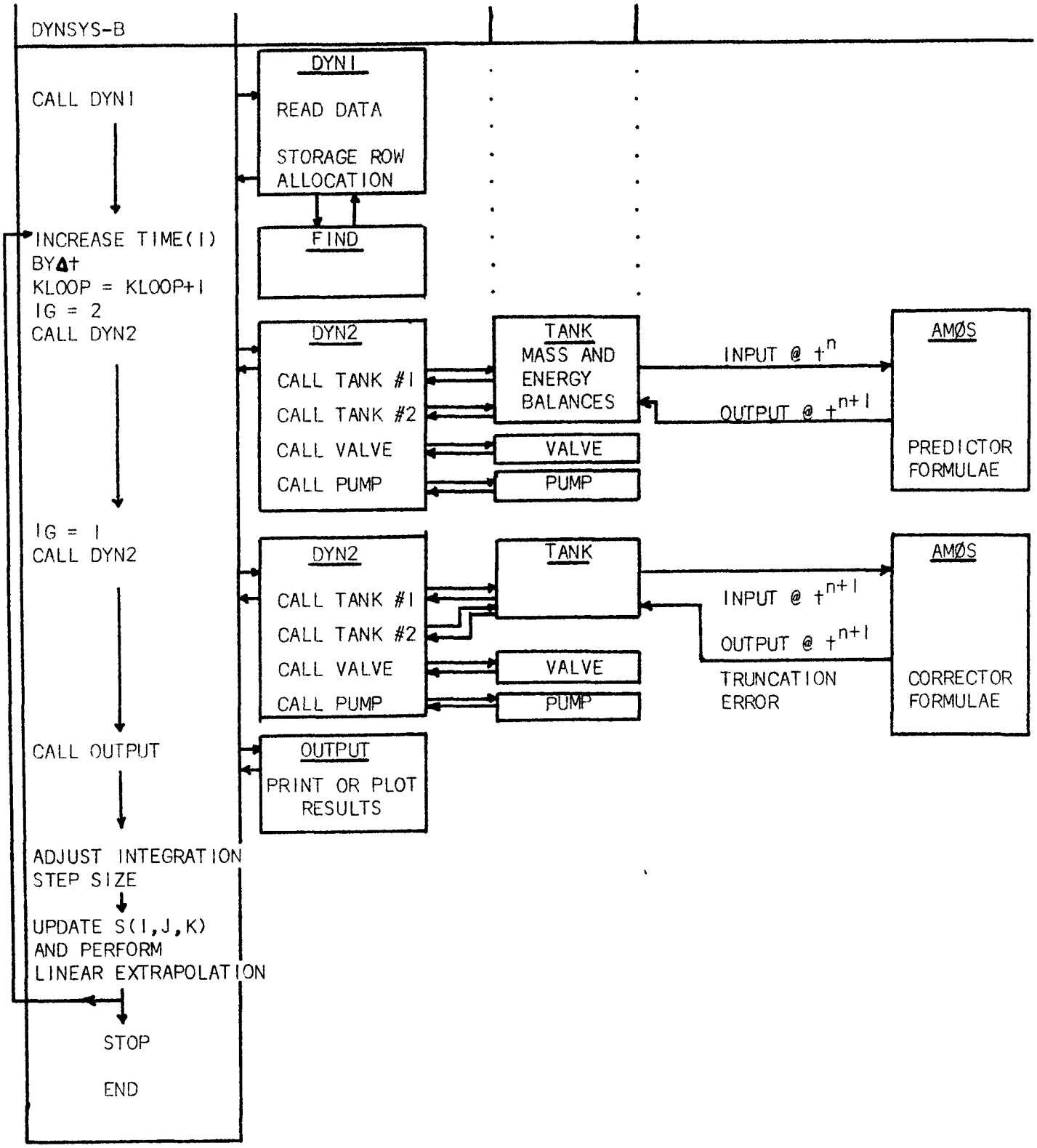
In this section, a description is given of the DYNSYS-B nomenclature. In particular, the use of the various tables defined in subroutine DYNI is discussed. As far as possible, only variables common throughout the program are described - those which relate only to one particular subroutine will be discussed under the detailed description of that subroutine.

i) TIME (I) - TIME(I) = time for which values of process variables are being calculated (t^{n+1}).
TIME(2), TIME(3)... etc. = times at which history points were calculated.

FIGURE 2-3

INFORMATION FLOW DIAGRAM FOR THE

DYNSYS-B SYSTEM



ii) S(I,J,K) - Stream values

I = 1 - values at TIME(1)

I = 2 - values at TIME(2)

J = row- one row is allocated for each stream.

It is convenient to allocate row (J) to stream number (J).

K = stream property. The arrangement of K is arbitrary, but would be uniform for any particular simulation problem under consideration. The arrangement we have used is:

K = 1 - stream number

2 - stream flag

3 - specific heat

4 - specific gravity

5 - total mass flow rate

6 - stream temperature

7 - concentration of component #1

8 - concentration of component #2.

Thus the storage location S(2,7,6) would represent the temperature of stream number (7) at TIME(2). Stream values are stored for TIME(1) and TIME(2) only.

iii) EC1(I,J,K) - Equipment Conditions

The arrangement of this matrix is similar to that described for S(I,J,K) above. In this case, however, the values stored would relate to pieces of equipment (e.g. liquid levels). A matrix EC2(I,J,K) is also defined with a function similar to that of EC1(I,J,K). Different types of equipments would require different maximum values for the parameter K, and

defining EC2 is intended to aid efficient utilization of computer storage.

It has been found inconvenient to require that values for equipment number (J) be stored in row (J) of the EC matrices. Rather, a matrix LOOKI( ) which is described below enables location of the storage row in EC( ) for a piece of equipment. The first column in EC( ) contains equipment numbers.

iv) BVI(I,J,K) - Back Values

Again, I represents time, J equipment number and K property. The values stored in this matrix are the values of the time derivatives of mass, energy etc. for individual pieces of equipment. For the form of AMØS used in the DYNSYS-B program, three history points for the time derivatives are required - therefore, I takes on the values 1, 2 and 3. The LOOKI( ) matrix is again employed to allocate values of J for items of equipment.

A BV2(I,J,K) matrix is also defined to avoid excessive computer storage.

v) EPI(J,K) - Equipment Parameters

This matrix stored values of equipment parameters (e.g. the size of a valve, the proportional band of a controller). J is a row unique for any equipment item. An EP2(J,K) matrix is defined for reasons of computer storage. The LOOKI( ) matrix is used to allocate values of J for equipment items. The first column of EPI or EP2 contains equipment numbers.

vi) MP(J,K) - Process Matrix

The Process Matrix, as mentioned previously, describes the equipment configuration of the plant being studied. J is a row unique for each piece of equipment

K = 1	Equipment Number
K = 2	Equipment Type Number
K = 3, ... N	Numbers of Streams entering equipment (with a positive sign)
K = N+1,...M	Numbers of Streams leaving equipment (with a negative sign)

To make MP(J,K) of uniform breadth, zeroes are entered where necessary. Specific rules may of course be adopted for any particular equipment subroutine (e.g. for an evaporator, the first output stream listed in the process matrix could represent liquid product, and the second output stream vapour product).

vii) PP(J,K) - Physical Properties Table

J is the storage row allocated for component number J. In the columns K = 1, 2 ... etc. are listed various constants required for physical property calculations. For example, suppose we wish to represent the specific heat of a component by a linearized function of temperature,

$$CP = A + B * (TEMPERATURE)$$

the values of the constants A and B would be stored in the PP( ) table.

viii) ERR(J) - Error Vector

This vector contains the truncation errors of the numerical integration of the system's differential equations.

ix) LOOKI(J,K) -

This matrix establishes correspondence between the row in which a particular item of equipment is located in the Process Matrix, and the storage rows for that equipment in EC( ), BV( ) and EP( ). The convention adopted is:

K = 1 row in EC1( ) or EC2( )

K = 2 row in BV1( ) or BV2( )

K = 3 row in EP1( ) or EP2( )

For example, suppose an equipment subroutine called TANK has been assigned a type number of 7, and that this equipment subroutine requires storage rows in EC1( ), BV2( ) and EP1( ). There are several TANK's in the process we wish to simulate, and one of these has been assigned an equipment number of 13, and is listed in the 18th row of the Process Matrix. Then the value of the number stored in location

LOOKI(18,3)

will be the storage row for equipment number 13 in the Equipment Parameter Matrix EP1( ). The storage row in BV2( ) will be given by LOOKI(18,2).

The reasons for this somewhat complicated procedure will become evident after examination of some equipment subroutines presented in Section 2.4. It should be noted

that the LOOK1( ) matrix is set up automatically in subroutine DYN1 with the aid of subroutine FIND (described in Section 2.3). The DYNSYS-B user need only specify which of the storage matrices EC1( ), EC2( ), BV1( )...etc. are required for the equipment subroutine.

x) NCOMP - number of components

xi) NE - number of equipments

xii) NS - number of streams

xiii) KLOOP - numbers of times the system has been calculated. Each loop corresponds to an integration step.

xiv) IG - this variable is given the value 2 during the predictor step, and the value 1 during the corrector step.

xv) NEC1 - number of equipments requiring storage in EC1( )

xvi) MEC1 - number of columns in EC1( )

xvii) NEC2 - number of equipment requiring storage in EC2( )

xviii) MEC2 - number of columns in EC2( )

xix) NBV1 - number of equipments requiring storage in BV1( )

xx) NBV2 - number of equipments requiring storage in BV2( )

xxi) NEP1 - number of equipments requiring storage in EP1( )

xxii) MEP1 - number of columns in EP1( )

xxiii) NEP2 - number of equipments requiring storage in EP2( )

xxiv) MEP2 - number of columns in EP2( )

xxv) NMP - number of columns in process matrix

xxvi) NERR - number of values stored in the ERR( ) vector

xxvii) SX(I,J,K) -

This matrix is similar to S(I,J,K). However, the parameter I (representing time) takes on values greater than 2. SX(I,J,K) is intended for use of streams requiring a larger number of history points, such as those associated with time lag elements.

xxviii) NSX -

number of streams stored in SX( ).

2.3 Description of the DYNSYS-B Executive Subroutines

In this section, detailed descriptions are given of the subroutines which comprise the DYNSYS-B executive program. Listings are included.

2.3.1 DYNSYS-B

This is the main calling program, and controls the information flow to and from the other subroutines. The steps are:

STEP	COMMENTS
CALL DYN1	Data input in storage space allocation for history points.
INCREMENT LOOP COUNT	
INCREMENT TIME( ) vector	
IG = 2 CALL DYN2	Calculation of equipment subroutines - predictor step.
IG = 1 CALL DYN2	Calculation of equipment subroutines - corrector step.
CALL OUTPUT	Printout of results - this statement is executed at a user - specified frequency

FIND THE MAXIMUM TRUNCATION ERROR (EMAX)

Error values are calculated in AMØS and stored in the ERR( ) vector.

IF EMAX IS OUTSIDE THE RANGE TOLL-TOLU ADJUST THE STEP SIZE

TOLL,TOLU = Lower, upper truncation error tolerances, specified by the user. The AMØS third-order formulae used in DYNSYS have a truncation error proportional to the fourth power of the integration step size. Accordingly, step size adjustment is proportional to the factor

$$4\sqrt{(TOLU/EMAX)}$$

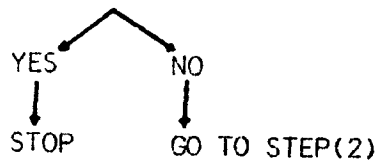
A factor of 0.9 is introduced to reduce the frequencies of changes in step size and has a bearing on computation time (see description of AMØS).

UPDATE S(I,J,K) AND LINEARLY EXTRAPOLATE

UPDATE THE TIME( ) VECTOR

IS TIME (2) > TMAX ?

TMAX is the user-specified terminal integration time.



```
C     PROGRAM DYNSYS-B .THE MAIN CALLING PROGRAM
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK3/NERR,ERR(100)
      CALL DYN1(TMAX,NE,NS,TOLL,TOLU,NPRINT,DELT )
      IG=2
      KLOOP=0
      CALL OUTPUT (NS,NE,KLOOP)
  100 KPRINT=0
  120 TIME(1)=TIME(2)+DELT
      KLOOP=KLOOP+1
      KPRINT=KPRINT+1
      CALL DYN2 (NE,KLOOP)
      IG=1
      NERR=0
      CALL DYN2 (NE,KLOOP)
      IG=2
      IF(KPRINT.LT.NPRINT) GO TO 140
      CALL OUTPUT (NS,NE,KLOOP)
      KPRINT=0
C     FIND THE MAXIMUM ERROR
  140 EMAX=ABS(ERR(1))
      DO 160 I=1,NERR
      IF(ABS(ERR(I)).GT.EMAX) EMAX=ABS(ERR(I))
  160 CONTINUE
C     ADJUST THE STEP SIZE IF NECESSARY
      DT=DELT
      FACT=(0.9*TOLU/EMAX)**0.25
      IF(EMAX.GT.TOLU.OR.EMAX.LT.TOLL)DELT=DELT*FACT
C     UPDATE AND PREDICT FOR THE S MATRIX
      NC6=NCOMP+6
      DO 180 I=1,NS
      S(1,I,1)=S(2,I,1)
      S(1,I,2)=S(2,I,2)
      DO 180 J=3,NC6
      X=S(1,I,J)+(S(1,I,J)-S(2,I,J))*DELT/DT
      S(2,I,J)=S(1,I,J)
  180 S(1,I,J)=X
C     UPDATE THE TIME() VECTOR
      DO 190 I=1,29
      II=30-I
  190 TIME(II+1)=TIME(II)
      IF(TIME(2).GT.TMAX) GO TO 200
      GO TO 120
 200  TIME(1)=TIME(2)
      CALL OUTPUT (NS,NE,KLOOP)
      STOP
      END
```

2.3.2 Subroutine DYN1

This subroutine is called upon by DYNSYS-B and performs the following functions:

Reading in Data
Storage Row Allocations
Printout of Data

Nomenclature:

NPRINT	-	Print interval. Printout is obtained every NPRINT loops
NPP	-	Number of columns in PP( ) matrix.
IOUTPUT	-	Output mode (+1 = Print, -1 = Plot)
DELT	-	Initial time step (increment)
TMAX	-	Final time
TOLL,TOLU	-	Lower and Upper truncation error tolerances.

Steps through subroutine DYN1:

STEP	COMMENTS
READ VARIOUS SYSTEM CONSTANTS	
READ PROCESS MATRIX, PHYSICAL PROPERTY TABLE	
MARCH DOWN PROCESS MATRIX TO CALCULATE NEC1, NEC2, NBV1, NBV2 NEP1, NEP2, NSX	The statements NEC1 = NEC1 + 1 etc. in this section must be supplied by the user according to which storage matrices are required for the equipment subroutines. The parameters NEC1 etc. are used for data reading, data printout and printout of results. They are also a useful check for errors in the data deck and whether the various matrices have been properly dimensioned.

READ S( ) INITIAL VALUES	For convenience in preparation of the data deck, the specific heat (CP) and specific gravity (SG) of each stream is calculated at this point. It is assumed that CP and SG are linear functions of temperature for any pure component. For a mixture, an average is calculated.
READ EC1( ), EC2( ), EP1( ), EP2( )	
SCAN THE PROCESS MATRIX TO ALLOCATE STORAGE ROWS FOR THE EQUIPMENTS - SET UP THE LOOK1( ) MATRIX	See subroutine FIND for further explanation. The user must ensure that the CALLFIND statements in this section are suitable for the required equipment subroutines.
WRITE OUT THE DATA	
IF REQUIRED, READ IN DATA FOR PLOTTING OUTPUT	See subroutine OUTPUT for an explanation of the nomenclature.

```
      SUBROUTINE DYN1(  TMAX,NE,NS,TOLL,TOLU,NPRINT,DELT)
C     NOMENCLATURE....NMP=NO.COL S  IN PROCESS MATRIX
C     NE=NO.EQUIPS      NS=NO.STREAMS   NCOMP=NO.COMPONENTS
C     MP(I,J)=PROCESS MATRIX   COL 1 =EQ.NO.      COL 2 =EQ TYPE
C                 IN REMAINING COLUMNS LIST +(IN STREAMS)  AND  -(OUT ST)
C     EP(I,J)=EQUIPMENT PARAMETERS      ONE ROW PER EQUIPMENT
C     NPRINT=PRINTOUT INTERVAL      NMP=NO.COL,S IN PROCESS MATRIX
C     DELT=INITIAL TIME STEP    TMAX=FINAL TIME  TOLL(TOLU=LOWER (UPPPER
C                                          ERROR TOLERANCES
C
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK3/NERR,ERR(100)
      COMMON/BLK4/NPLOTS,NPLOTE1,PLOTS(20,4),PLOTE1(15,4),PLOTT,
     1   PLOTIME,IOUTPUT,LKPLTE(85)
      COMMON/BLK5/NEC2,MEC2,NBV2,NEP2,MEP2,EC2(1 ,1 ,1 ),BV2(3,25,3),
     1  EP2(15,10)
      COMMON/BLK6/NSX,SX(15,25,14)
      COMMON/BLK8/NNBV1,NNBV2,NNSX
C
      READ(5,500)NMP,NE,NS,NCOMP,NPRINT,NPP,IOUTPUT,MEC1,MEC2,MEP1,MEP2
      READ(5,501) DELT,TMAX,TOLL,TOLU
      DO 50 I=1,NE
      READ(5,500) (MP(I,J),J=1,NMP)
   50 CONTINUE
      DO 60 I=1,NCOMP
      READ(5,501) (PP(I,J),J=1,NPP)
   60 CONTINUE
C
C     STORAGE ALLOCATION
      NEC1=0
      NBV1=0
      NEP1=0
      NEC2=0
      NBV2=0
      NEP2=0
      NSX=0
      DO 200 I=1,NE
      NTYPE=MP(I,2)
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,114,115
     1)NTYPE
C     TYPE -1-  XTRAC01
  101 NEC1=NEC1+1
      NBV1=NBV1+1
      NEP1=NEP1+1
      GO TO 200
C     TYPE -2- LAG02
  102 NSX=NSX+1
      NEP1=NEP1+1
      GO TO 200
```

```
C       TYPE -3-  CNTRL01
 103    NBV2=NBV2+1
        NEP2=NEP2+1
        GO TO 200
C       TYPE -4-  VALV01
 104    NEP1=NEP1+1
        GO TO 200
C      TYPE -5-  MXPLT02
 105    NEP1=NEP1+1
        GO TO 200
C       TYPE -6-  SETL01
 106    NEC1=NEC1+1
        NBV2=NBV2+1
        NEP1=NEP1+1
        GO TO 200
C       TYPE -7-  SETT01
 107    NEP1=NEP1+1
        GO TO 200
C       TYPE -8-  STIR01
 108    NEC1=NEC1+1
        NEP1=NEP1+1
        NBV1=NBV1+1
        GO TO 200
C       TYPE -9-  SETFLO
 109    GO TO 200
C       TYPE -10-  CONTROL
 110    GO TO 200
 111    CONTINUE
 112    CONTINUE
 113    CONTINUE
 114    CONTINUE
 115    CONTINUE
  200   CONTINUE
C
        NC6=NCOMP+6
        DO 220 J=1,NS
        READ(5,501)(S(1,J,K),K=1,NC6)
        CP=0.
        SG=0.
        DO 218 K=7,NC6
        KK=K-6
        CP =CP+(PP(KK,1)+PP(KK,2)*S(1,J,5))*S(1,J,K)
  218   SG=SG+(PP(KK,3)+PP(KK,4)*S(1,J,5))*S(1,J,K)
        S(1,J,3)=CP
        S(1,J,4)=SG
        DO 220 KK=1,NC6
        S(2,J,KK)=S(1,J,KK)
  220   CONTINUE
```

```
      IF(NEC1.EQ.0) GO TO 224
      DO 222 I=1,NEC1
      READ(5,501) (EC1(1,I,J),J=1,MEC1)
      DO 222 J=1,MEC1
      EC1(2,I,J)=EC1(1,I,J)
  222 CONTINUE
  224 IF(NEC2.EQ.0) GO TO 228
      DO 226 I=1,NEC2
      READ(5,501) (EC2(1,I,J),J=1,MEC2)
      DO 226 J=1,MEC2
      EC2(2,I,J)=EC2(1,I,J)
  226 CONTINUE
  228 IF(NEP1.EQ.0) GO TO 232
      DO 230 I=1,NEP1
      READ(5,501) (EP1(I,J),J=1,MEP1)
  230 CONTINUE
  232 IF(NEP2.EQ.0) GO TO 236
      DO 234 I=1,NEP2
      READ(5,501) (EP2(I,J),J=1,MEP2)
  234 CONTINUE
  236 CONTINUE
C     STORAGE ALLOCATIONS
      NNBV1=0
      NNBV2=0
      NNSX=0
      DO 350 I=1,NE
      NTYPE=MP(I,2)
      NEQ=MP(I,1)
      GO TO (251,252,253,254,255,256,257,258,259,260,261,262,263,264,265
     1)NTYPE
C     TYPE -1-  XTRAC01
  251 CALL FIND(I,1,NEQ)
      CALL FIND (I,3,NEQ)
      CALL FIND (I,5,NEQ)
      GO TO 350
C     TYPE -2-  LAG02
  252 CALL FIND(I,7,NEQ)
      CALL FIND(I,5,NEQ)
      GO TO 350
C     TYPE -3-  CNTRL01
  253 CALL FIND(I,4,NEQ)
      CALL FIND (I,6,NEQ)
      GO TO 350
C     TYPE 4-  VALV01
  254 CALL FIND(I,5,NEQ)
      GO TO 350
C     TYPE -5-  MXPLT02
  255 CALL FIND(I,5,NEQ)
      GO TO 350
C     TYPE -6-  SETL01
  256 CALL FIND(I,4,NEQ)
      CALL FIND(I,1,NEQ)
      CALL FIND(I,5,NEQ)
      GO TO 350
```

```
C       TYPE -7-  SETT01
 257    CALL FIND(I,5,NEQ)
        GO TO 350
C       TYPE -8-  STIR01
 258    CALL FIND(I,1,NEQ)
        CALL FIND(I,3,NEQ)
        CALL FIND(I,5,NEQ)
        GO TO 350
C       TYPE -9-  SETFLO
 259    GO TO 350
C       TYPE -10-  CONTROL
 260    GO TO 350
 261    CONTINUE
 262    CONTINUE
 263    CONTINUE
 264    CONTINUE
 265    CONTINUE
  350   CONTINUE
C       SET TIME(),ERR(),BV1(),BV2()  TO ZERO
        DO 410 I=1,100
        ERR(I)=0.
  410   CONTINUE
        DO 415 I=1,30
        TIME(I)=0.
  415   CONTINUE
        DO 425 I=1,3
        DO 425 J=1,NBV2
        DO 425 K=2,3
        BV2(I,J,K)=0.
 425    CONTINUE
        NN=2*NCOMP+2
        DO 430 I=1,3
        DO 430 J=1,NBV1
        DO 430 K=2,NN
        BV1(I,J,K)=0.
  430   CONTINUE
C
C       PRINTOUT OF DATA
        WRITE(6,507)
        WRITE(6,508)
        DO 435 I=1,NE
        WRITE(6,509) (MP(I,J),J=1,NMP)
  435   CONTINUE
        WRITE(6,510)
        DO 440 I=1,NCOMP
        WRITE(6,511) I,(PP(I,J),J=1,NPP)
  440   CONTINUE
        WRITE(6,513)
        DO 445 I=1,NEP1
        N=EP1(I,1)
        WRITE(6,514) N,(EP1(I,J),J=2,MEP1)
  445   CONTINUE
```

```
      IF(NEP2.EQ.0) GO TO 455
      DO 450 I=1,NEP2
      N=EP2(I,1)
      WRITE(6,514) N,(EP2(I,J),J=2,MEP2)
 450  CONTINUE
 455  WRITE(6,515)NE,NMP
      WRITE(6,517) NCOMP,NS
      WRITE(6,518) NPRINT,DELT
      WRITE(6,520) TOLU,TOLL
      WRITE(6,535)NEC1,NEC2
      WRITE(6,536)NBV1,NBV2
      WRITE(6,537)NEP1,NEP2
      WRITE(6,530)
      WRITE(6,531)
      DO 460 I=1,NS
      N1=S(1,I,1)
      N2=S(1,I,2)
      WRITE(6,532) N1,N2,(S(1,I,J),J=3,NC6 )
 460  CONTINUE
      WRITE(6,533)
      DO 465 I=1,NEC1
      N1=EC1(1,I,1)
      N2=EC1(1,I,2)
      WRITE(6,534) N1,N2,(EC1(1,I,J),J=3,MEC1)
 465  CONTINUE
      IF(NEC2.EQ.0) GO TO 480
      DO 470 I=1,NEC2
      N1=EC2(1,I,1)
      N2=EC2(1,I,2)
      WRITE(6,534) N1,N2,(EC2(1,I,J),J=3,MEC2)
 470  CONTINUE
 480  CONTINUE
      WRITE(6,550)
C
C     FORMAT STATEMENTS
 500  FORMAT(1515)
 501  FORMAT(5F10.4)
 507  FORMAT(1H1,15X,35(1H$),5X,18HD A T A  I N P U T  ,5X,35(1H$)///)
 508  FORMAT(1H0,40X,15HPROCESS MATRIX  ,//23X,6HEQ.NO.,5X,4HTYPE,
     1 5X,10HIN STREAMS,5X,11HOUT STREAMS  )
 509  FORMAT(1H0,15X,10I10)
 510  FORMAT(       ////40X,20HPHYSICAL PROPERTIES  ,//11X,9HCOMPONENT,
     1  40X,9HCONSTANTS  )
 511  FORMAT(1H0,15X,I2,5X,7(3X,  E10.3)/23X,7(3X,  E10.3))
 513  FORMAT(       ////40X,21HEQUIPMENT PARAMETERS   )
 514  FORMAT(1H0,2X,1H-,I3,1H-,2X,8(3X,F12.4),/12X,8(3X,F10.4))
 515  FORMAT(       ////42X,17HSYSTEM CONSTANTS  //10X,I5,12X,
     1 17H-NE=NO.EQUIPMENTS,17X,I5,15X,29H-NMP=NO.COL,S IN PROC.MATRIX )
 516  FORMAT(1H0,9X,I5,12X,14H-NS=NO.STREAMS,20X,I5,15X,
     1  19H-NSX=NO. X STREAMS   )
 517  FORMAT(1H0,9X,I5,12X,20H-NCOMP=NO.COMPONENTS,14X,I5,15X,
     1  11H-NO.STREAMS  )
```

```
  518 FORMAT(1H0,9X,I5,12X,22H-NPRINT=PRINT INTERVAL,12X,F7.5,13X,
     1  21H-DELT=INIT.TIME STEP  )
  520 FORMAT(1H0,9X,F10.7,7X,21H-TOLU=UPPER ERROR TOL,13X,F10.7,10X,
     1  22H-TOLL=LOWER ERROR TOL  )
  530 FORMAT(       ////40X,19HINITIAL CONDITIONS  )
  531 FORMAT(1H0,5X,6HSTREAM,3X,4HFLAG,10X,2HCP,13X,2HSG,11X,4HTEMP,9X,
     1  8HTOT.FLOW,12X,16HCONCENTRATIONS      )
  532 FORMAT(1H0,6X,I3,5X,I3,4(3X,F12.4),3(3X,F12.8),/18X,7(3X,F12.8))
  533 FORMAT(       ///6X,5HEQUIP,3X,4HFLAG,20X,11HPROPERTIES  )
  534 FORMAT(1H0,6X,I3,4X,I3,1X,7(3X,F12.4),/20X,7(3X,F12.4))
  535 FORMAT(1H0,9X,I5,12X,22H-NEC1=NO.EQUIPS IN EC1,12X,I5,     15X,
     1  23H-NEC2=NO.EQUIPS IN EC2  )
  536 FORMAT(1H0,9X,I5,12X,22H-NBV1=NO.EQUIPS IN BV1,12X,I5,     15X,
     1  23H-NBV2=NO.EQUIPS IN BV2  )
  537 FORMAT(1H0,9X,I5,12X,22H-NEP1=NO.EQUIPS IN EP1,12X,I5,     15X,
     1  23H-NEP2=NO.EQUIPS IN EP2  )
  550 FORMAT(       /////10X,50(2H$ ),//10X,50(2H$ )////)
      IF(IOUTPUT.LT.0) GO TO 600
      RETURN
  600 READ(5,500) NPLOTS,NPLOTE1
      READ(5,501) PLOTIME
      IF(NPLOTS.EQ.0) GO TO 620
      READ(5,1000) ((PLOTS(I,J),J=1,4),I=1,NPLOTS)
620   IF(NPLOTE1.EQ.0) GO TO 680
      READ(5,1000) ((PLOTE1(I,J),J=1,4),I=1,NPLOTE1)
      DO 660 I=1,NPLOTE1
      NEQ=PLOTE1(I,1)
      DO 640 J=1,NE
      MM=MP(J,1)
      IF(MM.EQ.NEQ) GO TO 650
640   CONTINUE
650   LKPLTE(I)=LOOK1(J,1)
660   CONTINUE
680   CONTINUE
      PLOTT=0.
1000  FORMAT(4F10.4)
      RETURN
      END
```

2.3.3 Subroutine FIND

This routine is called by DYNI and is responsible for setting up the LOOKI( ) matrix.

Nomenclature:

I - row in process matrix

KLOOK - the value of this variable (in the argument list) directs the calculations to a specific section of FIND corresponding to whether storage space allocation is required in EC1, EC2, BV1, BV2, EP1, EP2 or SX, as follows:

Storage Allocated in Matrix	Value of KLOOK
EC1( )	1
EC2( )	2
BV1( )	3
BV2( )	4
EP1( )	5
EP2( )	6
SX( )	7

LOOKI(I,J) - I corresponds to row in process matrix

J = 1 corresponds to EC1( ) or EC2( )

J = 2 corresponds to BV1( ) or BV2( )

J = 3 corresponds to EP1( ) or EP2( )

J = 4 corresponds to SX( )

LOOKI(I,J) = storage row for equipment listed in Ith row of process matrix in storage matrix corresponding to J.

NEQ - Number of equipment

NNEC1, NNEC2, NNBV1, NNBV2

NNEP1, NNEP2, NNSX -

A count on the number of storage rows already allocated in the corresponding matrices.

```
      SUBROUTINE FIND (I,KLOOK,NEQ)
C     THIS ROUTINE SETS UP THE LOOK1() MATRIX WHICH HOLDS THE STORAGE
C     ROWS FOR EACH EQUIPMENT IN EC1(),EC2(),BV1(),BV2(),EP1(),EP2(),
C     SX(), SUCH THAT A ROW IN LOOK1() CORRESPONDS TO A ROW IN THE PROCES
C                                                          MATRIX
C     CONVENTION THAT COL.(1) IN LOOK1()=ROW FOR EC
C                          (2)                   BV
C                          3                     EP
C                          4                     SX
C     EXAMPLE..   LOOK1(17,2)=STORAGE ROW FOR 17TH EQUIPMENT IN PROCESS
C                MATRIX IN MATRIX BV1(),OR,BV2  (DEPENNDS ON USER SPEC.)
C     NOMENCLATURE FOR PARAMETERS FOR ROUTINE ..FIND..
C      KLOOK=1-EC1()   2-EC2()   3-BV1()   4-BV2()   5-EP1()   6-EP2()
C                                                              7-SX()
C     NEQ=EQUIPMENT NUMBER
C     I=ROW IN PROCESS MATRIX
C
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK5/NEC2,MEC2,NBV2,NEP2,MEP2,EC2(1 ,1 ,1 ),BV2(3,25,3),
     1  EP2(15,10)
      COMMON/BLK6/NSX,SX(15,25,14)
      COMMON/BLK8/NNBV1,NNBV2,NNSX
C
      GO TO (10,20,30,40,50,60,70      )KLOOK
   10 DO 12 J=1,NEC1
      K=EC1(1,J,1)
      IF(K.EQ.NEQ) GO TO 14
   12 CONTINUE
   14 LOOK1(I,1)=J
      RETURN
   20 DO 22 J=1,NEC2
      K=EC2(1,J,1)
      IF(K.EQ.NEQ) GO TO 24
   22 CONTINUE
   24 LOOK1(I,1)=J
      RETURN
   30 NNBV1=NNBV1+1
      LOOK1(I,2)=NNBV1
      BV1(1,NNBV1,1)=NEQ
      RETURN
   40 NNBV2=NNBV2+1
      LOOK1(I,2)=NNBV2
      BV2(1,NNBV2,1)=NEQ
      RETURN
```

```
50  DO 52 J=1,NEP1
    K=EP1(J,1)
    IF(K.EQ.NEQ) GO TO 54
52  CONTINUE
54  LOOK1(I,3)=J
    RETURN
60  DO 62  J=1,NEP2
    K=EP2(J,1)
    IF(K.EQ.NEQ) GO TO 64
62  CONTINUE
64  LOOK1(I,3)=J
    RETURN
70  NNSX=NNSX+1
    LOOK1(I,4)=NNSX
    SX(1,NNSX,1)=NEQ
    RETURN
    END
```

2.3.4 Subroutine DYN2

This subroutine is called by DYNSYS-B. Its function is to march down the 2nd column of the process matrix (which contains equipment type numbers) and to call on the equipment subroutines in order.

```
      SUBROUTINE DYN2 (NE,KLOOP)
C     THIS ROUTINE SCANS THE PROCESS MATRIX AND CALLS THE UNIT COMPUTATION
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      DO 500 I=1,NE
      NMOD=MP(I,2)
      GO TO (100,105,110,115,120,125,130,135,140,145,150,155)NMOD
C     TYPE -1-  XTRAC01
  100 CALL XTRAC01(KLOOP,I)
      GO TO 500
C     TYPE -2-  LAG02
 105  CALL LAG02 (KLOOP,I)
      GO TO 500
C     TYPE -3-  CNTRL01
 110  CALL CNTRL01 (I)
      GO TO 500
C     TYPE -4-  VALV01
 115  CALL VALV01 (I)
      GO TO 500
C     TYPE -5-  MXPLT02
 120  CALL MXPLT02(I)
      GO TO 500
C     TYPE -6-  SETL01
 125  CALL SETL01 (KLOOP,I)
      GO TO 500
C     TYPE -7-  SETT01
 130  CALL SETT01(I)
      GO TO 500
C     TYPE -8-  STIR01
 135  CALL STIR01(KLOOP,I)
      GO TO 500
C     TYPE -9-  SETFLO
 140  CALL SETFLO(I)
      GO TO 500
C     TYPE -10-  CONTROL
 145  CALL CONTROL(I)
      GO TO 500
 150  CONTINUE
 155  CONTINUE
  500 CONTINUE
      RETURN
      END
```

2.3.5 Subroutine AMØS

AMØS is called by the equipment subroutines to integrate the differential equations. The routine contains the predictor and corrector formulae for the 1st, 2nd and 3rd order Adams Moulton Shell equations. The lower order formulae are only used for "starting-up", that is, to supply initial history points. In all subsequent calculations, the 3rd order formulae are used. The truncation error vector ERR( ) is formed in this subroutine.

Nomenclature:

FA, FB, FC -

These are the values of the derivatives at previous times (i.e. the history points). Suppose we are to calculate values for the dependent variable at a time t^{n+1}, and that previous calculations were made at t^n, t^{n-1} etc.

For predictor formulae:

FA = derivative at time t^n
FB = derivative at time t^{n-1}
FC = derivative at time t^{n-2}

For corrector formulae:

FA = derivative at time t^{n+1}
FB = derivative at time t^n
FC = derivative at time t^{n-1}

Note that the meanings of FA, FB and FC are displaced by one time interval between predictor and corrector formulae. The AMØS 3rd order predictor corrector pair actually requires four history points for the derivatives. Only three points however, are required for any given step. Consequently, the BV matrices

need only be dimensioned for 3 instants in time, provided they are updated after calculation of the predictor equations has been completed. The DYNSYS-B user should be aware of this pitfall before adding equipment subroutines to the DYNSYS-B library.

Y1 - value of dependent variable of time t^{n+1}

Y2 - value of dependent variable at time t^{n}

II - = 2 when predictor formulae are required

= 1 when corrector formulae are required

IERR - when greater than zero, the truncation error is inserted in the ERR( ) vector.

XERR - the predicted value of the dependent variable. Used to estimate the truncation error.

IFLAG - this takes on the value 1 when no adjustment has been made to the integration step size. In this case, the AMØS formulae may be simplified and a reduction in computation time achieved.

```
      SUBROUTINE AMOS (KLOOP,II,IERR,Y1,Y2,FA,FB,FC,XERR)
C     THIS ROUTINE CONTAINS THE PREDICTOR/CORRECTOR ROUTINES,AND CALCUL
C     -ATES  THE ERRORS
C     IERR.GT.0=INCLUDE IN ERROR MATRIX.OTHERWISE,DO NOT INCLUDE
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1    LOOK1(85,4)
      COMMON/BLK3/NERR,ERR(100)
      H0=TIME(1)-TIME(2)
      H1=TIME(2)-TIME(3)
      H2=TIME(3)-TIME(4)
      HK1=TIME(1)-TIME(3)
      HK2=TIME(2)-TIME(4)
      IFLAG=0
      IF(ABS(H0-H1).LT.1.E-9.AND.ABS(H1-H2).LT.1.E-9)IFLAG=1
      IF(2-II)300,5,10
C     II=2....PREDICTOR ROUTINES
    5 IF(KLOOP-2)50,100,150
C     II=1....CORRECTOR ROUTINES
   10 IF(KLOOP-2)75,125,175
C     ONE POINT FORMULAE
   50 Y1=Y2+H0*FA
      RETURN
   75 Y1=Y2+(FA+FB)*H0/2.
      FACT=2.
      IF(IERR.GT.0) GO TO 250
      RETURN
C     TWO POINT FORMULAE
  100 Y1=Y2+H0*(FA+0.5*H0/H1*(FA-FB))
      RETURN
  125 Y1=Y2+H0/2.*(FA+FB)
      FACT=3.*(1.+H1/H0)
      IF(IERR.GT.0) GO  TO 250
      RETURN
C     THREE POINT FORMULAE
  150 IF(IFLAG.EQ.1) GO TO 155
      Y1=Y2+H0*(FA*(1.+H0/H1/HK2*((H1+HK2)/2.+H0/3.))
     1   +H0/H2*(-FB*(HK2/2.+H0/3.)/H1+FC*(H1/2.+H0/3.)/HK2))
      RETURN
  155 Y1=Y2+H0/12.*(FA*23.-FB*16.+FC*5.)
      RETURN
  175 IF(IFLAG.EQ.1) GO TO 180
      Y1=Y2+H0*(FA*(H1/2.+H0/3.)/HK1-FB*(((H1-H0)/2.+H0/3.)/H1-1.)
     1  -FC/H1/HK1/6.*H0**2)
      FACT=1.+(H1*HK2/2.+H0*(H1+HK2)/3.+H0**2/4.)/(H0*H1/2.
     1  +H0*(H0-H1)/3.-H0**2/4.)
      GO TO 185
  180 Y1=Y2+H0/12.*(FA*5.+FB*8.-FC)
      FACT=10.
  185 IF(IERR.GT.0) GO TO 250
      RETURN
```

```
C     ERROR CALCULATIONS AND STORAGE IN ERR VECTOR
  250 NERR=NERR+1
      IF(NERR.GT.100) GO TO 300
      Z=(XERR-Y1)/FACT
      Y1=Y1+Z
      ERR(NERR)=Z/Y1
      RETURN
  300 WRITE(6,301)
  301 FORMAT(1H0,5(1H*),17HMISTAKE IN AMOS    ,90(1H*))
      RETURN
      END
```

2.3.6 Subroutine OUTPUT (Version I)

This routine is called by DYNSYS-B to print or plot the results.

Nomenclature:

NPLOTS - number of values in stream matrix S( ) to be plotted

NPLOTEI - number of values in ECI( ) matrix to be plotted

PLOTS(I,J) - input data table for plotting variables from the stream matrix. One row in PLOTS(I,J) is required for each variable.

- J = 1 Stream number
- J = 2 Position of variable in stream matrix row (e.g. 6 = Temperature etc.)
- J = 3 Minimum anticipated value
- J = 4 Maximum anticipated value

PLOTEI(I,J)- similar to PLOTS(I,J), but for variables stored in ECI( ) matrix.

LKPLTE(I) - this vector relates the row in PLOTEI(I,J) to the row an equipment is stored in ECI( ). Thus LKPLTE(I) = storage row for Ith variable in PLOTEI( ) in ECI( ).

SYMBOL(I) - plotting symbols used in graph

PROPS(I) - names of variables to be plotted out from the stream matrix (e.g. TEMPERATURE, FLOW etc.)

PLOTIME - approximate time interval represented by the space between consecutive lines on the output page.

ALINE( ) - this vector is constructed in the plotting routine, and contains the symbols to be plotted out. The position of the symbols in ALINE( ) corresponds to numerical values they represent. ALINE( ) is then printed out in A format.

PLOTT - time at which previous values were plotted.

The steps through subroutine OUTPUT are:

STEP	COMMENTS
PRINT OR PLOT MODE?	
PRINT MODE:-	
PRINT STREAM MATRIX PRINT ECI MATRIX	A convention is adopted whereby the second column in these matrices contain flag numbers. Only streams or equipments with positive flag values are printed out.
PLOT MODE:-	
WRITE OUT DATA, INDICATING WHICH SYMBOLS HAVE BEEN ASSIGNED TO THE PROCESS VARIABLES	Symbols are assigned in the sequence A,B,C ... as the variables occur in PLOTS( ) and PLOTEI( ).
CLEAR ALINE( )	i.e. fill with blanks.
CALCULATE NUMBER OF LINES TO SKIP	We recall that each line approximately represents a time interval of PLOTIME
SCALE NUMERICAL VALUES ACCORDING TO THE MINIMUM/ MAXIMUM VALUES SPECIFIED IN THE DATA. IGNORE VARIABLES WHICH FALL OUTSIDE THIS RANGE	The scaled (0-100) numerical value will be the position of the symbol in vector ALINE( ).
INSERT THE SYMBOL IN ALINE( )	However, if another symbol (other than a blank) occupies the required location in ALINE( ), indicate this at the side of the graph.
HAVING INSERTED ALL SYMBOLS INTO ALINE( ), PRINT OUT ALINE( ) in A FORMAT.	

```
      SUBROUTINE OUTPUT (NS,NE,KLOOP)
C     THIS ROUTINE PRINTS OUT THE RESULTS
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK4/NPLOTS,NPLOTE1,PLOTS(20,4),PLOTE1(15,4),PLOTT,
     1   PLOTIME,IOUTPUT,LKPLTE(85)
      COMMON/BLK5/NEC2,MEC2,NBV2,NEP2,MEP2,EC2(1 ,1 ,1 ),BV2(3,25,3),
     1  EP2(15,10)
      DIMENSION SYMBOL(20),SYMBOL1(4),ALINE(125),PROPS(10),PROPE( 2)
      DATA SYMBOL/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     1   1HN,1HO,1HP,1HQ,1HR,1HS,1HT/
      DATA SYMBOL1/1H.,1H-,1H ,1H,/
      DATA PROPE/4HMASS,5HLEVEL/
      DATA PROPS/4HTEMP,7HTOTFLOW,5HCONCA,5HCONCB,5HCONCC,5HCONCD,5HCONC
     1E,5HCONCF,5HCONCG,5HCONCH/
      IF(IOUTPUT.LT.0) GO TO 900
      NC6=NCOMP+6
      WRITE(6,500) TIME(1)
      WRITE(6,502)
      DO 100 I=1,NS
      IF(S(2,I,2).LT.0.) GO TO 100
      NN=S(2,I,1)
      WRITE(6,504) NN,(S(1,I,J),J=5,NC6 )
  100 CONTINUE
      WRITE(6,506)
      DO 150 I=1,NEC1
      IF(EC1(2,I,2).LT.0.) GO TO 150
      NN=EC1(2,I,1)
      WRITE(6,503)NN,(EC1(1,I,J),J=3,MEC1)
  150 CONTINUE
      IF(NEC2.EQ.0) GO TO 180
      DO 175 I=1,NEC2
      IF(EC2(2,I,2).LT.0.) GO TO 175
      NN=EC2(2,I,1)
      WRITE(6,503) NN,(EC2(1,I,J),J=3,MEC2)
175   CONTINUE
180   CONTINUE
  500 FORMAT(       ////20X,5(1H$) ,5X,30HPROCESS VARIABLES AT TIME =    ,
     1  F9.5,5X,5(1H$)/)
  502 FORMAT(1H0,5X,6HSTREAM,5X,4HTEMP,10X,8HTOT.FLOW,15X,14HCONCENTRATI
     1ONS )
503   FORMAT(1H0,5X,I3,5X,6(3X,F12.5))
  504 FORMAT(1H0,5X,I3,2(3X,F12.4),5(3X,F12.8),/39X,5(3X,F12.8))
  506 FORMAT(1H0,5X,5HEQUIP, 15X,     ,20X,9HHOLD-UPS  )
      RETURN
900   IF(KLOOP.GT.0) GO TO 1000
```

```
C     FOR FIRST OUTPUT,SET UP HEADINGS
      WRITE(6,2000)
      IF(NPLOTS.EQ.0) GO TO 920
      DO 910 I=1,NPLOTS
      N1=PLOTS(I,1)
      N1=S(1,N1,1)
      N2=PLOTS(I,2)
      N2=N2-4
  910 WRITE(6,2005) SYMBOL(I),N1,PROPS(N2),PLOTS(I,3),PLOTS(I,4)
  920 IF(NPLOTE1.EQ.0) GO TO 930
      DO 925 I=1,NPLOTE1
      II=I+NPLOTS
      N1=PLOTE1(I,1)
      N2=PLOTE1(I,2)
      N2=N2-2
  925 WRITE(6,2010) SYMBOL(II),N1,PROPE(N2),PLOTE1(I,3),PLOTE1(I,4)
  930 X=0.
      DO 940 I=1,21
      ALINE(I)=X
  940 X=X+5.
      WRITE(6,2015) (ALINE(I),I=1,21)
      WRITE(6,2020)
 1000 CONTINUE
C     CALCULATE NUMBER OF LINES TO SKIP
      XSPACE=(TIME(1)-PLOTT)/PLOTIME
      NSPACE=XSPACE
      XXSPACE=NSPACE
      IF((XSPACE-XXSPACE).GT.0.5) NSPACE=NSPACE+1
      IF(KLOOP.EQ.0) NSPACE=2
      IF(NSPACE.LT.1)RETURN
C      SET UP ALINE()
      DO 1010 I=2,120
      ALINE(I)=SYMBOL1(3)
 1010 CONTINUE
      ALINE(1)=SYMBOL1(1)
      ALINE(101)=SYMBOL1(1)
      PLOTT=TIME(1)
      NSPACE=NSPACE-1
      IF(NSPACE.EQ.0) GO TO 1022
      DO 1020 I=1,NSPACE
      WRITE(6,2200) (ALINE(J),J=1,101)
 2200 FORMAT(19X,101A1)
 1020 CONTINUE
C     SET UP NEW ALINE()
 1022 N5=1
      N6=NPLOTS
      IF(NPLOTS.EQ.0) GO TO 1080
 1025 DO 1060 I=N5,N6
      IF(N5.GT.1.OR.NPLOTS.EQ.0) GO TO 1035
      N1=PLOTS(I,1)
      N2=PLOTS(I,2)
      XN3=(S(1,N1,N2)-PLOTS(I,3))*100./(PLOTS(I,4)-PLOTS(I,3))+1.0
      GO TO 1040
```

```
1035 II=I-N5+1
     N1=LKPLTE(II)
     N2=PLOTE1(II,2)
     XN3=(EC1(1,N1,N2)-PLOTE1(II,3))*100./(PLOTE1(II,4)-PLOTE1(II,3))
    1   +1.0
1040 N3=XN3
     XXN3=N3
     IF((XN3-XXN3).GT.0.5) N3=N3+1
     IF(N3.LT.2.OR.N3.GT.100) GO TO 1060
     IF(ALINE(N3).NE.SYMBOL1(3)) GO TO 1045
     ALINE(N3)=SYMBOL(I)
     GO TO 1060
1045 DO 1050 J=102,108
     IF(ALINE(J).EQ.SYMBOL1(3)) GO TO 1055
1050 CONTINUE
1055 ALINE(J)=ALINE(N3)
     ALINE(J+1)=SYMBOL(I)
     ALINE(J+2)=SYMBOL1(4)
1060 CONTINUE
1080 IF(N6.GT.NPLOTS) GO TO 1090
     N5=NPLOTS+1
     IF(NPLOTE1.EQ.0) GO TO 1090
     N6=N5+NPLOTE1-1
     GO TO 1025
1090 WRITE(6,2205) TIME(1),SYMBOL1(2),(ALINE(J),J=1,110)
2000 FORMAT(1H1,43X,5(1H$),2X,11HL E G E N D  ,2X,5(1H$)//
    1  20X,6HSYMBOL,9X,6HSTREAM,10X,6HEQUIP.,10X,8HVARIABLE,19X,
    2  5HRANGE  )
2005 FORMAT(1H0,22X,A2,13X,I3,28X,A7 ,3X,2(2X,F12.5))
2010 FORMAT(1H0,22X,A2,27X,I3,14X,A7,3X,2(2X,F12.5))
2015 FORMAT(////15X,21(1X,F4.0))
2020 FORMAT(     14X,5(1H.),21(5HI.... ))
2205 FORMAT(10X,F6.3,2X,112A1)
     RETURN
     END
```

2.3.7 Subroutine OUTPUT (Version II)

This routine is called by DYNSYS-B and is used to write the calculation results onto magnetic tape. Only streams and equipments with flag values greater than zero are written out.

OUTPUT uses a dummy variable XFLAG to mark the end of output for a particular instant of time.

```
      SUBROUTINE OUTPUT(NS,NE,KLOOP)
C     ROUTINE FOR WRITING OUT THE RESULTS ON TAPE
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      XFLAG=-1000.
      NC6=NCOMP+6
      WRITE(8,500) TIME(1)
      DO 50 I=1,NS
      KK=S(1,I,2)
      IF(KK.LT.0) GO TO 50
      WRITE(8,500) (S(1,I,J),J=1,NC6)
   50 CONTINUE
      WRITE(8,500) XFLAG
      DO 100 I=1,NEC1
      KK=EC1(1,I,2)
      IF(KK.LT.0) GO TO 100
      WRITE(8,500) (EC1(1,I,J),J=1,4)
  100 CONTINUE
      WRITE(8,500) XFLAG
  500 FORMAT(F15.7)
      RETURN
      END
```

2.3.8 Routine DYNPLOT

This routine reads results from magnetic tape and calls subroutine OUTPUT to plot out the numbers. This process is repeated for each graph we wish to make.

Nomenclature:

NGRAPH - number of graphs to be made

XFLAG - see subroutine OUTPUT (Version II)

```
C       ROUTINE DYNPLOT
C      THIS ROUTINE READS VALUES FROM TAPE AND CALLS
C      (OUTPUT) TO PLOT OUT THE RESULTS
C      NGRAPH=NO.GRAPHS TO BE MADE
       COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
      1   LOOK1(85,4)
       COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
      1  EP1(65,5)
       COMMON/BLK4/NPLOTS,NPLOTE1,PLOTS(20,4),PLOTE1(15,4),PLOTT,
      1   PLOTIME,IOUTPUT,LKPLTE(85)
       READ(5,1000)NGRAPH
       DO 300 KGRAPH=1,NGRAPH
       REWIND 8
       IOUTPUT=-1
       KS=0
       KLOOP=-1
       PLOTT=0.
       READ(5,1000)NPLOTS,NPLOTE1,NCOMP
       NC6=NCOMP+6
       READ(5,1001)PLOTIME,TMAX
       IF(NPLOTS.EQ.0) GO TO 50
       READ(5,1001) ((PLOTS(I,J),J=1,4),I=1,NPLOTS)
  50   IF(NPLOTE1.EQ.0) GO TO 75
       READ(5,1001) ((PLOTE1(I,J),J=1,4),I=1,NPLOTE1)
  75   KLOOP=KLOOP+1
       READ(8,1010) TIME(1)
 100   READ(8,1010)XFLAG
       IF(XFLAG.LT.-999.9) GO TO 125
       NFLAG=XFLAG
       S(1,NFLAG,1)=XFLAG
       READ(8,1010) (S(1,NFLAG,J),J=2,NC6)
       GO TO 100
 125   KS=0
 150   KS=KS+1
       READ(8,1010)XFLAG
       IF(XFLAG.LT.-999.9) GO TO 175
       EC1(1,KS,1)=XFLAG
       READ(8,1010) (EC1(1,KS,J),J=2,4)
       GO TO 150
 175   IF(KLOOP.EQ.0) GO TO 225
       DO 200 I=1,KS
       LKPLTE(KS)=KS
 200   CONTINUE
 225   CALL OUTPUT(NS,NE,KLOOP)
       IF(TIME(1).GE.TMAX) GO TO 300
       GO TO 75
 300   CONTINUE
       STOP
1000   FORMAT(10I5)
1001   FORMAT(4F10.4)
1010   FORMAT(F15.7)
       END
```

2.4 The DYNSYS-B Library

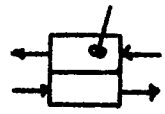
The unit computations (equipment subroutines) in the DYNSYS-B library are available to the user as FORTRAN source decks. In general, the subroutine names consist of four or five alphabetic characters describing the type of equipment, followed by one or two digits indicating the degree of sophistication of the model. The numerical value of the digits increases as the model is made more comprehensive.

Some of the equipment subroutines which follow were written specifically for use in simulating a liquid-liquid extraction unit, and may not be of general use. They are included in the library as a guide to the user who wishes to write additional unit computations.

Equipment subroutine symbols are given, indicating the equipment type number. The purpose of the subroutine, mathematical techniques, and required data input are described. A FORTRAN listing follows.

DYNSYS-B Subroutine Name - XTRACØI

Symbol



Purpose: This unit receives two mutually insoluble liquid feed streams, and calculates the mass transfer between the phases. It is assumed that the total mass hold-up of the vessel is constant.

Mathematical Description:

For any component (I), the material balances for the two phases are:

$$\frac{d(WF(I))}{dt} = FIN(I) - FOUT(I) + k_1 * Y(I) - k_2 * X(I)$$

$$\frac{d(WV(I))}{dt} = VIN(I) - VOUT(I) - k_1 * Y(I) + k_2 * X(I)$$

where :

FIN(I), FOUT(I) = flows of component (I) entering and leaving the vessel in phase F.

VIN(I), VOUT(I) = flows of component (I) entering and leaving the vessel in phase V.

Y(I) = concentration of component (I) in the vessel in the V phase

X(I) = concentration of component (I) in the vessel in the F phase.

k_1, k_2 = mass transfer coefficients.

The mass transfer coefficients obey the Arrhenius equation

$$k_1 = k_1^o * EXP(\alpha_1/T)$$

$$k_2 = k_2^o * EXP(\alpha_2/T)$$

where:

k_1^o, α_1, k_2^o, α_2 = constants

T = absolute temperature.

It is assumed that k_1 and k_2 are not functions of concentration. For each unit mass of component (I) which is transferred from one phase to the other, a quantity of heat $\lambda(I)$ is produced or absorbed. The total rate of heat production is therefore

$$\frac{d}{dt}(HEAT) = \sum_I (k_1 * Y(I) - k_2 * X(I)) * \lambda(I)$$

The specific heat and gravity for any component are assumed to be linear functions of temperature. Thus

$$CP(I) = \alpha_1 + \alpha_2 * T$$

$$SG(I) = \beta_1 + \beta_2 * T$$

For a mixture, the individual specific heats and gravities are added on a fractional basis. The total energy balance is therefore

$$\frac{dQ}{dt} = FIN * CP_F * TIN_F + VIN * CP_V * TIN_V$$
$$- FOUT * CP_F * TOUT - VOUT * CP_V * TOUT$$
$$+ \frac{d}{dt}(HEAT)$$

where: Q = heat

CP_F,CP_V = specific heats at corresponding temperatures

TIN_F,TIN_V,TOUT = inlet, outlet temperatures

Use of the AMØS routine allows calculation of total heat content of the vessel at time t^{n+1}. The temperature is then calculated using the Newton Raphson method.

Data Card Preparation:

XTRACØI uses ECI(J,K), BVI(,J,K) and EPI(J,K)

K =	1	2	3	4
ECI(J,K) Property =	Eq. No.	Flag	V-Phase Mass	F-Phase Mass
EPI(J,K) Property =	Eq. No.	Total Mass		

For the process matrix, the following convention is observed:

1st input stream - V phase feed

2nd input stream - F phase feed

1st output stream - V phase product

2nd output stream - F phase product

```
      SUBROUTINE XTRAC01(KLOOP,IM)
C     TYPE -1-
C     LIQUID/LIQUID EXTRACTOR WITH HEAT EFFECTS. CONSTANT MASS
C     EC1(I,J,K)    K=1-NO.    2-FLAG    3-WH(MASS HC PHASE)
C                   4-WS(MASS SOLVENT PHASE)
C     BV1(I,J,K)    K=1-NO.    2-DHEAT/DTIME    3,4...N-D(MASS HC PHASE COMPS
C                N+1,N+2...-D(MASS SOLVENT PHASE COMPS)/DTIME
C     EP1(I,J)    J=1-NO.    2-TOTAL VOLUME
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      DIMENSION RATE(10)
C     N1=ROW IN EC1()    N2=ROW IN BV1()    N3=ROW IN EP1()
C     STORAGE LOCATIONS
      N1=LOOK1(IM,1)
      N2=LOOK1(IM,2)
      N3=LOOK1(IM,3)
      MHO=-MP(IM,5)
      MHI=MP(IM,3)
      MSO=-MP(IM,6)
      MSI=MP(IM,4)
      TEMP=S(IG,MSO,5)
      T=1000./(460.+TEMP)-2.06
C     ADJUST TOTAL OUTFLOWS TO MAINTAIN CONSTANT MASS HOLD-UP
      IF(IG.EQ.2)GO TO 10
      X=EC1(1,N1,3)+EC1(1,N1,4)
      WO=S(1,MSI,6)+S(1,MHI,6)
      S(1,MSO,6)=WO*EC1(1,N1,4)/X
      S(1,MHO,6)=WO*EC1(1,N1,3)/X
C     CALCULATE DERIVATIVES AT TIME  N
   10 HT=0.
      DO 20 I=1,NCOMP
C     INTER PHASE TRANSFER RATES
      CK1=PP(I,5)*EXP(PP(I,6)*T)
      CK2=PP(I,7)*EXP(PP(I,8)*T)
      II=I+6
      RATE(I)=CK1*S(IG,MHO,II)-CK2*S(IG,MSO,II)
C     COMPONENT MASS BALANCES
      BV1(1,N2,I+2)=S(IG,MHI,6)*S(IG,MHI,I+6)-S(IG,MHO,6)*S(IG,MHO,I+6)
     1  -RATE(I)
      III=I+2+NCOMP
      BV1(1,N2,III)=S(IG,MSI,6)*S(IG,MSI,I+6)-S(IG,MSO,6)*S(IG,MSO,I+6)
     1   +RATE(I)
C     HEAT PRODUCED
   20 HT=HT+RATE(I)*PP(I,9)
      BV1(1,N2,2)=HT+S(IG,MSI,6)*S(IG,MSI,5)*S(IG,MSI,3)+S(IG,MHI,6)
     1  *S(IG,MHI,5)*S(IG,MHI,3)-(S(IG,MSO,6)*S(IG,MSO,3)+
     2  S(IG,MHO,6)*S(IG,MHO,3))*S(IG,MHO,5)
C
```

```
C     PREDICT/CORRECT VALUES AT TIME N+1
C     HEAT
      Y2=(EC1(2,N1,3)*S(2,MHO,3)+EC1(2,N1,4)*S(2,MSO,3))*S(2,MSO,5)
      IF(IG.EQ.1)X=(EC1(1,N1,3)*S(1,MHO,3)+EC1(1,N1,4)*S(1,MSO,3))
     1     *S(1,MSO,5)
      CALL AMOS(KLOOP,IG,1,Q,Y2,BV1(1,N2,2),BV1(2,N2,2),BV1(3,N2,2),X)
C     COMPONENT MASSES
      SUMH=0.
      SUMS=0.
      VH=0.
      VS=0.
      DO 40 I=1,NCOMP
      I6=I+6
      SG=PP(I,3)+PP(I,4)*S(1,MSO,5)
      Y2=EC1(2,N1,3)*S(2,MHO,I+6)
      IF(I.EQ.2) KK=1
      IF(I.NE.2) KK=-1
      IF(I.EQ.2)X=EC1(1,N1,3)*S(1,MHO,I6)
      II=I+2
      CALL AMOS(KLOOP,IG,KK,S(1,MHO,I6 ),Y2,BV1(1,N2,II),BV1(2,N2,II),
     1     BV1(3,N2,II),X)
      SUMH=SUMH+S(1,MHO,I6)
      VH=VH+S(1,MHO,I6)/SG
      Y2=EC1(2,N1,4)*S(2,MSO,I6 )
      II=I+2+NCOMP
      IF(I.LT.3) KK=1
      IF(I.LT.3)X=EC1(1,N1,4)*S(1,MSO,I6 )
      CALL AMOS(KLOOP,IG,KK,S(1,MSO,I+6),Y2,BV1(1,N2,II),BV1(2,N2,II),
     1     BV1(3,N2,II),X)
      SUMS=SUMS+S(1,MSO,I6)
   40 VS=VS+S(1,MSO,I6)/SG
C
C     NORMALIZE TO MAINTAIN TOTAL MASS CONSTANT
      FACT=EP1(N3,2)/(SUMS+SUMH)
      EC1(1,N1,3)=SUMH*FACT
      EC1(1,N1,4)=SUMS*FACT
C     OUTLET CONCENTRATIONS
      DO 60 I=1,NCOMP
      I6=I+6
      S(1,MHO,I6 )=S(1,MHO,I6 )/SUMH
   60 S(1,MSO,I6 )=S(1,MSO,I6 )/SUMS
C
C     NEWTON RAPHSON SEARCH FOR THE TEMPERATURE AT TIME N+1
      T1=S(1,MSO,5)
      X1=0.
      X2=0.
      X3=0.
      X4=0.
      DO 100 I=1,NCOMP
      II=I+6
      X1=X1+PP(I,1)*S(1,MSO,II)
      X2=X2+PP(I,2)*S(1,MSO,II)
      X3=X3+PP(I,1)*S(1,MHO,II)
  100 X4=X4+PP(I,2)*S(1,MHO,II)
```

```
      X1=X1*EC1(1,N1,4)
      X2=X2*EC1(1,N1,4)
      X3=X3*EC1(1,N1,3)
      X4=X4*EC1(1,N1,3)
      X5=X2+X4
      X6=X1+X3
   80 F=Q-(X6+X5*T1)*T1
      DF=-X6-2.0*X5*T1
      T2=T1-F/DF
      IF(ABS(T2-T1).LT.0.0001) GO TO 120
      T1=T2
      GO TO 80
  120 S(1,MSO,5)=T2
      S(1,MHO,5)=T2
C     THE NEW CP,S AND SG,S
      CPH=0.
      CPS=0.
      SGS=0.
      SGH=0.
      DO 125 I=1,NCOMP
      II=I+6
      CP=PP(I,1)+PP(I,2)*T2
      SG=PP(I,3)+PP(I,4)*T2
      CPH=CPH+CP*S(1,MHO,II)
      CPS=CPS+CP*S(1,MSO,II)
      SGH=SGH+SG*S(1,MHO,II)
  125 SGS=SGS+SG*S(1,MSO,II)
      S(1,MSO,3)=CPS
      S(1,MHO,3)=CPH
      S(1,MSO,4)=SGS
      S(1,MHO,4)=SGH
C     ADJUST TOTAL OUTFLOWS TO MAINTAIN CONSTANT MASS
      X=EC1(1,N1,3)+EC1(1,N1,4)
      WO=S(1,MSI,6)+S(1,MHI,6)
      S(1,MSO,6)=WO*EC1(1,N1,4)/X
      S(1,MHO,6)=WO*EC1(1,N1,3)/X
C     UPDATE BV1() AFTER PREDICTOR STEP
      IF(IG.EQ.1) GO TO 160
      NN=2*NCOMP+2
      DO 140 I=1,2
      II=3-I
      DO 140 J=1,NN
      BV1(II+1,N2,J)=BV1(II,N2,J)
  140 CONTINUE
      GO TO 200
C     UPDATE EC1() AFTER CORRECTOR STEP
  160 DO 180 J=1,4
      EC1(2,N1,J)=EC1(1,N1,J)
  180 CONTINUE
  200 RETURN
      END
```

DYNSYS-B Subroutine Name - LAGØ2

Symbol

Purpose: This unit represents a time lag, such as plug flow in a pipe. A "channeling" effect may be included (indicated by the broken line in the symbol). Only intensive properties are subject to time delay - there is no lag in bulk flow.

Subroutine Outline: A record of history points of the (single) inlet stream is kept and transferred to the (single) output stream after elapse of the time delay. A simple linear interpolation is used to estimate values between history points.

The user specifies the fraction of the input stream subject to time delay (ie. the remainder is the extent of channeling). The user also specifies the total number of history points we wish to store. The subroutine "spaces out" this number of history points over the length of the time delay.

The record of history points is stored in the SX(I,J,K) matrix.

Data Card Preparation:

Subroutine LAGØ2 uses EPI(J,K)

K =	1	2	3	4	5
Property =	Eq. No.	Time Lag	Fraction subject to lag.	Maximum number of history points	*

* EPI(J,5) for subroutine LAGØ2 is reserved for use in the program (See listing)

In the process matrix, the convention is:

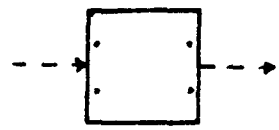
column	3	:	+	input stream number
column	4	:	-	output stream number.

```
      SUBROUTINE LAG02 (KLOOP,IM)
C     TYPE -2-
C     COMBINATION TIME LAG ELEMENT AND BYPASS
C     NOMENCLATURE   SX(I,J,K)   K=1-EQ.NO.     2-TIME OF VALUES
C           3-CP 4-SG 5-TEMP 6-FLOW 7,8..-COMP.CONCS.
C     EP1(I,J)   J=1-EQ.NO.    2-TIME LAG     3-FRACTION OF STREAM
C      PASSING THRU LAG  4-MAX.NO.VALUES TO BE STORED
C      5-TIME OF NEXT STORAGE (THIS CALCULATED IN SUBROUTINE)
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK6/NSX,SX(15,25,14)
C     NO DELAY IN FLOW RATE
      MI=MP(IM,3)
      MO=-MP(IM,4)
      S(1,MO,6)=S(1,MI,6)
C     BYPASS ROUTINE FOR CORRECTOR STEP
      IF(IG.EQ.1) GO TO 500
C     STORAGE LOCATIONS
      N3=LOOK1(IM,3)
      N4=LOOK1(IM,4)
      NV=EP1(N3,4)
      NC6=NCOMP+6
C     FOR FIRST LOOP,FILL WITH INITIAL VALUES
      IF(KLOOP.GT.1) GO TO 80
      IF(NV.GT.15) WRITE(6,1000) MP(IM,1)
      DO 40 I=1,NV
      SX(I,N4,1)=MP(IM,1)
      SX(I,N4,2)=0.
      DO 40 K=3,NC6
      SX(I,N4,K)=S(1,MO,K)
   40 CONTINUE
      EP1(N3,5)=EP1(N3,2)/EP1(N3,4)
C     TIME OF NEXT STORAGE
   80 T1=TIME(2)+EP1(N3,2)/EP1(N3,4)
C     STORE THE SOONEST VALUE
      IF(T1.LT.EP1(N3,5))EP1(N3,5)=T1
C     IS THIS VALUE CALCULATED YET
      IF(EP1(N3,5).GT.TIME(2)) GO TO 150
C     UPDATE SX() AND INSERT NEW VALUES
      NN=NV-1
      DO 100 J=1,NC6
      DO 100 I=1,NN
      II=NV-I
  100 SX(II+1,N4,J)=SX(II,N4,J)
      DO 120 J=3,NC6
      SX(1,N4,J)=S(2,MI,J)
  120 CONTINUE
```

```
      SX(1,N4,1)=MP(IM,1)
      SX(1,N4,2)=TIME(2)
C     DUMMY VALUE FOR NEXT STORAGE TIME
      EP1(N3,5)=1.0E+10
C     INTERPOLATION AND INSERTION OF VALUES INTO STREAM VECTOR
C     FIND INTERPOLATION POINT
 150  T=TIME(1)-EP1(N3,2)
      IF(T.LT.0.0) GO TO 200
      DO 160 I=1,NV
      IF(SX(I,N4,2).LT.T) GO TO 170
 160  CONTINUE
 170  IF(I.EQ.1)I=I+1
      FACT=(T-SX(I,N4,2))/(SX(I-1,N4,2)-SX(I,N4,2))
      GO TO 220
 200  I=NV
      FACT=0.0
 220  EE=1.0-EP1(N3,3)
      DO 240 J=3,NC6
      IF(J.EQ.6) GO TO 240
      S(1,MO,J)=EP1(N3,3)*(SX(I,N4,J)*(1.-FACT)+SX(I-1,N4,J)*FACT)
     1                        +EE*S(1,MI,J)
 240  CONTINUE
 500  RETURN
 1000 FORMAT(1H0,20(1H*),30HMISTAKE IN LAGO2,EQ.NO. IS      ,I10,50(1H*),
      END
```

DYNSYS-B Subroutine Name - CNTRLØI

Symbol



Purpose: Simulation of a 3-mode controller.

Mathematical Description: The input stream to the controller is the measured value of the process variable being controlled (x). The error (ε') is the difference between this value and the set-point (S.P.). Thus

$$\varepsilon' = x - S.P.$$

This error is scaled as a percentage of the range of the measuring device (ε). The output signal (Y) from the controller is a result of three terms: proportional, integral and derivative action. Thus

$$Y = K_p * \varepsilon + K_I * \int_0^t \varepsilon dt + K_D \frac{d\varepsilon}{dt}$$

where K_P, K_I, K_D = proportional, integral and derivative constants respectively.

Over a small interval of time (Δt) between time t^{n+1} and t^n, we have

$$Y^{n+1} - Y^n = K_p * (\varepsilon^{n+1} - {}^n) + K_I * \int_{t_n}^{t^{n+1}} \varepsilon dt$$

$$+ K_D * \{ (\frac{d\varepsilon}{dt})^{n+1} - (\frac{d\varepsilon}{dt})^n \}$$

Replacing the integral and derivative expressions by simple two-point finite difference approximations, we get

$$Y^{n+1} = Y^n + K_p*(\varepsilon^{n+1}-\varepsilon^n) + K_I*(\varepsilon^{n+1}+\varepsilon^n) * \frac{\Delta t}{2}$$

$$+ K_D * \left(\frac{\varepsilon^{n+1} - \varepsilon^n}{t^{n+1} - t^n} - \frac{\varepsilon^n - \varepsilon^{n-1}}{t^n - t^{n-1}}\right)$$

<u>Data Card Preparation:</u> Subroutine CNTRLØI uses BV2( ) and EP2(J,K)

K =	1	2	3	4	5
Property =	Eq. No.	Property Being Controlled	Minimum Value	Maximum Value	Set Point

K =	6	7	8
Property =	Proportional Const (K_p)	Integral Const.(K_I)	Derivative Const.(K_D)

Notes: K = 2, the property being controlled is the position of the variable in the stream list, that is the value of KK in S(I,J,KK) for the input stream.

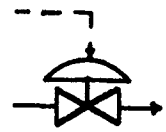
K = 3,4, the minimum and maximum values, represent the range of the measuring device.

In the process matrix, the input signal is entered in column (3) and the output signal (with a minus sign) in column (4).

```
      SUBROUTINE CNTRL01  (IM)
C     THREE MODE CONTROLLER  EQ.TYPE NUMBER IS  -3-
C     NOMENCLATURE  BV2(I,J,K)   K=1-EQ.NO.   2-ERROR
C       EP2(J,K)   K=1-EQ.NO.   2-PROPERTY BEING CONTROLLED  (5=TEMP,
C                                6=LEVEL,7,8...=CONCENTRATIONS)
C            3-MIN.VALUE    4-MAX.VALUE    5-SET POINT    6-PROP.CONST.
C             7-INTEGRAL CONST.    8-DERIVATIVE CONST.
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1    LOOK1(85,4)
      COMMON/BLK5/NEC2,MEC2,NBV2,NEP2,MEP2,EC2(1 ,1 ,1 ),BV2(3,25,3),
     1  EP2(15,10)
C     STORAGE LOCATIONS
      N2=LOOK1(IM,2)
      N3=LOOK1(IM,3)
      MI=MP(IM,3)
      MO=-MP(IM,4)
      NPR=EP2(N3,2)
C     SCALE THE ERROR            (-50 ...... +50)
      ER=(S(1,MI,NPR)-EP2(N3,5))/(EP2(N3,4)-EP2(N3,3))*50.
      IF(ER.LT.-50.)ER=-50.
      IF(ER.GT.50.)ER=50.
      BV2(1,N2,2)=ER
C     PROPORTIONAL TERM
      ER=ER-BV2(2,N2,2)
C     INTEGRAL TERM
      ERI=(TIME(1)-TIME(2))*(BV2(1,N2,2)+BV2(2,N2,2))/2.
C     DERIVATIVE TERM
      ERD=(BV2(1,N2,2)-BV2(2,N2,2))/(TIME(1)-TIME(2))
      BV2(1,N2,3)=ERD
      ERD=ERD-BV2(2,N2,3)
C     THE OUTPUT SIGNAL
      S(1,MO,6)=S(2,MO,6)+EP2(N3,6)*ER+EP2(N3,7)*ERI+EP2(N3,8)*ERD
C     UPDATE BV2() AFTER THE CORRECTOR STEP
      IF(IG.EQ.2) GO TO 50
      DO 25 K=1,3
      BV2(2,N2,K)=BV2(1,N2,K)
   25 CONTINUE
   50 RETURN
      END
```

DYNSYS-B Subroutine Name - VALVØI

Symbol



Purpose: This routine simulates a control valve with a parabolic characteristic. (such as a V-Port valve). Constant pressure drop across the valve is assumed.

Description: The input signal to the valve dome may range between 0 (psig) and 100 (psig) representing the range of valve stem travel. The flow through the valve is parabolically related to the input signal:

DIRECT ACTION : FLOW = CONST * $(SIGNAL)^2$

REVERSE ACTION : FLOW = CONST * $(100. - SIGNAL)^2$

Data Card Preparation: VALVØI uses EPI(J,K)

K =	1	2	3
Property =	Eq. No.	VALVE CONST.	ACTION

Note: For direct action, EPI(J,3) = +1.

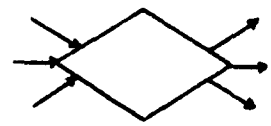
For reverse action, EPI(J,3) = -1.

In the process matrix, the valve signal is entered into column (3) (with a + sign) and the material flow stream number into column (4) (with a - sign).

```
      SUBROUTINE VALV01 (IM)
C     TYPE -4-
C     V-PORT (PARABOLIC) CONTROL VALVE
C     NOMENCLATURE  EP1(I,J)  J=1-EQ.NO.  2-VALVE CONSTANT
C                        3-ACTION (+=DIRECT,-=REVERSE)
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
C     STORAGE LOCATIONS
      N3=LOOK1(IM,3)
      MI=MP(IM,3)
      MO=-MP(IM,4)
C     CHECK THAT THE VALVE SIGNAL IS IN THE RANGE   0 ....100
      V=S(1,MI,6)
      IF(V.LT.0.)V=0.
      IF(V.GT.100.)V=100.
C     ACTION
      A=EP1(N3,3)
      IF(A.LT.0.)S(1,MO,6)=EP1(N3,2)*(100.-V)**2
      IF(A.GE.0.)S(1,MO,6)=EP1(N3,2)*V**2
      RETURN
      END
```

DYNSYS-B Subroutine Name - MXPLTØ2

Symbol



Purpose: Mixing of a number of input streams. Total input is split up into a number of output streams.

Description: The total heat and component mass input flows are calculated, and the resulting "mixed" concentrations are found. The "mixed" temperature is calculated by the Newton Raphson technique. Each output total flow rate is a constant fraction of the total input flow rate. All output flows have the same temperature and concentrations. The specific heat of an input flow is the weighted average of the specifications of its components. Each component specific heat is a linear function of temperature.

Note: The use of the Newton Raphson technique is not necessary, as the "mixed" temperature may be determined analytically. The technique has been adopted, however, to facilitate adaption of subroutine MXPLTØ2 to the case where component specific heats are not linear functions of temperature.

Data Card Preparation: Subroutine MXPLTØ2 uses EPI(J,K).

K =		1	2	3 .....
Property =	Equipment Number		α_1	α_2 .....

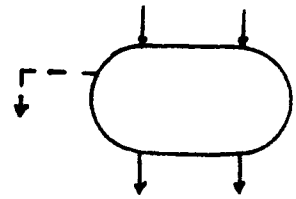
where $\alpha(I)$ is the fraction of total flow input leaving in the Ith output stream (as listed in the process matrix).

```
      SUBROUTINE MXPLT02 (IM)
C     TYPE -5-
C     JUNCTION/CONSTANT FRACTION SPLITTER
C     NOMENCLATURE  EP1()  1-EQ.NO.   2...5-FRACTIONS OF TOTAL INPUT
C                  IN OUTPUT STREAMS AS LISTED IN PROCESS MATRIX
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1    LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
C     STORAGE LOCATION
      N3=LOOK1(IM,3)
      NC6=NCOMP+6
C     TOTAL MASS AND HEAT INPUTS
      FT=0.
      Q=0.
      DO 25 I=3,NMP
      IS=MP(IM,I)
      IF(IS.LT.0) GO TO 35
      FT=FT+S(1,IS,6)
   25 Q=Q+S(1,IS,6)*S(1,IS,5)*S(1,IS,3)
   35 IIS=-IS
      NI=I-1
C     COMPUTE CONCENTRATIONS AND INSERT VALUES IN 1ST OUTPUT STREAM
      DO 50 K=7,NC6
      SUM=0.
      DO 40 I=3,NI
      IS=MP(IM,I)
   40 SUM=SUM+S(1,IS,6)*S(1,IS,K)
   50 S(1,IIS,K)=SUM/FT
C     NEWTON RAPSON SEARCH FO R THE TEMPERATURE
      T1=S(1,IIS,5)
      A1=0.
      B1=0.
      DO 70 I=1,NCOMP
      II=I+6
      A1=A1+PP(I,1)*S(1,IIS,II)
   70 B1=B1+PP(I,2)*S(1,IIS,II)
   90 F=Q-FT*T1*(A1+B1*T1)
      DF=-FT*(A1+2.0*B1*T1)
      T2=T1-F/DF
      IF(ABS(T2-T1).LT.0.0001) GO TO 100
      T1=T2
      GO TO 90
C     CALCULATE CP AND SG
  100 CP=A1+B1*T2
      SG=0.
      DO 110 I=1,NCOMP
      II=I+6
  110 SG=SG+(PP(I,3)+PP(I,4)*T2)*S(1,IIS,II)
```

```
C       INSERT VALUES IN OUTPUT STREAMS
        NI=NI+1
        DO 150 J=NI,NMP
        IS=-MP(IM,J)
        IF(IS.EQ.0) GO TO 175
        S(1,IS,3)=CP
        S(1,IS,5)=T2
        S(1,IS,4)=SG
        NII=J-NI+2
        S(1,IS,6)=EP1(N3,NII)*FT
        DO 150 K=7,NC6
        S(1,IS,K)=S(1,IIS,K)
  150   CONTINUE
  175   RETURN
        END
```

DYNSYS-B Subroutine Name - SETLØI

Symbol



Purpose: This unit represents a settler with two insoluble liquid phases.

Description: Total mass balances are performed for each phase. This subroutine does not carry out individual component mass or heat balances; (temperatures and concentrations are transmitted directly from input to output streams). The total hold-up of the light phase is transmitted as a controller signal. Total volume of the material in the vessel is held constant.

Remarks: This subroutine was written for use in combination with LAGØ2 to represent a horizontally baffled two phase liquid settler.

Data Card Preparation: Subroutine SETLØI uses ECI(I,J,K), BV2(I,J,K) and EPI(J,K)

K =	1	2	3	4
ECI( ) Property = Eq. No.		FLAG	Light Phase Mass	Heavy Phase Mass
EPI( ) Property = Eq. No.		Total Volume		

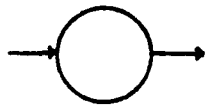
In the process matrix:

1st input stream = light phase feed
2nd input stream = heavy phase feed
1st output stream= light phase product
2nd output stream= heavy phase product
3rd output stream= controller signal (light phase hold-up)

```
      SUBROUTINE SETL01 (KLOOP,IM)
C     TYPE -6-
C     HOLD-UP TANK (SETTLER)   H.C.PHASE MASS BALANCE ONLY
C     NOMENCLATURE
C     EC1()  K=U-EQ.NO.   2-FLAG   3-HC PHASE MASS   4-SOLVENT PHASE MASS
C     BV2()  K=1-EQ.NO.   2-D(HC PHASE MASS)/D(TIME)
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      COMMON/BLK5/NEC2,MEC2,NBV2,NEP2,MEP2,EC2(1 ,1 ,1 ),BV2(3,25,3),
     1  EP2(15,10)
C     STORAGE LOCATIONS
      N1=LOOK1(IM,1)
      N2=LOOK1(IM,2)
      N3=LOOK1(IM,3)
      MHI=MP(IM,3)
      MSI=MP(IM,4)
      MHO=-MP(IM,5)
      MSO=-MP(IM,6)
      MSIG=-MP(IM,7)
C     CALCULATE DERIVATIVE
      BV2(1,N2,2)=S(IG,MHI,6)-S(IG,MHO,6)
C     PREDICT/CORRECT  NEW VALUES
      CALL AMOS(KLOOP,IG,-1,EC1(1,N1,3),EC1(2,N1,3),BV2(1,N2,2),BV2(2,
     1   N2,2),BV2(3,N2,2),EC1(1,N1,3) )
C     TRANSMIT INTENSIVE VALUES FROM INPUT STREAMS TO OUTPUT STREAMS
      NC6=NCOMP+6
      DO 30 I=3,NC6
      IF(I.EQ.6) GO TO 30
      S(1,MHO,I)=S(1,MHI,I)
      S(1,MSO,I)=S(1,MSI,I)
   30 CONTINUE
C     THE NEW SOLVENT PHASE HOLD-UP
      V=EC1(1,N1,3)/S(1,MHI,4)
      EC1(1,N1,4)=(EP1(N3,2)-V)*S(1,MSI,4)
      S(1,MSIG,6)=EC1(1,N1,3)
C     UPDATE BV2() AFTER THE PREDICTOR STEP
      IF(IG.EQ.1) GO TO 80
      DO 50 I=1,2
      II=3-I
      DO 50 K=1,3
      BV2(II+1,N2,K)=BV2(II,N2,K)
   50 CONTINUE
      GO TO 120
C     UPDATE EC1() AFTER THE CORRECTOR STEP
   80 DO 100 K=1,4
      EC1(2,N1,K)=EC1(1,N1,K)
  100 CONTINUE
  120 RETURN
      END
```

DYNSYS-B Subroutine Name - SETTØI

Symbol



Purpose: This unit sets the stream temperature to a specified value.

Description: This is the simplest representation of an exit temperature controlled heat exchanger. Values in the (single) input stream list are transmitted directly to the output stream list. The output temperature is set to a specified value, and the corresponding specific heat and specific gravity are calculated. It is assumed that specific heat and gravity for a pure component are linear functions of temperature, and for a mixture the pure component values are added on a fractional basis.

Data Card Preparation: The subroutine uses EPI(J,K)

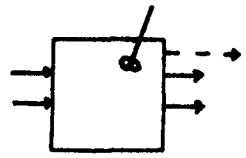
K =	1	2
Property =	Eq. No.	Temperature

In the process matrix, the input stream number (with a + sign) is entered in column (3), and the output stream number (with a - sign) in column (4).

```
      SUBROUTINE SETTU1(IM)
C     TYPE -7-
C     THIS ROUTINE SETS A STREAM TEMP.AND CALCULATES THE
C      CORRESPONDING CP AND SG.
C     NOMENCLATURE  EP1()  1-EQ.NO.   2-TEMP
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1    LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
      N3=LOOK1(IM,3)
      MI=MP(IM,3)
      MO=-MP(IM,4)
      NC6=NCOMP+6
      CP=0.
      SG=0.
      DO 20 I=7,NC6
      II=I-6
      S(1,MO,I)=S(1,MI,I)
      CP=CP+(PP(II,1)+PP(II,2)*EP1(N3,2))*S(1,MI,I)
   20 SG=SG+(PP(II,3)+PP(II,4)*EP1(N3,2))*S(1,MI,I)
      S(1,MO,6)=S(1,MI,6)
      S(1,MO,5)=EP1(N3,2)
      S(1,MO,4)=SG
      S(1,MO,3)=CP
      RETURN
      END
```

SYNSYS-B Subroutine Name - STIRØI

Symbol



Purpose: A perfectly-stirred tank

Description: Heat and component mass balances are performed. Information streams such as controller signals are flagged with absolute value greater than 10., and are not included in the heat and mass balances.

Component specific heats and gravities are linear functions of temperature. The specific heat and gravity of a mixture is the sum of the component values added on a fractional basis. The AMØS routine is used to estimate total heat content of the vessel at the new time t^{n+1}, and the Newton Rapson method is used to calculate the temperature.

Data Card Preparation: Subroutine STIRØI uses

ECI(I,J,K), BVI(I,J,K) and EPI(J,K). For ECI(I,J,K) -

K =	1	2	3
Property =	Eq. No.	Flag	Total Mass Hold-up

For EPI(J,K)

K =	1	2
Property =	Eq. No.	0.0

In EPI(J,2) the position of the first output stream is stored. This is calculated in the subroutine - the user simply enters a zero in the data deck.

```
      SUBROUTINE STIR01 (KLOOP,IM)
C     SIMPLE WELL STIRRED TANK
C     EC1(I,J,K)  K=1-EQ.NO.   2-FLAG   3-TOTAL MASS
C     BV1(I,J,K)  K=1-EQ.NO. 2-D(HEAT)/D(TIME) 3,4..D(COMP.MASS)/D(TIME)
C     EP1(J,K)    K=1-EQ.NO.   2-POSITION OF FIRST OUTPUT STREAM
C                          (THIS IS CALCULATED WITHIN THE SUBROUTINE)
C     ASSUMED THAT CP,SG FOR PURE COMPONENTS ARE LINEAR FUNCTIONS OF
C     TEMPERATURE.WEIGHTED AVERAGES TAKEN FOR A MIXTURE
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      COMMON/BLK2/NEC1,MEC1,NBV1,NEP1,MEP1,EC1(2,25,5),BV1(3,15,18),
     1  EP1(65,5)
C     STORAGE LOCATIONS
      N1=LOOK1(IM,1)
      N2=LOOK1(IM,2)
      N3=LOOK1(IM,3)
      NC6=NCOMP+6
      NC2=NCOMP+2
      IF(KLOOP.GT.1) GO TO 40
C     FIND THE FIRST OUTPUT STREAM
      DO 25 J=3,NMP
      IS=IABS(MP(IM,J))
      IF(ABS(S(1,IS,2)).GT.10.) GO TO 25
      IF(MP(IM,J).LT.0) GO TO 35
   25 CONTINUE
   35 EP1(N3,2)=J
      IF(ABS(S(1,IS,2)).GT.10.) WRITE(6,500) MP(IM,1)
  500 FORMAT(1H0,20X,15HERROR EQ. NO.  ,I5,23HONLY OUTPUT IS SIGNAL   )
   40 NOUT=EP1(N3,2)
      IOUT=-MP(IM,NOUT)
      T1=S(1,IOUT,5)
C     CALCULATION OF DERIVATIVES
      DO 50 J=2,NC2
      BV1(1,N2,J)=0.
   50 CONTINUE
      DO 100 J=3,NMP
      IS=IABS(MP(IM,J))
      IF(IS.EQ.0) GO TO 100
C     DO NOT INCLUDE CONTROLLER SIGNALS IN MASS AND HEAT BALANCES
      IF(ABS(S(1,IS,2)).GT.10.) GO TO 100
      SIGN=IS/MP(IM,J)
C     HEAT
      BV1(1,N2,2)=BV1(1,N2,2)+SIGN*S(IG,IS,3)*S(IG,IS,5)*S(IG,IS,6)
C     MASS BALANCES
      DO 75 K=3,NC2
      KK=K+4
   75 BV1(1,N2,K)=BV1(1,N2,K)+SIGN*S(IG,IS,KK)*S(IG,IS,6)
  100 CONTINUE
```

```
C       PREDICT/CORRECT NEW VALUES
C       MASSES
        SUM=0.
        DO 150 K=7,NC6
        KK=K-4
        Y2=EC1(2,N1,3)*S(2,IOUT,K)
        IF(IG.EQ.1) X=EC1(1,N1,3)*S(1,IOUT,K)
        CALL AMOS (KLOOP,IG,1,S(1,IOUT,K),Y2,BV1(1,N2,KK),BV1(2,N2,KK),
       1           BV1(3,N2,KK),X)
 150    SUM=SUM+S(1,IOUT,K)
C       HEAT
        Y2=EC1(2,N1,3)*S(2,IOUT,5)*S(2,IOUT,3)
        IF(IG.EQ.1) X=EC1(1,N1,3)*S(1,IOUT,5)*S(1,IOUT,3)
        CALL AMOS (KLOOP,IG,1,S(1,IOUT,5),Y2,BV1(1,N2,2),BV1(2,N2,2),
       1            BV1(3,N2,2),X)
C       THE NEW CONCENTRATIONS AND HOLD-UP
        EC1(1,N1,3)=SUM
        DO 175 K=7,NC6
        S(1,IOUT,K)=S(1,IOUT,K)/SUM
 175    CONTINUE
C       THE NEW TEMPERATURE
        X1=0.
        X2=0.
        X3=0.
        X4=0.
        DO 185 I=1,NCOMP
        II=I+6
        X1=X1+PP(I,1)*S(1,IOUT,II)
        X2=X2+PP(I,2)*S(1,IOUT,II)
        X3=X3+PP(I,3)*S(1,IOUT,II)
 185    X4=X4+PP(I,4)*S(1,IOUT,II)
 190    F=S(1,IOUT,5)-(X1+X2*T1)*SUM*T1
        DF=-SUM*(2.*X2*T1+X1)
        T2=T1-F/DF
        IF(ABS(T2-T1).LT.0.0001) GO TO 200
        T1=T2
        GO TO 190
 200    S(1,IOUT,5)=T2
C       THE NEW CP AND SG
        CP=X1+X2*T2
        SG=X3+X4*T2
```

```
C       INSERT THE NEW VALUES IN THE OUTPUT STREAMS
        DO 225 J=3,NMP
        IS=-MP(IM,J)
        IF(IS.LE.0) GO TO 225
        S(1,IS,3)=CP
        S(1,IS,4)=SG
        S(1,IS,5)=S(1,IOUT,5)
        DO 220 K=7,NC6
        S(1,IS,K)=S(1,IOUT,K)
 220    CONTINUE
C       FOR CONTROLLER OUTPUT,HOLD-UP STORED   IN PLACE OF FLOW
        IF(ABS(S(1,IS,2)).GT.10.) S(1,IS,6)=SUM
 225    CONTINUE
C       UPDATE BV1() AFTER PREDICTOR STEP
        IF(IG.EQ.1) GO TO 275
        DO 250 I=1,2
        II=3-I
        DO 250 K=1,NC2
        BV1(II+1,N2,K)=BV1(II,N2,K)
 250    CONTINUE
        RETURN
C       UPDATE EC1() AFTER THE CORRECTOR STEP
 275    DO 300 K=1,3
        EC1(2,N1,K)=EC1(1,N1,K)
 300    CONTINUE
        RETURN
        END
```

2.5 Illustrative Examples on the Use of DYNSYS-B in Solving Transient Problems

Example 1

The equipment subroutines used in this example are in the DYNSYS-B library, and are described fully in Section 2.4 of this report. They are:

Subroutine STIRØ1 -

A simple well-stirred tank

Equipment type number = 8

STIRØ1 uses EC1, BV1 and EP1

On an information flow diagram, this model is represented by:

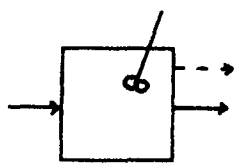


Figure 2-4 Information Flow Diagram Representation of STIRØ1

Subroutine MXPLTØ2 -

This routine models the junction of several input streams, and splits up the input such that each output stream is a constant fraction of the total input.

Equipment type number = 5

MXPLTØ2 uses EP1.

The graphical representation is:

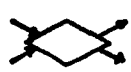


Figure 2-5 Information Flow Diagram Representation of MXPLTØ2

The system to be simulated is shown in Figure 2-6.

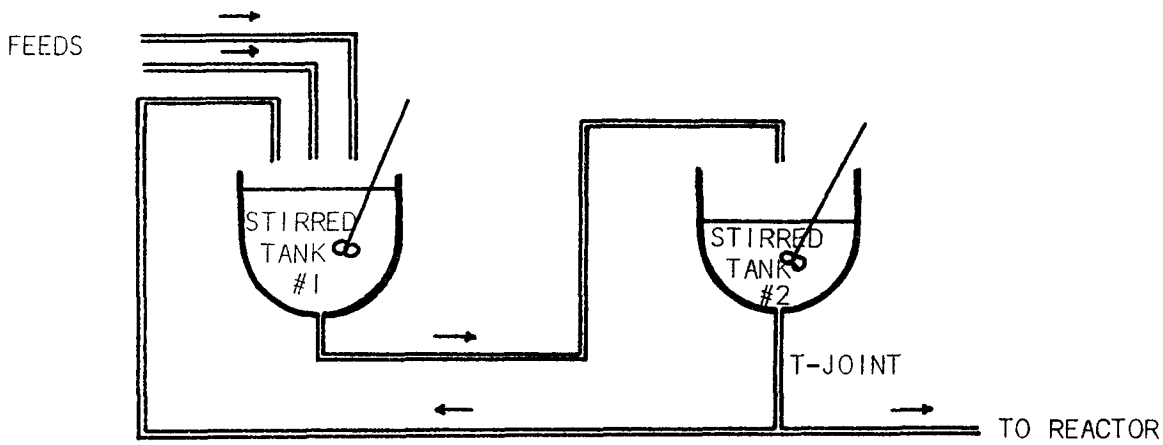


Figure 2-6 System for Example I

Three chemical components (A, B, C) are present in the system, which is used to blend the components into proper proportions before feeding the mixture to a reactor further down-stream. We are interested in observing the time response of the system to a disturbance in concentration of one of the feed streams to the first stirred-tank. The stirred-tanks are perfectly mixed, and by reason of constant pressure drops in the pipes we can assume that the flows leaving the T-joint are constant fractions of the flow to the T-joint.

The information flow diagram for the blending system is shown below:

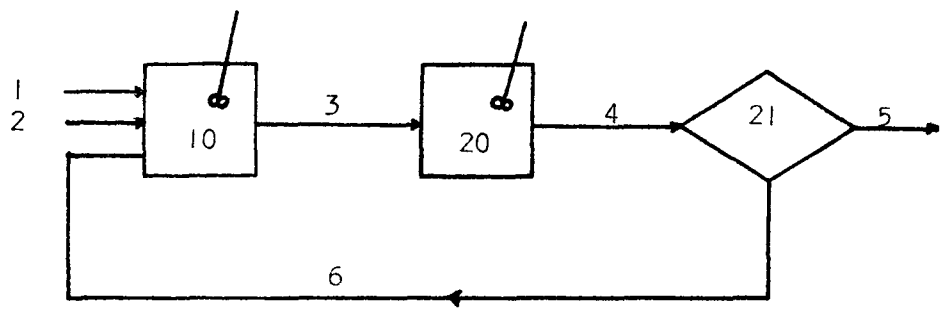


Figure 2-7 Information Flow Diagram for Blending System of Example 1

We note that equipment numbers may be assigned in any sequence. However, streams are numbered consequentively starting from 1. The process matrix is now constructed:

Eq. No.	Eq. Type No.	Associated Streams			
10	8	+1	+2	+6	-3
20	8	+3	-4	0	0
21	5	+4	-5	-6	0

The Physical Property Table

The subroutines STIRØ1 and MXPLTØ2 make the assumptions that the specific heat and specific gravity for any pure component are linear

functions of temperature. Thus

$$CP = \alpha_1 + \alpha_2 * T$$

$$SG = \beta_1 + \beta_2 * T$$

where

CP = specific heat for pure chemical component

SG = specific gravity for pure chemical component

T = temperature

$\alpha_1, \alpha_2, \beta_1, \beta_2$ = constants

The units selected are arbitrary, but must of course be consistent throughout the system. The constants α_1, α_2, β_1, β_2 are assumed to be in columns 1, 2, 3, 4 respectively in the physical property (PP) table. For a mixture of components, the specific heats and gravities of the components are added on a fractional basis. Given the physical data, the PP( ) table will be:

Component	α_1	α_2	β_1	β_2
A	0.70	-0.0001	0.80	-0.004
B	0.65	-0.0005	0.85	-0.003
C	0.60	+0.008	0.80	-0.002

For example, at T = 100

$$CP(A) = 0.70 + (-0.0001) * 100 = 0.69$$

$$CP(B) = 0.65 + (-0.0005) * 100 = 0.60$$

The specific heat of a mixture consisting of 70% of component A and 30% of component B will be:

$$CP(A,B) = 0.69 * 70\% + 0.60 * 30\% = 0.663$$

The Stream Matrix - S(I,J,K)

Subroutines STIRØI and MXPLTØ2 assume that values of the parameter K are chosen as follows:

K =	1	2	3	4
Property =	Stream No.	Flag	Spec. Heat	Spec. Gravity

K =	5	6	7,8, ...
Property =	Temperature	Tot. Mass Flow Rate	Concentrations of Components A,B, ...

Given the steady-state values for the system, we can write down the stream matrix.

No.	FLAG	CP	SG	TEMP.	FLOW	CONC(A)	CONC(B)	CONC(C)
1.	-1.	0.	0.	50.	1000.	1.000	0.000	0.000
2.	-1.	0.	0.	50.	2000.	0.000	0.750	0.250
3.	+1.	0.	0.	50.	4000.	0.333	0.500	0.167
4.	+1.	0.	0.	50.	4000.	0.333	0.500	0.167
5.	+1.	0.	0.	50.	3000.	0.333	0.500	0.167
6.	-1.	0.	0.	50.	1000.	0.333	0.500	0.167

Remarks:

a) Streams 1, 2 and 6 are flagged negatively and will not appear in the output.

b) The values for specific heat and gravity are entered simply as zeroes. As explained in the discussion on subroutine DYNI, these values will be calculated automatically during the reading in of the data.

c) The values in the stream matrix as written above are steady state values. To study the effect of a step change in concentrations of feed stream 2 , we would rewrite the list for stream 2 as follows:

2.	-1.	0.	0.	50.	2000.	0.000	0.500	0.500

We thereby introduce a step change in the initial conditions (i.e. at time zero).

The Equipment Conditions Matrix - ECI(I,J,K)

In this example, only subroutine STIRØI requires storage in the ECI( ) matrix. For STIRØI, the values of parameter K are chosen as follows:

K =	1	2	3
Property =	Eq. No.	Flag	Total Mass Hold-up

Suppose that the first tank contains 10,000 lbs and the second contains 2,500 lbs. We are not interested in obtaining a print out of either vessel in ECI, and therefore assign negative flag numbers.

Eq. No.	Flag	Total Mass Hold-up
10.	-1.	10000.0
20.	-1.	2500.0

The Equipment Parameter Matrix - EPI(J,K)

Both STIRØI and MXPLTØ2 use EPI( ). The values of parameter K for MXPLTØ2 are:

K =	1	2	3	4,5,..
Property =	Eq. No.	Fraction of total input in first output stream (order of occurrence in MP( ))	Fraction of total input in second output stream (order of occurrence in MP( ))	etc.

The output streams are 5 , 6 and occur in that order in the process matrix. The flows are 3000 lbs/hr (75% of total input) and 1000 lbs/hr (25% of total input) respectively. STIRØI uses EPI( ) to store the position of the first output stream - this number is calculated within the subroutine itself and the user need simply enter zeroes.

The EPI( ) matrix is therefore:

Eq. No.		
10.	0.	0.
20.	0.	0.
21.	0.75 (1st output)	0.25 (2nd output)

Other Matrices

Data input is never required for BV1( ) and BV2( ). For this example, EC2( ) and EP2( ) are not required.

System Constants

NMP	= 6	(number column's in process matrix)
NE	= 3	(number equipments)
NS	= 6	(number streams)
NCOMP	= 3	(number chemical components)
NPRINT	= 10	(print-out results every 10 time increments)
NPP	= 4	(number columns in PP( ) matrix)
IOUTPUT	= +1	(output mode = print)
MEC1	= 3	(number columns for EC1 matrix)
MEC2	= 0	(number columns for EC2 matrix)
MEP1	= 3	(number columns for EP1 matrix)
MEP2	= 0	(number columns for EP2 matrix)
DELT	= 0.005	(initial time increment - approximately $\frac{1}{100}$ th of the smallest time constant in the system. The time constant for the second tank is $\frac{2500 \text{ lbs}}{4000 \text{ lbs/hr}} \approx 0.6$ hrs.)
TMAX	= 10.0	(final time - we anticipate that the system will be close to steady-state after 10 hours)
TOLL	= 0.000005	(lower limit for truncation error).
TOLU	= 0.00001	(upper limit for truncation error)

The choice of TOLL and TOLU is somewhat arbitrary. We have found that the values chosen above generally give satisfactory results. It is advisable to rerun the simulation with lower error limits as a check. (see also Appendix 11).

Data Deck for Example I

The data must now be assembled in the proper sequence:

1. System Constants (in the order shown above)
2. Process Matrix (MP())
3. Physical Property Matrix (PP())
4. Stream Matrix (S())
5. Equipment Conditions Matrix (EC1())
6. Equipment Conditions Matrix (EC2())
7. Equipment Parameter Matrix (EP1())
8. Equipment Parameter Matrix (EP2())

Format:

All integer numbers are entered in I5 format.

All floating point numbers are entered in 5F10. format.

The data deck for the example (including a step change in stream 2 concentations) is listed below. The punched card data fields are indicated.

FIELD	5	10	15	20	25	30	35	40	45	50	55
	6	3	6	3	10	4	1	3	0	3	0
	0.005		10.		0.000005		0.00001				
	10	8	1	2	6	-3					
MP( )	20	8	3	-4	0	0					
	21	5	4	-5	-6	0					
	.70		-.0001		.80		-.004				
PP( )	.65		-.0005		.85		-.003				
	.60		+.008		.80		-.002				
	1.		-1.		0.		0.		50.		
	1000.		1.00		0.		0.				
	2.		-1.		0.		0.		50.		
	2000.		0.0		0.5		0.5				
	3.		1.		0.		0.		50.		
S( )	4000.		0.333		0.500		0.167				
	4.		1.		0.		0.		50.		
	4000.		0.333		0.500		0.167				
	5.		1.		0.		0.		50.		
	3000.		0.333		0.500		0.167				
	6.		-1.		0.		0.		50.		
	1000.		0.333		0.500		0.167				
ECI( )	10.		-1.		10000.						
	20.		-1.		2500.						
	10.		0.		0.						
EPI( )	20.		0.		0.						
	21.		0.75		0.25						

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ D A T A I N P U T $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

DYNSYS-B Output for Example #1

PROCESS MATRIX

EQ.NO.	TYPE	IN STREAMS		OUT STREAMS	
10	8	1	2	6	-3
20	8	3	-4	0	0
21	5	4	-5	-6	0

PHYSICAL PROPERTIES

COMPONENT	CONSTANTS			
1	7.000E-01	-1.000E-04	8.000E-01	-4.000E-03
2	6.500E-01	-5.000E-04	8.500E-01	-3.000E-03
3	6.000E-01	8.000E-03	8.000E-01	-2.000E-03

EQUIPMENT PARAMETERS

-10-	0.0000	0.0000
-20-	0.0000	0.0000
-21-	.7500	.2500

SYSTEM CONSTANTS

3	-NE=NO.EQUIPMENTS	6	-NMP=NO.COL.S IN PROC.MATRIX
3	-NCOMP=NO.COMPONENTS	6	-NO.STREAMS
10	-NPRINT=PRINT INTERVAL	.00500	-DELT=INIT.TIME STEP
.0000100	-TOLU=UPPER ERROR TOL	.0000050	-TOLL=LOWER ERROR TOL
2	-NEC1=NO.EQUIPS IN EC1	0	-NEC2=NO.EQUIPS IN EC2
2	-NBV1=NO.EQUIPS IN BV1	0	-NBV2=NO.EQUIPS IN BV2
3	-NEP1=NO.EQUIPS IN EP1	0	-NEP2=NO.EQUIPS IN EP2

INITIAL CONDITIONS

STREAM	FLAG	CP	SG	TEMP	TOT.FLOW	CONCENTRATIONS		
1	-1	.6950	.6000	50.0000	1000.0000	1.00000000	0.00000000	0.00000000
2	-1	.8125	.7000	50.0000	2000.0000	0.00000000	.50000000	.50000000
3	1	.7109	.6667	50.0000	4000.0000	.33300000	.50000000	.16700000
4	1	.7109	.6667	50.0000	4000.0000	.33300000	.50000000	.16700000
5	1	.7109	.6667	50.0000	3000.0000	.33300000	.50000000	.16700000
6	-1	.7109	.6667	50.0000	1000.0000	.33300000	.50000000	.16700000

EQUIP	FLAG	PROPERTIES
10	-1	10000.0000
20	-1	2500.0000

$ $

$ $

$$$$$ PROCESS VARIABLES AT TIME = 0.00000 $$$$$

STREAM	TEMP	TOT.FLOW	CONCENTRATIONS		
3	50.0000	4000.0000	.33300000	.50000000	.16700000
4	50.0000	4000.0000	.33300000	.50000000	.16700000
5	50.0000	3000.0000	.33300000	.50000000	.16700000

EQUIP HOLD-UPS

$$$$$ PROCESS VARIABLES AT TIME = .82454 $$$$$

STREAM	TEMP	TOT.FLOW	CONCENTRATIONS		
3	50.0000	4000.0000	.33307117	.46441326	.20251557
4	50.0000	4000.0000	.33303286	.48357070	.18339644
5	50.0000	3000.0000	.33303286	.48357070	.18339644

EQUIP HOLD-UPS

$$$$$ PROCESS VARIABLES AT TIME = 2.68995 $$$$$

STREAM	TEMP	TOT.FLOW	CONCENTRATIONS		
3	50.0000	4000.0000	.33317817	.41091739	.25590444
4	50.0000	4000.0000	.33314610	.42694936	.23990454
5	50.0000	3000.0000	.33314610	.42694936	.23990454

EQUIP HOLD-UPS

$$$$$ PROCESS VARIABLES AT TIME = 7.97711 $$$$$

STREAM	TEMP	TOT.FLOW	CONCENTRATIONS		
3	50.0000	4000.0000	.33329782	.35108913	.31561305
4	50.0000	4000.0000	.33329032	.35484229	.31186739
5	50.0000	3000.0000	.33329032	.35484229	.31186739

EQUIP HOLD-UPS

$$$$$ PROCESS VARIABLES AT TIME = 10.08175 $$$$$

STREAM	TEMP	TOT.FLOW	CONCENTRATIONS		
3	50.0000	4000.0000	.33331812	.34094104	.32574085
4	50.0000	4000.0000	.33331489	.34255425	.32413086
5	50.0000	3000.0000	.33331489	.34255425	.32413086

Example 2

The equipment subroutines used in this example are:

Subroutine STIRØI - (see Example I)

Subroutine VALVØI -

A control valve with a parabolic characteristic.

VALVØI uses EPI( ).

Equipment type number is 4

On an information flow diagram, VALVØI is represented by:

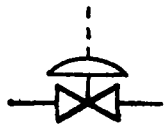


Figure 2-8 Information Flow Diagram Representation of Subroutine VALVØI

Subroutine CNTRLØI -

A three-mode controller

Equipment type number is 3

CNTRLØI uses BV2( ), EP2( )

The information flow diagram representation is:

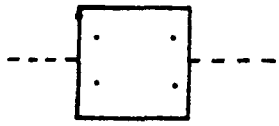


Figure 2-9 Information Flow Diagram Representation of CNTRLØI

Subroutine LAGØ2 -

Time lag element with by-pass.

Equipment type number is 2

LAGØ2 uses EPI( )

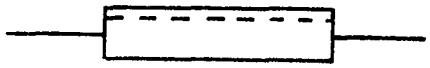


Figure 2-10 Information Flow Diagram Representation of LAGØ2

Subroutine MXPLTØ2 - See Example I.

The system to be simulated is sketched below:

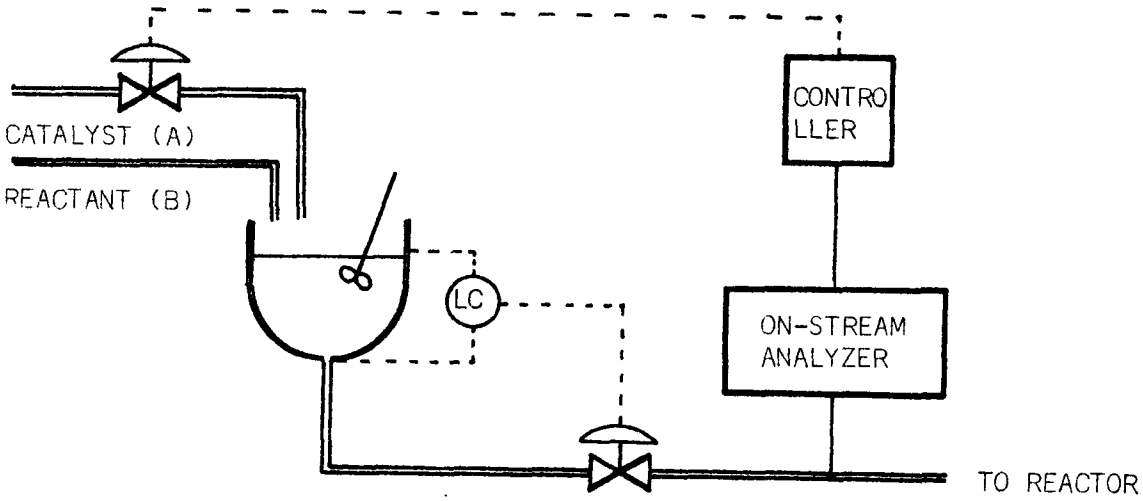


Figure 2-11 System for Example 2

The reactant (B) is mixed with a catalyst (A) in a well-stirred tank before being charged to a reactor further downstream. Catalyst concentration is measured by an on-stream analyzer and is automatically controlled. The stirred tank hold-up is measured by a differential-pressure cell and is automatically controlled. Catalyst concentration analysis is continuous, but there is a time lag between measurement and transmitting the concentration information to the controller. We assume that measurement of hold-up is instantaneous. The information flow diagram is

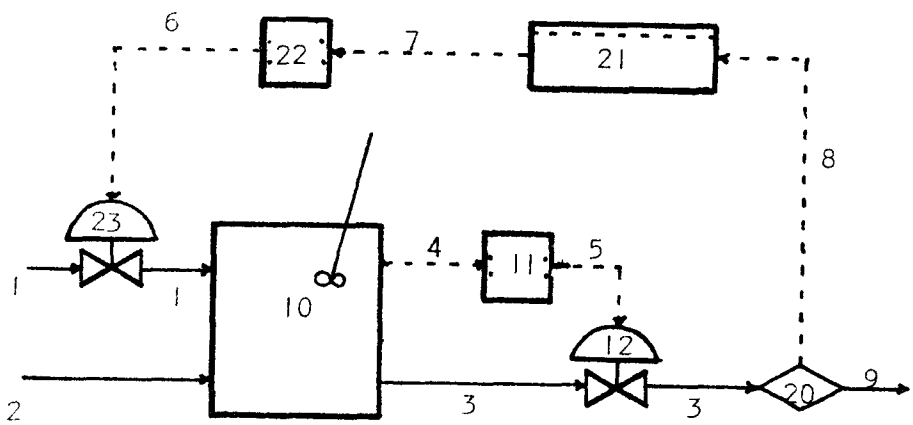


Figure 2-12(a) Information Flow Diagram
For Example 2

Comments on Figure 2-12(a):

Controller and valve signals are assigned stream numbers. These are regarded as "information flows". However, in order that information flows not be confused with material flows when carrying out mass and energy balances (e.g. around the stirred-tank) they are assigned flag numbers whose absolute value is greater than 10. Thus stream 4 could be flagged +11 or -11 depending on whether printout is desired. Information flows are represented as broken lines on the information flow diagram.

The MXPLTØ2 model (20) has no physical significance. In the EPI( ) matrix, its parameters (the splitting fractions) would be unity for each output stream. This is simply a method of directing information flow from one point to another. In fact, we may simplify the information flow diagram by further reducing its similarity to the physical system as shown in Figure 2-12(b).

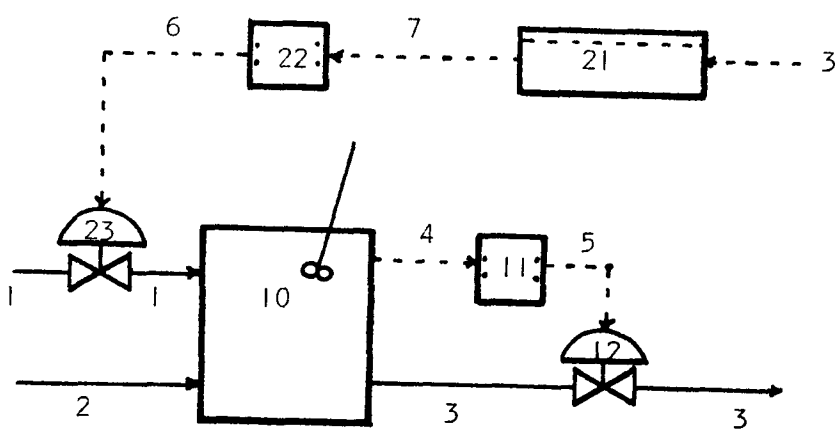


Figure 2-12(b) Improved Information Flow Diagram for Example 2

In Figure 3-12(b), the information stored in row 3 of the S( ) matrix is transmitted directly to the time lag element 21 . Clearly, we have reduced the amount of computation required (by reducing the number of equipments and streams). With this scheme, there is an inconsistency concerning the flag number to be assigned to stream 3 . However, examination of subroutine LAGØ2 reveals that no computational error would be introduced if stream 3 is flagged as a material flow.

Note that material flow through VALVØ1 is the output information from VALVØ1. Hence, the same stream number is assigned to the material flow to and from the valve.

The data deck for this system, including step changes in stream 2 (Flow: 4900 → 6000; TEMP. 20 → 60) is shown below.

McMASTER UNIVERSITY LIBRARY

FIELD	5	10	15	20	25	30	35	40	45	50	50
	6	6	7	2	10	4	1	3	0	5	8
	.001		10.	0.000005		0.00001					
MP( )	10	8	1	2	-3	-4					
	11	3	4	-5	0	0					
	12	4	5	-3	0	0					
	21	2	3	-7	0	0					
	22	3	7	-6	0	0					
	23	4	6	-1	0	0					
PP( )	0.3		+0.0006	1.5	1.5		0.0				
	0.8		-0.0044		0.9		+0.001				
S( )	1.	1.		0.		0.		20.			
	100.	1.		0.							
	2.	-1.		0.		0.		60.			
	6000.	0.		1.							
	3.	1.		0.		0.		20.			
	5000.	0.02		0.98							
	4.	-11.		0.		0.		0.			
	1000.	0.		0.							
	5.	-11.		0.		0.		0.			
	81.5	0.		0.							
	6.	-11.		0.		0.		0.			
	29.3	0.		0.							
	7.	+11.		0.		0.		20.			
	5000.	.02		.98							
EC1( )	10.	1.		1000.		0.		0.			
	10.	0.		0.		0.		0.			
EP1( )	12.	.75		+1.		0.		0.			
	23.	.02		-1.		0.		0.			
	21.	.1		1.		14.		0.			
EP2( )	11.	6.		0.		2000.		1000.			
	3.	5.0		0.001							
	22.	7.		0.		0.1		0.02			
	4.	7.0		0.005							

Comments: The stream lists for information (controller signal) streams are filled largely with zeroes - we need only enter those values which will be required in the calculations. Note that signals to valves are located in column 6 (normally used for flow rate). We have assumed that valves 12 and 23 have maximum through-puts of 10000 lbs/hr and 200 lbs/hr respectively. The on-line analyzer measures in the range 0 - 0.1.

To obtain a plotted output, we set

IOUTPUT = -1

and add the PLOTS(I,J) and PLOTEI(I,J) tables to the input data list. (The print interval number may also be reduced). The variables to be plotted are:

Stream 1 : Flow

Stream 3 : Temperature, Flow, Catalyst (A) Concentration

Equipment 10: Level

Therefore, we have 4 stream variables and 1 equipment variable. We wish 1 line on the output paper to represent 0.05 hours. The additional data (including ranges for the variables) are shown below:

```
 4        1
0.05
 1.          6.      50.       100.
 3.          6.       0.     10000.
 3.          5.       0.       100.
 3.          7.       0.015      0.025
10.          3.     500.      1500.
```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ D A T A I N P U T $$

DYNSYS-B Output for Example #2

PROCESS MATRIX

EQ.NO.	TYPE	IN STREAMS		OUT STREAMS	
10	8	1	2	-3	-4
11	3	4	-5	0	0
12	4	5	-3	0	0
21	2	3	-7	0	0
22	3	7	-6	0	0
23	4	6	-1	0	0

PHYSICAL PROPERTIES

COMPONENT	CONSTANTS			
1	3.000E-01	6.000E-04	1.500E+00	0.
2	8.000E-01	-4.000E-04	9.000E-01	1.000E-03

EQUIPMENT PARAMETERS

- 10-	0.0000	0.0000	0.0000	0.0000			
- 12-	.7500	1.0000	0.0000	0.0000			
- 23-	.0200	-1.0000	0.0000	0.0000			
- 21-	.1000	1.0000	14.0000	0.0000			
- 11-	6.0000	0.0000	2000.0000	1000.0000	3.0000	5.0000	.0010
- 22-	7.0000	0.0000	.1000	.0200	4.0000	7.0000	.0050

SYSTEM CONSTANTS

6	-NE=NO.EQUIPMENTS	6	-NMP=NO.COL.S IN PROC.MATRIX
2	-NCOMP=NO.COMPONENTS	7	-NO.STREAMS
1	-NPRINT=PRINT INTERVAL	.00100	-DELT=INIT.TIME STEP
.0000100	-TOLU=UPPER ERROR TOL	.0000050	-TOLL=LOWER ERROR TOL
1	-NEC1=NO.EQUIPS IN EC1	0	-NEC2=NO.EQUIPS IN EC2
1	-NBV1=NO.EQUIPS IN BV1	2	-NBV2=NO.EQUIPS IN BV2
4	-NEP1=NO.EQUIPS IN EP1	2	-NEP2=NO.EQUIPS IN EP2

INITIAL CONDITIONS

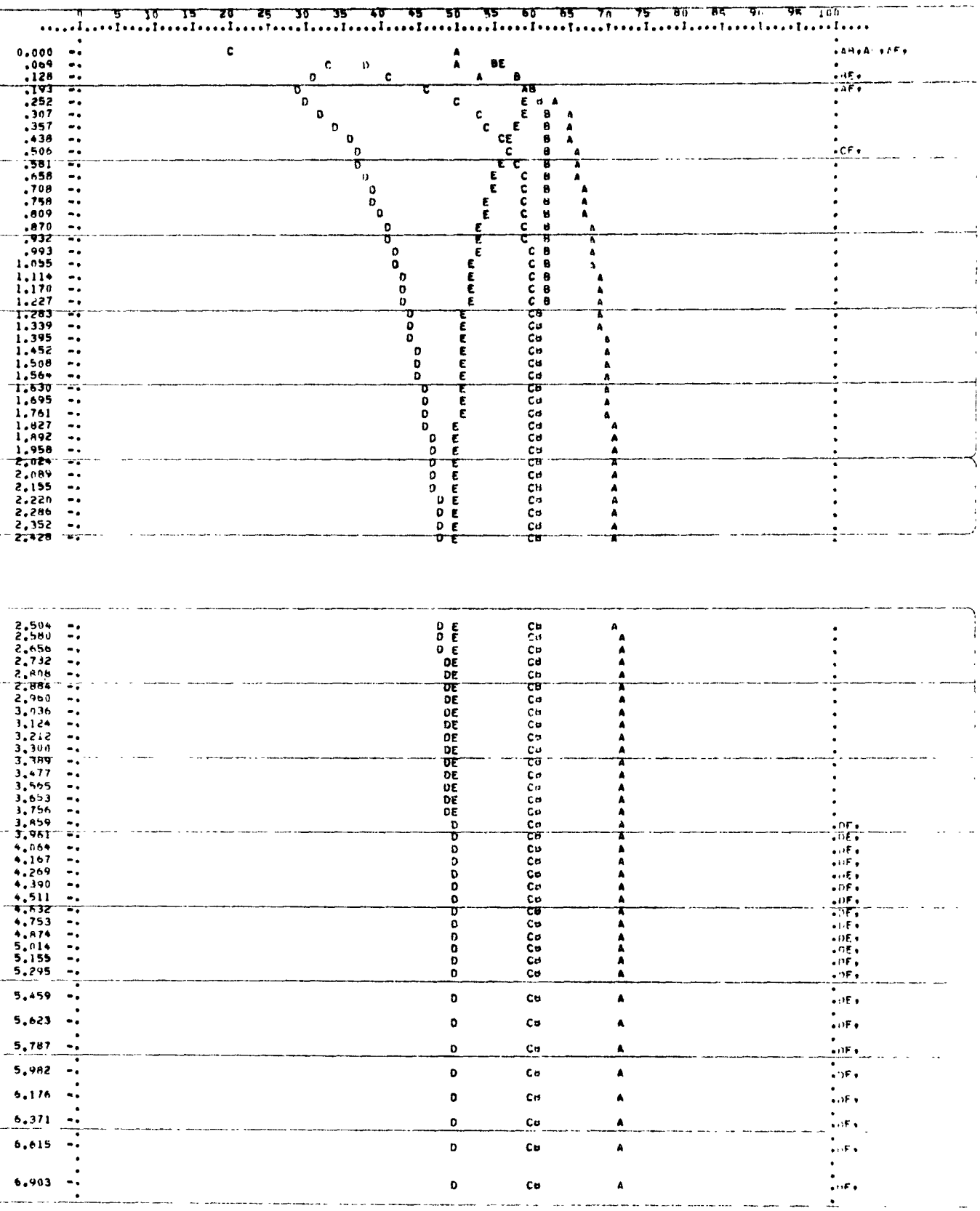
STREAM	FLAG	CP	SG	TEMP	TOT.FLOW	CONCENTRATIONS	
1	1	.3120	1.5000	20.0000	100.0000	1.00000000	0.00000000
2	-1	.7760	.9600	60.0000	6000.0000	0.00000000	1.00000000
3	1	.7824	.9316	20.0000	5000.0000	.02000000	.98000000
4	-11	0.0000	0.0000	0.0000	1000.0000	0.00000000	0.00000000
5	-11	0.0000	0.0000	0.0000	81.5000	0.00000000	0.00000000
6	-11	0.0000	0.0000	0.0000	29.3000	0.00000000	0.00000000
7	11	.7824	.9316	20.0000	5000.0000	.02000000	.98000000

EQUIP	FLAG	PROPERTIES
10	1	1000.0000

$ $

$$$$$ L E G E N D $$$$$

SYMBOL	STREAM	EQUIP.	VARIABLE	RANGE	
A	1		TOTFLOW	50.00000	150.00000
B	3		TOTFLOW	0.00000	10000.00000
C	3		TEMP	0.00000	100.00000
D	3		CONCA	.01500	.02500
E		10	MASS	500.00000	1500.00000



3. APPLICATION OF DYNSYS-B TO A LIQUID-LIQUID EXTRACTION UNIT WITH HEAT EFFECTS

3.1 Aim of the Simulation

The aim of this study was to test the applicability of the DYNSYS-B approach to a full-scale, real non-linear system. It was considered desirable to gain some experience as to the ease of setting up the simulation, the computer storage and computation time required.

With this in mind, we were concerned mainly with the approach to the problem and the principles involved, and did not deem it profitable to be overly concerned with obtaining exact results. For example, rather than carrying out laboratory experiments to measure equilibrium constants, we used available plant data to estimate these numbers, endeavouring, of course, to make the estimates as realistic as possible (see Section 3-5). Where applicable, one or more methods are indicated by which more exact numbers could be obtained by carrying out plant experiments

3.2 Description of the System

3.2.1 Chemical

The purpose of the unit is to extract 1-3 Butadiene from a mixture of hydrocarbons containing

1-3 Butadiene	∿	30% (wt.)
cis, trans Butylenes	∿	30% (wt.)
iso, normal Butenes	∿	26% (wt.)
iso, normal Butanes	∿	14% (wt.)

The hydrocarbon feed stock also contains small amounts of 1-2 Butadiene, propane, propene and traces of acetylenes and carbonyls.

The solvent used is formed by mixing together water, acetic-acid, copper and ammonia. The resulting aqueous solution contains the cupric ion (∿ 0.3 moles/litre), the cuprous ion (∿ 3 moles per litre) and the ammonium acetate ion (∿ 4.5 moles acetate per litre). There is an excess of ammonia, with total ammonia content being about 10 moles/litre.

The solution is highly selective to 1-3 butadiene, with which an exothermic reaction takes place to form a complex. This reaction is reversible, and the equilibrium coefficient is strongly dependent on temperature. At low temperatures (∿ 10°F), formation of salt is favoured, while at higher temperatures (90°F) the hydrocarbon form predominates.

3.2.2 Physical

The extraction unit is essentially a series of mixer/settler stages with the hydrocarbon phase flowing counter-current to the solvent phase. A typical stage is shown in Figure 3-1.

The hydrocarbon flow (about 10-40 Mlbs/hr) from each stage is controlled by the hydrocarbon phase level in the settler. Extraction of the 1-3 butadiene to the solvent phase takes place primarily in the mixer units, whose volumes are about 250 ft^3 each. The settlers are equipped with horizontal baffles to facilitate separation of the two phases. Settler volume is about 2000 ft^3, but varies from stage to stage.

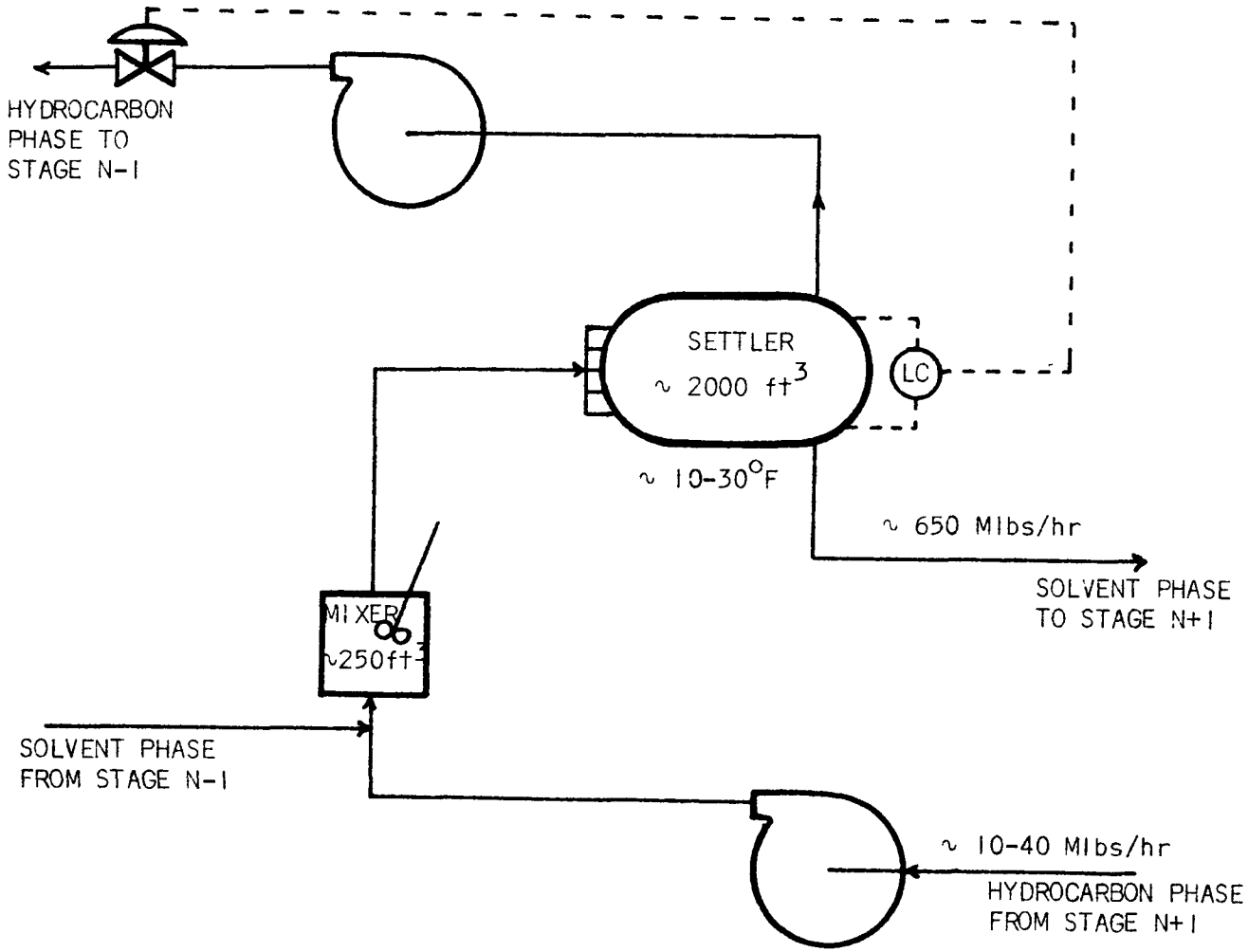


FIGURE 3-1 A TYPICAL MIXER/SETTLER STAGE

A somewhat simplified schematic diagram of the extraction unit is shown in Figure 3-2. The hydrocarbon feed enters stage 5 (actually, several alternate feed points are available). The series of stages to the left of the feed point comprise the "stripping" section, where the bulk of the extraction of 1-3 butadiene takes place. Stages 4, 6, 7 comprise the rectification section, where part of the hydrocarbon is refluxed to improve product purity. The rich solvent leaving stage 7 (the hydrocarbon content at this point is mainly 1-3 butadiene) is passed through a heater. At the higher temperature, the reverse reaction predominates, forming a hydrocarbon phase which is refluxed. Part of this reflux hydrocarbon is first passed through the Recycle Rerun Unit for purification (mainly removal of high molecular weight by products such as polymers) before entering stage 7. The hydrocarbon content of the solvent phase leaving stage 8 is about half of that leaving stage 7. The rich solvent enters a desorber, where hydrocarbon and solvent phases are separated. The hydrocarbon product is purified in a Wash Tower (for removal of 1-2 Butadiene, cis and trans Butylenes). Other purification steps such as removal of acetylenes, ammonia etc. are not included in the diagram. The solvent leaving the Desorber Tower is cooled and recycled to the first stage (stage 0).

3.2.3 Operational

In this section, a short description of the operational aspects and control philosophy of the plant is given, with the intention of providing some insight into the behaviour of the process.

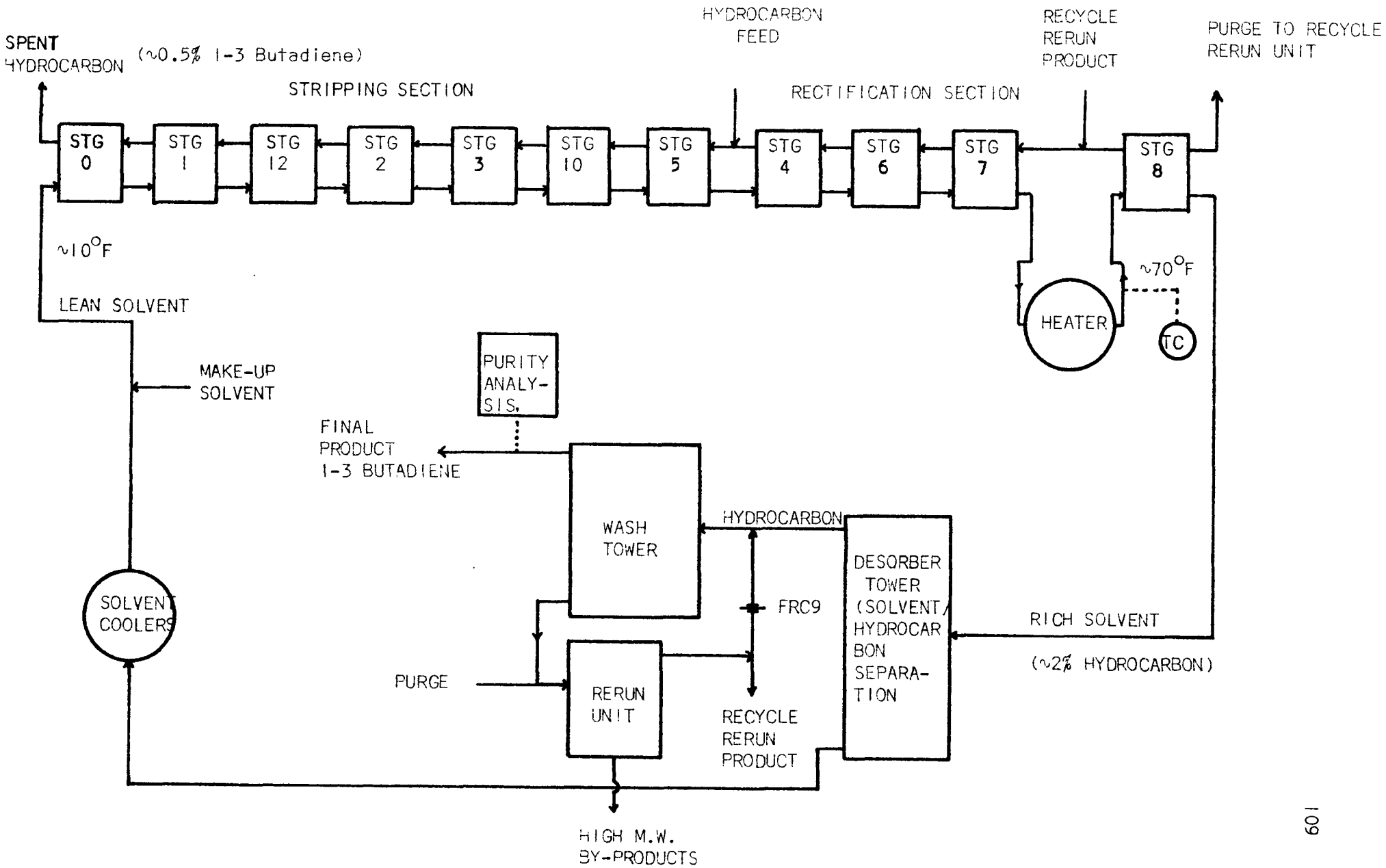


FIGURE 3-2 EXTRACTION UNIT AND PRODUCT PURIFICATION

The lean solvent enters stage 0 at about 10°F, but heats up in the stripping section to about 30°F, as a result of the exothermic reaction with 1-3 butadiene. The transfer rate of 1-3 butadiene to the solvent phase is greater at the cooler (left-hand) end of the stripping section due to the lower process temperature. To increase the transfer rate, the operator has the facility of solvent phase inter-stage coolers (not shown in Figure 3-2) but these are seldom used in practice (see Section 3-6). It is worth noting that the bulk of the heat flow through the extraction train is carried by the solvent phase, both by reason of the high ratio of solvent flow to hydrocarbon flow (∿ 35:1) and the higher specific heat of the solvent phase.

The spent hydrocarbon contains about 0.6% 1-3 butadiene. The concentration of 1-3 butadiene in the solvent phase increases rapidly in the low temperature zone, rising by a factor of about 15 between stages 0 and 3. Between stages 3 and 7, the concentration increases by a factor of 2, largely due to extraction of 1-3 butadiene reflux in the Rectification Section. There is a very large solvent phase flow, such that the total hydrocarbon content of the solvent phase never exceeds about 4%.

Control of the process is based on product purity analysis leaving the Wash Tower, and related adjustment of the reflux hydrocarbon flow. In practice, the operator has no control over the Rerun Unit, which is physically located elsewhere in the plant. He must accept a given make flow from the Rerun Unit, and decide what fraction of the total will pass through the flow control valve FRC-9 to the

Wash Tower, with the remainder being returned as reflux to stage 7. Thus, to improve 1-3 Butadiene product purity, the operator would reduce the flow through FRC-9, thereby increasing the flow of Recycle Rerun Unit product to be refluxed to stage 7.

3.3 Scope of the Simulation

We have limited this study to a relatively detailed simulation of the extraction train alone. Therefore, the Desorber and Wash Towers and the Rerun Unit are condensed into a short subroutine (CONTROL) whose purpose is to estimate the final 1-3 butadiene product purity by performing simple (steady-state) mass balances around these units. As mentioned previousely, this purity measurement is the main guideline for the operator in controlling plant operation. (hence the name subroutine CONTROL).

Within the extraction unit, we have chosen to ignore problems such as entrainment related to separation of the solvent and hydrocarbon phases. However, it is worth noting that to some extent these factors are implicitly taken into account since estimation of mass transfer coefficients etc. is based on real plant measurements.

A total of eight components are included in the study. They are (in the order used in the study):

A pure solvent (i.e. water, + Ammonia + acetic-acid + copper)
B 1-3 butadiene
C cis butylene
D trans butylene
E iso butene
F normal butene
G normal butane
H 1-2 butadiene

It is further assumed that the lean solvent feed to stage 0 is known and constant.

3.4 Construction of the Information Flow Diagram

Each mixer-settler stage was broken down into the following operations:

a) A perfectly mixed stirred tank, wherein the components of the hydrocarbon feed passed from one phase to the other (in either direction according to the temperature dependent mass transfer or reaction rates). Heat production is included. (subroutine XTRACØ1)

b) The horizontally baffled settler was modelled as two time lags, one for each phase. To allow for backmixing, a certain specified fraction of the flow to each time lag element was allowed to bypass the element. (subroutine LAGØ2). To incorporate the fluctuation of hydrocarbon phase level in the settler, a "tank" was conceived such that the intensive properties (i.e. temperature, concentrations etc.) of the feed, "tank" contents, and "tank" effluent were identical (within each of the two mutually insoluble phases) - only the total quantity of each phase (i.e. hydrocarbon and solvent phase levels) could vary. (subroutine SETLØ1)

c) Information on hydrocarbon phase level in "tank" SETLØ1 is passed to a controller (subroutine CNTRLØ1) and thence to a control valve (subroutine VALVØ1) which controls the hydrocarbon phase flow from SETLØ1.

d) To preserve overall material balances on each stage, the solvent phase flow from SETLØI was adjusted in subroutine SETFLØ. In practice, of course, this occur due to the constant total volume of each mixer - settler stage.

Note that the effect on flow of line pressure drop and pump delivery pressure fluctuation has been simplified and incorporated into subroutine VALVØI such that total flow is a simple (parabolic) function of valve opening. We believe the latter assumption to be a good one, since in practice changes in flow are small. The information flow diagram for a mixer/settler stage is shown in Figure 3-3.

The solvent heater between stages 7 and 8 is temperature controlled, and has a capacity very much smaller than that of a typical settler.

The influence of the heater on the dynamic behaviour of the process would be relatively unimportant, and it was concluded that this heater could be adequately represented by subroutine SETTØI.

As mentioned previousely (Section 3-3), the Wash and Desorber Towers and the Rerun unit are condensed into subroutine CONTROL.

The complete information flow diagram is shown in Figure 3-4.

To conclude section 3-4, listings of subroutines SETFLO and CONTROL are given. (These routines do not form part of the DYNSYS-B library, since they are specific to this particular process and will have no general use).

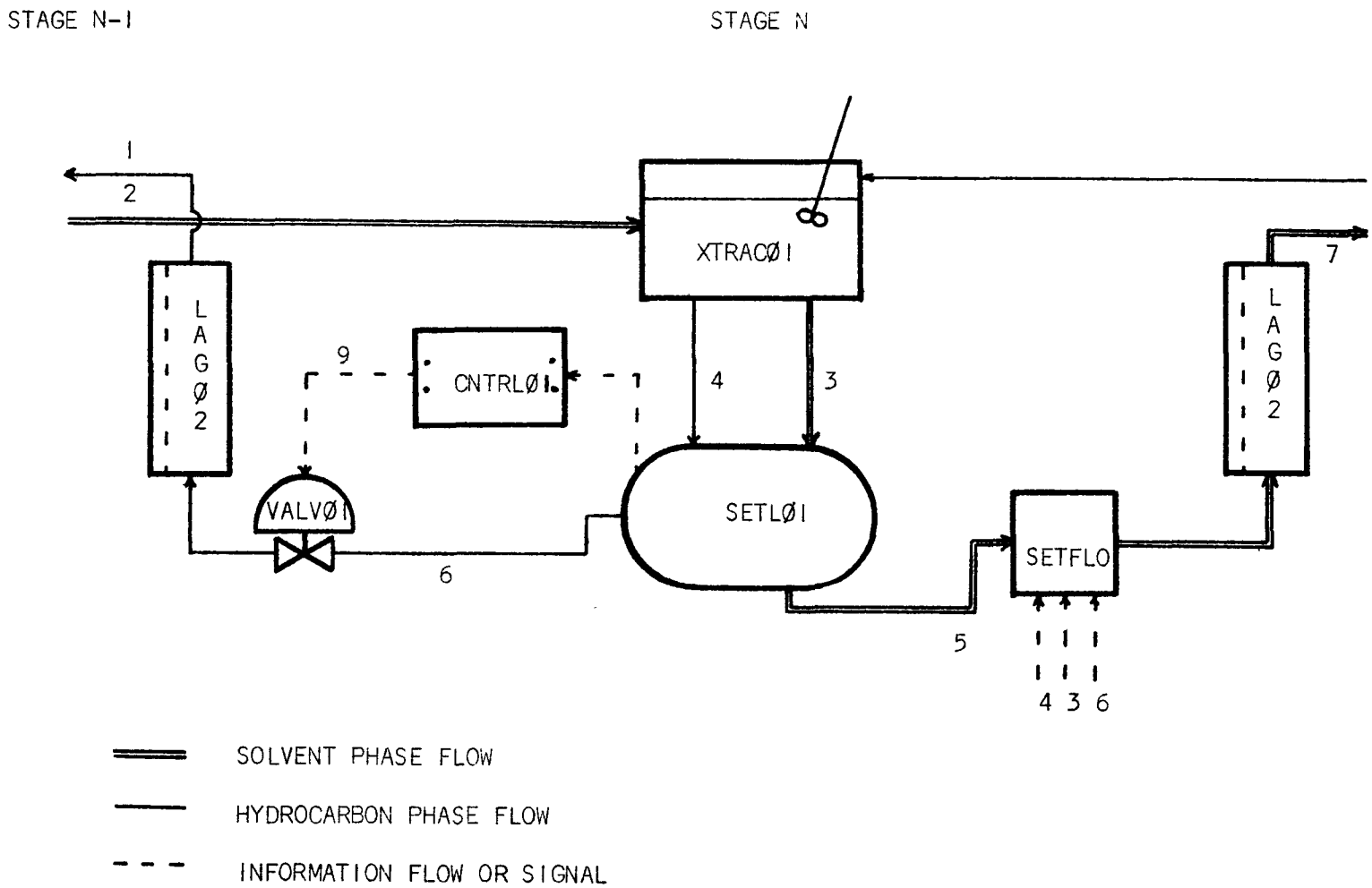


FIGURE 3-3 INFORMATION FLOW DIAGRAM FOR A MIXER/SETTLER STAGE

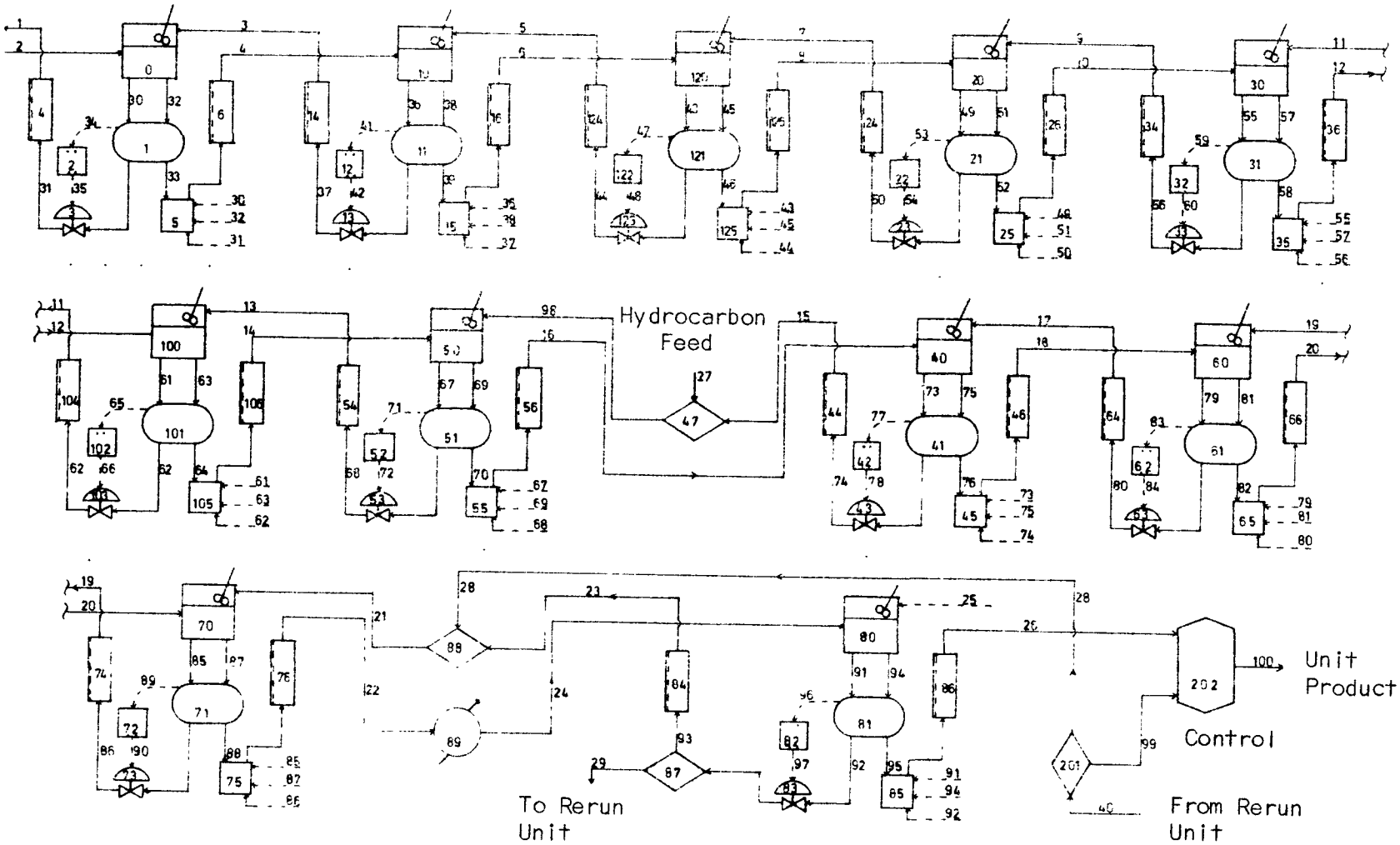


FIGURE 3-4 INFORMATION FLOW DIAGRAM FOR EXTRACTION PROCESS

```
      SUBROUTINE SETFLO (IM)
C     TYPE -9-
C     THIS  ROUTINE ADJUSTS OUTFLOW TO MAINTAIN THE MASS BALANCE
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      MHI=MP(IM,3)
      MSI=MP(IM,4)
      MHO=MP(IM,5)
      MSO=-MP(IM,6)
      S(1,MSO,6)=S(1,MSI,6)+(S(1,MHI,6)-S(1,MHO,6))*S(1,MSO,4)/S(1,MHO,
     1   4)
      RETURN
      END
```

```
      SUBROUTINE CONTROL (IM)
C     TYPE -10-
C     THIS ROUTINE PERFORMS A SIMPLIFIED SIMULATION OF THE CONTROL
C     PHILOSOPHY OF THE EXTRACTION UNIT
C     THE STEPS ARE -    (1) SEPARATE HYDROCARBONS FROM SOLVENT LEAVING
C                                                        DS8      (DESORBER)
C                        (2) MIX WITH STREAM FRC9
C                        (3) REMOVE BD1-3,CIS,TRANS.  (RERUN TOWER)
C     NO HEAT BALANCES ARE PERFORMED
C     LISTING IN PROCESS MATRIX-1ST IN STREAM=SOLVENT PHASE FROM DS8
C                              2ND IN STREAM=FRC9
      COMMON/BLK1/NMP,IG,NCOMP,TIME(30),MP(85,8),S(2,100,14),PP(10,10),
     1   LOOK1(85,4)
      NC6=NCOMP+6
      MSI=MP(IM,3)
      MHI=MP(IM,4)
      MHO=-MP(IM,5)
      SUM=0.
      DO 50 K=8,NC6
      S(1,MHO,K)=S(1,MSI,K)*S(1,MSI,6)+S(1,MHI,K)*S(1,MHI,6)
      IF(K.EQ.9.OR.K.EQ.10.OR.K.EQ.14)SUM=SUM+S(1,MHO,K)
   50 CONTINUE
      HCF=(1.0-S(1,MSI,7))*S(1,MSI,6)+S(1,MHI,6)-SUM
      DO 75 K=7,NC6
      S(1,MHO,K)=S(1,MHO,K)/HCF
   75 CONTINUE
      S(1,MHO,6)=HCF
      S(1,MHO,9)=0.
      S(1,MHO,10)=0.
      S(1,MHO,14)=0.
      RETURN
      END
```

3.5 Estimation of Physical Properties

In this section a description is given of the approaches used to estimate the physical properties of the materials in the system.

Available plant steady-state data are used to estimate mass transfer coefficients and heats of reaction. Data on specific heat and gravity are taken from tables in the literature [17, 18].

As described in subroutine XTRACØI (Type I in the DYNSYS-B library), it was decided to lump together the mass transfer and reaction rates of the hydrocarbons, such that the rate of transfer of component (I) from hydrocarbon to solvent phases is given by:

$$\text{Rate}(I) = k_1 * Y - k_2 * X \qquad (1)$$

where Rate (I) = lbs/hr of component (I) changing phases

Y = concentration of component (I) in hydrocarbon phase

X = concentration of component (I) in solvent phase (see below)

k_1, k_2 = forward and reverse transfer rate coefficients

Note that in fact (for the case of 1-3 butadiene at least) the hydrocarbon exists as a complex in the solvent phase, so that X really represents the concentration of hydrocarbon as this complex.

The rate coefficients, k_1 and k_2 are assumed to be Arrhenius type functions of temperature, i.e.

$$k_1 = \alpha_1 \exp(\alpha_2/T) \qquad (2)$$

$$k_2 = \alpha_2 \exp(\alpha_4/T)$$

where $\alpha(i)$ = constant

T = absolute temperature

Ideally, α_2 and α_4 are given by

$$\alpha_2 = \Delta H_F/R \qquad (3)$$

$$\alpha_4 = \Delta H_R/R$$

where ΔH_F = forward heat of "transfer" (reaction)

ΔH_R = reverse heat of "transfer" (reaction)

R = universal gas constant

For any stage at steady state, the mass balances for component (I) are:

Hydrocarbon Phase -

0 = Flow of (I) into stage in hydrocarbon phase

- Flow of (I) out of stage in hydrocarbon phase

- Rate (I)

$$= \Delta_{HC} - \text{Rate (I)} \qquad (4)$$

Similarly, for the solvent phase,

$$0 = \Delta_{SOL} + \text{Rate (I)}$$

Now, at equilibrium, Rate (I) = 0

$$k_1 Y - k_2 X^* = 0 \qquad (5)$$

where X^* is the concentration in equilibrium with Y. Consider that in fact, there is some "approach to equilibrium" p, such that

$$X = p * X^* \qquad (6)$$

then by combining (4), (5), (6), we have

$$k_1 = -\Delta_{HC}/Y(1-p) \qquad (7)$$

$$k_2 = p*k_1*Y/X \qquad (8)$$

It was considered reasonable to assume that the plant operated at about 80%-90% of equilibrium. In practice, this could be calculated by measuring equilibrium coefficients (the ratio k_2/k_1) or perhaps by taking steady state data of the plant at a very low throughput (i.e. long residence times). Since each stage operates at a different temperature, equations (7) and (8) together with an assumed value for p enabled calculation of k_1 and k_2 as a function of temperature (this assumes no dependence on concentration).

One point should be noted with regard to plant data. As previousely suggested, the residence times and lags in this process are quite large (about 15 hours are required for the system to approach closely to steady state). It may be surmised that it is therefore difficult in practice to collect good steady-state data, and in fact we found that there were some fairly large inconsistancies in the plant data. (In certain cases, mass balances did not close to better than 200%). As a result, the original plant data were first used as a set of initial conditions containing numerous step changes (corresponding to the inconsistencies) and allowed to run to steady-state (in the computer) to provide a set of consistent steady-state data. Some adjustments were made to the original estimates of k_1 and k_2 improve the matching of plant and computer steady state data.

Examination of the data revealed that the bulk of total hydrocarbon changing phases consists of the desired product 1-3 butadiene (more than 90%). We therefore decided to attribute all of the reaction heat produced to 1-3 butadiene. This heat of reaction was assumed constant, and an average value taken according to steady-state heat

balances. For the same reasons, and also since the original estimates of k_1 and k_2 as functions of temperature showed poor correlation (except for the case of 1-3 butadiene where the correlation was good), we set the values of α_2 and α_4 in equations(2) to zero for all components other than for 1-3 butadiene. There is no measurable inter-phase transfer of the solvent phase, and the corresponding rate constants were taken as zero. To facilitate future improvements to this simulation, the programming was done as if all component k_1 and k_2 values were functions of temperature, and heats of reactions are not equal to zero for all cases. Rather, appropriate zero's are entered into the data deck.

As mentioned previousely, the bulk of the heat flow through the process is carried in the solvent phase (about 99%). Further, the solvent phase contains no more than 4% hydrocarbon. Thus it was considered adequate to calculate the specific heats and gravities of the phases as concentration weighted averages of the specific heats and gravities of the pure components. For each case, these properties are taken to be linear functions of temperature. Component specific heats and gravities were taken from the literature [17, 18].

No plant information was available as to the extent of back-mixing in the settlers. We therefore assumed somewhat arbitrarily that 10% of the flow by-passed the lag element (subroutine LAGØ2). Real data on this point could be fairly readily obtained by carrying out tracer studies in the plant. Similarly, there is no information as to what percentage of the inter phase mass transfer takes place in

the mixers and what percentage takes place in the settlers. Again, we have arbitrarily assumed that 10% of the volume of the settlers is mixed (available for mass transfer). Thus 10% of the volume for subroutine SETLØI is assigned to subroutine XTRACØI. If our original assumption of an 80%-90% approach to equilibrium is correct, this reassignment of volumes will have little effect on the dynamic behaviour.

In the following pages the complete data deck used in the simulation is given.

```
  7   83  100    8   50    9   -1    5    0    5   10          123
.00001     15.       .000008    .00001
  0    1    3    2  -30  -32    0
  1    6   30   32  -31  -33  -34
  2    3   34  -35    0    0    0
  3    4   35  -31    0    0    0
  4    2   31   -1    0    0    0
  5    9   30   32   31  -33    0
  6    2   33   -4    0    0    0
 10    1    5    4  -36  -38    0
 11    6   36   38  -37  -39  -41
 12    3   41  -42    0    0    0
 13    4   42  -37    0    0    0
 14    2   37   -3    0    0    0
 15    9   36   38   37  -39    0
 16    2   39   -6    0    0    0
120    1    7    6  -43  -45    0
121    6   43   45  -44  -46  -47
122    3   47  -48    0    0    0
123    4   48  -44    0    0    0
124    2   44   -5    0    0    0
125    9   43   45   44  -46    0
126    2   46   -8    0    0    0
 20    1    9    8  -49  -51    0
 21    6   49   51  -50  -52  -53
 22    3   53  -54    0    0    0
 23    4   54  -50    0    0    0
 24    2   50   -7    0    0    0
 25    9   49   51   50  -52    0
 26    2   52  -10    0    0    0
 30    1   11   10  -55  -57    0
 31    6   55   57  -56  -58  -59
 32    3   59  -60    0    0    0
 33    4   60  -56    0    0    0
 34    2   56   -9    0    0    0
 35    9   55   57   56  -58    0
 36    2   58  -12    0    0    0
100    1   13   12  -61  -63    0
101    6   61   63  -62  -64  -65
102    3   65  -66    0    0    0
103    4   66  -62    0    0    0
104    2   62  -11    0    0    0
105    9   61   63   62  -64    0
106    2   64  -14    0    0    0
 40    1   17   16  -73  -75    0
 41    6   73   75  -74  -76  -77
 42    3   77  -78    0    0    0
 43    4   78  -74    0    0    0
 44    2   74  -15    0    0    0
 45    9   73   75   74  -76    0
```

46	2	76	-18	0	0	0
47	5	27	15	-98	0	0
50	1	98	14	-67	-69	0
51	6	67	69	-68	-70	-71
52	3	71	-72	0	0	0
53	4	72	-68	0	0	0
54	2	68	-13	0	0	0
55	9	67	69	68	-70	0
56	2	70	-16	0	0	0
60	1	19	18	-79	-81	0
61	6	79	81	-80	-82	-83
62	3	83	-84	0	0	0
63	4	84	-80	0	0	0
64	2	80	-17	0	0	0
65	9	79	81	80	-82	0
66	2	82	-20	0	0	0
80	1	25	24	-91	-94	0
81	6	91	94	-92	-95	-96
82	3	96	-97	0	0	0
83	4	97	-92	0	0	0
87	5	92	-93	-29	0	0
84	2	93	-23	0	0	0
201	5	40	-28	-99	0	0
88	5	23	28	-21	0	0
85	9	91	94	92	-95	0
86	2	95	-26	0	0	0
202	10	26	99	-100	0	0
89	7	22	-24	0	0	0
70	1	21	20	-85	-87	0
71	6	85	87	-86	-88	-89
72	3	89	-90	0	0	0
73	4	90	-86	0	0	0
74	2	86	-19	0	0	0
75	9	85	87	86	-88	0
76	2	88	-22	0	0	0

1.02	-.00205	75.	0.	0.
0.	0.	0.	0.	
.6	.000625	38.5	-.042	86000.
16.5	780000.	-3.4	660.	
.61	.000625	41.4	-0.0181	8000.
0.	2000000.	0.	0.	
.61	.000625	40.6	-0.0181	8000.
0.	2000000.	0.	0.	
.615	.000625	39.9	-0.0181	22000.
0.	2000000.	0.	0.	
.615	0.000625	39.9	-0.0181	15000.
0.	2000000.	0.	0.	
0.64	0.000605	38.7	0.035	5000.
0.	2000000.	0.	0.	
.63	0.000625	37.4	-0.035	50000.
0.	200000.	0.	0.	

1.00	1.00	.6242	39.9865	11.7469
38287.46	0.0000000	.0254456	.1892177	.2266214
.1494052	.2361228	.1728162	.0003711	
2.00	-1.00	.9995	74.8092	9.0000
650000.00	.9946400	.0016700	.0005800	.0004200
.0004200	.0021000	.0001000	.0000700	
3.00	1.00	.6263	39.8109	16.8102
41667.53	0.0000000	.0818541	.1759954	.2140148
.1517721	.2132083	.1627291	.0004260	
4.00	1.00	.9919	74.6212	11.7468
653380.17	.9894946	.0053905	.0007126	.0007862
.0013417	.0018493	.0003502	.0000751	
5.00	1.00	.6280	39.5371	21.9717
46357.16	0.0000000	.1721500	.1581551	.1931728
.1400235	.1891095	.1469003	.0004887	
6.00	-1.00	.9790	74.3576	16.8100
658069.94	.9824432	.0122967	.0007049	.0008374
.0015860	.0016578	.0003922	.0000820	
7.00	-1.00	.6288	39.2988	25.3619
50802.86	0.0000000	.2485037	.1436550	.1756913
.1274274	.1703254	.1338279	.0005692	
8.00	-1.00	.9663	74.1070	21.9717
662515.81	.9758507	.0192247	.0006494	.0007874
.0015490	.0014751	.0003728	.0000909	
9.00	-1.00	.6292	39.1585	27.1246
53574.91	0.0000000	.2921040	.1355510	.1658436
.1195224	.1597284	.1265623	.0006883	
10.00	1.00	.9581	73.9503	25.3619
665287.99	.9717848	.0236912	.0005926	.0007231
.0014369	.0013253	.0003437	.0001025	
11.00	1.00	.6293	39.0892	27.9478
54959.91	0.0000000	.3131814	.1316880	.1611354
.1154269	.1545707	.1231312	.0008664	
12.00	-1.00	.9540	73.8718	27.1246
666673.10	.9697661	.0259864	.0005545	.0006779
.0013445	.0012291	.0003231	.0001184	
13.00	-1.00	.6293	39.0585	28.2363
55598.32	0.0000000	.3227480	.1299312	.1589909
.1134398	.1521783	.1215833	.0011284	
14.00	-1.00	.9520	73.8354	27.9478
667311.60	.9688385	.0270582	.0005336	.0006527
.0012881	.0011765	.0003116	.0001409	
15.00	-1.00	.6269	38.2766	31.8699
5962.94	0.0000000	.5606535	.0615745	.0773769
.1152432	.1414664	.0429487	.0007367	
16.00	1.00	.9513	73.8149	28.2363
667676.30	.9683095	.0276403	.0005231	.0006401
.0012577	.0011500	.0003058	.0001734	
17.00	1.00	.6255	37.6728	34.6281
8773.18	0.0000000	.7507436	.0261387	.0338327
.0792275	.0913566	.0179102	.0007908	

18.00	-1.00	.9428	73.6549	31.8704
670486.78	.9642516	.0323618	.0003153	.0003918
.0012640	.0010822	.0001569	.0001764	
19.00	-1.00	.6248	37.3389	36.1294
10527.95	0.0000000	.8569888	.0117516	.0159836
.0474213	.0572900	.0095870	.0009777	
20.00	1.00	.9365	73.5528	34.6281
672241.51	.9617345	.0359011	.0001574	.0001996
.0009694	.0007843	.0000729	.0001809	
21.00	-1.00	.6292	36.8642	44.3975
11219.20	0.0000000	.9140732	.0060824	.0089192
.0244483	.0378791	.0068048	.0017930	
22.00	-1.00	.9333	73.5108	36.1294
672932.73	.9607465	.0376963	.0000747	.0000980
.0006341	.0005188	.0000363	.0001953	
23.00	-1.00	.6330	36.4589	51.6472
4211.20	0.0000000	.9541342	.0028913	.0037922
.0193699	.0177084	.0014875	.0006164	
24.00	-1.00	.8771	73.4642	65.0000
672933.03	.9607465	.0376963	.0000747	.0000980
.0006341	.0005188	.0000363	.0001953	
25.00	-1.00	0.0000	0.0000	0.0000
0.00	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
26.00	1.00	.9076	74.1058	51.6472
661850.53	.9768333	.0223514	.0000276	.0000362
.0003204	.0002309	.0000120	.0001882	
27.00	-1.00	.6278	39.1828	25.0000
50000.00	0.0000000	.3000000	.1370000	.1674000
.1120000	.1520000	.1300000	.0016000	
28.00	-1.00	.6268	37.1009	40.0000
7008.00	0.0000000	.8900000	.0080000	.0120000
.0275000	.0500000	.0100000	.0025000	
29.00	-1.00	.6330	36.4589	51.6472
6870.91	0.0000000	.9541345	.0028913	.0037922
.0193698	.0177084	.0014875	.0006163	
30.00	-1.00	.6242	39.9865	11.7468
38287.59	0.0000000	.0254439	.1892182	.2266220
.1494051	.2361230	.1728167	.0003711	
31.00	-1.00	.6242	39.9865	11.7468
38287.46	0.0000000	.0254439	.1892182	.2266220
.1494051	.2361230	.1728167	.0003711	
32.00	-1.00	.9919	74.6212	11.7468
653379.94	.9894947	.0053904	.0007126	.0007862
.0013417	.0018493	.0003502	.0000751	
33.00	-1.00	.9919	74.6212	11.7468
653380.17	.9894947	.0053904	.0007126	.0007862
.0013417	.0018493	.0003502	.0000751	
34.00	-1.00	0.0000	0.0000	0.0000
17599.85	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
35.00	-1.00	0.0000	0.0000	0.0000
69.18	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	

36.00	-1.00	.6263	39.8109	16.8099
41667.69	0.0000000	.0818495	.1759963	.2140160
.1517728	.2132094	.1627299	.0004260	
37.00	-1.00	.6263	39.8109	16.8099
41667.53	0.0000000	.0818495	.1759963	.2140160
.1517728	.2132094	.1627299	.0004260	
38.00	-1.00	.9790	74.3576	16.8099
658069.63	.9824433	.0122966	.0007049	.0008374
.0015860	.0016578	.0003922	.0000820	
39.00	-1.00	.9790	74.3576	16.8099
658069.94	.9824433	.0122966	.0007049	.0008374
.0015860	.0016578	.0003922	.0000820	
40.00	-1.00	.6268	37.1009	40.0000
12000.00	0.0000000	.8900000	.0080000	.0120000
.0275000	.0500000	.0100000	.0025000	
41.00	-1.00	0.0000	0.0000	0.0000
17599.80	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
42.00	-1.00	0.0000	0.0000	0.0000
72.17	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
43.00	-1.00	.6280	39.5371	21.9716
46357.34	0.0000000	.1721477	.1581557	.1931734
.1400237	.1891099	.1469008	.0004887	
44.00	-1.00	.6280	39.5371	21.9716
46357.16	0.0000000	.1721477	.1581557	.1931734
.1400237	.1891099	.1469008	.0004887	
45.00	-1.00	.9663	74.1070	21.9716
662515.46	.9758509	.0192245	.0006494	.0007874
.0015490	.0014751	.0003728	.0000909	
46.00	-1.00	.9663	74.1070	21.9716
662515.81	.9758509	.0192245	.0006494	.0007874
.0015490	.0014751	.0003728	.0000909	
47.00	-1.00	0.0000	0.0000	0.0000
7999.80	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
48.00	-1.00	0.0000	0.0000	0.0000
76.12	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
49.00	-1.00	.6288	39.2989	25.3619
50803.02	0.0000000	.2485029	.1436552	.1756915
.1274275	.1703256	.1338281	.0005691	
50.00	-1.00	.6288	39.2989	25.3619
50802.86	0.0000000	.2485029	.1436552	.1756915
.1274275	.1703256	.1338281	.0005691	
51.00	-1.00	.9581	73.9503	25.3619
665287.70	.9717849	.0236911	.0005926	.0007231
.0014369	.0013253	.0003437	.0001025	
52.00	-1.00	.9581	73.9503	25.3619
665287.99	.9717849	.0236911	.0005926	.0007231
.0014369	.0013253	.0003437	.0001025	
53.00	-1.00	0.0000	0.0000	0.0000
5599.87	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	

```
    54.00      -1.00     0.0000     0.0000     0.0000
    79.69 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    55.00      -1.00      .6292    39.1585    27.1246
 53575.03 0.0000000  .2921038  .1355510  .1658437
 .1195224  .1597284  .1265624  .0006883
    56.00      -1.00      .6292    39.1585    27.1246
 53574.91 0.0000000  .2921038  .1355510  .1658437
 .1195224  .1597284  .1265624  .0006883
    57.00      -1.00      .9540    73.8718    27.1246
666672.87  .9697661  .0259864  .0005545  .0006779
 .0013445  .0012291  .0003231  .0001184
    58.00      -1.00      .9540    73.8718    27.1246
666673.10  .9697661  .0259864  .0005545  .0006779
 .0013445  .0012291  .0003231  .0001184
    59.00      -1.00     0.0000     0.0000     0.0000
  5599.94 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    60.00      -1.00     0.0000     0.0000     0.0000
    81.83 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    61.00      -1.00      .6293    39.0892    27.9478
 54960.01 0.0000000  .3131816  .1316880  .1611354
 .1154268  .1545706  .1231312  .0008663
    62.00      -1.00      .6293    39.0892    27.9478
 54959.91 0.0000000  .3131816  .1316880  .1611354
 .1154268  .1545706  .1231312  .0008663
    63.00      -1.00      .9520    73.8354    27.9478
667311.41  .9688385  .0270582  .0005336  .0006527
 .0012881  .0011765  .0003116  .0001409
    64.00      -1.00      .9520    73.8354    27.9478
667311.60  .9688385  .0270582  .0005336  .0006527
 .0012881  .0011765  .0003116  .0001409
    65.00      -1.00     0.0000     0.0000     0.0000
  7999.99 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    66.00      -1.00     0.0000     0.0000     0.0000
    82.89 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    67.00      -1.00      .6293    39.0585    28.2363
 55598.40 0.0000000  .3227492  .1299311  .1589907
 .1134395  .1521782  .1215832  .0011281
    68.00      -1.00      .6293    39.0585    28.2363
 55598.32 0.0000000  .3227492  .1299311  .1589907
 .1134395  .1521782  .1215832  .0011281
    69.00      -1.00      .9513    73.8149    28.2363
667676.14  .9683095  .0276403  .0005231  .0006401
 .0012577  .0011500  .0003058  .0001734
    70.00      -1.00      .9513    73.8149    28.2363
667676.30  .9683095  .0276403  .0005231  .0006401
 .0012577  .0011500  .0003058  .0001734
    71.00      -1.00     0.0000     0.0000     0.0000
 20000.01 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
```

```
    72.00      -1.00     0.0000     0.0000     0.0000
    83.37 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    73.00      -1.00      .6269    38.2767    31.8707
  5963.21 0.0000000  .5605748  .0615799  .0773837
 .1152666  .1414841  .0429520  .0007588
    74.00      -1.00      .6269    38.2767    31.8707
  5962.94 0.0000000  .5605748  .0615799  .0773837
 .1152666  .1414841  .0429520  .0007588
    75.00      -1.00      .9428    73.6549    31.8707
670486.27  .9642514  .0323621  .0003153  .0003918
 .0012640  .0010822  .0001569  .0001763
    76.00      -1.00      .9428    73.6549    31.8707
670486.78  .9642514  .0323621  .0003153  .0003918
 .0012640  .0010822  .0001569  .0001763
    77.00      -1.00     0.0000     0.0000     0.0000
  5200.13 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    78.00      -1.00     0.0000     0.0000     0.0000
    48.84 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    79.00      -1.00      .6255    37.6728    34.6281
  8773.13 0.0000000  .7507430  .0261388  .0338329
 .0792276  .0913567  .0179102  .0007907
    80.00      -1.00      .6255    37.6728    34.6281
  8773.18 0.0000000  .7507430  .0261388  .0338329
 .0792276  .0913567  .0179102  .0007907
    81.00      -1.00      .9365    73.5528    34.6281
672241.61  .9617345  .0359010  .0001574  .0001996
 .0009694  .0007843  .0000729  .0001809
    82.00      -1.00      .9365    73.5528    34.6281
672241.51  .9617345  .0359010  .0001574  .0001996
 .0009694  .0007843  .0000729  .0001809
    83.00      -1.00     0.0000     0.0000     0.0000
  5200.06 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    84.00      -1.00     0.0000     0.0000     0.0000
    59.24 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
    85.00      -1.00      .6248    37.3389    36.1294
 10527.93 0.0000000  .8569886  .0117516  .0159836
 .0474214  .0572901  .0095870  .0009776
    86.00      -1.00      .6248    37.3389    36.1294
 10527.95 0.0000000  .8569886  .0117516  .0159836
 .0474214  .0572901  .0095870  .0009776
    87.00      -1.00      .9333    73.5108    36.1294
672932.78  .9607465  .0376963  .0000747  .0000980
 .0006341  .0005188  .0000363  .0001953
    88.00      -1.00      .9333    73.5108    36.1294
672932.73  .9607465  .0376963  .0000747  .0000980
 .0006341  .0005188  .0000363  .0001953
    89.00      -1.00     0.0000     0.0000     0.0000
  5200.02 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000
```

90.00	-1.00	0.0000	0.0000	0.0000
64.89	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
91.00	-1.00	.6330	36.4589	51.6472
11082.08	0.0000000	.9541345	.0028913	.0037922
.0193698	.0177084	.0014875	.0006163	
92.00	1.00	.6330	36.4589	51.6472
11082.11	0.0000000	.9541345	.0028913	.0037922
.0193698	.0177084	.0014875	.0006163	
93.00	-1.00	.6330	36.4589	51.6472
4211.20	0.0000000	.9541345	.0028913	.0037922
.0193698	.0177084	.0014875	.0006163	
94.00	-1.00	.9076	74.1058	51.6472
661850.60	.9768334	.0223514	.0000276	.0000362
.0003204	.0002309	.0000120	.0001882	
95.00	-1.00	.9076	74.1058	51.6472
661850.53	.9768334	.0223514	.0000276	.0000362
.0003204	.0002309	.0000120	.0001882	
96.00	-1.00	0.0000	0.0000	0.0000
14000.02	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
97.00	-1.00	0.0000	0.0000	0.0000
66.58	0.0000000	0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000	0.0000000	
98.00	-1.00	.6277	39.0934	25.7311
55962.94	0.0000000	.3277731	.1289633	.1578079
.1123456	.1508776	.1207245	.0015080	
99.00	-1.00	.6268	37.1009	40.0000
4992.00	0.0000000	.8900000	.0080000	.0120000
.0275000	.0500000	.0100000	.0025000	
100.00	1.00	.6007	38.5531	0.0000
20045.76	0.0000000	.9596116	0.0000000	0.0000000
.0174269	.0200763	.0028852	0.0000000	
0.000	-1.000	1660.664	28339.336	0.000
10.000	-1.000	1786.429	28213.571	0.000
120.000	-1.000	2877.418	41122.582	0.000
20.000	-1.000	2128.349	27871.651	0.000
30.000	-1.000	2231.524	27768.476	0.000
100.000	-1.000	3348.105	40651.895	0.000
50.000	-1.000	2921.075	35078.925	0.000
40.000	-1.000	264.466	29735.534	0.000
60.000	-1.000	386.473	29613.527	0.000
70.000	-1.000	462.116	29537.884	0.000

80.000	-1.000	658.733	39341.267	0.000
1.000	-1.000	17599.848	74610.380	0.000
11.000	-1.000	17599.802	74202.493	0.000
121.000	-1.000	7999.796	155451.550	0.000
21.000	-1.000	5599.872	95950.937	0.000
31.000	-1.000	5599.943	95811.150	0.000
101.000	-1.000	7999.985	154710.381	0.000
51.000	-1.000	20000.010	121643.107	0.000
41.000	-1.000	5200.130	96056.625	0.000
61.000	-1.000	5200.062	95763.346	0.000
71.000	-1.000	5200.020	95618.054	0.000
81.000	-1.000	14000.021	119755.274	0.000
0.0	30000.	0.	0.	0.
10.	30000.	0.	0.	0.
120.	44000.	0.	0.	0.
20.	30000.	0.	0.	0.
30.	30000.	0.	0.	0.
100.	44000.	0.	0.	0.
50.	38000.	0.	0.	0.
40.	30000.	0.	0.	0.
60.	30000.	0.	0.	0.
70.	30000.	0.	0.	0.
80.	40000.	0.	0.	0.
1.	1440.	0.	0.	0.
11.	1440.	0.	0.	0.
121.	2300.	0.	0.	0.
21.	1440.	0.	0.	0.
31.	1440.	0.	0.	0.
101.	2300.	0.	0.	0.
51.	2160.	0.	0.	0.
41.	1440.	0.	0.	0.
61.	1440.	0.	0.	0.
71.	1440.	0.	0.	0.
81.	2000.	0.	0.	0.
4.	.4	.8	14.	0.
14.	.4	.8	14.	0.
124.	.2	.8	14.	0.
6.	.09	.8	14.	0.
16.	.09	.8	14.	0.
126.	.20	.8	14.	0.
24.	.1	.8	14.	0.
34.	.1	.8	14.	0.
104.	.15	.8	14.	0.
26.	.12	.8	14.	0.
36.	.12	.8	14.	0.
106.	.19	.8	14.	0.
54.	.4	.8	14.	0.
56.	.15	.8	14.	0.
44.	.7	.8	14.	0.

64.	.7	.8	14.	0.
74.	.7	.8	14.	0.
84.	1.	.8	14.	0.
46.	.12	.8	14.	0.
66.	.12	.8	14.	0.
76.	.12	.8	14.	0.
86.	.13	.8	14.	0.
3.	8.	1.	0.	0.
13.	8.	1.	0.	0.
123.	8.	1.	0.	0.
23.	8.	1.	0.	0.
33.	8.	1.	0.	0.
103.	8.	1.	0.	0.
53.	8.	1.	0.	0.
43.	2.5	1.	0.	0.
63.	2.5	1.	0.	0.
73.	2.5	1.	0.	0.
83.	2.5	1.	0.	0.
47.	1.	0.	0.	0.
88.	1.	0.	0.	0.
201.	0.5	0.5	0.	0.
87.	.38	.62	0.	0.
89.	65.	0.	0.	0.
2.	6.	0.	50000.	17600.
3.	5.	0.01	0.	0.
12.	6.	0.	50000.	17600.
3.	5.	0.01	0.	0.
122.	6.	0.	50000.	8000.
3.	5.	0.01	0.	0.
22.	6.	0.	50000.	5600.
3.	5.	0.01	0.	0.
32.	6.	0.	50000.	5600.
3.	5.	0.01	0.	0.
102.	6.	0.	50000.	8000.
3.	5.	0.01	0.	0.
52.	6.	0.	50000.	20000.
3.	5.	0.01	0.	0.
42.	6.	0.	50000.	5200.
3.	5.	0.01	0.	0.
62.	6.	0.	50000.	5200.
3.	5.	0.01	0.	0.
72.	6.	0.	50000.	5200.
3.	5.	0.01	0.	0.
82.	6.	0.	50000.	14000.
3.	5.	0.01	0.	0.

3.6 Results of the Simulation

A number of runs were carried out, each run involving a single step change. Firstly, step changes were made on the reflux splitter (equipment number 201) parameters, to provide a table summarizing the effect of division of reflux on final product purity (Table 3-1). In practice, a similar table is the primary tool used by plant personnel in controlling the plant. Further runs were made involving step changes in feed flows and temperatures.

A typical DYNSYS-B graphical output is shown in Figure 3-5. For the sake of brevity, only a few of the calculated variables are included in the plot.

Comments

Bearing in mind the objectives of this study, the following comments on the process are of a tentative nature.

Generally speaking, plant personnel feel that the simulation results appear reasonable in the light of their experience. The behaviour of process conditions following the step changes is close to expectation. The lag time in the plant is, however, somewhat larger than calculated (about 10%-15%), probably due to the lags in the product purification steps (Wash Tower etc.) which were not included in this study.

The simulation indicates that the temperature in the final settler (Eq. No. 80) is lower than the solvent phase heat (Eq. No. 89) exit temperature. This is reasonable on the basis of our assumptions re the extraction process, namely that the process is reversible and

Recycle Product To:
~~STG 7 → 6000~~
FRC 9 → 6000

$$$$$ L E G E N D $$$$$

SYMBOL	STREAM	EQUIP.	VARIABLE	RANGE	
A	1		CONCH	0.00000	.10100
B	1		TOTFLOW	30000.00000	40000.00000
C	1		TEMP	10.00000	20.00000
D	4		CONCH	0.00000	.01000
E	17		CONCB	.66000	.76000
F	20		CONCB	.02600	.03600
G	17		TEMP	30.00000	40.00000
H	5		CONCB	.10000	.20000
I	26		TOTFLOW	655000.00000	665000.00000
J	26		TEMP	50.00000	60.00000
K	100		CONCH	.95000	.96000
L	26		CONCF	.00020	.00030
M	3		CONCD	.15000	.25000
N	26		CONCC	.00002	.00004

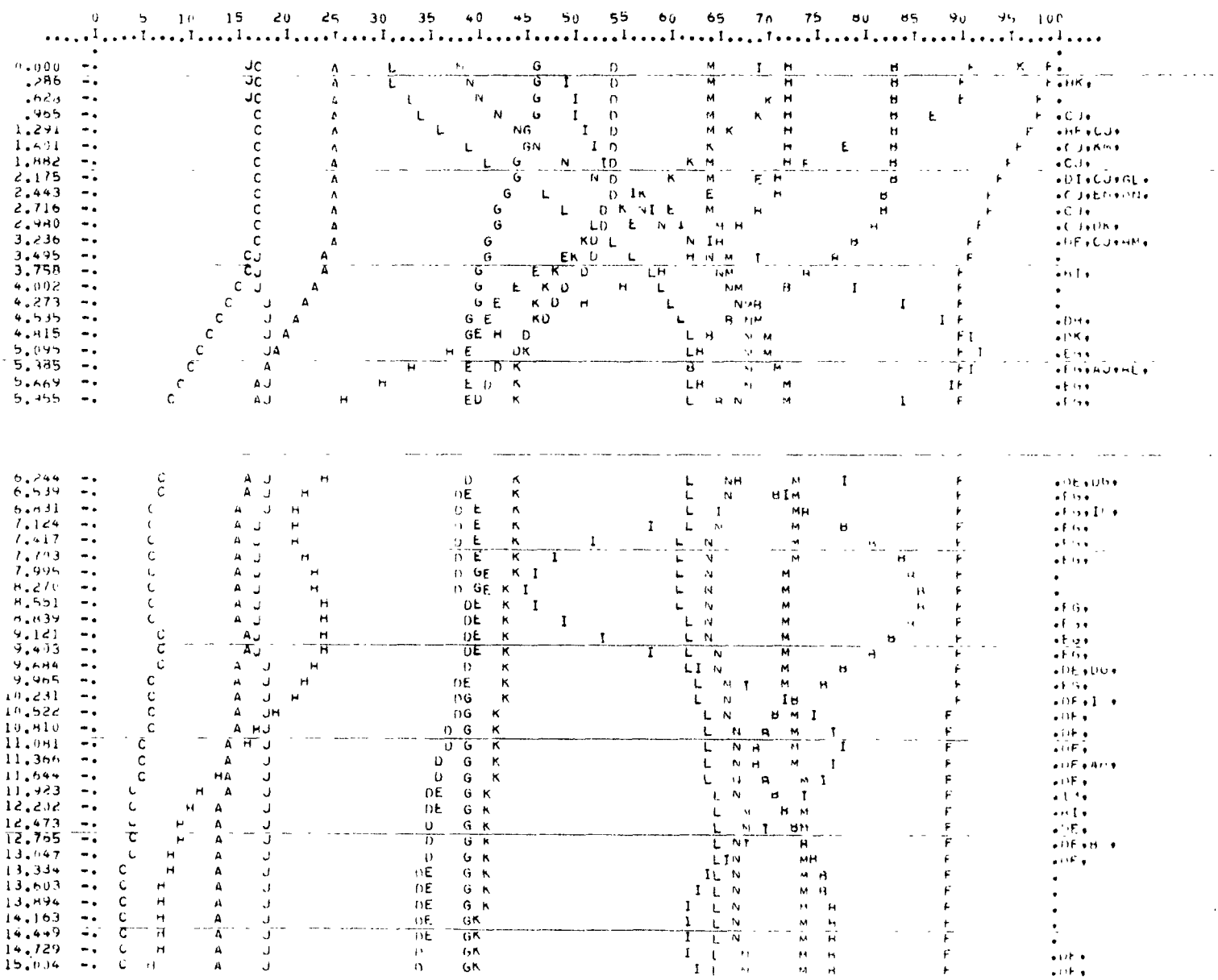


FIGURE 3-5 RESPONSE TO A STEP CHANGE

exothermic when 1-3 butadiene passes to the solvent phase. At the higher temperature after the heater, the temperature dependent rate coefficients are such that the reverse reaction (which should then be endothermic) predominates, thereby providing hydrocarbon phase for the reflux. This reduction in temperature has not been observed in the process, although it should be mentioned that no true measurement of final stage temperature is available. It is possible that a considerable amount of the reverse reaction takes place within the heater itself (Equipment 89 actually represents a bank of heaters). This would reduce the amount of cooling between heater and final stage.

It has been observed previously that inter-stage solvent coolers are provided in the unit, but have not proved to be of benefit and are not used. The coolers are located on the solvent stream leaving the 5th stage (Eq. No. 30). The simulation indicates that the 1-3 butadiene concentration in the hydrocarbon phase changes most rapidly in the first three or four stages. This concentration changes quite slowly from this point through to the final stages in the rectification section. This suggests that the most effective location for the inter-coolers would be between the first three stages (0, 10, 120). We have not tested this hypothesis in this study.

The results of the simulation cannot be claimed to be truly respresentative of plant performance; however, it seems reasonable to suppose that the simulation should indicate trends in plant operation. One possible application would be to test the effect of changing inter-cooler location, as described above. (This is a steady-state rather

than a transient problem, and use of DYNSYS-B would not be efficient use of computer time - however, the study could under the circumstances be rapidly set-up by a plant engineer. Computation time would be reduced by raising the error tolerances).

A further application is evaluation of different control schemes for the process. A major problem at present is inherent in the large process time lag - final product purity changes are observed at a considerable distance in time from the point at which corrective action was taken. This problem could be reduced by using an anticipatory (feed-forward) control system - for example, control concentration measurements could be made at a different point in the process and correlated with final product purity using DYNSYS-B . In this way, corrective control action could be taken before off-specification product is made.

TABLE 3-1

Effect of Division of Rerun Product on Product Purity

Rerun Product To		Product Purity
Wash Tower	Stage 7	
6250	5750	.9524
6000	6000	.9540
5500	6500	.9570
5000	7000	.9596
4750	7250	.9608
4000	8000	.9643

4. CONCLUSIONS AND RECOMMENDATIONS

Based on our somewhat limited experience, we feel that the DYNSYS-B approach should be a useful tool in solving many types of problems which chemical engineers often meet. Writing new unit computations is easily done - the required mass and energy balances are familiar in many cases, and linear or non-linear ordinary differential equations are handled with equal ease. Furthermore, if no new unit computations are required for a particular study, the modular nature of DYNSYS-B renders setting up the study an elementary task.

On a theoretical basis [8], the DYNSYS-B user need not concern himself with the problem of numerical instability - the system will always be stable provided the integration step size is sufficiently small (this may, of course, result in very lengthy computer runs).

The price to be paid for the ease in setting up a DYNSYS-B simulation is its lack of efficiency (from the point of view of computation time). About 10 minutes were required on a CDC 6400 to simulate 15 hours of plant time for the extraction unit study. The small examples presented in Section 2-5 required about 10 seconds until the systems approached closely to steady state.

While the DYNSYS-B approach will (theoretically at least) be applicable to any problem involving ordinary differential and algebraic equations, it is likely that a more efficient solution exists for any individual problem. Firstly, the modular approach does not permit one

to take advantage of the particular structural form of the equations. (The simplest example is combining equations and cancelling out terms). Secondly, for any particular study there will be a particular iteration technique which will have superior computation efficiency (such as Holland's Θ-method [20] for distillation columns) - such special techniques will often be difficult to incorporate in DYNSYS-B. (However, special iteration techniques will usually be readily applied within a unit computation, such as the use of the Newton Rapson method in subroutines XTRACØI and MXPLTØ2 - it is those situations involving information between unit computations that will present difficulties).

One major area for improvement of the DYNSYS-B program lies in the fact that at present a single step size is used throughout the system (for each time step). For a given truncation error, the step size is a function of the time constant associated with any particular differential equation. Equations with smaller time constants require smaller step sizes. In the typical chemical engineering system, the process units will have a wide range of time constants, ranging perhaps from a few seconds to an hour. To keep the truncation error within the specified tolerance limits, the integration step size will be determined by the system's smallest time constant, and therefore much of the system will be computed using a smaller than necessary step-size. For example, in the 1-3 butadiene extraction process the time constant for 1-3 butadiene ranges from about 0.015 hours (at the cold end) to about 0.07 hours (at the warm end). It is suggested that there could be a considerable benefit in reduced computation time if a means were devised

whereby various sections of the process were computed using their own optimum step sizes.

Another area which may be desirable to investigate is the use of higher or lower order AMØS formulae. Further, there may be some advantage in additional applications of the corrector step.

It would perhaps be interesting to examine the application of DYNSYS-B in the solution of steady-state problems. In its present form, DYNSYS-B yields the steady-state solution when allowed to run for several system time constants. Upward adjustment of the error tolerances would reduce computation time(accuracy of the transient results would not be of interest). The limiting step size would be determined by numerical instability, and would probably be about 50 times larger than that required for an accurate transient simulation. Provided the step size is below the point of instability, it is possible that problems of lack of convergence would not be encountered.

In future case studies using DYNSYS, we anticipate the study to be carried to its logical conclusion in that control aspects be evaluated. It may often be possible to proceed from a comprehensive DYNSYS-B simulation to a simple mathematical model, linearized about some given set of operating conditions. (One technique is the generation of frequency response curves by numerical integration [22]). Analysis of control systems, possibly involving direct digital control (DDC), could then be carried out according to the classical methods. In the case of the extraction process described previously, the results in Table 3.1 suggest that a linear approximation may be quite successful.

In other cases, the system may be highly non-linear and classical analysis will not be feasible. When control studies are made using the comprehensive model, there would be a need to make a compromise between accuracy of the results and computation time (integration step size) by adjustment of the error tolerances.

REFERENCES

* 1. Mosler, H.A., (1964)
PACER - A Digital Computer Executive Program for Process Simulation and Design,
M.S. Thesis, Chemical Engineering Dept., Purdue University.

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

3. Johnson, A.I. and Toong, T., (1968)
GEMCS - Information Handling Program for Analysis and Design of Engineering and Management Systems,
McMaster University, Hamilton.

4. Northcott, T.H., (1967)
The Numerical Solution of Chemical Engineering Problems Using MIMIC. Part A of a Major Report.
Dept. of Chemical Engineering, McMaster University.

* 5. MIDAS - An Analog Approach to Digital Computation
Simulation 3, No. 3, September 1964, page 17.

* 6. Syn, W.M. and Linebarger, R.N.,
DSL/90 - A Digital Simulation Program for Continuous System Modeling,
Proceedings, Spring Joint Computer Conference 1966,
pp. 165 - 187

7. Lapidus, L., (1962)
Digital Computation for Chemical Engineers,
McGraw-Hill, New York.

8. G.P. Distefano, (1968)
Stability of Numerical Integration Techniques,
A.I.Ch.E., 14, No. 6, 946.

* 9. Shell, D.L., (1958)
General Electric Company Technical Information Series Number DF58AFT679
General Electric Company, Cincinnati, Ohio.

* References not read in original form.

10. Fairchild, B.T., Wengrow, H.R. and May, F.P., (1965)
AMØS - Numerical Integration of Differential Equations with the Adams-Moulton-Shell Method,
Department of Chemical Engineering, University of Florida.

11. Dorf, R.C., (1967)
Modern Control Systems,
Addison-Wesley Publishing Company, Don Mills, Ont.

12. Eckman, D.P., (1958)
Automatic Process Control,
John Wiley & Sons, Inc., New York.

13. Amundson, N.R., (1966)
Mathematical Methods in Chemical Engineering,
Prentice-Hall, Englewood Cliffs, New Jersey.

14. Schultz, D.G., and Melsa, J.L.,
State Functions and Linear Control Systems,
McGraw-Hill, New York.

* 15. Motard, R.L., Lee, H.M., Barkley, R.W., and Ingels, D.M. (1968)
CHESS - Chemical Engineering Simulation System,
Department of Chemical Engineering, University of Houston.

* 16. Young, Jin, Kwon, and Elzy (1968)
DISCOSSA System Manual,
Department of Chemical Engineering, Oregon State University.

17. Maxwell, J.B., (1950)
Data Book on Hydrocarbons,
D. Van Nostrand Company, Princeton, N.J.

18. Perry, J.H., (1963)
Chemical Engineers' Handbook,
McGraw-Hill, Toronto.

19. Pugh, A.L., (1963)
Dynamo User's Manual,
M.I.T. Press.

20. Holland, C.D., (1966)
Unsteady-State Processes with Applications in Multicomponent Distillation,
Prentice-Hall, Englewood Cliffs, N.J.

21. Bobrow, S., Leather, A.J., and Prowse, D.R. (1969)
DYNSYS - Dynamic Systems Simulation,
McMaster University, Hamilton.

22. Dreifke, G.E., and Hougen, J.O., (1963)
Paper presented at 4th Joint Automatic Control
Conference, June 19-21, University of Minnesota.

APPENDIX I

THE AMØS EQUATIONS

Derivation of the AMØS Equations

Define the divided differences

$$f[t,t_o] = \frac{f(t_o) - f(t)}{t_o - t} \tag{1}$$

$$f[t,t_{-1},t_o] = \frac{f[t-1,t_o] - f[t,t_o]}{t_{-1} - t} \tag{2}$$

where t_{-1}, t_o, t - points in time (independent variable)

$f(t)$ - first time derivative of dependent variable

from (1) we have

$$f(t) = f(t_o) + (t-t_o)*f[t,t_o] \tag{3}$$

substituting (2) into (3), we have

$$f(t) = f(t_o)+(t-t_o)*f[t_{-1},t_o]+(t-t_o)*(t-t_{-1})*f[t,t_{-1},t_o] \tag{4}$$

This process may be continued, obtaining successively higher order equations. Only the development of the two point AMØS formulae will be given.

We may rewrite (4) as:

$$f(t) = f(t_o) + (t-t_o)*f[t_{-1},t_o] + E_o \tag{5}$$

$$E_o = (t-t_o)*(t-t_{-1})*f[t,t_{-1},t_o] \tag{6}$$

E_o then is a truncation error.

Letting X denote the dependent variable, (i.e. $dX/dt = f$) we have:

$$X(t) = X(t_o) + \int_{t_o}^{t} f(t)\, dt \tag{7}$$

Combining (5), (6), (7) and performing the integration from t_o to t_1 we get the 2-point predictor formula:

$$X_1^p = X_o + h_o*f_o*(1+\frac{h_o}{2h-1}) - h_o*f_{-1}(\frac{h_o}{2h-1}) \tag{8}$$

where we have abbreviated:

$X_1 = X(t_1)$ (the p signifies predictor)
$X_o = X(t_o)$
$h_o = t_1 - t_o$
$h-1 = t_o - t_{-1}$
$f_o = f(t_o)$
$f_{-1} = f(t_{-1})$

The truncation error is

$$\varepsilon^p = \int_{t_o}^{t_1} E_o\, dt$$

$$= \int_0^{h_o} E_o\, dh \tag{9}$$

The corrector equation is obtained in a similar way, but equation (7) is integrated from the point t_{-1} to t_o, giving

$$X_o^c = X_{-1} + \frac{h_{-1} * f_o}{2} + \frac{h_{-1} * f_{-1}}{2}$$

and raising the subscripts by 1,

$$X_1^c = X_o + \frac{h_o}{2} * (f_1 + f_o) \tag{10}$$

with the error term

$$\varepsilon^c = \int_o^{h_o} E_1 \, dh \tag{11}$$

The error term ε^p, is calculated by substituting (6) into (9) To perform the integration, the 2nd divided difference term is assumed to be a constant, denoted by f^{II}.

This gives

$$\varepsilon^p = f^{II} * \left(\frac{h_o^3}{3} + \frac{h_o^2 * h_{-1}}{2} \right) \tag{12}$$

Similarly,

$$\varepsilon^c = -f^{II} * \left(\frac{h_o^3}{6} \right) \tag{13}$$

Denoting the true value as X_T, we have

$$X_T = X^p + \varepsilon^p \tag{14}$$

and

$$X_T = X^c + \varepsilon^c \tag{15}$$

Combining equations (12), (13), (14) and (15) we find

$$\varepsilon^c = \frac{(X^p - X^c)}{3*(1 + h_{-1}/h_o)} \tag{16}$$

APPENDIX II

A COMPARISON OF DYNSYS-B NUMERICAL RESULTS WITH AN ANALYTICAL SOLUTION

The system studied has two well-stirred tanks in series, with process temperature as the only variable. A step change in temperature from 0-100 degrees is introduced in the feed stream to the first tank. The information flow diagram is given in Figure A-1.

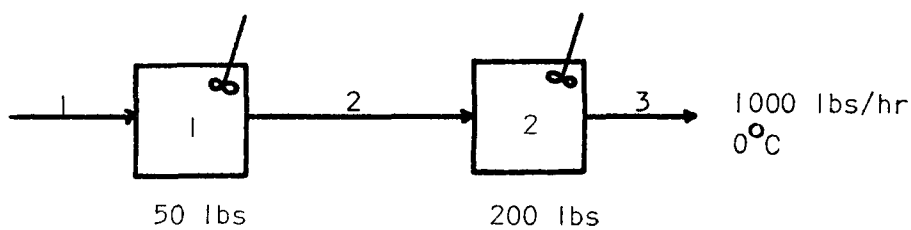


FIGURE A-1

We have assumed one component only, and a constant specific heat equal to unity. The analytical solutions for the temperatures in streams 2 and 3 are:

$$T(2) = 100. * (1.-EXP(-t/\tau_1))$$

$$T(3) = 100.*[1. + \left(\frac{1.}{\frac{1}{\tau_1} - \frac{1}{\tau_2}}\right)*\left(\frac{1}{\tau_2}*EXP\left(-\frac{t}{\tau_1}\right) - \frac{1}{\tau_1}*EXP\left(-\frac{t}{\tau_2}\right)\right)$$

where τ_1, τ_2 = time constants for tanks (1), (2)

t = time

τ_1 = 50/1000 = .050 hrs

τ_2 = 200/1000 = 0.200 hrs

The calculations were performed for three values for the upper truncation error tolerance. The results are tabulated below.

ERROR TOL.	TIME	T(2) ANALYT.	T(2) NUMER.	T(3) ANALYT.	T(3) NUMER.
.001	.00025	1.6856	1.6857	.0036	.0036
	.01925	31.9549	31.9546	1.5834	1.5834
	.18565	97.5596	97.4899	48.1142	48.0920
	.39045	99.9594	99.8933	81.0863	81.0289
.00001	.00332	6.4337	6.4338	.0538	.0537
	.01932	32.0569	32.0570	1.5948	1.5948
	.08012	79.8607	79.8604	17.3929	17.3929
	.15052	95.0733	95.0727	38.8251	38.8250
	.38732	99.9567	99.9565	80.7891	80.7881
.000001	.00335	6.4805	6.4806	.0546	.0546
	.01775	29.8827	29.8827	1.3625	1.3625
	.08495	81.7134	81.7134	18.9044	18.9044
	.14895	94.9156	94.9155	38.3810	38.3810
	.42415	99.9793	99.9793	84.0146	84.0146