NGL-44-005-084

A HYBRID COMPUTER STUDY OF THE DYNAMICS OF A TUBULAR CHEMICAL REACTOR

by

Hong-Mou Lee

FACILITY FORM 602

N71-70739
(ACCESSION NUMBER)

264
(PAGES)

CR-116208
(NASA CR OR TMX OR AD NUMBER)

(THRU)

(CODE)

(CATEGORY)

DRA

FEB 1971
RECEIVED
NASA STI FACILITY
INPUT BRANCH

A HYBRID COMPUTER STUDY OF THE DYNAMICS OF A TUBULAR CHEMICAL REACTOR

A Dissertation

Presented to

the Faculty of the Department of Chemical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Chemical Engineering

---

by

Hong-Mou Lee

December, 1970

ACKNOWLEDGEMENTS

I want to give my special thanks to Professor R. L. Motard, for his constant encouragement and guidance. Without his assistance this work would not have been completed. The financial support from Project THEMIS, ONR Contract: N00014-68-A-0151, NASA Grant NGL-44-005-084, and NSF Grant GK-2436 made this study possible.

I wish to thank the members of the Engineering Systems Simulation Laboratory, especially Messrs. Henry W. Pekar and David K. Lowell. Many thanks are also due Miss Pat Malek, who did the typing.

Finally to my wife, Seing-whe, her assistance and understanding is deeply appreciated.

ABSTRACT

The stable hybrid computer solution of a time-dependent tubular chemical reactor represented by a system of parabolic or elliptic-parabolic partial differential equations is studied. In the classical approach to the serial hybrid solution of the one-dimensional diffusion equation using the continuous-space-discrete-time (CSDT) technique, there exists an undesirably large amount of positive analog loop feedback. This makes the classical hybrid method highly unstable in the study of higher frequency transient behavior.

The serial decomposition method used in this study replaces the linear second order differential operator by two stable first order operators integrating in opposite directions and yields one-pass solutions instead of the usual iterative solutions. Thus, considerable computation economy can be expected.

Application of the serial decomposition method to the two-space dimension problem is also possible. The results of the analysis of tubular reactor dynamics in two space dimensions using the continuous-space-discrete-space-discrete-time (CSDSDT) approach not only support the findings of the previous digital computer steady-state simulation of a homogeneous turbulent flow gas phase $SOCl_2$ decomposition problem, but also give insight to the transient behavior which is not so easy to obtain otherwise.

Dynamic studies of chemical processes appears promising with the hybrid decomposition method. Special interest may be in the area of process sensitivity and stability analysis.

The difficulties and major errors involved in hybrid computation of partial differential equations are also discussed.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	INTRODUCTION	1
2.	SURVEY AND PRELIMINARY WORK	10
	DIGITAL COMPUTER SIMULATION	15
	2.1 Finite-Difference Method	15
	2.2 CSMP Method	16
	ANALOG COMPUTER SIMULATION	18
	2.3 Steady-State Analog Solution	18
	2.4 Unsteady-State Analog Solution	19
	HYBRID COMPUTER SIMULATION	22
	2.5 Steady-State Hybrid Solution	22
	2.6 Unsteady-State Hybrid Solution	24
	2.6.1 DSDT Hybrid Method	24
	2.6.2 DSCT Hybrid Method	26
	2.6.3 CSDT Hybrid Method	28
	2.6.4 Functional Approximations and Other Methods	38
	2.7 Classical CSDT Hybrid Simulation of An Isothermal Tubular Flow Reactor With Axial Diffusion	44
3.	SERIAL HYBRID DECOMPOSITION METHOD IN ONE-SPACE DIMENSION AND TIME	48
	3.1 Mathematical Development of Serial Hybrid Decomposition Method	48
	3.2 Hybrid Computer Set-up	53

CHAPTER PAGE

3.3 Simulation of An Isothermal Tubular Reactor with Axial Diffusion 56

4. SERIAL HYBRID DECOMPOSITION METHOD IN TWO-SPACE DIMENSION AND TIME 63

4.1 Mathematical Development 63

4.2 Simulation of An Unsteady-State Isothermal Tubular Reactor 68

5. SIMULATION OF A NON-ISOTHERMAL TUBULAR FLOW CHEMICAL REACTOR - ENERGY AND MATERIAL BALANCES 87

5.1 Energy Balance 89

5.2 Material Balance 95

5.3 Complete Simulation Procedure 98

6. CONCLUDING REMARKS 116

BIBLIOGRAPHY 126

APPENDIX 130

A. Finite Difference Solutions of The Steady-State One Dimensional Isothermal Tubular Flow Reactor 130

B. CSMP Prediction of the Steady-State Solutions Integrating in the Forward Direction 136

C. Steady-State Hybrid Solution of

$$\ddot{y} - 2U\dot{y} - UR_n y = 0 \quad . . . . . 139$$

D. Classical CSDT Hybrid Simulation of an Isothermal Tubular Reactor with Axial Diffusion 151

APPENDIX PAGE

E. CSDT Hybrid Simulation with the Decomposition Method . . . 159

F. Solution of An Isothermal Tubular Reactor in Two-Space Dimension and Time -- CSDSDT Decomposition Method 169

G. Non-isothermal Tubular Reactor Simulation -- Energy and Material Balances 191

H. Derivation of CSCSDT Expression in the Radial Direction . 246

LIST OF FIGURES

FIGURE		PAGE
2.1	The Most Probable Numerical Solutions	12
2.3	Analog Mechanization of $\ddot{y} = -2U\dot{y} + R_n y^n$	18
2.4	Portion of the DSCT Circuit Diagram	20
2.5	Hybrid System for Steady-State Simulation	23
2.6.2	Hybrid Block Diagram for Equation 2.10	27
2.6.3	A Classical CSDT Set-up	29
2.6.4	Green's Function Method	33
2.6.5	Hybrid Computer Mechanization of the CSDT Functional Optimization Approach	37
2.7	Typical Unstable Results of the Classical CSDT Approach . .	46
3.1	Hybrid Computing Circuit Diagram for the Serial Decomposition Method	54
3.2	Sequence of Operations	55
3.3.1	Comparison of the Transient Solutions for $\alpha\Delta t = 0.025$ . .	61
3.3.2	Comparison of the Transient Solutions for $\alpha\Delta t = 0.25$ . . .	62
4.2.1	Computing Sequence Block Diagram	78
4.2.2	Reactant Concentration Profiles in Axial Direction, Transient Results at $20\Delta t$	80
4.2.3	Radial Concentration Profiles	81
5.1	Two-Dimensional, Non-Isothermal Tubular Reactor	88
5.3.1	Heat and Mass Balance Computing Sequence Block Diagram . .	100
5.3.2	Overall Computing Schedule	102

FIGURE		PAGE
5.3.3	Transient Axial Temperature Profiles at one Δt	106
5.3.4	Transient Axial Concentration Profiles at one Δt	107
5.3.5	Transient Axial Temperature Profiles at $2\Delta t$	108
5.3.6	Transient Axial Concentration Profiles at $2\Delta t$	109
5.3.7	Transient Radial Temperature Profiles at $2\Delta t$	110
A-1	Concentration Profile of Reactant in Reactor for $n = 1/4$	131
A-2	Concentration Profile of Reactant in Reactor for $n = 1/2$	132
A-3	Concentration Profile of Reactant in Reactor for $n = 1$.	133
A-4	Concentration Profile of Reactant in Reactor for $n = 2$.	134
A-5	Concentration Profile of Reactant in Reactor for $n = 3$.	135
B-1	CSMP Prediction of Concentration Profile for 1/4 Order Reaction, $R_n = 0.2$, $U = 5$	137
B-2	CSMP Prediction of Concentration Profile for 1/4 Order Reaction, $R_n = 1.0$, $U = 50$	138
C-1	Flowchart for the Digital Portion of the Steady-State Hybrid Simulation	144
D-1	Logic and Analog Wiring Diagram	153
F-1	Analog Mechanization of $L_B(\ )$ and $L_F(\ )$ for $r \neq 1$ . .	175
F-2	Analog Mechanization of $L_B(\ )$ and $L_F(\ )$ for $r = 1$ . .	176
F-3	Transient Results at One Δt	177
F-4	Transient Results at $2\Delta t$	178
F-5	Transient Results at $3\Delta t$	179

FIGURE PAGE

F-6 Transient Results at $4\Delta t$ 180

F-7 Transient Results at $5\Delta t$ 181

F-8 Transient Results at $20\Delta t$ 182

G-1 Predicted Velocity and Eddy Viscosity Profiles 228

G-2 Curve Fitted Velocity and Eddy Viscosity Profiles Using 5th Order Polynomials 229

LIST OF TABLES

TABLE		PAGE
2.1	Number of Iterations Required to Converge Within 0.001%	16
2.7.1	Potentiometer Settings	45
2.7.2	Comparison of the Transient Results at the end of the First Time Increment for $\alpha\Delta t = 0.025$	47
3.3	Comparison of the Results ($\alpha\Delta t = 0.025$)	58
5.3	Hybrid Hardware Requirements for Energy and Material Balance	104
C-1	Result of Steady-State Hybrid Simulation	145
C-2	Comparison of Steady-State Results	147
E-1	Steady-State Solutions of the Decomposition Method for Different Δt	160

LIST OF DIGITAL PROGRAM LISTINGS

LISTING		PAGE
C	Classical Hybrid Steady-State Simulation of An Isothermal Tubular Reactor With Axial Diffusion	148
D	Classical 'CSDT' Hybrid Simulation of a Tubular Reactor With Axial Diffusion	154
E	Serial Decomposition Solution of an Isothermal Tubular Reactor in One-Space Dimension	163
F	Simulation of an Isothermal Tubular Reactor in Two-Space Dimension with the Hybrid 'CSDSDT' Decomposition Method . .	183
G-1	The Hybrid 'CSDSDT' Decomposition Solution of a Tubular Reactor with Heat and Mass Transfer	202
G-2	The Velocity and Eddy Viscosity Evaluation in a Turbulent Flow Tubular Reactor	230

CHAPTER 1

INTRODUCTION

The solution of partial differential equations is a problem which is encountered throughout the spectrum of engineering practice. It is still one of the major time consumers on large-scale scientific digital computer operations.

Consider the chemical reaction scheme,

$$nA \rightarrow B + C \tag{1.1}$$

taking place in a chemical reactor. The application of the laws of conservation of momentum, thermal energy, and mass leads to a system of differential equations which accomplishes a complete mathematical description of the reacting system. When certain assumptions are made, it is possible to classify many chemical reactors by noting that the mathematical model describing their behavior requires only a single independent variable. They are called the lumped-parameter systems. The single independent variable may be the time of operation or the distance along the reactor length. For example, in an isothermal tubular reactor with a specified amount of axial mixing, a material balance on component A at steady-state leads to the equation

$$\frac{1}{P_e} \frac{d^2C_A}{dx^2} - \frac{dC_A}{dx} - R \cdot C_A^n = 0 \tag{1.2}$$

with the necessary boundary conditions

$$C_A^0 = C_A - \frac{1}{P_e}\frac{dC_A}{dx}, \quad x = 0 \tag{1.3a}$$

$$\frac{dC_A}{dx} = 0, \quad x = 1 \tag{1.3b}$$

where

C_A = concentration of reactant A

n = reaction order

P_e = axial Peclet number (Lu/D)

R = reaction rate group

x = axial coordinate measured from reactor inlet

A constant composition of A is fed to the reactor. The Peclet number characterizes the amount of axial mixing in terms of the effective axial diffusivity D. In the asymtotic limit as D approaches infinity, $P_e \to 0$, the axial mixing becomes so large that a perfectly mixed reactor is obtained. On the other hand, if D approaches zero, $P_e \to \infty$, the axial mixing disappears and a plug flow reactor (or a lumped constant reactor (1)) is obtained (7).

If the system were adiabatic, a heat balance equation would be also required,

$$\frac{1}{P_e}\frac{d^2T}{dx^2} - \frac{dT}{dx} + R_T C_A^{\,n} = 0 \tag{1.4}$$

and the boundary conditions may be

$$T_0 = T - \frac{1}{P_e}\frac{dT}{dx}, \quad x = 0 \tag{1.5a}$$

$$\frac{dT}{dx} = 0, \qquad x = 1 \tag{1.5b}$$

where the mechanism for heat and mass diffusion have been assumed to be the same, and R_T represents the heat generation term. Based on these equations Amundson and Luss (1) have presented a detailed digital analysis of the adiabatic tubular reactor. They have found no unusual or pathological results from the behavior of the general lumped constant reactor. That is, there is always a unique stable solution which may be obtained by a marching technique starting at the reactor inlet. However, when axial dispersion is allowed, coupled with an exothermic reaction, the situation becomes much more complicated. Since the equations are second order, the integration cannot be started until the initial values of the dependent variables and its first derivatives are specified at the same starting point. In general, for this case the forward marching technique does not produce useful results since the equation is very stiff with the result that the solutions tend to diverge to plus and minus infinity for very small values of x. The problem may, however, be solved by backward marching technique assuming values of dependent variables at $x = 1$, since their first derivatives are always zero. Even then some difficulties have been encountered, particularly, when the outlet temperature $T(1)$ is near the adiabatic equilibrium temperature, and usually the computations must be carried out in double precision. Extensive computations have shown that the solution may not be unique. In fact, they have shown that there are multiple (three) steady states existing for any finite reactor (i.e., P_e is finite), and that the

solution for the semi-infinite reactor is always unique and may serve as a bounding solution for all finite reactors with the same parameters.

In summary the problem of the chemical reactor classified as lumped-parameter system is one of ordinary integration plus some iterative process for nonlinearities. Analog and/or digital simulation is completely feasible except for the special cases involving very complex chemical reactions; in such cases the analog hardware requirements or the digital computational time may become excessively large. However the basic problems are adequately defined, and there seems little further that can be done which will prove a significant contribution.

If the dependent variables of concentration and temperature are functions of at least two independent variables the chemical reactors are classified as distributed-parameter systems. These may include the various space dimensions and time. To illustrate, consider the previous reaction scheme in a tubular reactor. When one assumes steady state operation and axial symmetry, the material balance equation for component A and the corresponding heat balance become respectively,

$$D_x \frac{\partial^2 C_A}{\partial x^2} + D_r \left[\frac{\partial^2 C_A}{\partial r^2} + \frac{1}{r} \frac{\partial C_A}{\partial r} \right] - \frac{\partial C_A U}{\partial x} - RC_A^{\,n} = 0 \qquad (1.6)$$

$$K_x \frac{\partial^2 T}{\partial x^2} + K_r \left[\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right] - GC_p \frac{\partial T}{\partial x} - R\Delta H = 0 \qquad (1.7)$$

where

D_x = effective axial diffusivity

D_r = effective radial diffusivity

K_x = effective axial conductivity

K_r = effective radial conductivity

ΔH = heat of reaction

Appropriate initial and boundary conditions may be listed as

$$C_A(x,r,0) = g(x,r) \tag{1.8a}$$

$$T(x,r,0) = G(x,r) \tag{1.8b}$$

$$C_A(0,r,t) - \frac{1}{P_{e_x}} \frac{\partial C_A(0,r,t)}{\partial x} = f(r,t) \tag{1.8c}$$

$$T(0,r,t) - \frac{1}{P_{e_x}'} \frac{\partial T(0,r,t)}{\partial x} = F(r,t) \tag{1.8d}$$

$$\frac{\partial C_A(L,r,t)}{\partial x} = \frac{\partial T(L,r,t)}{\partial x} = 0 \tag{1.8e}$$

$$\frac{\partial C_A(x,0,t)}{\partial r} = \frac{\partial T(x,0,t)}{\partial r} = 0 \tag{1.8f}$$

$$\frac{\partial C_A(x,R,t)}{\partial r} = 0 \tag{1.8g}$$

$$\frac{\partial T(x,R,t)}{\partial r} = H[T(x,R,t) - Tw(x,t)] \tag{1.8h}$$

where

g,G = initial condition functionality

f,F = inlet condition functionality

P_{e_x}, P_{e_x}' = axial Peclet numbers for mass and heat respectively

L = length of reactor

R = radius of reactor

Tw = wall temperature

H = heat transfer functionality at tube wall

These mathematical representations consist of a set of nonlinear partial differential equations, coupled through the reaction and heat generation terms, are of the split boundary value type along the space domains. Simulation of this type of distributed-parameter reactor can be performed on a digital computer (14), but a considerable amount of computation time and effort is required before any results can be obtained. A possible approach for digital simulation would be to use an explicit or implicit difference technique to replace all the partial derivatives and solve the resultant non-linear simultaneous algebraic equations. Problems of stability and convergence of these equations are normally involved, the choices of the value of the spatial increment ratios, and the absolute magnitude of one of the increments determine how satisfactorily the solution to the difference equation will agree with the solution to the original partial differential equations. There are two kinds of errors usually and severely restricting the freedom which one has in choosing these parameters. The first, and more obvious, is known as the truncation error when the partial derivatives are replaced by the finite differences. The second error is known as the round-off error which is inherent in the numerical work, that numbers must be carried with a finite number of digits and the rest is always rounded off to the nearest digit. Besides, the problem of obtaining proper transport properties for a reacting system usually exists, since very little, if any, data are available for reacting systems.

Analog simulation of this type of problem, although possible, is extremely difficult due to the limited dynamic range and amount of hardware

required. In addition, the analog equipment required might become excessive, since the number of integrators required would correspond to the number of points used in the discretized space domain.

When the assumption of steady state operation is removed, accumulation terms must be added to the right sides of the previously described steady state expressions. The resultant nonlinear partial differential equations are of elliptic-parabolic type, and the chemical reactor dynamics now become initial value problems in the time domain and split boundary value problems in the space domains. To date no work has been reported on this problem. Even with the most reasonable simplifying assumptions it has been practically impossible to solve these equations numerically. With three independent variables involved, analog simulation is not feasible; also, the time and storage requirements for even the largest digital computer seem to exclude them as well.

The recent advent of hybrid computers, in which a parallel linkage exists between analog and digital computers, opens up new opportunities for tackling a great number of important chemical processes, especially in the area of solving partial differential equations.

For the solution of partial differential equations hybrid computation offers faster solution times than pure digital computation and much greater economy of analog hardware than pure analog computation. For instance in the serial method of solution, the equations can be solved in an iterative fashion on a hybrid computer by time-sharing an analog circuit for a simple differential equation. The iterative procedure updates the integration pointer from run-to-run, stores the appropriate

variables in the digital computer, and seeks interpolated values of the dependent variable from the digital computer. However, the diverse nature of the individual computers plus the complications which are involved because of interface equipment tend to introduce a number of different errors into the computation. For example, to solve a simple one dimensional diffusion equation, one usually is faced with an extremely severe problem of stability when the serial method is employed. On the other hand using the classical "parallel approach", while free of stability problems as the boundary conditions in space are always satisfied in the simulation, nevertheless, the amount of equipment required grows proportionally larger as the quantization of the space variable x is made finer to reduce trancation error. The parallel approach is obviously not a very efficient hybrid technique of solving the partial differential equations.

The purpose of this research is outlined as follows:

1. Explore the numerical algorithms which are applicable to the hybrid computer solution of the partial differential equations.
2. Use the most appropriate hybrid algorithm to study the solution of an isothermal tubular flow reactor.
3. Develop the hybrid algorithm which will be applicable to the system of coupled partial differential equations describing the non-isothermal tubular flow reactor in two-space dimensions and time. To date no work has been reported on the solution of the coupled diffusion type equations.

The momentum balance in the chemical reactor has been constantly ignored from most discussions. The problem of fluid flow alone has been studied in much more detail than either mass or heat transfer problems. One of the basic assumptions involved in this work is the application of the analogy between momentum, heat, and mass transfer to describe the transport processes in the turbulent flow tubular reactor. The problem of momentum transport was fully discussed in the early work (14). For this reason, it will not be repeated here.

The hybrid computing system in the Engineering Systems Simulation Laboratory of the Cullen College of Engineering, University of Houston, consists of an IBM 360 Model 44 digital computer and a Hybrid Systems Incorporated, Model SS-100 analog computer. Communication between the 360/44 and the SS-100 is through a HSI Model 1044 Hybrid Linkage Unit. Data transfer between the linkage and the digital computer is split between the two high-speed multiplexor channels. Control information and A-to-D input from the linkage unit are through two subchannels on one high-speed multiplexor channel. D-to-A data is transmitted from the 360/44 to the linkage over a subchannel of the second high-speed multiplexor channel. This split provides much higher effective data transfer rates with the I/O activity operating asynchronously over two separate channels.

Through the use of DAMPS2 (21,22) and the Hybrid Executive (11) the hybrid programming is easily accomplished in a FORTRAN type coding system.

CHAPTER 2

SURVEY AND PRELIMINARY WORK

One dimensional isothermal tubular flow reactor yields the following material balance equation:

$$\bar{D}\frac{\partial^2 C}{\partial x^2} - \bar{U}\frac{\partial C}{\partial x} - r(C) = \frac{\partial C}{\partial \theta} \tag{2.1}$$

where $\bar{D}$ = mean axial diffusivity of reactant.

$\bar{U}$ = mean axial velocity of mixture.

$r(C) = k_n C^n$, (k_n: n-th order reaction rate constant)

Appropriate initial and boundary conditions are:

$$\begin{aligned} &C(x,0) = g(x) \quad ; \quad 0 \leq x \leq L \\ &\frac{\partial C(L,\theta)}{\partial x} = 0 \quad ; \quad \theta \geq 0 \\ &C(0,\theta) - \frac{\bar{D}}{\bar{U}}\frac{\partial C(0,\theta)}{\partial x} = f(\theta) \quad ; \quad \theta > 0 \end{aligned} \tag{2.2}$$

where $f(\theta)$ may be represented by one of the following three general conditions:

a. Step input; $f(\theta) = C_0$; $\theta > 0$

b. Square pulse input.

c. Impulse input.

Let $U = \dfrac{\bar{U}L}{2\bar{D}}$

$$R_n = \frac{2k_n L}{\bar{U}} C_0^{n-1}$$

$$\Theta = \frac{L^2}{\bar{D}T}$$

and the transformations of the variables,

$$y = \frac{C}{C_0}, \quad z = \frac{x}{L}, \quad t = \frac{\theta}{T}$$

reduces the equation (2.1) to the following dimensionless form:

$$\frac{\partial^2 y}{\partial z^2} - 2U \frac{\partial y}{\partial z} - UR_n y^n = \Theta \frac{\partial y}{\partial t} \tag{2.3}$$

with the initial condition:

$$y(z,0) = g(z)$$

and the boundary conditions:

$$\frac{\partial y}{\partial z}(0,t) = 2U[y(0^+,t) - 1] \qquad \text{(for the step input)}$$

$$\frac{\partial y}{\partial z}(1,t) = 0$$

If $T \to \infty$, $\Theta \to 0$, the equation represents a steady-state problem.

Numerous studies have been made of the solution of the steady-state problem (7,15). The results reveal that the backmixing has profound effect on the concentration profile in the reactor. It has little or negligible effect on overall conversion when R_n is small.

The numerical solution to this type of split boundary value problem is no trivial matter, even for a steady-state case, because numerical integration from $z = 0$ to $z = 1$ is unstable and its errors tend to grow rather alarmingly.

Figure 2.1 illustrates the most probable results one could obtain in the general practice

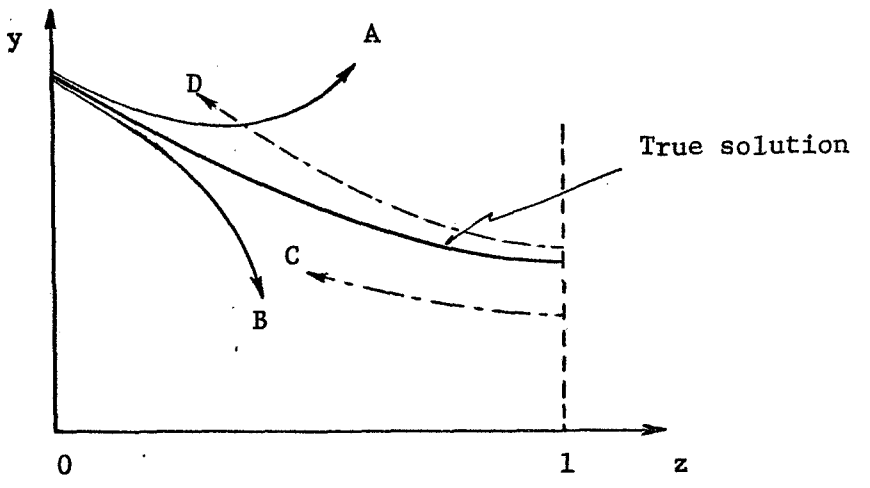


Figure 2.1 The Most Probable Numerical Solutions

Since to integrate from $z = 0$ involves an implicit boundary condition in $(\frac{dy}{dz})_{z=0}$ and $(y)_{z=0}$. Guessing a slightly larger or smaller value of $(y)_{z=0}$ will give unreasonable solution A or B, and in fact, however accurate the guess at $(y)_{z=0}$, it might prove impossible to get a physically reasonable solution out to $z=1$. Here we see that a component of the homogeneous solution of

$$\frac{d^2y}{dz^2} - 2U\frac{dy}{dz} - UR_n y^n = 0 \qquad (2.4)$$

when $n = 1$

is e^{mz}, a growing exponential, so that all the rounding off and other errors (such as the noise level of the analog computer) are multiplied by a positive exponential and therefore can blow up. Integrating from the reverse direction, however, seems very stable (as C or D). For $\frac{dy}{dz} = 0$ is fixed at $z = 1$, and it would be quite reasonable to guess $(y)_{z=1}$ (as long as it is smaller than or equal to the true solution) and integrate back to $z = 0$, iterating until the boundary condition at $z = 0$ is satisfied.

For the solution of unsteady-state problems, analog methods are complicated by the fact that such problems have more than one independent variable. Moreover, analog computers are useful for initial value problems rather than boundary value problems. The digital computer demands stepwise integration in the time domain and most often leads to uneconomically long computer times.

In applying analog methods there exist four basic approaches:

1. Continuous-space-continuous-time (CSCT).
2. Discrete-space-continuous-time (DSCT).
3. Continuous-space-discrete-time (CSDT).
4. Discrete-space-discrete-time (DSDT).

Only the DSDT method is feasible for the digital computer,

The concept of the hybrid computer appears to be the most efficient means to solve the unsteady-state problems. Among the most commonly used hybrid techniques are:

1. DSCT method.
2. CSDT method.

3. DSDT method.

4. Monte Carlo method.

Some investigators (8,12) have found the Monte Carlo method to be impractical because of the long time to complete a random walk. However, using analog noise generators and allowing the random walks to be taken in the analog domain the method may become competitive in its problem-solving-capability. The advantage of this method over more conventional methods of solution is that the value of the solution in a selected point in space can be computed without having to solve the entire multidimensional problem. Since this approach generates the solution at a point rather than continuously no further discussion will be made hereafter.

DIGITAL COMPUTER SIMULATION

Steady-state solutions of the problem described by Equation (2.4) were obtained using the finite-difference method and the CSMP (Continuous System Modeling Program) method.

2.1 Finite-Difference Method

The finite-difference representation of the differential equation given by Equation (2.4) yields

$$(1 + Uh)y_{i-1} - 2y_i + (1 - Uh)y_{i-1} - UR_n h^2 y_i^{\,n} = 0 \qquad (2.5)$$

where $h = 1/(N-1)$

i = i-th mesh point (total points = N)

The solution of these N simultaneous equations may be obtained in a straight forward manner when boundary conditions are specified:

at $i = 1$ (or $z = 0$)

$$(1 + 2Uh)y_1 - y_2 = 2Uh$$

at $i = N$ (or $z = 1$)

$$y_N = y_{N+1}$$

y_is are obtained in a recursive manner:

$$y_i = g_i - b_i y_{i+1} \quad ; \qquad i = 1, 2, \ldots, N - 1$$

where g_i and b_i are the recursive coefficients.

The method works successfully except at high $R_n(=10)$ and low reaction order $(n < 1)$, where y_i approaches to zero at some point in the reactor tube. Since there exists a numerical gap between absolute zero and the minimum allowable magnitude in any digital computer, successive readjustment of the integration range Z becomes necessary whenever y_i becomes intolerably small in magnitude, in order to overcome the numerical underflow problem. Results are shown in the Appendix A. The following table reveals the typical uneconomical feature of the digital computer integration.

TABLE 2.1

Number of iterations required to converge within 0.001% for the case $R_n = 10$.

n \ U	1	2	5	500
1/4	67	60	67	58
1/2	37	37	35	33
1 *	2	2	2	2
2	25	18	15	12
3	198	59	28	14

* Linear system

2.2 CSMP Method

The Continuous System Modeling Program is used to simulate the steady-state analog computer solution. Difficulties involved in the for-

ward direction integration are made clear in this method. In Appendix B are shown the typical results for two runs corresponding to the curves A and B in Figure 2.1.

Integration in the reverse direction requires the following transformation.

Let $w = 1 - z$

Then equation (2.4) becomes

$$\frac{d^2y}{dw^2} + 2U\frac{dy}{dw} - UR_n y^n = 0 \tag{2.6}$$

with the boundary conditions:

$$\text{at } w = 0, \qquad \frac{dy}{dw} = 0$$

$$w = 1, \qquad \frac{dy}{dw} = 2U[1 - (y)_{w = 1^-}]$$

The results are fairly consistent compared to the results obtained from the finite-difference method.

ANALOG COMPUTER SIMULATION

2.3 Steady-State Analog Solution

The analog computer is restricted to one independent variable, and it is particularly suitable for the initial value problem. Whereas in the split boundary value problem, analog memory pairs will be required to adjust the initial value assumption, and the repetitive operation of the analog computer iterates until the boundary conditions are met.

Integrating Equation (2.4) in the reverse direction gives the following unscaled circuit diagram.

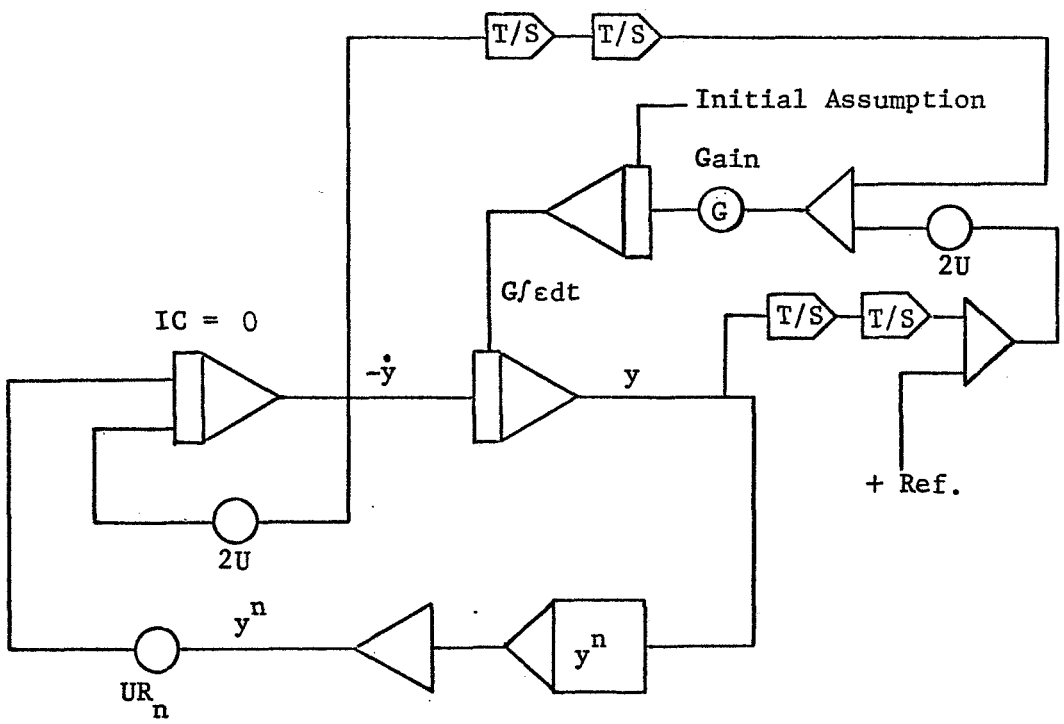


Figure 2.3 Analog Mechanization of

$$\ddot{y} = -2U\dot{y} + UR_n y^n$$

2.4 Unsteady-State Analog Solution

With more than one independent variable, discretization of the space or time domains is usually required in the analog computer. Although problems have been solved using probabilistic techniques without any actual discretization, the Continuous-Space-Continuous-Time technique (CSCT) has not been practical in the analog computation. The most widely used approach to the analog simulation of unsteady-state problems involves the discretization of all the space variables (DSCT).

By using central difference approximation, equation (2.3) is represented by a set of discretely-spaced points as

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta z^2} - 2U\left(\frac{y_{i+1} - y_{i-1}}{2\Delta z}\right) - UR_n y_i^{\,n} = \Theta \frac{dy_i}{dt} \qquad (2.7)$$

with the boundary conditions

$$\frac{y_2 - y_1}{\Delta z} = 2U(y_1 - 1) \qquad \text{at} \quad z = 0$$

and

$$y_{N+1} = y_{N-1} \qquad \text{at} \quad z = 1$$

The general analog then takes the form of a rectangular network of resistors in a one-dimensional array with a capacitor linking each node point to ground. For this purpose at least an integrator and an adder are required at each grid point. Equation (2.7) is rewritten for each grid point i

$$y_i = \frac{1}{\alpha\Delta z^2}\int_0^t [y_{i+1}(1 - U\Delta z) + y_{i-1}(1 + U\Delta z) - y_i(2 + UR_n\Delta z^2 y_i^{n-1})]\, dt \qquad (2.8)$$

and for the two boundary points (at $i = 1$ and N)

$$y_1 = (y_2 + 2U\Delta z)/(1 + 2U\Delta z)$$

$$y_N = \frac{1}{\alpha\Delta z^2}\int_0^t [2y_{N-1} - y_N(2 + UR_n\Delta z^2 y_N^{n-1})]\, dt$$

The mechanization of the analog system then takes the form shown in Figure 2.4, with the initial condition applied to each integrator corresponding to each space point at time $t = 0$.

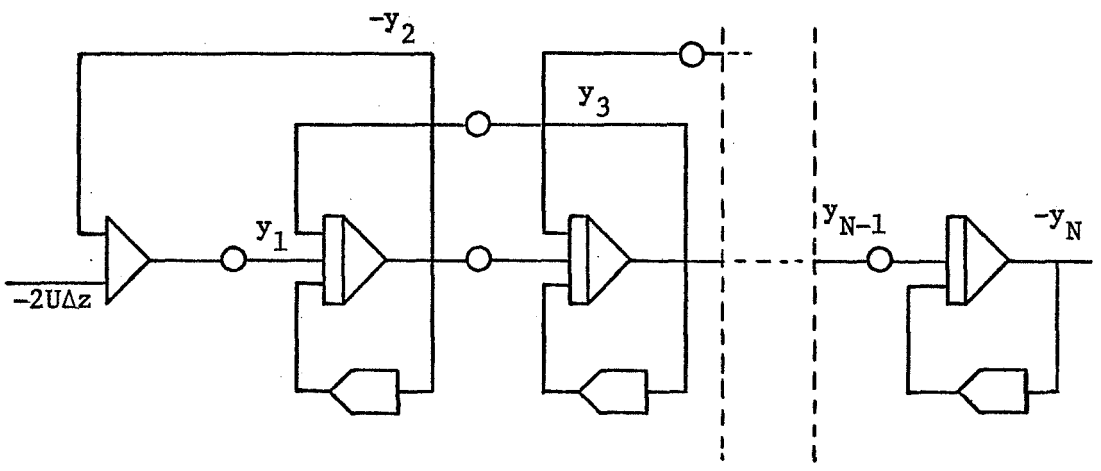


Figure 2.4 Portion of the DSCT circuit diagram

An application of dynamic memory (13) has led to continuous-space-discrete-time (CSDT) method. In this method the finite-difference approximation is taken for the time variable t and the computer time variable represents the problem space variable z. Equation (2.4) then becomes

$$\ddot{y}_t - 2U\dot{y}_t - UR_n y_t^{\,n} = \text{Ⓗ}\ \frac{y_t - y_{t-\Delta t}}{\Delta t} \tag{2.9}$$

where the function $y_{t-\Delta t}$ has been stored from the previous calculation, so that y_t is the only unknown.

Using conventional analog computer techniques the application of the CSDT method presents three major problems: continuous memory and play back, the determination of the integrator initial condition to match the split boundary values, and finally the most important of all, the stability of the analog feedback loop. A more detailed analysis of the stability problem of the CSDT method will appear in the later section.

If all the partial derivatives are replaced by finite-difference expressions the technique is then called the discrete-space-discrete-time (DSDT) method. For each time level there are as many simultaneous algebraic equations as there are points in the space domain, this is not too different from the pure digital approach which is based on simultaneous algebraic equations.

HYBRID COMPUTER SIMULATION

The most important distinction between an analog computer and a digital computer is that the analog computer is a parallel device while the digital computer is a sequential device.

The completely integrated hybrid computer makes optimum use of the best features of both analog and digital machines - the rep-op and the speed of the analog, and the logic, memory and accuracy of the digital. Hence, hybrid computation is practical for the solution of engineering problems that have not been solved at all in the past, or have been solved inefficiently on the analog or digital machine alone.

For Steady-State problems (or more explicitly, ordinary differential equations) the most common hybrid technique is:

(1) Guess at the unknown initial value.

(2) Solve the problem as an initial-value problem.

(3) Compare the specified boundary conditions against the computed values.

(4) Update the step (1) and iterate the process until the solution converges.

In the Unsteady-State problems (partial differential equations) hybrid computation offers faster solution times than pure digital computation and much greater analog hardware economy than pure analog computation.

2.5 Steady-State Hybrid Solution

Following the classical approach to the solution of the steady-state problems, the analog circuit described in Figure 2.3 may be modified to give the following hybrid system diagram:

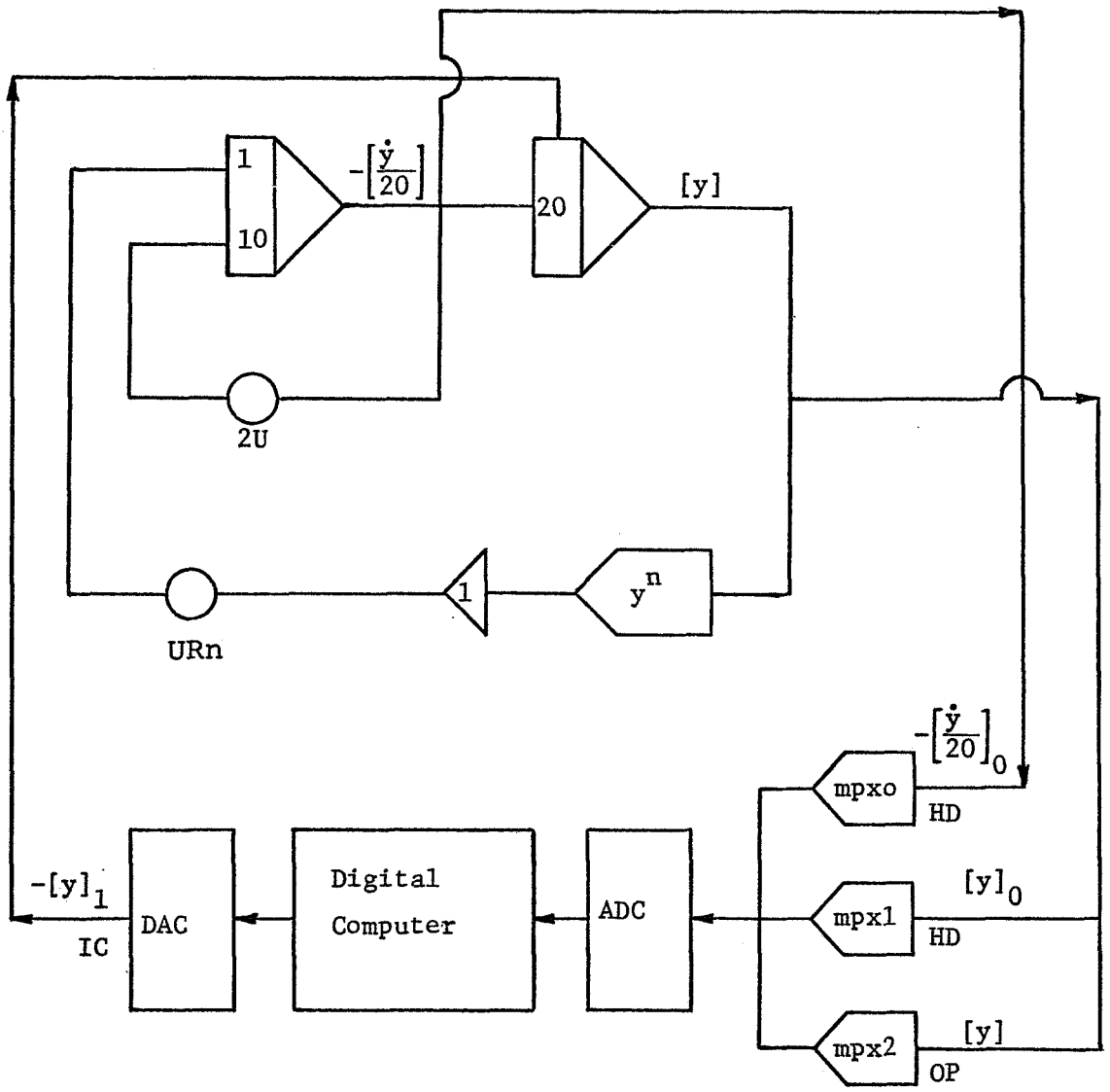


Figure 2.5 Hybrid System for Steady-State Simulation

The computed boundary values may be sampled at the end of each analog rep-op cycle, and the digital program then updates the initial value for the next analog cycle until the final convergence has achieved.

A case study was made to demonstrate the adaptability of the hybrid simulation of boundary value problems. The results are quite satisfactory when compared with the analytic and digital solutions. Of course, with this simple steady-state problem, numerous features of the hybrid technique cannot be fully exploited. The details and results are presented in Appendix C.

2.6 Unsteady-State Hybrid Solution

One of the most important applications of hybrid computers is in solving partial differential equations.

The problem of unsteady-state processes usually assumes that one of the independent variables is time and the rest are spatial variables. For a one-space dimensional partial differential equation, both the CSDT and DSCT methods yield a set of ordinary differential equations while the DSDT method leads to a set of algebraic equations.

2.6.1 DSDT Hybrid Method

Here the problem to be solved is first programmed for conventional digital computer solution. Then, analog hardware is considered as comprising subroutines employed in a digital computer solution. The advantage of the hybrid technique over a pure digital solution is that the calculation of the non-linear terms and the inversion of the matrix can be accomplished much more rapidly. However, the problem of convergence

for more complicated problems makes further development of this method debatable.

Chan (4) reported the application of a block relaxation technique to avoid the large amount of analog equipment needed for a high-order algebraic system. Analog arrays were constructed to solve simultaneous algebraic equations instantaneously. This analog block was time-shared as the relaxation proceeded from a block of spatial and temporal cells to the neighboring block.

As for a simple example, consider the following system of equations.

$$\left(\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2}\right)_{ij} \approx \frac{u_{ij} - u_{i-1,j}}{\Delta t} - \frac{u_{i,j+1} - 2U_{ij} + U_{i,j-1}}{(\Delta x)^2} = 0$$

where $U_{ij} = U(t_i, x_j) = U(i\Delta t, j\Delta x)$

In matrix form we can write:

$$\begin{bmatrix} A & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & A & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & A & -1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & .. & .. & 0 & -1 & A & -1 \\ 0 & .. & .. & .. & 0 & -1 & A \end{bmatrix} \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ij} \end{bmatrix} = \begin{bmatrix} \delta u_{i-1,1} + u_{i,0} \\ \delta u_{i-1,2} \\ \vdots \\ \delta u_{i-1,J} + u_{i,J+1} \end{bmatrix}$$

where J is the number of points along the x-coordinate, and $\delta = (\Delta x)^2/\Delta t$, $A = 2 + \delta$ respectively. Instead of setting up the complete matrix, a block of five x-points is set up. Assuming boundary conditions at each ends, the whole process is repeated again and again until con-

vergence of the results at the certain time increment is achieved, then the operation advances to the next block of temporal points.

2.6.2 DSCT Hybrid Method

The chief disadvantage of this classical analog technique is the direct proportion between the number of finite-difference grid points in the space domain and the analog hardware requirement. Accordingly, it has been suggested for the hybrid computing techniques that a basic analog computing block could be time-shared. The analog system is then used repeatedly to provide solutions for each of the sections of the space domain. It is therefore necessary to use iterative techniques to "match" all the time-shared sections.

Coulman, Svetlik and Clifford (6) applied the DSCT method to linear one space dimension, parabolic equation. The resulting ordinary differential equations were uncoupled using matrix transformation thus allowing any number to be solved at one time on the analog.

Bishop and Green (2) used the DSCT and the alternating direction implicit technique to solve for the pressure history in a two-space dimensional oil reservoir problem. Results compared quite favorably with an all digital solution in accuracy and the hybrid solution was accomplished in much short period of time.

Rewriting Equation (2.8) for a first order problem leads

$$y_i = \frac{1}{\alpha \Delta z^2} \int_0^t [y_{i+1}(1 - U\Delta z) + y_{i-1}(1 + U\Delta z) - y_i(2 + UR_n \Delta z^2)]\, dt \tag{2.10}$$

when taking each grid point as a basic analog computing block, the hybrid block diagram for Equation (2.10) is illustrated in Figure 2.6.2.

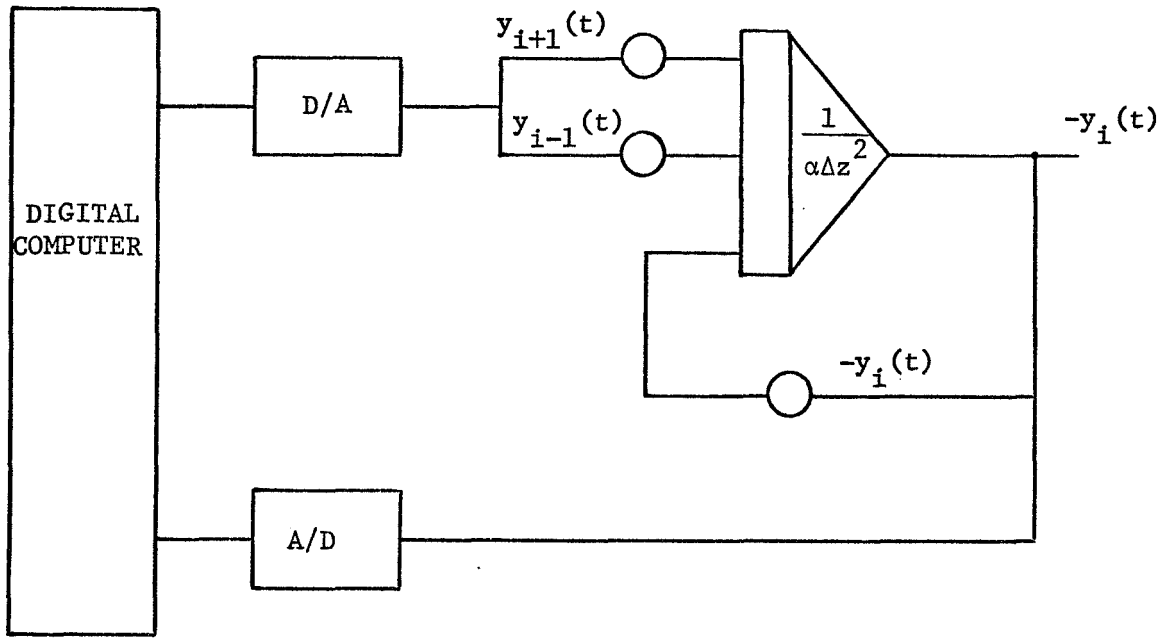


Figure 2.6.2 Hybrid Block Diagram for Equation 2.10

The analog system is used successively to represent the grid point of the space domain, starting with one end and terminating at the other end. The solution values are played back and stored until convergence has been obtained. Unfortunately, the need for these iterations, as well as the errors introduced in the hybrid loop, greatly limits the applicability of this technique.

2.6.3 CSDT Hybrid Methods

Contrasting to the DSDT hybrid method the serial or CSDT hybrid method is an analog computer oriented technique.

Because of the engineering importance and the abundance of problems characterized by one-dimensional diffusion equations a considerable amount of effort (9) has been devoted to this method for solving non-linear parabolic partial differential equations in one space-dimension.

In the CSDT method, the problem time-variable is discretized and the analog computer time-variable represents the problem space-variable. A closed analog loop is employed to integrate a spatial-dependent ordinary differential equation at successive time levels. By controlling the analog integration interval, this method has an additional advantage over others in handling moving boundaries.

Another important feature of the CSDT method is that the analog hardware requirements are very modest because a relatively small analog circuit is time-shared to solve the entire problem. In practice, however, considerable difficulties were encountered in obtaining dependable results using the CSDT method (9,24). The major difficulty is that in solving the basic spatial sweep from boundary to boundary, at each discrete time level, one usually is faced with an extremely severe problem of stability due to high analog loop gains.

A. The Classical CSDT Approach

As an example, consider a simple diffusion equation

$$\frac{\partial^2 y}{\partial x^2} = \phi \frac{\partial y}{\partial t} \tag{2.11}$$

with CSDT approach, denoting $y(x,t_i)$ by $y_i(x)$ where $t_i = i\Delta t$, Equation 2.11 becomes;

$$\frac{d^2 y_i(x)}{dx^2} = \frac{\phi}{\Delta t} [y_i(x) - y_{i-1}(x)] \tag{2.12}$$

Schematically the hybrid setup for the equation (2.12) is shown in Figure 2.6.3. The configuration in Figure 2.6.3., however, involves a positive feedback loop of four operational amplifiers with an overall loop gain of $\phi/\Delta t$.

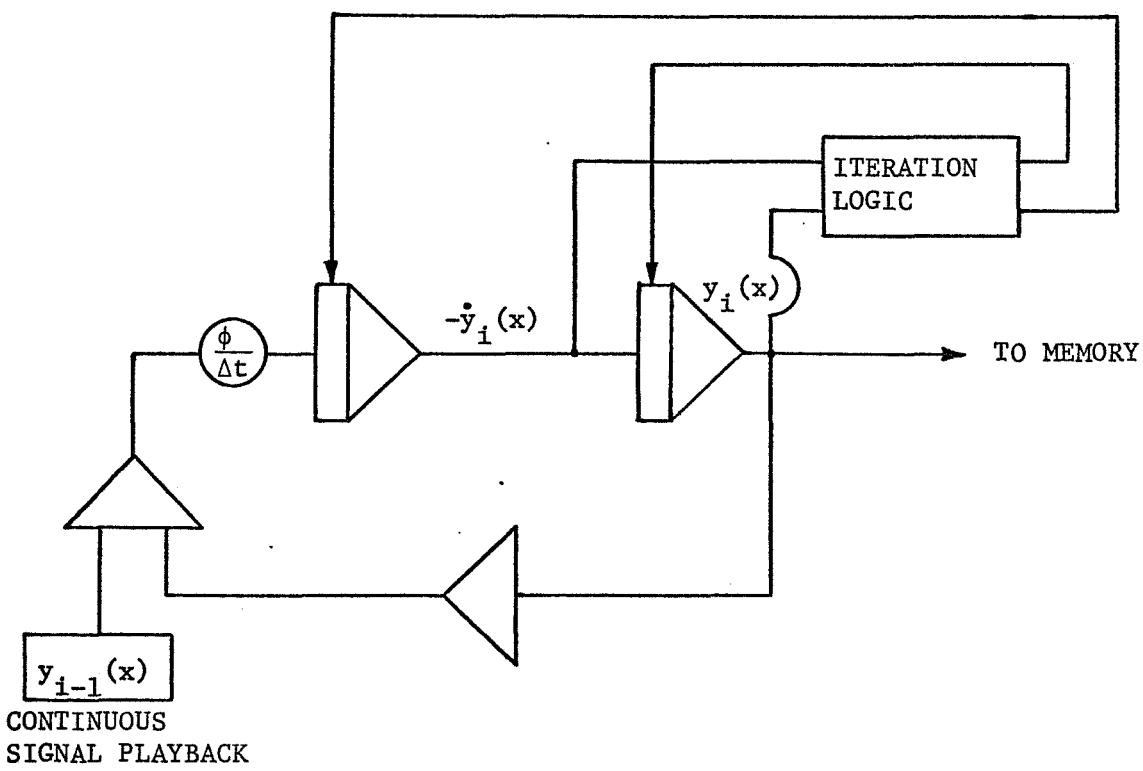


Figure 2.6.3 A Classical CSDT set up

Again, we see that a component of the homogeneous solution of Equation (2.12) is a growing exponent, $\exp(\sqrt{\phi/\Delta t}\ x)$. If we are to be assured that the transient study results will be meaningful the size of Δt must first be determined. Moreover, to keep the truncation error in the difference quotient scheme small, one must keep Δt small. Yet, a smaller Δt implies higher gains in the analog loop which is inherently constrained by the phase margin of the amplifiers. On the other hand using a large Δt will not only produce a larger error, but may also lose much of the transient information. If we try to perform the integration of Equation (2.12) in the backward sense, the method is still computationally unstable. This will be illustrated later in this chapter.

For this reason numerous approaches have been proposed by many investigators (5,9,16,24-32) in order to alleviate the inherent instability problem of the analog components.

B. The Green's Function Method (5,31)

Briefly, the Green's function method is equivalent to the so-called method of variation of parameters in elementary differential equations, with the restriction that ϕ in Equation (2.11) is a constant.

Consider equation (2.12) with the subscript i dropped, and let $g(x)$ representing the known function $y_{i-1}(x)$

$$\frac{d^2y}{dx^2} = \frac{1}{\beta^2}\ [y(x) - g(x)] \tag{2.13}$$

where $\beta^2 = \Delta t/\phi$

satisfying the boundary conditions

$$y(0) = y(1) = 0 \tag{2.14}$$

Let $y_1(x)$ and $y_2(x)$ be two independent homogeneous solutions, one then writes

$$y(x) = A(x)\ y_1(x) + B(x)\ y_2(x) \tag{2.15}$$

substituting $y(x)$ into equation (2.13), we have, following the standard method of variation of parameters.

If $y' = A\,y_1' + B\,y_2'$

provided that $A'y_1 + B'y_2 = 0$

Also

$$y'' = A'y_1' + B'y_2' + Ay_1'' + By_2''$$

Thus $y'' - \frac{1}{\beta^2}\,y = A'y_1' + B'y_2' + Ay_1'' + By_2'' - \frac{1}{\beta^2}\,(Ay_1 + By_2)$

and since y_1 and y_2 are the solutions of the homogeneous equation

$$y'' - \frac{1}{\beta^2}\,y = 0$$

Finally we have

$$\begin{cases} A' y_1 + B' y_2 = 0 & (2.16a) \\ A' y_1' + B' y_2' = -\frac{1}{\beta^2} g(x) & (2.16b) \end{cases}$$

as conditions for A and B. The independency of y_1 and y_2 implies then a unique solution for A' and B'

$$A'(x) = \frac{1}{w(x)} [+ \frac{1}{\beta^2} g(x) \; y_2(x)] \tag{2.17a}$$

$$B'(x) = \frac{1}{w(x)} [- \frac{1}{\beta^2} g(x) \; y_1(x)] \tag{2.17b}$$

where

$$w(x) = \begin{vmatrix} y_1(x) & y_2(x) \\ y_1'(x) & y_2'(x) \end{vmatrix} \neq 0$$

is the Wronskian determinant, or

$$y_1(x) \; y_2'(x) - y_2(x) \; y_1'(x) \neq 0 \tag{2.17c}$$

Upon integration, we obtain

$$y(x) = c_1 y_1(x) + c_2 y_2(x) + \frac{1}{\beta^2} y_1(x) \int_0^x \frac{1}{w(\xi)} g(\xi) \; y_2(\xi) \; d\xi$$
$$- \frac{1}{\beta^2} y_2(x) \int_0^x \frac{1}{w(\xi)} g(\xi) \; y_1(\xi) \; d\xi \tag{2.18}$$

where c_1 and c_2 are constants to be determined by the boundary conditions, equation (2.14).

By substituting $y_1(x) = e^{\frac{1}{\beta}x}$, $y_2(x) = e^{-\frac{1}{\beta}x}$ and after some algebraic manipulation, denoting $\frac{1}{\beta}$ by σ

$$y(x) = \frac{1}{\beta^2} \int_0^1 G(x,\xi)\, y_{i-1}(\xi)\, d\xi \tag{2.19}$$

where

$$G(x,\xi) = \frac{1}{\sigma \operatorname{Sinh} \sigma} \begin{cases} \operatorname{Sinh} \sigma\, \xi \operatorname{Sinh} \sigma\, (1-x) & 0 \leq \xi \leq x \leq 1 \\ \operatorname{Sinh} \sigma\, x \operatorname{Sinh} \sigma\, (1-\xi) & 0 \leq x \leq \xi \leq 1 \end{cases} \tag{2.20}$$

known as the Green's function of the operator $\frac{d^2}{dx^2} - \sigma^2 = 0$ satisfying the homogeneous boundary conditions.

The hybrid set-up is shown in Figure 2.6.4.

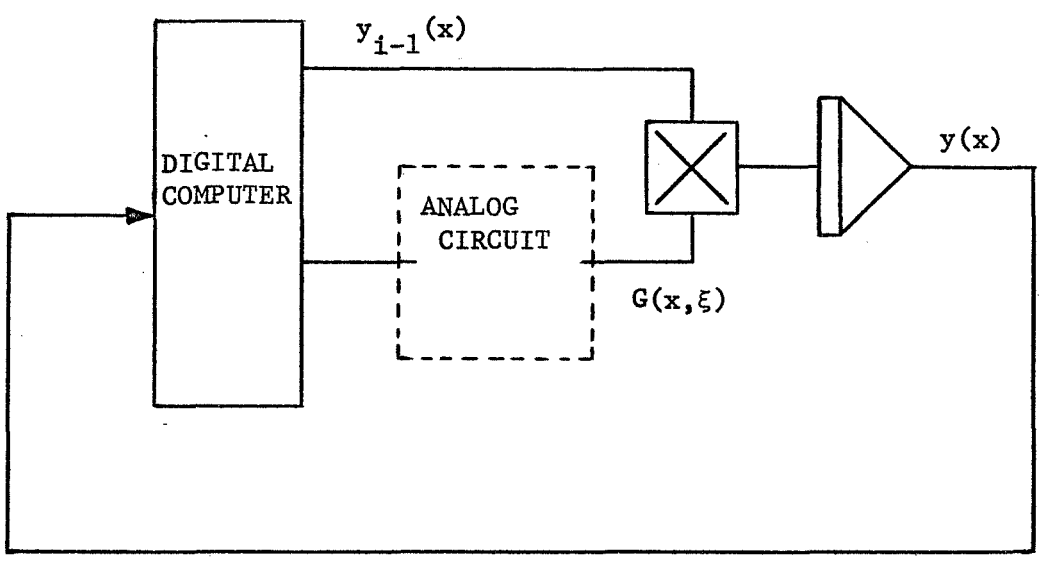


Figure 2.6.4 Green's Function Method

The speed of solution is much slower compared to that of the classical CSDT approach. One may notice that a complete integration of equation (2.19) in ξ is necessary to get one point of $y_i(x)$. Since the Green's functions of the spatial differential operator should be pre-computed, and the solution to the differential problem is replaced by an integral relation to the non-homogeneous terms, depending on the number of samples used in the space-direction, it will require the same number of integrations per time step.

The physical significance of the Green's function $G(x,\xi)$ is the response of the system to an impulse excitation at $x = \xi$. Thus, equation (2.19) is nothing more than the result of superposition. The advantage here is that the boundary condition is automatically satisfied, i.e., it is already part of the whole system. In other words, the Green's Function Approach solves equation (2.13) not as an initial value problem, but as part of a system given by equations (2.13) and (2.14), incorporating both equations together at the same time. Therefore, there should have no stability problem at all.

This method, however, has only limited application. The success of the whole scheme hinges on the fact that an analytic form of the Green's function can indeed be determined. Such a requirement, generally speaking, is difficult to meet.(24). More details are discussed in Witsenhausen (31) and Chan (5).

C. The Integral Equation Method (4,5)

In essence, the principle of the integral equation approach in not very much different from that of the Green's function approach.

Using the same equation (2.12) again, with homogeneous boundary conditions. Instead of seeking the Green's function for $\frac{d^2}{dx^2} - \frac{\phi}{\Delta t}$, the method seeks the Green's function for $\frac{d^2}{dx^2}$, satisfying the same boundary conditions.

The solution to equation (2.13) is then given by

$$y(x) = - \int_0^1 \frac{k(x,\xi)}{\beta^2} [y(\xi) - y_{i-1}(\xi)] \, d\xi \tag{2.21}$$

The function $k(x,\xi)$ is called the kernel of the integral equation. The major difference between equations (2.19) and (2.21) is that the unknown function $y(x)$ appears on both sides of the latter. It is also known as the Fredholm equation of the second kind. A practical way of solving this integral equation is the so-called Neumann series, which is not always convergent (5). The implementation of the Neumann series has been well studied (5), therefore will not be discussed at any length here. In fact, the solution time for the integral equation method is even longer than the Green's function method, for it takes the same amount of time for each iteration as it would for one time step in the Green's function method, and several iterations for one time step solution.

D. The Functional Optimization Approach

Hara and Karplus (9) reported a different way of handling the instability problem by transforming the problem into one of optimal control. Introducing a control function $\hat{y}_i$ in Equation(2.12) results in

$$\frac{d^2y_i}{dx^2} = \frac{\phi}{\Delta t}(2\hat{y}_i - y_i - y_{i-1}) \tag{2.22}$$

As can be seen from Figure 2.6.3, the classical approach of the CSDT set up to the equation (2.12) always leads to the analog computer circuit containing a closed loop comprised of an even number of operational amplifiers. Such a circuit is inherently unstable and can be expected to exhibit marked sensitivities to errors in initial conditions and component inaccuracies. By inverting the sign of y_i as shown in the equation (2.22), the method retains the advantageous features of the serial CSDT method while obviating the need for an even number of operational amplifiers in the analog loop.

The problem then becomes one to find a control policy $\hat{y}_i$ in order to minimize the criterion function which is defined by

$$0 = \int_{x=0}^{\ell} (e_i)^2 dx \tag{2.23}$$

where

$$e_i = \hat{y}_i - y_i \tag{2.24}$$

(for when $\hat{y}_i \to y_i$, $e_i \to 0$)

With the substitutions, the state equations to be solved are

$$\begin{aligned} \frac{dy_i}{dx} &= u_i \\ \frac{du_i}{dx} &= \frac{\phi}{\Delta t}(2\hat{y}_i - y_i - y_{i-1}) \\ \frac{dv_i}{dx} &= e_i^2 \end{aligned} \tag{2.25}$$

along with the proper initial conditions.

Updating of the control policy involves solving the two sets of adjoint equations. Actual updating of the control policy is accomplished by

$$\hat{y}_i \text{ (NEW)} = \hat{y}_i \text{ (OLD)} + \delta\hat{y}_i \tag{2.26}$$

This method, although feasible and possibly extensible to higher dimensions, requires considerable mathematical manipulation and hardware. The simplified hybrid mechanization is shown in Figure 2.6.5.

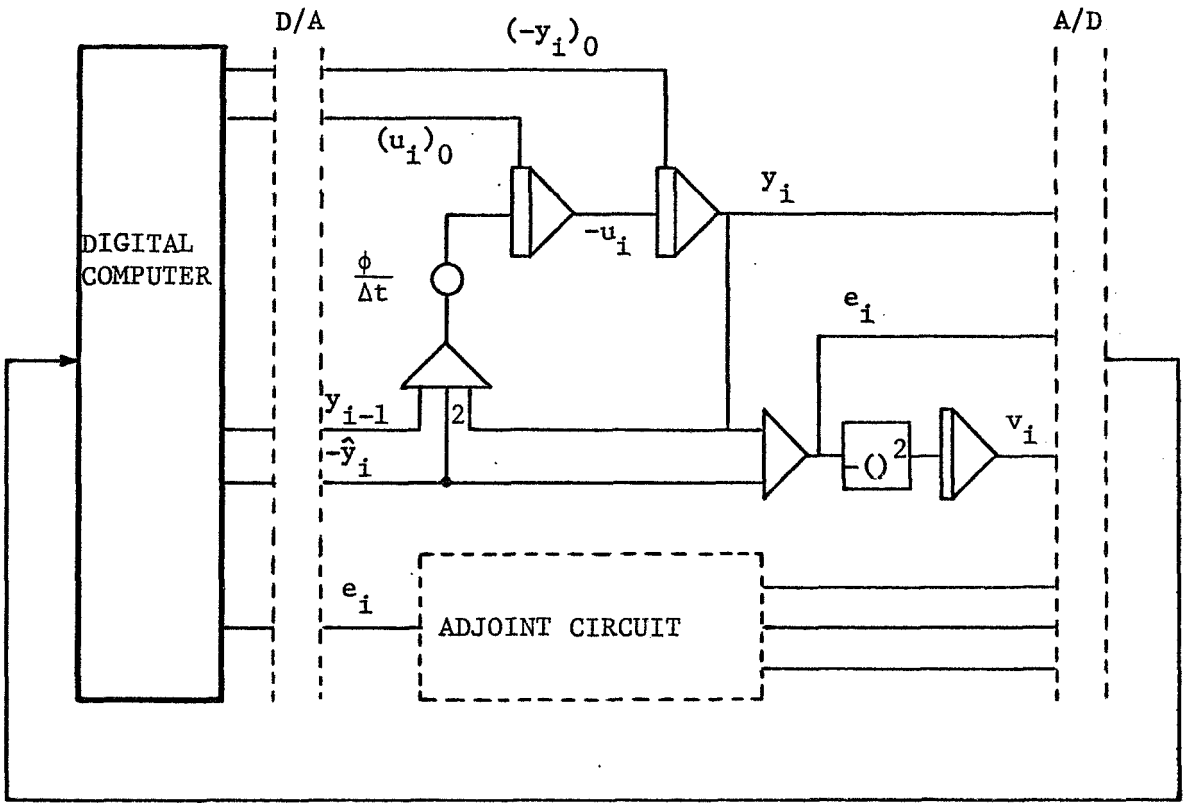


Figure 2.6.5 Hybrid Computer Mechanization of the CSDT Functional Optimization Approach

E. The Decomposition Method

Vichnevetsky (24,25,28) proposed a stable method which had been used in the optimal control problems (called the "Riccatti transformation") (32). The general solutions are obtained first by integrating the factored differential equation for the homogeneous case. One of the factored equations is stable in the forward direction and the other in the backward direction. The homogeneous solutions are obtained prior to the actual calculations since they are time independent. The complete solution is then obtained by adding a particular solution obtained at each time step to the homogeneous solutions. This method allows one to replace the usual iteration schemes for one dimension and time with one-pass per time step at a cost of augmenting the system with additional equations.

Detailed discussion of the decomposition method will be given in the later section since it is the technique selected for the major part of this study.

2.6.4 Functional Approximations and Other Methods

Another approach to the solution of initial value partial differential equation problems which is widely used in digital computers are the Functional Approximation Methods. These methods have an objective identical to that of the parallel finite difference method; that is, they provide a device by which the solution of these problems can be approximated by the solution of initial value problems in sets of ordinary differential equations.

One of the earliest methods in functional approximation was proposed by Rayleigh and later extended by Ritz and Galerkin (27,19). The application of these methods to the reduction of partial differential equations to ordinary differential equations was proposed independently by Kantorovich and by Poritzky (27). What all these methods have in common is the fact that the solution $u(x,t)$ of a partial differential equation in space x and time t is approximated by a function $u^*(x,a_1(t),a_2(t),\ldots,a_N(t))$, where the dependence upon x is prescribed.

Consider the following initial value problem:

$$\frac{\partial u}{\partial t} = X(u) + g(x,t) \tag{2.27}$$

$$x \in D;\ t \geq 0;$$

where $X(\cdot)$ denotes a partial differential operator involving spatial derivatives of $u(x,t)$, $g(x,t)$ is a given function specified over spatial domain D for time, $t \geq 0$ with boundary S.

The appropriate initial and boundary conditions are:

$$u(x,0) = u^0(x) \quad \text{in } D \tag{2.28a}$$

$$\phi_m[u(x,t)] = \beta_m(x,t) \quad \text{on } S, \quad m = 1,2,\ldots,M \tag{2.28b}$$

Assume an approximation solution of the form

$$u^*(x,t) = \sum_{i=1}^{N} a_i(t) \cdot f_i(x) \tag{2.29}$$

where the $f_i(x)$ constitute a set of prescribed functions of space over D.

The convergence of approximate solution $u^*(x,t)$ to the exact solution perhaps is best measured by the departure from zero of all the corresponding residuals:

The equation residual:

$$R(u^*) = \frac{\partial u^*}{\partial t} - X(u^*) - g(x,t)$$

$$= \sum_{i=1}^{N} \frac{da_i}{dt} f_i(x) - X(\sum_{i=1}^{N} a_i f_i) - g(x,t) \qquad (2.30)$$

The initial residual:

$$I(u^*) = u^*(0) - u^0(x) = \sum_{i=1}^{N} a_i(0)\ f_i(x) - u^0(x) \qquad (2.31)$$

The boundary residuals:

$$B_m(u^*) = [\phi_m(u^*(x,t)) - \beta_m(x,t)] \quad \text{on} \quad S,\ m=1,2,\ldots,M \qquad (2.32)$$

Obviously, when u^* is the exact solution of the problem, all residuals are identically zero.

In this respect, the choice of the form of u^* defines the following methods.

1) An interior method: The boundary conditions are always satisfied ($B_m \equiv 0$).

2) A boundary method: The differential equations are identically solved ($R \equiv 0$).

3) A mixed method: One in which neither the equation nor the boundary conditions are satisfied.

The approximation process consists in minimizing the residuals for the above methods, of which, the least squares procedure and methods of weighted residuals are most commonly employed. Some special cases of methods of weighted residuals are the Ritz-Galerkin method (orthogonality method), the method of collocation (requiring that the residual vanish at a selected set of points), the method of moments, and the method of subdomains:

1) The least squares procedure: The inner product

$$< R,R > = \int_D R^2 dD \tag{2.33}$$

is formed and its minimum is sought.

2) Methods of weighted residuals: The minimization of R is interpreted as equating to zero N weighted integrals or inner products:

$$< R,w_j > = 0, \quad j=1,2,\ldots,N \tag{2.34}$$

The w_j are a set of weighting functions which can be selected arbitrarily,

a) The Ritz-Galerkin method, where the weighting functions are the coordinating functions $f_j(x)$ themselves.

$$< R,f_j > = \int_D R \cdot f_j(x) dD = 0, \quad j=1,2,\ldots,N \tag{2.35}$$

Substituting equation (2.30) into (2.35) yields

$$\sum_{i=1}^{N} \frac{da_i}{dt} \cdot \langle f_i, f_j \rangle = \langle X(\sum_{i=1}^{N} a_i f_i), f_j \rangle + \langle g, f_j \rangle,$$

$$j=1,2,\ldots,N \qquad (2.36)$$

which is a system of ordinary differential equations of order N.

b) Collocation, in this case the w_j are N dirac delta functions

$$w_j = \delta(x-x^j), \qquad j=1,2,\ldots,N \qquad (2.37)$$

c) Method of moments, one selects a set of linearly independent polynomials, $p_j(x)$, in the variables x.

$$w_j = p_j(x), \qquad j=1,2,\ldots,N \qquad (2.38)$$

d) Subdomain method, the domain D is subdivided into N subdomains, not necessarily disjoint,

D_j, $j=1,2,\ldots,N$ and

$$w_j \begin{cases} = 1 & \text{in } D_j \\ = 0 & \text{outside of } D_j \end{cases} \qquad (2.39)$$

and

$$\int_D R dD = 0, \quad j=1,2,\ldots,N \qquad (2.40)$$

In most applications, u^* is linear in the $a_i(t)$, i.e., $u^* \sum_{i=1}^{N} a_i(t) f_i(x)$. The $a_i(t)$ satisfy a set of ordinary differential equations obtained as the result of the approximation process. This

system of ordinary differential equations is then integrated by analog, digital or hybrid computer methods.

The freedom that exists in the choice of the coordinate functions $f_i(x)$ (or sometimes called the eigen-functions) leads to the development of the Assumed Mode Solution (3,17), where in some cases properly selected modes provide a set of coordinate functions in which the solution series converges more rapidly than that of the normal modes. In spite of this, the technique requires a great deal of preliminary work before the problem is ready for the hybrid computer. The selection of the modes is not easy and is usually subjective.

Nelson (16) discussed the possible application of invariant imbedding to the CSDT hybrid method. Invariant imbedding is essentially a method for converting a two-point boundary value problem for a linear system of ordinary differential equations to an equivalent initial-value problem for an associated non-linear system or ordinary differential equations. Following the usual derivation which applied the CSDT method to the time-dependent heat diffusion equation Nelson concluded that the method can be applied, at least in principle, to any linear partial differential equation, provided the associated boundary conditions in the continuous variable can be put in the linear inhomogeneous form.

2.7 Classical CSDT Hybrid Simulation of An Isothermal Tubular Flow Reactor with Axial Diffusion

In order to illustrate the inherent instability problem of the classical CSDT technique discussed in the previous section, let us consider our basic problem.

$$\frac{\partial^2 y}{\partial z^2} - 2U\frac{\partial y}{\partial z} - UR_n y = \frac{1}{\alpha}\frac{\partial y}{\partial t} \tag{2.41}$$

where $\frac{1}{\alpha} = \frac{L^2}{\bar{D}T}$

Applying the classical CSDT approximation and reversing the integration direction leads to the following difference-differential expression:

$$\frac{d^2 y^i}{dz^2} \cong -2U\frac{dy^i}{dz} + UR_n y^i + \frac{y^i - y^{i-1}}{\alpha\Delta t} \tag{2.42}$$

with

$$y^i = y(z,t^i); \quad t^i = i\Delta t$$

Dropping the iteration index i from Equation (2.42) for the current values and rewriting, we have

$$\ddot{y} = -2U\dot{y} + UR_n y + \frac{1}{\alpha\Delta t}[y - y^{i-1}] \tag{2.43}$$

with the following initial and boundary conditions.

$$y^0(z) \equiv 0 \quad , \quad 0 \leq z \leq 1$$

$$\left.\begin{aligned} [\dot{y}]_{z=0} &= 2U[(y)_{z=0} - 1] \\ [\dot{y}]_{z=1} &= 0 \end{aligned}\right\} \quad t > 0 \tag{2.44}$$

The details of the hybrid programming procedure are discussed in Appendix D. For simplicity, since our purpose here is to illustrate the difficulties involved in the classical CSDT approach, let

$$U = 1 \quad \text{and}$$

$$R_n = 1$$

The instability problem which is directly related to the magnitude of $\alpha\Delta t$, i.e., the loop gain, is examined by adjusting the potentiometer setting for the various $\alpha\Delta t$ values.

TABLE 2.7.1 Potentiometer Settings

Pot-4	corresponding $\alpha\Delta t$ value	stable results obtained?
0.1	$\frac{1}{4}$ (=0.25)	yes
0.3	$\frac{1}{12}$ (= 0.0835)	partially
0.5	0.05	partially
0.6	$\frac{1}{24}$ (= 0.0417)	no
0.7	$\frac{1}{28}$ (= 0.0358)	no
0.8	$\frac{1}{32}$ (= 0.0312)	no
0.9	$\frac{1}{36}$ (= 0.0278)	no
1.0 (0.9999)	0.025	no

As the magnitude of $\alpha\Delta t$ becomes smaller the condition for the solution to be converged gets worse, and finally it becomes impossible to attain a stable result. Even using the smallest possible IC change within the system $(\frac{1}{81.91}$ volts) an enormous difference can be observed in the resulting terminal values. This typical problem is illustrated in Figure 2.7 for backward integration in z.

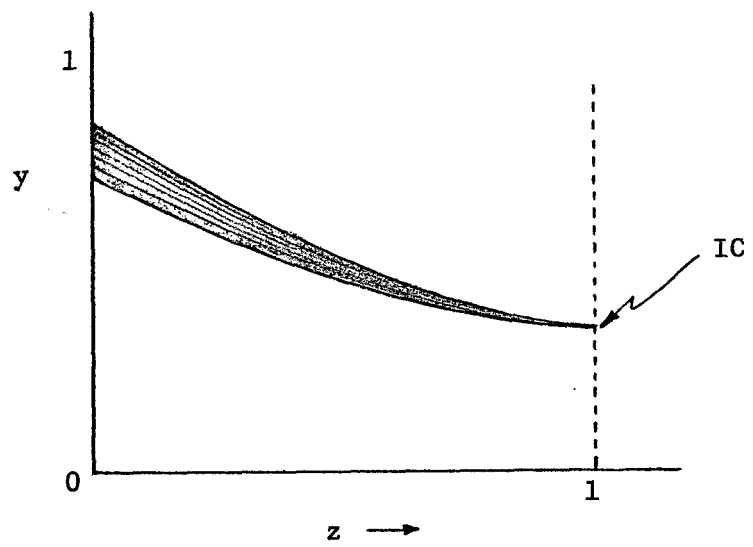


Figure 2.7 Typical Unstable Results of the Classical CSDT Approach

Table 2.7.2 exhibits another example of the major difficulties associated with the classical CSDT approach. In fact, as shown in Figure 2.7, to obtain a stable result is entirely out of the question in the classical CSDT when using $\alpha\Delta t = 0.025$.

TABLE 2.7.2 Comparison of the Transient Results at the End of the First Time Increment (Δt) for $\alpha\Delta t = 0.025$

Axial position	$y(z,\Delta t)$	
(z = 0 to 1)	Exact CSDT **	Classical CSDT *
1	0.2674	0.3122
2	0.2147	0.2539
3	0.1725	0.2039
4	0.1385	0.1637
5	0.1112	0.1315
6	0.0893	0.1056
7	0.0718	0.0846
8	0.0576	0.0678
9	0.0463	0.0543
10	0.0372	0.0436
11	0.0299	0.0352
12	0.0240	0.0283
13	0.0193	0.0227
14	0.0155	0.0182
15	0.0125	0.0147
16	0.0100	0.0155
17	0.0081	0.0092
18	0.0065	0.0073
19	0.0053	0.0059
20	0.0043	0.0046
21	0.0036	0.0038
22	0.0030	0.0031
23	0.0026	0.0026
24	0.0023	0.0022
25	0.0022	0.0021

* unstable results, terminated at the 21st iteration.

** analytic solution may be obtained by eliminating y^{i-1} from equation (2.43) ($y^{i-1} \equiv 0$ for $i = 1$)

CHAPTER 3

SERIAL HYBRID DECOMPOSITION METHOD IN ONE-SPACE DIMENSION AND TIME

3.1 Mathematical Development of Serial Hybrid Decomposition Method

The method of decomposition allows one to replace the usual iteration schemes with a one-pass scheme at a cost of augmenting the system with additional equations. Vichnevetsky (23-30) was the first to point out its advantages for hybrid computation.

To illustrate the application of the decomposition technique for the hybrid solution of the unsteady-state problem, again consider the following diffusion equation:

$$\frac{\partial^2 y}{\partial z^2} - 2U \frac{\partial y}{\partial z} - UR_n y = \frac{1}{\alpha} \frac{\partial y}{\partial t} \tag{3.1}$$

for $z \in (0,1)$

with the initial and boundary conditions:

$$y(z,0) = Y_z(0)$$

$$\left.\begin{aligned} \left(\frac{dy}{dz}\right)_{z=0} &= 2U[y(0,t) - 1] \\ \left(\frac{dy}{dz}\right)_{z=1} &= 0 \end{aligned}\right\} \quad t > 0 \tag{3.2}$$

The hybrid CSDT method of approximation consists in expressing the solution along equi-distant lines parallel to the z axis in the (z,t) plane. If we denote $y^i(z)$ the value of $y(z,t)$ at time $t^i = i \Delta t$, then Equation (3.1) can be approximated by

$$\theta \left[\frac{d^2y^{i+1}}{dz^2} - 2U \frac{dy^{i+1}}{dz} - UR_n y^{i+1} \right] + (1-\theta) \left[\frac{d^2y^i}{dz^2} - 2U \frac{dy^i}{dz} - UR_n y^i \right] = \frac{y^{i+1} - y^i}{\alpha \Delta t} \tag{3.3}$$

where θ is a constant which must be chosen in the interval $1/2 < \theta \leq 1$ to ensure stability in the time marching process. (29)

To produce the equation for y^{i+1}:

$$\frac{d^2y^{i+1}}{dz^2} - 2U \frac{dy^{i+1}}{dz} - \left(UR_n + \frac{1}{\alpha\theta \Delta t}\right) y^{i+1} = -\frac{y^i}{\alpha\theta \Delta t} - \frac{(1-\theta)}{\theta} \left[\frac{d^2y^i}{dz^2} - 2U \frac{dy^i}{dz} - UR_n y^i \right] \tag{3.4}$$

which can be rewritten:

$$\frac{d^2y^{i+1}}{dz^2} - 2U \frac{dy^{i+1}}{dz} - \left(UR_n + \frac{1}{\alpha\theta \Delta t}\right) y^{i+1} = R^i \tag{3.5}$$

where

$$R^i = -\frac{y^i}{\alpha\theta \Delta t} - \frac{(1-\theta)}{\theta} \left[\frac{d^2y^i}{dz^2} - 2U \frac{dy^i}{dz} - UR_n y^i \right] \tag{3.6}$$

Equation (3.5) is an ordinary differential equation in the independent variable z. The new value R^{i+1} to be stored at time t^{i+1} is computed by a recursive relationship, since from equation (3.6)

$$R^{i+1} = -\frac{y^{i+1}}{\alpha\theta \Delta t} - \frac{(1-\theta)}{\theta} \left[\frac{d^2y^{i+1}}{dz^2} - 2U \frac{dy^{i+1}}{dz} - UR_n y^{i+1} \right].$$

or

$$R^{i+1} = -\frac{y^{i+1}}{\alpha\theta\ \Delta\ t} - \frac{(1-\theta)}{\theta}\left[\frac{d^2y^{i+1}}{dz^2} - 2U\frac{dy^{i+1}}{dz} - (UR_n + \frac{1}{\alpha\theta\ \Delta\ t})y^{i+1}\right]$$
$$-\frac{(1-\theta)}{\theta}\frac{y^{i+1}}{\alpha\theta\ \Delta\ t}$$

taking equation (3.5) into account.

$$R^{i+1} = -\frac{1}{\theta}\left[\frac{y^{i+1}}{\alpha\theta\ \Delta\ t} + (1-\theta)\ R^i\right] \tag{3.7}$$

An alternative expression is:

$$R^{i+1} = R^i - \frac{1}{\theta}(R^i + \frac{y^{i+1}}{\alpha\theta\ \Delta\ t}) \tag{3.8}$$

Equation (3.5) is of the second order in Z, its characteristic equation is:

$$\beta^2 - 2U\beta - (UR_n + \frac{1}{\alpha\theta\ \Delta\ t}) = 0 \tag{3.9}$$

or

$$\beta = U \pm \sqrt{U^2 + UR_n + \frac{1}{\alpha\theta\ \Delta\ t}} \tag{3.10}$$

These two βs are real and of opposite sign. It was pointed out by Vichnevetsky (23) that the direct integration of Equation (3.5) as an initial value problem of the second order could result in unstable error propagation problems.

To apply the method of Decomposition let

$$L(\ ) = \frac{d^2}{dz^2} - 2U\frac{d}{dz} - [UR_n + \frac{1}{\alpha\theta\ \Delta\ t}] \tag{3.11}$$

be the second order operator which can be decomposed into the product of

two first order operators, L_B and L_F, which are intended to yield stable integrations in the backward and forward directions, respectively. We can write

$$\left.\begin{aligned} L(\ ) &= L_B(\ ) \cdot L_F(\ ) \\ L_B(\ ) &= \frac{d}{dz} - \lambda_B \\ L_F(\ ) &= \frac{d}{dz} - \lambda_F \end{aligned}\right\} \tag{3.12}$$

By identification we obtain

$$\left.\begin{aligned} \lambda_B + \lambda_F &= 2U \\ \text{and} \quad \lambda_B \lambda_F &= -\left[UR_n + \frac{1}{\alpha\theta\ \Delta\ t}\right] \end{aligned}\right\} \tag{3.13}$$

Conditions for the stable integration in these respective directions are $\lambda_B \geq 0$ and $\lambda_F \leq 0$, which yield

$$\left.\begin{aligned} \lambda_B &= U + \sqrt{U^2 + UR_n + \frac{1}{\alpha\theta\ \Delta\ t}} \\ \text{and} \quad \lambda_F &= U - \sqrt{U^2 + UR_n + \frac{1}{\alpha\theta\ \Delta\ t}} \end{aligned}\right\} \tag{3.14}$$

Now, to obtain the homogeneous solutions, any function $y_1(z)$ which is a solution of

$$\frac{dy_1}{dz} - \lambda_B y_1 = 0 \tag{3.15}$$

is also a solution of $L(y_1) = 0$, since

$$L(y_1) = \left(\frac{d}{dz} - \lambda_F\right) \cdot \left(\frac{d}{dz} - \lambda_B\right) \cdot y_1$$
$$= \left(\frac{d}{dz} - \lambda_F\right) \cdot 0 = 0 \tag{3.16}$$

The integration of (3.15) requires only one boundary value at $z=1$. Similarly, any solution of

$$\frac{dy_2}{dz} - \lambda_F y_2 = 0 \tag{3.17}$$

is also a solution of

$$L(y_2) = 0 \tag{3.18}$$

And the integration of (3.17) requires only one boundary value at $z=0$. We can choose $y_1(1) = 1$ and $y_2(0) = 1$ for convenience.

Finally, to obtain the particular solution of equation (3.5), if $x^{i+1}(z)$ is a solution of:

$$\frac{dx^{i+1}}{dz} - \lambda_B x^{i+1} = R^i \tag{3.19}$$

and $y_3^{i+1}(z)$ is a solution of:

$$\frac{dy_3^{i+1}}{dz} - \lambda_F y_3^{i+1} = x^{i+1} \tag{3.20}$$

then we have

$$L(y_3^{i+1}) = \left(\frac{d}{dz} - \lambda_B\right) \left(\frac{d}{dz} - \lambda_F\right) y_3^{i+1}$$
$$= \left(\frac{d}{dz} - \lambda_B\right) x^{i+1} = R^i$$

Thus, we have

$$\frac{d^2 y_3^{i+1}}{dz^2} - 2U \frac{dy_3^{i+1}}{dz} - \left(UR_n + \frac{1}{\alpha\theta \, \Delta t} \right) y_3^{i+1} = R^i \tag{3.21}$$

This holds for any boundary conditions chosen to integrate equations (3.19) and (3.20). For simplicity, the associated boundary conditions can be chosen to be:

$$\left.\begin{aligned} & x^{i+1}(1) = 0 \\ \text{and} \quad & \\ & y_3^{i+1}(0) = 0 \end{aligned}\right\} \tag{3.22}$$

Now we can observe that any linear combination

$$y^{i+1} = a^{i+1} y_1 + b^{i+1} y_2 + y_3^{i+1} \tag{3.23}$$

satisfies equation (3.5). Once $y_1(z)$, $y_2(z)$ and $y_3^{i+1}(z)$ are known, the two constants a^{i+1} and b^{i+1} can always be found so that equation (3.23) satisfies the boundary conditions of the original problem, equation (3.2). This formulates the general solution to the problem.

3.2 Hybrid Computer Set-Up

The hybrid computer block diagram for the procedure described in the previous section is shown in Figure 3.1.

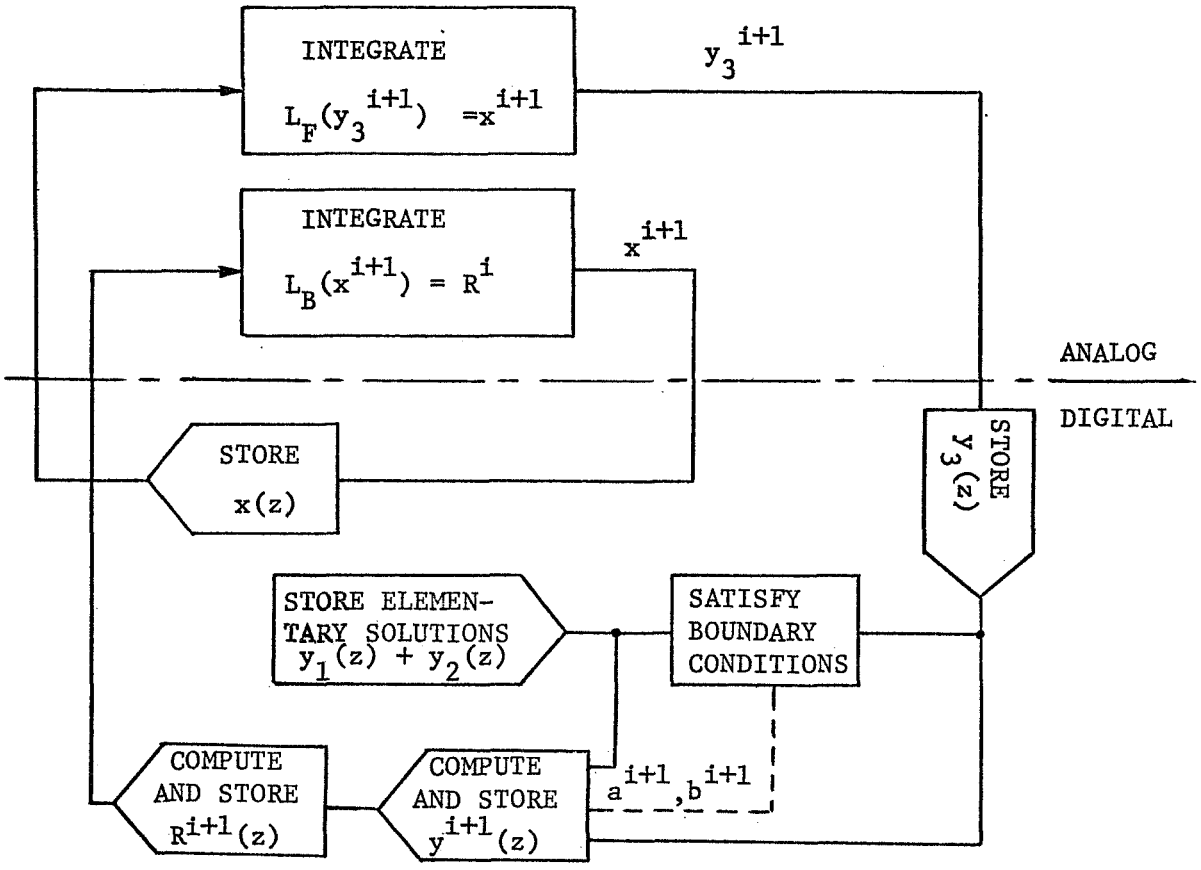


Figure 3.1 Hybrid Computing Circuit Diagram for The Serial-Decomposition Method

The overall integration procedure can be summarized as follows:

1) By integration of linear homogeneous equations (3.15) and (3.17) analytically and computing values digitally, obtain and store the time independent elementary solutions $y_1(z)$ and $y_2(z)$.

2) Integrate the two stable differential equations (3.19) and (3.20) with boundary conditions (3.22) on the analog computer.

3) Compute a^{i+1} and b^{i+1} by the application of the boundary conditions (3.2) on the digital computer.

4) Compute the solution $y^{i+1}(z)$ by the application of (3.23) on the analog computer.

5) Compute $R^{i+1}(z)$ by the application of (3.8) on the digital computer.

6) Return to step 2 for the next time increment.

The procedure is summarized in the following diagram:

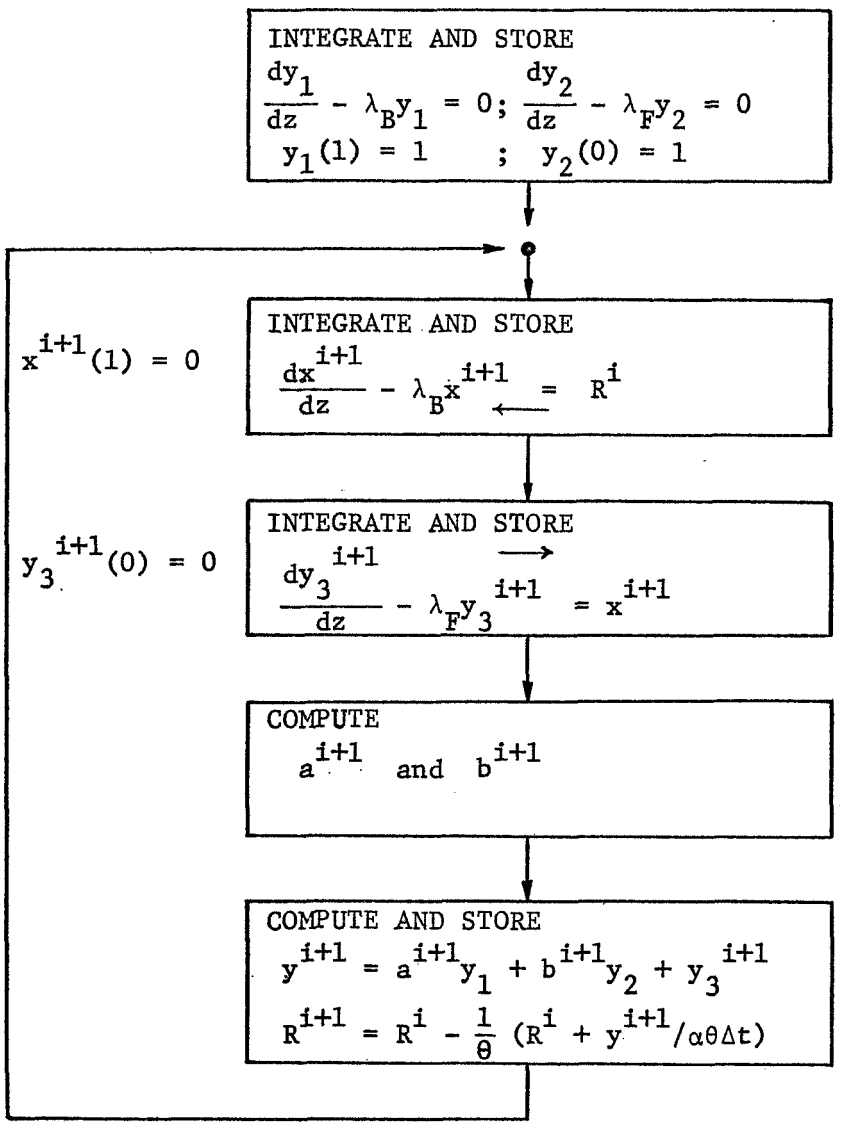


Figure 3.2 Sequence of Operations

3.3 Simulation of An Isothermal Tubular Reactor with Axial Diffusion

It becomes apparent that the classical CSDT method will not yield a stable solution when time increment reaches a certain lower limit. This is due to the fact that such a mechanization always leads to a very high positive gain in the integrator feedback loop.

By using the same case study from Section 2.7, the method of decomposition follows:

The two homogeneous solutions as described by Equations (3.15) and (3.17) are easily obtained digitally from the following expressions;

$$y_1(z) = 1/\exp[\lambda_1(1-z)] \tag{3.24}$$

$$y_2(z) = 1/\exp(\lambda_2 z) \tag{3.25}$$

which satisfy the required initial conditions

$$y_1(1) = 1 \quad \text{and} \quad y_2(0) = 1$$

Since they are time independent, the solutions are only computed once and stored for the later usage at all time steps.

The particular solution is arrived by integrating Equations (3.19) a and (3.20) successively. The actual mechanization of these two equations is described in Appendix E.

The starting value of R^i is directly obtained from the initial value when θ is set to 1.

Finally, to obtain the general solution

$$y = a\, y_1 + b\, y_2 + y_3 \tag{3.26}$$

the two constants a and b must be solved by applying the boundary conditions, since

$$\frac{dy}{dz} = a \frac{dy_1}{dz} + b \frac{dy_2}{dz} + \frac{dy_3}{dz} \tag{3.27}$$

and

$$\frac{dy_1}{dz} = \lambda_B y_1$$

$$\frac{dy_2}{dz} = \lambda_F y_2$$

$$\frac{dy_3}{dz} = \lambda_F y_3 + x$$

Substituting the above expressions into Equation (2.44) we obtain the following expressions:

$$a = \frac{-\lambda_F[b\ y_2(1) + y_3(1)]}{\lambda_B} \tag{3.28}$$

and

$$b = \frac{\lambda_B[x(0) + 2U] - y_1(0)(\lambda_B - 2U)[\lambda_F y_3(1)]}{y_1(0)\ y_2(1)\ \lambda_F(\lambda_B - 2U) - \lambda_B(\lambda_F - 2U)} \tag{3.29}$$

Digital portion of the program using DAMPS and Hybrid Executive is listed in Appendix E.

Table 3.3 is a comparison of the results from the classical method and the decomposition method for $\alpha \Delta t = 0.025$, which was far beyond the stability range of the classical method. Note that the result of the classical method is not the final converged result (since it is impossible to attain a stable solution), but rather, it is the result at the

end of the 21st iteration. From this table the significantly more stable nature of the serial-decomposition solution is clearly illustrated.

Table 3.3 Comparison of the Results ($\alpha\Delta t = 0.025$)

Time Step	y(0)		y(1)	
i	Classical	Decomposition	Classical	Decomposition
1	0.3122	0.2674	0.0021	0.0021
2	0.3930 *	0.3673	0.0104	0.0048
3	0.3901 *	0.4324	0.0278	0.0129
4	0.4977	0.4809	0.0509	0.0268
5	0.5831	0.5321	0.0812	0.0564
6	0.6766	0.5644	0.1178	0.0840
7	0.7043 *	0.5887	0.1596	0.1010
8	0.6796 *	0.6113	0.2035	0.1221
9	1.0000 **	0.6421	0.2487	0.1566

* erratic results
** amplifier overload

where $y(0)$ = concentration at $z = 0^+$

$y(1)$ = concentration at $z = 1$

In principle, the analytic solution can be obtained by means of the Laplace transformation satisfying the specified boundary conditions, then, carry out the inverse transformation using the Heaviside partial fraction theorem. However, the mathematical derivations involved in this operation are extremely complex even in the simplest linear case of equation. The analytic solution of equation (3.1) with the corresponding boundary condition (3.2) was given by Yaws (33), who integrated the inverse of the position transfer function (33) by introducing the result obtained from the pulse response and yielded the following general expression:

$$y(z,\theta) = \exp\left[\frac{Pe}{2} z\right] \sum_{n=1}^{\infty} \{A_n(z)/a_n\}$$

$$- \exp\left[\frac{Pe}{2} z\right] \sum_{n=1}^{\infty} \{A_n(z)/a_n\} \exp[-a_n\theta] \qquad (3.30)$$

where

$$a_n = \beta_n/Pe + Pe/4 + R_n/2$$

$$A_n(z) = \frac{\beta_n(2\beta_n \text{ Cos } \beta_n z + Pe \text{ Sin } \beta_n z)}{\beta_n^2 + Pe^2/4 + Pe}$$

β_n = nth positive root to the transcendental equation

$$\text{Tan}(\beta_n) = \frac{\beta_n Pe}{\beta_n^2 - Pe^2/4} \quad , \qquad n = 1,2, \ldots.$$

$$Pe = \frac{1}{2U}$$

$$\theta = 2\alpha Ut$$

The convergence of the series is known to be very slow for large Pe values. Instead, the digital computer solution was obtained using the implicit finite difference approach (using 50 mesh points in the space coordinate) to compare the result from the serial decomposition method. The results shown in Figures 3.3.1 and 3.3.2 agree fairly well. In fact, the steady-state solution from the digital computer finite difference method (see Table E-1) does not match exactly to the analytic solution due to the errors resulted from the finite difference approximations to the differential equation.

Additional details of the decomposition technique are shown in Appendix E.

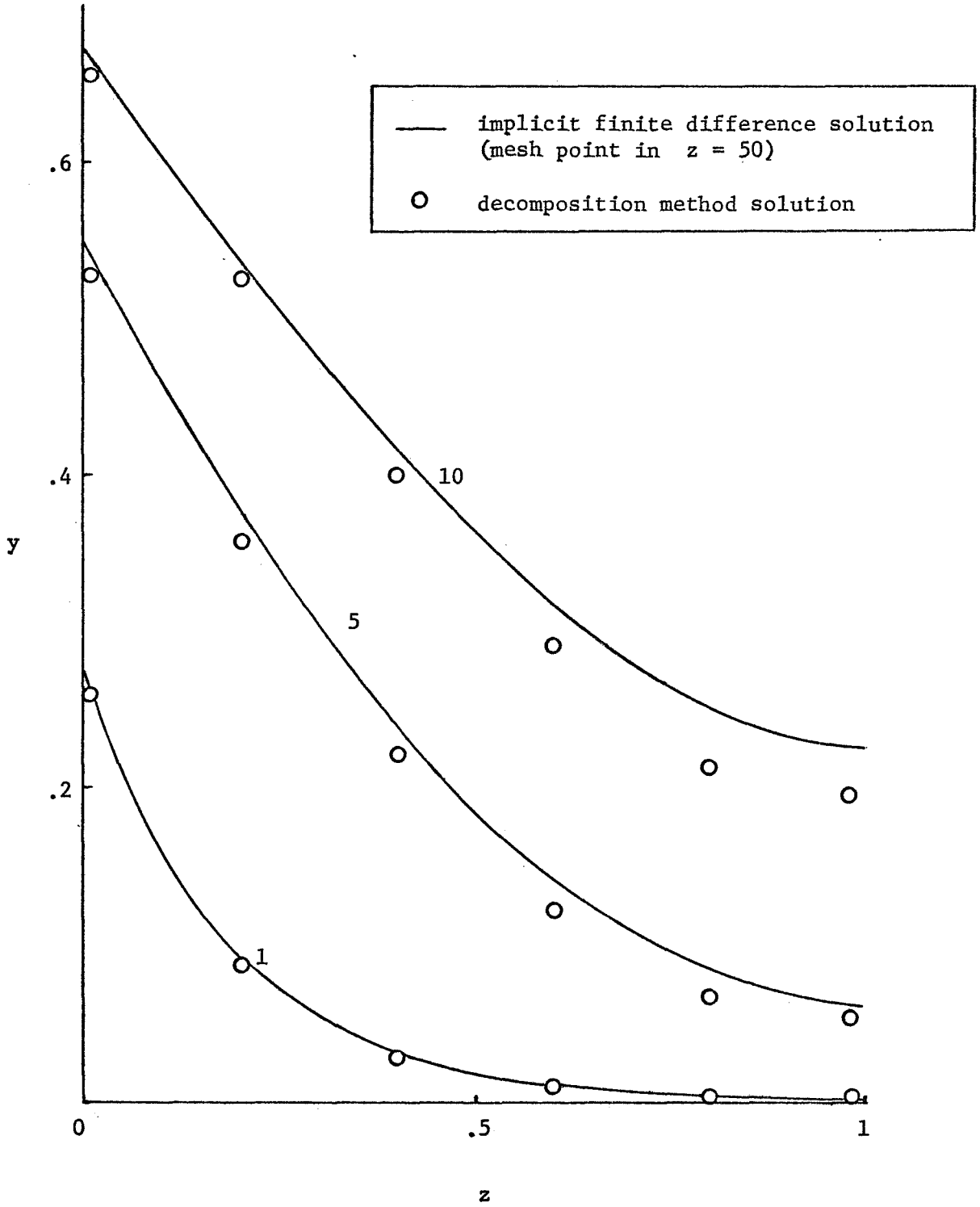


Figure 3.3.1 Comparison of the Transient Solutions for $\alpha\Delta t = 0.025$, parameter = time step

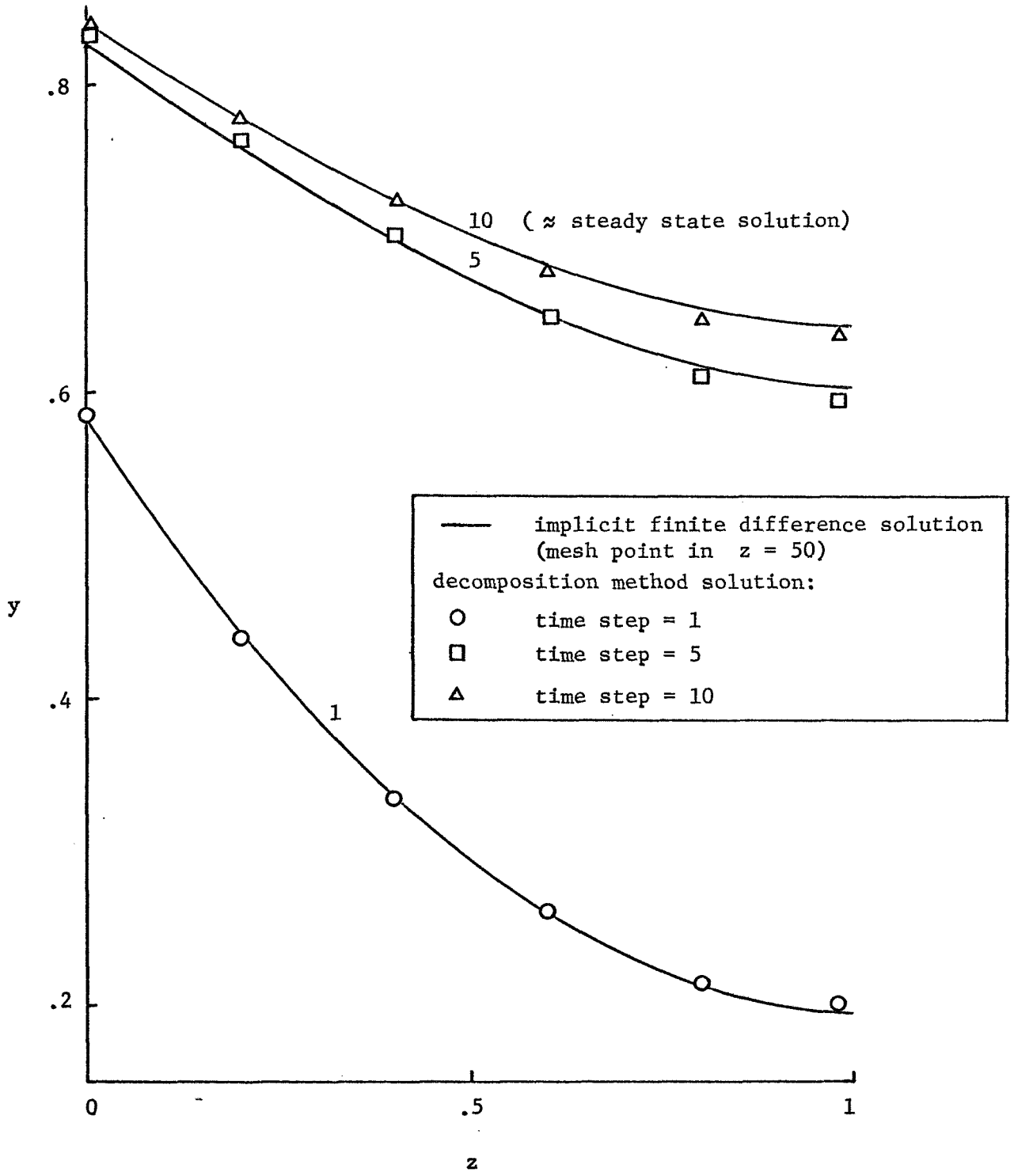


Figure 3.3.2 Comparison of the Transient Solutions for $\alpha\Delta t = 0.25$, parameter = time step

CHAPTER 4

SERIAL HYBRID DECOMPOSITION METHOD IN TWO-SPACE DIMENSION AND TIME

41. Mathematical Development

To extend the decomposition method into an additional space domain, the related system equations require some basic modifications in order to apply the existing serial hybrid techniques. Since in the serial approach only one independent space variable is considered in the entire computation process, it seems most natural to follow some of the well established numerical methods for solving the multidimensional problems. Among them, the method of lines and the alternating direction implicit method (18) perhaps are the two most promising techniques applicable to the serial solution of the multidimensional problems. When these approaches are followed it will normally result in either an explicit/implicit or the all implicit formulation of the original problem. Hara and Karplus (9) proposed a possible extension to their technique; a combination of the alternating direction implicit method and the functional optimization technique to obtain the serial solution of the two-space dimension problem. Judging from the complications involved in the one dimensional problem by the functional optimization, the practical application into the two dimensional case may be very limited. Vichnevetsky and Tomalesky (29,30) discussed various approaches in conjunction with the serial decomposition method for the two space dimension problem.

In this section we present two basic approaches using the serial decomposition method; a CSCSDT method and a CSDSDT method (29).

4.1.1 Alternating Direction Implicit Formulation by A Continuous Space, Continuous Space, Discrete Time Method (CSCSDT)

The serial decompositon method and the Peaceman-Rachford Method (18) (known as the alternating direction implicit method) can be combined to solve two dimensional unsteady state problems. As an illustration, consider the following simple equation with a constant diffusivity:

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} = \frac{1}{\alpha} \frac{\partial u(x,y,t)}{\partial t} \tag{4.1}$$

where

$x \in (0,X)$

$y \in (0,Y)$

$t \in (0,T)$

and with certain initial and boundary conditions which will be omitted from the present discussion since our purpose here is primarily to introduce the combined technique. The basic process in passing from t^i to t^{i+1} has two parts; each part involving the solution of a set of ordinary differential equations by the CSDT decomposition technique.

For the first part, integrating in the x direction along the lines of constant y's from t^i to $t^{i+1/2}$, the CSDT representation of equation (4.1) is

$$\frac{d^2 u_m^{i+1/2}(x)}{dx^2} - \frac{u_m^{i+1/2}(x)}{\alpha \frac{\Delta t}{2}} = -\frac{u_m^{i}(x)}{\alpha \frac{\Delta t}{2}} - \frac{d^2 u_m^{i}(x)}{dy^2} \tag{4.2}$$

with $m = 1,2,\ldots,M-1$ (note that $y = 0$, $y = y_M$ are boundaries)

where

$$u_m^{\,i}(x) = u(x,y_m,t^i)$$

Equation (4.2) can be rewritten as

$$\frac{d^2u_m^{\,i+1/2}}{dx^2} - \frac{u_m^{\,i+1/2}}{\alpha\,\frac{\Delta t}{2}} = R_m^i \tag{4.3}$$

where

$$R_m^i = -\frac{u_m^i}{\alpha\,\frac{\Delta t}{2}} - \frac{d^2u_m^{\,i}}{dy^2} \tag{4.4}$$

The serial decomposition method described in Chapter 3 can be applied to solve equation (4.3) to obtain $u_m^{\,i+1/2}(x)$.

The second part will consist of finding $u_\ell^{\,i+1}(y)$ along the lines $x = x_\ell$; $\ell = 1,2,\ldots, L-1$ (where $x = 0$ and $x = x_L$ are boundaries, $u_\ell^{\,i+1}(y) = u(x_\ell,y,t^{i+1})$).

We now alternate the directions of equation (4.2) and obtain

$$\frac{d^2u_\ell^{\,i+1}(y)}{dy^2} - \frac{u_\ell^{\,i+1}(y)}{\alpha\,\frac{\Delta t}{2}} = -\frac{u_\ell^{\,i+1/2}(y)}{\alpha\,\frac{\Delta t}{2}} - \frac{d^2u_\ell^{\,i+1/2}(y)}{dx^2} \tag{4.5}$$

or rewriting the above equation:

$$\frac{d^2u_\ell^{\,i+1}}{dy^2} - \frac{u_\ell^{\,i+1}}{\alpha\,\frac{\Delta t}{2}} = R_\ell^{\,i+1/2} \tag{4.6}$$

where

$$R_\ell^{i+1/2} = -\frac{u_\ell^{i+1/2}}{\alpha \frac{\Delta t}{2}} - \frac{d^2 u_\ell^{i+1/2}}{dx^2} \tag{4.7}$$

By comparing equations (4.3) and (4.7) where m and ℓ represent the same position on the $(M+1) \times (L+1)$ grid at $t^{i+1/2}$

$$R_\ell^{i+1/2} = -\left(\frac{d^2 u_m^{i+1/2}}{dx^2} - \frac{u_m^{i+1/2}}{\alpha \frac{\Delta t}{2}}\right) - \frac{2u_m^{i+1/2}}{\alpha \frac{\Delta t}{2}}$$

or

$$R_\ell^{i+1/2} = -R_m^i - \frac{2u_m^{i+1/2}}{\alpha \frac{\Delta t}{2}} \tag{4.8}$$

Knowing $R_\ell^{i+1/2}$, equation (4.6) can be solved by the decomposition method to obtain $u_\ell^{i+1}(y)$ over $\ell = 1,2,\ldots, L-1$. This completes the second part of the alternating direction process and also one time step of the CSDT procedure.

We can conclude from the above derivations that the following general recursive relations exist in this alternating-direction marching process:

$$\left.\begin{aligned} R_m^i &= -R_\ell^{i-1/2} - \frac{2u_\ell^i}{\alpha \frac{\Delta t}{2}} \\ &\text{and} \\ R_\ell^{i+1/2} &= -R_m^i - \frac{2u_m^{i+1/2}}{\alpha \frac{\Delta t}{2}} \end{aligned}\right\} \tag{4.9}$$

Or more general if we drop m and ℓ from Rs

$$R^{i} = -R^{i-1/2} - \frac{2u^{i}}{\alpha \frac{\Delta t}{2}} \tag{4.10}$$

with $i = 1/2, 1, 3/2, 2, \ldots, \frac{T}{\Delta t}$.

The limitations of this method are discussed in the example at the end of this chapter.

4.1.2 Explicit/Implicit Formulation (29,30) by a Continuous Space, Discrete Space, Discrete Time Method (CSDSDT)

Another approach to the two-space dimension problem is by discretizing one of the space variables. Then, the two-space dimension problem is reduced to a set of one space dimension and time problems by considering N adjacent spacelines covering the discretized space domain. This approach is first applied by Vichnevetsky and Tomalesky (29,30) with the decomposition method and is known to the usual numerical methodology as the difference-differential technique or the method of lines which is more familiar in the Russian literature.

If we now discretize the y and t domains and integrating continuously along x, the two space dimension problem described by Equation (4.1) may be approximated by:

$$\frac{d^2 u_n^{i+1}}{dx^2} = -\frac{u_{n+1}^{i} - 2u_n^{i} + u_{n-1}^{i}}{\Delta y^2} + \frac{u_n^{i+1} - u_n^{i}}{\alpha \Delta t} \tag{4.11}$$

for $n = 1, 2, \ldots, N$ (spacelines in y domain).

From Equation (4.11) it is clear that this explicit/implicit representation of the approximation permits an immediate application of the decomposition method independently to the N equations described by (4.11).

4.2 Simulation of An Unsteady-State Isothermal Tubular Reactor

Equation (2.1) may be modified to yield the following material balance equation of an isothermal tubular reactor in the cylindrical coordinates with axial symmetry.

$$D_x \frac{\partial^2 c}{\partial x^2} + D_y \left[\frac{\partial^2 c}{\partial y^2} + \frac{1}{y} \frac{\partial c}{\partial y} \right] - u \frac{\partial c}{\partial y} - K_c c^n = \frac{\partial c}{\partial \theta} \qquad (4.12)$$

where

$x \in (0,L)$; longitudinal directive

$y \in (0,R)$; radial directive

$\theta \in (0,T)$; time

Let

$$z = x/L, \quad r = y/R, \quad t = \theta/T \quad \text{and} \quad f = c/c_o$$

After making variable substitutions with $n = 1$, equation (4.12) is transformed into:

$$\frac{1}{P_{e_z}} \frac{\partial^2 f}{\partial z^2} + \frac{1}{P_{e_r}} \left[\frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} \right] - \frac{\partial f}{\partial z} - \beta f = \delta \frac{\partial f}{\partial t} \qquad (4.13)$$

where $P_{e_z} = \frac{Lu}{D_x}$; axial Peclet number

$P_{e_r} = \frac{Ru}{D_y} \left(\frac{R}{L}\right)$; radial Peclet number

$\beta = \frac{LKc}{u}$

$\delta = \frac{L}{uT}$

with

$$z \in (o,1), \quad r \in (0,1) \quad \text{and} \quad f \in (0,1)$$

and

$$f = f(z,r,t)$$

Equation (4.13) is a second order equation in both z and r and first order in time t. The appropriate initial and boundary conditions are given by

$$\left.\begin{array}{l} \text{For} \quad t \leq 0, \quad f(z,r,t) = 0 \\ \text{For} \quad t > 0 \\ \qquad f - \dfrac{1}{P_{e_z}} \dfrac{df}{dz} = 1, \quad z = 0, \quad r \neq 1 \\ \qquad f(0,1,t) = 1 \\ \qquad \dfrac{df}{dz} = 0, \quad z = 1 \\ \qquad \dfrac{df}{dr} = 0, \quad r = 0 \text{ and } r = 1 \end{array}\right\} \qquad (4.14)$$

Unfortunately, application of the all implicit CSCSDT method to this problem introduces a basic difficulty which makes this method unable to satisfy the radial boundary conditions imposed by Equation (4.14). This subject matter will be further discussed in Appendix H.

By applying the CSDSDT approach, if we consider N adjacent straight lines covering the radius of the reactor tube and let $f_n(z,t)$ represent the reactant concentration along the nth line, then equation (4.13) may be approximated by:

$$\frac{1}{P_{e_z}} \frac{\partial^2 f_n}{\partial z^2} - \frac{\partial f_n}{\partial z} - \beta f_n + \frac{1}{P_{e_r}} \left[\frac{f_{n+1} - 2f_n + f_{n-1}}{\Delta r^2} + \frac{f_{n+1} - f_{n-1}}{2n\Delta r^2} \right]$$

$$= \delta \frac{\partial f_n}{\partial t}$$

or

$$\frac{1}{P_{e_z}} \frac{\partial^2 f_n}{\partial z^2} - \frac{\partial f_n}{\partial z} - \beta f_n + \left[P\, f_{n+1} + Q\, f_n + T\, f_{n-1} \right] = \delta \frac{\partial f_n}{\partial t} \qquad (4.15)$$

where

$$P = \frac{2n+1}{P_{e_r}(2n\Delta r^2)} \; ; \; Q = -\frac{2}{P_{e_r}\Delta r^2} \; ; \quad T = \frac{2n-1}{P_{e_r}(2n\Delta r^2)}$$

If we now apply discrete time approximation to this equation with implicit formulation in the z direction, we obtain

$$\frac{1}{P_{e_z}} \frac{d^2 f_n^{\,i+1}}{dz^2} - \frac{df_n^{\,i+1}}{dz} - (\beta + \frac{\delta}{\Delta t})\, f_n^{\,i+1} = -\, Pf_{n+1}^i - (Q + \frac{\delta}{\Delta t})\, f_n^i - Tf_{n-1}^i$$

The above equation represents an implicit/explicit approximation in z and r directions respectively, or we can rewrite in the form

$$\frac{d^2 f_n^{\,i+1}}{dz^2} - P_{e_z} \frac{df_n^{\,i+1}}{dz} - P_{e_z} (\beta + \frac{\delta}{\Delta t})\, f_n^{\,i+1} = R_n^i \qquad (4.16)$$

where $\quad R_n^i = -P_{e_z} [Pf_{n+1}^i + (Q + \frac{\delta}{\Delta t})\, f_n^i + Tf_{n-1}^i]$

with

$$R_n^0 \equiv 0$$

Since P_{e_z} is space dependent in the actual sense the second order operator

$$L(\ ) = \frac{d^2}{dz^2} - P_{e_z}\frac{d}{dz} - P_{e_z}(\beta + \frac{\delta}{\Delta t}) \tag{4.17}$$

is decomposed into the product of two first order operators

$$\left.\begin{aligned} L &= L_B \cdot L_F \\ L_B &= \frac{d}{dz} - \lambda_B(z) \\ L_F &= \frac{d}{dz} - \lambda_F(z) \end{aligned}\right\} \tag{4.18}$$

conditions for the stable integration in these respective directions are $\lambda_B \geq 0$ and $\lambda_F \leq 0$. By identifying (4.17) and (4.18) yields additional conditions on λ_F and λ_B:

$$\begin{aligned} L &= \frac{d^2}{dz^2} - P_{e_z}\frac{d}{dz} - P_{e_z}(\beta + \frac{\delta}{\Delta t}) \\ &= \left(\frac{d}{dz} - \lambda_B\right) \cdot \left(\frac{d}{dz} - \lambda_F\right) \\ &= \frac{d^2}{dz^2} - (\lambda_B + \lambda_F)\frac{d}{dz} - \frac{d\lambda_F}{dz} + \lambda_F\lambda_B \end{aligned} \tag{4.19}$$

or $$\lambda_B + \lambda_F = P_{e_z} \tag{4.20}$$

and $$\frac{d\lambda_F}{dz} - \lambda_F\lambda_B = P_{e_z}(\beta + \frac{\delta}{\Delta t}) \tag{4.21}$$

which becomes

$$\lambda_B = P_{e_z} - \lambda_F \tag{4.22}$$

$$\frac{d\lambda_F}{dz} - \lambda_F(P_{e_z} - \lambda_F) = P_{e_z}(\beta + \frac{\delta}{\Delta t}) \tag{4.23}$$

This latter equation (or sometimes called the Ricatti equation (29)) is stable when integrated in the reverse $(z = 1 \text{ to } 0)$ direction, with any in initial condition satisfying: $\lambda_F(1) \leq 0$.

If P_{e_z} is constant, we can solve λ_F and λ_B directly

$$\lambda_B + \lambda_F = P_{e_z}$$

$$\lambda_F\lambda_B = -P_{e_z}(\beta + \frac{\delta}{\Delta t}) \tag{4.24}$$

then

$$\lambda_F = \frac{P_{e_z} - \sqrt{P_{e_z}^{\ 2} + 4P_{e_z}(\beta + \frac{\delta}{\Delta t})}}{2} \tag{4.25}$$

$$\lambda_B = \frac{P_{e_z} + \sqrt{P_{e_z}^{\ 2} + 4P_{e_z}(\beta + \frac{\delta}{\Delta t})}}{2} \tag{4.26}$$

Now, a particular solution of equation (4.16) is obtained by the following sequence of integrations:

(1) $$L_B[x(z)] \equiv \frac{d}{dz}x - \lambda_B x = R_n^i(z) \tag{4.27}$$

(2) $$L_F[f_3^{i+1}(z)] \equiv \frac{d}{dz}f_3^{i+1} - \lambda_F f_3^{i+1} = x(z) \tag{4.28}$$

That f_3^{i+1} satisfies equation (4.16) is easily shown by:

$$L(f_3^{i+1}) = L_B \cdot L_F(f_3^{i+1}) = L_B(L_F(f_3^{i+1})) = L_B(x) = R_n^i$$

The method of finding the general solution is essentially the same as the procedure described in Chapter 3.

The integration of (4.23) needs to be done only once. However, if P_{e_z} is also time dependent, then (4.23) must be integrated at each time increment.

Now, at the radial boundaries where $r = 0$ and $r = 1$ equation (4.13) must be reformulated, we have

at $r = 0$, $\frac{\partial f}{\partial r} = 0$ and

at $r = 1$, $\frac{\partial f}{\partial r} = 0$ also $u = 0$

(1) At $r = 0$ and since $\lim_{r \to 0} \frac{1}{r} \frac{\partial f}{\partial r} = \frac{\partial^2 f}{\partial r^2}$, then we have

$$\frac{1}{P_{e_z}} \frac{\partial^2 f}{\partial z^2} + \frac{2}{P_{e_r}} \frac{\partial^2 f}{\partial r^2} - \frac{\partial f}{\partial z} - \beta f = \delta \frac{\partial f}{\partial t} \tag{4.29}$$

Following the same procedure, the above equation may be put in the form of the Equation (4.16) with

$$P = \frac{2}{P_{e_r} \Delta r^2} ; \quad Q = -\frac{4}{P_{e_r} \Delta r^2} ; \quad T = P$$

This then can be solved in the similar manner.

(2) At $r = 1$ where $u = 0$ and $P_{e_z} = P_{e_r} = 0$.

Multiply equation (4.13) by P_{e_z}, we obtain

$$\frac{\partial^2 f}{\partial z^2} + \frac{P_{e_z}}{P_{e_r}} \left[\frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} \right] - P_{e_z} \frac{\partial f}{\partial z} - P_{e_z} \beta f = \delta P_{e_z} \frac{\partial f}{\partial t} \tag{4.30}$$

where $\frac{\partial f}{\partial r} = 0$

and let

$$\frac{P_{e_z}}{P_{e_r}} = \frac{L^2 D_y}{D_x R^2} = V$$

$$P_{e_z} \beta = \frac{L^2 K_c}{D_x} = W$$

$$P_{e_z} \delta = \frac{L^2}{D_x T} = X$$

Then we have

$$\frac{\partial^2 f}{\partial z^2} + V \frac{\partial^2 f}{\partial r^2} - Wf = X \frac{\partial f}{\partial t} \tag{4.31}$$

The second derivative in r direction may be represented by

$$\frac{\partial^2 f}{\partial r^2} = \frac{f_{n+1} - 2f_n + f_{n-1}}{\Delta r^2}$$

$$\cong \frac{-f_n + f_{n-1}}{\Delta r^2} \qquad \text{since } f_{n+1} = f_n.$$

Finally:

$$\frac{d^2 f_n^{i+1}}{dz^2} - (W + \frac{X}{\Delta t})\, f_n^{i+1} = \left(\frac{V}{\Delta r^2} - \frac{X}{\Delta t}\right) f_n^i - \left(\frac{V}{\Delta r^2}\right) f_{n-1}^i$$

$$= R_n^i \qquad (4.32)$$

Since V, W, and X are all functions of z, by following the previous discussion, we have

$$L(\ ) = \frac{d^2}{dz^2} - (W + \frac{X}{\Delta t})$$

$$= (\frac{d}{dz} - \lambda_B)(\frac{d}{dz} - \lambda_F)$$

$$= \frac{d^2}{dz^2} - (\lambda_B + \lambda_F)\frac{d}{dz} - \frac{d\lambda_F}{dz} + \lambda_F\lambda_B \qquad (4.33)$$

By identifying the terms, it them becomes

$$\lambda_B = -\lambda_F$$

$$\frac{d\lambda_F}{dz} + \lambda_F^2 = W + \frac{X}{\Delta t} \qquad (4.34)$$

This last equation (Riccati equation) is stable when integrated from $z = 1$ to $z = 0$, with any initial conditions satisfying: $\lambda_F(1) \leq 0$.

If coefficients are constant, we can solve λ_F and λ_B directly:

$$\lambda_F = -\sqrt{W + \frac{X}{\Delta t}}$$

$$\lambda_B = +\sqrt{W + \frac{X}{\Delta t}} \qquad (4.35)$$

It is interesting to point out at this stage that in general case when integration of the Riccati equation is required, the particular solution becomes the required solution. This is easily proven by applying the axial boundary conditions.

Since $\left.\frac{\partial f}{\partial z}\right|_{z=1} = 0$

with the conditions $\lambda_F(1) = 0$ and $x(1) = 0$, we can immediately see that, at $z=1$, the particular solution is automatically satisfied:

$$\left.\frac{df_n^{i+1}}{dz}\right|_{z=1} = \lambda_F(1)\, f_n^{i+1} + x(1) \equiv 0 \qquad \text{g.e.d.}$$

The initial condition $f_n^{i+1}(0)$ for the above equation may be obtained

$$\left.\frac{\partial f}{\partial z}\right|_{z=0} = Pe_z(f - 1) \qquad \text{for } r \neq 1$$

$$= \lambda_F(0)\, f + x(0)$$

then $$f_n^{i+1}(0) = \frac{x(0) + Pe_z}{-\lambda_F(0) + Pe_z} \tag{4.36}$$

and for $r = 1$, simply $f_n^{i+1}(0) \equiv 1.$ (4.37)

The computation sequence may be outlined as follows:

(1) Integrate the Riccati equation, which is stable in the backward direction ($z = 1$ to 0), with any initial condition $\lambda_F(1) \leq 0$ to obtain $\lambda_F(z)$ and consequently $\lambda_B(z)$.

(2) The backward integration of

$$\frac{dx}{dz} = \lambda_B(z)\ x + R_n^i(z)$$

with $x(1) = 0$

(3) The forward integration of

$$\frac{df_n^{i+1}}{dz} = \lambda_F(z)\ f_n^{i+1} + x(z)$$

This sequence is repeated for $n = 1,2,\ldots,N$, for each time step t^{i+1}.

To demonstrate the method of solution, lets consider a simple first-order irreversible reaction of the form

$$A \rightarrow B + C \tag{4.38}$$

Based on the past work (14), the transport properties can be calculated numerically assuming that the fully developed velocity profile exists. For simplicity, the velocities and eddy diffusivities are kept constant along the axial direction. The complete details are further discussed in Appendix F.

The successful application of this explicit/implicit approach requires that for this algorithm to be stable the following condition should be met (30)

$$\frac{\Delta t}{\Delta r^2} \leq \frac{\delta Pe_r}{4}$$

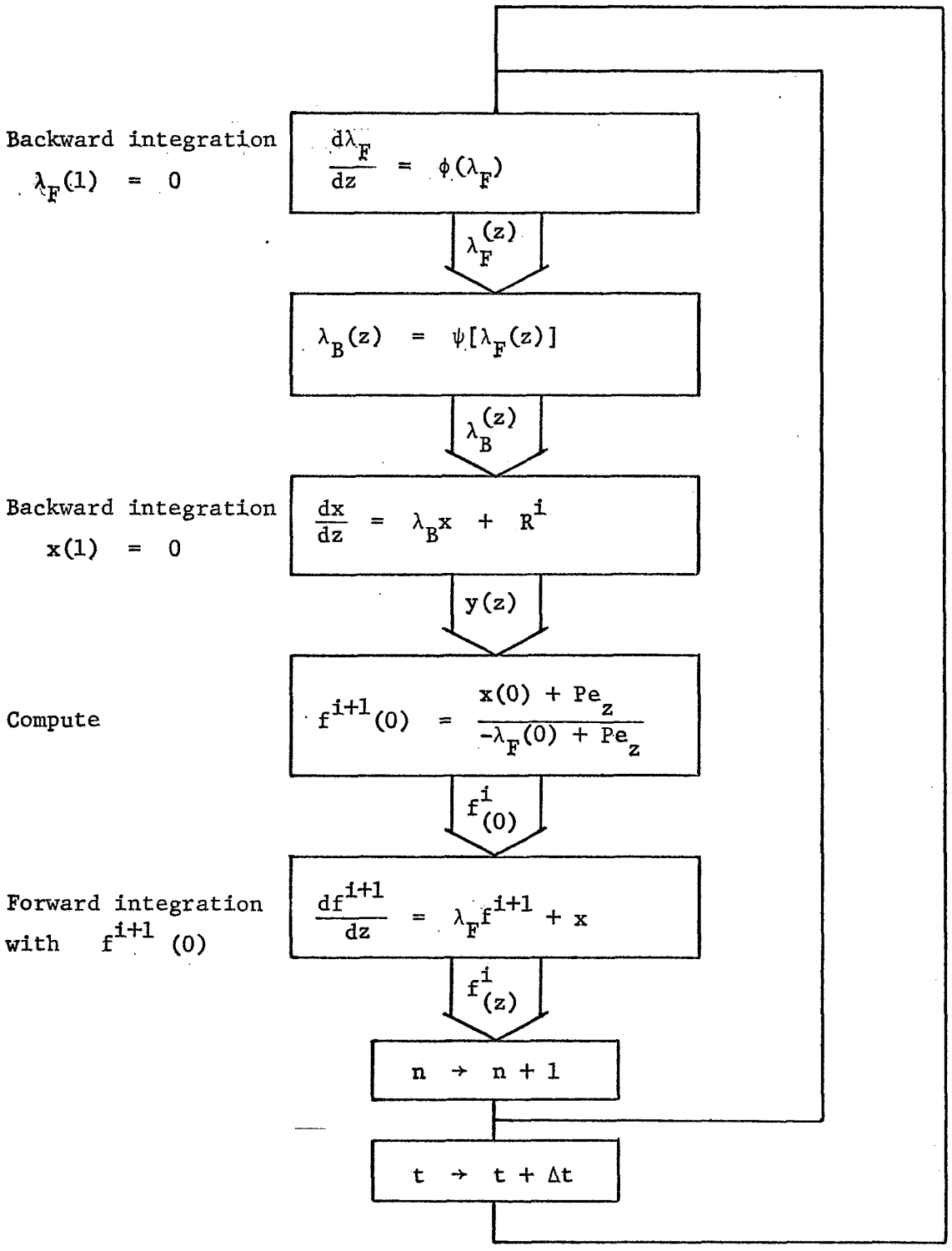


Figure 4.2.1 Computing Sequence Block Diagram

so that R^i in equation (4.16) will always have the same sign (negative). A similar criterion can be found from any text discussing the same approach.

Figure 4.2.2 is the transient result, a plot of reactant concentration along the reactor axis at various radial lines at the end of 20 time steps after a step input in concentration of A has been introduced. Due to the limitation of the plot routine used the graphs do not represent all the radial increments used in the simulation, instead only the following nine curves are shown:

Curve 1 : n = 1 (center line)

Curve 2 : n = 3

Curve 3 : n = 5

Curve 4 : n = 6

Curve 5 : n = 7

Curve 6 : n = 8

Curve 7 : n = 9

Curve 8 : n = 10

Curve 9 : n = 11 (wall line)

where n represents radial increment.

Plots of the reactant concentration profiles in the radial direction are presented in Figures 4.2.3 (each figure represents 9 axial points for the total of 50 axial points).

Transient results at the early time steps are also shown in Appendix F (Figures F-3 through F-8).

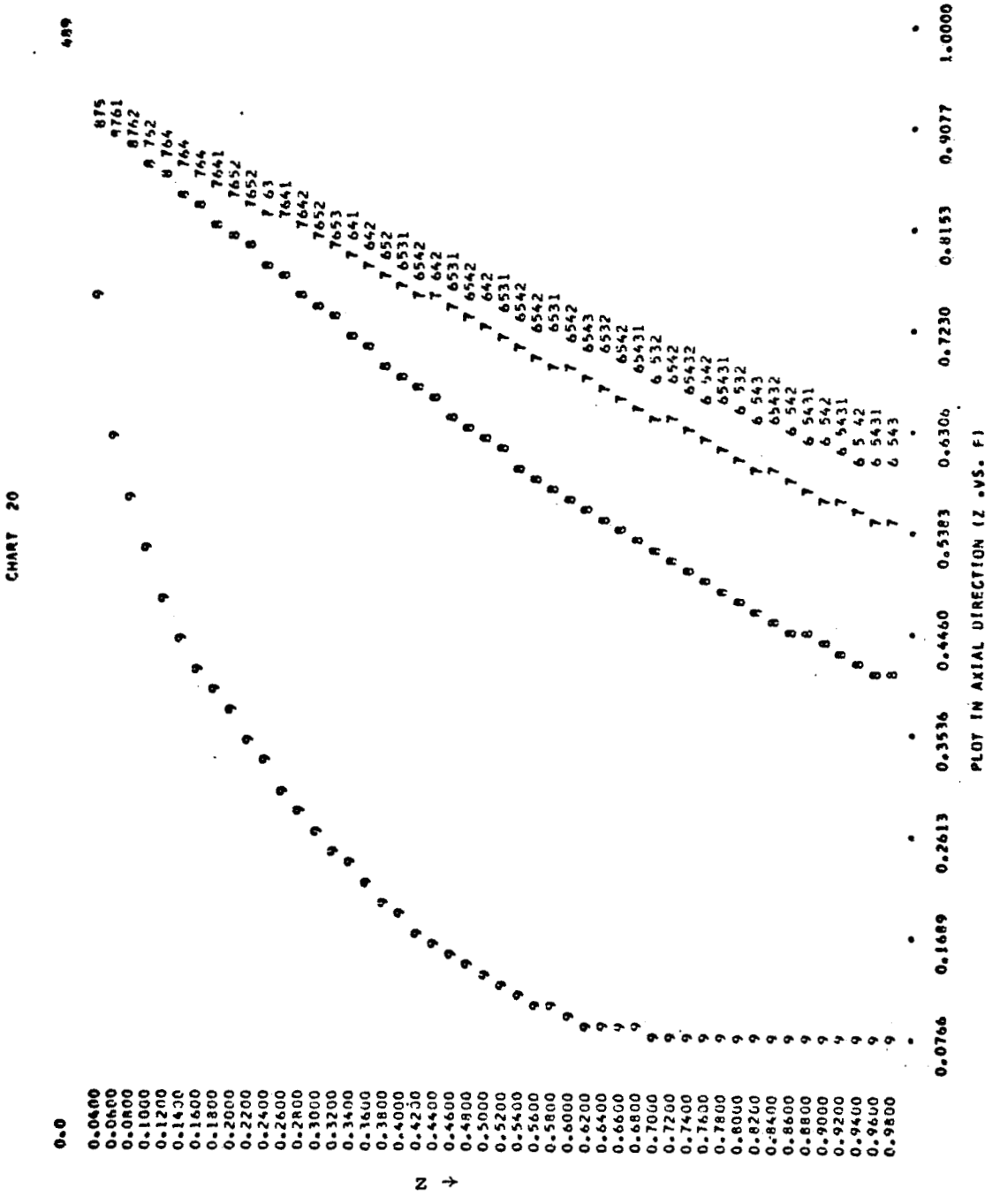


Figure 4.2.2 Reactant Concentration Profiles in Axial Direction
Transient Results at 20 Δt

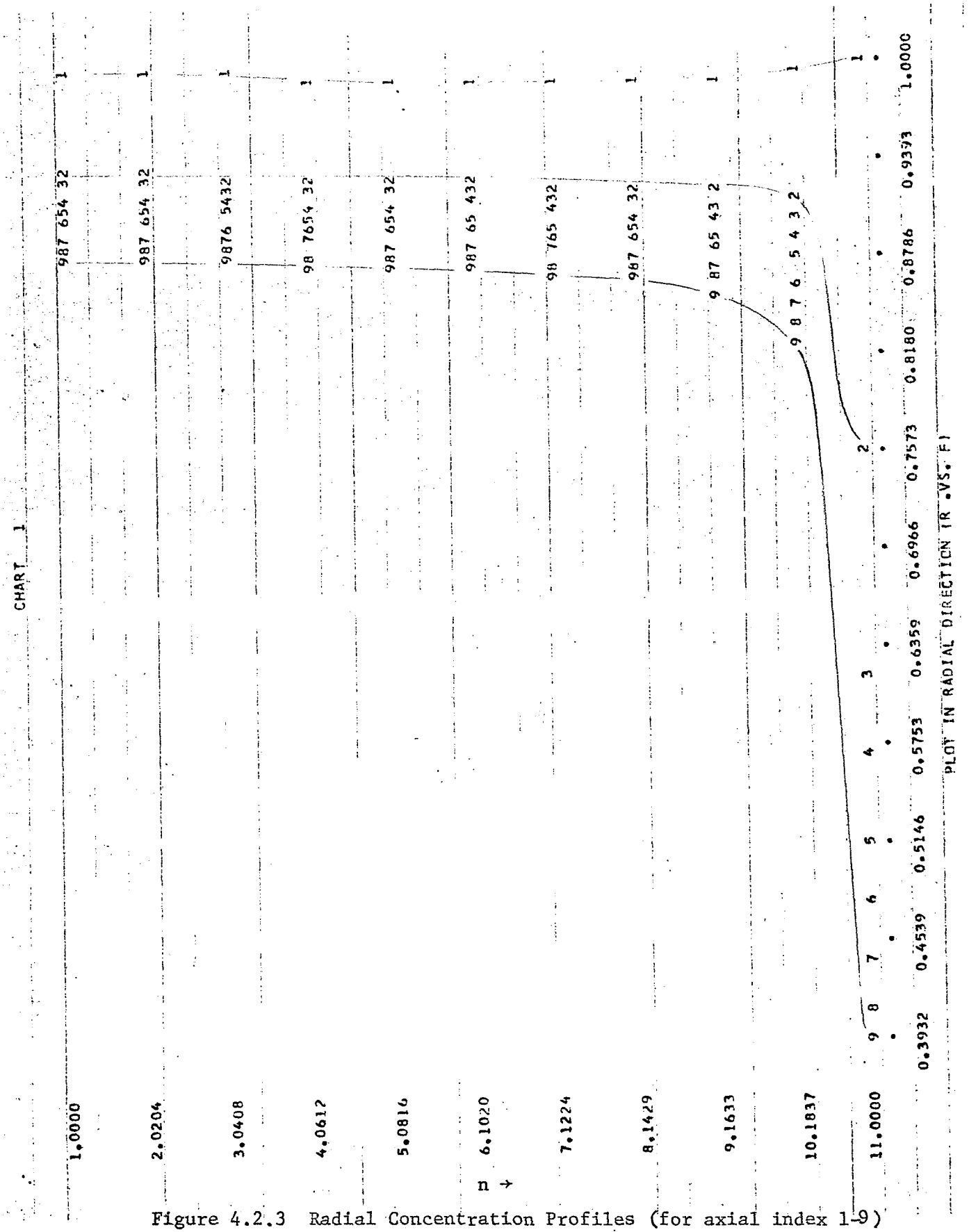


Figure 4.2.3 Radial Concentration Profiles (for axial index 1-9)

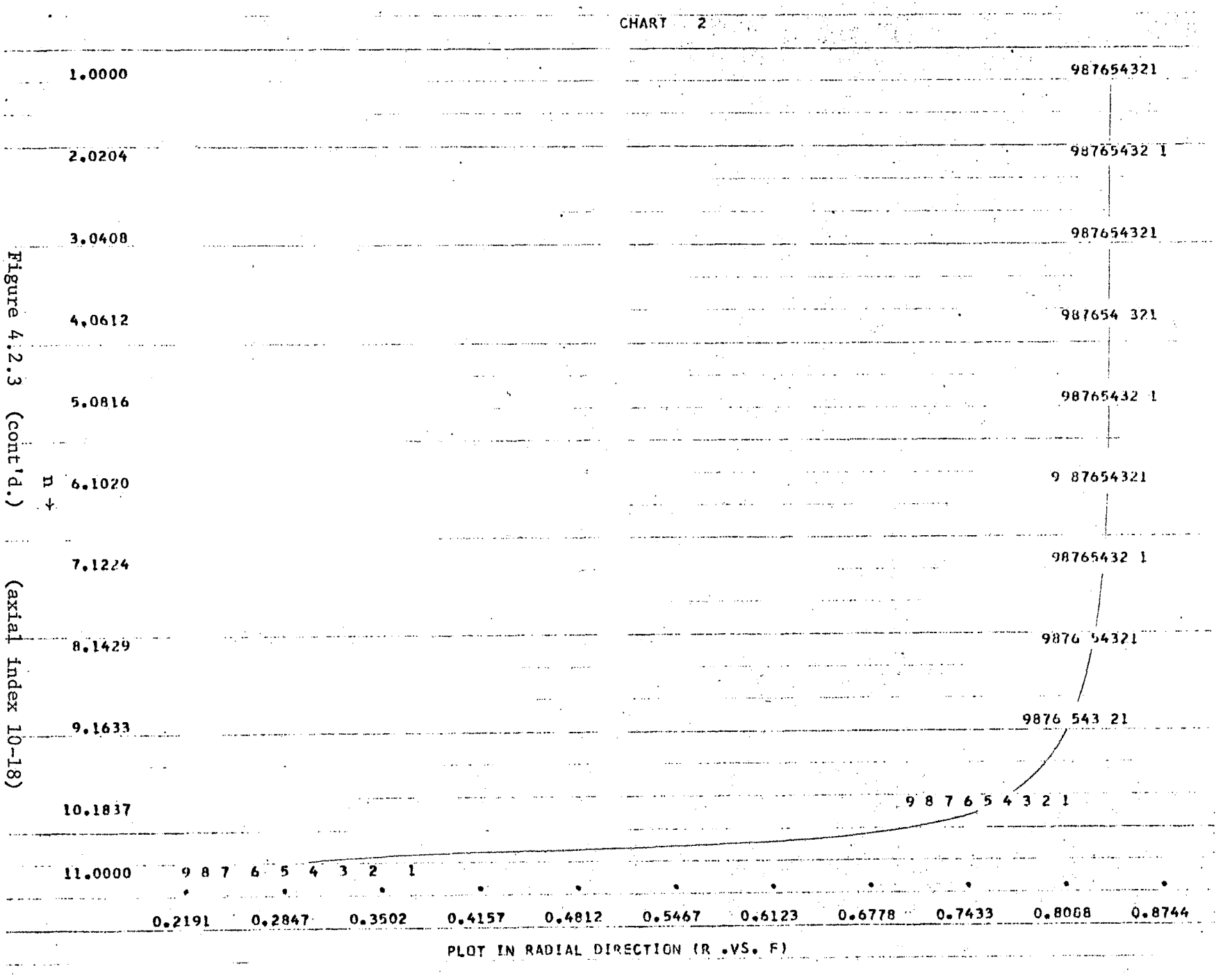


Figure 4.2.3 (cont'd.) (axial index 10-18)

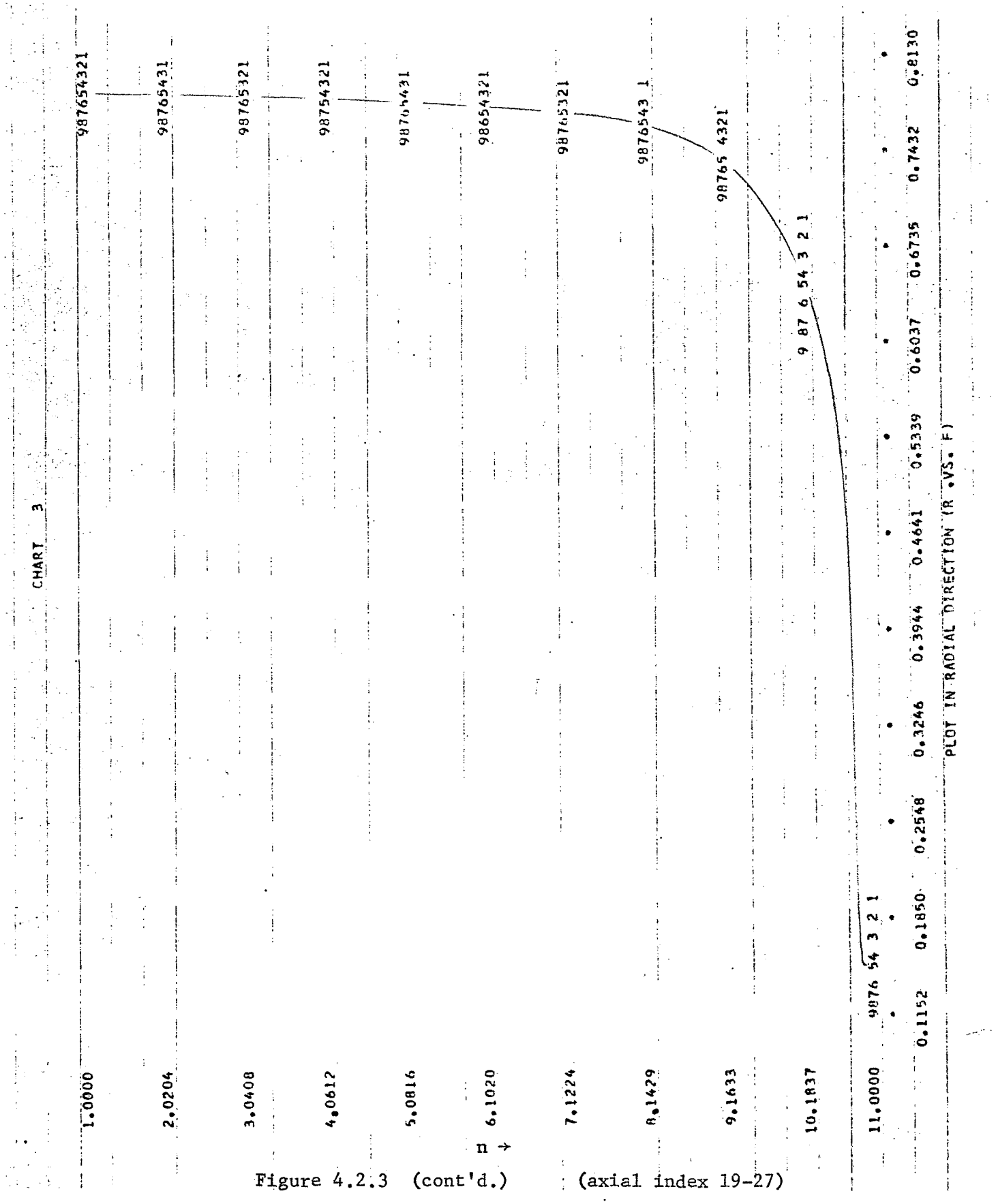


Figure 4.2.3 (cont'd.) (axial index 19-27)

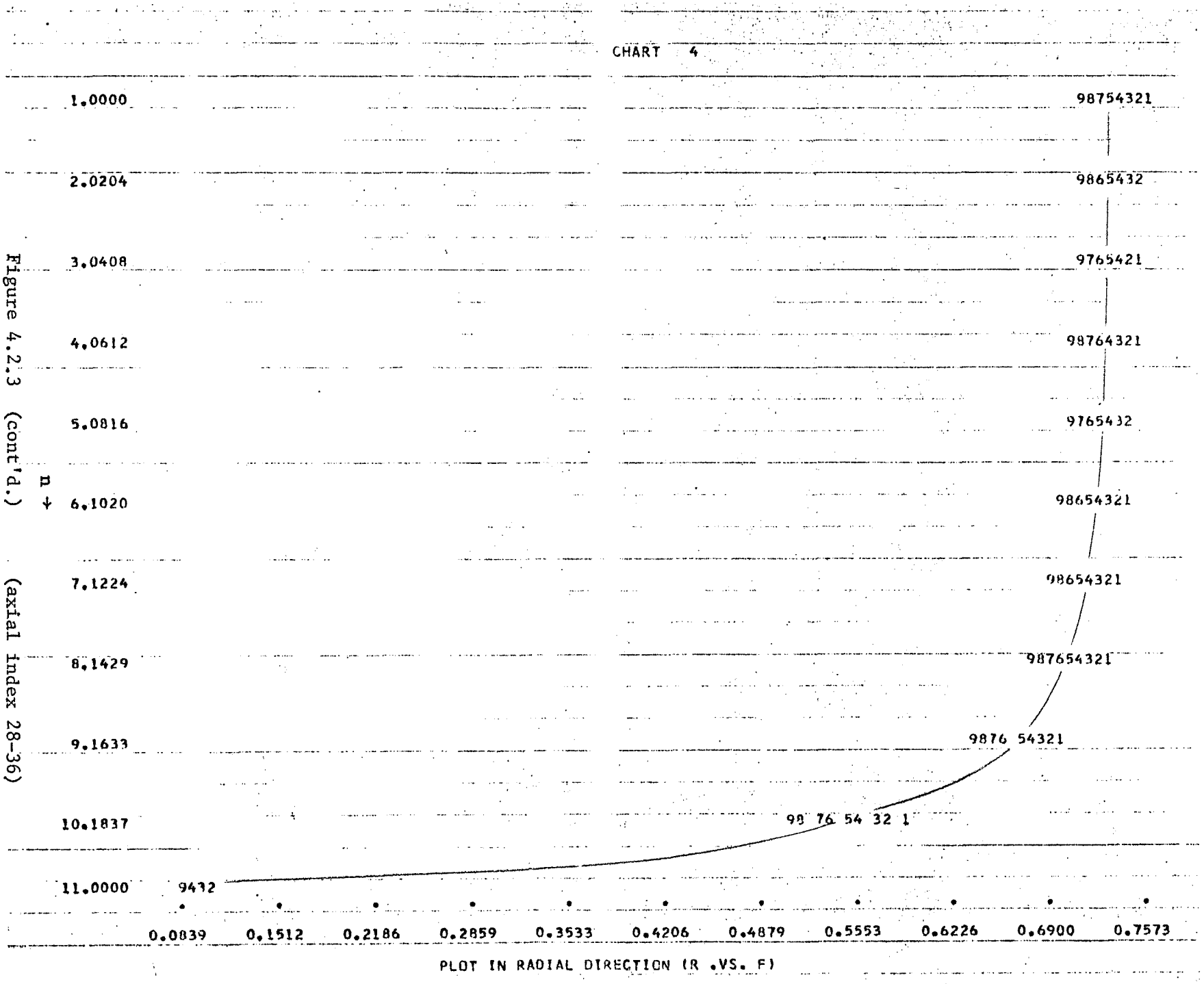


Figure 4.2.3 (cont'd.) (axial index 28-36)

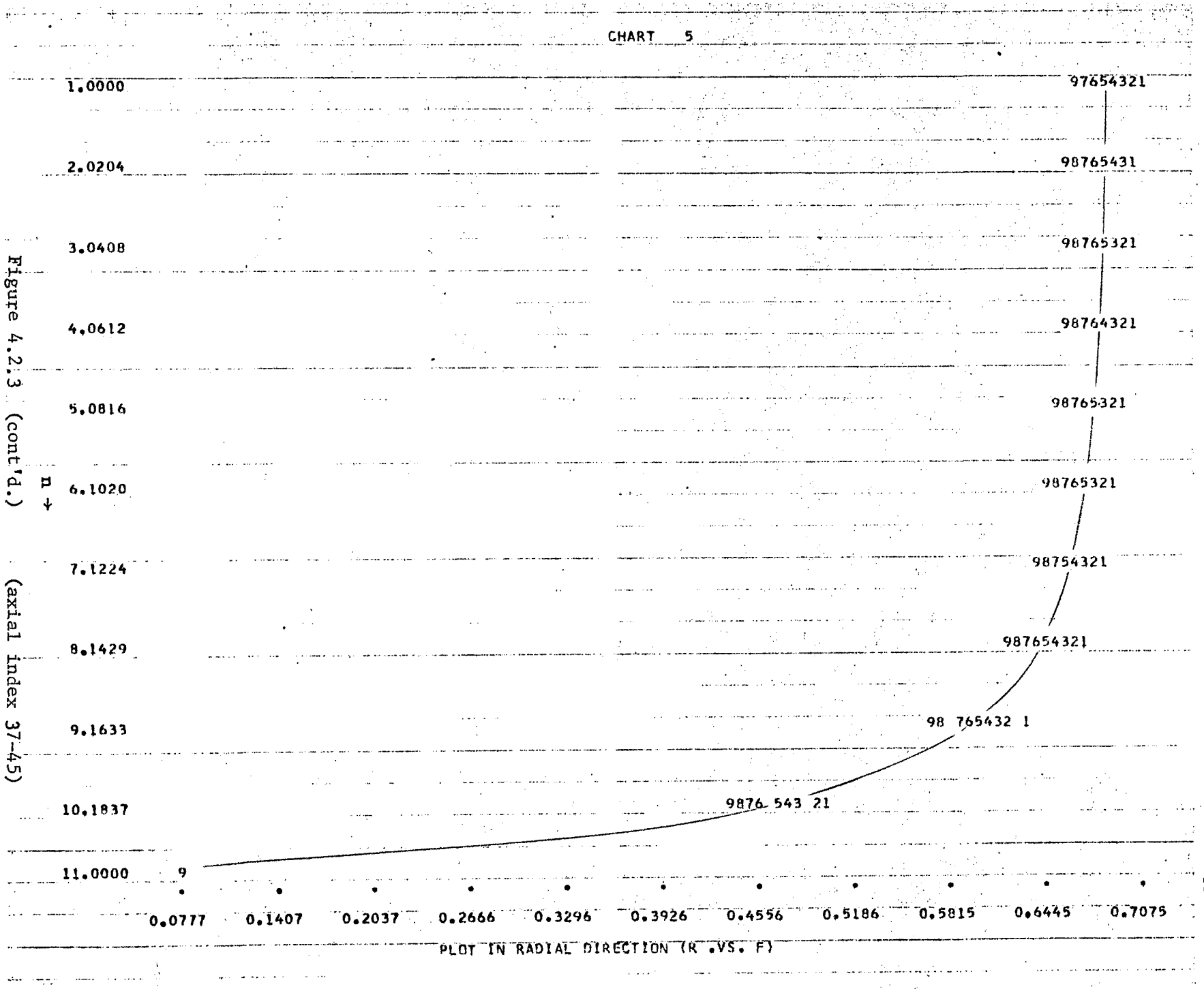


Figure 4.2.3 (cont'd.) (axial index 37-45)

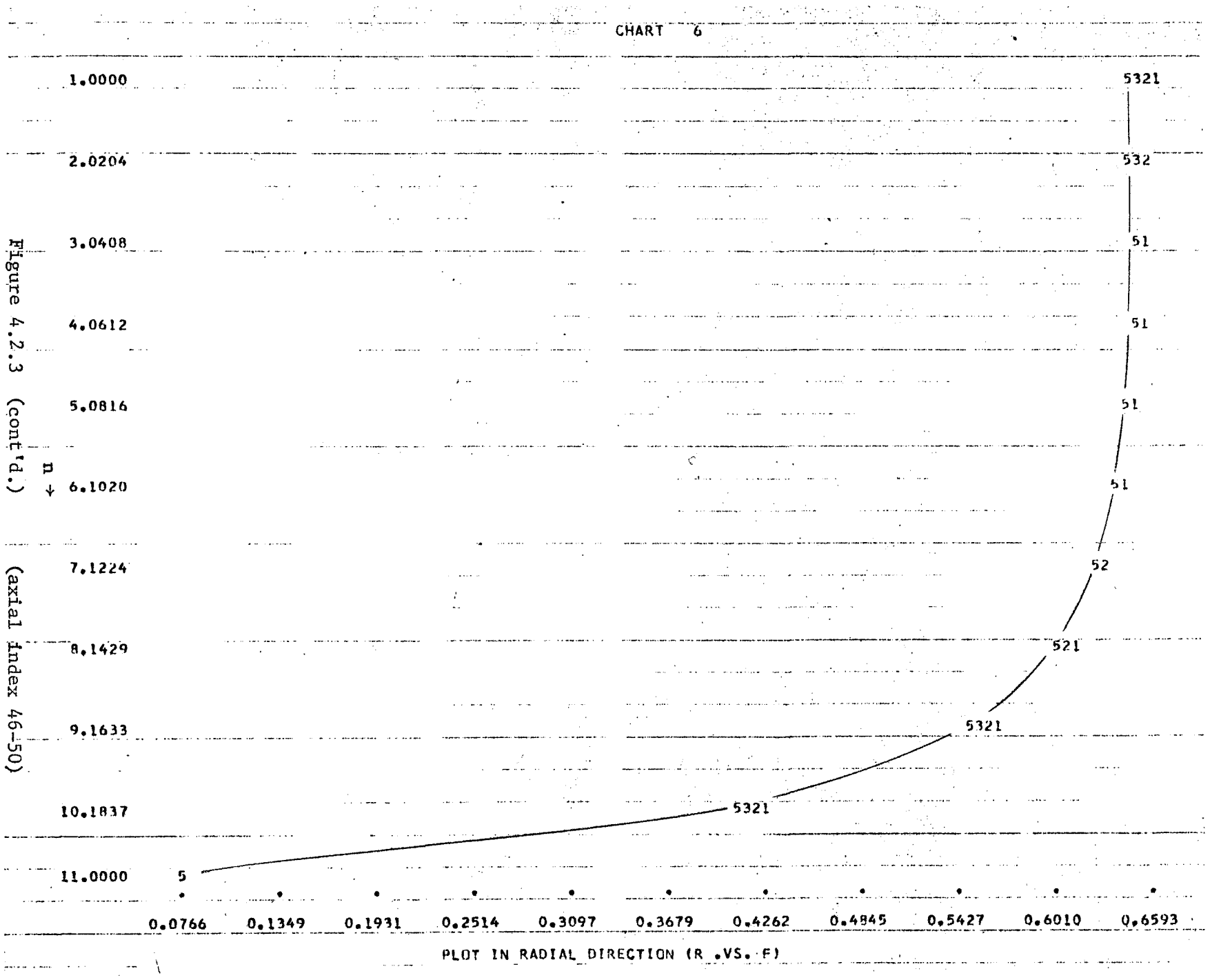


Figure 4.2.3 (cont'd.) (axial index 46-50)

CHAPTER 5

Simulation of A Non-isothermal Tubular Flow Chemical Reactor -- Energy and Material Balances

All tubular flow chemical reactors in which heat is exchanged with the surroundings are at least two-space-dimensional. However, a major difficulty usually arises in analyzing the performance of these reactors because the set of partial differential equations describing the system are all highly nonlinear and coupled. From past experience (14) it has been demonstrated that for a steady-state operation, simulation of this type of distributed-parameter system is feasible using digital computers, but a considerable amount of computation time and effort is required before any results can be obtained. When the assumption of steady-state operation is removed, the chemical reactor dynamics now involve three independent variables, two of space and one of time. Unfortunately, there has not been a successful result reported on this problem. In the previous chapters we have shown the superior potential of the hybrid method in solving partial differential equations. Despite some difficulties in implementation and inaccuracies of hardware components, it appears to be the most attractive approach available to date for tackling such a highly nonlinear coupled system.

Consider a homogeneous turbulent flow tubular reactor with heat transfer from the wall of the reactor as shown in Figure 5.1.

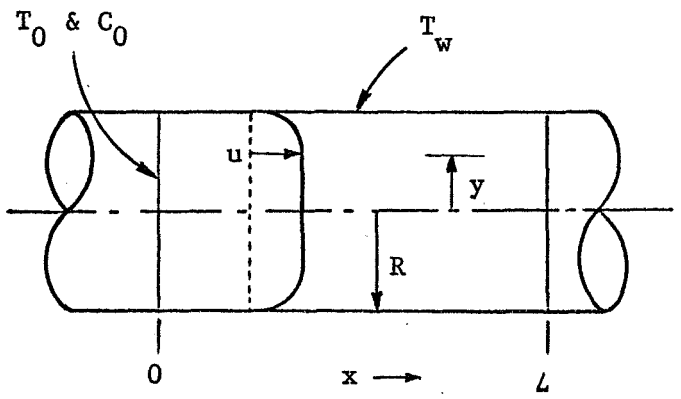


Figure 5.1 Two-Dimensional, Non-Isothermal Tubular Reactor

A complete mathematical description of this reacting system leads to a system of nonlinear partial differential equations which consists of momentum, energy, and material balances. Following the previous work the velocity and eddy diffusivity profiles are developed by the direct numerical integration of Deissler's analogy to the boundary layer zone and of the Prandtl-von Karman mixing length theory to the turbulent core (14). The general form for this relationship may be expressed as

$$\tau g_c = -(\mu + \rho\varepsilon) \frac{du}{dy} \tag{5.1}$$

where τ : Shear

g_c : gravitational constant

μ : viscosity

ρ : density

ε : eddy diffusivity of momentum

u : velocity

y : radial distance

The same analogy is also assumed to exist for energy and material transfer. If the velocity and eddy diffusivity can be calculated, the problem is reduced to one consisting of two simultaneous partial differential equations for the conservation of energy and matter. Since the primary objective of this research is to explore the feasibility of the hybrid computing system in solving partial differential equations, the method which was used to evaluate the velocity and eddy diffusivity profiles is briefly discussed in Appendix G and will not be elaborated here again. The detailed discussions of fluid flow problems can be found elsewhere (14).

5.1 Energy Balance

Heat transfer in a tubular flow chemical reactor may be expressed by

$$K_x \frac{\partial^2 T'}{\partial x^2} + K_r \left[\frac{\partial^2 T'}{\partial y^2} + \frac{1}{y} \frac{\partial T'}{\partial y} \right] - GC_p \frac{\partial T'}{\partial x} - R\Delta H = C_p \rho \frac{\partial T'}{\partial \theta} \qquad (5.2)$$

By making variable substitutions

$$z = \frac{x}{L}, \quad r = \frac{y}{R}, \quad T = \frac{T'}{T_w}, \quad t = \frac{\theta}{\Theta}, \quad f = \frac{C}{C_0}$$

the above equation is transformed into equation (5.3) for a system with a simple first order reaction

$$\frac{1}{Pe'_z} \frac{\partial^2 T}{\partial z^2} + \frac{1}{Pe'_r} \left[\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right] - \frac{\partial T}{\partial z} - H(T)f = \phi' \frac{\partial T}{\partial t} \qquad (5.3)$$

where $Pe_z' = \frac{GC_pL}{K_x}$ (Longitudinal Peclet number for heat transfer)

$Pe_r' = \frac{GC_pR^2}{K_rL}$ (Radial Peclet number for heat transfer)

$H(T) = \frac{L\Delta HK_cC_0}{T_wGC_p}$ (where $K_c = A_ce^{-E/RT'}$, Arrhenius type rate constant)

$\phi' = \frac{L}{u\Theta}$

with $T = T(z,r,t)$

$z \varepsilon (0,1)$

$r \varepsilon (0,1)$

T_w = tube wall temperature

And the following initial and boundary conditions:

$$T(z,r,0) = T_0$$

$$T(0,r,t) = T_0 \qquad \text{(constant inlet temperature)}$$

$$\frac{\partial T}{\partial z}(1,r,t) = 0 \qquad (5.4)$$

$$\frac{\partial T}{\partial r}(z,0.t) = 0$$

$$\frac{\partial T}{\partial r}(z,1,t) = h[T(z,1,t) - T_w(x,t)]$$

where h = tube wall heat transfer functionality.

Discretizing the r and t components with a most simplified assumption that Pe_z' remains constant along the constant radial line (axial direction), Equation (5.3) may be transformed into the following

expression

$$\frac{d^2T}{dz^2} - Pe_z' \frac{dT}{dz} - Pe_z' \, T \frac{\phi'}{\Delta t} - H(T)f = R_T^i \tag{5.5}$$

where R_T^i can be obtained from the previous time step similar to that in equation (4.16). Equation (5.5) is a nonlinear equation, and is coupled through the term $H(T)f$ with the material balance equation which will be discussed later.

In order that the decomposition method may be applied to this nonlinear problem, it is necessary to use a successive linearization technique. A linear operator $L(\ )$ is developed, whose coefficients are functions of the solution at the preceding time step i.

$$L(T) = \frac{d^2T}{dz^2} - Pe_z' \frac{dT}{dz} - Pe_z' \frac{\phi}{\Delta t} T = R_T^i + H(T)^i f \tag{5.6}$$

Iterative procedure is then used to update $H(T)^i$ until convergence has achieved (two steps are sufficient in most cases).

By decomposing $L(\ )$, we have

$$\begin{aligned} L(\ ) &= \frac{d^2}{dz^2} - Pe_z' \frac{d}{dz} - Pe_z' \frac{\phi}{\Delta t} \\ &= \left(\frac{d}{dz} - \lambda_B\right)\left(\frac{d}{dz} - \lambda_F\right) \\ &= \frac{d^2}{dz^2} - (\lambda_B + \lambda_F) \frac{d}{dz} + \lambda_F \lambda_B \end{aligned} \tag{5.7}$$

identifying each term leads to the expressions:

$$\lambda_F = \frac{Pe_z' - \sqrt{Pe_z'^2 + 4Pe_z' \phi/\Delta t}}{2}$$

$$\lambda_B = \frac{Pe_z' + \sqrt{Pe_z'^2 + 4Pe_z' \phi/\Delta t}}{2} \tag{5.8}$$

Note that (5.8) does not apply to the case when $r = 1$, since $Pe_z' = 0$.

Then a particular solution is obtained by:

(1) $$L_B[x(z)] = \frac{d}{dz} x - \lambda_B x = R_T^i + H(T)f = y(z) \tag{5.9}$$

(2) $$L_F[T_3(z)] = \frac{d}{dz} T_3 - \lambda_F T_3 = x(z) \tag{5.10}$$

To find a general solution

$$T(z) = a\ T_1(z) + b\ T_2(z) + T_3(z) \tag{5.11}$$

apply the boundary conditions

$$T(0) = T_0$$

$$\frac{dT}{dz}(1) = 0 \tag{5.12}$$

leading to the following relations

$$a\ T_1(0) + b\ T_2(0) + T_3(0) = T_o$$

$$a\ \lambda_B T_1(1) + b\ \lambda_F T_2(1) + \lambda_F T_3(1) + x(1) = 0 \tag{5.13}$$

Since we know:

$$T_1(1) = 1, \quad T_2(0) = 1, \quad T_3(0) = 0 \quad \text{and}$$

$$x(1) = 0$$

these simplify the relations to

$$\begin{aligned} &a\, T_1(0) + b = T_0 \\ &a\, \lambda_B + b\, \lambda_F T_2(1) + \lambda_F T_3(1) = 0 \end{aligned} \tag{5.14}$$

Solving this relation we get for a and b:

$$\begin{aligned} a &= \frac{T_0 + T_3(1)/T_2(1)}{T_1(0) - \lambda_B/\lambda_F T_2(1)} \\ b &= T_0 - a\, T_1(0) \end{aligned} \tag{5.15}$$

The general term of R_T^i can be expressed as:

$$R_T^i = -\frac{Pe_z'}{Pe_r'}\left[P T_{n+1}^i + \left(Q + \frac{\phi\, Pe_r'}{\Delta t}\right) T_n^i + S T_{n-1}^i\right] \tag{5.16}$$

where

$$P = \frac{2n+1}{2n\Delta r^2}$$

$$Q = -2/\Delta r^2$$

$$S = \frac{2n-1}{2n\Delta r^2}$$

Particular cases:

1) At the center line where $r = 0$, the above expressions may be reduced to

$$P = \frac{2}{\Delta r^2} \; ; \quad Q = -\frac{4}{\Delta r^2} \; ; \quad S = P$$

2) At $r = 1$ (the wall line), with

$$\frac{dT}{dr} = h(1 - T), \text{ for heating} \tag{5.17}$$

Multiply equation (5.3) by Pe'_z to obtain

$$\frac{\partial^2 T}{\partial z^2} + \frac{Pe'_z}{Pe'_r}\left[\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right] - Pe'_z\, H(T) f = Pe'_z\, \phi\, \frac{\partial T}{\partial t} \tag{5.18}$$

Since $u = 0$ at the wall, using (5.17) equation (5.18) may be rewritten as

$$\frac{\partial^2 T}{\partial z^2} + V\left[\frac{\partial^2 T}{\partial r^2} + h(1 - T)\right] - Wf = X\,\frac{\partial T}{\partial t} \tag{5.19}$$

where $V = \dfrac{L^2 K_r}{R^2 K_x}$

$$W = \frac{L^2 \Delta H\, C_0}{K_x\, T_w} K_c$$

$$X = \frac{C_p \rho L^2}{K_x \Theta}$$

applying DSDT in r and t domains leads to

$$\frac{d^2T_n}{dz^2} - \left(Vh + \frac{X}{\Delta t}\right)T_n = R_T^i + Wf \tag{5.20}$$

with

$$R_T^i = \left(\frac{2V}{\Delta r^2} - \frac{X}{\Delta t}\right)T_n^i - \left(\frac{V}{\Delta r^2}\right)T_{n-1}^i - V\left(h + \frac{1}{\Delta r^2}\right) \tag{5.21}$$

we also obtain for $r = 1$ that,

$$\begin{aligned} \lambda_F &= -\sqrt{Vh + \frac{X}{\Delta t}} \\ \lambda_B &= +\sqrt{Vh + \frac{X}{\Delta t}} \end{aligned} \tag{5.22}$$

5.2 Material Balance

The expression for mass transfer in a tubular reactor accompanied by heat transfer is identical to that of an isothermal reactor except that the reaction term now becomes a space dependent variable rather than a constant as in the isothermal problem.

Rewrite equation (4.13) to give:

$$\frac{1}{Pe_z}\frac{\partial^2 f}{\partial z^2} + \frac{1}{Pe_r}\left[\frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r}\right] - \frac{\partial f}{\partial z} - \beta(T)f = \delta\frac{\partial f}{\partial t} \tag{5.23}$$

where $\beta(T) = \frac{L}{u}K_c(T)$

The initial and boundary conditions are essentially the same as that of the isothermal case which are described by equation (4.14). Here we

can see that equations (5.3) and (5.23) are coupled through the reaction and heat generation terms $\beta(T)f$ and $H(T)f$. Equation (5.23) is linear in f. Applying the CSDSDT approach, we obtain

$$\frac{d^2f}{dz^2} - Pe_z \frac{df}{dz} - g(z)f = R_f^i \tag{5.24}$$

where $g(z)$ is unknown function of T, and will be evaluated after each iteration of the evergy balance is performed.

Identification of the decomposed equation gives

$$\begin{aligned} \lambda_B(z) + \lambda_F(z) &= Pe_z \\ \frac{d\lambda_F}{dz} - \lambda_F\lambda_B &= g(z) \end{aligned} \tag{5.25}$$

Equation (5.25) leads to the Riccati equation of the following form

$$\frac{d\lambda_F}{dz} - \lambda_F(Pe_z - \lambda_F) = g(z) \tag{5.26}$$

which is stable when integrated from $z = 1$ to 0

with $\lambda_F(z) \leq 0$

For $r = 1$ we can obtain the same expression from equation (4.32)

$$\frac{d^2f}{dz^2} - g(z)f = R_f^i \tag{5.27}$$

where $g(z) = W(x) + \frac{X}{\Delta t}$

The corresponding Riccati equation for $r = 1$ is

$$\frac{d\lambda_F}{dz} + \lambda_F^2 = g(z) \tag{5.28}$$

with

$$\lambda_B(z) = -\lambda_F(z) \tag{5.29}$$

The solution of the mass transfer equation is obtained by integrating the following two first order equations:

(1) Backward simultaneous integration of the Riccati equation and

$$L_B[x(z)] = \frac{d}{dx} x - \lambda_B(z)x = R_f^i(z) \tag{5.30}$$

with the initial conditions $x(1) = 0$

$\lambda_F(1) = 0$

and the relation $\lambda_B(z) = Pe_z - \lambda_F(z)$

(2) Forward integration of

$$L_F[f(z)] = \frac{d}{dz} f - \lambda_F(z)f = x(z) \tag{5.31}$$

with $f(0) = \dfrac{Pe_z + x(0)}{Pe_z - \lambda_F(0)}$ for $r \neq 1$

and $f(0) \equiv 1$ for $r = 1$

5.3 Complete Simulation Procedure

Based on the foregoing developments the integration procedure for a complete time-step of these coupled heat and mass transfer equations can be summarized as follows:

1) Begin with the energy balance equation, since the eigenvalues are space independent. First, we obtain the two elementary solutions $T_1(z)$ and $T_2(z)$ on the digital computer.

2) Integrate on the analog computer the two stable equations with the assumed valued for $H(t)f$ to obtain $T_3(z)$.

3) Digitally compute a and b by the application of the boundary conditions.

4) Compute the general solution for the temperature

$$T(z) = aT_1(z) + bT_2(z) + T_3(z)$$

on the digital computer.

5) Then start the material balance equation; digitally compute $g(z)$ using the results from step 4.

6) Integrate the Riccati equation with $\lambda_F(1) = 0$ to obtain $\lambda_F(z)$ and $\lambda_B(z)$ on the analog computer and store these values in the digital memory.

7) Backward integration (concurrent with step 6) of

$$\frac{dx}{dz} = \lambda_B(z)x + R_f^i(z)$$

with $x(1) = 0$ on the analog computer.

8) Digitally compute the initial value $f(0)$.

9) Forward integration of the mass transfer solution equation

$$\frac{df}{dz} = \lambda_F(z)f + x(z)$$

on the analog computer.

10) Check convergence of $T(z)$ and $f(z)$, if not, then update $H(T)f$ values and return to step 2. If converged,then, step to the next adjacent radial increment.

The simplified block diagram of the above procedure is illustrated in Figure 5.3.1.

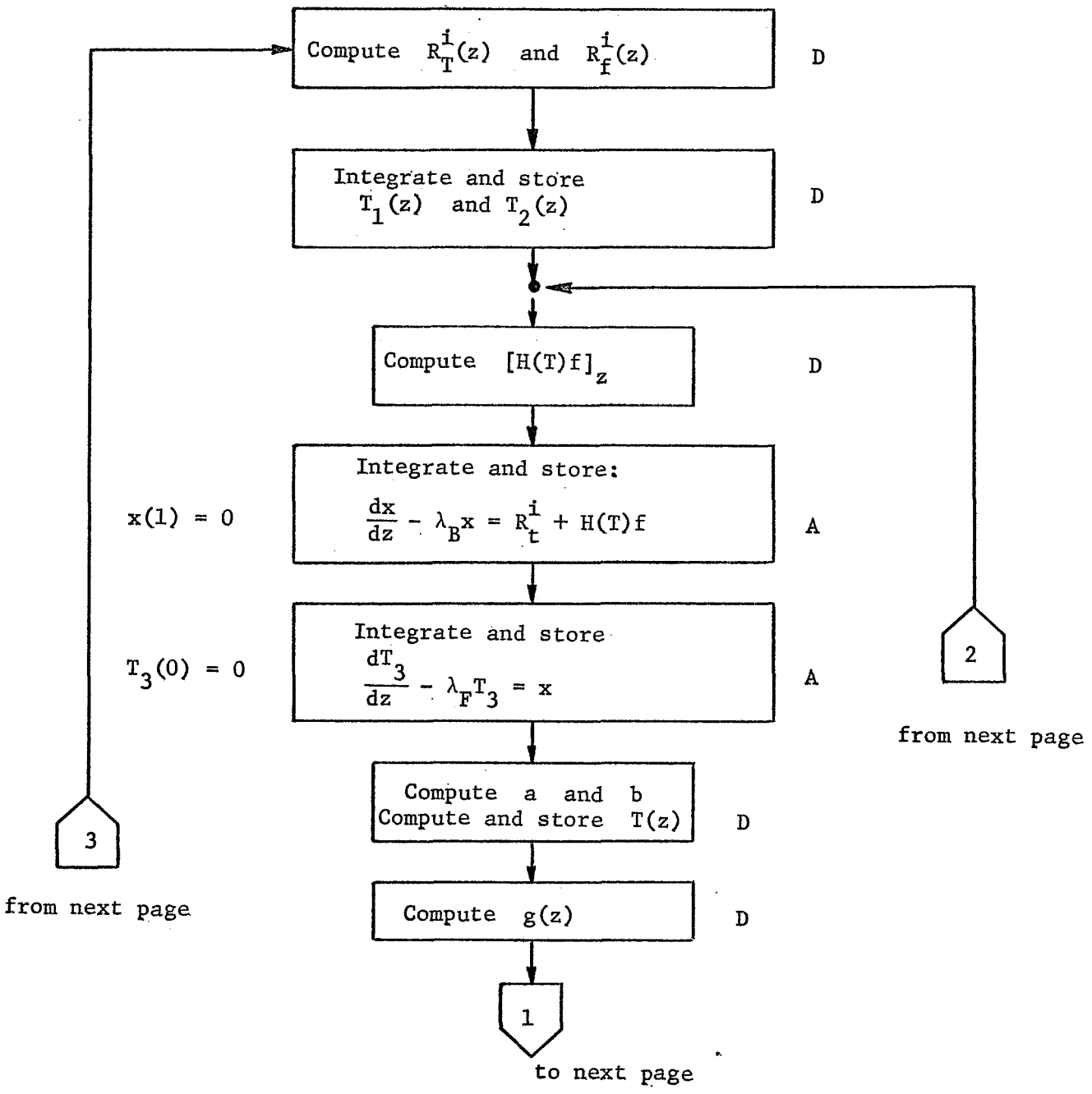


Figure 5.3.1 Heat and Mass Balance Computing Sequence Block Diagram (computing functions are indicated by: A = analog, D = digital)

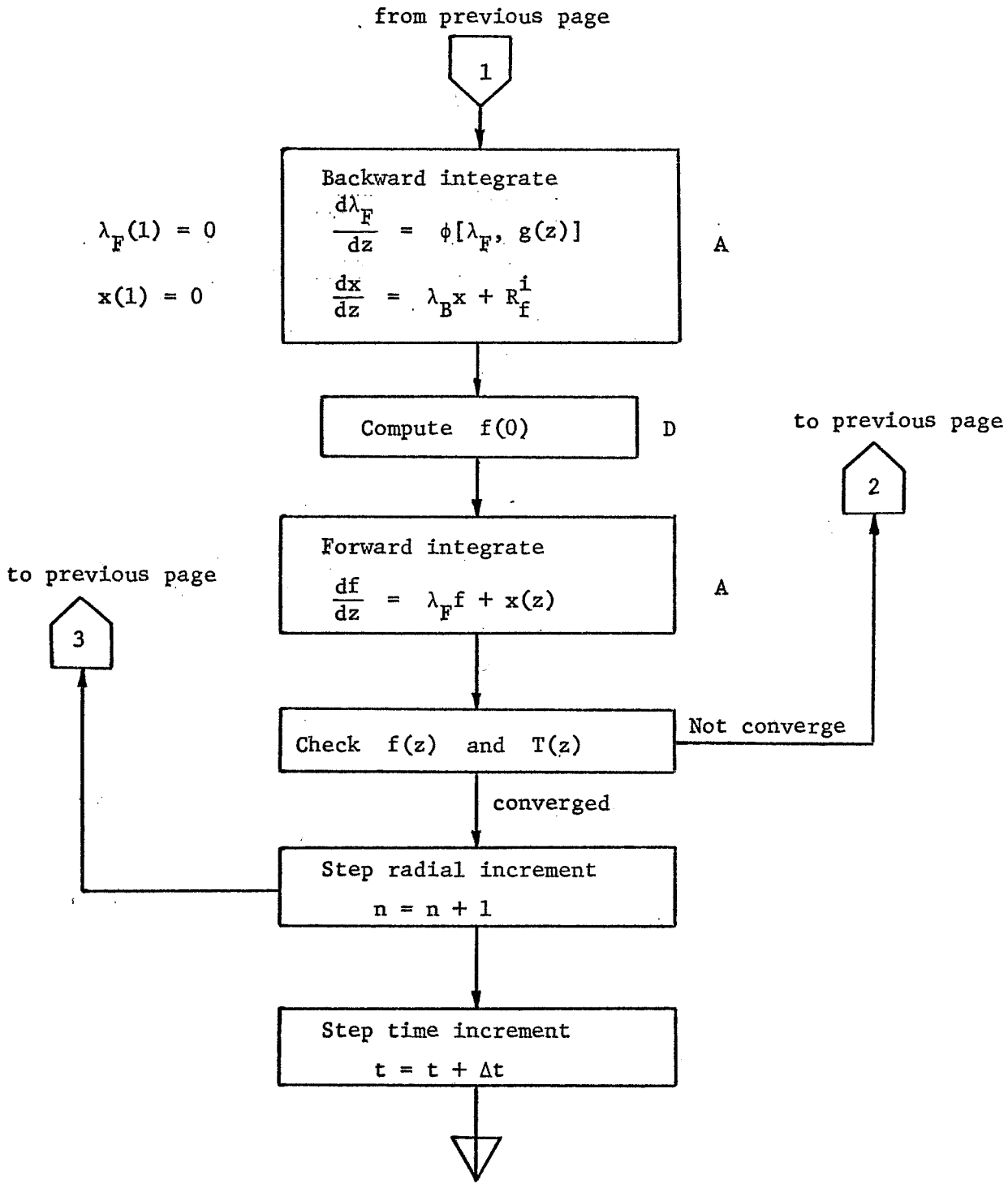


Figure 5.3.1 (Cont'd.)

The overall hybrid simulation procedure may be summarized in the following figure.

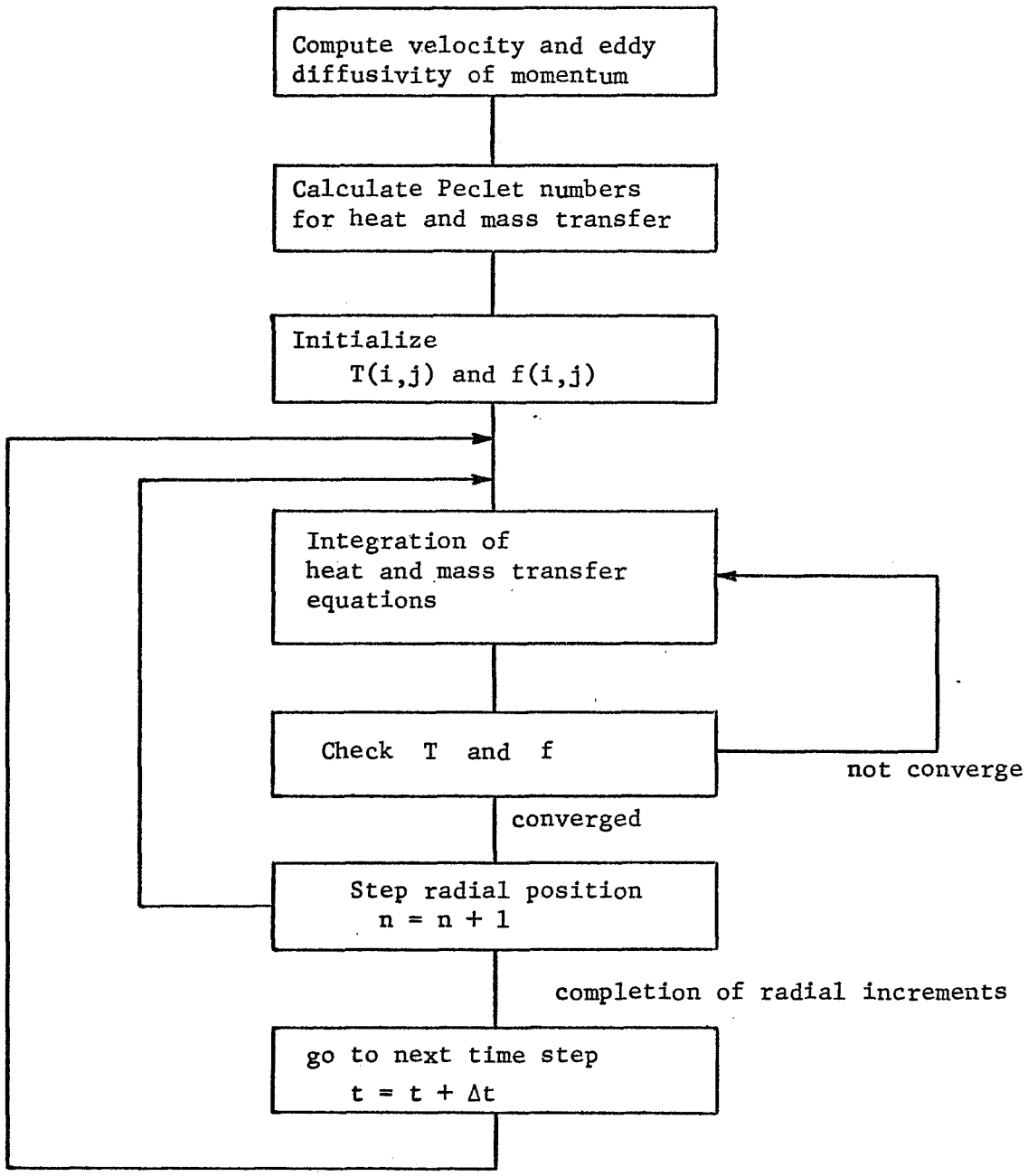


Figure 5.3.2 Overall Computing Schedule

It is important to point out that our preceding discussion is based on the constant velocity and eddy diffusivity profiles. However, this is not true especially for the case in which a notable volume change would occur as a result from the reaction. Under this condition it is necessary to reformulate the decomposed individual linear first order operator, since the Peclet numbers are no longer constant along the z axis. In addition, within each time step convergence of the velocity and eddy diffusivity profiles will be required.

To demonstrate the method of solution, again we chose the example problem described in Appendix F. The constant wall temperature, T_w, is assumed. The reactor at its initial state is supposed to have zero concentration of the reactant A and with a uniform temperature (equals to the input temperature, T_0) throughout the entire length of the reactor tube. At time zero a step input is introduced, and a constant wall temperature provides heating or cooling depending on whether the reaction is an endothermic or exthothermic type.

With a simplified assumption which does not require the reevaluation of the velocity and eddy diffusivity profiles, the following table shows approximately the extent of the required hybrid computer hardware other than the digital computer for this particular case study.

The complete details pertaining to the hybrid computing procedure for this case study are shown in Appendix G.

Table 5.3 Hybrid Hardware Requirements for Energy and Material Balance

Hardware Components	Energy Balance	Material Balance	Total
DAC	4	11	15
ADC	4	6	10
Multiplier	-	4	4
Integrator	4	6	10
Summer/Invertor	8	11	19
Servoset Pot.	7	3	10
Sample Hold Amp.	-	4	4

The actual transient results up to the second time step are shown in the Figures 5.3.3 through 5.3.7. No significant difference is found in the concentration curves comparing to the result from the isothermal case, however, the temperature profiles exhibit some interesting results. The result of the first time step shows the temperature curves (other than the wall line) actually went down for a while because an endothermic reaction was taking place in this tubular reactor. And the temperatures are lower the more the lines are closer to the wall, because we can visualize from the concentration curves that in fact more reaction has occurred toward the wall. In the next time step (Figure 5.3.5), the relative positions of the temperature lines now begin to move, for example, curve 8, which is the nearest line to the wall, is showing the heating effect from the wall and is moving upward. The hybrid simulation

was discontinued after the second time increment because of the errors from the wall line (curve 9) become unbearable to further produce any meaningful result. The analog sealing for the wall line leads to extremely high analog loop gain which not only can reduce the accuracies of the analog results but the moderate noise may even become a disaster.

Transient concentration profiles (Figures 5.3.4, 5.3.6 and Figures F-3, F-4) show significant differences between the non-isothermal and isothermal cases, although material balance equations have the same parameters except in the reaction term of the non-isothermal case.

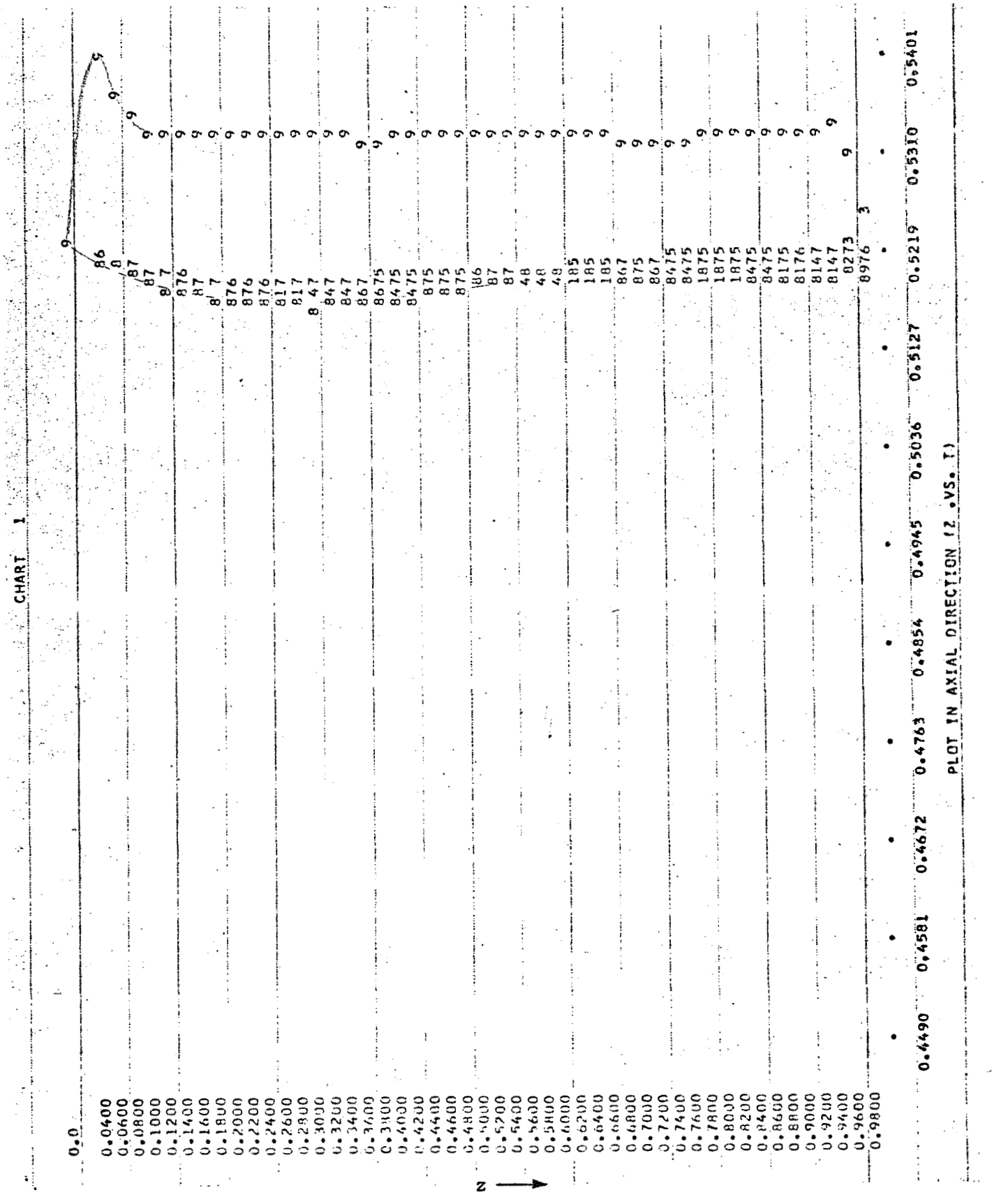


Figure 5.3.3 Transient Axial Temperature Profiles at one Δt

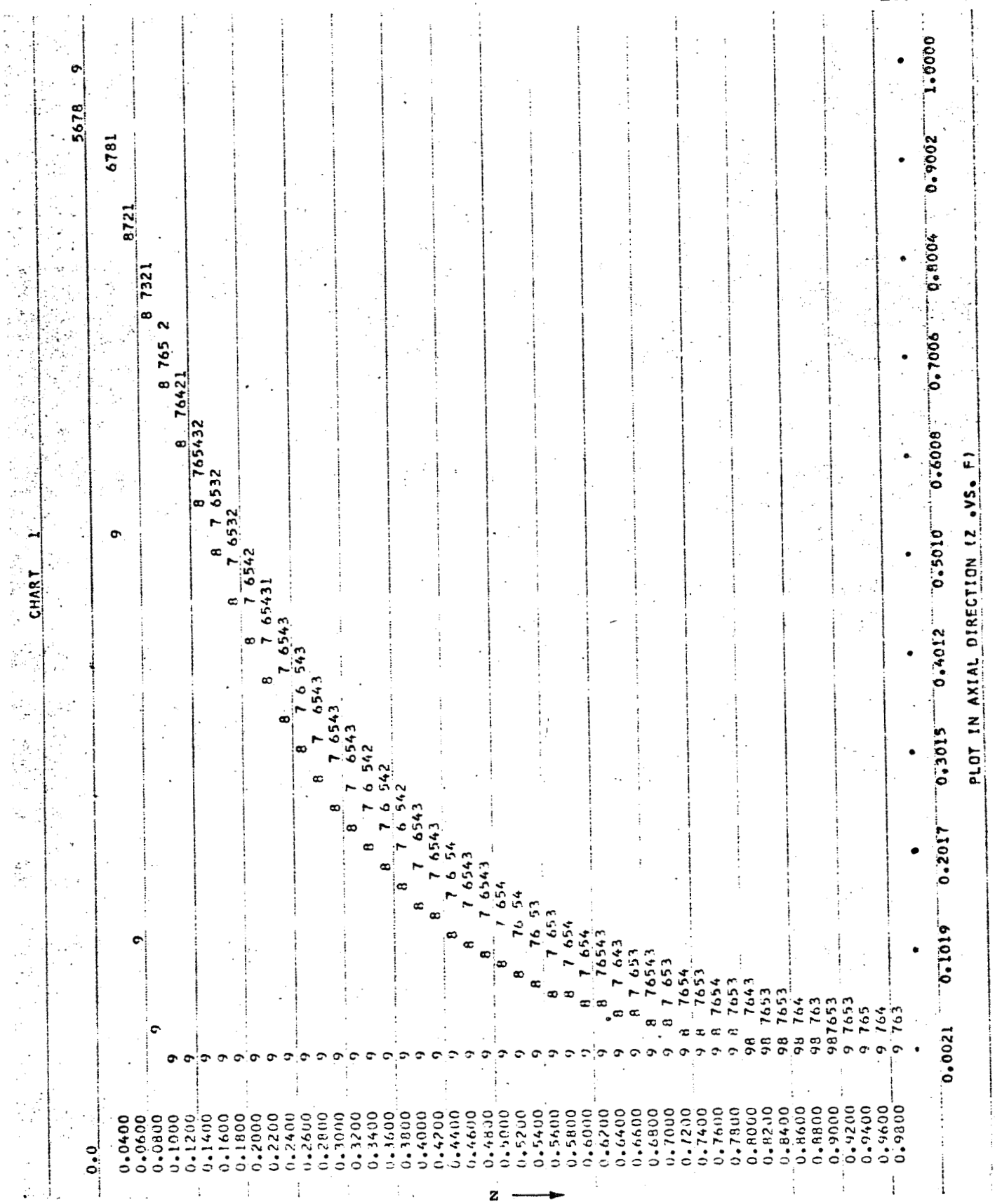


Figure 5.3.4 Transient Axial Concentration Profiles at one Δt

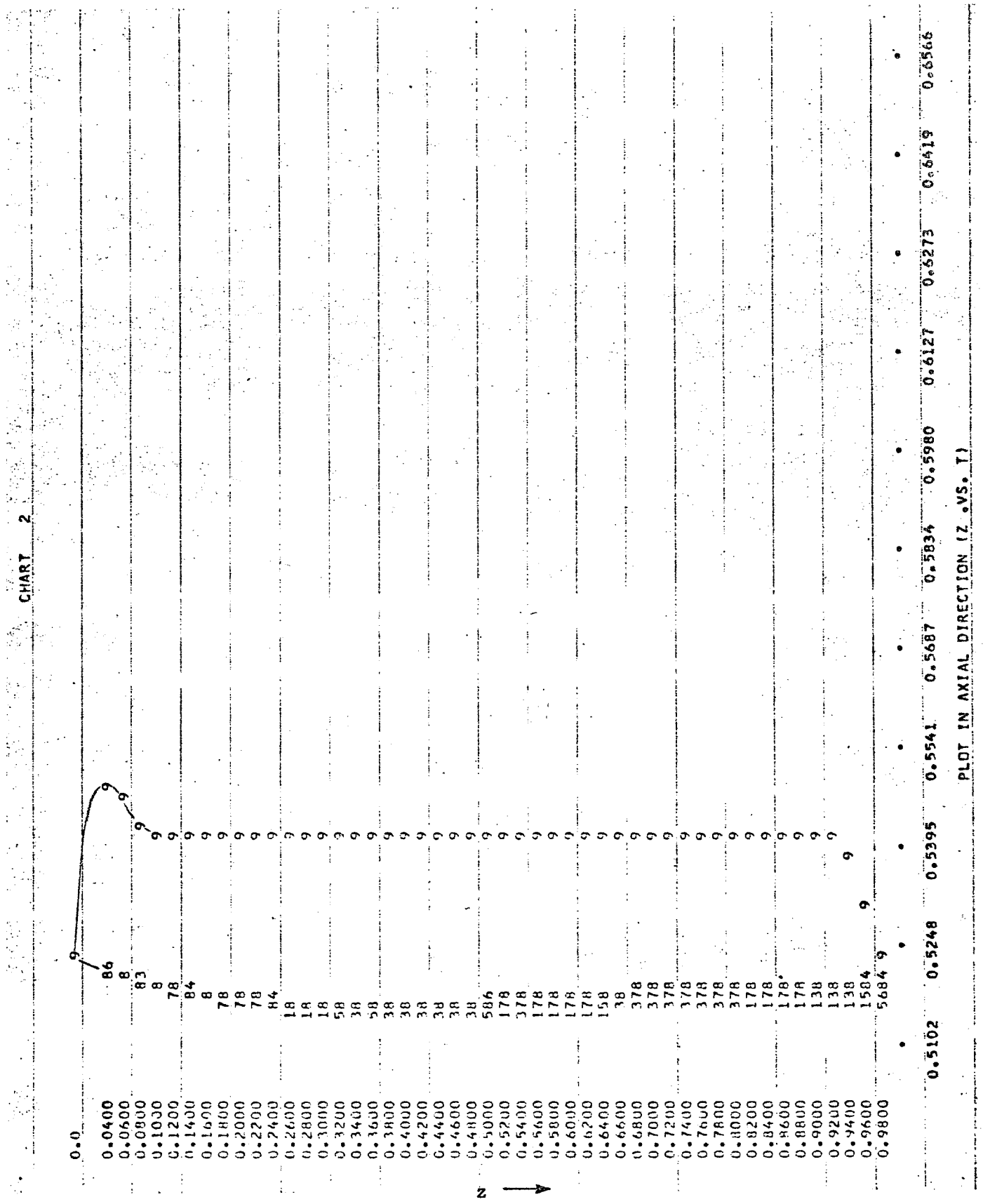


Figure 5.3.5 Transient Axial Temperature Profiles at $2\Delta t$

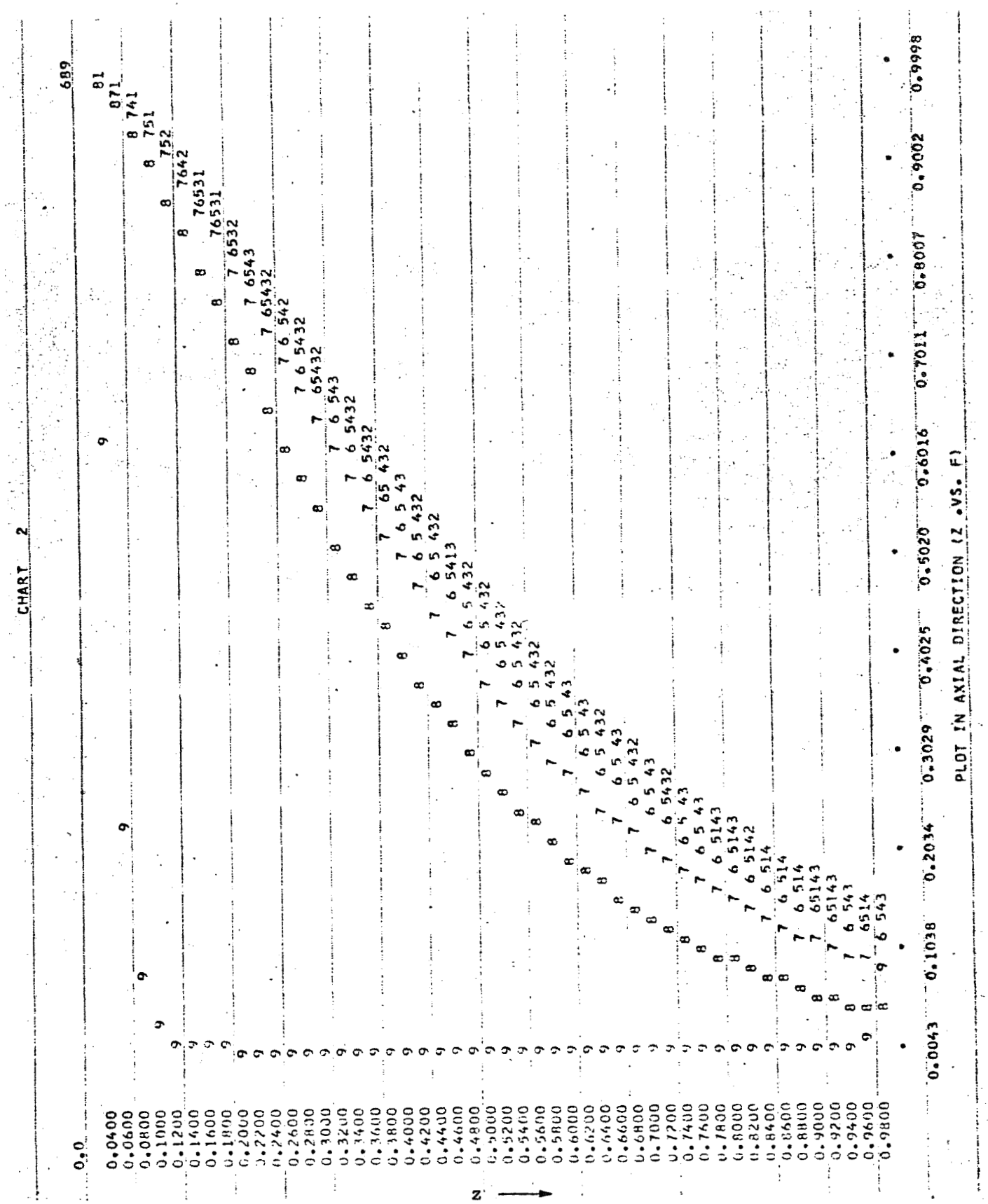


Figure 5.3.6 Transient Axial Concentration Profiles at $2\Delta t$

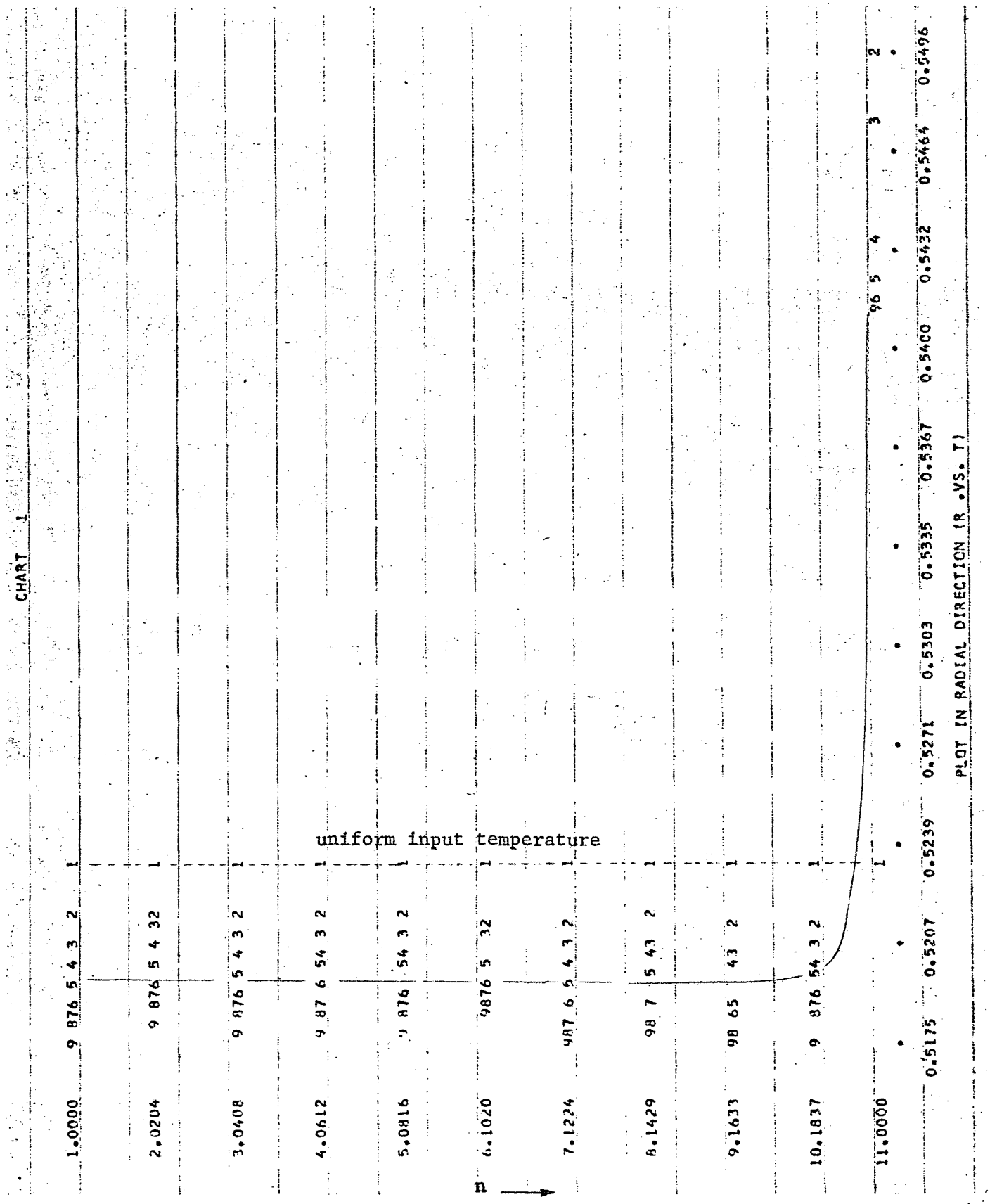


Figure 5.3.7 Transient Radial Temperature Profiles at $2\Delta t$ (axial index 1 ~ 9)

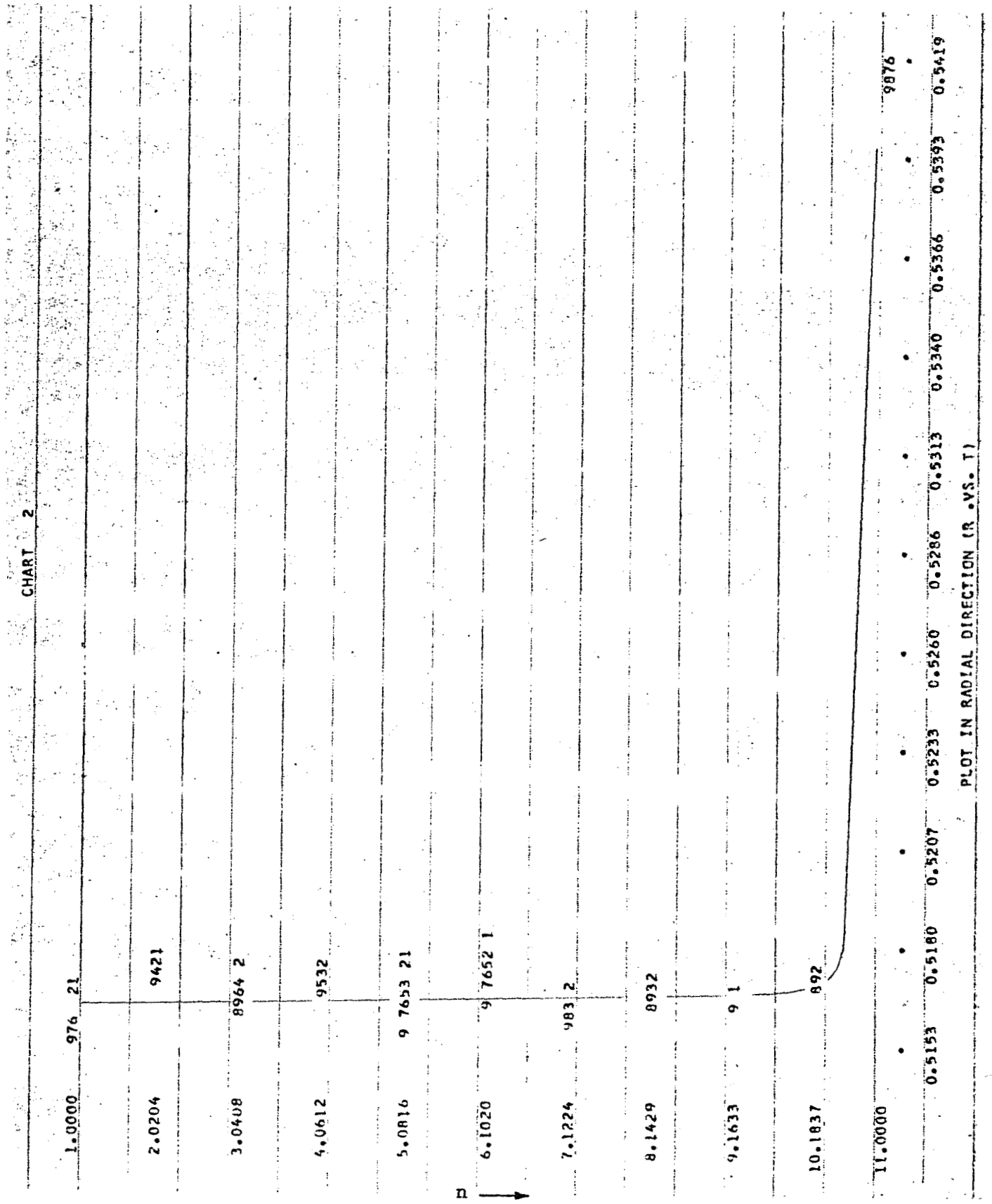


Figure 5.3.7 (Cont'd.) (axial index 10 ~ 18)

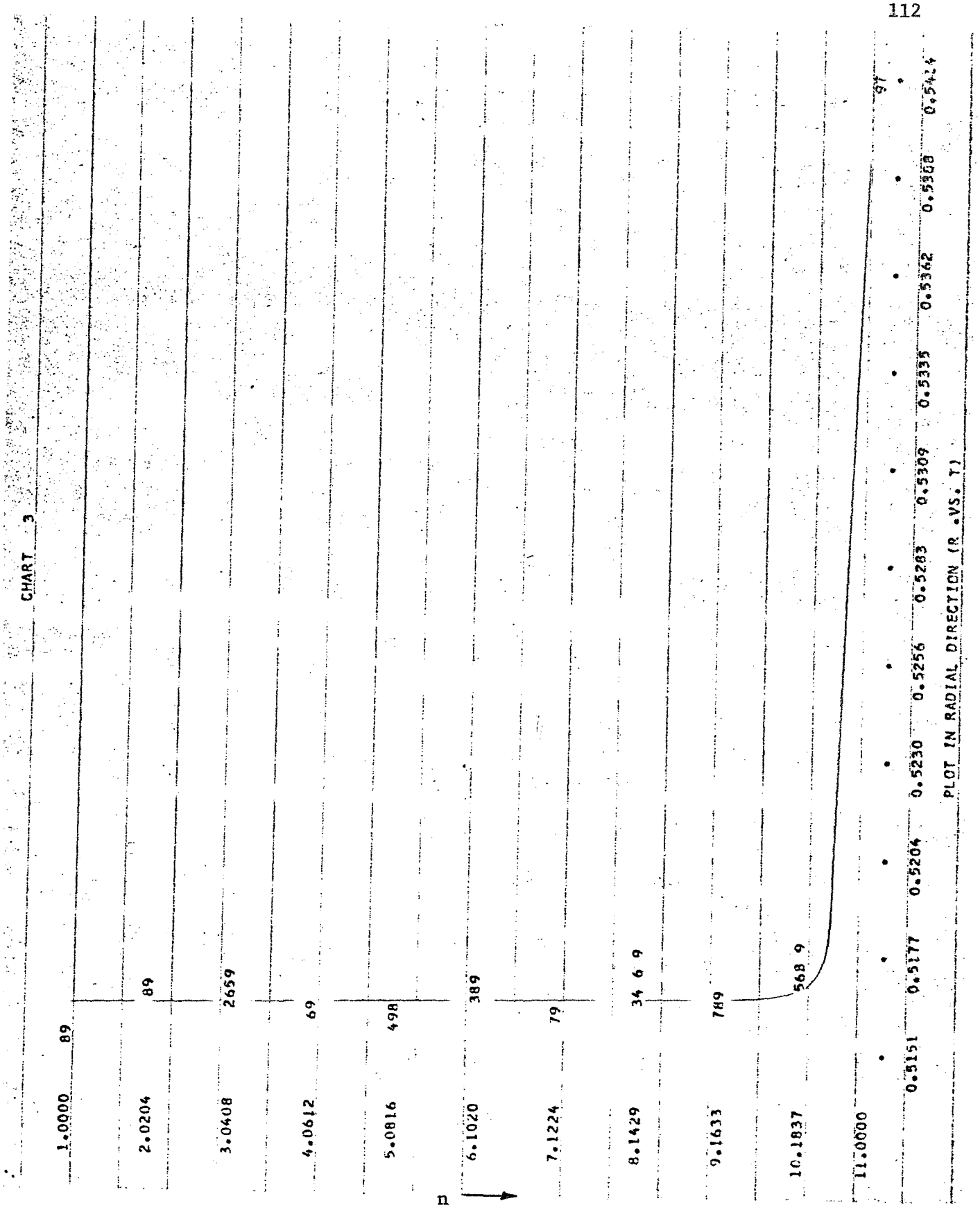


Figure 5.3.7 (Cont'd.) (axial index 19 ~ 27)

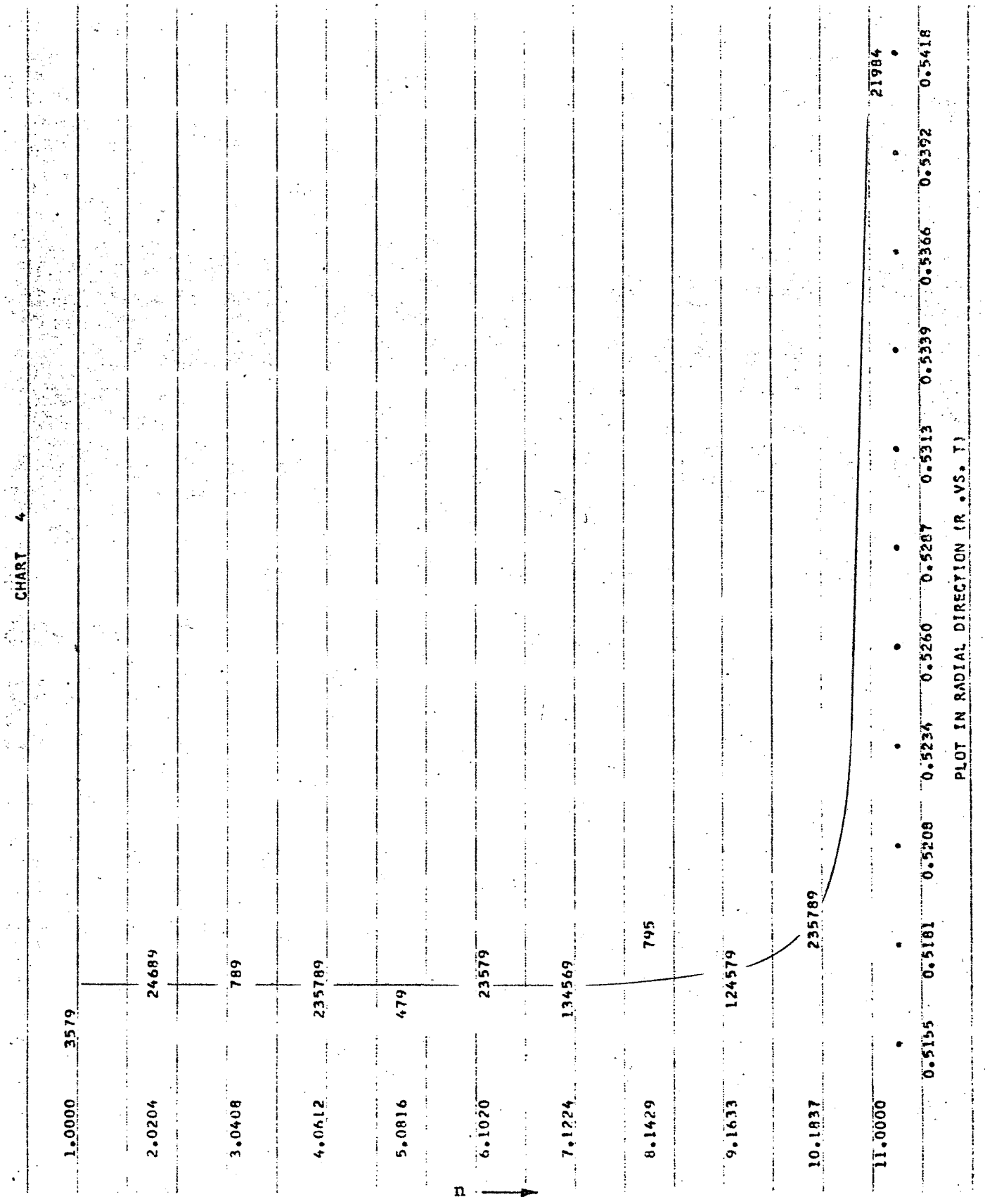


Figure 5.3.7 (Cont'd.) (axial index 28 ~ 36)

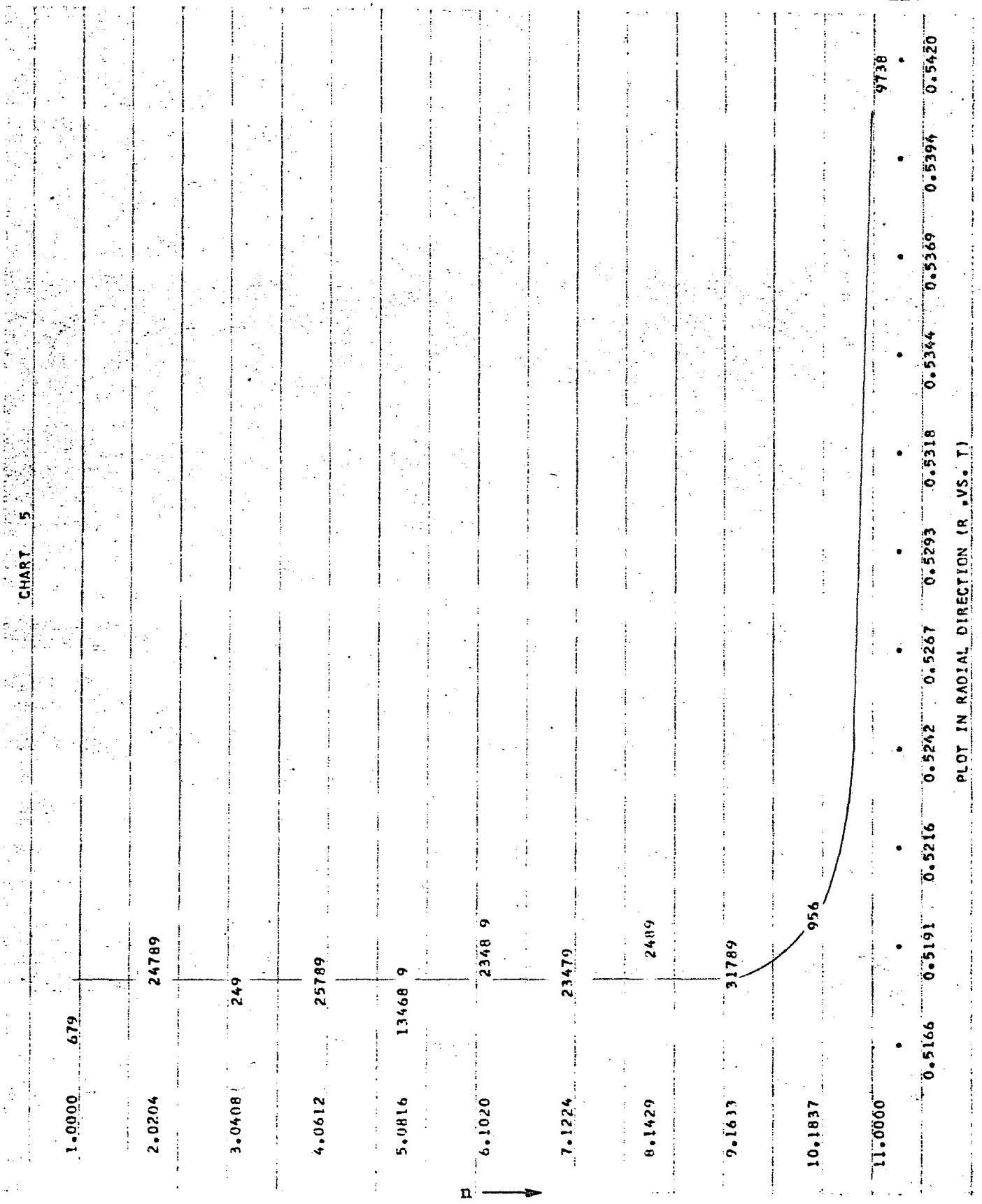


Figure 5.3.7 (Cont'd.) (axial index 37 ~ 45)

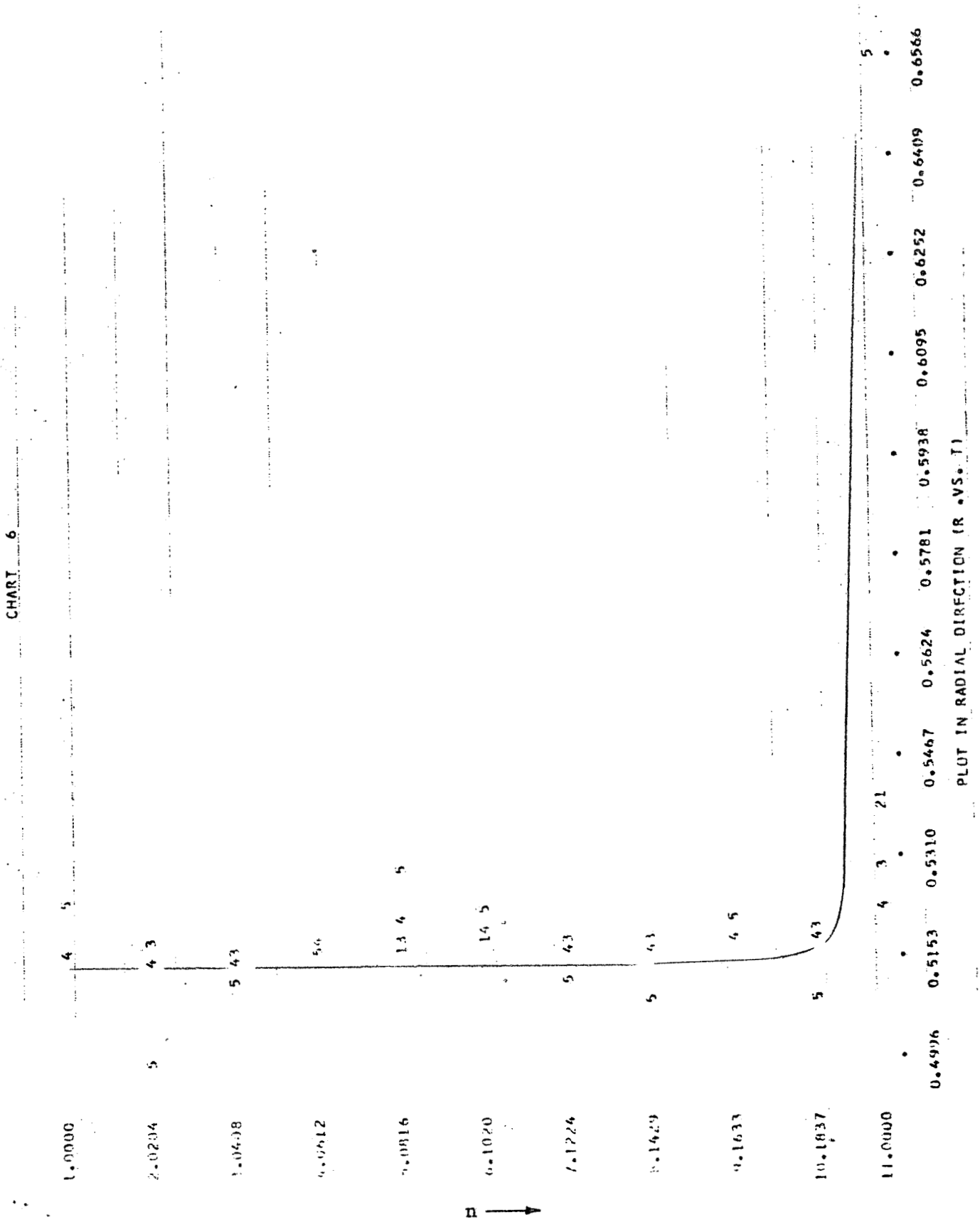


Figure 5.3.7 (Cont'd.) (axial index 46 ~ 50)

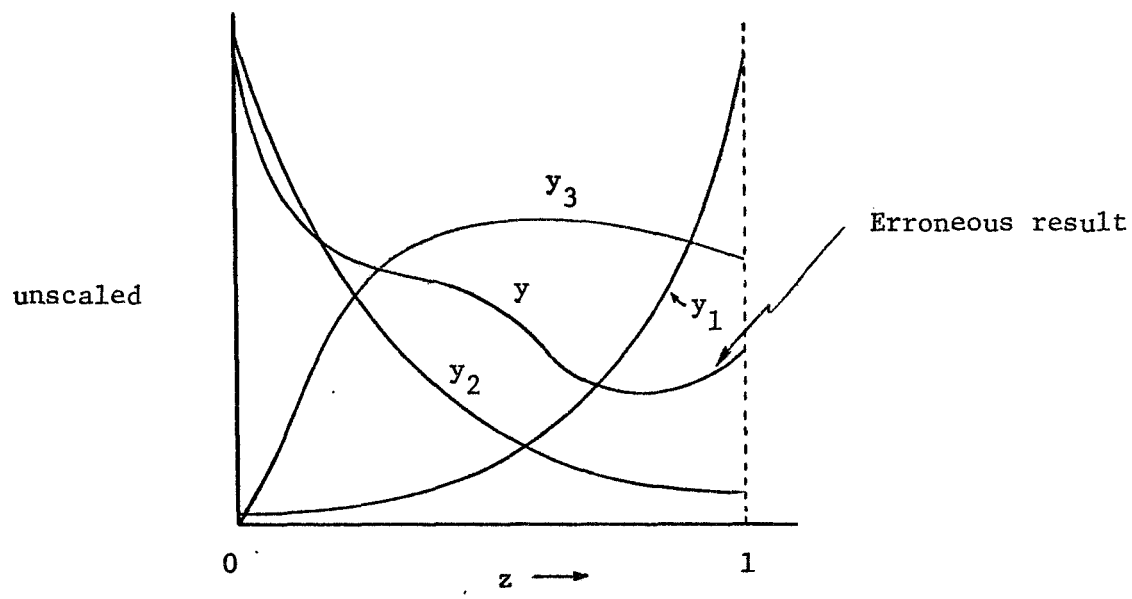
CHAPTER 6

CONCLUDING REMARKS

In the previous chapters, we have studied the application of hybrid techniques to both the steady-state and the unsteady-state solutions of the tubular flow chemical reactor. A problem in one space-dimension was first examined to establish the basis for problems in the two-space dimensions. In the steady-state case, a hybrid solution using the classical approach was obtained without difficulty at the rate of approximately 15 milliseconds (including IC, OP and Hold periods) per iteration. On the other hand, stability difficulties were encountered in solving the unsteady-state problem with the classical CSDT formulation. Among the possible non-classical approaches, the serial decomposition method was judged to be the most economical approach with the least complexity involved in the computational problems. Most important of all, the decomposition method yields a one pass solution for the linear boundary value equations, whereas iterative solution techniques generally lead to numerical difficulties due to inherent analog component inaccuracies.

Perhaps this problem of component inaccuracy associated with the various errors introduced by the involvement of interfacing equipment is by far the most serious difficulties one has to face in the hybrid computations. From our experience, the most probable errors normally occur at both ends of the data sequence being transmitted. Unfortunately, to solve the boundary value problem with the decomposition process, the general solution is calculated based on the results obtained from these

two terminal values. For example, the expression of general solution y is $y = ay_1 + by_2 + y_3$. Both y_1 and y_2 are time independent exponential functions, any significant error in either a or b will result error in the exponential order. This is shown in the following figure.



By examining the final apparent steady state values obtained from the case studies for different time increment, $\alpha\Delta t$ (refer to Table E-1), we can immediately come to a conclusion that the deviations of the predicted hybrid values from the true steady-state solutions are inversely proportional to the magnitude of the time increment, $\alpha\Delta t$, which seems contradictory to the basic assumption of the difference approximation process. This is mainly due to the inaccuracies that

exist in the overall hybrid loop. Since the smaller time increment will result a higher gain in the analog loop, it will also require more steps or successive calculations to reach the steady-state, hence, the error could also be magnified by summation over a larger number of iterations.

Sullivan, Bott and Gay (20) who used the same hybrid computing installation in their preliminary search for the partial differential equation solutions reported that their final steady-state hybrid values were consistently higher than the digital values. Hsu and Howe (10) also reported a similar experience. They concluded that this was likely due to digital/analog conversions not being transferred to the analog at the precise point in the solution at which it was previously generated. With the same zero-order extrapolation and the digital to analog playback function advanced in time by one-half the sampling period, Hsu and Howe were able to compensate the result for the average time delay, except that the hybrid method used was a DSCT method instead of the CSDT. However, this adjustment in synchronization exerted little or no effect on the results of the decomposition method, since the integrations are carried twice over the reversed directions.

As a matter of fact, the errors in the hybrid computation consist of many complex components. Vichnevetsky and Tomalesky (29) contributed a very interesting error analysis based on the first order correction of the CSDT approximation. By using the Taylor Series expansion and the CSDT approach to the simple fluid transport equation

$$\frac{\partial y}{\partial t} = -V \frac{\partial y}{\partial x} + D \frac{\partial^2 y}{\partial x^2} \tag{6.1}$$

to yield the expression

$$\frac{\partial y}{\partial t} = -V \frac{\partial y}{\partial x} + (\theta - 1/2)V^2 \Delta t \frac{\partial^2 y}{\partial x^2} + D \frac{\partial^2 y}{\partial x^2} + \ldots \qquad (6.2)$$

and they concluded that the approximation method introduced a truncation error which had the effect of "dispersing" the solution y by a diffusion-like term:

$$(\theta - 1/2)V^2 \Delta t$$

in addition to the actual diffusion term D. By adjusting θ and Δt an "exact" solution can be obtained.

Now, from the purely digital results tabulated here for $R_n = 1$ and $n = 1$

U	y(0)	y(1)
1	0.8388	0.6404
2	0.9001	0.6289
5	0.9547	0.6164
500	0.9995	0.6036

where $U = \frac{VL}{2D}$

there seems no conclusive relation existing between this table and the values in Table E-1. If the truncation-induced diffusion effect is to be the major factor one would expect that the trend in $y(1)$ values in Table E-1 should appear in the reversed order.

Now, if we take into account the reaction term and repeat the previous analysis, we have

$$\frac{\partial y}{\partial t} = -V \frac{\partial y}{\partial x} - Ry \tag{6.3}$$

where the diffusion term has been temporarily neglected.

The CSDT approximation of the above equation is

$$\frac{y^{i+1} - y^{i}}{\Delta t} = -\left[\theta (V \frac{dy^{i+1}}{dx} + Ry^{i+1}) + (1 - \theta) (V \frac{dy^{i}}{dx} + Ry^{i})\right] \tag{6.4}$$

by expressing the individual terms in a Taylor series around the point $y^{i}(x)$:

$$y^{i+1} = y^{i} + \frac{\partial y^{i}}{\partial t} \cdot \Delta t + \frac{\partial^2 y^{i}}{\partial t^2} \cdot \frac{\Delta t^2}{2} + \ldots. \tag{6.5}$$

$$\frac{dy^{i+1}}{dx} = \frac{\partial y^{i+1}}{\partial x} = \frac{\partial y^{i}}{\partial x} + \frac{\partial^2 y^{i}}{\partial x \partial t} \cdot \Delta t + \ldots. \tag{6.6}$$

Upon substitution of these expressions and omitting the index i, we have the CSDT approximation in the following form:

$$\frac{\partial y}{\partial t} + \frac{\partial^2 y}{\partial t^2} \cdot \frac{\Delta t^2}{2} + \ldots = -\Big\{\theta [V (\frac{\partial y}{\partial x} + \frac{\partial^2 y}{\partial x \partial t} \cdot \Delta t + \ldots) + R(y + \frac{\partial y}{\partial t} \cdot \Delta t + \ldots)] + (1 - \theta) (V \frac{\partial y}{\partial x} + Ry)\Big\} \tag{6.7}$$

Since we have for the exact solution

$$\frac{\partial y}{\partial t} = -V \frac{\partial y}{\partial x} - Ry \tag{6.3}$$

which yields

$$\frac{\partial^2 y}{\partial x \partial t} = -V \frac{\partial^2 y}{\partial x^2} - R \frac{\partial y}{\partial x} \tag{6.8}$$

and

$$\frac{\partial^2 y}{\partial t^2} = V^2 \frac{\partial^2 y}{\partial x^2} + 2VR \frac{\partial y}{\partial x} + R^2 y \tag{6.9}$$

Thus, again after substituting these relations we obtain:

$$\frac{\partial y}{\partial t} = -V \frac{\partial y}{\partial x} - Ry + (\theta - 1/2)\Delta t V^2 \frac{\partial^2 y}{\partial x^2} + 2(\theta - 1/2)\Delta t VR \frac{\partial y}{\partial x}$$

$$+ (\theta - 1/2)\Delta t R^2 y + \ldots. \tag{6.10}$$

By the identification of the equations, we find that not only does there exist an induced diffusion effect, but convection-like and reaction-like terms have also appeared. The value of θ has an important influence on the stability of the CSDT approximation.

When the serial method of hybrid integration is applied to problems in more than one space dimension, partial derivatives other than those in the space coordinate along which continuous integration takes place must be replaced by an approximation.

In two-space dimension problems, the alternating direction implicit method proposed in Section 4.1.1 will not require the approximation in the second space coordinate. However, the resulting first order equations may not always satisfy the specified boundary conditions. For this case, perhaps a method other than the decomposition technique should be recommended (9).

Various approaches have been suggested for the approximation in the second space coordinate:

1. The finite difference expression (CSDSDT) as we have employed in this work (Section 4.1.2).
2. Functional approximation by using the Ritz-Galerkin approach (30) or, elsewhere called the modal technique (26,27) replaces the partial derivatives in the second coordinate with prescribed functions of the modes in the second space coordinate. For instance, one may reduce the following equation

$$\frac{\partial y}{\partial t} = D\left(\frac{\partial^2 y}{\partial z} + \frac{\partial^2 y}{\partial r^2} + \frac{1}{r}\frac{\partial y}{\partial r}\right) + Ry \tag{6.11}$$

with the radial modes:

$$y(z,r,t) = \sum_n \phi_n (z,t) f_{mr}(r) \tag{6.12}$$

which leads to the eigenvalue problem:

$$\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r}\right) f_{nr} = \lambda_{nr} f_{nr} \tag{6.13}$$

By obtaining λ_{nr}, we obtain

$$\frac{\partial \phi_n}{\partial t} = \left[D\left(\frac{d^2}{dz^2} + \lambda_{nr}\right) + R\right] \phi_n \tag{6.14}$$

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

which is a set of N ordinary differential equations in the z-coordinate, which can be integrated using the CSDT approach. In the general application, however, this latter method is complicated by the difficulties of mathematical formulation.

Despite the inaccuracies existing in hybrid computation, the results of the isothermal two-space dimensional problem exhibit the stable nature of the decomposition technique as well as the fantastic speed of the hybrid computer solution. There still remains the scaling problem which has been the main drawback of the analog computation. This most unwanted problem usually occurs at the place where most accurate results are needed. For example, in a turbulent flow tubular reactor, where most of the significant transport processes take place in the laminar boundary layer, analog scaling poses significant problems due to high gain requirements. We must cope with this difficulty as long as we attempt to solve the tubular reactor problem in two-space dimensions.

In the non-isothermal tubular reactor simulation the problems are basically identical to those in the isothermal case. However, the increased non-linearity and the coupling of two equations probably develops more inaccuracies and perhaps will generate additional stability problems. This is where future work should be concentrated.

In summary, in order to determine the validity of the hybrid computation various sources of errors should be carefully examined. Theoretically, the integration should be able to be conducted in a matter of one or two milliseconds, however, it is still limited by the hardware problem. Although the non-linear equations have not been explored in depth, a

successive linearization approach within the decomposition method described in Chapter 5 seems to behave rather well in our example run. With the non-linear components involved, such as the analog multipliers, the noise effect sometimes may become very critical.

In the previous work (14) the example problem used was a homogeneous gas phase first order decomposition of SO_2Cl_2 to SO_2 and Cl_2 in a tubular reactor and had the following boundary conditions:

1. Constant wall temperature
2. Constant temperature and concentration at the inlet.
3. Vanishing longitudinal gradients at the outlet.
4. A finite radial concentration gradient at the wall.

The steady-state solution was attempted on the digital computer. Although final convergence was not reached in the previous study, the results indicated that the contribution of the boundary layer in this reacting system was more important than that obtained by a mere super-position of individual heat and mass transfer phenomena onto the chemical reaction process. The hybrid simulation of the non-isothermal case in this work, employed an almost identical example, strongly supporting the foregoing conclusion, though the boundary conditions and the space-increment sizes may be somewhat different.

The effect of the thermal conditions upon the reacting system was clearly illustrated by the concentration profiles of the isothermal and non-isothermal cases. When the reaction is endothermic, the velocity and temperature profiles work together to cause more radial transport of matter as shown in the non-isothermal profiles, while in the

isothermal case the radial concentration profiles are indeed flatter. For the exothermic reaction it was found that the lower velocity near the wall tended to flatten the concentration profile.

The most important application of this CSDT hybrid approach, although not yet fully realized, appears to be in the sensitivity and stability analysis of tubular reactor systems. These problems can be very time-consuming on the digital computer and usually require very small time steps and space increments in order to be able to pinpoint the precise transition which might lead to some multiple equilibrium states. The serial hybrid decomposition method illustrated throughout this work is continuous at least in one space coordinate and permits small time increments as long as the analog circuit scaling will accept it. The immediate problem still rests on the reliability and accuracy of the hardware components and the interfacing equipment. Unless these can be satisfied, the hybrid computer solution of the partial differential equations is a frustrating task.

BIBLIOGRAPHY

1. Amundson, N. R. and Dan Luss, "Qualitative and Quantitative Observations on the Tubular Reactor", The Canadian Jr. of Chem. Eng., 46, 424-433, Dec. 1968.

2. Bishop, K. and D. W. Green, AIChE Jr., 16, 193 (1970).

3. Braun, K. N. and A. B. Clymer, "Assumed Mode Methods for Structural Analysis and Simulation", 1963 Proc. Conf. on Simulation for Aerospace Flight, pp. 244-260.

4. Chan, S. K., "Analog-Digital Methods of Solving Partial Differential Equations", Final Report ESL-FR-330, M.I.T. DSR Projects 79128 and 75161, Contract NOnr-1841(85), Oct. 1967.

5. Chan, S. K., "The Serial Solution of the Diffusion Equation Using Nonstandard hybrid techniques," IEEE Trans. on Computers, Vol. C-18, No. 9, Sept. 1969.

6. Coulman, G. A., J. F. Svetlik and W. H. Clifford, AFIPS Conf. Proc. Vol. 33, 1968, FJCC, p. 593 ff.

7. Fan, L. T. and R. C. Bailie, Chem, Eng. Sci., 13, 63-68 (1960).

8. Handler, H., IEEE Trans. on Electronic Computers, EC-16, 608

9. Hara, H. H. and W. J. Karplus, "Application of Functional Optimization Techniques for the Serial Hybrid Computer Solution of Partial Differential Equations", AFIPS Proc. FJCC 1968, 33, pt. 1, pp. 565-574.

10. Hsu, S. K. T. and R. M. Howe, "Preliminary Investigation of a Hybrid Method for Solving P.D.Es", AFIPS Proc. FJCC, 1968, pp. 601-609.

11. Hybrid Executive, Programming Guide, Hybrid Systems, Inc., Houston, Texas, Publication 6033.2, 10/68.

12. Jermann, W. H., Report to NASA, Computation Lab., Simulation Branch, Contract No. NSR-01-002-033, Aug. 15, 1969.

13. Jury, S. H., "Solving Partial Differential Equations", I & EC, 53, 177-180 (1961).

14. Lee, H. M., M.S. Thesis, Department of Chemical Engineering, University of Houston, (1962).

15. Levenspiel, O. and K. B. Bischoff, I & EC, 51, 1431 (1959).

16. Nelson, P. Jr., "Application of Invariant Imbedding to the Solution of P.D.Es by CSDT Method", AFIPS Proc. SJCC 1970, pp. 397-402.

17. Newman, D. J. and J. C. Strauss, "Hybrid Assumed Mode Solution of Non-linear P.D.Es", AFIPS Proc. FJCC 1968, 33, pt. 1, pp. 575-583.

18. Peaceman, D. W. and H. H. Rachford, Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations", J. Soc. Inc. Appl. Math., 3, 28 (1955).

19. Snyder, L. J., T. W. Spriggs, and W. E. Stewart, AIChE Jr. 10, No. 4, 535 (1964).

20. Sullivan, S. S. Jr., J. L. Bott, and J. N. Gay, "The Search for a More Economical Solution to P.D.Es Using Hybrid Computer Techniques - A Preliminary Investigation", Hybrid Systems, Inc., Houston, Texas (1969).

21. System/360 Continuous System Modeling Program (360A-CX-16X), User's Manual, IBM Application Program H20-0367-1.

22. System/360 Model 44 (DAMPS),(360A-CX-20X) Version 2, Program Description Manual, IBM Application Program H20-0537-1.

23. Vichnevetsky, R., "Error Analysis in Computer Simulation of Dynamic Systems: Variational Aspects of the Problem", IEEE, EC-16, No. 4, 403-411, Aug. 1967.

24. Vichnevetsky, R., "A New Stable Computing Method for the Serial Hybrid Computer Integration of Partial Differential Equations", AFIPS, Proc. SJCC 1968, pp. 143-150.

25. Vichnevetsky, R., "Application of Hybrid Computers to the Integration of P.D.Es of the First and Second Order", IFIP Congress 68, 1-8, Aug. 5-10, 1968.

26. Vichnevetsky, R., "Analog/Hybrid Solution of P.D.Es in the Nuclear Industry", Simulation, 269-281, Dec. 1968.

27. Vichnevetsky, R., "Use of Functional Approximation Methods in the Computer Solution of Initial Value Partial Differential Equation Problems", IEEE Trans. on Computers, Vol. C-18, No. 6, pp. 499-512, June, 1969.

28. Vichnevetsky, R., "Serial Solution of Parabolic P.D.Es: The Decomposition Method for Non-linear and Space-dependent problems", Simulation, 47-48, July, 1969.

29. Vichnevetsky, R. and A. W. Tomalesky, "A Hybrid Computer Method for the Analysis of Time Dependent River Pollution Problems", AFIPS Proc. SJCC 1970, pp. 43-49.

30. Vichnevetsky, R. and A. W. Tomalesky, "Hybrid Computer Programs for the Study of Pollution in Rivers and Estuaries", Proc. SCSC 1970, Vol. 2, pp. 922-924.

31. Witsenhausen, H. S., "On the Hybrid Solution of P.D.Es", Proceedings of IFIP Congress 1965, Vol. 2, pp. 425-431.

32. Wozny, M. J., P. S. Bartells and R. E. Bailey, "Hybrid Computer Simulation of Xenon Spatial Oscillators", Simulation, 239, May 1970.

33. Yaws, C. L., Ph.D. thesis, University of Houston (1965).

APPENDIX A

Finite Difference Solutions of the Steady-State One Dimensional Isothermal Tubular Flow Reactor

Equation: $$\frac{d^2y}{dz^2} - 2U\frac{dy}{dz} - UR_n y^n = 0$$

Boundary Conditions:

$$z = 0 \; ; \; \frac{dy}{dz} = 2U[y_{0+} - 1]$$

$$z = 1 \; ; \; \frac{dy}{dz} = 0$$

Parameters:

$$R_n = 0.2,\ 1.,\ 10.$$

$$U = 1,\ 2,\ 5,\ 500$$

$$n = 1/4,\ 1/2,\ 1,\ 2,\ 3$$

Legend:

Curve	U
1	1
2	2
3	5
4	500

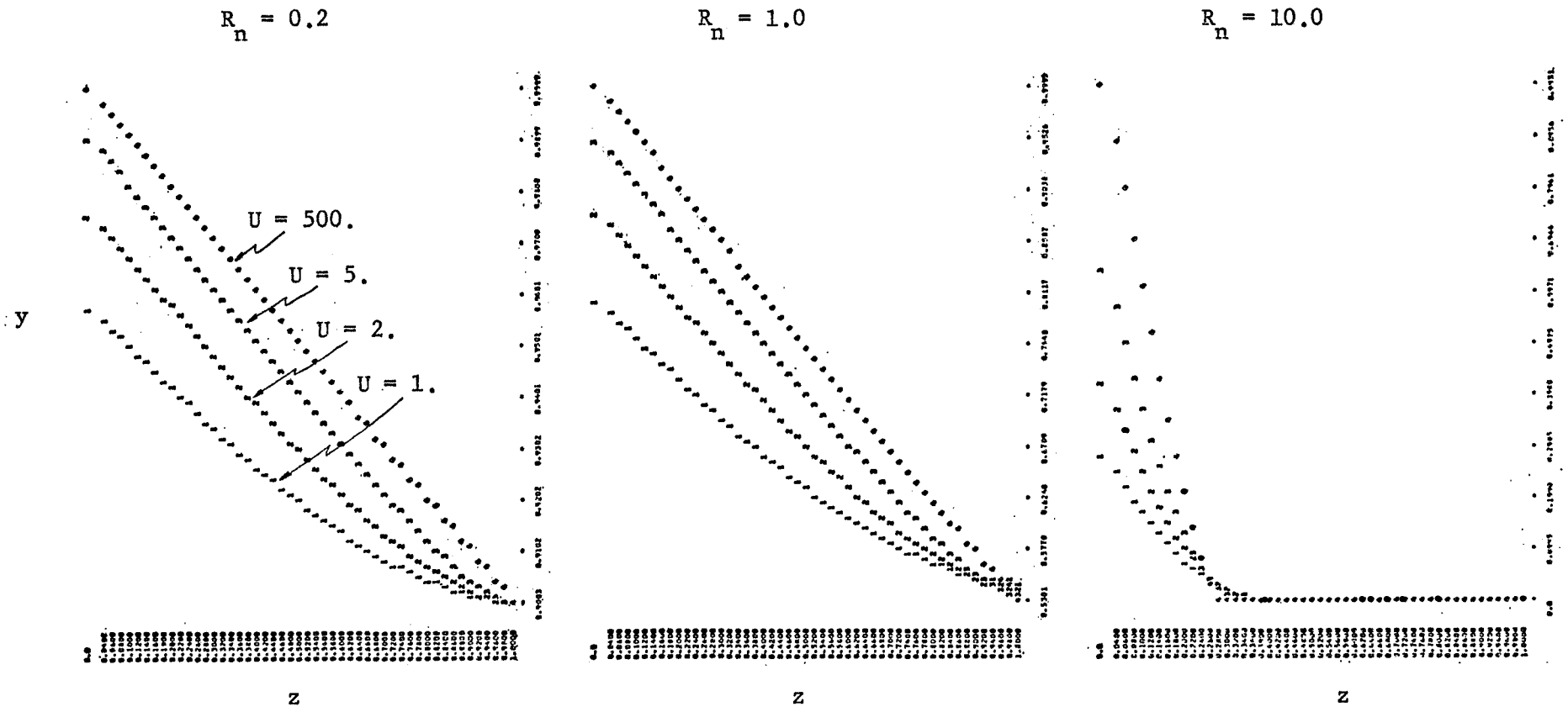


Figure A-1 Concentration Profile of Reactant in Reactor (y_0- = 1.0) for n = 1/4, U = Parameter

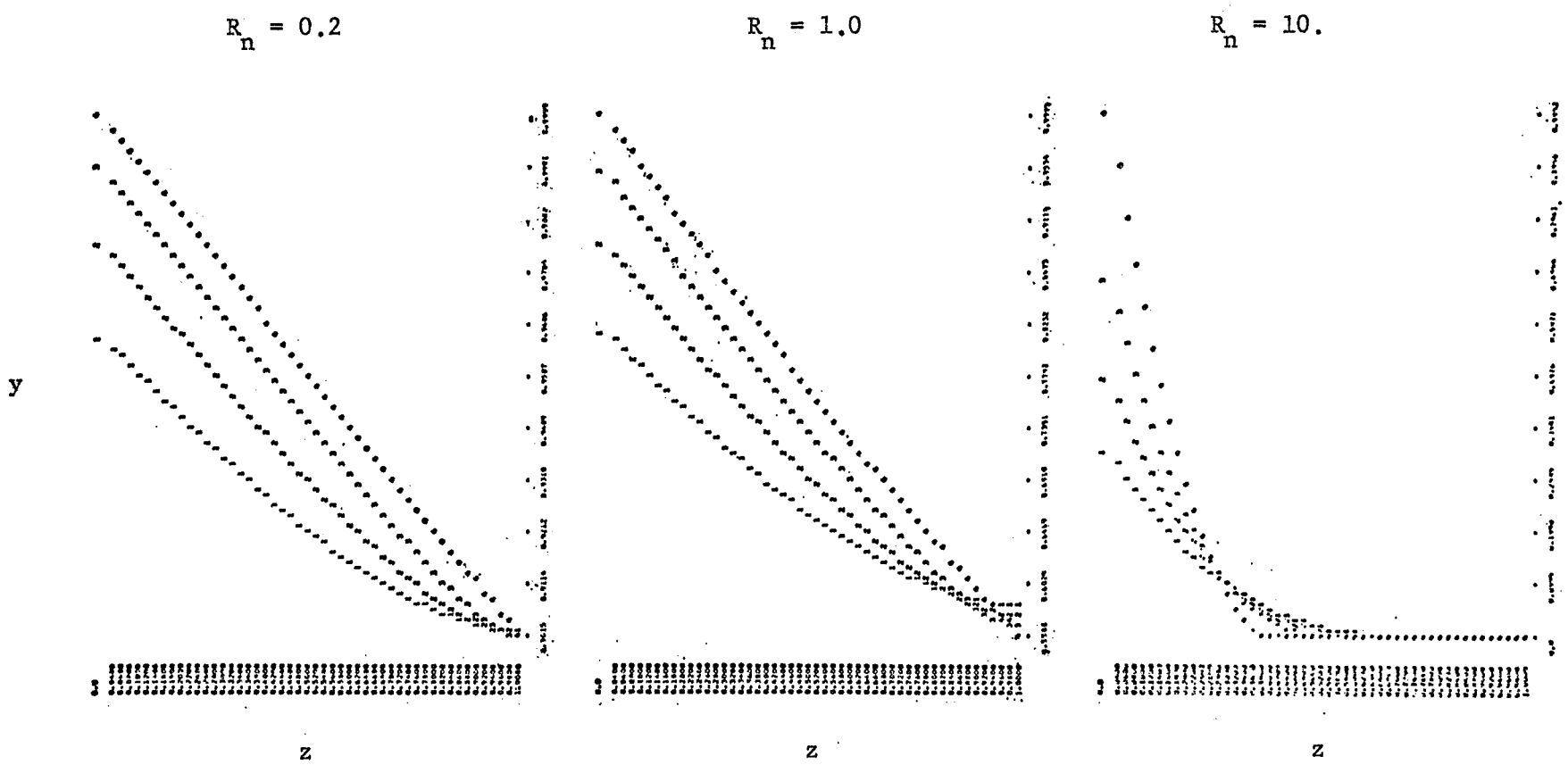


Figure A-2 Concentration Profile of Reactant in Reactor (y_0- = 1.0) for n = 1/2, U = Parameter

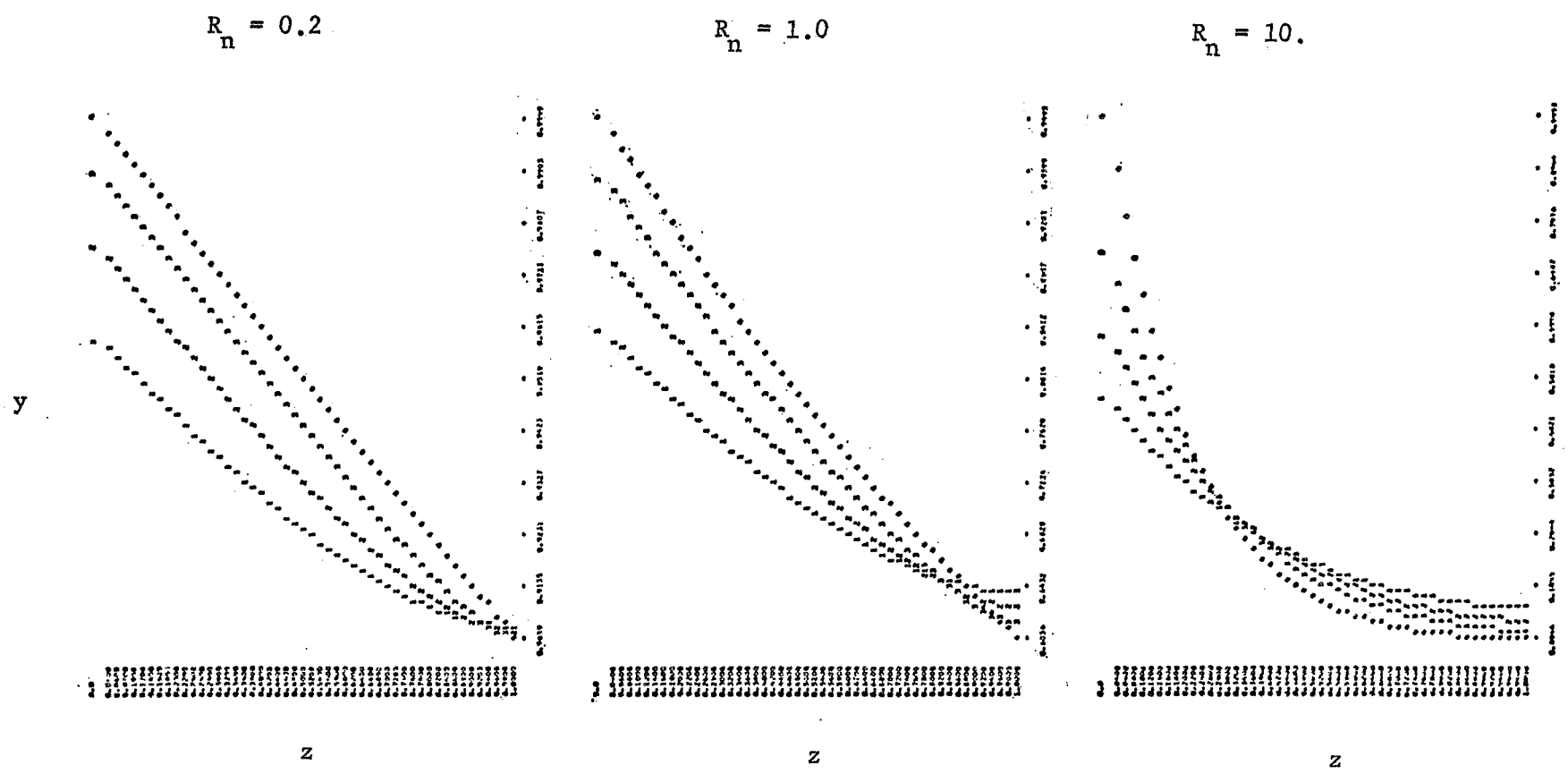


Figure A-3 Concentration Profile of Reactant in Reactor (y_{0^-} = 1) for n = 1, U = Parameter

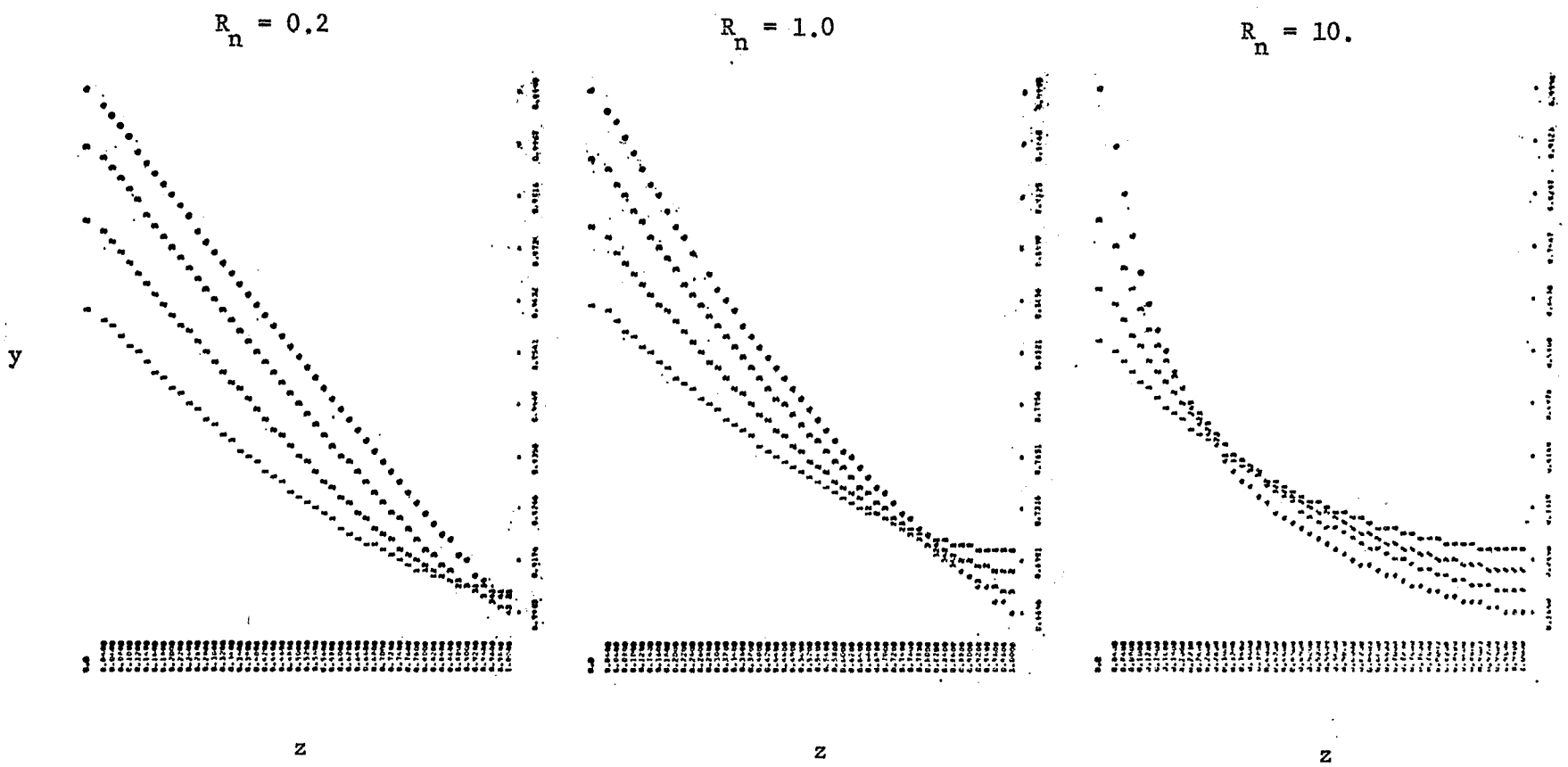


Figure A-4 Concentration Profile of Reactant in Reactor (y_0- = 1.0) for n = 2, U = Parameter

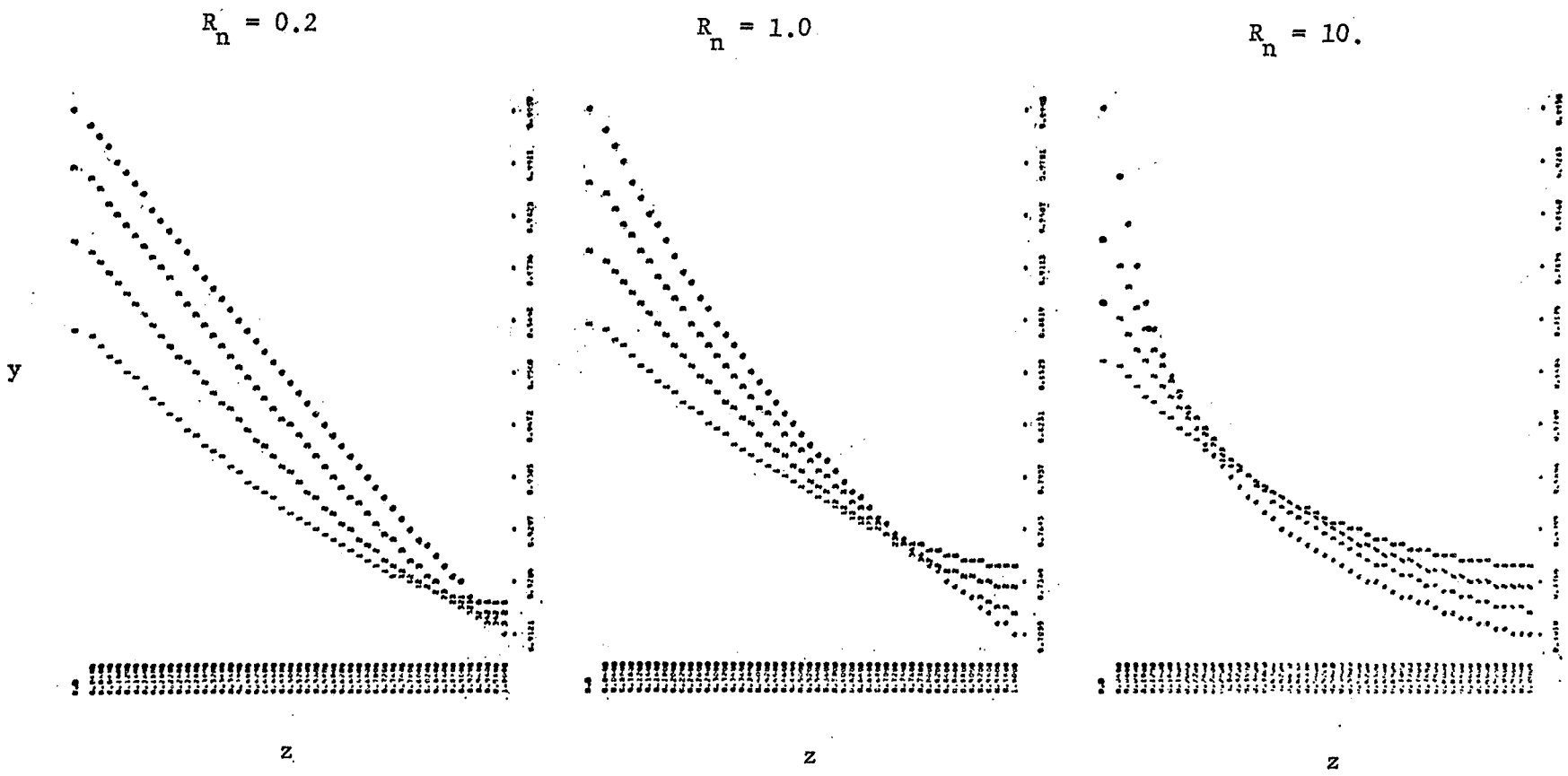


Figure A-5 Concentration Profile of Reactant in Reactor (y_{0-} = 1.0) for n = 3, U = parameter

APPENDIX B

CSMP (21) Prediction of the Steady-State Solutions Integrating in the Forward Direction

In order to show the difficulties involved in the forward integration, two of the typical unsuccessful results corresponding to the curves A and B of Figure 2.1 are presented in the following figures.

Equation: $\ddot{y} - 2U\dot{y} - UR_n y^n = 0$

Boundary conditions:

$$\dot{y} = 2U[y_{0+} - 1] \; ; \quad x = 0$$

$$\dot{y} = 0 \; ; \quad x = 1$$

Legend:

YY represents y

DYDX represents $\dot{y}$

MINIMUM 8.3136E-01 YY VERSUS X MAXIMUM 9.9005E-01

X	YY	DYDX
0.0	9.9005E-01	-9.9525E-02
2.5000E-02	9.8756E-01	-9.9470E-02
5.0000E-02	9.8507E-01	-9.9417E-02
7.5000E-02	9.8259E-01	-9.9366E-02
1.0000E-01	9.8011E-01	-9.9319E-02
1.2500E-01	9.7762E-01	-9.9277E-02
1.5000E-01	9.7514E-01	-9.9240E-02
1.7500E-01	9.7266E-01	-9.9211E-02
2.0000E-01	9.7018E-01	-9.9191E-02
2.2500E-01	9.6770E-01	-9.9185E-02
2.5000E-01	9.6522E-01	-9.9194E-02
2.7500E-01	9.6274E-01	-9.9224E-02
3.0000E-01	9.6026E-01	-9.9280E-02
3.2500E-01	9.5778E-01	-9.9371E-02
3.5000E-01	9.5529E-01	-9.9505E-02
3.7500E-01	9.5280E-01	-9.9696E-02
4.0000E-01	9.5031E-01	-9.9960E-02
4.2500E-01	9.4780E-01	-1.0032E-01
4.5000E-01	9.4529E-01	-1.0079E-01
4.7500E-01	9.4276E-01	-1.0142E-01
5.0000E-01	9.4022E-01	-1.0225E-01
5.2500E-01	9.3765E-01	-1.0334E-01
5.5000E-01	9.3505E-01	-1.0475E-01
5.7500E-01	9.3240E-01	-1.0658E-01
6.0000E-01	9.2971E-01	-1.0895E-01
6.2500E-01	9.2695E-01	-1.1202E-01
6.5000E-01	9.2410E-01	-1.1597E-01
6.7500E-01	9.2114E-01	-1.2107E-01
7.0000E-01	9.1804E-01	-1.2765E-01
7.2500E-01	9.1475E-01	-1.3611E-01
7.5000E-01	9.1121E-01	-1.4701E-01
7.7500E-01	9.0737E-01	-1.6103E-01
8.0000E-01	9.0313E-01	-1.7906E-01
8.2500E-01	8.9837E-01	-2.0224E-01
8.5000E-01	8.9296E-01	-2.3205E-01
8.7500E-01	8.8670E-01	-2.7037E-01
9.0000E-01	8.7935E-01	-3.1963E-01
9.2500E-01	8.7060E-01	-3.8294E-01
9.5000E-01	8.6005E-01	-4.6431E-01
9.7500E-01	8.4719E-01	-5.6888E-01
1.0000E 00	8.3136E-01	-7.0327E-01

Figure B-1 CSMP Prediction of Concentration Profile for 1/4 Order Reaction, $R_n = 0.2$, $U = 5.0$

MINIMUM 9.5939E-01 YY VERSUS X MAXIMUM 1.0000E 00

X	YY	DYDX
0.0	9.9501E-01	-4.9858E-01
2.5000E-02	9.8258E-01	-4.9547E-01
5.0000E-02	9.7036E-01	-4.7494E-01
7.5000E-02	9.6029E-01	-2.4282E-01
1.0000E-01	9.7610E-01	2.5764E 00
1.2500E-01	1.0000E 00	3.6959E 01
1.5000E-01	1.0000E 00	4.5573E 02
1.7500E-01	1.0000E 00	5.5562E 03
2.0000E-01	1.0000E 00	6.7678E 04
2.2500E-01	1.0000E 00	8.2429E 05
2.5000E-01	1.0000E 00	1.0040E 07
2.7500E-01	1.0000E 00	1.2228E 08
3.0000E-01	1.0000E 00	1.4893E 09
3.2500E-01	1.0000E 00	1.8139E 10
3.5000E-01	1.0000E 00	2.2092E 11
3.7500E-01	1.0000E 00	2.6907E 12
4.0000E-01	1.0000E 00	3.2772E 13
4.2500E-01	1.0000E 00	3.9915E 14
4.5000E-01	1.0000E 00	4.8614E 15
4.7500E-01	1.0000E 00	5.9210E 16
5.0000E-01	1.0000E 00	7.2115E 17
5.2500E-01	1.0000E 00	8.7833E 18
5.5000E-01	1.0000E 00	1.0698E 20
5.7500E-01	1.0000E 00	1.3029E 21
6.0000E-01	1.0000E 00	1.5869E 22
6.2500E-01	1.0000E 00	1.9328E 23
6.5000E-01	1.0000E 00	2.3540E 24
6.7500E-01	1.0000E 00	2.8671E 25
7.0000E-01	1.0000E 00	3.4920E 26
7.2500E-01	1.0000E 00	4.2531E 27
7.5000E-01	1.0000E 00	5.1801E 28
7.7500E-01	1.0000E 00	6.3092E 29
8.0000E-01	1.0000E 00	7.6843E 30
8.2500E-01	1.0000E 00	9.3591E 31
8.5000E-01	1.0000E 00	1.1399E 33
8.7500E-01	1.0000E 00	1.3883E 34
9.0000E-01	1.0000E 00	1.6909E 35
9.2500E-01	1.0000E 00	2.0595E 36
9.5000E-01	1.0000E 00	2.5084E 37
9.7500E-01	1.0000E 00	3.0551E 38
1.0000E 00	1.0000E 00	3.7209E 39

Figure B-2 CSMP Prediction of Concentration Profile for 1/4 Order Reaction, $R_n = 1.$, $U = 50.$

APPENDIX C

Steady-State Hybrid Solution of

$$\ddot{y} - 2U\dot{y} - UR_n y = 0$$

In order to mechanize our problem on the analog computer, the original equation is rewritten as:

$$\ddot{y} = 2U\dot{y} + UR_n y \qquad (z = 0 \text{ to } 1)$$

To conduct the integration in the reverse direction we simply invert the sign of the $\dot{y}$ term to obtain

$$\ddot{y} = -2U\dot{y} + UR_n y \qquad (z = 1 \text{ to } 0)$$

One of the advantages of inverting direction is that the initial condition of the first derivative is fixed at $z = 1$ (i.e., $\dot{y}_{z=1} = 0$).

Since the dynamic range of the variables is limited in the analog computation(-100 volts to +100 volts), the maximum value of each variable has to be predetermined for scaling purposes.

The estimated values are:

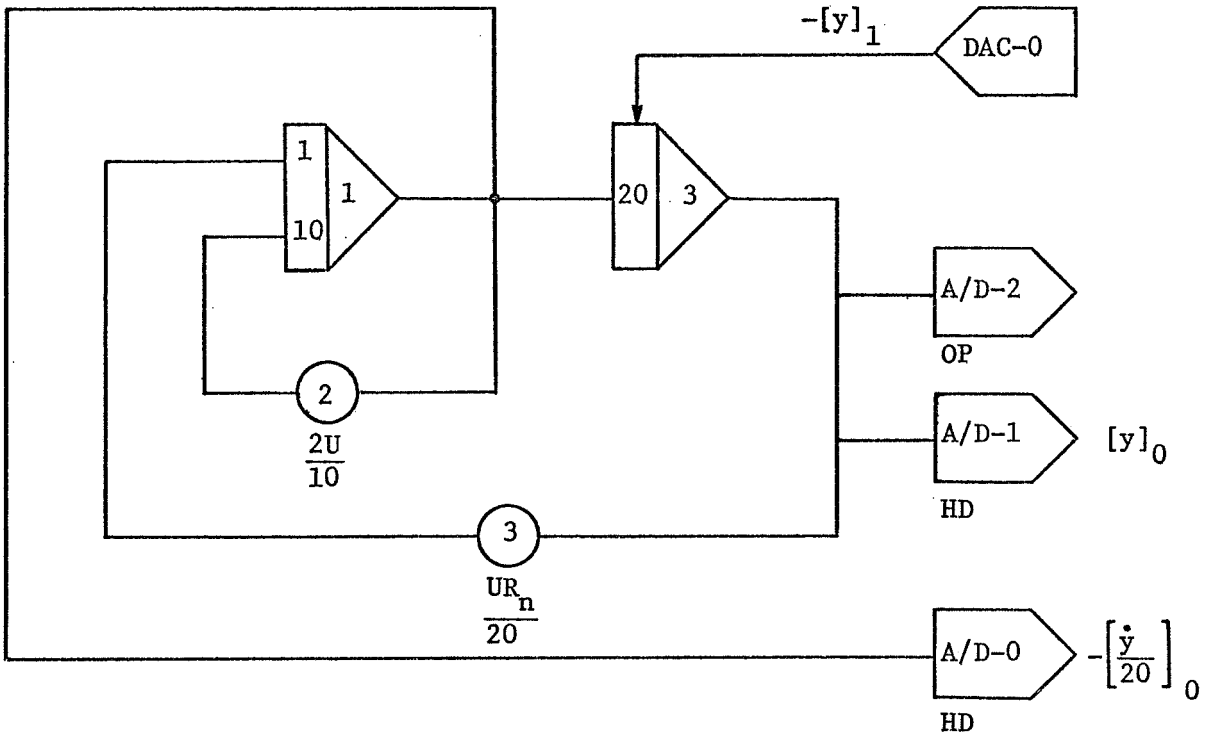
$$[\dot{y}]_{max} = 20$$

$$[y]_{max} = 1$$

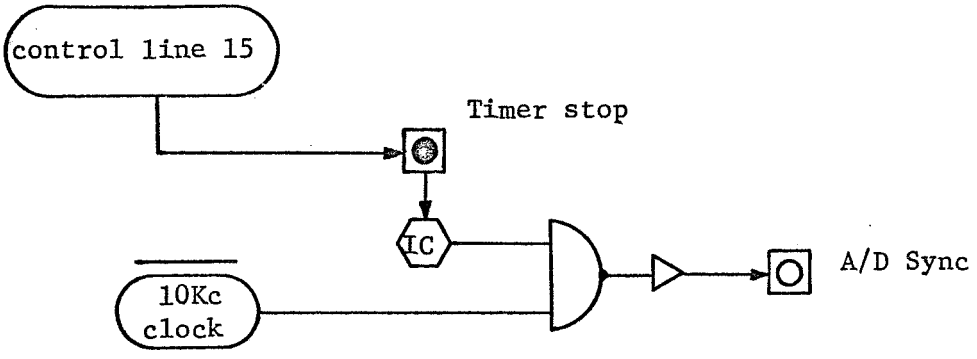
Then the following scaled equation is obtained

$$\left[\frac{\ddot{y}}{20}\right] = -2U\left[\frac{\dot{y}}{20}\right] + \frac{UR_n}{20}[y]$$

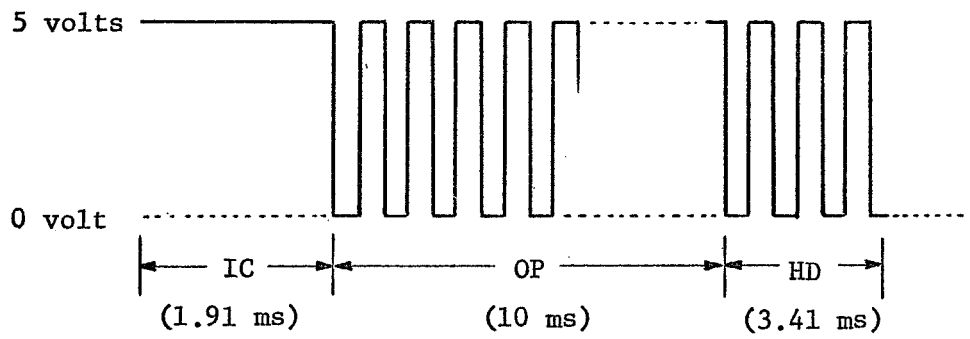
This gives the following mechanized analog diagram



The initial value of y at $z = 1$ is fetched through DAC-0 from the digital program. The profile is sampled during the operating period, and when it reaches $z = 0$ a sufficient holding time is required to sample y and $\dot{y}$. For this, following logic diagram is wired to synchronize the clock pulse with the integration schedule so that the A/D sampling will be triggered at the right moment.



When the timer is started, the following signals control the A/D operation.



Operation with a 100 KC clock signal and 1 millisecond operating period was not successful due to insufficient settling time for the capacitors.

Using the boundary condition at $z = 0$ and $\dot{y}$ at $z = 0$ obtained from A/D - 0 we can calculate y at $z = 0$:

$$[y]_{0_{calc.}} = \frac{10}{U}\left[\frac{\dot{y}}{20}\right]_0 + 1 \qquad \text{(step input)}$$

The initial condition $[y]_{z=1}$ is simply up-dated by

$$[y]_1^{i+1} = [y]_1^i + \frac{[y]_{0\ calc.} - [y]_{0\ measured}}{10}$$

Figure C-1 depicts the digital algorithm which controls the entire hybrid process. The digital program using DAMPS2 and the Hybrid Executive is listed in Listing C. Tables C-1 and C-2 show the results obtained from the following conditions:

$U = 1$

$R_n = 1$

Timer : IC = 1.91 msec.

OP = 10 msec.

HD = 3.41 msec.

Capacitor : 0.01 mfd.

Clock : 10 KC

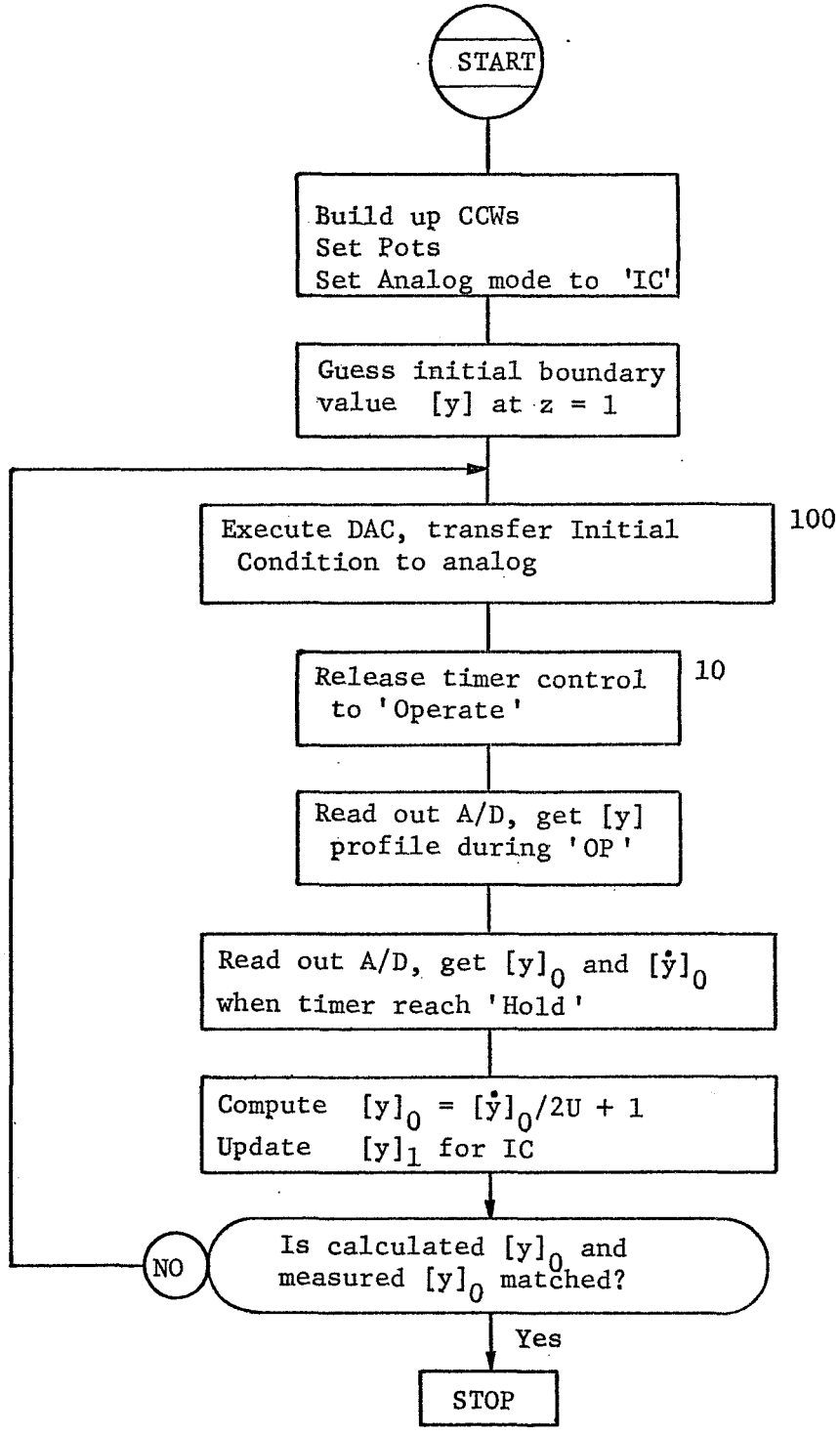


Figure C-1 Flow Chart for the Digital Portion of the Steady-State Hybrid Simulation

TABLE C-1 Result of Steady-State Hybrid Simulation

Iteration No.	$-[\dot{y}_0]$ measured	$[y_0]$ measured	$[y_0]$ calc.	$[y_1]$ calc.	Remarks
1	0.2539	0.6554	0.8730	0.5218	
2	0.2637	0.6799	0.8681	0.5406	
3	0.2808	0.7043	0.8596	0.5561	
4	0.2856	0.7248	0.8572	0.5694	
5	0.2906	0.7422	0.8547	0.5806	
10	0.3101	0.7969	0.8450	0.6161	
15	0.3223	0.8186	0.8388	0.6303	
20	0.3247	0.8293	0.8376	0.6372	
21	0.3223	0.8305	0.8388	0.6381	Within 1% error
22	0.3199	0.8315	0.8401	0.6389	
23	0.3272	0.8326	0.8364	0.6393	
24	0.3199	0.8335	0.8401	0.6399	
25	0.3247	0.8347	0.8376	0.6402	
26	0.3247	0.8343	0.8376	0.6406	
27	0.3296	0.8347	0.8352	0.6406	Within 0.1% error

Mathematically, for this simple system, the exact solution can be obtained analytically with very little effort. The general solution of

$$\ddot{y} - 2U\dot{y} - UR_n y = 0$$

may be expressed as

$$y = Ae^{\lambda_1 z} + Be^{\lambda_2 z}$$

where two eigenvalues are

$$\lambda_1 = U + \sqrt{U^2 + UR_n}$$

$$\lambda_2 = U - \sqrt{U^2 + UR_n}$$

Applying the boundary conditions at $z = 0$ and $z = 1$ to solve for A and B, we find

$$B = 2U/[\lambda_2 e^{\lambda_2}(\lambda_1 - 2U)/\lambda_1 e^{\lambda_1} - (\lambda_2 - 2U)]$$

and

$$A = -B\lambda_2 e^{\lambda_2}/\lambda_1 e^{\lambda_1}$$

Using this expression the computed results are tabulated in Table C-2.

Table C-2 Comparison of Steady-State Results

Axial position (z = 0 to 1)	Exact Solution	Digital Solution * (Finite-difference)	Hybrid Solution **
1	0.8383	0.8388	0.8347
2	0.8255	0.8257	0.8240
3	0.8130	0.8130	0.8117
4	0.8009	0.8006	0.7964
5	0.7890	0.7885	0.7848
6	0.7775	0.7767	0.7734
7	0.7664	0.7653	0.7622
8	0.7555	0.7542	0.7516
9	0.7451	0.7436	0.7407
10	0.7350	0.7333	0.7307
11	0.7253	0.7233	0.7216
12	0.7160	0.7139	0.7122
13	0.7071	0.7048	0.7037
14	0.6986	0.6962	0.6949
15	0.6906	0.6881	0.6872
16	0.6831	0.6804	0.6798
17	0.6761	0.6734	0.6726
18	0.6696	0.6668	0.6665
19	0.6637	0.6609	0.6604
20	0.6585	0.6556	0.6554
21	0.6539	0.6510	0.6507
22	0.6500	0.6472	0.6469
23	0.6468	0.6441	0.6442
24	0.6445	0.6419	0.6416
25	0.6430	0.6407	0.6406

* iteration terminated at 0.001% error limit

** iteration terminated at 0.1% error limit

DIGITAL PROGRAM LISTING C

Classical Hybrid Steady-State Simulation
of an Isothermal Tubular Reactor
With Axial Diffusion

```
C ------ CLASSICAL HYBRID STEADY-STATE SIMULATION.
      DIMENSION Y(200)
      REAL*8 CCWMD(2),RCBMD(4),CCWHD(2),RCBHD(4),CCWOP(2),RCBOP(4)
      REAL*8 CCWDA(2),RCBDA(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4)
      INTEGER POTS(2)/'P002','P003'/
      INTEGER*2 PVAL(2)/2000,500/,IC/16/,CTLSTP/1/,CTLGO/0/
      INTEGER*2 IY1,IY0,IYDOT,LOCHD(3),LOCOP(201),LOCDA(2)
      LOCDA(1)=0
      LOCHD(1)=1
      LOCOP(1)=514
      U=1.0
      Y1C=0.5
      PAUSE ' SET UP ANALOG-REMOTE, TIMER-3 MODE '
      CALL MODE(CCWMD,IC)
      CALL FRCBSU(RCBMD,28,CCWMD)
      CALL READAD(CCWHD,2,2,LOCHD)
      CALL FRCBSU(RCBHD,29,CCWHD)
      CALL READAD(CCWOP,100,10,LOCOP)
      CALL FRCBSU(RCBOP,29,CCWOP)
      CALL WRITDA(CCWDA,8,1,LOCDA)
      CALL FRCBSU(RCBDA,30,CCWDA)
      CALL CONTRL(CCWGO,CTLGO,2)
      CALL FRCBSU(RCBGO,28,CCWGO)
      CALL CONTRL(CCWCT,CTLSTP,2)
      CALL FRCBSU(RCBCT,28,CCWCT)
      CALL FRCBSU(RCBPT,28,CCWPT)
      CALL POTSS(CCWPT,2,POTS,PVAL,RCBPT)
      CALL GOGO(RCBMD)
      PAUSE 'EOB TO START.'
      WRITE(6,5)
5     FORMAT(T15,'YOM',T35,'YOC',T55,'Y1C',T75,'YDOT',T95,'IY1')
      GO TO 100
10    CONTINUE
      CALL GOGO(RCBGO)
```

```
      CALL GOGO(RCBOP)
      CALL GOGO(RCBHD)
      CALL GOGO(RCBCT)
      YOM= (LOCHD(3)/8191.)
      YDOT= (LOCHD(2)/8191.)
      YOC=10.*YDOT/U+1.0
      Y1C=Y1C+(YOC-YOM)/10.
      WRITE(6,20) YOM,YOC,Y1C,YDOT,IY1
20    FORMAT(1H0,5G20.8)
      IF(ABS(YOC-YOM)/YOM .GT. 0.001) GO TO 100
      DO 30 I=1,100
30    Y(I)=LOCOP(102-I)/8191.
      WRITE(6,32) (Y(I),I=1,100)
32    FORMAT('0Y(1)...Y(100)='/(10F10.4))
      STOP
100   IY1=Y1C*8191
      LOCDA(2)=-IY1
      CALL GOGO(RCBDA)
      GO TO 10
      END
```

```
      SUBROUTINE GOGO(RCB)
      REAL*8 RCB(1)
      CALL FRTIO(RCB,IRET)
      CALL FCHECK(RCB,IRET,1)
      RETURN
      END
```

APPENDIX D

Classical CSDT Hybrid Simulation of An Isothermal Tubular Reactor with Axial Diffusion

$$\boxed{\frac{\partial^2 y}{\partial z^2} - 2U\frac{\partial y}{\partial z} - UR_n y = \frac{1}{\alpha}\frac{\partial y}{\partial t}}$$

For the analog mechanization the estimated maximums are:

$$\max |\ddot{y}| = 40$$

$$\max |\dot{y}| = 40$$

$$\max |y| = 1$$

From these, we obtain the following scaled equations

$$\left[\frac{\ddot{y}}{40}\right] = -\frac{2U}{10}(10)\left[\frac{\dot{y}}{40}\right] + \frac{UR_n}{40}[y] + \frac{1}{40\ \alpha\Delta t}\ [y - y^{i-1}]$$

with the step input at $z = 0$ as one of the boundary conditions,

$$\left[\frac{\dot{y}}{40}\right] = \frac{2U}{40}[y_0 - 1]$$

and at the other end, $z = 1$

$$\left[\frac{\dot{y}}{40}\right] \equiv 0$$

The hybrid procedure has the same basic set-up as in the steady-state case which is described in Appendix C except that in this case an additional DAC is required to simultaneously play back the previous profile, y^{i-1}. The basic logic and analog wirings are shown in Figure D-1. Instead of using the crude approach to update the initial condition (like the one used in Appendix C) the method of false position (regular falsi) is employed in the digital portion of the hybrid program which is listed in Listing D.

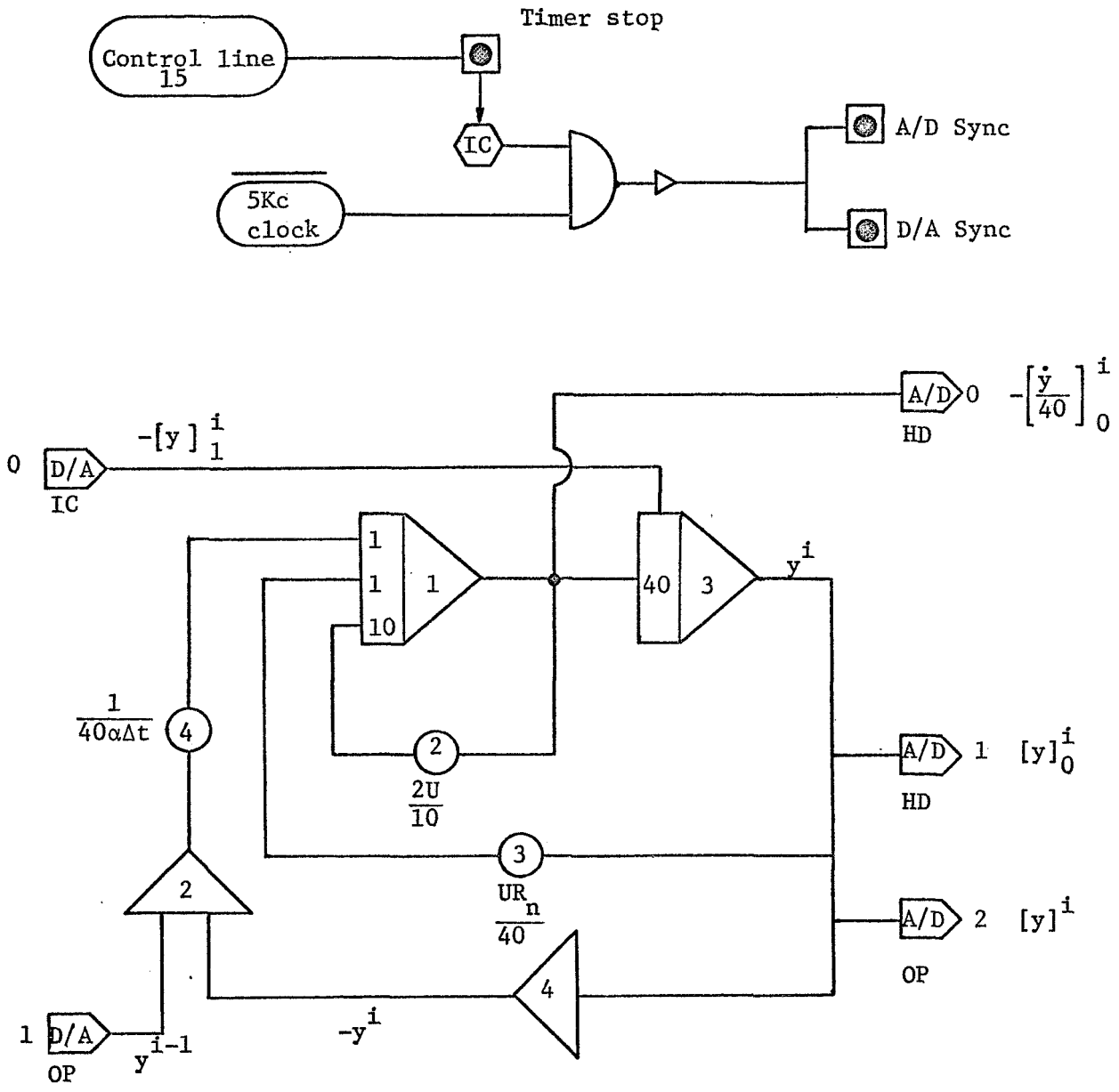


Figure D-1 Logic and Analog Wiring Diagram

DIGITAL PROGRAM LISTING D

Classical 'CSDT' Hybrid Simulation of
A Tubular Reactor With Axial Diffusion

```
C ...... CLASSICAL 'CSDT' HYBRID SIMULATION OF A TUBULAR FLOW REACTOR
C ......  WITH AXIAL DIFFUSION   ---------------------------
      INTEGER*2 DISP(2000)
      DIMENSION Y(200),P(1000)
      REAL*8 CCWMD(2),RCBMD(4),CCWHD(2),RCBHD(4),CCWOP(2),RCBOP(4)
      REAL*8 CCWDA(2),RCBDA(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWYI(2),RCBYI(4)
      INTEGER POTS(3)/'P002','P003','P004'/
      INTEGER*2 PVAL(3),          IC/16/,CTLSTP/1/,CTLGO/0/
      INTEGER*2 IY1,IY0,IYDOT,LOCHD(3),LOCOP(201),LOCDA(2),LOCYI(201)
      LOGICAL RIGHT
      PVAL(1)=2000
      PVAL(2)=250
      LOCDA(1)=0
      LOCHD(1)=1
      LOCOP(1)=514
      LOCYI(1)=257
      U=1.0
1       CONTINUE
      WRITE(15,2)
2     FORMAT(' SET ''NW'' IN ''I4''')
      READ(15,3) NW
3     FORMAT( I4)
      NW2=NW+2
      DO 4 I=1,NW
4     P(I)=FLOAT(I-1)/NW
      CALL READAD(CCWOP, NW,10,LOCOP)
      CALL FRCBSU(RCBOP,29,CCWOP)
      CALL READAD(CCWHD,2,2,LOCHD)
      CALL FRCBSU(RCBHD,29,CCWHD)
      CALL WRITDA(CCWYI,10,NW,LOCYI)
      CALL FRCBSU(RCBYI,30,CCWYI)
      CALL WRITDA(CCWDA,8,1,LOCDA)
      CALL FRCBSU(RCBDA,30,CCWDA)
      CALL MODE(CCWMD,IC)
```

```
      CALL FRCBSU(RCBMD,28,CCWMD)
      CALL CONTRL(CCWGO,CTLGO,2)
      CALL FRCBSU(RCBGO,28,CCWGO)
      CALL CONTRL(CCWCT,CTLSTP,2)
      CALL FRCBSU(RCBCT,28,CCWCT)
      CALL FRCBSU(RCBPT,28,CCWPT)
      PAUSE ' SET UP ANALOG-REMOTE, TIMER-3 MODE, EOB TO START.'
      CALL POTSS(CCWPT,2,POTS,PVAL,RCBPT)
200     CONTINUE
      WRITE(15,5)
5     FORMAT(' SET ''P004'' IN ''I4''')
      READ(15,3) PVAL(3)
      CALL POTSS(CCWPT,1,POTS(3),PVAL(3),RCBPT)
      WRITE(6,6) NW,PVAL
6     FORMAT(' NW=',I5,T15,'POT-2=',I5,T30,'POT-3=',I5,T45,'POT-4=',I5)
      IT=1
      RIGHT=.FALSE.
      IRF=0
      IY1=0
      Y1C=0.0
      DO 7 I=1,NW
      P(I+NW)=Y1C
      DISP(I)  =IY1
7     LOCYI(I+1)=IY1
      CALL GOGO(RCBMD)
      WRITE(6,9)
9     FORMAT(T15,'YOM',T35,'YOC',T55,'Y1C',T75,'YDOT',T95,'IY1')
      GO TO 100
10    CONTINUE
      CALL GOGO(RCBCT)
      CALL GOGO(RCBGO)
      CALL FRTIO(RCBYI,IRET)
      CALL GOGO(RCBOP)
      CALL GOGO(RCBHD)
      YOM= (LOCHD(3)/8191.)
      YDOT= (LOCHD(2)/8191.)*40.
```

```
      YOC=0.5*YDOT/U+1.0
      IRF=IRF+1
      FX=YOM-YOC
      IF(IRF.LE.1) GO TO 12
      IF(IRF.GT.20.OR.ABS(FX/YOM).LE.0.005) GO TO 21
      IF(FX*FXL.GT.0.) GO TO 12
      FXR=FX
      XR=Y1C
      RIGHT=.TRUE.
11    Y1C=XL+FXL*(XR-XL)/(FXL-FXR)
      GO TO 100
12    FXL=FX
      XL=Y1C
      IF(RIGHT) GO TO 11
      Y1C=Y1C-(FX/10.)*IRF
      GO TO 100
21    IT=IT+1
      WRITE(6,102) YOM,YOC,Y1C,YDOT,IY1,IRF
      IRF=0
      RIGHT=.FALSE.
      J=IT*NW
      DO 30 I=1,NW
      DISP(J+I)=LOCOP(NW2-I)
      LOCYI(I+1)=LOCOP(I+1)
      Y(I)=LOCOP(NW2-I)/8191.
      P(J+I)=Y(I)
30     CONTINUE
      WRITE(6,32) (Y(I),I=1,NW)
32    FORMAT('0Y(1)...Y( NW) ='/(10F10.4))
      IF(IT.LT.10) GO TO 10
      IDT=PVAL(3)/10
      CALL PLOT(IDT,P,NW,10,0,0)
      WRITE(6,36)
36    FORMAT('0',T55,'X  VS.  Y'/'1')
      GO TO 200
100   IY0=IY1
```

```
      IY1=Y1C*8191.
102   FORMAT(1H0,6G20.8)
      LOCDA(2)=-IY1
      CALL GOGO(RCBDA)
      GO TO 10
      END
```

APPENDIX E

'CSDT' Hybrid Simulation With The Decomposition Method

———— * ————

Simulation of a Tubular Flow Reactor
With Axial Diffusion

TABLE E-1 Steady-State Solutions of the Decomposition Method for Different Δt

$\alpha\Delta t$	y(0)	y(1)	no. of time step ≠
∞ *	0.8383	0.6425	-
DSDT **	0.8393	0.6435	109
0.5	0.8381	0.6370	5 ~ 7
0.25	0.8404	0.6360	9 ~ 11
0.125	0.8430	0.6399	14
0.05	0.8600	0.6500	30 ~ 40
0.025	0.8880	0.6584	50^{+}

* analytic solution

≠ reached apparent steady-state

** implicit finite difference solution using $\alpha\Delta t = 0.025$ and 0.001% tolerance.

Mechanization of the two first order equations

1) $$\frac{dx}{d(-z)} - \lambda_B x = -\frac{y^i}{\alpha\Delta t}$$

with the following estimated maximums

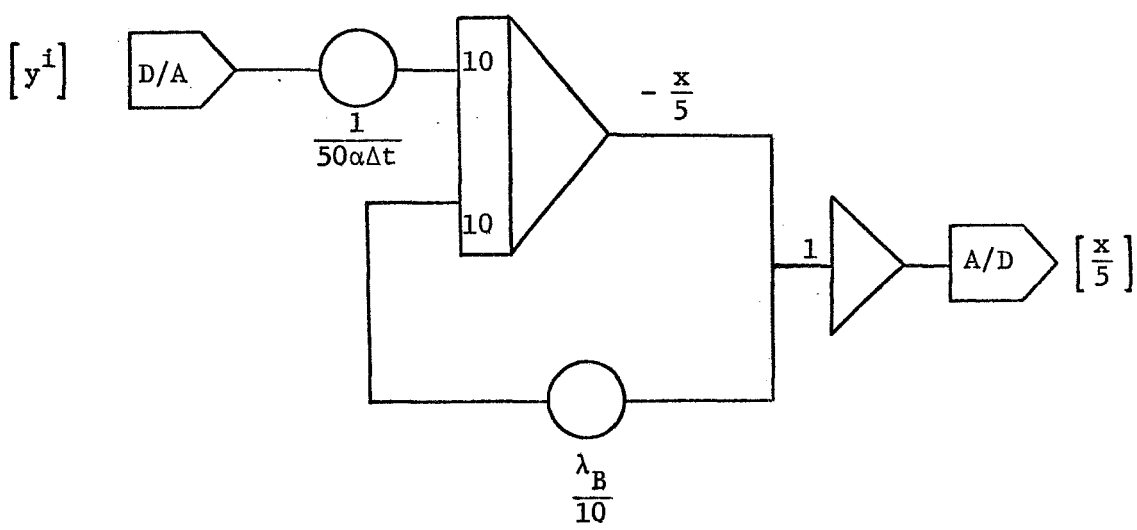
$$\max \left|\frac{1}{\alpha\Delta t}\right| = 50$$

$$\max |x| = 5$$

$$\max |\lambda_B| = 10$$

We have the scaled equation:

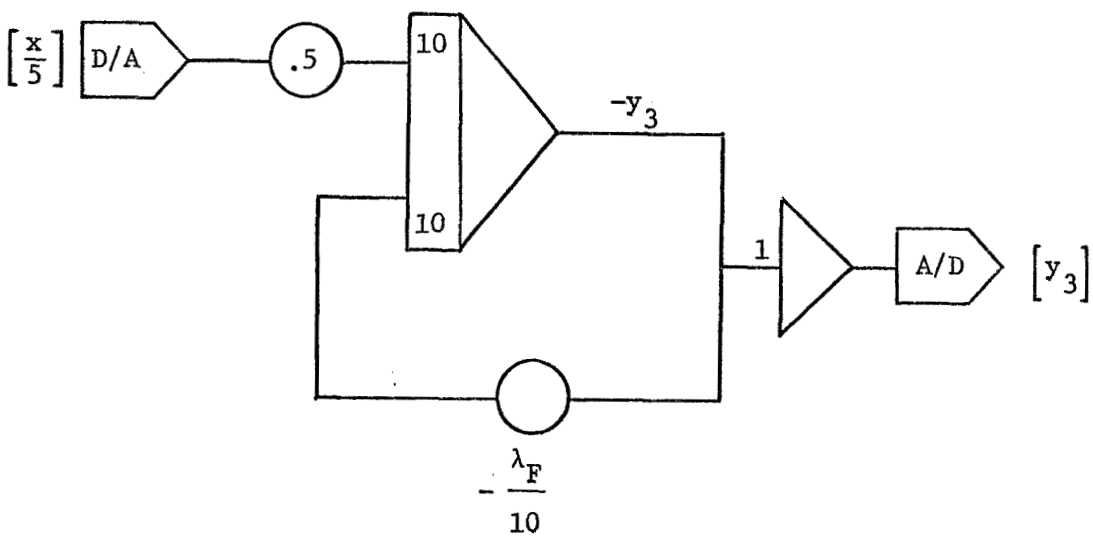
$$\left[\frac{\dot{x}}{5}\right] = \left[\frac{\lambda_B}{10}\right](10)\left[\frac{x}{5}\right] + \left[\frac{y^i}{50\ \alpha\Delta t}\right](10)$$



2) $\frac{dy_3}{dz} - \lambda_F y_3 = x$

where $\max |\lambda_F| = 10$, then

$$\left[\dot{y}_3\right] = -\left[-\frac{\lambda_F}{10}\right](10)(y_3) + \left[\frac{x}{5}\right](5)$$



DIGITAL PROGRAM LISTING E

Serial Decomposition Solution of an
Isothermal Tubular Reactor in One-space
Dimension

```
C ..... 'CSDT' IN ONE-SPACE DIMENSION WITH DECOMPOSITION METHOD.
      COMMON/BEGMNT/ DUM(7),P(500)
      NW=50
      NW1=NW+1
      NW2=NW+2
      REAL*8 CCWAN(2),RCBAN(8)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      INTEGER POTS(5)/'P201','P202','P203','P204','P205'/
      INTEGER*2 PVAL(5),LOCS(51),LOCX(51),IC/16/,CTLSTP/1/,CTLGO/0/
      REAL LAM1,LAM2,Y1(50),Y2(50),Y3(50),Y(50),S(50),X(50)
1     WRITE(15,2)
2     FORMAT(' ENTER ADT = ( 0.025 - 0.25 )')
      READ(15,3) ADT
3     FORMAT(F10.5)
      WRITE(6,4) ADT
4     FORMAT('1''CSDT'' WITH DECOMPOSITION, ALPHA*DT=',F7.3)
      SQ=SQRT(2.+1./ADT)
      LAM1=1.+SQ
      LAM2=1.-SQ
      Z=0.
      DZ=1./(NW-1)
      YMAX=EXP(LAM1)
      DO 5 I=1,NW
      Y1(I)=EXP(LAM1*Z)/YMAX
      Y2(I)=EXP(LAM2*Z)
5     Z=Z+DZ
      BB=Y1(1)*Y2(NW)*LAM2*(LAM1-2.)-LAM1*(LAM2-2.)
      DO 7 I=1,NW
      P(I+NW)=0.
      Y(I)=0.
7     S(NW1-I)=Y(I)
      WRITE(6,75) LAM1,LAM2,Y1
75    FORMAT('0LAM1=',G10.4,5X,'LAM2=',G10.4/' Y1..'/(10G12.4))
      WRITE(6,76) Y2
```

```
76    FORMAT('0Y2..'/(10G12.4))
      CALL MODE(CCWMD,IC)
      CALL FRCBSU(RCBMD,28,CCWMD)
      CALL CONTRL(CCWGO,CTLGO,2)
      CALL FRCBSU(RCBGO,28,CCWGO)
      CALL CONTRL(CCWCT,CTLSTP,2)
      CALL FRCBSU(RCBCT,28,CCWCT)
      PVAL(1)=MINO(IFIX(100./ADT),9999)
      PVAL(2)=LAM1*1000
      PVAL(3)=5000
      PVAL(4)=8000
      PVAL(5)=-LAM2*1000
      CALL FRCBSU(RCBPT,28,CCWPT)
      CALL POTSS(CCWPT,5,POTS,PVAL,RCBPT)
      CALL ANALOG(CCWAN,5,POTS,LOCS,LOCX)
      CALL FRCBSU(RCBAN,28,CCWAN(1),4,29,CCWAN(2))
      CALL FRTIO(RCBAN,IRET)
      CALL FCHECK(RCBAN,IRET,1)
      DO 81 I=1,5
81    LOCX(I)=LOCX(I)/0.8191
      WRITE(6,82) POTS,PVAL,(LOCX(I),I=1,5)
82    FORMAT('0''POTSS'' RESULTS..'/T22,5(3X,A4)/'  VALUES ASSIGNED -->'
     -,T22,5I7/'  SET RESULTS -->',T22,5I7)
      CALL FRTIO(RCBMD,IRET)
      CALL FCHECK(RCBMD,IRET,1)
      IT=0
9     CONTINUE
      IT=IT+1
      DO 10 I=1,NW
10    LOCS(I+1)=S(I) *8191.
      CALL DACADC(CCWDA,RCBDA,LOCS,0,CCWAD,RCBAD,LOCX,0,NW)
      CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
      IF(IT.GT.1) GO TO 13
      DO 12 I=1,NW
12    LOCX(I+1)=0
      X(NW)=0.
```

```
13      DO 14 I=2,NW
        LOCS(I   )=LOCX(NW2-I)
14      X(I-1)=LOCX(NW2-I)/8191.*5.
        LOCS(NW1)=0
        CALL DACADC(CCWDA,RCBDA,LOCS,257,CCWAD,RCBAD,LOCX,257,NW)
        CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
        DO 16 I=1,NW
16      Y3(I)=LOCX(I+1)/8191.
        IF(IT.GT.1) GO TO 17
        DO 160 I=1,NW
160     Y3(I)=0.
17      B=(LAM1*(X(1)+2.+LAM2*Y3(1))-Y1(1)*Y3(NW)*LAM2*(LAM1-2.))/BB
        A=-LAM2*(B*Y2(NW)+Y3(NW))/LAM1
        DO 18 I=1,NW
        Y(I)=A*Y1(I)+B*Y2(I)+Y3(I)
18      S(NW1-I)=Y(I)
        WRITE(6,20) IT,A,B,Y
20      FORMAT('0IT=',I3,5X,'A=',G12.4,5X,'B=',G12.4/' Y.. '/(10G12.4))
21      FORMAT(' Y3..'/(10G12.4))
15      FORMAT('0X..'/(10G12.4))
        IF(IT/10*10.EQ.IT) GO TO 22
        GO TO 9
22      PAUSE ' WANT TO CONTINUE ? ? HIT ''EOB''. IF NO, HIT ''STOP''.'
        GO TO 9
        END
```

```
SUBROUTINE DACADC(CCWDA,RCBDA,LDA,DA,CCWAD,RCBAD,LAD,AD,NW)
REAL*8 CCWDA(1),RCBDA(1),CCWAD(1),RCBAD(1)
INTEGER*2 LDA(1),LAD(1)
INTEGER  DA,AD
LDA(1)=DA
LAD(1)=AD
CALL WRITDA(CCWDA,10,NW,LDA)
CALL FRCBSU(RCBDA,30,CCWDA)
CALL READAD(CCWAD,NW,10,LAD)
CALL FRCBSU(RCBAD,29,CCWAD)
RETURN
END
```

```
SUBROUTINE REALGO(CT,GO,DA,AD)
REAL*8 CT(1),GO(1),DA(1),AD(1)
CALL FRTIO(CT,IRT)
CALL FCHECK(CT,IRT,1)
CALL FRTIO(GO,IRT)
CALL FCHECK(GO,IRT,1)
CALL FRTIO(DA,IRT)
CALL FRTIO(AD,IRT)
CALL FCHECK(AD,IRT,1)
CALL FRTIO(CT,IRT)
RETURN
END
```

APPENDIX F

Solution of an Isothermal Tubular Reactor in Two-Space Dimension and Time --- CSDSDT Decomposition Method

For a simple first order irreversible reaction

$$A \rightarrow B + C$$

taking place in a homogeneous tubular flow reactor, a material balance of the reactant A may be represented by the following equation

$$\frac{1}{Pe_z}\frac{\partial^2 f}{\partial z^2} + \frac{1}{Pe_r}\left[\frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r}\right] - \frac{\partial f}{\partial r} - \beta f = \delta\frac{\partial f}{\partial t} \qquad (4.13)$$

where $f = f(z,r,t)$, the dimensionless concentration.

The initial and boundary conditions are:

$$f(z,r,0) \equiv 0 \quad ; \quad z \in (0,1),\ r \in (0,1)$$

$$f - \frac{1}{Pe_z}\frac{df}{dz} = 1 \quad ; \quad z = 0,\ r \neq 1,\ t > 0$$

$$f = 1 \quad ; \quad z = 0,\ r = 1,\ t > 0$$

and

$$\frac{df}{dz} = 0 \quad ; \quad z = 1,\ t > 0$$

$$\frac{df}{dr} = 0 \quad ; \quad r = 0 \text{ and } 1$$

Since the coefficients are considered to be constant in the axial direction, we have the general solution in the form

$$f = af_1 + bf_2 + f_3 \qquad (F\text{-}1)$$

for each radial spaceline, where f_1 and f_2 represent their elementary (or homogeneous) solutions, and f_3 represents the particular solution.

To calculate constants a and b, since from equation F-1, we obtain

$$\frac{df}{dz} = a\frac{df_1}{dz} + b\frac{df_2}{dz} + \frac{df_3}{dz} \tag{F-2}$$

We have

$$\begin{aligned} \frac{df_1}{dz} &= \lambda_B f_1 \\ \frac{df_2}{dz} &= \lambda_F f_2 \\ \frac{df_3}{dz} &= \lambda_F f_3 + x \end{aligned} \tag{F-3}$$

with initial conditions:

$$\begin{aligned} x(1) &= 0 \\ f_1(1) &= 1 \\ f_2(0) &= 1 \\ f_3(0) &= 0 \end{aligned} \tag{F-4}$$

Apply the respective axial boundary conditions.

(1) For $r \neq 1$, substituting (F-3) into (F-2) leads to the final expressions for a and b as

$$a = \frac{-\lambda_F[Rf_2(1) + Pf_3(1)]}{\lambda_B P + \lambda_F Q f_1(0)\ f_2(1)}$$

$$b = \frac{a\,Qf_1(0) + R}{P}$$

where $P \equiv Pe_z - \lambda_F$

$Q \equiv \lambda_B - Pe_z$

$R \equiv x(0) + Pe_z$

(2) For $r = 1$ (at the wall) we have

$$af_1(0) + bf_2(0) + f_3(0) = 1$$

or apply (F-4), then

$$af_1(0) + b = 1$$

and

$$a\lambda_B + b\lambda_F f_2(1) + \lambda_F f_3(1) = 0$$

Solving these two equations obtain

$$a = \frac{\lambda_F[f_2(1) + f_3(1)]}{\lambda_F f_1(0)\ f_2(1) - \lambda_B}$$

or, since $\lambda_F = -\lambda_B$

$$a = \frac{f_2(1) + f_3(1)}{1 + f_1(0)\ f_2(1)}$$

and

$$b = 1 - a\, f_1(0)$$

Since both $f_1(0)$ and $f_2(1)$ are in the order of 10^{-40} or less, while f_3 is bounded between 0 and 1, then

$$a = \frac{1 + f_3(1)/f_2(1)}{f_1(0) + 1/f_2(1)} \quad (\dot{\approx} 0)$$

$$b = a\left[\frac{1}{a} - f_1(0)\right]$$

Assuming there are ten equally-spaced radial increments across the reactor tube radius, the following data are given for the example case

$L = 10$ ft.

$R = 1$ ft.

$\Delta r = 0.1$

$\Delta t = 0.01$

$N = 11 \quad (n = 1,2, \ldots, N)$

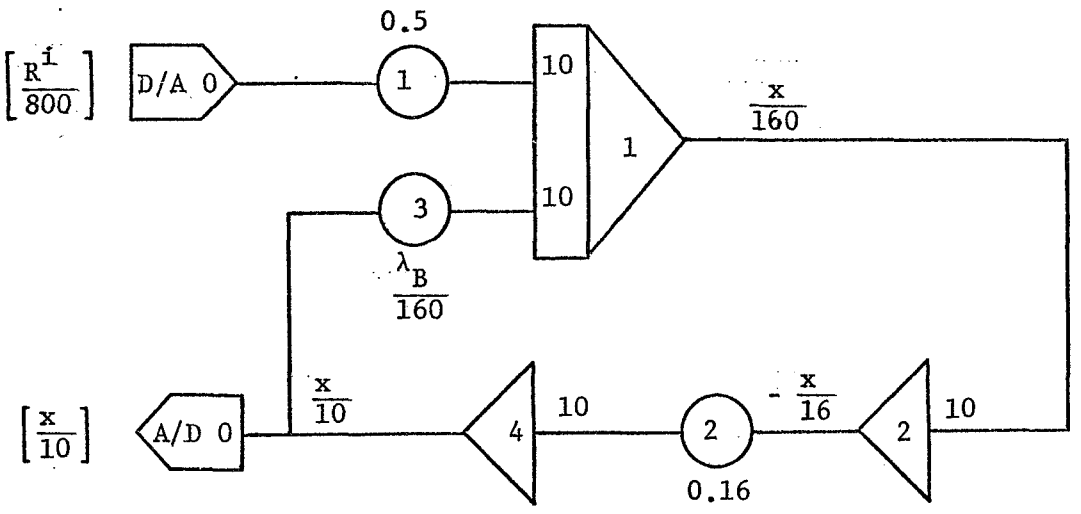
$K_c = 1$

The velocities and Peclet numbers are:

n (from center line)	Pe_z	u (ft/sec)	D_x (ft/sec^2)
1	110	25.8	
2	100	25.7	
3	80	25.5	
4	65	25.1	
5	55	24.8	
6	55	24.4	
7	60	23.8	
8	70	23.1	
9	90	21.9	
10	150	19.8	
11	0	0	0.2

The analog mechanization of the decomposed two first order equations are shown in Figures F-1 and F-2 with their corresponding scaled equations.

$$-\left[\frac{\dot{x}}{160}\right] = \left[\frac{\lambda_B}{160}\right]\left[\frac{x}{10}\right](10) + \left[\frac{R^i}{800}\right]\left(\frac{10}{2}\right)$$



$$\dot{f}_3 = -\left[\frac{-\lambda_F}{10}\right](10)\left[f_3\right] + \left[\frac{x}{10}\right](10)$$

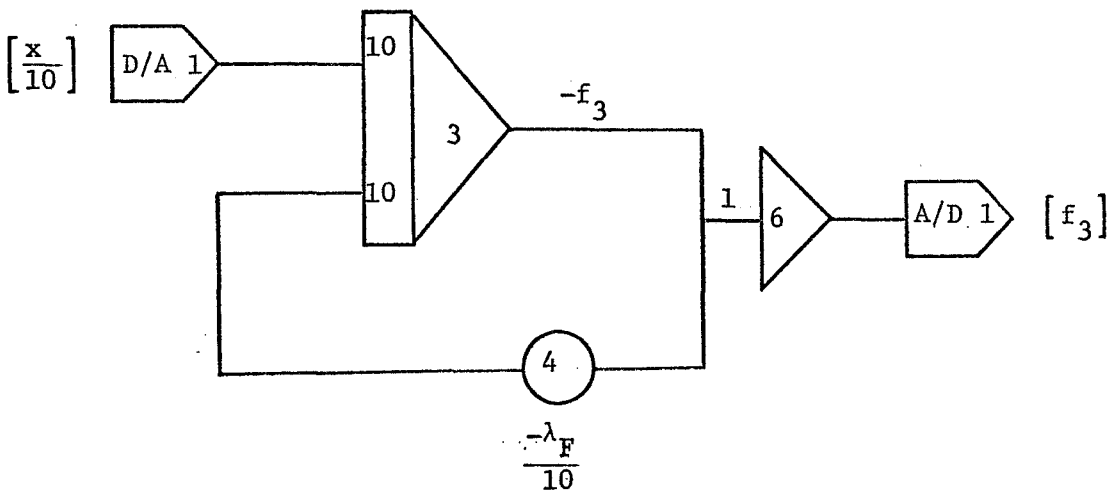
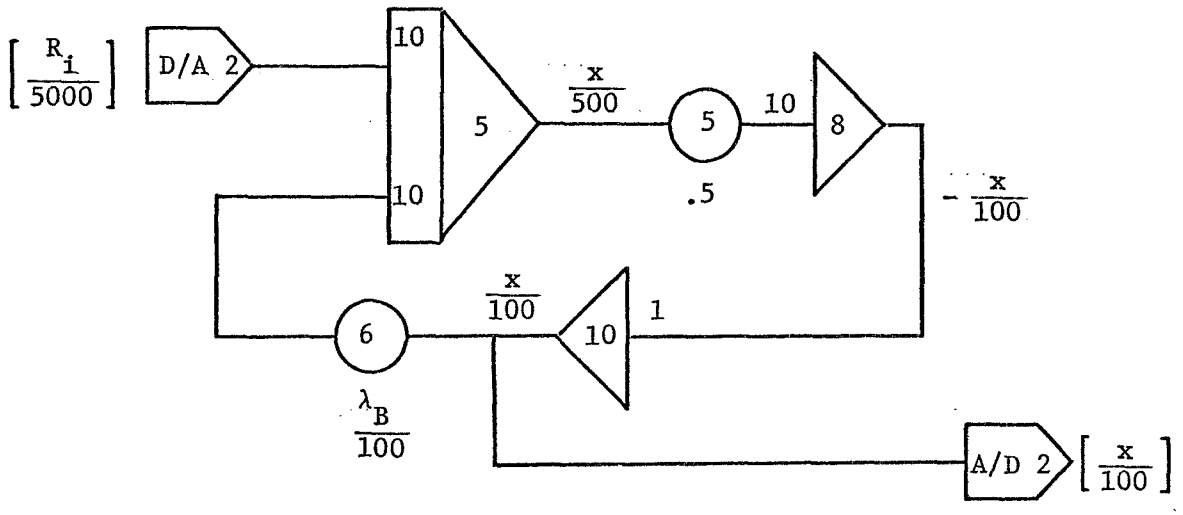


Figure F-1 Analog Mechanization of $L_B(\ )$ and $L_F(\ )$ for $r \neq 1$

$$-\frac{1}{5}\left[\frac{\dot{x}}{100}\right] = \left[\frac{\lambda_B}{100}\right]\left[\frac{x}{100}\right](20) + \left[\frac{R_i}{5000}\right](10)$$



$$\left[\frac{\dot{f}_3}{10}\right] = -\left[\frac{-\lambda_F}{100}\right](f_3)(10) + \left[\frac{x}{100}\right](10)$$

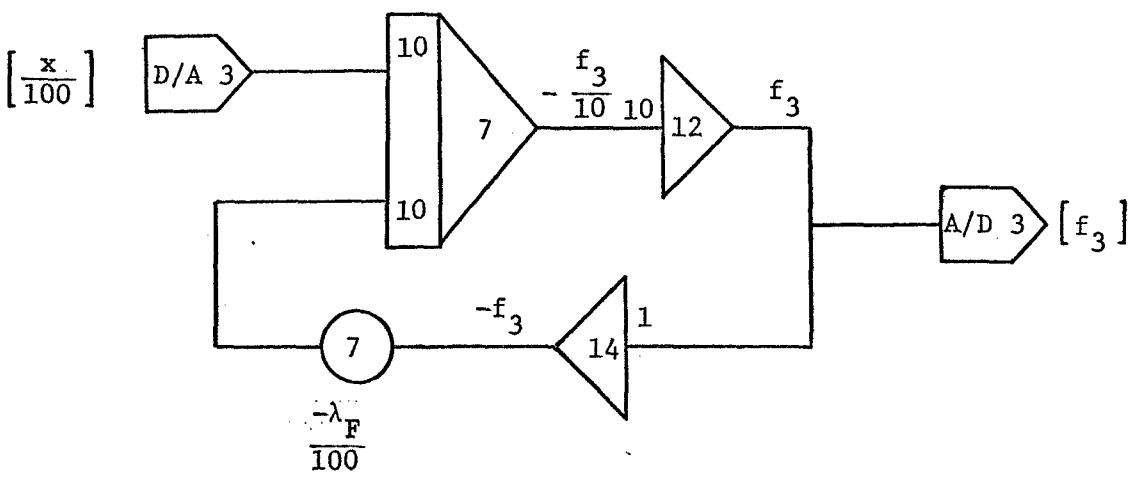


Figure F-2 Analog Mechanization of $L_B(\ )$ and $L_F(\ )$ for $r = 1$

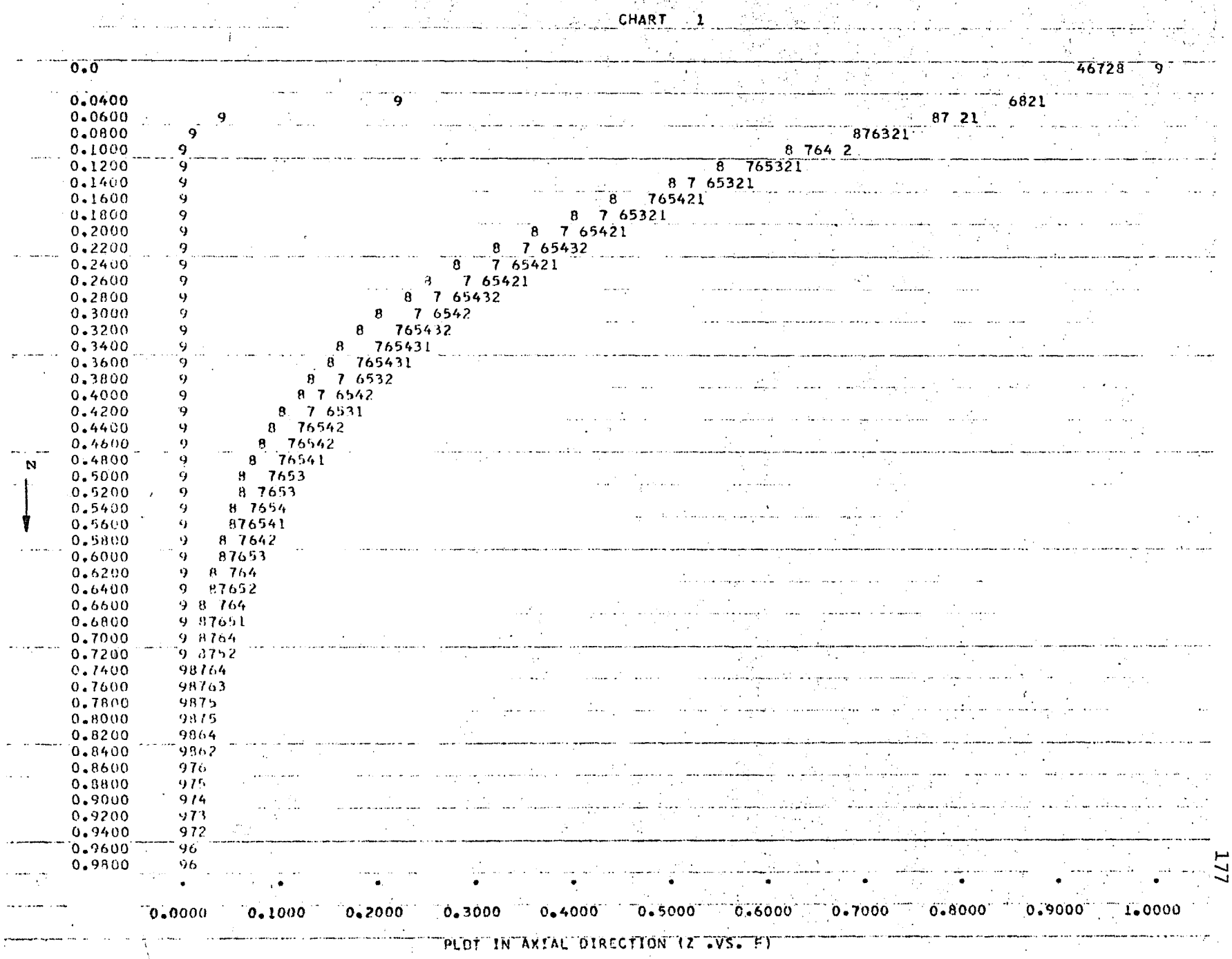


Figure F-3 Transient Results at one Δt

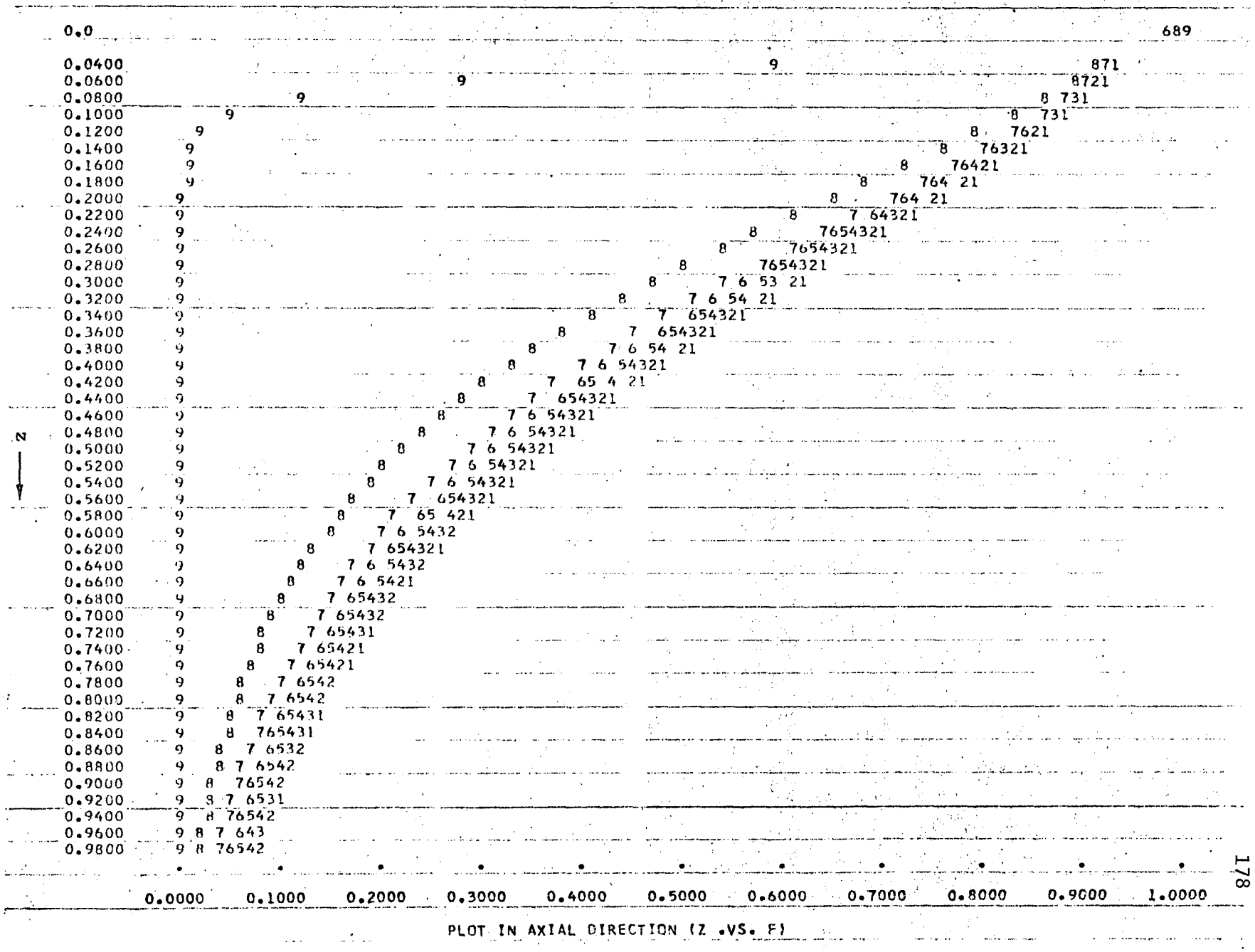


Figure F-4 Transient Results at 2Δt

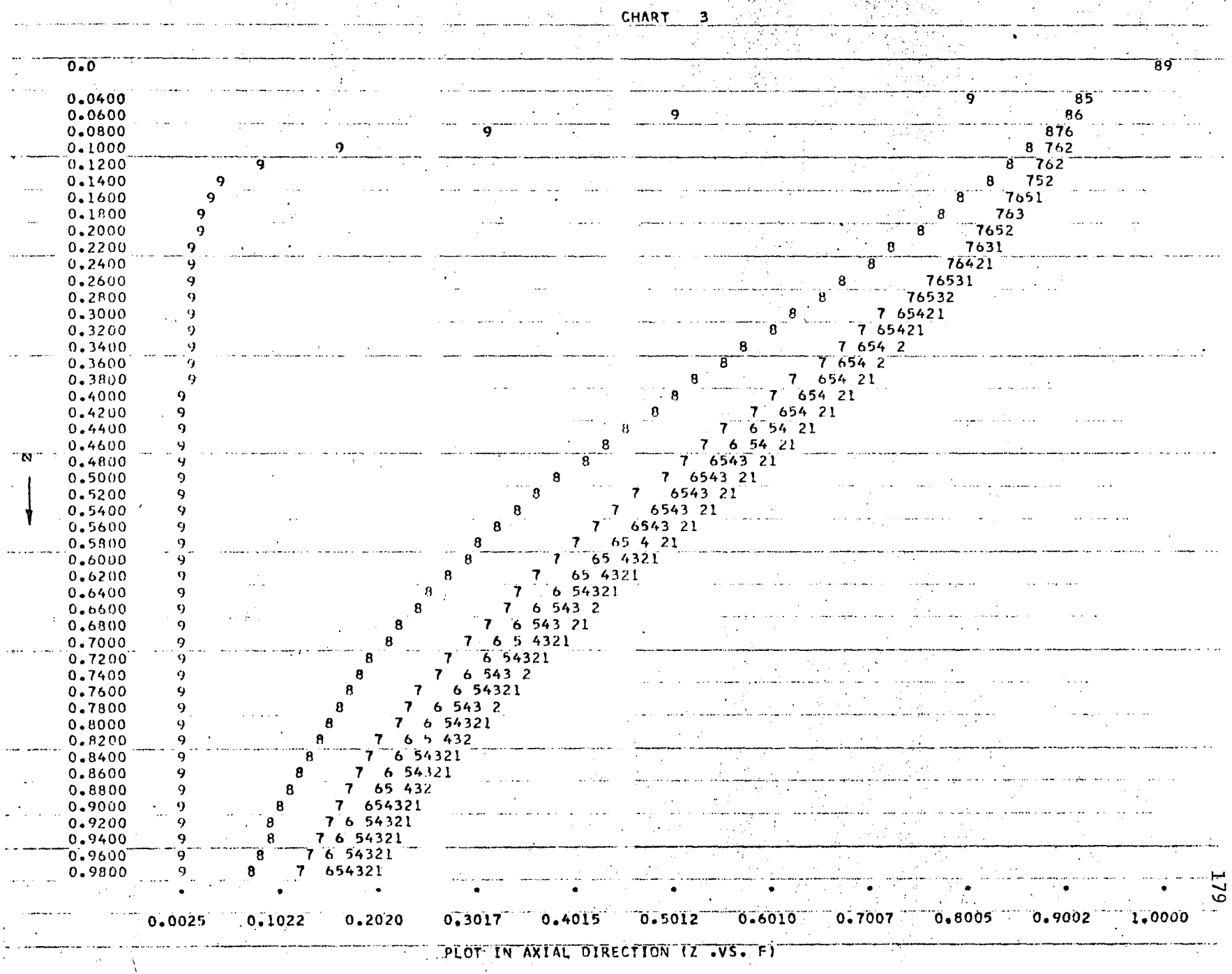


Figure F-5 Transient Results at 3Δt

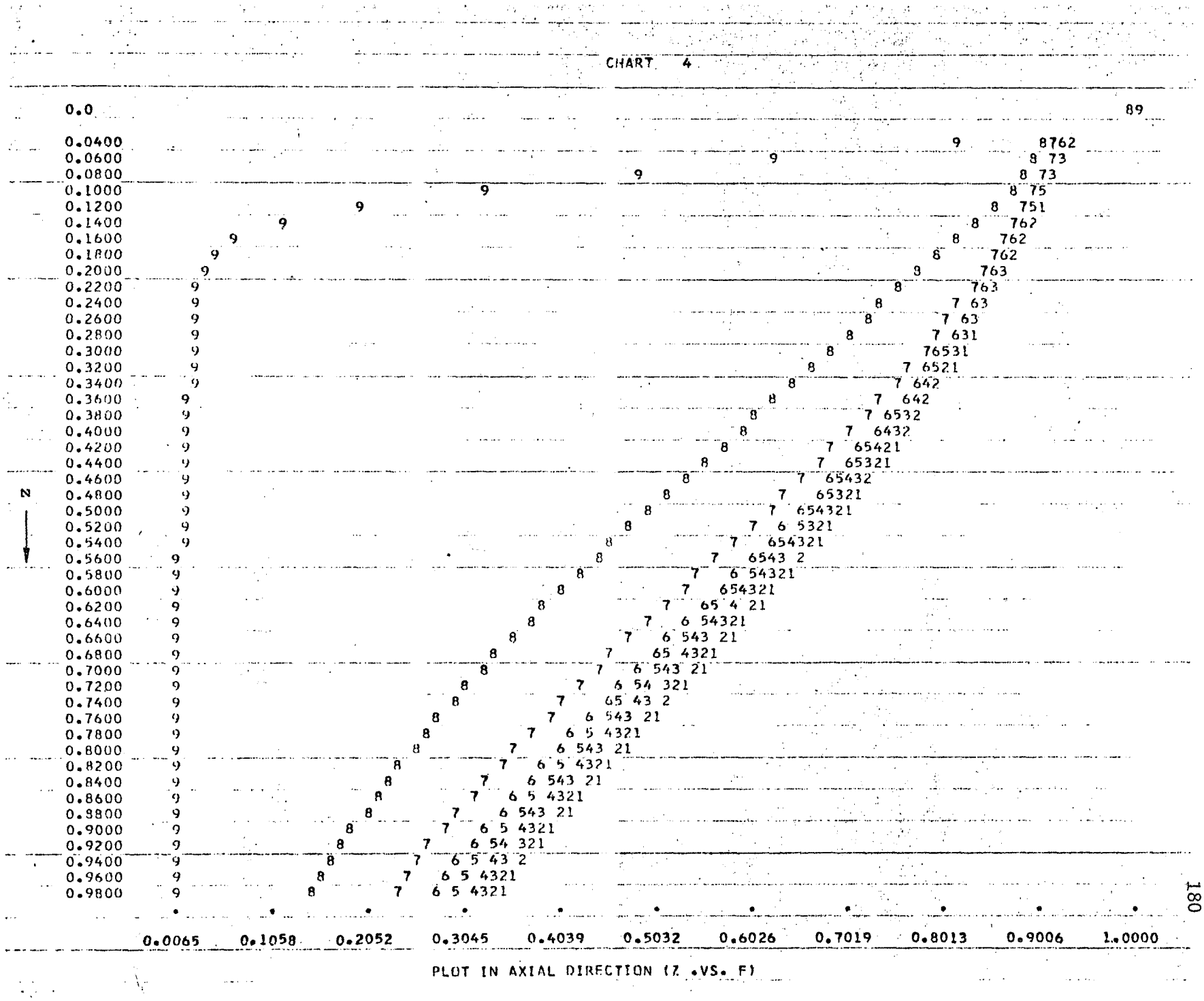


Figure F-6 Transient Results at 4Δt

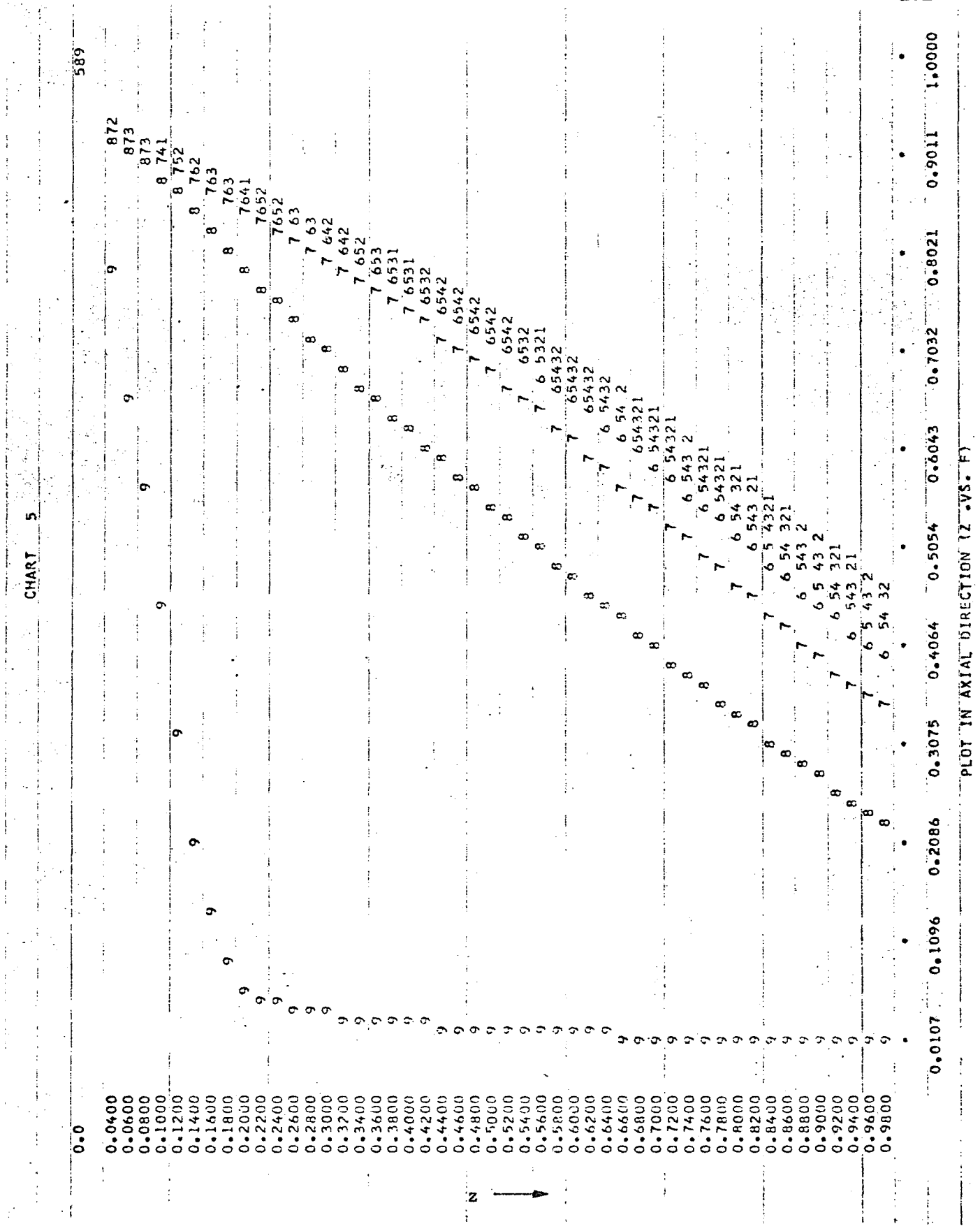


Figure F-7 Transient Results at $5\Delta t$

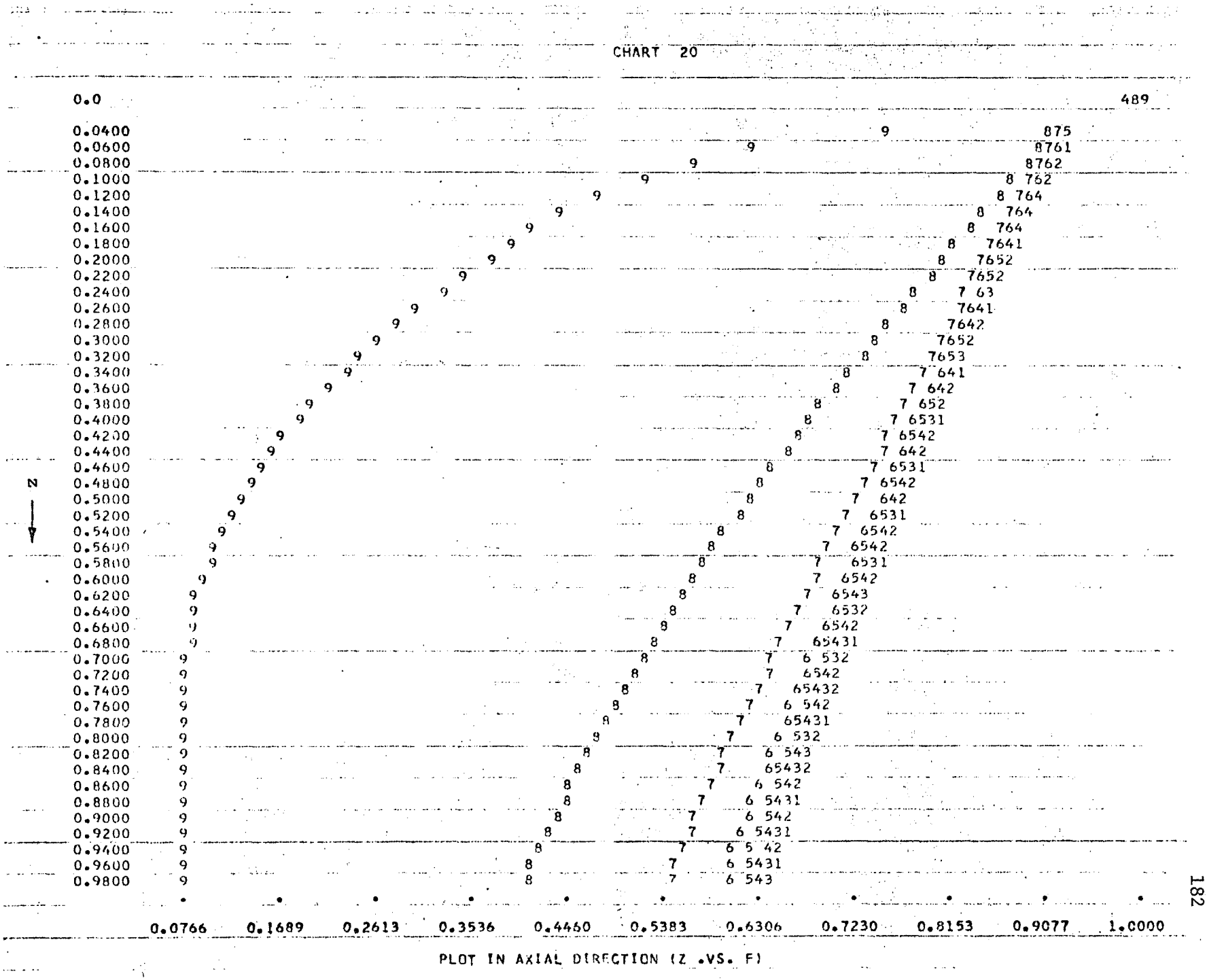


Figure F-8 Transient Results at 20Δt

DIGITAL PROGRAM LISTING F

Simulation of an Isothermal Tubular Reactor
in Two-space Dimension With The Hybrid 'CSDSDT'
Decomposition Method

```
C ..... ''ISOTHERMAL TBULAR FLOW REACTOR WITH AXIAL/RADIAL DIFFUSION''.
C ..... 'CSDSDT' IN TWO-SPACE DIMENSION WITH DECOMPOSITION METHOD.
      REAL PEZ(10),U(10),LAMF,LAMB
C     DATA PEZ/2*200.,160.,130.,2*110.,120.,140.,180.,300./
C     DATA PEZ/2*100., 80., 65.,2* 55., 60., 70., 90.,150./
      DATA PEZ/110.,100.,80.,65.,2*55.,60.,70.,90.,150./
      DATA U/25.8,25.7,25.5,25.1,24.8,24.4,23.8,23.1,21.9,19.8/
      COMMON NW,NW1,NW2,DZ,IT,Y
      REAL F(50,11),R(50),Y(50)
C     REAL L/10./,KC/1./,T/1./,DXW/0.01/,RI/0.1/
C     REAL L/10./,KC/1./,T/10./,DXW/0.02/,RI/0.1/
      REAL L/10./,KC/1./,T/10./,DXW/0.2/,RI/1.0/
      DO 1 I=1,50
      DO 1 J=1,11
1     F(I,J)=0.0
      WRITE(15,2)
      READ(15,3) ITMAX,DT
      WRITE(6,4) DT,ITMAX
2     FORMAT(' ENTER ITMAX & DT IN ''I2,F5.3''')
3     FORMAT(I2,F5.3)
4     FORMAT('1''CSDSDT'', DR=0.1, DT=',F6.3,'  ITMAX=',I3)
C     DT=0.01
      DR=0.1
      DRR=DR*DR
      NW=50
      NW1=NW+1
      NW2=NW+2
      DZ=1./(NW-1)
      DO 100 IT=1,ITMAX
      DO 50 J=1,11
      IF(J.EQ.11) GO TO 30
      PZ=PEZ(J)
C     PER=PZ/100.
      PER=PZ
      BETA=L*KC/U(J)
```

```
      DDT=L/(U(J)*T*DT)
      SQ=SQRT(PZ*(PZ+4.*(BETA+DDT)))
      LAMF=(PZ-SQ)/2.
      LAMB=(PZ+SQ)/2.
      IF(J.GT.1) GO TO 20
C ------ CENTER  LINE
      CP=4./DRR
      DO 18 I=1,50
18    R(I)=-PZ/PER*(CP*F(I,2)+(DDT*PER-CP)*F(I,1))
      GO TO 40
C ------ GENERAL LINES
20    JJ=J-1
      CP=(2.*JJ+1)/(2*JJ*DRR)
      CQ=-2/DRR
      CT=(2*JJ-1)/(2*JJ*DRR)
22    DO 24 I=1,50
24    R(I)=-PZ/PER*(CP*F(I,J+1)+(CQ+DDT*PER)*F(I,J)+CT*F(I,J-1))
      GO TO 40
C ------ WALL BOUNDARY
30    WW=L*L*KC/DXW
      XW=L*L/(DXW*T)
      VW=L*L/(RI*RI*100.)
      LAMF=-SQRT(WW+XW/DT)
      LAMB=-LAMF
      CQ=VW/DRR
      CP=CQ-XW/DT
      PZ=0.
      DO 32 I=1,50
32    R(I)=CP*F(I,11)-CQ*F(I,10)
40    CONTINUE
      IF(IT.NE.ITMAX) GO TO 45
      IF(J.EQ.1) WRITE(6,99)
      WRITE(6,44) J,LAMF,LAMB,R
44    FORMAT('0'///' J=',I3,5X,'LAMF=',F8.2,5X,'LAMB=',F8.2/
     $  ' R(50)=...'/( ' ',10F10.2))
45    CONTINUE
```

```
      IF(J.EQ.1) GO TO 47
      DO 46 I=1,50
46    F(I,J-1)=Y(I)
47    CALL HYBGO(LAMB,LAMF,R,PZ,J)
50    CONTINUE
      DO 52 I=1,50
52    F(I,11)=Y(I)
      IF(IT.NE.ITMAX) GO TO 60
      DO 55 J=1,11
      WRITE(6,54) J,(F(I,J),I=1,50)
54    FORMAT('0F(50,',I2,')=...'/(10G12.4))
55    CONTINUE
C ......... PLOT ROUTINE ........
      REAL P(500)
C ------ PLOT IN AXIAL DIRECTION
60    NLINE=1
      K=2
      DO 64 J=1,11,K
      IF(J.GE.5) K=1
      INW=NLINE*NW
      DO 62 I=1,NW
62    P(INW+I)=F(I,J)
64    NLINE=NLINE+1
      DO 66 I=1,NW
66    P(I)=FLOAT(I-1)/NW
      CALL PLOT(IT,P,NW,NLINE,0,0)
      WRITE(6,68)
68    FORMAT('0',T45,'PLOT IN AXIAL DIRECTION (Z .VS. F)')
      IF(IT.NE.ITMAX) GO TO 100
C ------ PLOT IN RADIAL DIRECTION
      DO 70 I=1,11
70    P(I)=I
      NLINE=10
      NPLOT=NW/9+1
      DO 74 I=1,NPLOT
      IF(I.EQ.NPLOT) NLINE=NW-9*I+10
```

```
      NN=NLINE-1
      DO 72 IN=1,NN
      INW=9*(I-1)+IN
      NNN=IN*11
      DO 72 J=1,11
72    P(NNN+J)=F(INW,J)
      CALL PLOT(I,P,11,NLINE,0,0)
      WRITE(6,73)
73    FORMAT('0',T45,'PLOT IN RADIAL DIRECTION (R .VS. F)')
74    CONTINUE
90    WRITE(6,99)
99    FORMAT('1')
100     CONTINUE
      STOP
      END
```

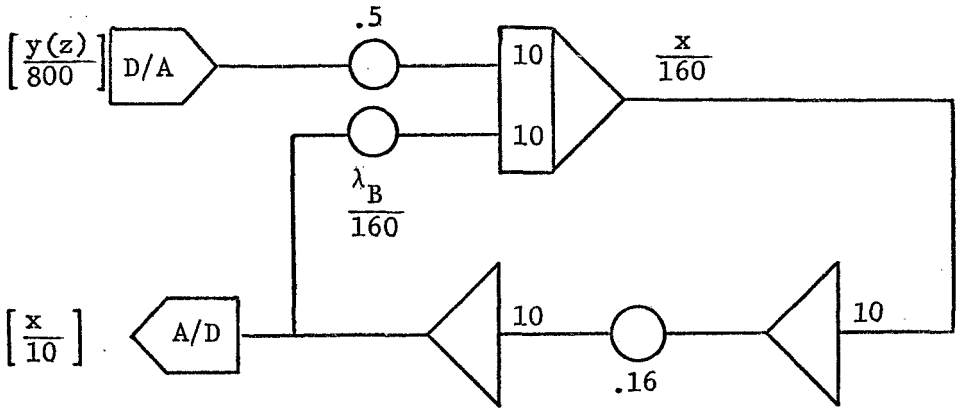
```
      SUBROUTINE HYBGO(LAM1,LAM2,X,PZ,IR)
      COMMON NW,NW1,NW2,DZ,IT,Y
      REAL*8 CCWAN(2),RCBAN(8)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      INTEGER POTS(7)/'P001','P002','P003','P004','P005','P006','P007'/
      INTEGER*2 PVAL(5),LOCS(51),LOCX(51),IC/16/,CTLSTP/1/,CTLGO/0/
      REAL LAM1,LAM2,Y1(50),Y2(50),Y3(50),Y(50),S(50),X(50)
      POLY(A,B,C,D,E,F,X)=A+(B+(C+(D+(E+F*X)*X)*X)*X)*X
2     Z=0.
      YMAX=EXP(LAM1)
      DO 5 I=1,NW
      Y1(I)=EXP(LAM1*Z)/YMAX
      Y2(I)=EXP(LAM2*Z)
5     Z=Z+DZ
      DO 7 I=1,NW
7     S(NW1-I)=X(I)
      CALL MODE(CCWMD,IC)
      CALL FRCBSU(RCBMD,28,CCWMD)
      CALL CONTRL(CCWGO,CTLGO,2)
      CALL FRCBSU(RCBGO,28,CCWGO)
      CALL CONTRL(CCWCT,CTLSTP,2)
      CALL FRCBSU(RCBCT,28,CCWCT)
      CALL FRCBSU(RCBPT,28,CCWPT)
      IF(IR.EQ.11) GO TO 71
      SCALER=800.
      SCALEX=10.
      PVAL(1)=5000
      PVAL(2)=1600
      PVAL(3)=LAM1/1.6*100
      PVAL(4)=-LAM2*1000
      MPA=0
      MPB=257
      CALL POTSS(CCWPT,4,POTS,PVAL,RCBPT)
      GO TO 74
```

```
71    CONTINUE
      SCALER=5000.
      SCALEX=100.
      PVAL(1)=5000
      PVAL(2)=LAM1*100
      PVAL(3)=-LAM2*100
      MPA=514
      MPB=771
      CALL POTSS(CCWPT,3,POTS(5),PVAL,RCBPT)
74    CONTINUE
      CALL FRTIO(RCBMD,IRET)
      CALL FCHECK(RCBMD,IRET,1)
9     CONTINUE
      DO 10 I=1,NW
10    LOCS(I+1)=S(I) *8191./SCALER
      WRITE(6,100)
100   FORMAT('+')
      CALL DACADC(CCWDA,RCBDA,LOCS,MPA,CCWAD,RCBAD,LOCX,MPA,NW)
      CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
11    FORMAT(' LOCS & LOCX..'/(25I5))
      IF(IT.GT.1) GO TO 13
      DO 12 I=1,NW
12    LOCX(I+1)=0
13    DO 14 I=1,NW
      LTEMP=LOCX(NW2-I)
      LOCS(I+1)=MAX0(LTEMP,0)
14    X(I   )=LOCX(NW2-I)/8191.*SCALEX
      X(NW)=0.
      LOCS(NW1)=0
      CALL DACADC(CCWDA,RCBDA,LOCS,MPB,CCWAD,RCBAD,LOCX,MPB,NW)
      CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
      DO 16 I=2,NW
16    Y3(I)=AMAX1(LOCX(I   )/8191.,0.)
169   CONTINUE
      IF(IT.GT.1) GO TO 17
      DO 160 I=1,NW
```

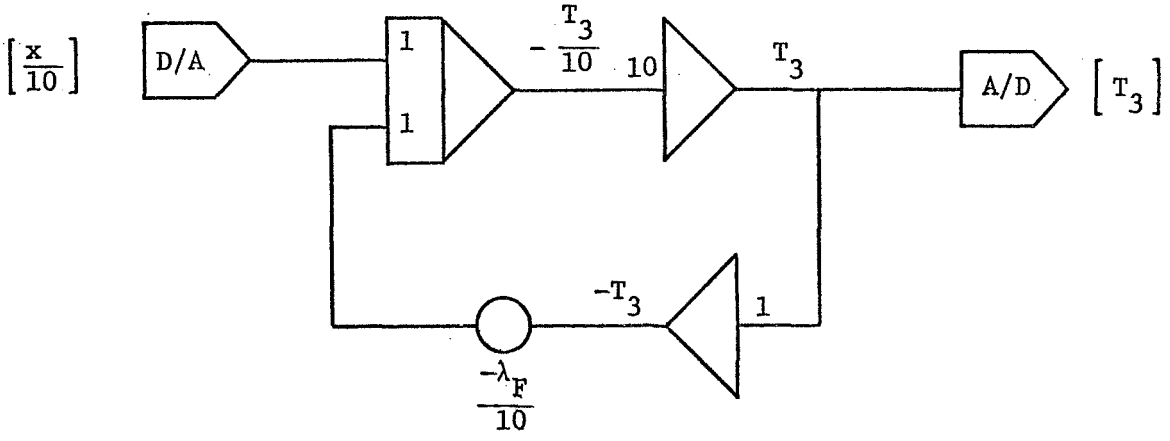
```
160   Y3(I)=0.
17    IF(PZ.LT.1.E-10) GO TO 176
      PP=PZ-LAM2
      QQ=LAM1-PZ
      RR=X(1)+PZ
      A=-LAM2*(RR*Y2(NW)+PP*Y3(NW))/(LAM1*PP+LAM2*QQ*Y1(1)*Y2(NW))
      B=(A*QQ*Y1(1)+RR)/PP
      GO TO 179
176   A=(1.+Y3(NW)/Y2(NW))/(Y1(1)+1./Y2(NW))
      B=A*(1./A-Y1(1))
179   DO 18 I=1,NW
18    Y(I)=A*Y1(I)+B*Y2(I)+Y3(I)
20    FORMAT('0IT=',I3,5X,'A=',G12.4,5X,'B=',G12.4/' Y.. '/(10G12.4))
21    FORMAT(' Y3..'/(10G12.4))
15    FORMAT('0X..'/(10G12.4))
      DO 35 I=25,NW
      IF(Y(I).GT.Y(I-1)) Y(I)=Y(I-1)
35      CONTINUE
      IF(IR.LT.11.OR.IT.EQ.1) RETURN
      X(1)=ALOG(Y(1))
      Y1(1)=0.1
      DO 37 I=3,10
      X(I-1)=ALOG(Y(I))
37    Y1(I-1)=I/10.
      CALL CVFIT(4,9,Y1,X)
      Y1(1)=0.2
      Y(2)=EXP(POLY(X(1),X(2),X(3),X(4),X(5),X(6),0.2))
      RETURN
      END
```

Energy Balance Mechanization for General Radial Lines

$$-\left[\frac{\dot{x}}{160}\right] = \left[\frac{\lambda_B}{160}\right]\left[\frac{x}{10}\right](10) + \left[\frac{y(z)}{800}\right][5] \qquad \text{(G-1)}$$

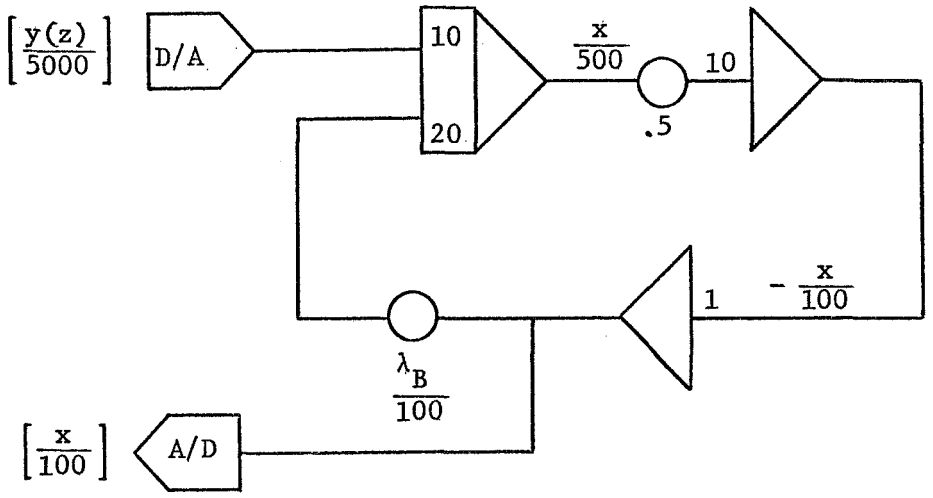


$$\left[\frac{\dot{T}_3}{10}\right] = -\left[\frac{-\lambda_F}{10}\right]\left[T_3\right] + \left[\frac{x}{10}\right] \qquad \text{(G-2)}$$

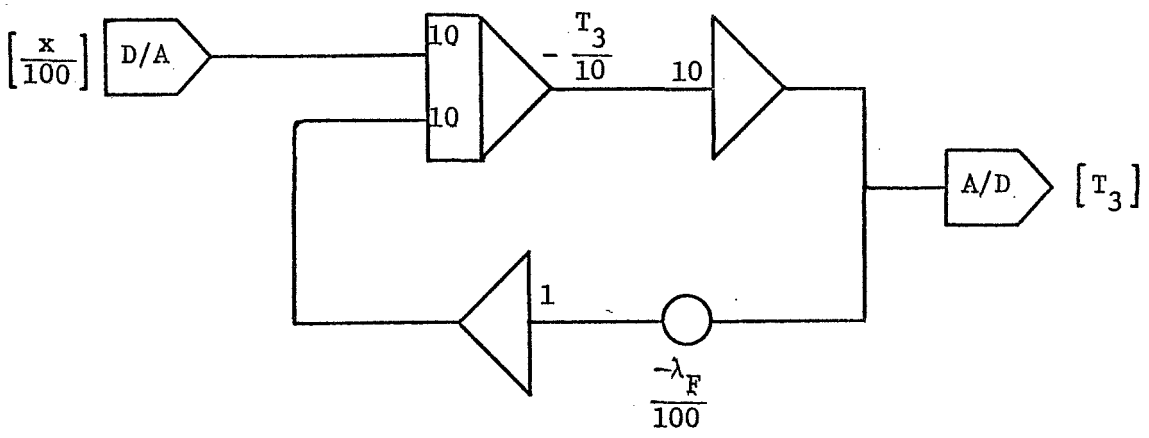


Energy Balance Mechanization at Wall

$$-\left[\frac{\dot{x}}{500}\right] = \left[\frac{\lambda_B}{100}\right]\left[\frac{x}{100}\right](20) + \left[\frac{y(z)}{5000}\right](10) \qquad \text{(G-3)}$$



$$\left[\frac{\dot{T}_3}{10}\right] = \left[\frac{-\lambda_F}{100}\right]\left[-T_3\right](10) + \left[\frac{x}{100}\right](10) \qquad \text{(G-4)}$$



Scaling & Mechanization of Mass Transfer Equation for General Radial Lines

The Riccati Equation:

$$-\dot{\lambda}_F = \lambda_F(Pe_z - \lambda_F) + g(z) \qquad \text{(G-5)}$$

with the estimated maximum values:

$$\text{Max} \mid \dot{\lambda}_F \mid = 100$$

$$\text{Max} \mid \lambda_F \mid = 10$$

$$\text{Max} \mid Pe_z - \lambda_F \mid = 200$$

$$\text{Max} \mid g(z) \mid = 1000$$

Then, we have

$$-\frac{\dot{\lambda}_F}{100} = \left(\frac{\lambda_F}{10}\right)\left[\frac{Pe_z - \lambda_F}{200}\right](20) + \left[\frac{g(z)}{1000}\right][10] \qquad \text{(G-5a)}$$

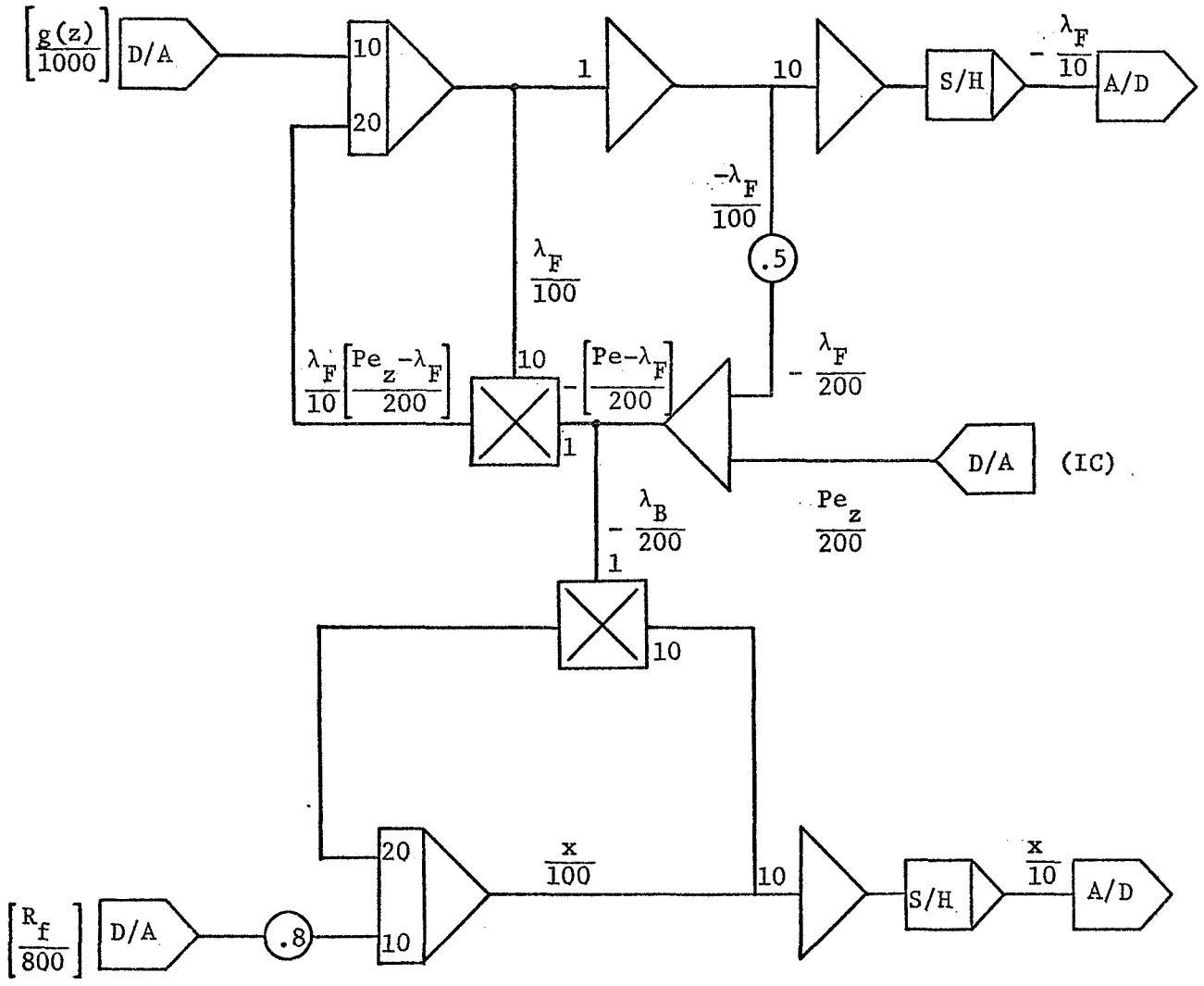
Similarly for:

$$-\dot{x} = (Pe_z - \lambda_F)x + R_f^i(z) \qquad \text{(G-6)}$$

$$-\frac{\dot{x}}{100} = \left[\frac{Pe_z - \lambda_F}{200}\right]\left(\frac{x}{10}\right)(20) + \left[\frac{R_f}{800}\right][8] \qquad \text{(G-6a)}$$

The actual mechanization is shown in the next diagram.

Analog Mechanization of Equations (G.5a) and (G.6a)



Finally for

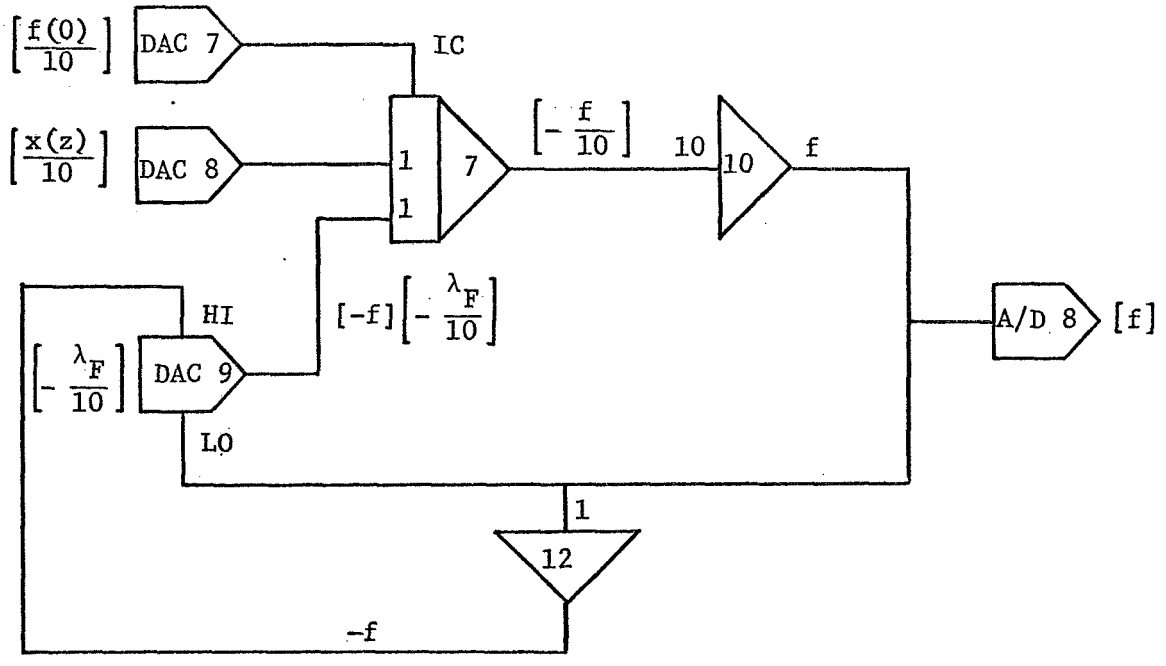
$$\dot{f} = [-\lambda_F]\,[-f] + x \tag{G-7}$$

or in the scaled form

$$\left[\frac{\dot{f}}{10}\right] = \left[-\frac{\lambda_F}{10}\right][-f] + \left[\frac{x}{10}\right] \tag{G-7a}$$

and the IC at $z = 0$,

$$f(0) = \frac{Pe_z + x(0)}{Pe_z - \lambda_F(0)} \qquad (< 1) \tag{G-8}$$



Mechanization of Mass Transfer Equation at Wall Line

The Riccati equation is

$$-\dot{\lambda}_F = -\lambda_F^2 + g(z) \tag{G-9}$$

When the following estimated maximums are applied,

$$\text{Max} \mid \lambda_F \mid = 100$$

$$\text{Max} \mid g(z) \mid = 10000$$

$$\text{Max} \mid \dot{\lambda}_F \mid = 1000$$

the scaled equation becomes

$$-\left[\frac{\dot{\lambda}_F}{1000}\right] = -\left[\frac{\lambda_F}{100}\right]\left[\frac{\lambda_F}{100}\right](10) + \left[\frac{g(z)}{10000}\right](10) \tag{G-9a}$$

Similarly for the equation

$$-\dot{x} = \lambda_B x + R_f^i \tag{G-10}$$

The maximums are

$$\mid \lambda_B \mid = 100$$

$$\mid R_f \mid = 5000$$

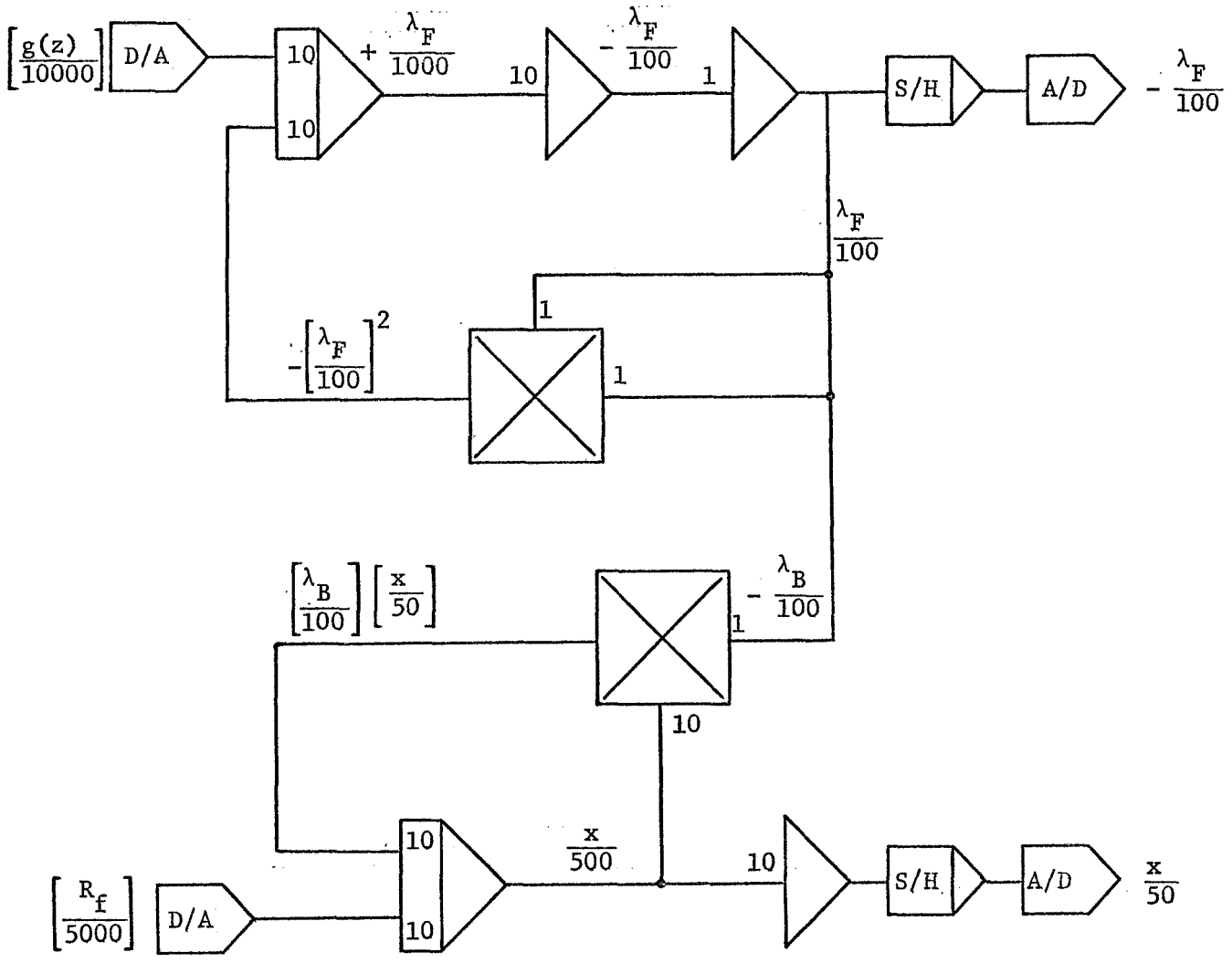
$$\mid x \mid = 50$$

Then, we have

$$-\left[\frac{\dot{x}}{500}\right] = \left[\frac{\lambda_B}{100}\right]\left[\frac{x}{50}\right](10) + \left[\frac{R_f^i}{5000}\right](10) \tag{G-10a}$$

The analog circuit is shown in the next page.

Analog Circuit for Equations (G.9a) and (G.10a)



And finally to integrate

$$\dot{f} = \left[-\lambda_F\right][-f] + x \tag{G-11}$$

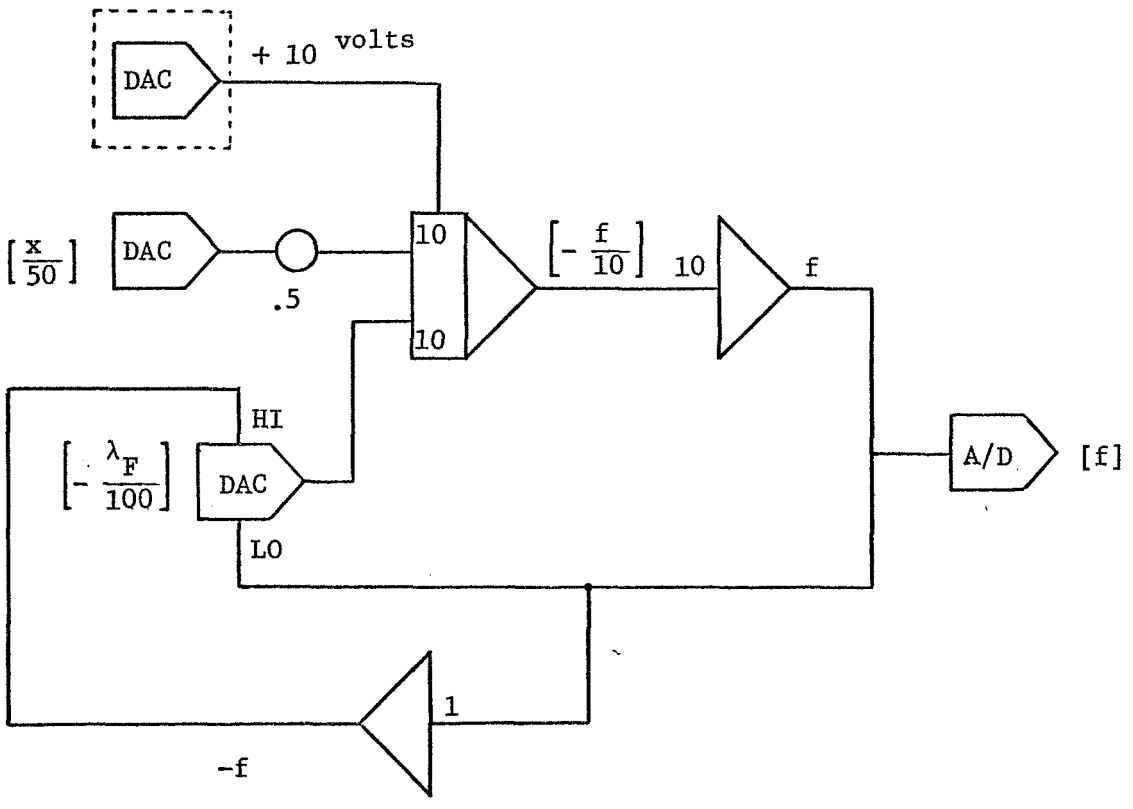
which is scaled to obtain

$$\left[\frac{\dot{f}}{10}\right] = \left[-\frac{\lambda_F}{100}\right][-f](10) + \left[\frac{x}{50}\right](5) \tag{G-11a}$$

The initial condition at the wall line is simply

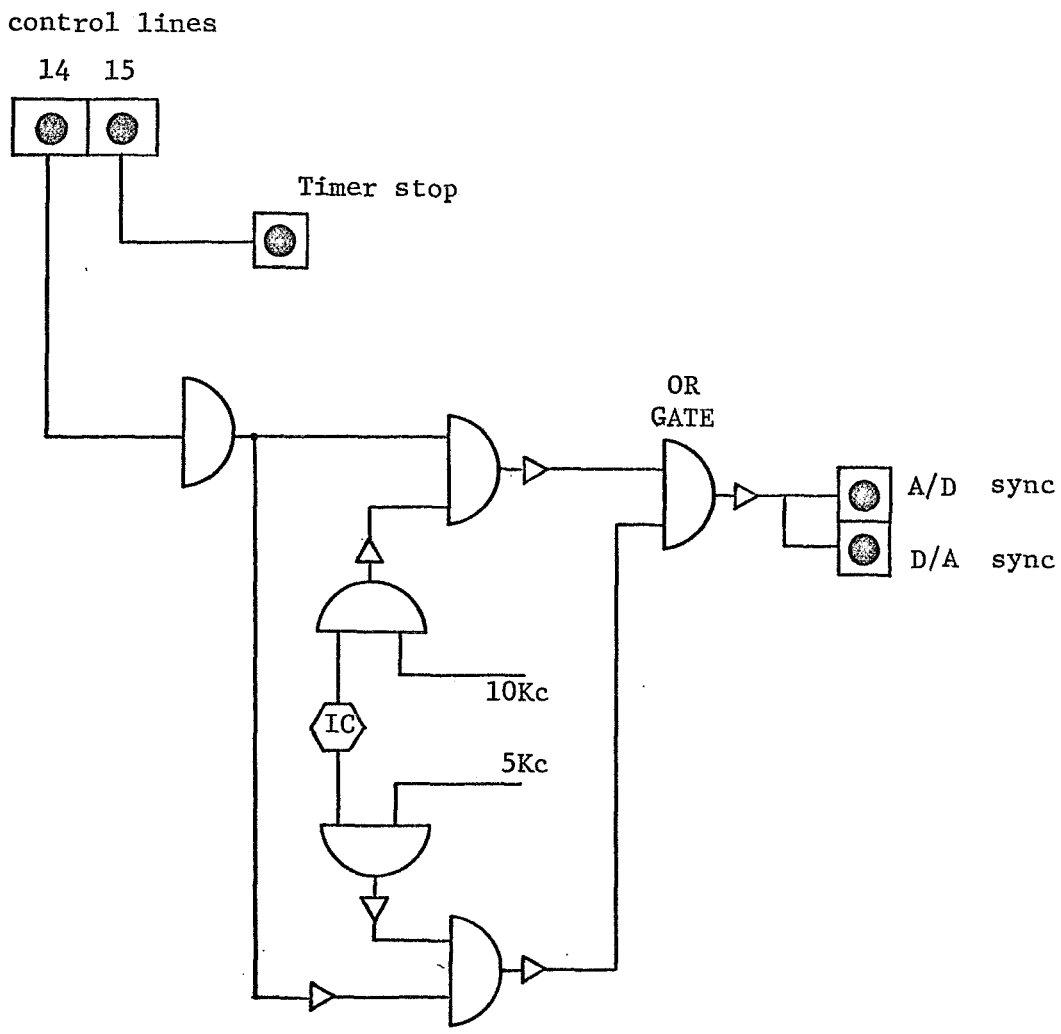
$$f(0) \equiv 1 \tag{G-12}$$

We have, then, the following circuit for equations (G-11a) and (G-12)



Hybrid Control Logic Design

The integration of material balance equation requires the sampling rate twice the speed of the energy balance equation since there are simultaneous sample and hold amplifiers and playback DACs involved. For this particular requirement an additional control line (14) is used to switch the sampling rates from 5 KC to 10 KC and vice versa.



The following digital program listing consists of 13 routines:

1. Main program
2. Subroutine HYBSET : to build command block and channel command word and set some servo-set pots.
3. BLOCK DATA
4. Subroutine HEATUP : set up coefficients for energy equation
5. Subroutine HEATGO : energy balance integration
6. Subroutine MASSUP : set up coefficients for mass equation
7. Subroutine MASSGO : mass balance integration
8. Subroutine MASSIC : set IC for mass balance calculation
9. Subroutine DACADC : set up D/A and A/D channels
10. Subroutine REALGO : realtime I/O, actual integration
11. Subroutine SETPLT : arrange results to be plot
12. Subroutine SETZ : set up plot in axial direction
13. Subroutine SETR : set up plot in radial direction

The subroutine PLOT is a standard SSP routine

DIGITAL PROGRAM LISTING G-1

The Hybrid 'CSDSDT' Decomposition Solution of a Tubular Reactor With Heat and Mass Transfer

```
C ......                                                  ......
C ...... MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN ......
C ......                                                  ......
C ..... ''NON-ISOTHERMAL TUBULAR FLOW REACTOR IN TWO-SPACE DIMENSION''.
C ..... 'CSDSDT' IN TWO-SPACE DIMENSION WITH DECOMPOSITION METHOD.
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
      COMMON/BLK3/ LAM1(50),LAM2(50),RF(50),G(50)
      COMMON/BLK4/ LAMB,LAMF,R(50),T1(50),T2(50)
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      REAL L,LAM1,LAM2,LAMB,LAMF
      LOGICAL FLAG
      EQUIVALENCE(TONORM,TO)
      PAUSE'IF READY, HIT ''EOB''.'
      DT=0.01
      DR=0.1
      DRR=DR*DR
      DZ=1./(NW-1)
99    READ(5,90,END=100) TO,FN,ITMAX
90    FORMAT(2F10.2,I2)
      TONORM=TO/TS
      WRITE(6,91) TONORM,FN,ITMAX
91    FORMAT('1 TONORM=',F8.3,5X,'FN=',F8.3 ,5X,'ITMAX=',I3)
      DO 1 I=1,50
      DO 1 J=1,11
      T(I,J)=TONORM
1     F(I,J)=FN
      CALL HYBSET
      DO 88 IT=1,ITMAX
      DO 80 J=1,11
```

```
      FLAG=.TRUE.
      DO 80 IX=1,2
      CALL HEATUP
      CALL MASSUP
      FLAG=.FALSE.
80    CONTINUE
      DO 82 I=1,NW
      T(I,11)=TNEW(I)
82    F(I,11)=FNEW(I)
      CALL SETPLT(ITMAX)
      IF(IT.NE.ITMAX) GO TO 88
      DO 85 J=1,11
      WRITE(6,84) J,(T(I,J),I=1,NW)
84    FORMAT('0T(50,',I2,')=...'/(10G12.4))
85    CONTINUE
      DO 87 J=1,11
      WRITE(6,86) J,(F(I,J),I=1,NW)
86    FORMAT('0F(50,',I2,')=...'/(10G12.4))
87    CONTINUE
88    CONTINUE
      GO TO 99
100     CONTINUE
      STOP
      END
```

```
      SUBROUTINE HYBSET
C ...... USING CONTROL LINE '15' TO TRIGGER TIMER.
C ......        CONTROL LINE '14' TO CHANGE SAMPLING RATE
      INTEGER*2 IC/16/,CTLSTP/1/,CTLGO/Z2/,CTLGM/0/
      INTEGER POTS(6)/'P205','P206','P207','P002','P003','P011'/
      INTEGER*2 PVAL(6)/5000,1600,2*5000,8000,5000/
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
1     CALL MODE(CCWMD,IC)
      CALL FRCBSU(RCBMD,28,CCWMD)
      CALL CONTRL(CCWGO,CTLGO,2)
      CALL FRCBSU(RCBGO,28,CCWGO)
      CALL CONTRL(CCWGM,CTLGM,2)
      CALL FRCBSU(RCBGM,28,CCWGM)
      CALL CONTRL(CCWCT,CTLSTP,2)
      CALL FRCBSU(RCBCT,28,CCWCT)
      CALL FRCBSU(RCBPT,28,CCWPT)
      CALL POTSS(CCWPT,6,POTS,PVAL,RCBPT)
      RETURN
      END
```

```
BLOCK DATA
COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
INTEGER NW/50/,NW1/51/,NW2/52/,NNW/100/,NNW1/101/,NNW2/102/
REAL L/10./,TIME/10./,DXW/0.2/,RI/1.0/,TS/700./,HC/1./
DATA PEZ/110.,100.,80.,65.,2*55.,60.,70.,90.,150./
DATA U/25.8,25.7,25.5,25.1,24.8,24.4,23.8,23.1,21.9,19.8/
END
```

```
      SUBROUTINE HEATUP
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
      COMMON/BLK3/ LAM1(50),LAM2(50),RF(50),G(50)
      COMMON/BLK4/ LAMB,LAMF,R(50),T1(50),T2(50)
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
      LOGICAL FLAG
      REAL KC,LAMF,LAMB,L,LAM1,LAM2
      KC(T)=6.427E15/EXP(50610./(1.987*T*TS))
      IF(.NOT.FLAG) GO TO 49
      IF(J.EQ.11) GO TO 30
      PZ=PEZ(J)
      PER=PZ
      DDT=L/(U(J)*TIME*DT)
      SQ=SQRT(PZ*(PZ+4.*DDT))
      LAMF=(PZ-SQ)/2.
      LAMB=(PZ+SQ)/2.
      IF(J.GT.1) GO TO 20
C ------ CENTER LINE
      CP=4./DRR
      DO 18 I=1,NW
18    R(I)=-PZ/PER*(CP*T(I,2)+(DDT*PER-CP)*T(I,1))
      GO TO 40
C ------ GENERAL LINES
20    CONTINUE
      CP=(1.+0.5/(J-1))/DRR
      CQ=-2./DRR
      CS=-CP-CQ
22    DO 24 I=1,NW
24    R(I)=-PZ/PER*(CP*T(I,J+1)+(CQ+DDT*PER)*T(I,J)+CS*T(I,J-1))
      GO TO 40
C ------- WALL BOUNDARY
30    CONTINUE
      XW=L*L/(DXW*TIME)
      VW=L*L/(RI*RI*100.)
```

```
      LAMB= SQRT(VW*HC+XW/DT)
      LAMF=-LAMB
      CQ=VW/DRR
      CP=CQ+CQ-XW/DT
      CS=VW*HC+CQ
      PZ=0.
      DO 32 I=1,NW
32    R(I)=CP*T(I,11)-CQ*T(I,10)-CS
40    CONTINUE
      IF(J.EQ.1) GO TO 47
      DO 46 I=1,NW
46    T(I,J-1)=TNEW(I)
47    DO 48 I=1,NW
      LAM2(I)=FNEW(I)
      FNEW(I)=F(I,J)
48    TNEW(I)=T(I,J)
C ------ COMPUTE REACTION TERM
49    CONTINUE
      DO 50 I=1,NW
50    Y(I)=KC(TNEW(I))*FNEW(I)
C ..... Y(I) IN COMMON, UPON RETURN TO BE STORED IN TNEW(I)
      CALL HEATGO
      RETURN
      END
```

```
      SUBROUTINE HEATGO
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
      COMMON/BLK4/ LAM1,LAM2,RT(50),Y1(50),Y2(50)
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      LOGICAL FLAG
      INTEGER POTS(4)/'P201','P202','P203','P204'/
      INTEGER*2 PVAL(2),LOCS(51),LOCX(51)
      REAL LAM1,LAM2,S(50),X(50),Y3(50),L
      POLY(A,B,C,D,E,F,X)=A+(B+(C+(D+(E+F*X)*X)*X)*X)*X
1     IF(.NOT.FLAG) GO TO 6
2     Z=0.
      YMAX=EXP(LAM1)
      DO 5 I=1,NW
      Y1(I)=EXP(LAM1*Z)/YMAX
      Y2(I)=EXP(LAM2*Z)
5     Z=Z+DZ
6     CONTINUE
      DO 7 I=1,NW
7     S(NW1-I)=RT(I)+Y(I)
      IF( J.EQ.11) GO TO 71
      SCALER=800.
      SCALEX=10.
      PVAL(1)=LAM1/1.6*100
      PVAL(2)=-LAM2*1000
      MPA=0
      MPB=257
      CALL POTSS(CCWPT,2,POTS,PVAL,RCBPT)
      GO TO 74
71    CONTINUE
```

```
      SCALER=5000.
      SCALEX=100.
      PVAL(1)=LAM1*100
      PVAL(2)=-LAM2*100
      MPA=514
      MPB=771
      CALL POTSS(CCWPT,2,POTS(3),PVAL,RCBPT)
74    CONTINUE
      CALL FRTIO(RCBMD,IRET)
      CALL FCHECK(RCBMD,IRET,1)
9     CONTINUE
      DO 10 I=1,NW
10    LOCS(I+1)=S(I) *8191./SCALER
      WRITE(6,100)
100   FORMAT('+')
      CALL DACADC(LOCS,MPA,LOCX,MPA,NW,10)
      CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
11    FORMAT(' LOCS & LOCX..'/(1X,25I5))
13    DO 14 I=1,NW
      LTEMP=LOCX(NW2-I)
      LOCS(I+1)=MAX0(LTEMP,0)
14    X(I   )=LOCX(NW2-I)/8191.*SCALEX
      X(NW)=0.
      LOCS(NW1)=0
      CALL DACADC(LOCS,MPB,LOCX,MPB,NW,10)
      CALL REALGO(RCBCT,RCBGO,RCBDA,RCBAD)
      DO 16 I=2,NW
16    Y3(I)=AMAX1(LOCX(I+1)/8191.,0.)
169   CONTINUE
      Y3(1)=0.
176   A=(TO+Y3(NW)/Y2(NW))/(Y1(1)-LAM1/(Y2(NW)*LAM2))
      B=A*(TO/A-Y1(1))
179   DO 18 I=1,NW
18    TNEW(I)=A*Y1(I)+B*Y2(I)+Y3(I)
      WRITE(6,20) IT,A,B,TNEW
20    FORMAT('0IT=',I3,5X,'A=',G12.4,5X,'B=',G12.4/' T.. '/(10G12.4))
```

```
21    FORMAT(' Y3..'/(10G12.4))
15    FORMAT('0X..'/(10G12.4))
      RETURN
      END
```

```
      SUBROUTINE MASSUP
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
      COMMON/BLK3/ LAM1(50),LAM2(50),RF(50),G(50)
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
      REAL KC,L,LAM1,LAM2
      LOGICAL FLAG
      KC(T)=6.427E15/EXP(50610./(1.987*T*TS))
C ...... COMPUTE RF(I)'S
      IF(J.EQ.11) GO TO 30
      PZ=PEZ(J)
      PER=PZ
      DDT=L/(U(J)*TIME*DT)
      IF(.NOT.FLAG) GO TO 25
      IF(J.GT.1) GO TO 20
      CP=4./DRR
      DO 18 I=1,NW
18    RF(I)=-PZ/PER*(CP*F(I,2)+(DDT*PER-CP)*F(I,1))
      GO TO 25
20    CP=(1.+0.5/(J-1))/DRR
      CQ=-2./DRR
      CS=-CP-CQ
      DO 24 I=1,NW
24    RF(I)=-PZ/PER*(CP*F(I,J+1)+(CQ+DDT*PER)*F(I,J)+CS*F(I,J-1))
25    DO 26 I=1,NW
26    G(I)=PZ*(L*KC(TNEW(I))/U(J)+DDT)
      GO TO 40
30    ELL=L*L
      XT=ELL/(DXW*TIME*DT)
      IF(.NOT.FLAG) GO TO 34
      VW=ELL/(RI*RI*100.)
      CQ=VW/DRR
      CP=CQ-XT
      DO 32 I=1,NW
32    RF(I)=CP*F(I,11)-CQ*F(I,10)
```

```
34    DO 36 I=1,NW
36    G(I)=XT+ELL*KC(TNEW(I))/DXW
40    CONTINUE
      IF(J.EQ.1.OR..NOT.FLAG) GO TO 47
      DO 42 I=1,NW
42    F(I,J-1)=LAM2(I)
47    CONTINUE
      WRITE(6,50) J,FNEW(1),RF(1),G(1)
50    FORMAT(' J=',I3,'  FNEW,RF,G..',3G12.4)
      CALL MASSGO
      RETURN
      END
```

```
      SUBROUTINE MASSGO
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/BLK2/ DZ,L,DR,RI,DT,TIME,DXW,DRR,HC,TS,TO
      COMMON/BLK3/ LAM1(50),LAM2(50),RF(50),G(50)
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      REAL L,LAM1,LAM2
      INTEGER*2 LOCS(101),LOCX(101)
      IF(J.EQ.11) GO TO 10
      PZ=PEZ(J)
      LOCS(1)=1028
      LOCS(2)=PZ/200.*8191.
      CALL MASSIC(LOCS,1)
      SCALEG=8.191
      SCALER=8191./800.
      MPA=1286
      MPB=2057
      GO TO 12
10    SCALEG=0.8191
      SCALER=8191./5000.
      MPA=2571
      MPB=3085
12    DO 14 I=1,NW
      IX=2*(I-1)
      LOCS(NNW-IX)=G(I)*SCALEG
14    LOCS(NNW1-IX)=RF(I)*SCALER
      CALL DACADC(LOCS,MPA,LOCX,MPA,NNW, 3)
      CALL REALGO (RCBCT,RCBGM,RCBDA,RCBAD)
11    FORMAT(' LOCS & LOCX..'/(1X,25I5))
      IF(J.EQ.11) GO TO 16
      X0=LOCX(NNW1)/819.1
```

```
      RAM=LOCX(NNW)/819.1
      FO=(XO+PZ)/(RAM+PZ)
      WRITE(6,15) J,XO,RAM,FO
15    FORMAT(' J,XO,LAMFO,FO..',I5,3G12.4)
      LOCS(1)=1799
      LOCS(2)=AMIN1(FO,1.)*819.1
      GO TO 18
16    LOCS(1)=3598
      LOCS(2)=819.1
18    CALL MASSIC(LOCS,1)
      DO 20 I=1,NNW
20    LOCS(NNW2-I)=LOCX(I+1)
      CALL DACADC(LOCS,MPB,LOCX,MPB,NNW, 3)
      CALL REALGO(RCBCT,RCBGM,RCBDA,RCBAD)
      DO 22 I=1,NW
22    FNEW(I)=LOCX(2*I) /8191.
      WRITE(6,25) FNEW
25    FORMAT(' FNEW..'/(10G12.4))
      RETURN
      END
```

```
      SUBROUTINE MASSIC(LOCS,N)
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      INTEGER*2 LOCS(1)
      CALL WRITDA(CCWDA,0,N,LOCS)
      CALL FRCBSU(RCBDA,30,CCWDA)
      CALL FRTIO(RCBDA,IRET)
      CALL FCHECK(RCBDA,IRET,1)
      RETURN
      END
```

```
      SUBROUTINE DACADC(LDA,DA,LAD,AD,NW,ICON)
      COMMON/BLKH/ CCWAN,RCBAN,CCWMD,RCBMD,CCWGO,RCBGO,CCWCT,RCBCT,
     $ CCWPT,RCBPT,CCWDA,RCBDA,CCWAD,RCBAD,CCWGM,RCBGM
      REAL*8 CCWAN(2),RCBAN(8),CCWGM(2),RCBGM(4)
      REAL*8 CCWMD(2),RCBMD(4),CCWGO(2),RCBGO(4),CCWCT(2),RCBCT(4)
      REAL*8 CCWPT(2),RCBPT(4),CCWDA(2),RCBDA(4),CCWAD(2),RCBAD(4)
      INTEGER*2 LDA(1),LAD(1)
      INTEGER  DA,AD
      LDA(1)=DA
      LAD(1)=AD
      CALL WRITDA(CCWDA,ICON,NW,LDA)
      CALL FRCBSU(RCBDA,30,CCWDA)
      CALL READAD(CCWAD,NW,ICON,LAD)
      CALL FRCBSU(RCBAD,29,CCWAD)
      RETURN
      END
```

```
SUBROUTINE REALGO(CT,GO,DA,AD)
REAL*8 CT(1),GO(1),DA(1),AD(1)
CALL FRTIO(CT,IRT)
CALL FCHECK(CT,IRT,1)
CALL FRTIO(GO,IRT)
CALL FRTIO(DA,IRT)
CALL FRTIO(AD,IRT)
CALL FCHECK(AD,IRT,1)
CALL FRTIO(CT,IRT)
RETURN
END
```

```
      SUBROUTINE SETPLT(ITMAX)
      COMMON/BLK1/ NW,NW1,NW2,NNW,NNW1,NNW2,IT,J,FLAG
      COMMON/B4/PEZ(10),U(10),Y(50),FNEW(50),TNEW(50),F(50,11),T(50,11)
C ......... PLOT ROUTINE ........
      REAL P(500)
C ------ PLOT IN AXIAL DIRECTION
      DO 66 I=1,NW
66    P(I)=FLOAT(I-1)/NW
      CALL SETZ(P,T,NW,NLINE,IT)
      WRITE(6,68)
      CALL SETZ(P,F,NW,NLINE,IT)
      WRITE(6,69)
68    FORMAT('0',T45,'PLOT IN AXIAL DIRECTION (Z .VS. T)')
69    FORMAT('0',T45,'PLOT IN AXIAL DIRECTION (Z .VS. F)')
      IF(IT.NE.ITMAX) GO TO 100
C ------ PLOT IN RADIAL DIRECTION
      DO 70 I=1,11
70    P(I)=I
      NLINE=10
      NPLOT=NW/9+1
      DO 72 I=1,NPLOT
      CALL SETR(I,NPLOT,NLINE,NW,P,T)
      WRITE(6,71)
71    FORMAT('0',T45,'PLOT IN RADIAL DIRECTION (R .VS. T)')
72    CONTINUE
      NLINE=10
      DO 74 I=1,NPLOT
      CALL SETR(I,NPLOT,NLINE,NW,P,F)
      WRITE(6,73)
73    FORMAT('0',T45,'PLOT IN RADIAL DIRECTION (R .VS. F)')
74    CONTINUE
90    WRITE(6,99)
99    FORMAT('1')
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETZ(P,F,NW,NLINE,IT)
      REAL P(1),F(50,11)
60    NLINE=1
      K=2
      DO 64 J=1,11,K
      IF(J.GE.5) K=1
      INW=NLINE*NW
      DO 62 I=1,NW
62    P(INW+I)=F(I,J)
64    NLINE=NLINE+1
      CALL PLOT(IT,P,NW,NLINE,0,0)
      RETURN
      END
```

```
      SUBROUTINE SETR(I,NPLOT,NLINE,NW,P,F)
      REAL P(1),F(50,11)
      IF(I.EQ.NPLOT) NLINE=NW-9*I+10
      NN=NLINE-1
      DO 72 IN=1,NN
      INW=9*(I-1)+IN
      NNN=IN*11
      DO 72 J=1,11
72    P(NNN+J)=F(INW,J)
      CALL PLOT(I,P,11,NLINE,0,0)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE PLOT
C
C        PURPOSE
C           PLOT SEVERAL CROSS-VARIABLES VERSUS A BASE VARIABLE
C
C        USAGE
C           CALL PLOT (NO,A,N,M,NL,NS)
C
C        DESCRIPTION OF PARAMETERS
C           NO - CHART NUMBER (3 DIGITS MAXIMUM)
C           A  - MATRIX OF DATA TO BE PLOTTED. FIRST COLUMN REPRESENTS
C                BASE VARIABLE AND SUCCESSIVE COLUMNS ARE THE CROSS-
C                VARIABLES (MAXIMUM IS 9).
C           N  - NUMBER OF ROWS IN MATRIX A
C           M  - NUMBER OF COLUMNS IN MATRIX A (EQUAL TO THE TOTAL
C                NUMBER OF VARIABLES). MAXIMUM IS 10.
C           NL - NUMBER OF LINES IN THE PLOT. IF 0 IS SPECIFIED, 50
C                LINES ARE USED.
C           NS - CODE FOR SORTING THE BASE VARIABLE DATA IN ASCENDING
C                ORDER
C                  0  SORTING IS NOT NECESSARY (ALREADY IN ASCENDING
C                     ORDER).
C                  1  SORTING IS NECESSARY.
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C     ..................................................................
C
```

```
      SUBROUTINE PLOT(NO,A,N,M,NL,NS)
      DIMENSION OUT(101),YPR(11),ANG(9),A(1)
C
    1 FORMAT(1H1,60X,7H CHART ,I3,//)
    2 FORMAT(1H ,F11.4,5X,101A1)
    3 FORMAT(1H )
    7 FORMAT(1H ,16X,101H.         .         .         .         .
     1   .         .         .         .         .         .)
    8 FORMAT(1H0,9X,11F10.4)
C
C     ..............................................................
C
      NLL=NL
C
      IF(NS) 16, 16, 10
C
C        SORT BASE VARIABLE DATA IN ASCENDING ORDER
C
   10 DO 15 I=1,N
      DO 14 J=I,N
      IF(A(I)-A(J)) 14, 14, 11
   11 L=I-N
      LL=J-N
      DO 12 K=1,M
      L=L+N
      LL=LL+N
      F=A(L)
      A(L)=A(LL)
   12 A(LL)=F
   14 CONTINUE
   15 CONTINUE
C
C        TEST NLL
C
   16 IF(NLL) 20, 18, 20
   18 NLL=50
```

```
C
C        PRINT TITLE
C
   20 WRITE(6,1)NO
C
C        DEVELOP BLANK AND DIGITS FOR PRINTING
C
      DATA BLANK,ANG/' ','1','2','3','4','5','6','7','8','9'/
C
C        FIND SCALE FOR BASE VARIABLE
C
      XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
C
C        FIND SCALE FOR CROSS-VARIABLES
C
      M1=N+1
      YMIN=A(M1)
      YMAX=YMIN
      M2=M*N
      DO 40 J=M1,M2
      IF(A(J)-YMIN) 28,26,26
   26 IF(A(J)-YMAX) 40,40,30
   28 YMIN=A(J)
      GO TO 40
   30 YMAX=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/100.0
C
C        FIND BASE VARIABLE PRINT POSITION
C
      XB=A(1)
      L=1
      MY=M-1
      I=1
   45 F=I-1
      XPR=XB+F*XSCAL
```

```
      IF(A(L)-XPR) 50,50,70
C
C        FIND CROSS-VARIABLES
C
   50 DO 55 IX=1,101
   55 OUT(IX)=BLANK
      DO 60 J=1,MY
      LL=L+J*N
      JP=((A(LL)-YMIN)/YSCAL)+1.0
      OUT(JP)=ANG(J)
   60 CONTINUE
C
C        PRINT LINE AND CLEAR, OR SKIP
C
      WRITE(6,2)XPR,(OUT(IZ),IZ=1,101)
      L=L+1
      GO TO 80
   70 WRITE(6,3)
   80 I=I+1
      IF(I-NLL) 45, 84, 86
   84 XPR=A(N)
      GO TO 50
C
C        PRINT CROSS-VARIABLES NUMBERS
C
   86 WRITE(6,7)
      YPR(1)=YMIN
      DO 90 KN=1,9
   90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
      YPR(11)=YMAX
      WRITE(6,8)(YPR(IP),IP=1,11)
      RETURN
      END
```

Generation of Velocity and Eddy Viscosity Profiles

The method is essentially following the previous work ([14]). The representative formulae are

1. Laminar layer

$$\tau_0 g_c = (\mu + \rho n^2 uy) \frac{du}{dy}$$

with $y_0 = 0$; $u_0 = 0$ and $\varepsilon_0 = 0$

also $\varepsilon = n^2 u y$

2. Turbulent core

$$\tau g_c = (\mu + \rho\varepsilon) \frac{du}{dy}$$

In the turbulent flow tubular reactor, laminar layer thickness, though, may be small, its contribution could affect the entire profiles. Because these velocities and eddy viscosities are evaluated at the different incremental intervals, in order that they could be played back in the equal sampling period, curve fits for U and ε are performed by using subroutine UEFIT. The equations provided the best fits are:

$$u \cdot y = a + by + cy^2 + dy^3 + ey^4 + fy^5$$

and

$$\varepsilon = a' + b'y + c'y^2 + d'y^3 + e'y^4 + f'y^5$$

The predicted and the fitted results are shown in Figures G-1 and G-2 respectively.

The digital program which calculates velocity and eddy viscosity profiles is shown in Listing G-2.

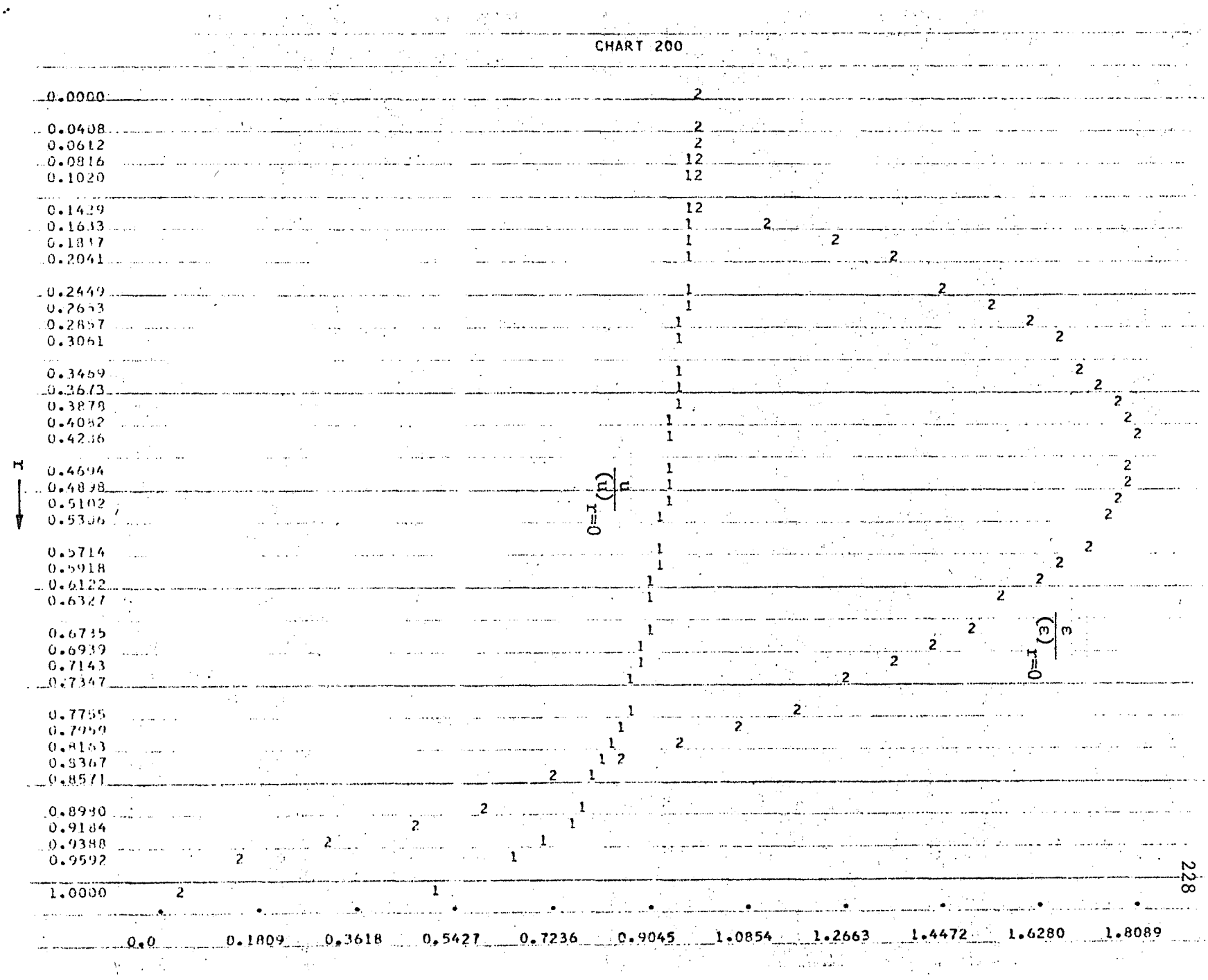


Figure G-1 Predicted Velocity and Eddy Viscosity Profiles

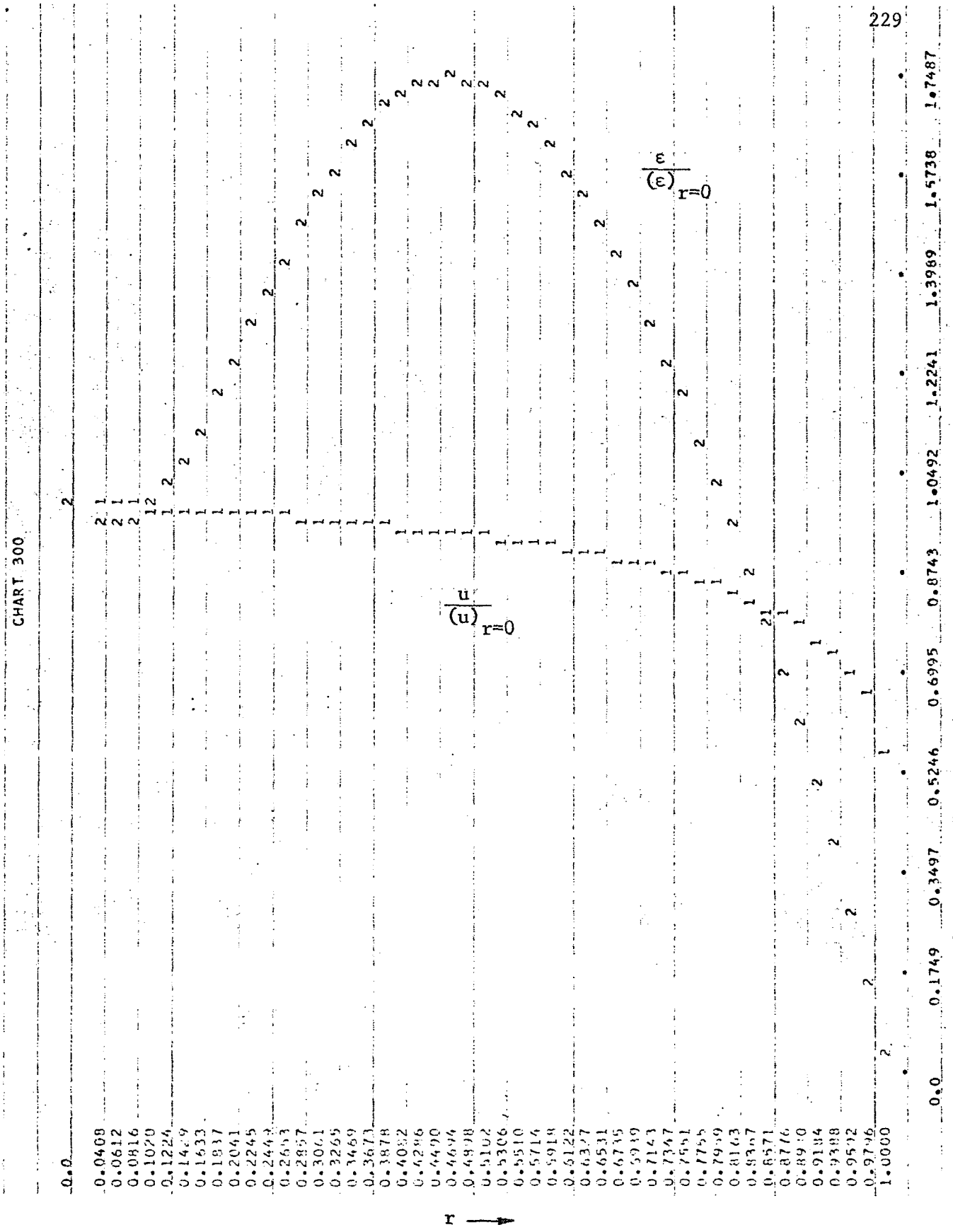


Figure G-2 Curve Fitted Velocity and Eddy Viscosity Profiles Using 5th Order Polynomials

DIGITAL PROGRAM LISTING G-2

The Velocity and Eddy Viscosity Evaluation in a Turbulent Flow Tubular Reactor

```
      SUBROUTINE BEGIN (IDENT)
      COMMON/BEGMNT/ IPRINT,LOOP,ITER,XBL,MPM,LPL,PFLAG,P(500)
      COMMON/A/ TB(50),CB(50),XB(50),VB(50)
      COMMON/B/ T(1,30),C(1,30),X(1,30),TC(1,30)
      COMMON/DIM/ R,AR,DZ,TWODZ,L,M,N,LP1,LM1,M1,N1,RR(50),DR
      COMMON/G/ ALPHA,Q,FO,RO,TO,TS,DFS,CO,CON
      COMMON/ENERGY/ G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     - G16,THKA,THKB,THKC,THKD,PRN,SCN,REY,EA,AC,ORDER,INBC,IZBND,IRBND
      CBIF(X,Y)=(ALPHA*(1.-X)*Y)/(1.+(Q-1.)*ALPHA*X)
      NAMELIST/INPUT/ IPROB,L,M,N,IRBND,IZBND,LOOP,INBC,R,DZ,DFS,CON,
     - ORDER,TO,TS,Q,ALPHA,FO,RO,AC,EA,THKA,THKB,THKC,THKD,TB,XB,
     - G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,G16,IPRINT
C
      READ(5,INPUT,END=901)
      WRITE (6,500)
500   FORMAT('1',T34,'<  HYBRID SIMULATION OF A TUBULAR REACTOR  >'//
     $T30,'(CSDT WITH ALTERNATING-DIRECTION IMPLICIT METHOD)'///)
          IDENT=IPROB
      WRITE (6,535) IPROB
      IF(IRBND.EQ.1) WRITE(6,531)
      IF(IRBND.EQ.2) WRITE(6,532)
      WRITE(6,542)
      IF(INBC .EQ.1) WRITE(6,543)
      IF(INBC .EQ.2) WRITE(6,544)
      IF(IZBND.EQ.1) WRITE(6,533)
      IF(IZBND.EQ.2) WRITE(6,534)
      WRITE(6,511) L,M,N,      LOOP,DZ,DFS,CON,FO,RO,AC,EA,ORDER,TO,TS,
     -Q,ALPHA,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,G16,R,
     -THKA,THKB,THKC,THKD
      WRITE            (6, 512) (TB(I),I=1,L)
      WRITE            (6, 513) (XB(I),I=1,L)
      WRITE (6,514)
      MPM=M+M
      LPL=L+L
      M1=M-1
```

```
      N1=N-1
      LP1=L+1
      LM1=L-1
      TWODZ=2.*DZ
      AR=3.1415927D0*R*R
      CO=CBIF(XB(1),RO)
C  ..... GENERATE 2-DIMENSIONAL VALUES :
      DO 12  I=1,L
      DO 11  J=1,N
      C (I,J)=CO
      T (I,J)=TB(I)
   11 X (I,J)=XB(I)
C  ..... CONSTANT WALL TEMPERATURE:
   12 T (I,N)=TS
C  ..... GENERATE SPACE VARIABLES; RR(I) & DR
      FM1=M1
      DR=R/FM1
      DO 15 J=1,M
15    RR(J)=R*(J-1)/FM1
      RETURN
  901 STOP
  511 FORMAT(40X,' LIST OF INPUT DATA'//T4,'L,M,N,LOOP,'
     1T27,4I12/   26H   DZ,DFS,CON,FO,RO,AC,EA,  7E12.4/23H   ORDER,TO,T
     3S,Q,ALPHA,  5F12.2/
     526H   SPECIFIC HEAT CONSTANTS /18H   G1,G2,G3,G4,G5,8X, 5E12.4 /
     619H   G6,G7,G8,G9,G10, 7X, 5E12.4 /26H   G11,G12,G13,G14,G15,
     75E12.4 /22H   G16,RADIUS,THKABCD,  4X,6E12.4 /)
  512 FORMAT (13H   TB(1-L)..../ (10X,10F10.2) )
  513 FORMAT (/13H   XB(1-L)... / (10X,10F10.7) )
  514 FORMAT (/11H   UNITS.../11X,16HENERGY   -   CAL/11X,17HMASS     -
     1  GMOL/ 11X,15HLENGTH   -   FT/11X,16HTIME     -   SEC/11X,14HTEMP
     2.    -   K)
531   FORMAT(7X,'RADIAL MASS BOUNDARY :  (DC/DR)=0'/)
  532 FORMAT(7X,'RADIAL MASS BOUNDARY :  -2(D/R)*(DC/DR) = K*C'/)
  533 FORMAT(27X,36HNO REACTION OUTLET, (DC/DZ)=0  @ Z=L // )
  534 FORMAT(27X,'(D(DC/DZ)/DZ) @ L-DZ = (D(DC/DZ)DZ) @ L'///)
```

```
535 FORMAT (5X,7HPROBLEMI10//5X,  'THE BOUNDARY CONDITIONS :'//
   $ 7X,'CONSTANT WALL TEMPERATURE'/)
542 FORMAT(7X,'AXIAL BOUNDARY CONDITIONS:')
543 FORMAT(27X,44HCONSTANT INLET WITH CONTINUOUS FUNCTIONS AND)
544 FORMAT(27X,'C(0)=C(0+)-(D/U)*(DC/DZ), AND ')
    END
```

```
      SUBROUTINE MNTUM(Z,*)
      REAL KONF
      COMMON/A/ TB(50),CB(50),XB(50),VB(50)
      COMMON/DIM/ R,AR,DZ,TWODZ,L,M,N,LP1,LM1,M1,N1,RR(50),XDR
      COMMON/G/ ALPHA,Q,FO,RO,TO,TS,DFS,CO,CON
      COMMON/ENERGY/ DUMMY(20),PRN,SCN,REY,EA,AC,ORDER,INBC,IZBND,IRBND
      COMMON/BEGMNT/ IPRINT,LOOP,ITER,XBL,MPM,LPL,PFLAG,PP(500)
      LOGICAL PFLAG
C     HASTINGS CONSTANTS
      REAL P/.47047/,A1/.3084284/,A2/-.0849713/,A3/.6627698/
C ------ USING 10/39 INCREMENTS FOR LAM/TURB ZONES
      DIMENSION UT(50),EDT(50),Y(50)
      RHOX(T,X)=RO*TO/(T*(1.-X))
      TB1=TB(1)
      TBL=TB(L)
      TBA=(TB1+TBL)/2.
      XB1=XB(1)
      XBL=XB(L)
      XS=(XB1+XBL)/2.
      RHOIN=RHOX(TB1,XB1)
      RHOUT=RHOX(TBL,XBL)
      RHOAV=(RHOIN+RHOUT)/2.
C     FANNINGS FRICTION FACTOR, WALL SHEAR STRESS AND FILM THICKNESS
C      CALCULATIONS WITH WILKE VISCOSITY EQUATION TO ESTIMATE THE
C       MIXTURE VISCOSITY.
      IF (ITER-1) 25,25,26
   25 UB=FO*(1.+(Q-1.)*ALPHA*XS)/(AR*RHOAV)
      SCN=VISC(TB1,XB(1))/(RO*DFS)
      GO TO 28
   26 UB=(VB(1)+VB(L))/2.
28    VISIN=VISC(TB1,XB1)
      VISOT=VISC(TBL,XBL)
      VISMB=(VISIN+VISOT)/2.
      REY=2.*R*UB*RHOAV/VISMB
      EFU=0.046/(REY**0.2)
```

```
      TAU=EFU*UB*UB*RHOAV/2.
      S=26.*VISC(TS,XS)/((TAU*RHOX(TS,XS))**0.5)
      IF(S.GE.R.OR.REY.LT.10000.) GO TO 600
      DS=S/10.
      DR=(R-S)/39.
      PRN=(CPMF(XB1,TB1)*VISIN/THKF(XB1,TB1)+CPMF(XBL,TBL)*VISOT/THKF
     1     (XBL,TBL))/2.
      SCN=(VISIN/(RHOIN*DIFF(TB1,Z))+VISOT/(RHOUT*DIFF(TBL,Z)))/2.
      WRITE            (6, 521) ITER,EFU,TAU,S,DR,DS,TBA,UB,VISMB
      WRITE            (6, 541) REY,PRN,SCN
C     SUBROUTINE FOR VELOCITY AND EDDY
      DO 24 I=1,L
      ROST=RHOX(TS,XB(I))
      ROBK=RHOX(TB(I),XB(I))
      IF(ITER.GT.1) UB=VB(I)
      VISMB=VISC(TB(I),XB(I))
      EFU=0.046/((2.*R*UB*ROBK/VISMB)**0.2)
      TAU   =EFU*UB*UB*ROST/2.
      S=26.*VISC(TS,XB(I))/((TAU*ROST)**0.5)
      DS=S/10.
      DR=(R-S)/39.
      DRM=2.*DR*DR
C ... LAMINAR LAYER CALCULATIONS
C ... DEISSLERS RELATION, EDDY=N*N*U*Y, UP TO Y+ =26
C ... NEWTON-RAPSON ITERATION APPROACH AND HASTINGS APPROXIMATION METHOD
      Y(1)=0.
      UT(1)=0.
10    EDT(1)=0.
      S1=0.
      U1=0.
      DO 14 K=2,11
      A=0.011881*ROBK/(2.*TAU)
      B=VISMB/TAU
      G=SQRT(TAU/ROBK)
      S2=S1+DS
      U2=G*(-3.05+5.*ALOG(G*ROBK *S2/VISMB))
```

```
      SQRTA=SQRT (A)
      E1=1./(1.+P*U1*SQRTA)
      F1=E1*(A1+E1*(A2+E1*A3))
   11 E2=1./(1.+P*U2*SQRTA)
      F2=E2*(A1+E2*(A2+E2*A3))
      DEN=2.*A*U2*S2+B
      DU=(S2+B*F2/SQRTA-(S1+B*F1/SQRTA)*EXP (A*(U2*U2-U1*U1)))/DEN
      U2=U2+DU
      IF (ABS (DU)-1.E-4) 13,13,11
   13 U1=U2
      S1=S2
      UT(K)=U2
      Y(K)=Y(K-1)+DS
14    EDT(K)=0.011881*UT(K)*(K-1)*DS
C
C     TURBULENT CORE CALCULATIONS: VON KARMAN RELATION
      CNST=DS/(UT(11)-UT(10))
      R2=R-S
      ROA=ROBK
      CNST=CNST+0.72*SQRT (R*R2*ROA/TAU)
      DO 19 J=12,50
      B1=0.36*SQRT (R*ROA/TAU)
      R1=R2-DR
      IF(R1.LT.0.0) R1=0.
      SQR1=SQRT (R1)
      SQR2=SQRT (R2)
      AL=(CNST-2.*B1*SQR1)/(CNST-2.*B1*SQR2)
      DL=(SQR1-SQR2)/B1
      EL=CNST*ALOG(AL)/(2.*B1*B1)
      UT(J)=UT(J-1)+EL+DL
      R2=R1
   19 CONTINUE
      DO 20 J=12,50
      Y(J)=Y(J-1)+DR
      IF(J-46) 22,21,21
21    EDT(J)=EDT(J-1)
```

```
      GO TO 20
   22 EDT(J)= TAU*(50-J)*DRM/(R*ROBK          *(UT(J+1)-UT(J-1)))
   20 CONTINUE
      IF(I.EQ.L) WRITE(6,522) I,(Y(J),UT(J),EDT(J),J=1,50)
      IF(I.NE.L) GO TO 24
      DO 3 J=1,M
      PP(J)=(R-Y(J))/R
      II=J+M
      PP(II)=UT(J)/UT(M)
      III=II+M
3     PP(III)=EDT(J)/EDT(M)
      CALL PLOT (200,PP,M,3,0,1)
      DO 30 J=1,M
       IF(Y(J).GE.XDR) GO TO 31
30     CONTINUE
31    CALL UEFIT( MINO(M,M-J+3),UT,EDT,Y)
      DZCENT=DZ/100.
      G=FO/AR
      GCP=G*CPMF(XS,TBA)
      NAMELIST/HEAT/ FO,AR,G,GCP
      WRITE(6,HEAT)
      DO 4 J=1,M
      XB(J)=KONF(EDT(J),TB(J),XS)
      VB(J)=DZCENT*GCP/XB(J)
      CB(J)=R*R*GCP/XB(J)/DZ
      PP(J)=   RR(J)/R
      Y (J)=R-RR(J)
      II=J+M
      PP(II)=UT(J)/UT(1)
      III=II+M
4     PP(III)=EDT(J)/EDT(1)
      Y(M)=0.
      WRITE(6,524) I,(Y(J),UT(J),EDT(J),XB(J),VB(J),CB(J),J=1,M)
524   FORMAT('1FITTED RESULTS, I=',I3,T40,'Y,U,EDDY,K,PE-Z,PE-R'''/
     -  (6(6X,G14.5)))
      CALL PLOT(300,PP,M,3,0,1)
```

```
24    CONTINUE
33     RETURN
  521 FORMAT (32H1           RESULTS OF ITERATION I5/43H            FANNIN
     1GS FRICTION FACTOR, EFU = E14.5/31H            SHEAR STRESS, TAU =
     2E26.5/31H            FILM THICKNESS, S = E26.5,4H FT./37H
     3 TURBULENT INCREMENT, DR = E20.5,4H FT./35H            LAMINAR INCR
     4EMENT, DS = E22.5,4H FT./32H            AVERAGE TEMPERATURE =F18.2,
     52H K/26H            MEAN VELOCITY =F24.2,8H FT/SEC./27H            M
     6EAN VISCOSITY = E30.5,13H GMOL/FT.SEC.//)
522   FORMAT('0VELOCITY & EDDY PROFILES, I=',I3/(3(6X,E14.5)))
  541 FORMAT (/5X,17HREYNOLDS NUMBER =E13.5,7X,16HPRANDTL NUMBER =E13.5,
     1        7X,16HSCHMIDT NUMBER =E13.5/)
600   WRITE(6,601) S,R
601   FORMAT('0*****LAMINAR FLOW, FILM THICKNESS=',G10.3,
     - ' (RADIUS=',G10.3,')')
      RETURN 1
      END
```

```
      SUBROUTINE UEFIT(I,V,E,Y)
      DIMENSION V(1),E(1),Y(1)
      COMMON/DIM/ R,AR,DZ,TWODZ,L,M,N,LP1,LM1,M1,N1,RR(50),DR
      POLY(A,B,C,D,E,F,X)=A+(B+(C+(D+(E+F*X)*X)*X)*X)*X
      I1=M-I
      DO 2 J=1,I
      V(J)=V(J+I1)*Y(J+I1)
      E(J)=E(J+I1)
2     Y(J)=Y(J+I1)
      I5=I-5
      I1=I-1
      DD=E(I-6)-E(I5)
      DO 5 J=I5,I1
      DD=DD*0.77
5     E(J)=E(J)+DD
      CALL CVFIT(5,I,Y,V)
      CALL CVFIT(5,I,Y,E)
      DO 3 J=1,6
      Y(J)=V(J)
3     Y(J+6)=E(J)
      DO 4 J=1,M1
      RX=R-RR(J)
      V(J)=POLY(Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),RX)/RX
4      E(J)=POLY(Y(7),Y(8),Y(9),Y(10),Y(11),Y(12),RX)
      V(M)=0.
      E(M)=0.
      RETURN
      END
```

```
      REAL FUNCTION FA(X)
      COMMON/G/ ALPHA,Q,FO,RO,TO,TS,DFS,CO,CON
      COMMON/ENERGY/ G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     - G16,THKA,THKB,THKC,THKD,PRN,SCN,REY,EA,AC,ORDER,INBC,IZBND,IRBND
      ENTRY RHOF(T,X)
      RHOF=RO*TO/(T*(1.-X))
      FA=RHOF
      RETURN
      ENTRY COMP1F(X)
      COMP1F=   ALPHA*(1.-X)
      FA=COMP1F
      RETURN
      ENTRY COMP2F(X)
      COMP2F   =ALPHA*X
      FA=COMP2F
      RETURN
      ENTRY CAYF(X)
      CAYF   =AC*2.7182818D0**(-EA/(1.987*X))
      FA=CAYF
      RETURN
      ENTRY CPMF(X,Y)
      CPMF     =((1.-ALPHA)*(G14+(G15+G16*Y)*Y)+ALPHA*((1.-X)*(G5+(G6+G7
     1          *Y)*Y)+X*(G8+G11+(G9+G12+(G10+G13)*Y)*Y)))/(1.+X*ALPHA)
      FA=CPMF
      RETURN
      ENTRY CPTF(X,Y)
      CPTF     =((1.-ALPHA)*(G14+(G15/2.+G16*Y/3.)*Y)+ALPHA*((1.-X)*(G5
     1          +(G6/2.+G7*Y/3.)*Y)+X*(G8+G11+((G9+G12)/2.+(G10+G13)*Y/
     2          3.)*Y)))/(1.+X*ALPHA)
      FA=CPTF
      RETURN
      ENTRY THKF(X,Y)
      THKF     =( (1.-ALPHA)*THKD+ALPHA*((1.-X)*THKA+X*(THKB+THKC)) )*
     1           (Y/TO)**0.4/(1.+ALPHA*X)
      FA=THKF
```

```
RETURN
ENTRY DIFF(X,Y)
DIFF     =DFS*(X/TO)**1.5+Y/SCN
FA=DIFF
RETURN
ENTRY DHRF(X)
DHRF   =G1+(G2+(G3+G4*X)*X)*X
FA=DHRF
RETURN
END
```

```
REAL FUNCTION FB(X)
COMMON/G/ ALPHA,Q,FO,RO,TO,TS,DFS,CO,CON
COMMON/ENERGY/ DUMMY(20),PRN,SCN,REY,EA,AC,ORDER,INBC,IZBND,IRBND
REAL KONF
ENTRY KONF(X,Y,Z)
KONF   =THKF(Z,Y)/CPMF(Z,Y)+X*RHOF(Y,Z)/PRN
FB=KONF
RETURN
ENTRY CDCF(T,C)
CDCF     =CAYF(T)*DHRF(T)*C**ORDER
FB=CDCF
RETURN
ENTRY CACF(T,C)
CACF     =CAYF(T)*C**(ORDER-1.)
FB=CACF
RETURN
END
```

```
      FUNCTION VISC(T,X)
C     VISCOSITY SUBPROGRAM
      COMMON/G/ ALPHA,Q,FO,RO,TO,TS,DFS,CO,CON
      REAL W(4)/135.,64.07,70.91,1./,EN(4)/2*.68,2*1./,Y(4)/4*0./
      REAL UO(4)/2.835E-5,7.758E-5,7.697E-5,1./
      VISF(X,Y,Z)=X*(Y/TO)**Z
      VISC=0.
      Y(1)=ALPHA*(1.-X)
      Y(2)=ALPHA*X
      Y(3)=Y(2)
      DO 6 IA=1,3
      SPHI=0.
      VISIA=VISF(UO(IA),T,EN(IA))
      DO 5 JA=1,3
      VISJA=VISF(UO(JA),T,EN(JA))
      HELP=W(IA)/W(JA)
      PHI=SQRT(1.+HELP)
      PHI=((1.+SQRT(VISIA/VISJA)*((1./HELP)**.25))**2)/PHI
    5 SPHI=Y(JA)*PHI+SPHI
    6 VISC=VISIA*Y(IA)/SPHI+VISC
      VISC=2.828428*VISC
      RETURN
      END
```

```
      SUBROUTINE CVFIT (MXORD,ND,XX,YY)
      DIMENSION A(42),B(42),AA(6,7),BB(6,7),XX(ND),YY(ND)
      EQUIVALENCE (A(1),AA(1,1)),(B(1),BB(1,1))
      DO 1 I=1,42
      A(I)=0.
    1 B(I)=0.
      A(1)=ND
      DO 10 I=1,ND
      X=XX(I)
      Y=YY(I)
      A(37)=A(37)+Y
      W=1.0
      Z=X**5
      DO 10 J=2,6
      W=W*X
      Z=Z*X
      A(J)=A(J)+W
      AA(J,6)=AA(J,6)+Z
10    AA(J,7)=AA(J,7)+Y*W
      DO 31 I=1,5
      DO 31 J=1,I
      K=5*J+I+1
   31 A(K)=A(I+1)
      DO 32 I=1,4
      DO 32 J=1,I
      L=36-I
      K=L-5*J
   32 A(K)=A(L)
      DO 54 K=1,5
      DO 33 I=K,6
      DO 33 J=K,6
   33 AA(I,J+1)=AA(I,J+1)/AA(I,K)
      DO 34 I=K,5
      DO 34 J=K,7
   34 AA(I+1,J)=AA(I+1,J)-AA(K,J)
```

```
54 CONTINUE
   A(42)=A(42)/A(36)
   DO 36 I=1,5
   L=37-I
   DO 35 J=1,6
   M=43-6*J-I
   K=43-J
35 B(M)=A(K)-B(L)*A(K-6)-B(L-6)*A(K-12)-B(L-12)*A(K-18)-B(L-18)*A(K-2
  14)-B(L-24)*A(K-30)
   II=43-I
36 A(II)=0.
39 DO 43 I=1,6
43 YY(I)=BB(MXORD+1,I)
   RETURN
   END
```

APPENDIX H

Derivation of CSCSDT Expression in the Radial Direction

When implicit in the radial direction, that is to pass from time $t^{i+1/2}$ to t^{i+1}, we can rewrite equation 4.13 in terms of a radial direction and applying the CSDT approximation yields the following ordinary differential equation

$$\frac{d^2f^{i+1}}{dr^2} + \frac{1}{r}\frac{df^{i+1}}{dr} - Pe_r(\beta + \frac{2\delta}{\Delta t})\ f^{i+1}$$
$$= -Pe_r\left[\frac{1}{Pe_z}\frac{d^2f^{i+1/2}}{dz^2} - \frac{df^{i+1/2}}{dz} + \frac{2\delta f^{i+1/2}}{\Delta t}\right]$$

or

$$\frac{d^2f^{i+1}}{dr^2} + \frac{1}{r}\frac{df^{i+1}}{dr} - Pe_r(\beta + \frac{2\delta}{\Delta t})\ f^{i+1} = R^{i+1/2} \qquad \text{(H-1)}$$

where

$$R^{i+1/2} = -Pe_r\left[\frac{1}{Pe_z}\frac{d^2f^{i+1/2}}{dz^2} - \frac{df^{i+1/2}}{dz} + \frac{2\delta f^{i+1/2}}{\Delta t}\right] \qquad \text{(H-2)}$$

A relationship between (H-2) and the expression in time t^i to $t^{i+1/2}$ leads to the following form

$$R^{i+1/2} = -\frac{Pe_r}{Pe_z}R^i - Pe_r(\beta + \frac{4\delta}{\Delta t})\ f^{i+1/2} \qquad \text{(H-3)}$$

Obviously, the right hand side of (H-3) is known from the results of the first half time step. Now to apply the decomposition method to equation (H-1), first we multiply every term by r^2:

$$r^2\frac{d^2f^{i+1}}{dr^2} + r\frac{df^{i+1}}{dr} - r^2Pe_r(\beta + \frac{2\delta}{\Delta t})\ f^{i+1} = r^2R^{i+1/2} \qquad \text{(H-4)}$$

We assume that the second order differential operator

$$L(\ ) = r^2 \frac{d^2}{dr^2} + r \frac{d}{dr} - r^2 Pe_r (\beta + \frac{2\delta}{\Delta t}) \tag{H-5}$$

can be decomposed into two first order differential operators of the form:

$$\begin{aligned} L(\ ) &= L_B \cdot L_F \\ &= \left[r \frac{d}{dr} - \lambda_B(r) \right] \cdot \left[r \frac{d}{dr} - \lambda_F(r) \right] \\ &= r^2 \frac{d^2}{dr^2} + r(1 - \lambda_F - \lambda_B) \frac{d}{dr} - r \frac{d\lambda_F}{dr} + \lambda_F \lambda_B \end{aligned} \tag{H-6}$$

By identification of (H-5) and (H-6), we have:

$$1 - \lambda_F - \lambda_B = 1 \tag{H-7}$$

and

$$r \frac{d\lambda_F}{dr} - \lambda_F \lambda_B = r^2 Pe_r (\beta + \frac{2\delta}{\Delta t}) \tag{H-8}$$

which becomes

$$\lambda_B(r) = -\lambda_F(r) \tag{H-9}$$

$$r \frac{d\lambda_F}{dr} + \lambda_F^2 = r^2 Pe_r (\beta + \frac{2\delta}{\Delta t}) \tag{H-10}$$

Since the general solution f is expressed as

$$f = af_1 + bf_2 + f_3$$

where f_1 and f_2 are being the solutions of the two homogeneous equations:

$$L_B(f_1) = 0; \; f_1(1) = 1$$

$$L_F(f_2) = 0; \; f_2(0) = 1$$

The two boundary conditions stated that

$$\frac{df}{dr} = 0 \qquad \text{for} \quad r = 0 \quad \text{and} \quad r = 1$$

but

$$\frac{df}{dr} = a\frac{df_1}{dr} + b\frac{df_2}{dr} + \frac{df_3}{dr}$$

$$= \frac{1}{r}\left[a\lambda_B f_1 + b\lambda_F f_2 + \lambda_F f_3\right]$$

It could be easily seen that even if $\lambda_F(1) = 0$ and since we know $\lambda_F(r) \leq 0$ and $f_3(0) = 0$

$\frac{df}{dr} = 0$ at $r = 1$ can be satisfied $(a = 0)$, yet at $r = 0$

$$\frac{df}{dr} = \frac{\lambda_F}{r}(bf_2 + f_3) = \frac{\lambda_F}{r} b \neq 0.$$

Vichnevetsky (25) has successfully demonstrated the application of the decomposition method using an example problem describing non-steady state burning of a solid propellant. The final transformed system equation yields the differential operator

$$L(\;) = x^2\frac{d^2}{dx^2} + x\beta\frac{d}{dx} - \frac{1}{\alpha\theta\Delta t}$$

for the dependent variable T (temperature).

This is then decomposed into two first order differential operators of the form:

$$L(\ ) = L_1(\ ) \cdot L_2(\ )$$

$$= \left(x \frac{d}{dx} - \lambda_1 \right) \cdot \left(x \frac{d}{dx} - \lambda_2 \right)$$

with the boundary conditions

$$T(0,t) = 0$$

$$\frac{\partial T}{\partial x}(x,t)\Big|_{x=1^-} = f(t) \quad \text{(assumed to be given)}$$

for $x \in (0,1)$

We now define $y(x)$, $\tau_1(x)$ and $\tau_3(x)$ as being the solution of the three following equations and respective boundary conditions:

$$L_1(y) = -(\text{forcing function from the previous time step})$$

$$y(1) = 0$$

$$L_2(\tau_3) = y$$

$$\tau_3(0) = 0$$

$$L_1(\tau_1) = 0$$

$$\tau_1(1) = 1$$

We note that $\tau_1(0) = 0$ will always be satisfied, in view of the

$\frac{\tau_1}{x}$ term present in the following equation, going to ∞ for x going to zero

$$\frac{d\tau_1}{dx} = +\frac{\lambda_1}{x}\tau_1$$

We now observe that any linear combination of the form

$$T(x) = \tau_3(x) + a\tau_1(x)$$

satisfies the system equation and that $T(0) = 0$ is always satisfied.

The boundary condition at $x = 1$ is then used to find the value of a.