6-1970

A Numerical Algorithm For Solving the Nonlinear Differential Equations that Describe a Multistage Flash Evaporator

Maurice Manning Anderson Jr.
University of Tennessee, Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Maurice Manning Anderson Jr. entitled "A Numerical Algorithm For Solving the Nonlinear Differential Equations that Describe a Multistage Flash Evaporator." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

J.C. Robinson, Major Professor

We have read this thesis and recommend its acceptance:

T.W. Kerlin, Hale C. Roland

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

May 8, 1970

To the Graduate Council:

I am submitting herewith a thesis written by Maurice Manning Anderson, Jr. entitled "A Numerical Algorithm For Solving the Nonlinear Differential Equations that Describe a Multistage Flash Evaporator." I recommend that it be accepted for nine quarter hours of credit in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

J C Robinson

Major Professor

We have read this thesis
and recommend its acceptance

Julian T Mott

T. W. Kerlin

Hall C. Roland

Accepted for the Council:

Vice Chancellor for
Graduate Studies and Research

A NUMERICAL ALGORITHM FOR SOLVING THE NONLINEAR DIFFERENTIAL EQUATIONS THAT DESCRIBE A MULTISTAGE FLASH EVAPORATOR

A Thesis

Presented to

the Graduate Council of

The University of Tennessee

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Maurice Manning Anderson, Jr.

June 1970

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Dr. J. C. Robinson for his guidance and instruction during the course of this study. The author would also like to thank Dr. T. W. Kerlin, Dr. J. H. Swanks, and Mr. S. J. Ball for their assistance.

Finally, the financial support provided by a National Defense Education Act (Title IV) Fellowship and by the Instrumentation and Controls Division of Oak Ridge National Laboratory is gratefully acknowledged.

ABSTRACT

A numerical algorithm is formulated to solve the first order, nonlinear differential equations that describe a multistage flash evaporator. The nonlinearities appearing in the formulation of the algorithm are products of up to three terms with each term being a dependent variable raised to some power.

To develop the algorithm, the first order differential equations are written in integral form. The dependent variables are then assumed to have a purely exponential dependence over a finite time step thereby allowing for the explicit integration of all terms. The solution of the differential equations is then reduced to the determination of the exponential dependences. The exponential dependences are determined by an iterative method.

A computer code based upon the aforementioned algorithm was written. Before the algorithm was used to obtain solutions to a flash evaporator system, it was applied to several differential equations with known solutions. The algorithm was then used to obtain solutions for two perturbations in the twenty-third order system that describes a three stage flash evaporator. These solutions are compared with solutions obtained by other methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

I. PURPOSE OF THE STUDY

Although considerable effort has gone into the development and the study of linearized models of multistage flash evaporators,[1,2,3,4,5,6*] solutions of the nonlinear models are needed to aid in the interpretation of experimental data and in the design of control systems.[7] Currently numerical solutions of the nonlinear models of multistage flash evaporators are being obtained using MATEXP,[8] a general purpose computer program for solving differential equations.[6]

The purpose of this thesis is to develop a solution algorithm applicable to nonlinear differential equations and in particular to the nonlinear models of multistage flash evaporators. The algorithm will be used to check the use of MATEXP in the solution of the nonlinear models of flash evaporators by presenting an alternate and independent solution algorithm. If the computation time permits, the solution algorithm may become a useful tool for solving the large sets of nonlinear differential equations that describe multistage flash evaporator systems.

---

[*] Superscript numbers in the text refer to similarly numbered entries in the bibliography.

II. PREVIOUS USE OF THE EXPONENTIAL ALGORITHM AND ITS EXTENSION TO NONLINEAR DIFFERENTIAL EQUATIONS

Hansen et al.[9,10] have developed a computational algorithm for solving the time-dependent neutron multi-group diffusion equations that is numerically stable, rapid in operation, and accurate. In essence, Hansen's algorithm was developed by integrating the differential equations describing the time-dependent neutron fluxes and precursor concentrations over a finite time step and assuming an exponential time dependence of the fluxes and precursor concentrations. In this thesis numerical algorithms for solving differential equations by assuming an exponential dependence of the dependent variables over a finite time step are referred to as exponential algorithms.

Others have since applied modifications of Hansen's exponential algorithm to other problems. Specifically, Swanks[11] used an exponential algorithm to obtain solutions to the time-dependent discrete ordinate neutron transport equations, and Stevenson and Bingham[12] used an exponential algorithm for a liquid metal fast breeder transient analysis.

Because of its speed and accuracy in the solution of large sets of linear differential equations, an extension of the exponential algorithm to large sets of nonlinear differential equations seems appropriate. However, the question of numerical stability remains unanswered when applying the exponential algorithm to nonlinear differential equations.

To develop the exponential algorithm for first order, nonlinear differential equations, the dependent variables are assumed to behave as pure exponentials over a finite time step. The differential equations are then integrated over the finite time step, and the solution of the differential equations is reduced to the determination of the exponential dependences of the variables.

CHAPTER II

FLASH EVAPORATORS

I. A DESCRIPTION OF A THREE STAGE FLASH EVAPORATOR AND THE VARIABLES USED TO DESCRIBE A MULTISTAGE FLASH EVAPORATOR

In the model employed in this thesis,[6] the differential equations describing the state of any interior stage of a multistage flash evaporator are a function of the variables used to describe that stage and the variables used to describe the two adjacent stages. For this reason, a three stage evaporator system is sufficiently general to include the coupling that arises in multistage flash evaporator systems and is used as a reference system in this study.

A schematic diagram of the three stage flash evaporator used as the reference system is shown in Figure 1. The brine is heated in the steam heater section and pumped into Stage 1 where partial flashing occurs as the brine enters the stage. Pressure differences due to the vapor pressures and the hydraulic heads support the brine flow from Stage 1 to Stage 2 and from Stage 2 to Stage 3, and partial flashing occurs as the brine enters the stages. Fresh water is used in the coolant loop to condense the vapor, and the condensate is removed from the system.

As shown in Figure 1, six dependent variables are used to describe the state of each stage. Specifically, these variables are

1. The tray brine mass,

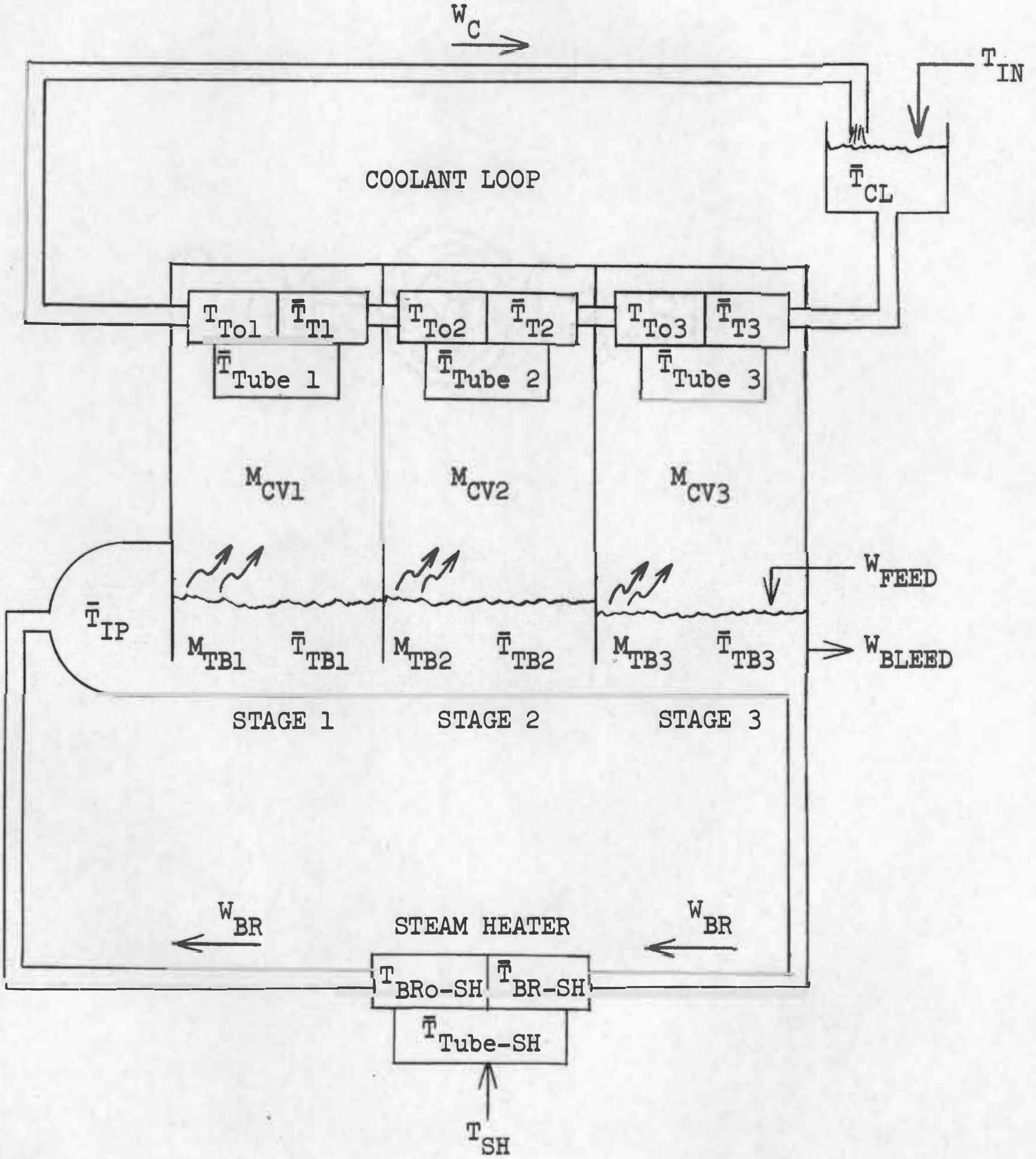

FIGURE 1

SCHEMATIC DIAGRAM OF THREE STAGE EVAPORATOR SYSTEM

2. The average tray brine temperature,
3. The cell vapor mass,
4. The average condenser tube temperature,
5. The average coolant temperature, and
6. The coolant outlet temperature.

In addition, five dependent variables are used to describe the heating and cooling loops.

A state variable vector representation, $\bar{X}$, is used to describe the state of the entire system. Table 1 defines the variables used to describe the three stage evaporator system and denotes the ordering of the variables in the state vector.

II. THE DIFFERENTIAL EQUATIONS USED TO DESCRIBE A THREE STAGE FLASH EVAPORATOR

The development of the differential equations used to describe a multistage flash evaporator is well documented.[1,2,6] Therefore, no attempt will be made here to present their development. The nonlinear differential equations used to describe a three stage flash evaporator were obtained from the equations given in the Appendix of Reference 6, except for the introduction of the nonlinear flashing flowrate discussed in Reference 6.

The nonlinear differential equations used to describe a three stage flash evaporator are

TABLE 1

DEFINITION OF THE VARIABLES USED IN THE THREE STAGE EVAPORATOR MODEL

Model Variable (units)	State Variable	Physical Significance
T_{To1}(°F)	x_1	Coolant outlet temperature in Stage 1
$\bar{T}_{T1}$(°F)	x_2	Average coolant temperature in Stage 1
$\bar{T}_{Tube\ 1}$(°F)	x_3	Average condenser tube temperature in Stage 1
M_{CV1}(pounds)	x_4	Cell vapor mass in Stage 1
$\bar{T}_{TB1}$(°F)	x_5	Average tray brine temperature in Stage 1
M_{TB1}(pounds)	x_6	Tray brine mass in Stage 1
T_{To2}(°F)	x_7	Coolant outlet temperature in Stage 2
$\bar{T}_{T2}$(°F)	x_8	Average coolant temperature in Stage 2
$\bar{T}_{Tube\ 2}$(°F)	x_9	Average condenser tube temperature in Stage 2
M_{CV2}(pounds)	x_{10}	Cell vapor mass in Stage 2
$\bar{T}_{TB2}$(°F)	x_{11}	Average tray brine temperature in Stage 2
M_{TB2}(pounds)	x_{12}	Tray brine mass in Stage 2
T_{To3}(°F)	x_{13}	Coolant outlet temperature in Stage 3
$\bar{T}_{T3}$(°F)	x_{14}	Average coolant temperature in Stage 3
$\bar{T}_{Tube\ 3}$(°F)	x_{15}	Average condenser tube temperature in Stage 3
M_{CV3}(pounds)	x_{16}	Cell vapor mass in Stage 3

TABLE 1 (continued)

Model Variable (units)	State Variable	Physical Significance
$\bar{T}_{TB3}$(°F)	x_{17}	Average tray brine temperature in Stage 3
M_{TB3}(pounds)	x_{18}	Tray brine mass in Stage 3
$\bar{T}_{CL}$(°F)	x_{19}	Average coolant temperature in reservoir
$\bar{T}_{IP}$(°F)	x_{20}	Average inlet plenum temperature
$\bar{T}_{BR\text{-}SH}$(°F)	x_{21}	Average brine temperature in brine-heater
$T_{BRo\text{-}SH}$(°F)	x_{22}	Brine outlet temperature in brine-heater
$\bar{T}_{Tube\text{-}SH}$(°F)	x_{23}	Average tube temperature in brine-heater
W_{BR}(pounds/second)		Brine flowrate
W_{C}(pounds/second)		Coolant flowrate
T_{IN}(°F)		Temperature of coolant feed
T_{SH}(°F)		Temperature of steam in brine heater
W_{FEED}(pounds/second)		Brine feed rate
W_{BLEED}(pounds/second)		Brine extraction rate

$$\frac{dx_1}{dt} = -\left(\frac{2W_C}{M_{T1}}\right)x_1 + \left[\frac{2W_C}{M_{T1}} - \left(\frac{h_iA_i}{(MC_p)_T}\right)_1\right]x_2 + \left(\frac{h_iA_i}{(MC_p)_T}\right)_1 x_3 \qquad \text{(1a)}$$

$$\frac{dx_2}{dt} = -\left[\frac{2W_C}{M_{T1}} + \left(\frac{h_iA_i}{(MC_p)_T}\right)_1\right]x_2 + \left(\frac{h_iA_i}{(MC_p)_T}\right)_1 x_3 + \left(\frac{2W_C}{M_{T1}}\right)x_7 \qquad \text{(1b)}$$

$$\frac{dx_3}{dt} = \left(\frac{h_iA_i}{(MC_p)_{Tube}}\right)_1 x_2 - \left[\left(\frac{h_iA_i}{(MC_p)_{Tube}}\right)_1 + \left(\frac{h_oA_o}{(MC_p)_{Tube}}\right)_1\right]x_3$$

$$+ \alpha_1\left(\frac{h_oA_o}{(MC_p)_{Tube}}\right)_1 x_4 + \beta_1\left(\frac{h_oA_o}{(MC_p)_{Tube}}\right)_1 \qquad \text{(1c)}$$

$$\frac{dx_4}{dt} = \left(\frac{h_oA_o}{h_{fg}}\right)_1 x_3 - \alpha_1\left[K_1(W_{BR})^{1/2} + \left(\frac{h_oA_o}{h_{fg}}\right)_1\right]x_4$$

$$+ K_1(W_{BR})^{1/2}\, x_{20} - \beta_1\, K_1(W_{BR})^{1/2} + \left(\frac{h_oA_o}{h_{fg}}\right)_1 \qquad \text{(1d)}$$

$$\frac{dx_5}{dt} = \alpha_1 K_1(W_{BR})^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_1 \frac{x_4}{x_6} - (W_{BR})\,\frac{x_5}{x_6}$$

$$+\left[W_{BR} - K_1(W_{BR})^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_1\right]\frac{x_{20}}{x_6} + K_1(W_{BR})^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_1 \frac{1}{x_6} \qquad \text{(1e)}$$

$$\frac{dx_6}{dt} = \alpha_1 K_1 (W_{BR})^{1/2} \; x_4 - K_1 (W_{BR})^{1/2} \; x_{20}$$

$$- (XWl_1) x_{24}^{1/2} + \beta_1 K_1 (W_{BR})^{1/2} + W_{BR} \tag{1f}$$

$$\frac{dx_7}{dt} = - \left(\frac{2W_C}{M_{T2}} \right) x_7 + \left[\frac{2W_C}{M_{C3}} - \left(\frac{h_i A_i}{(MC_p)_T} \right)_2 \right] x_8 + \left(\frac{h_i A_i}{(MC_p)_T} \right)_2 x_9 \tag{1g}$$

$$\frac{dx_8}{dt} = - \left[\frac{2W_C}{M_{T2}} + \left(\frac{h_i A_i}{(MC_p)_T} \right)_2 \right] x_8 + \left(\frac{h_i A_i}{(MC_p)_T} \right)_2 x_9 + \left(\frac{2W_C}{M_{T2}} \right) x_{13} \tag{1h}$$

$$\frac{dx_9}{dt} = \left(\frac{h_i A_i}{(MC_p)_{Tube}} \right)_2 x_8 - \left[\left(\frac{h_i A_i}{(MC_p)_{Tube}} \right)_2 + \left(\frac{h_o A_o}{(MC_p)_{Tube}} \right)_2 \right] x_9$$

$$+ \alpha_2 \left(\frac{h_o A_o}{(MC_p)_{Tube}} \right)_2 x_{10} + \beta_2 \left(\frac{h_o A_o}{(MC_p)_{Tube}} \right)_2 \tag{1i}$$

$$\frac{dx_{10}}{dt} = \left(\frac{h_o A_o}{h_{fg}} \right)_2 x_9 - \alpha_2 \left(\frac{h_o A_o}{h_{fg}} \right)_2 x_{10}$$

$$- (\beta_2 K_2 XWl_1) x_{24}^{1/2} + (K_2 XWl_1) x_{24}^{1/2} \; x_5$$

$$- (\alpha_2 K_2 XWl_1) x_{24}^{1/2} \; x_{10} - \beta_2 \left(\frac{h_o A_o}{h_{fg}} \right)_2 \tag{1j}$$

$$\frac{dx_{11}}{dt} = (XW1_1)\frac{x_{24}^{1/2}x_5}{x_{12}} - K_2(XW1_1)^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_2 \frac{x_{24}^{1/4}x_5}{x_{12}}$$

$$+ \alpha_2 K_2(XW1_1)^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_2 \frac{x_{24}^{1/2}x_{10}}{x_{12}}$$

$$- (XW1_1)\frac{x_{24}^{1/2}x_{11}}{x_{12}} + \beta_2 K_2(XW1_1)^{1/2}\left(\frac{h_{fg}}{C_{pTB}}\right)_2 \frac{x_{24}^{1/4}}{x_{12}} \qquad (1k)$$

$$\frac{dx_{12}}{dt} = - K_2(XW1_1)^{1/2} \; x_{24}^{1/4}x_5 + \alpha_2 K_2(XW1_1)^{1/2} \; x_{24}^{1/4}x_{10}$$

$$+ \beta_2 K_2(XW1_1)^{1/2} \; x_{24}^{1/4} + (XW1_1)x_{24}^{1/2} - (XW1_2)x_{25}^{1/2} \qquad (1l)$$

$$\frac{dx_{13}}{dt} = -\left(\frac{2W_C}{M_{T3}}\right)x_{13} + \left[\frac{2W_C}{M_{T3}} - \left(\frac{h_i A_i}{(MC_p)_T}\right)_3\right]x_{14} + \left(\frac{h_i A_i}{(MC_p)_T}\right)_3 x_{15} \qquad (1m)$$

$$\frac{dx_{14}}{dt} = -\left[\frac{2W_C}{M_{T3}} + \left(\frac{h_i A_i}{(MC_p)_T}\right)_3\right]x_{14} + \left(\frac{h_i A_i}{(MC_p)_T}\right)_3 x_{15} + \left(\frac{2W_C}{M_{T3}}\right)x_{19} \qquad (1n)$$

$$\frac{dx_{15}}{dt} = \left(\frac{h_i A_i}{(MC_p)_{Tube}}\right)_3 x_{14} - \left[\left(\frac{h_i A_i}{(MC_p)_{Tube}}\right)_3 + \left(\frac{h_i A_o}{(MC_p)_{Tube}}\right)_3\right] x_{15}$$

$$+ \alpha_3 \left(\frac{h_o A_o}{(MC_p)_{Tube}}\right)_3 x_{16} + \beta_3 \left(\frac{h_o A_o}{(MC_p)_{Tube}}\right)_3 \qquad (1o)$$

$$\frac{dx_{16}}{dt} = \left(\frac{h_o A_o}{h_{fg}}\right)_3 x_{15} - \alpha_3 \left(\frac{h_o A_o}{h_{fg}}\right)_3 x_{16}$$

$$+ K_3 (XWl_2)^{1/2} x_{25}^{1/4} x_{11} - \alpha_3 K_3 (XWl_2)^{1/2} x_{25}^{1/4} x_{16}$$

$$- \beta_3 K_3 (XWl_2)^{1/2} x_{25}^{1/4} - \beta_3 \left(\frac{h_o A_o}{h_{fg}}\right)_3 \qquad (1p)$$

$$\frac{dx_{17}}{dt} = (XWl_2) \frac{x_{25}^{1/2} x_{11}}{x_{18}} - K_3 (XWl_2)^{1/2} \left(\frac{h_{fg}}{C_{pTB}}\right)_3 \frac{x_{25}^{1/4} x_{11}}{x_{18}}$$

$$+ \alpha_3 K_3 (XWl_2)^{1/2} \left(\frac{h_{fg}}{C_{pTB}}\right)_3 \frac{x_{25}^{1/4} x_{16}}{x_{18}}$$

$$- (XWl_2) \frac{x_{25}^{1/2} x_{17}}{x_{18}} + \beta_3 K_3 (XWl_2)^{1/2} \left(\frac{h_{fg}}{C_{pTB}}\right)_3 \frac{x_{25}^{1/4}}{x_{18}} \qquad (1q)$$

$$\frac{dx_{18}}{dt} = (XW1_2)x_{25}^{1/2} - K_3(XW1_2)^{1/2} x_{25}^{1/4} x_{11} + \alpha_3 K_3(XW1_2)^{1/2} x_{25}^{1/4} x_{16}$$

$$+ \beta_3 K_3(XW1_2)^{1/2} x_{25}^{1/4} - W_{BR} + W_{FEED} - W_{BLEED} \qquad (1r)$$

$$\frac{dx_{19}}{dt} = \left(\frac{W_C Recf}{M_{CL}}\right) x_1 - \left(\frac{W_C}{M_{CL}}\right) x_{19} + \left(\frac{W_C}{M_{CL}}\right)(1 - Recf)T_{IN} \qquad (1s)$$

$$\frac{dx_{20}}{dt} = - \left(\frac{W_{BR}}{M_{IP}}\right) x_{20} + \left(\frac{W_{BR}}{M_{IP}}\right) x_{22}(t - \tau_2) \qquad (1t)$$

$$\frac{dx_{21}}{dt} = \left(\frac{2W_{BR}}{M_{SH}}\right) x_{17}(t - \tau_1) - \left[\frac{2W_{BR}}{M_{SH}} + \left(\frac{h_i A_i}{(MC_p)_T}\right)_{SH}\right] x_{21}$$

$$+ \left(\frac{h_i A_i}{(MC_p)_T}\right)_{SH} x_{23} \qquad (1u)$$

$$\frac{dx_{22}}{dt} = \left[\frac{2W_{BR}}{M_{SH}} - \left(\frac{h_i A_i}{(MC_p)_T}\right)_{SH}\right] x_{21} - \left(\frac{2W_{BR}}{M_{SH}}\right) x_{22} + \left(\frac{h_i A_i}{(MC_p)_T}\right)_{SH} x_{23} \qquad (1v)$$

$$\frac{dx_{23}}{dt} = \left(\frac{h_i A_i}{(MC_p)_{Tube}}\right)_{SH} x_{21} - \left[\left(\frac{h_i A_i}{(MC_p)_{Tube}}\right)_{SH} + \left(\frac{h_o A_o}{(MC_p)_{Tube}}\right)_{SH}\right] x_{23}$$

$$+ \left(\frac{h_o A_o}{(MC_p)_{Tube}}\right)_{SH} T_{SH} \, . \qquad (1w)$$

The definitions of the coefficients used in Equation (1) are given in Appendix A.

In Equation (1), x_{24} and x_{25} are the effective pressure drops from Stage 1 to Stage 2 and from Stage 2 to Stage 3 respectively. These pressure drops are given by the following equations:

$$x_{24} = \gamma_1 x_4 + \left(\frac{1}{AFC_1}\right) x_6 - \gamma_2 x_{10} - \left(\frac{K2_1}{AFC_2}\right) x_{12} + \xi_1 - \xi_2$$

$$x_{25} = \gamma_2 x_{10} + \left(\frac{1}{AFC_2}\right) x_{12} - \gamma_3 x_{16} - \left(\frac{K2_2}{AFC_3}\right) x_{18} + \xi_2 - \xi_3 - XM3B. \tag{2}$$

The definitions of the coefficients used in Equation (2) are given in Appendix A.

In Equation (1) the nonlinear terms are products of the dependent variables and the effective pressure drops.* An exponential algorithm can be readily formulated for nonlinearities of this form.

---

*The effective pressure drops are formulated as dependent variables, and differential equations describing their time dependence can be obtained by differentiating Equation (2).

CHAPTER III

THE NUMERICAL ALGORITHM

I. DEVELOPMENT OF THE FINITE DIFFERENCED EQUATIONS

The first order, nonlinear differential equations for a three stage flash evaporator given in Equation (1) can be written in the form:

$$\frac{d\bar{X}}{dt} = C\bar{X} + \bar{F}(\bar{X}) + \bar{Z}(\bar{X}) + \bar{S} \qquad (3)$$

where $\bar{X}$ is a N dimensional, time dependent state vector; C is a linear, N dimensional square matrix; $\bar{F}(\bar{X})$ is a N dimensional column vector containing all the nonlinear terms; $\bar{Z}(\bar{X})$ is a N dimensional column vector containing all time-lagged terms and $\bar{S}$ is a N dimensional column vector. The C matrix in Equation (3) is factored into two parts:

$$C = A + D \qquad (4)$$

where D is strictly a diagonal matrix and A is the remaining part of the C matrix. Substitution of Equation (4) into Equation (3) and rearrangement of terms yields

$$\frac{d\bar{X}}{dt} - D\bar{X} = A\bar{X} + \bar{F}(\bar{X}) + \bar{Z}(\bar{X}) + \bar{S}. \qquad (5)$$

The algorithm used to obtain solutions to Equation (3) will be developed from the general concepts of Hansen's exponential algorithm[9,10]; however, the algorithm will be formulated by considering the differential equation describing an arbitrary dependent variable, x_i, instead of using a matrix representation. The differential equation describing this arbitrary variable is

$$\frac{dx_i}{dt} - d_{ii}x_i = \sum_{j=1}^{N} a_{ij}x_j + f_i(\bar{X}) + z_i(\bar{X}) + s_i \qquad (6)$$

where the definition of all terms can be inferred by comparison of Equations (5) and (6).

Equation (6) is multiplied by the integrating factor $\exp(-d_{ii}t)$, and the resulting equation is integrated from t_α to $t_\alpha+h$ yielding

$$x_i(t_\alpha+h)\exp(-d_{ii}(t_\alpha+h)) - x_i(t_\alpha)\exp(-d_{ii}t_\alpha)$$

$$= \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')[\sum_{j=1}^{N} a_{ij}x_j(t') + f_i(\bar{X}) + z_i(\bar{X}) + s_i]dt'. \qquad (7)$$

Multiplying Equation (7) by $\exp(+d_{ii}(t_\alpha+h))$ and rearranging yields

$$x_i(t_\alpha+h) = x_i(t_\alpha)\exp(d_{ii}h) + \exp(d_{ii}(t_\alpha+h)) \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')[\sum_{j=1}^{N} a_{ij}x_j(t')$$

$$+ f_i(\bar{X}) + z_i(\bar{X}) + s_i]dt'. \quad (8)$$

It should be noted at this point that $f_i(\bar{X})$ and $z_i(\bar{X})$ are functions of t'.

For the flash evaporator described by Equation (1), the nonlinear terms, $f_i(\bar{X})$'s, can in general be represented by

$$f_i(\bar{X}) = \sum_{j=1}^{J_i} b_j[x_k^p(t)x_\ell^q(t)x_m^r(t)]_j \quad (9)$$

where J_i is the number of nonlinearities in the i^{th} differential equation; the b_j's are coefficients of the individual nonlinear terms; the k's, ℓ's, and m's are integers corresponding to the indices of the state vector $\bar{X}$; and the p's, q's, and r's are real numbers; i.e., for each nonlinear term of the i^{th} differential equation there is a corresponding b, k, ℓ, m, p, q, and r.

The time-lagged terms are expressed in the form:

$$z_i(\bar{X}) = \sum_{j=1}^{T_i} e_j[x_n(t - \tau)]_j \quad (10)$$

where T_i is the number of time-lagged terms in the i^{th} differential equation; the e_j's are the coefficients of the individual time-lagged terms; the n's are integers corresponding to the indices of the dependent state vector $\bar{X}$; and the τ's are the time lags; i.e., for each time-lagged term in the i^{th} differential equation, there is a corresponding e, n, and τ.

Substitution of Equations (9) and (10) into Equation (8) yields

$$x_i(t_\alpha+h) = x_i(t_\alpha)\exp(d_{ii}h) + \exp(d_{ii}(t_\alpha+h)) \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')\Big(\sum_{j=1}^{N} a_{ij}x_j(t') + \sum_{j=1}^{J_i} b_j[x_k^p(t')x_\ell^q(t')x_m^r(t')]_j + \sum_{j=1}^{T_i} e_j[x_n(t'-\tau)]_j + s_i\Big)dt'. \tag{11}$$

At this point, the assumption that $x_n(t'-\tau)$, where $t_\alpha \leq t' \leq t_\alpha + h$, can be accurately approximated by $x_n(t_\alpha + \frac{h}{2} - \tau)$ is made. Using this assumption and the assumption that the a_{ij}'s, b_j's, e_j's, and s_i are constant over the time step t_α to $t_\alpha + h$, Equation (11) becomes

$$x_i(t_\alpha+h) = x_i(t_\alpha)\exp(d_{ii}h)$$

$$+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{N} a_{ij} \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')x_j(t')dt'$$

$$+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{J_i} b_j \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')[x_k^p(t')x_\ell^q(t')x_m^r(t')]_j dt'$$

$$+ \exp(d_{ii}(t_\alpha+h))\left(\sum_{j=1}^{T_i} e_j[x_n(t_\alpha+ \frac{h}{2} - \tau)]_j + s_i\right) \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')dt'. \qquad (12)$$

Integration of the last term in Equation (12) and rearrangement of the resulting equation yields

$$x_i(t_\alpha+h) = x_i(t_\alpha)\exp(d_{ii}h)$$

$$+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{N} a_{ij} \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')x_j(t')dt'$$

$$+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{J_i} b_j \int_{t_\alpha}^{t_\alpha+h} \exp(-d_{ii}t')[x_k^p(t')x_\ell^q(t')x_m^r(t')]_j dt'$$

$$+ \left(\sum_{j=1}^{T_i} e_j[x_n(t_\alpha+ \frac{h}{2} - \tau)]_j + s_i\right)(\exp(d_{ii}h) - 1)/d_{ii}. \qquad (13)$$

Some assumption must now be made concerning the behavior of the dependent variables over the time step t_α to $t_\alpha+h$. The dependent variables are assumed to have a purely exponential time dependence over the time step; i.e., on the interval $t_\alpha \le t' \le t_\alpha+h$, $x_i(t')$ is given by the expression:

$$x_i(t') = x_i(t_\alpha)\exp(\omega_i(t'-t_\alpha)) \tag{14}$$

where the ω's are a set of real parameters to be determined numerically.* A discussion of the method of determining the ω's is presented in the next section. Introduction of the exponential assumption into Equation (13) and rearrangement of the resulting equation yields

$$\begin{aligned} x_i(t_\alpha+h) &= x_i(t_\alpha)\exp(d_{ii}h) \\ &+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{N} a_{ij}x_j(t_\alpha)\exp(-\omega_j t_\alpha) \int_{t_\alpha}^{t_\alpha+h} \exp((\omega_j-d_{ii})t')dt' \\ &+ \exp(d_{ii}(t_\alpha+h)) \sum_{j=1}^{J_i} b_j[x_k^p(t_\alpha)x_\ell^q(t_\alpha)x_m^r(t_\alpha)\exp(-p\omega_k \\ &+ q\omega_\ell+r\omega_m)t_\alpha) \int_{t_\alpha}^{t_\alpha+h} \exp((p\omega_k+q\omega_\ell+r\omega_m-d_{ii})t')]_j dt' \\ &+ (\sum_{j=1}^{T_i} e_j[x_n(t_\alpha+\frac{h}{2}-\tau)]_j + s_i)(\exp(d_{ii}h)-1)/d_{ii}. \end{aligned} \tag{15}$$

---

*It is noted that the exponential assumption given by Equation (15) assumes that the values of the dependent variables do not change sign over a time step.

After evaluation of the integrals, Equation (15) is written as

$$x_i(t_\alpha+h) = x_i(t_\alpha)\exp(d_{ii}h)$$

$$+ \exp(d_{ii}h) \sum_{j=1}^{N} a_{ij}x_j(t_\alpha)(\exp((\omega_j-d_{ii})h) - 1)/(\omega_j - d_{ii})$$

$$+ \exp(d_{ii}h) \sum_{j=1}^{J_i} b_j[x_k^p(t_\alpha)x_\ell^q(t_\alpha)x_m^r(t_\alpha)(\exp((p\omega_k$$

$$+q\omega_\ell + r\omega_m - d_{ii})h) - 1)/(p\omega_k+q\omega_\ell+r\omega_m-d_{ii})]_j$$

$$+ \left(\sum_{j=1}^{T_i} e_j[x_n(t_\alpha + \frac{h}{2} - \tau)]_j + s_i\right)(\exp(d_{ii}h) - 1)/d_{ii}. \qquad (16)$$

Equation (16) forms the basis of the exponential algorithm. This equation expresses, in finite difference form, the components of the state vector $\bar{X}(t_\alpha+h)$ in terms of $\bar{X}(t_\alpha)$ and the undetermined exponential parameters.

II. DETERMINATION OF THE EXPONENTIAL PARAMETERS

Equation (16) shows that the solution of the differential equations for the time step t_α to $t_\alpha+h$ has been reduced to the problem of determining the appropriate ω's for the time step. The ω's are determined by an iterative method.

Solving Equation (14) for ω_i yields

$$\omega_i = \frac{1}{t'-t_\alpha} \; \ell n \left(\frac{x_i(t')}{x_i(t_\alpha')}\right). \tag{17}$$

By letting $t' = t_\alpha+h$, Equation (17) becomes

$$\omega_i = \frac{1}{h} \ell n \left(\frac{x_i(t_\alpha+h)}{x_i(t_\alpha)}\right) . \tag{18}$$

Once a set of ω's which satisfies Equation (18) is obtained, the solution is said to have converged on the interval t_α to $t_\alpha+h$. In general, Equation (18) can not be satisfied exactly. As a test upon convergence, the ω_i's are required to satisfy either

$$\left| \omega_i - \frac{1}{h} \ell n \left(\frac{x_i(t_\alpha+h)}{x_i(t_\alpha)}\right) \right| \leq \varepsilon_\ell \tag{19}$$

or

$$\left| \frac{\omega_i - \frac{1}{h} \ell n \left(\frac{x_i(t_\alpha+h)}{x_i(t_\alpha)}\right)}{\omega_i} \right| \leq \varepsilon_f \tag{20}$$

where ε_ℓ and ε_f are small positive numbers. For very small values of ω_i, the linear convergence test given by Equation (19) is used; otherwise, the fractional convergence test given by Equation (20) is used.

The iterative method used in this thesis to calculate the ω's is a modification of the method used by Swanks.[11] The method employs two schemes to estimate the ω's. Scheme 1 calculates new estimates of the ω's from Equation (18), while Scheme 2 calculates new estimates of the ω's by a linear interpolation based upon previous estimates of the ω's. For a detailed discussion of Scheme 2, the reader is referred to Reference 11.

The use of the iterative method in the determination of the ω's is diagramed in Figure 2. In Figure 2, the looping between control points 3 and 4 is an inner-iteration, and the looping between control points 2 and 5 is an outer-iteration. An inner-iteration estimates the individual ω_i's and hence the $x_i(t_\alpha + h)$'s. The completion of an outer-iteration yields a complete set of the ω's and hence an estimate of $\bar{X}(t_\alpha + h)$.

The initial estimates of the ω's are obtained from the derivatives of the dependent variables by using a first order Taylor series expansion of $\bar{X}(h)$ about t = 0.* The mathematical expression used to obtain the initial estimates of the ω's is

$$x_i(h) = x_i(0)\exp(\omega_i h) \sim x_i(0) + h \frac{dx_i}{dt}(0) \qquad (21)$$

or

*It has been found that it is advantageous, when introducing binary perturbations of the brine flowrate into a three stage flash evaporator, to use a modification of Equation (22) to estimate new exponential parameters whenever the sign of the perturbation changes.

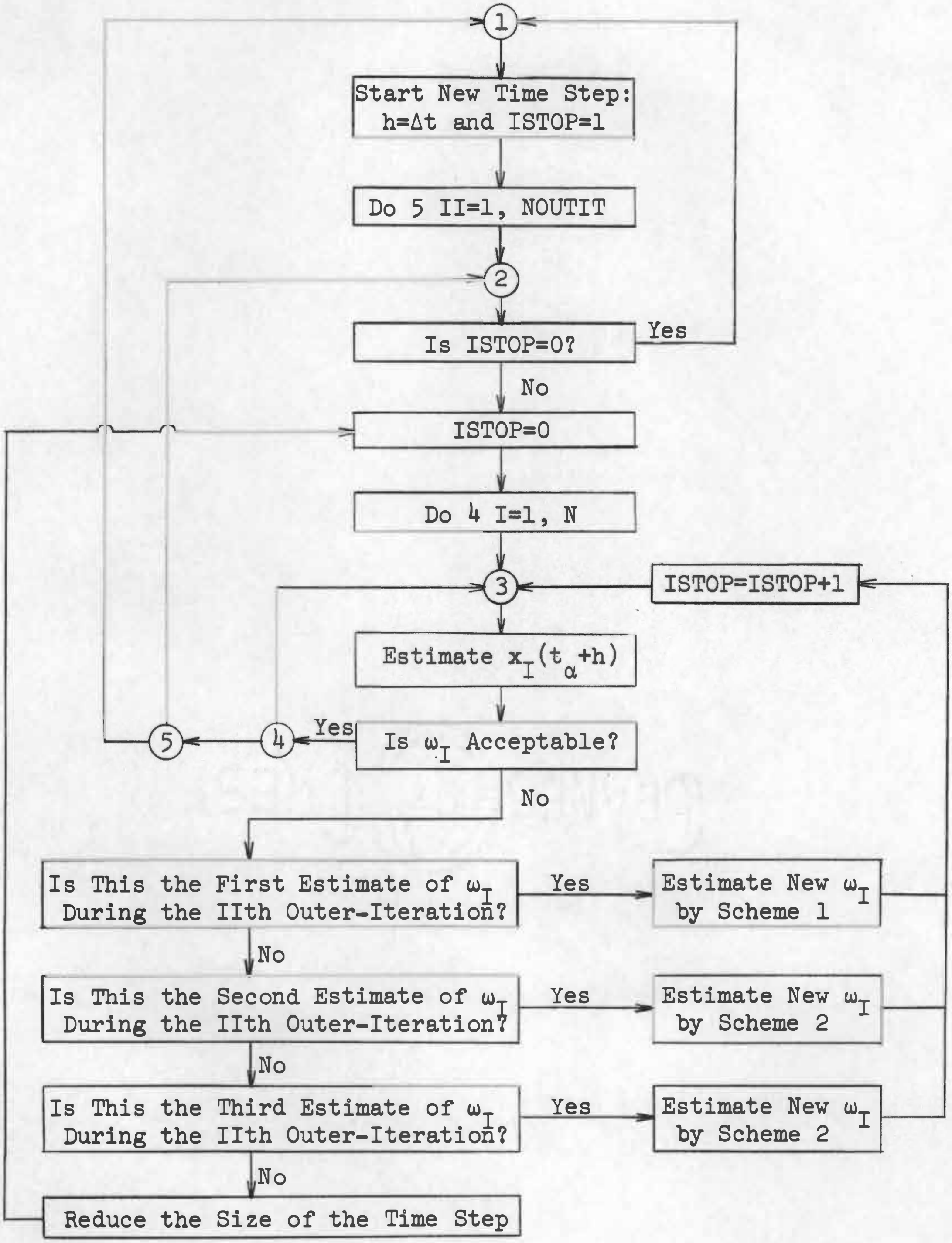

FIGURE 2

BLOCK DIAGRAM FOR DETERMINATION OF
EXPONENTIAL PARAMETERS

$$\omega_i \sim \frac{1}{h} \ln \left(1 + h \frac{dx_i}{dt}(0)/x_i(0)\right). \tag{22}$$

At the beginning of subsequent time steps, the ω's are set equal to the values calculated during the previous time step.

To initiate a time step, its size, h, is set equal to the input quantity Δt. An outer-iteration is then started and the values of $x_i(t_\alpha + h)$ are calculated individually. After the calculation of each $x_i(t_\alpha + h)$, a check is made to determine if ω_i satisfies the appropriate convergence criterion given by Equation (19) or Equation (20). If a ω_i does not satisfy the convergence criterion, up to three successive estimates of this ω_i are obtained. The first estimate is obtained from Scheme 1, and the next two estimates are obtained from Scheme 2. After each new estimate of ω_i is made, $x_i(t_\alpha + h)$ is calculated again and convergence is checked. Once a converged ω_i is found, the calculation proceeds to the next $x_i(t_\alpha + h)$. If a converged ω_i cannot be found in three estimates, the size of the time step, h, is reduced and the outer-iteration is started again. The above procedure is repeated until an outer-iteration is accomplished.

If a new estimate of any ω_i is made during an outer-iteration, an additional outer-iteration is performed (an additional outer-iteration is performed if ISTOP = 0). Although this requirement may be relaxed, it is employed because of the coupling of the differential equations. The maximum number of outer-iterations is specified by the input quantity NOUTIT. Once an outer-iteration in which all the ω's remain constant is completed or after NOUTIT outer-iterations have been performed, t_α is incremented and a new time step is started.

CHAPTER IV

NUMERICAL RESULTS

I. INTRODUCTION

The computer code ESNDE (Exponential Solution to Nonlinear Differential Equations) was developed from the exponential algorithm presented in Chapter III. A listing of this code and a discussion of input data is given in Appendix B. Although the code was developed specifically to solve the nonlinear multistage flash evaporator equations, it can be used to obtain solutions to sets of first order, nonlinear differential equations whose nonlinearities are of the form given by Equation (15) and whose dependent variables retain their original sign over the solution interval. The computer code handles variable coefficients by evaluating new coefficients at the beginning of each time step.

Solutions to several differential equations were obtained before the exponential algorithm was used to obtain solutions to a three stage evaporator system. Numerical solutions of two of these differential equations and the results obtained for two perturbations of the three stage flash evaporator described by Equation (1) are presented in this chapter.

II. A NUMERICAL SOLUTION TO THE MODIFIED BESSEL'S EQUATION

The modified Bessel's equation is[13]

$$t^2 \frac{d^2w}{dt^2} + t \frac{dw}{dt} - (t^2 + \nu^2)w = 0. \tag{23}$$

In order to apply the exponential algorithm to Equation (23), the equation is written as the following set of coupled, first order differential equations:

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = x_1 + \nu^2 \frac{x_1}{x_3^2} - \frac{x_2}{x_3}$$

$$\frac{dx_3}{dt} = 1 \tag{24}$$

where $x_1 = w$, $x_2 = \frac{dw}{dt}$, and $x_3 = t$. A solution of the modified Bessel's equation was obtained from the code ESNDE by solving Equation (24) with $\nu = 1$, $x_1(1) = 1$, $x_2(1) = 1$, and $x_3(1) = 1$. Tabulated results of the exponential algorithm solution are presented in Table 2 along with numerical values of the analytic solution,

$$w = aI_1(t) + bK_1(t), \tag{25}$$

where

$$a = (2K_1(1) + K_0(1))/(I_0(1)K_1(1) + I_1(1)K_0(1)),$$

TABLE 2

SOLUTIONS TO MODIFIED BESSEL'S EQUATION FOR $\nu = 1$, $w(1) = 1$, AND $dw(1)/dt = 1$

Time (Seconds)	Exponential Algorithm Solution	Numerical Values of Analytic Solution
1.0	1.000	1.000
2.0	2.603	2.604
4.0	1.586×10^{1}	1.586×10^{1}
6.0	9.967×10^{1}	9.967×10^{1}
8.0	6.497×10^{2}	6.497×10^{2}
10.0	4.340×10^{3}	4.340×10^{3}
12.0	2.947×10^{4}	2.948×10^{4}
14.0	2.026×10^{5}	2.026×10^{5}
16.0	1.405×10^{6}	1.406×10^{6}
18.0	9.816×10^{6}	9.819×10^{6}
20.0	6.896×10^{7}	6.898×10^{7}
22.0	4.867×10^{8}	4.868×10^{8}
24.0	3.448×10^{9}	3.449×10^{9}
26.0	2.451×10^{10}	2.451×10^{10}
28.0	1.745×10^{11}	1.747×10^{11}
30.0	1.248×10^{12}	1.248×10^{12}
32.0	8.935×10^{12}	8.938×10^{12}
34.0	6.409×10^{13}	6.412×10^{13}
36.0	4.605×10^{14}	4.607×10^{14}
38.0	3.314×10^{15}	3.315×10^{15}
40.0	2.388×10^{16}	2.389×10^{16}

$$b = (1 - aI_1(1))/K_1(1), \tag{26}$$

and I_1, I_0, K_0, and K_1 are modified Bessel functions.

The agreement of the exponential algorithm solution of the modified Bessel's equation with the analytic solution is excellent over an extremely large range of numerical values. Comparison of the results shows that the fractional difference between the two solutions is generally less than 10^{-3}.

III. NUMERICAL SOLUTIONS TO VAN DER POL'S EQUATION

Van der Pol's equation, describing a triode oscillator, is[14]

$$\frac{d^2y}{dx^2} - \varepsilon(1 - y^2)\frac{dy}{dx} + y = 0. \tag{27}$$

In order to apply the exponential algorithm to Equation (27), the transformations

$$x_1 = y + \gamma$$

$$x_2 = \frac{dy}{dx} + \rho \tag{28}$$

are made; and the resulting transformed differential equation is written as the following set of coupled, first order differential equations:

$$\frac{dx_1}{dx} = x_2 - \rho$$

$$\frac{dx_2}{dx} = -\ (2\varepsilon\gamma\rho + 1)\ x_1 + \varepsilon(1 - \gamma^2)x_2$$

$$+\ \varepsilon\rho x_1^2 + 2\varepsilon\gamma x_1 x_2 - \varepsilon x_1^2 x_2$$

$$+\ \varepsilon\rho(\gamma^2 - 1) + \gamma. \qquad (29)$$

In Equation (28), γ and ρ are positive real constants whose magnitudes are sufficiently large to insure that x_1 and x_2 are always positive.

Numerical solutions of Van der Pol's equation with initial conditions of $y(0) = 2$ and $\frac{dy}{dx}(0) = 0$ and values of ε between 0.5 and 5.0 were obtained by applying the exponential algorithm (ESNDE) to Equation (29). The results are presented in Figure 3 along with approximate solutions given by Davis.[14] Figure 3 shows good agreement between the exponential algorithm solutions and the approximate solutions of Davis.

IV. NUMERICAL SOLUTION OF THREE STAGE EVAPORATOR SYSTEM

Solutions for two perturbations in the three stage flash evaporator system described by Equation (1) were obtained from the code ESNDE. Two approaches were taken to include the time dependence of the pressure differences given by Equation (2) given on page 14. The first approach was the introduction of two additional differential equations to describe

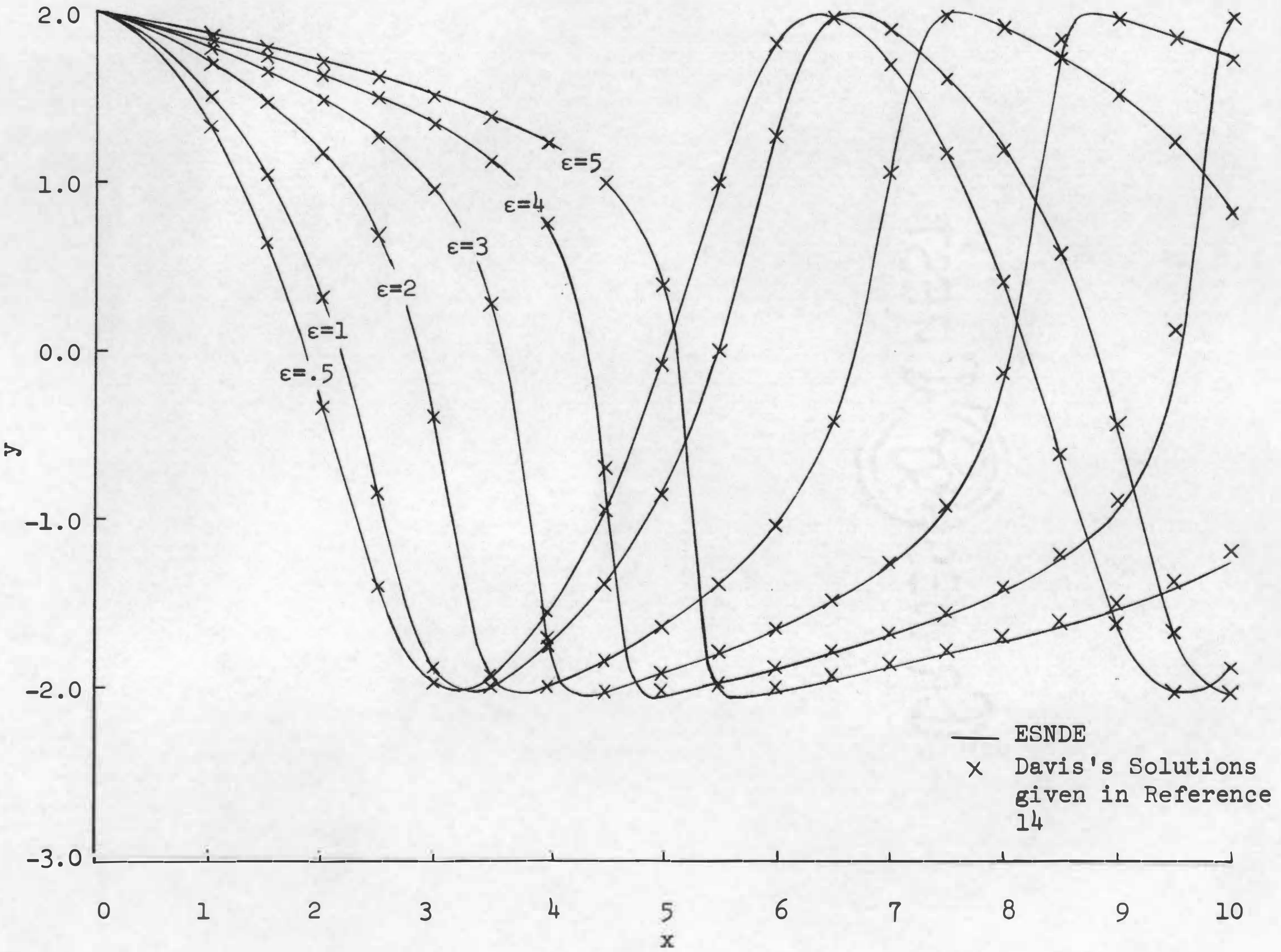

FIGURE 3

SOLUTIONS TO VAN DER POL'S EQUATION FOR y(0)=2 AND dy(0)/dx=0 the time dependence of the pressure differences by differentiating Equation (2). The second approach consisted of the evaluation of the pressure differences at the end of each time step from Equation (2) and an iterative solution of an assumed exponential dependence of the pressure differences based upon Equations (19) and (20) given on page 22. When the time dependence of the pressure differences was included, the computation time required to obtain a converged solution became prohibitive.

Solutions were then obtained by assuming that the pressure differences remained constant over each time step. This assumption is not very restrictive since the changes in the pressure differences are small over a time step and only fractional powers, less than or equal to one-half, of the pressure differences appear in Equation (1). The solutions obtained by including the time dependence of the pressure differences and by assuming the pressure difference remained constant over a time step agreed very well. For the above reasons, the pressure differences were assumed constant over a time step and were evaluated from Equation (2) at the end of each time step.

Solutions of the three stage evaporator system obtained from the exponential algorithm (ESNDE) are compared with solutions obtained by Ball[15] using MATEXP and from a computer program written to solve Equation (1) by the Euler method.[16] The solutions obtained by the Euler method are taken as the reference solutions because reductions by a factor of ten in the time step yielded no significant changes in any of the dependent variables.

The first perturbation was a ten degree (10°F) step change in the steam heater temperature, T_{SH}. The second perturbation was a twenty percent step change in the brine flowrate, W_{BR}. The initial conditions used were not the steady state values of the system. For this reason, the perturbations were introduced after a forty second interval.

Tabulated results of the solutions are presented in Tables 3 and 4. Plots of the transient responses of the vapor mass and brine mass in the first stage and the brine mass in the third stage are presented in Figures 4, 5, and 6 for the first perturbation and in Figures 7, 8, and 9 for the second perturbation. The time steps were 0.2 seconds for ESNDE, 0.1 seconds for MATEXP, and 0.02 seconds for the Euler method.

The tabulated results show a general agreement between the three methods of solution. The transient responses show excellent agreement between the Euler solution and the exponential algorithm solution and some discrepancies between the Euler solution and the MATEXP solution primarily due to the appearance of oscillations in the MATEXP solution. The oscillations in the MATEXP solution are attributed to either the use of too large a time step since similar oscillations appearing in other problems have been eliminated by a reduction in the size of the time step[15] or the possibility that a slightly different set of differential equations was used to obtain the MATEXP solutions.

The exponential algorithm results were obtained by requiring that at each time step none of the values of the exponential parameters

TABLE 3

TABULATED RESULTS FOR A 10°F STEP CHANGE IN THE STEAM HEATER TEMPERATURE OF A THREE STAGE FLASH EVAPORATOR

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
	ESNDE		90.34	93.06	97.15
x_1	Euler	89.00	90.33	93.05	97.10
	MATEXP		90.32	93.01	97.07
	ESNDE		88.46	91.00	94.95
x_2	Euler	87.00	88.45	90.99	94.90
	MATEXP		88.44	90.95	94.87
	ESNDE		98.17	102.0	106.6
x_3	Euler	89.00	98.16	102.0	106.6
	MATEXP		98.15	102.0	106.6
	ESNDE		.4256	.4902	.5530
x_4	Euler	.4114	.4256	.4900	.5524
	MATEXP		.4258	.4959	.5743
	ESNDE		107.1	112.1	117.3
x_5	Euler	108.1	107.1	112.1	117.3
	MATEXP		107.1	112.1	117.3
	ESNDE		433.7	407.3	393.3
x_6	Euler	421.4	433.6	407.6	393.7
	MATEXP		432.6	404.6	385.2
	ESNDE		86.59	88.94	92.74
x_7	Euler	84.99	86.58	88.92	92.70
	MATEXP		86.57	88.89	92.68
	ESNDE		84.61	86.77	90.43
x_8	Euler	82.99	84.59	86.76	90.39
	MATEXP		84.58	86.73	90.37
	ESNDE		95.21	98.63	103.0
x_9	Euler	84.99	95.20	98.62	103.0
	MATEXP		95.20	98.58	103.0

TABLE 3 (continued)

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
x_{10}	ESNDE		.6155	.6989	.7881
	Euler	.5901	.6155	.6988	.7876
	MATEXP		.6160	.7067	.8178
x_{11}	ESNDE		104.8	109.0	114.1
	Euler	105.7	104.8	108.9	114.0
	MATEXP		104.8	108.9	114.0
x_{12}	ESNDE		1069.	1051.	1041.
	Euler	1048.	1069.	1051.	1041.
	MATEXP		1069.	1048.	1030.
x_{13}	ESNDE		82.62	84.61	88.12
	Euler	80.98	82.61	84.59	88.08
	MATEXP		82.61	84.57	88.05
x_{14}	ESNDE		80.26	82.06	85.39
	Euler	78.98	80.25	82.05	85.35
	MATEXP		80.25	82.03	85.33
x_{15}	ESNDE		93.50	96.51	100.8
	Euler	80.98	93.49	96.50	100.8
	MATEXP		93.49	96.47	100.8
x_{16}	ESNDE		.3406	.3788	.4271
	Euler	.3222	.3406	.3787	.4266
	MATEXP		.3410	.3827	.4420
x_{17}	ESNDE		102.0	105.1	110.1
	Euler	102.8	101.9	105.0	110.1
	MATEXP		101.9	105.0	110.0
x_{18}	ESNDE		1721.	1800.	1834.
	Euler	1705.	1721.	1800.	1834.
	MATEXP		1720.	1801.	1840.
x_{19}	ESNDE		77.91	79.52	82.67
	Euler	77.00	77.89	79.50	82.63
	MATEXP		77.89	79.49	82.61

TABLE 3 (continued)

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
x_{20}	ESNDE	110.3	109.4	115.0	120.3
	Euler		109.4	115.0	120.2
	MATEXP		109.4	114.9	120.2
x_{21}	ESNDE	106.5	105.6	110.2	115.2
	Euler		105.6	110.2	115.2
	MATEXP		105.6	110.1	115.2
x_{22}	ESNDE	110.3	109.2	116.1	120.9
	Euler		109.2	116.0	120.8
	MATEXP		109.2	116.0	120.8
x_{23}	ESNDE	125.3	120.0	134.2	138.2
	Euler		120.0	134.2	138.2
	MATEXP		120.0	134.1	138.1

TABLE 4

TABULATED RESULTS FOR A 20% STEP CHANGE
IN THE BRINE FLOWRATE OF A
THREE STAGE FLASH EVAPORATOR

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
x_1	ESNDE	89.00	90.34	90.74	91.23
	Euler		90.33	90.71	91.17
	MATEXP		90.32	90.70	91.15
x_2	ESNDE	87.00	88.46	88.98	89.50
	Euler		88.45	88.95	89.46
	MATEXP		88.44	88.94	89.42
x_3	ESNDE	89.00	98.16	98.08	98.42
	Euler		98.16	98.06	98.38
	MATEXP		98.15	98.05	98.34
x_4	ESNDE	.4114	.4256	.4176	.4196
	Euler		.4256	.4174	.4192
	MATEXP		.4258	.4174	.4186
x_5	ESNDE	108.1	107.1	106.7	106.8
	Euler		107.1	106.7	106.8
	MATEXP		107.1	106.7	106.8
x_6	ESNDE	421.4	433.7	497.0	511.2
	Euler		433.7	497.0	511.3
	MATEXP		432.6	497.1	511.3
x_7	ESNDE	84.99	86.59	87.22	87.77
	Euler		86.58	87.20	87.73
	MATEXP		86.57	87.18	87.70
x_8	ESNDE	82.99	84.60	85.32	85.90
	Euler		84.59	85.30	85.86
	MATEXP		84.58	85.29	85.83
x_9	ESNDE	84.99	95.21	95.44	95.82
	Euler		95.20	95.42	95.78
	MATEXP		95.20	95.41	95.74

TABLE 4 (continued)

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
x_{10}	ESNDE		.6154	.6116	.6148
	Euler	.5901	.6155	.6113	.6142
	MATEXP		.6160	.6116	.6138
x_{11}	ESNDE		104.8	104.7	104.9
	Euler	105.7	104.8	104.7	104.9
	MATEXP		104.8	104.7	104.8
x_{12}	ESNDE		1069.	113.9	1174.
	Euler	1048.	1069.	113.9	1174.
	MATEXP		1069.	113.7	1173.
x_{13}	ESNDE		82.62	83.42	84.03
	Euler	80.98	82.61	83.40	83.99
	MATEXP		82.61	83.39	83.97
x_{14}	ESNDE		80.26	81.12	81.76
	Euler	78.98	80.25	81.10	81.72
	MATEXP		80.25	81.10	81.71
x_{15}	ESNDE		93.50	93.94	94.32
	Euler	80.98	93.49	93.91	94.28
	MATEXP		93.49	93.90	94.25
x_{16}	ESNDE		.3407	.3421	.3440
	Euler	.3222	.3406	.3419	.3437
	MATEXP		.3410	.3421	.3438
x_{17}	ESNDE		102.0	102.3	102.5
	Euler	102.8	101.9	102.3	102.5
	MATEXP		101.9	102.3	102.4
x_{18}	ESNDE		1721.	1696.	1766.
	Euler	1705.	1721.	1697.	1767.
	MATEXP		1720.	1692.	1762.
x_{19}	ESNDE		77.90	78.82	79.50
	Euler	77.00	77.89	78.80	79.46
	MATEXP		77.89	78.80	79.45

TABLE 4 (continued)

Variable	Method of Solution	Initial Condition	Value of Variable at 40 Seconds	Value of Variable at 100 Seconds	Value of Variable at 200 Seconds
	ESNDE		109.4	108.5	108.6
x_{20}	Euler	110.3	109.4	108.5	108.6
	MATEXP		109.4	108.5	108.5
	ESNDE		105.6	105.4	105.6
x_{21}	Euler	106.5	105.6	105.3	105.5
	MATEXP		105.6	105.3	105.5
	ESNDE		109.2	108.5	108.7
x_{22}	Euler	110.3	109.2	108.4	108.6
	MATEXP		109.2	108.4	108.6
	ESNDE		120.0	118.2	118.4
x_{23}	Euler	125.3	120.0	118.2	118.4
	MATEXP		120.0	118.2	118.3

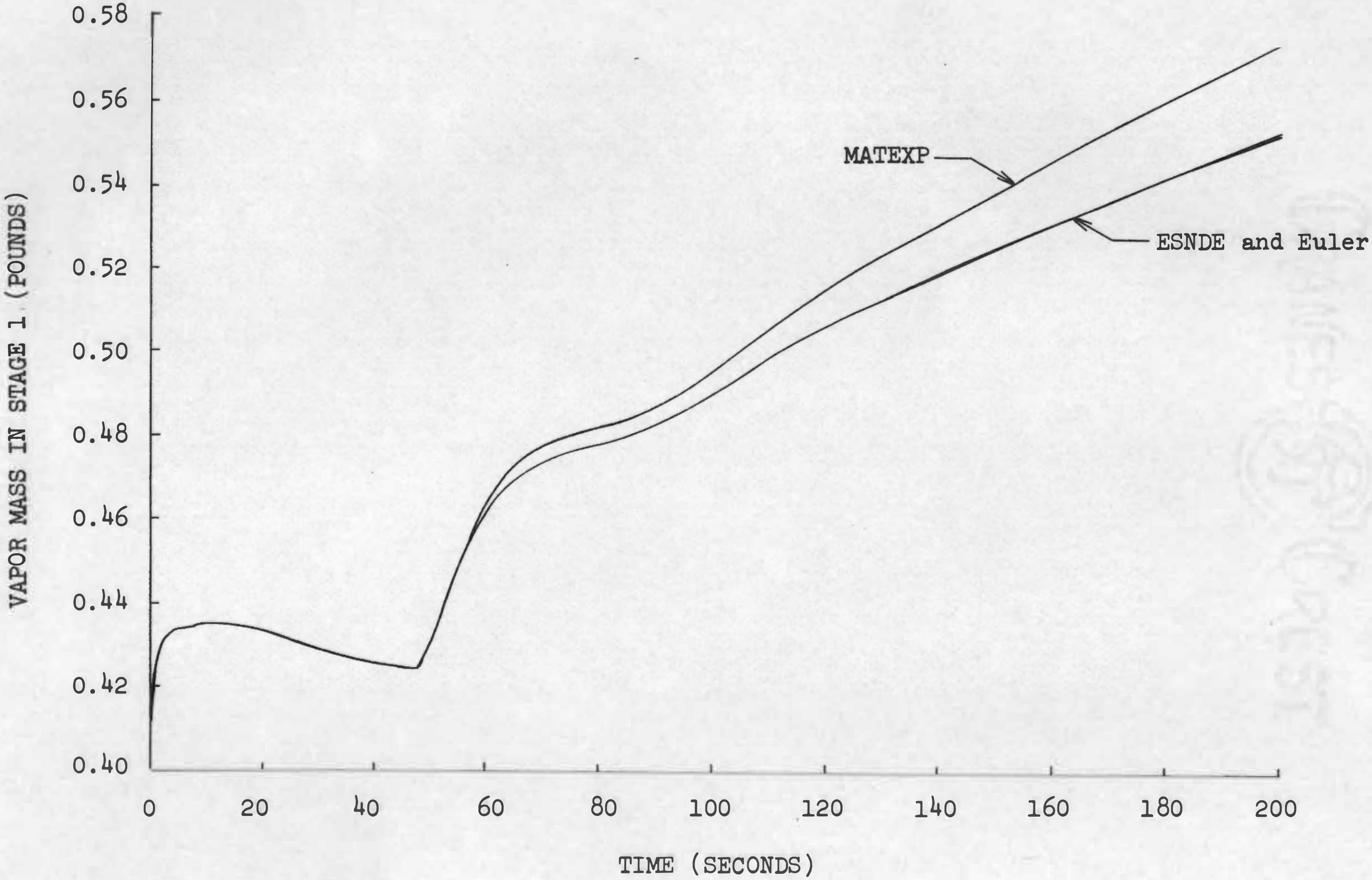

FIGURE 4

TRANSIENT RESPONSE OF VAPOR MASS IN STAGE 1 FOR A 10°F STEP CHANGE IN THE STEAM HEATER TEMPERATURE OF A THREE STAGE FLASH EVAPORATOR

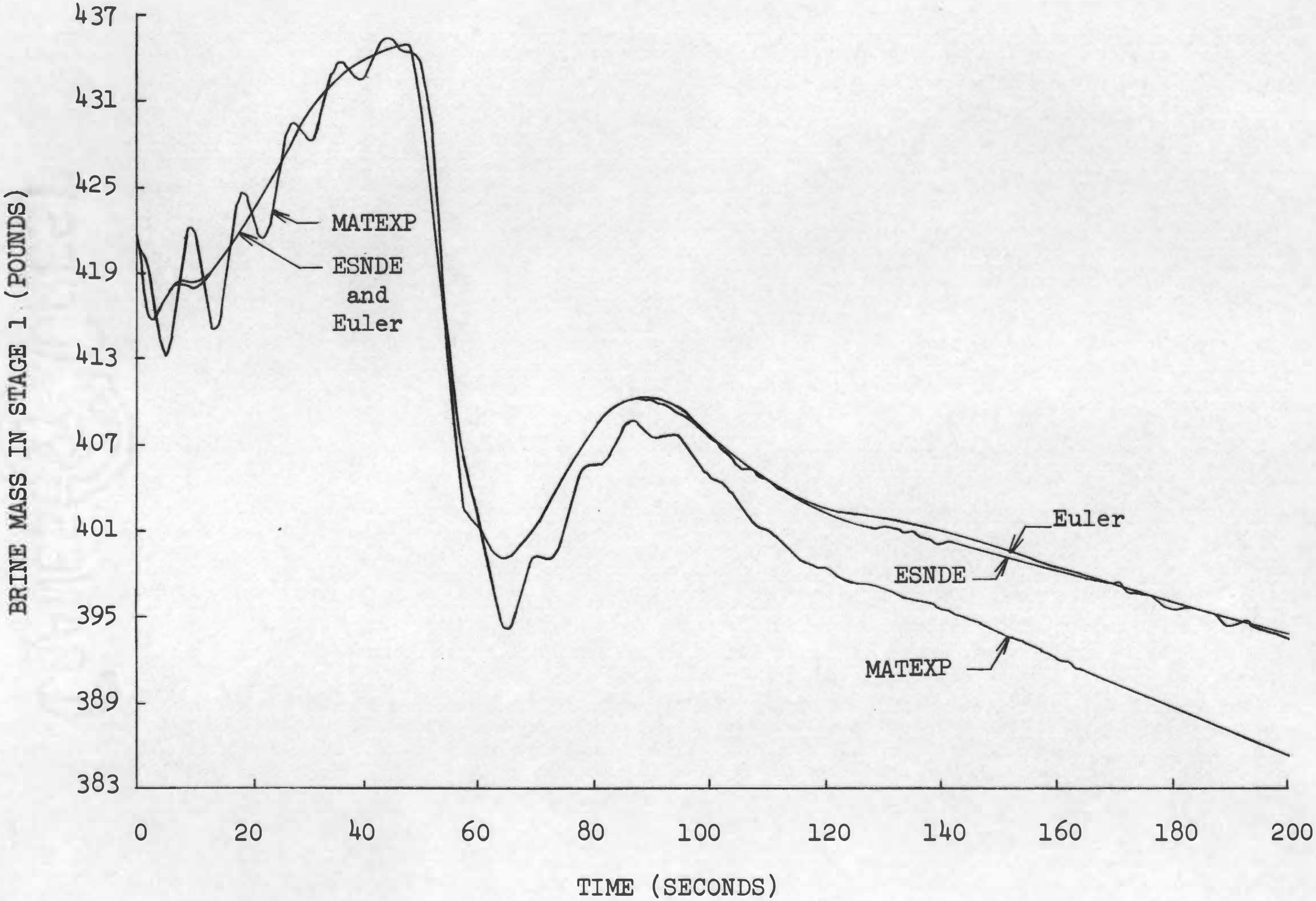

FIGURE 5

TRANSIENT RESPONSE OF BRINE MASS IN STAGE 1 FOR A 10°F STEP CHANGE IN THE STEAM HEATER TEMPERATURE OF A THREE STAGE FLASH EVAPORATOR

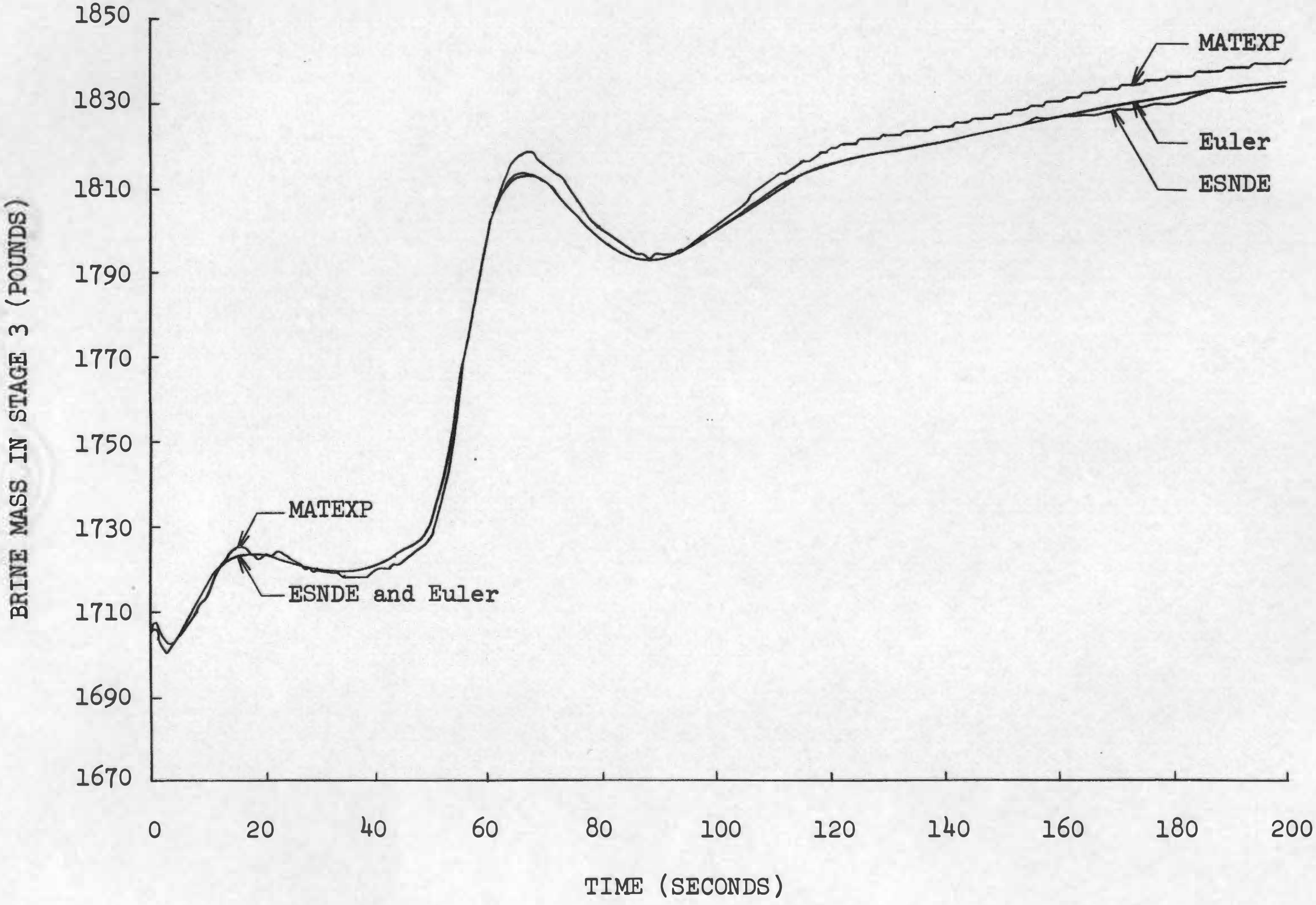

FIGURE 6

TRANSIENT RESPONSE OF BRINE MASS IN STAGE 3 FOR A 10°F STEP CHANGE IN THE STEAM HEATER TEMPERATURE OF A THREE STAGE FLASH EVAPORATOR

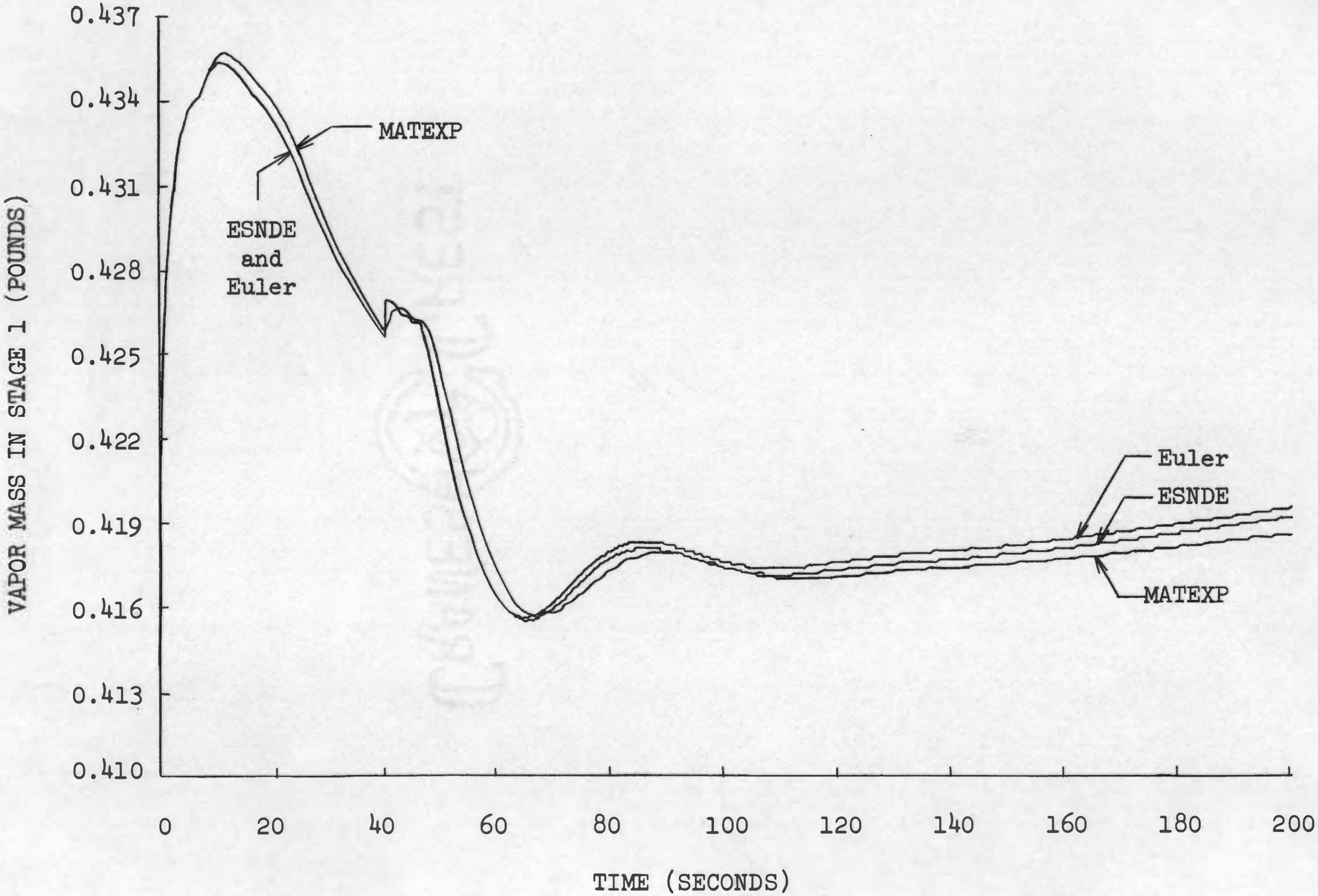

FIGURE 7

TRANSIENT RESPONSE OF VAPOR MASS IN STAGE 1 FOR A 20% STEP CHANGE IN THE BRINE FLOWRATE OF A THREE STAGE FLASH EVAPORATOR

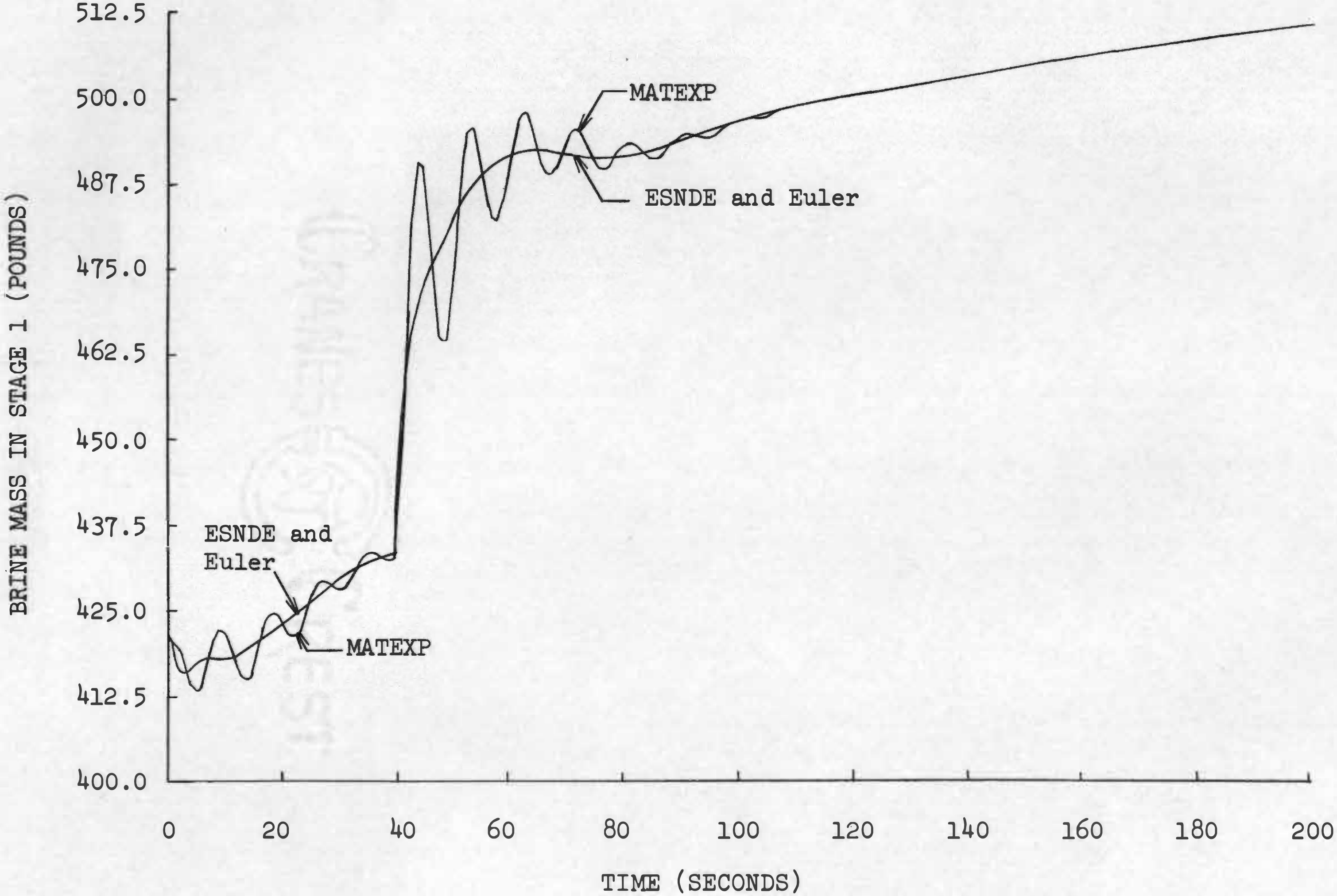

FIGURE 8

TRANSIENT RESPONSE OF BRINE MASS IN STAGE 1 FOR A 20% STEP CHANGE IN THE BRINE FLOWRATE OF A THREE STAGE FLASH EVAPORATOR

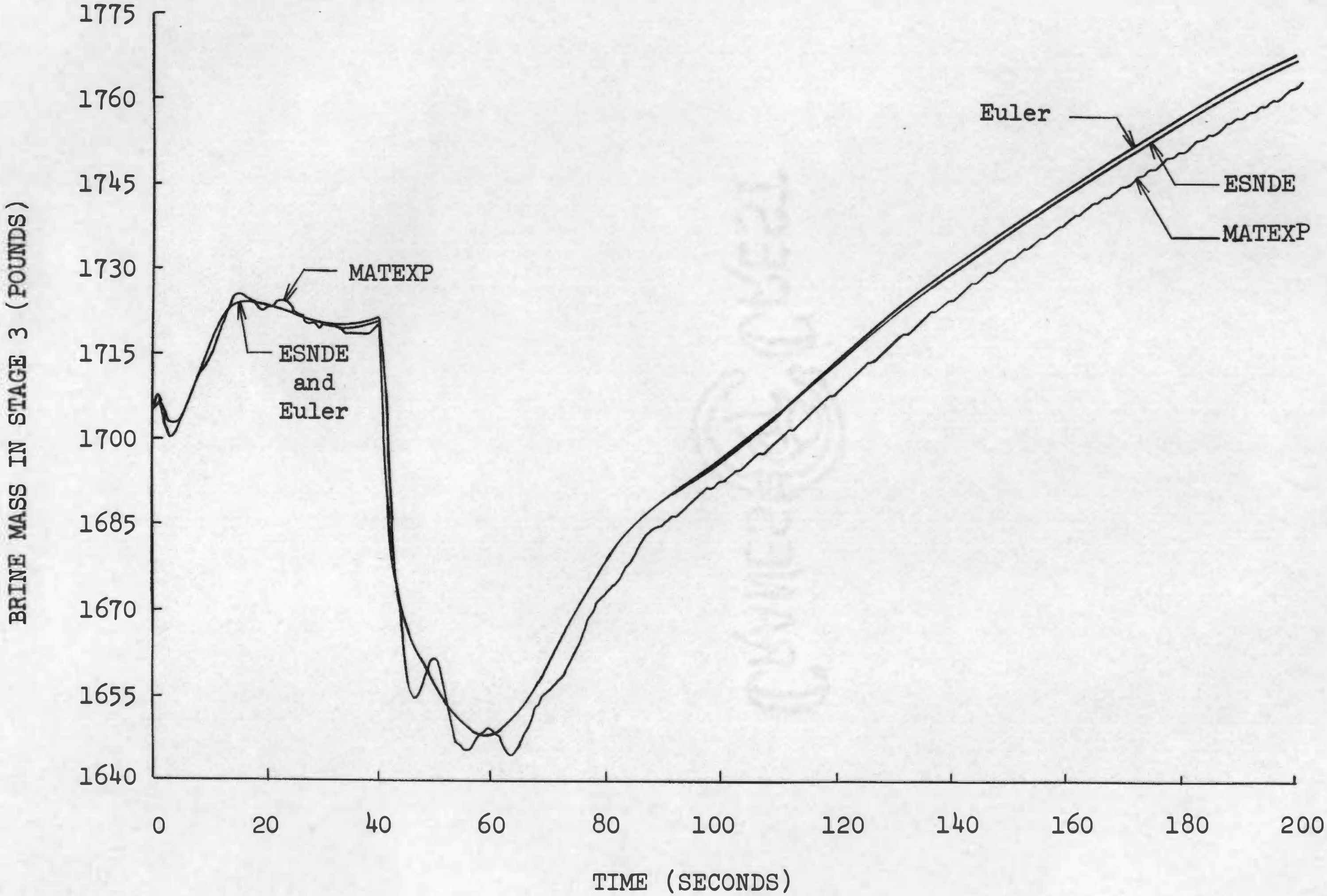

FIGURE 9

TRANSIENT RESPONSE OF BRINE MASS IN STAGE 3 FOR A 20% STEP CHANGE IN THE BRINE FLOWRATE OF A THREE STAGE FLASH EVAPORATOR changed during the last outer-iteration. In order to reduce the computation time, this requirement was relaxed, and only one outer-iteration was performed at each time step. The solutions obtained in this manner were essentially identical to the solutions presented in this section. In addition, it was found that comparable results could be obtained by increasing the size of the time step to 0.6 seconds. The time step of 0.6 seconds was found to be the maximum acceptable time step regardless of the number of outer-iterations allowed. At larger time steps, several of the dependent variables attempted to take on negative values which are physically unacceptable.

The computation time required by the exponential algorithm to obtain solutions to the three stage evaporator system with only one outer-iteration and a time step size of 0.6 seconds is almost identical to the time required by MATEXP to obtain the solutions presented in this section.

CHAPTER V

CONCLUSIONS

The results presented in Chapter IV demonstrate the successful use of the exponential algorithm in the solution of specific non-linear differential equations. Analytic predictions of the numerical stability of the algorithm are not available, but the algorithm is useful for obtaining solutions to the type of nonlinear equations encountered in this study.

Since the exponential algorithm presented in this thesis is based upon an iterative method of solution, it should allow the use of larger time steps than are allowed in non-iterative methods of solution. The results presented in Chapter IV for the three stage evaporator system tend to confirm this statement. The ability to use a larger time step coupled with the option to reduce the time step when convergence is difficult makes the exponential algorithm attractive when estimates of a suitable time step are not available for other methods of solution.

As discussed in Chapter IV, a reduction in the MATEXP time step may be required in order to remove the oscillations in the three stage evaporator solutions. Since the computation time required by MATEXP is approximately proportional to the number of time intervals,[8] the time required to obtain accurate solutions for a three stage evaporator system will be less for the exponential algorithm than for MATEXP if a reduction of the MATEXP time step is required.

As noted in Chapter IV, only one outer-iteration was required to obtain solutions for the three stage evaporator system. When only one outer-iteration is specified, the computation time is approximately proportional to the number of equations to be solved. For this reason, the use of the exponential algorithm with one outer-iteration is attractive for the solution of the large sets of nonlinear differential equations that describe a multistage flash evaporator.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Ball, S. J., "Nuclear Desalination Dual-Purpose Plant Control Studies Interim Report," USAEC Report ORNL-TM-1618, Part I, Oak Ridge National Laboratory (October, 1966).

2. Ball, S. J., "Nuclear Desalination Dual-Purpose Plant Control Studies Interim Report Appendix," USAEC Report ORNL-TM-1618, Part II, Oak Ridge National Laboratory (January, 1967).

3. Kerlin, T. W., et al., "Nuclear Desalination Plant Dynamics: Modeling and Analysis of a Multistage Flash Evaporator," Report NEUT 2806-1, Nuclear Engineering Department, University of Tennessee, Knoxville (1967).

4. Kerlin, T. W., et al., "Nuclear Desalination Plant Dynamics: Modeling and Analysis of a Multistage Flash Evaporator," Report NEUT 2806-2, Nuclear Engineering Department, University of Tennessee, Knoxville (1968).

5. Wright, W. C. and T. W. Kerlin, "An Efficient Computer-Oriented Method for Stability Analysis of Large Multivariable Systems," Report NEUT 2806-3, Nuclear Engineering Department, University of Tennessee, Knoxville (1968).

6. Ball, S. J., et al., "Dynamics Experiments on the AMF Millstone Point Flash Evaporator," USAEC Report ORNL-TM-2188, Oak Ridge National Laboratory (July, 1968).

7. Kerlin, T. W., et al., "Nuclear Desalination Plant Dynamics: Modeling and Analysis of a Multistage Flash Evaporator," Report NEUT 2806-4, Nuclear Engineering Department, University of Tennessee, Knoxville (1969).

8. Ball, S. J. and R. K. Adams, "MATEXP, A General Purpose Digital Computer Program for Solving Ordinary Differential Equations by the Matrix Exponential Method," USAEC Report ORNL-TM-1933, Oak Ridge National Laboratory (August, 1967).

9. Andrews, J. B., II and K. F. Hansen, "Numerical Solution of the Time-Dependent Multigroup Diffusion Equations," Nuc. Sci. Eng., 31, 304-313 (February, 1968).

10. McCormick, W. T., Jr. and K. F. Hansen, "Numerical Solution of the Two-Dimensional Time-Dependent Multigroup Equations," Proceedings of Conference on the Effective Use of Computers in Nuclear Industry, USAEC Report CONF-690401, 76-101 (April, 1969).

11. Swanks, J. H., "Approximate Numerical Solutions to the Time-Dependent Neutron Transport Equations," Order No. 69-16, 535, Ann Arbor, Michigan: University Microfilms (1969).

12. Stevenson, M. G. and B. E. Bingham, "TART-An LMFBR Transient Analysis Project," Proceedings of Conference on the Effective Use of Computers in Nuclear Industry, USAEC Report CONF-690401, 16-29 (April, 1969)

13. Abramowitz, M. and I. A. Stegun, editors, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Washington: U. S. Government Printing Office (1964).

14. Davis, H. T., Introduction to Nonlinear Differential and Integral Equations, New York: Dover Publications Inc. (1962).

15. Ball, S. J., Oak Ridge National Laboratory, Oak Ridge, Tennessee, personal communication with the author.

16. McCracken, D. D. and W. S. Dorn, Numerical Methods and Fortran Programming, New York: John Wiley and Sons, Inc. (1964).

APPENDIXES

APPENDIX A

TERMINOLOGY USED FOR THE FLASH EVAPORATOR

The terminology used to describe the three stage flash evaporator is basically that used in Reference 6. The terms used in the development of Equations (1) and (2) are defined below:

M = mass (pounds)

W = mass flowrate (pounds/second)

T = temperature (°F)

P = pressure (psi)

C_p = specific heat (Btu/pound/°F)

h = heat transfer coefficient (Btu/second/ft^2/°F)

h_{fg} = heat of vaporization (Btu/pound)

ρ = density (pounds/ft^3)

A = heat transfer area (ft^2)

AFC = base area of channel (ft^2)

VV = vapor volume (ft^3)

Recf = coolant recirculation fraction

XW1 = flow coefficient between stages (pounds/second/$psi^{1/2}$)

K = flashing flow coefficient ($pounds^{1/2}$/$second^{1/2}$/°F)

XM3B = bias term accounting for curvature of Stage 3 sides (psi)

K2 = importance factor for downstream liquid level

τ = time delay (seconds)

$$\alpha = \left(\frac{\partial T_{V-SAT}}{\partial P_V}\right)_{P_{VO}} \left(\frac{1}{VV}\right) \left(\frac{\partial P_V}{\partial \rho_{V-SAT}}\right)_{\rho_{V-SATO}} \quad (^{\circ}F/pound)$$

$$\beta = T_{VO} - \alpha M_{VO} \quad (^{\circ}F)$$

$$\gamma = \left(\frac{\partial P_V}{\partial \rho_{V-SAT}}\right)_{P_{VO}} \left(\frac{1}{VV}\right) \quad (psi/pound)$$

$$\xi = P_{VO} - \gamma M_{VO} \quad (psi).$$

The following subscripts are applied to the above terms:

0 = initial

1 = Stage 1

2 = Stage 2

3 = Stage 3

i = inside

o = outside, outlet

T = water in tubes

TB = tray brine

CV = cell vapor

BR = brine

C = coolant water

CL = coolant in reservoir

IP = inlet plenum

SH = steam heater

Tube = tube

SAT = saturated

V = vapor.

APPENDIX B

THE COMPUTER CODE

I. INPUT INFORMATION

The following general representation of the i^{th} differential equation is used to discuss the input data required by the computer code ESNDE:

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} a_{ij}x_j + d_{ii}x_i + \sum_{j=1}^{I1_i} C1_j x_{V1_j}^{P1_j}$$

$$+ \sum_{j=1}^{I2_i} C2_j x_{V21_j}^{P21_j} x_{V22_j}^{P22_j} \sum_{j=1}^{I3_i} C3_j x_{V31_j}^{P31_j} x_{V32_j}^{P32_j} x_{V33_j}^{P33_j}$$

$$+ \sum_{j=1}^{I4_i} C4_j x_{V4_j} (t - \tau_j) + s_i$$

where

N is the number of dependent variables;

x_i is the i^{th} dependent variable;

a_{ij} is the ij^{th} element of the linear matrix A in Equation (5);

d_{ii} is the ii^{th} element of the diagonal matrix D in Equation (5);

$I1_i$ is the number of type-1 nonlinearities in the i^{th} differential equation;

$I2_i$ is the number of type-2 nonlinearities in the i^{th} differential equation;

$I3_i$ is the number of type-3 nonlinearities in the i^{th} differential equation;

$I4_i$ is the number of pure time delays in the i^{th} differential equation;

$C1_j$'s, $C2_j$'s, and $C3_j$'s are the coefficients of the type-1, type-2, and type-3 nonlinearities respectively;

$C4_j$'s are the coefficients of the time-lagged terms;

$V1_j$'s, $V21_j$'s, $V22_j$'s, $V31_j$'s, $V32_j$'s, $V33_j$'s, and $V4_j$'s are integers corresponding to the indices of the dependent variables;

$P1_j$'s, $P21_j$'s, $P22_j$'s, $P31_j$'s, $P32_j$'s, and $P33_j$'s are real numbers denoting the powers to which the dependent variables are raised;

τ_j's are the time delays; and

s_i is the i^{th} element of forcing function vector $\bar{S}$ in Equation (5).

The arrangement of the data cards for a typical problem is shown in Figure 10. The layouts of the individual data cards are:

Title Card

This card may contain up to 80 alphanumeric characters.

Control Card 1

The following data are specified on this card in a 2I4 format:

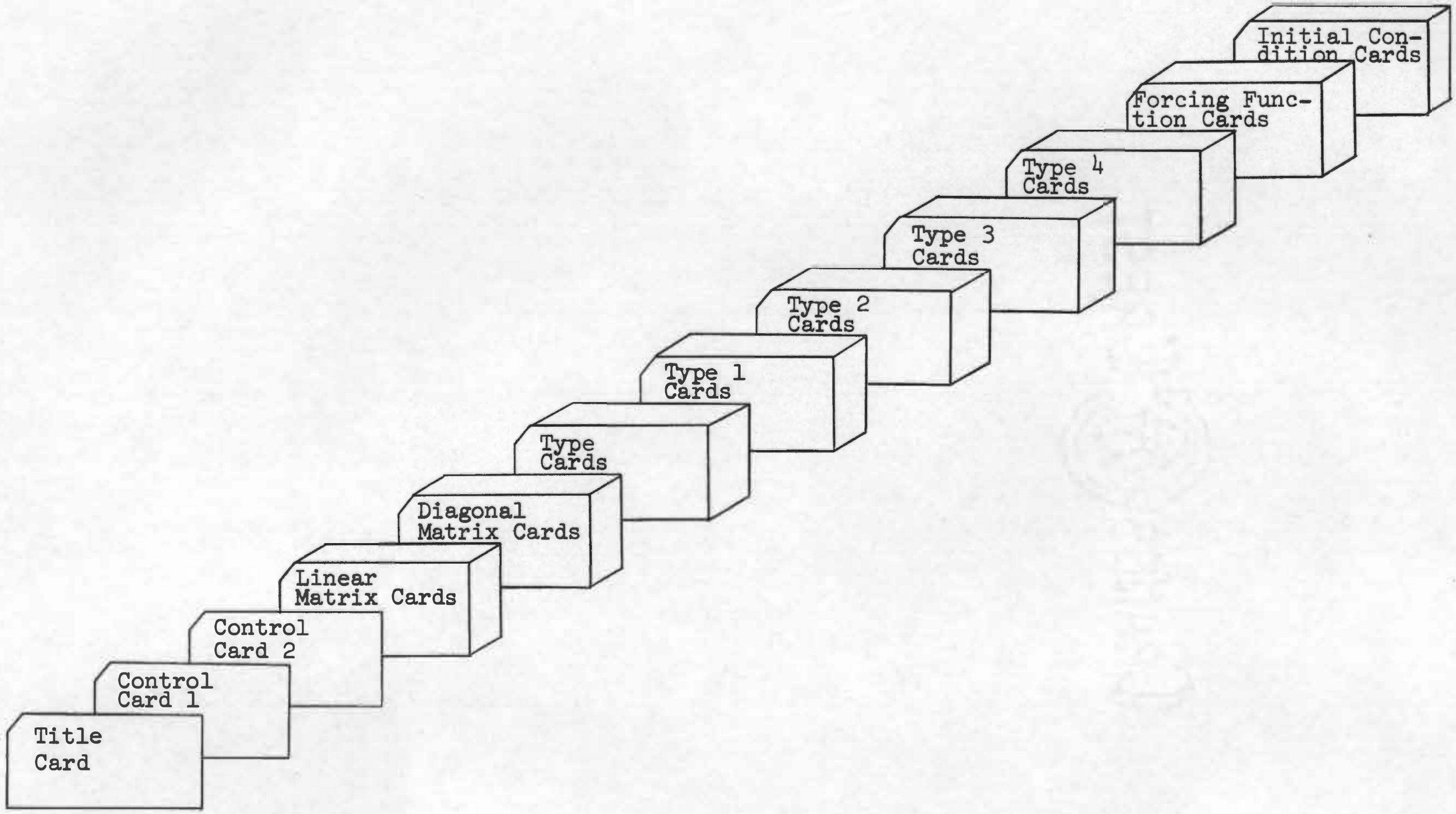

FIGURE 10

ESNDE INPUT DATA ARRANGEMENT

1. N = the number of dependent variables, and
2. NOUTIT = the maximum number of outer-iterations.

Control Card 2

The following data are specified on this card in a 7E10.3 format:

1. TO = starting time for solution,
2. DELTAT = maximum size of solution time step,
3. TMAX = final solution time,
4. HMIN = minimum size of solution time step,
5. CONVLR = line or convergence constant (ε_ℓ),
6. CONVFR = fractional convergence constant (ε_f), and
7. CONVCH = value of exponential parameters (ω's) above which fractional convergence test is applied and below which linear convergence test is applied.

Linear Matrix Cards

These cards are used to input the linear matrix A. The elements of the linear matrix, a_{ij}'s, are specified by rows in a 8E10.3 format.

Diagonal Matrix Cards

These cards are used to input the diagonal elements of the diagonal matrix D. The diagonal elements, d_{ii}'s, are specified in a 8E10.3 format.

Type Cards

For each differential equation, four integers are required to denote the number and type of nonlinearities and the number of time-lagged terms. These numbers, corresponding to $I1_i$, $I2_i$, $I3_i$, and $I4_i$ in Equation (30), are specified in the aforementioned order in a 20I4 format.

Type-1 Cards

For each nonlinearity of the form $C1_j x_{V1_j}^{P1_j}$, a type-1 card is required. Each type-1 card specifies a $C1_j$, $V1_j$, and $P1_j$ in a E10.3, I3, E10.3 format.

Type-2 Cards

For each nonlinearity of the form $C2_j x_{V21_j}^{P21_j} x_{V22_j}^{P22_j}$, a type-2 card is required. Each type-2 card specifies a $C2_j$, $V21_j$, $P21_j$, $V22_j$, and $P22_j$ in a E10.3, I4, E10.3, I4, E10.3 format.

Type-3 Cards

For each nonlinearity of the form $C3_j x_{V31_j}^{P31_j} x_{V32_j}^{P32_j} x_{V33_j}^{P33_j}$, a type-3 card is required. Each type-3 card specifies a $C3_j$, $V31_j$, $P31_j$, $V32_j$, $P32_j$, $V33_j$, and $P33_j$ in a E10.3, I4, E10.3, I4, E10.3, I4, E10.3 format.

Type-4 Cards

For each time lagged term, a type-4 card is required. Each type-4 card specifies a $C4_j$, $V4_j$, and a time lag τ_j in a E10.3, I4, E10.3 format.

Forcing Function Cards

These cards are used to input forcing function vector $\bar{S}$. The elements of this vector, s_i's, are specified in a 8E10.3 format.

Initial Condition Cards

These cards are to input the initial conditions. The initial conditions are specified in a 8E10.3 format.

At the beginning of each time step, the subroutine COEFF is called in order to update the values of time-varying coefficients. The user must supply his own COEFF subroutine.

II. LISTING OF THE CODE

A listing of the computer code ESNDE is presented on the following pages.

```
C     ESNDE---WRITTEN FOR IBM-360
C           THIS PROGRAM SOLVES SETS OF COUPLED FIRST ORDER
C     DIFFERENTIAL EQUATIONS WITH PURE TIME DELAYS AND
C     NONLINEARITIES WHICH ARE PRODUCTS OF FRACTIONAL POWERS
C     OF UP TO THREE DEPENDENT VARIABLES.
C     THE  PROGRAM ASSUMES AN EXPONENTIAL DEPENDENCE OF THE
C     DEPENDENT VARIABLES OVER A TIME STEP AND ITERATIVELY
C     SOLVES FOR THE EXPONENTIAL DEPENDENCE.
C     TIME VARING COEFFICIENTS MAY BE INCLUDED BY CHANGING
C     THE VALUES OF THE COEFFICIENTS AT THE BEGINNING OF
C     EACH TIME STEP.  THE USER MUST WRITE HIS OWN C-O-E-F-F
C     SUBROUTINE TO VARY THE COEFFICIENTS.
      COMMON/MATRIX/X(30),XT(30),LMAT(30,30),D(30),CONSOR(30
     1),ITYPE1(30),ITYPE2(30),ITYPE3(30),ITYPE4(30),COEF1(30
     2),COEF2(30),COEF3(30),COEF4(30),VAR1(30),POW1(30),
     3VAR21(30),POW21(30),VAR22(30),POW22(30),VAR31(30),
     4POW31(30),VAR32(30),POW32(30),VAR33(30),POW33(30),
     5VAR4(30),TAU(30)
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      COMMON/OMEG/OMEGA(30),H
      DIMENSION TITLE(20)
      REAL LMAT
      INTEGER VAR1,VAR21,VAR22,VAR31,VAR32,VAR33,VAR4
      INTEGER  TYPE1,TYPE2,TYPE3,TYPE4
      DATA TYPE1,TYPE2,TYPE3/0,0,0/
      TYPE4=0
      TYPE1=0
      TYPE2=0
      TYPE3=0
      TYPE4=0
      READ(5,800) (TITLE(I),I=1,20)
      WRITE(6,801) (TITLE(I),I=1,20)
      READ(5,900) N,NOUTIT
C     N=NUMBER OF DIFFERENTIAL EQUATIONS
C     NOUTIT=MAXIMUM NUMBER OF OUTER ITERATIONS
      READ(5,901) TO,DELTAT,TMAX,HMIN,CONVLR,CONVFR,CONVCH
C     TO=INITIAL TIME
C     DELTAT=MAXIMUM TIME STEP SIZE
C     TMAX=FINAL TIME
C     HMIN=MINIMUM TIME STEP SIZE
C     CONVLR=LINEAR CONVERGENCE CONSTANT
C     CONVFR=FRACTIONAL CONVERGENCE CONSTANT
C     CONVCH=VALUE OF EXPONENTIAL PARAMETERS(OMEGA'S) ABOVE
C     WHICH FRACTIONAL CONVERGENCE TEST IS USED AND BELOW
C     WHICH LINEAR CONVERGENCE TEST IS USED
      WRITE(6,777) N,NOUTIT,TO,DELTAT,TMAX
      WRITE(6,778) HMIN,CONVLR,CONVFR,CONVCH
      READ(5,901)((LMAT(I,J),J=1,N),I=1,N)
C     LMAT IS N TIMES N DIMENSION LINEAR MATRIX
      WRITE(6,3000)
      WRITE(6,3001) ((LMAT(I,J),J=1,N),I=1,N)
```

```
      READ(5,901) (D(I),I=1,N)
C     D IS N DIMENSION DIAGONAL VECTOR
      WRITE(6,3002)
      WRITE(6,3001) (D(I),I=1,N)
      READ(5,900)(ITYPE1(I),ITYPE2(I),ITYPE3(I),ITYPE4(I),I=
     11,N)
C     FOR THE I-TH DIFFENTIAL EQUATION#
C     ITYPE1=NUMBER OF TERMS OF THE FORM C(X(I)**A)
C     ITYPE2=NUMBER OF TERMS OF THE FORM C(X(I)**A)(X(J)**B)
C     ITYPE3=NUMBER OF TERMS OF THE FORM
C     C(X(I)**A)(X(J)**B)(X(K)**D)
C     ITYPE4=NUMBER OF PURE TIME DELAYS OF THE FORM
C     C(X(I))@ TIME = T - TAU
      WRITE(6,3003)
      WRITE(6,3004) (ITYPE1(I),ITYPE2(I),ITYPE3(I),ITYPE4(I)
     1,I=1,N)
      DO 11 I=1,N
      TYPE1=TYPE1    +ITYPE1(I)
      TYPE2=TYPE2    +ITYPE2(I)
      TYPE3=TYPE3    +ITYPE3(I)
      TYPE4=TYPE4    +ITYPE4(I)
   11 CONTINUE
      IF(TYPE1.LE.0) GO TO 12
C     WE HAVE TERMS OF THE FORM C(X(I)**A)
      READ(5,904)(COEF1(I),VAR1(I),POW1(I),I=1,TYPE1)
C     COEF1'S CORRESPOND TO THE C'S
C     VAR1'S CORRESPOND TO THE I'S
C     POW1'S CORRESPOND TO THE A'S
   12 IF(TYPE2.LE.0) GO TO 13
C     WE HAVE TERMS OF THE FORM C(X(I)**A)(X(J)**B)
      READ(5,905)(COEF2(I),VAR21(I),POW21(I),VAR22(I),POW22(
     1I),I=1,TYPE2)
C     COEF2'S CORRESPOND TO THE C'S
C     VAR21'S CORRESPOND TO THE I'S
C     POW21'S CORRESPOND TO THE A'S
C     VAR22'S CORRESPOND TO THE J'S
C     POW22'S CORRESPOND TO THE B'S
   13 IF(TYPE3.LE.0) GO TO 14
C     WE HAVE TERMS OF THE FORM C(X(I)**A)(X(J)**B)(X(K)**D)
      READ(5,906)(COEF3(I),VAR31(I),POW31(I),VAR32(I),POW32(
     1I),VAR33(I),POW33(I),I=1,TYPE3)
C     COEF3'S CORRESPOND TO THE C'S
C     VAR31'S CORRESPOND TO THE I'S
C     POW31'S CORRESPOND TO THE A'S
C     VAR32'S CORRESPOND TO THE J'S
C     POW32'S CORRESPOND TO THE B'S
C     VAR33'S CORRESPOND TO THE K'S
C     POW33'S CORRESPOND TO THE D'S
   14 IF(TYPE4.LE.0) GO TO 15
C     WE HAVE TERMS OF THE FORM C(X(I))@ TIME=T-TAU
C     THAT IS # WE HAVE PURE TIME DELAYS
```

```
      READ(5,907)(COEF4(I),VAR4(I),TAU(I),I=1,TYPE4)
C     COEF4'S CORRESPOND TO THE C'S
C     VAR4'S CORRESPOND TO THE I'S
C     TAU'S CORRESPOND TO THE TIME DELAYS( TO THE TAU'S)
   15 CONTINUE
      WRITE(6,3005)
      IF(TYPE1.LE.0) GO TO 4000
      WRITE(6,3006) (COEF1(I),VAR1(I),POW1(I),I=1,TYPE1)
 4000 WRITE(6,3008)
      IF(TYPE2.LE.0) GO TO 4001
      WRITE(6,3009) (COEF2(I),VAR21(I),POW21(I),VAR22(I),POW
     122(I),I=1,TYPE2)
 4001 WRITE(6,3010)
      IF(TYPE3.LE.0) GO TO 4002
      WRITE(6,3011) (COEF3(I),VAR31(I),POW31(I),VAR32(I),POW
     132(I),VAR33(I),POW33(I),I=1,TYPE3)
 4002 WRITE(6,3012)
      IF(TYPE4.LE.0) GO TO 4003
      WRITE(6,3013) (COEF4(I),VAR4(I),TAU(I),I=1,TYPE4)
 4003 CONTINUE
      READ(5,901)(CONSOR(I),I=1,N)
C     CONSOR  IS N DIMENSION CONSTANT SOURCE VECTOR
      WRITE(6,3014)
      WRITE(6,3001) (CONSOR(I),I=1,N)
C     READ IN INITIAL CONDITIONS
      READ(5,901) (X(I),I=1,N)
C     INITIALIZE TIME LAG ARRAYS
      CALL LAG(0)
      T=TO
C     THE SUBROUTINE OMGA1 ESTIMATES THE FIRST VALUE OF
C     THE OMEGA'S
      CALL OMGA1
    1 CONTINUE
      WRITE(6,1000) T
      WRITE(6,3001) (X(I),I=1,N)
      WRITE(6,3001) (OMEGA(I),I=1,N)
      IF(T.GE.TMAX) GO TO 2000
C     UPDATE TIME LAGGED ARRAYS
      IF(T.NE.TO) CALL LAG(1)
C     THE SUBROUTINE COEFF IS USED TO UPDATE TIME VARING
C     COEFFICIENTS AT THE BEGINNING OF EACH TIME STEP
      CALL COEFF
      H=DELTAT
    3 CONTINUE
      ISTOP=1
      DO 4 II=1,NOUTIT
      IF(ISTOP.EQ.0) GO TO 8
    7 CONTINUE
C     FIND TIME LAGGED VARIABLES
      CALL LAG(2)
      IFLAG=0
```

```
      ISTEP1=0
      ISTEP2=0
      ISTEP3=0
      ISTEP4=0
      ISTOP=0
      DO 5 I=1,N
      ISTEP=0
C     SUBROUTINE EVALXT CALCULATES THE X'S AT TIME=T+H
    6 CALL EVALXT(I,ISTEP1,ISTEP2,ISTEP3,ISTEP4,IFLAG)
      IFLAG=0
      OMESTR=(ALOG(XT(I)/X(I)))/H
      IF(ABS(OMEGA(I)).GE.CONVCH) GO TO 50
      IF(ABS(OMESTR-OMEGA(I)).LE.CONVLR) GO TO 5
      GO TO 51
   50 IF(ABS((OMESTR-OMEGA(I))/OMEGA(I)).LE.CONVFR) GO TO 5
   51 CONTINUE
      ISTEP=ISTEP+1
      ISTOP=ISTOP+1
C     SUBROUTINE ITER ITERATES UPON THE OMEGA'S AND RETURNS
C     NEW ESTIMATES OF THE OMEGA'S
      CALL ITER(I,OMESTR,ISTEP)
      IF(H.LE.HMIN) WRITE(6,1002) H,I
      IF(H.LE.HMIN) CALL EXIT
      IFLAG=1
      IF(ISTEP.NE.4) GO TO 6
      GO TO 7
    5 CONTINUE
      IF(II.EQ.NOUTIT) WRITE(6,1009)
    4 CONTINUE
    8 CONTINUE
      T=T+H
      DO 2 I=1,N
    2 X(I)=XT(I)
      GO TO 1
 2000 CONTINUE
  777 FORMAT(1H0,'N=',I3,3X,'NOUTIT=',I2,3X,'TO=',E12.4,3X,
     1'DELTAT=',E12.4,3X,'TMAX=',E12.4)
  778 FORMAT(1H0,'HMIN=',E12.4,3X,'CONVLR=',E12.4,3X,
     1'CONVFR=',E12.4,3X,'CONVCH=',E12.4)
  800 FORMAT(20A4)
  801 FORMAT(1H1,15X,20A4)
  900 FORMAT(20I4)
  901 FORMAT(8E10.3)
  904 FORMAT(E10.3,I3,E10.3)
  905 FORMAT(E10.3,I3,E10.3,I3,E10.3)
  906 FORMAT(E10.3,I3,E10.3,I3,E10.3,I3,E10.3)
  907 FORMAT(E10.3,I3,E10.3)
 1000 FORMAT(1H0,'X(I) AND OMEGA(I) FOR TIME=',F13.4,1X,
     1'ARE')
 1002 FORMAT(1H1,'CALCULATION TERMINATED H LE HMIN  H=',
     1E10.3,'I=',I4)
```

```
1009 FORMAT(1H0,'PROBLEM MAY NOT HAVE CONVERGED  AT THE ',
    1'FOLLOWING TIME STEP')
3000 FORMAT(1H1,'THE LINEAR MATRIX IS BY ROWS')
3001 FORMAT(1H ,10E12.4)
3002 FORMAT(1H1,'THE DIAGONAL ELEMENTS ARE')
3003 FORMAT(1H1,'ITYPE1,ITYPE2,ITYPE3,ITYPE4   ARE')
3004 FORMAT(1H ,7X,4I4,8X,4I4,8X,4I4,8X,4I4,8X,4I4)
3005 FORMAT(1H1,'COEF1,VAR1,POW1 ARE')
3006 FORMAT(1H ,E12.4,I4,E12.4)
3008 FORMAT(1H0,'COEF2,VAR21,POW21,VAR22,POW22 ARE')
3009 FORMAT(1H ,E12.4,I4,E12.4,I4,E12.4)
3010 FORMAT(1H0,'COEF3,VAR31,POW31,VAR32,POW32,VAR33,POW33'
    1,' ARE')
3011 FORMAT(1H ,E12.4,I4,E12.4,I4,E12.4,I4,E12.4)
3012 FORMAT(1H0,'COEF4,VAR4,TAU ARE')
3013 FORMAT(1H ,E12.4,I4,E12.4)
3014 FORMAT(1H1,'THE CONSTANT SOURCE TERMS ARE')
     CALL EXIT
     END
```

```
      SUBROUTINE ITER(I,OMESTR,ISTEP)
      COMMON/OMEG/OMEGA(30),H
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      INTEGER TYPE4
      DATA EPL,EPER,BMX,HRD/.001,1.0,0.75,0.9/
C     HRD=FACTOR BY WHICH TIME STEP IS REDUCED IF A
C     CONVERGED OMEGA CAN NOT BE FOUND IN THREE ESTIMATES
C     THE TIME STEP IS REDUCE
      EPLEPR=EPL*EPER
C     THE PRODUCT EPL*EPER SHOULD ALWAYS BE LESS THAN OR
C     EQUAL TO CONVLR
      IF(ISTEP.GT.1) GO TO 1
C     BELOW IS SCHEME 1
      ER1=OMESTR-OMEGA(I)
      OM1=OMEGA(I)
      OM2=OMESTR
      OMEGA(I)=OM2
      RETURN
    1 IF(ISTEP.GT.2) GO TO 2
C     BELOW IS SCHEME 2
      OM3=OMESTR
      ER2=OM3-OM2
      IF(ABS(ER2).GE.ABS(ER1)) GO TO 7
      IF(ABS(ER2).GE.EPLEPR) GO TO 7
      OMEGA(I)=OM3
      RETURN
    7 B=1.0-ER1/ER2
      IF(ER1*ER2.LT.0.0) GO TO 8
      IF(ABS(B).GE.BMX) GO TO 8
      IF(B.GT.0.0) GO TO 9
      B=-BMX
      GO TO 10
    9 B=BMX
   10 IF(ABS(ER1).GE.ABS(ER2)) GO TO 8
      A=ER1*(1.0-B)
      GO TO 11
    8 A=ER2-B*(OM3-OM1)
   11 OM3=OM1-A/B
      OMEGA(I)=OM3
      RETURN
    2 IF (ISTEP.GT.3) GO TO 3
C     BELOW IS SCHEME 2
      OM4=OMESTR
      ER3=OM4-OM3
      IF(ABS(ER3).GE.ABS(ER2)) GO TO 4
      IF(ABS(ER3).GE.EPLEPR) GO TO 4
      OMEGA(I)=OM4
      RETURN
    4 B=1.0-ER2/ER3
      IF(ER2*ER3.LT.0.0) GO TO 5
      IF(ABS(B).GE.BMX) GO TO 5
```

```
      IF(B.GT.0.0) GO TO 6
      B=-BMX
      GO TO 12
    6 B=BMX
   12 IF(ABS(ER2).GE.ABS(ER3)) GO TO  5
      A=ER2*(1.0-B)
      GO TO 13
    5 A=ER3-B*(OM4-OM2)
   13 OM4=OM2-A/B
      OMEGA(I)=OM4
      RETURN
C     TIME STEP IS REDUCED HERE
    3 H=H*HRD
      WRITE(6,20) H,I
   20 FORMAT(1H ,'TIME STEP REDUCED TO ',E12.6,'FOR I=',I4)
      RETURN
      END
```

```
      SUBROUTINE EVALXT(I,ISTEP1,ISTEP2,ISTEP3,ISTEP4,IFLAG)
      COMMON/MATRIX/X(30),XT(30),LMAT(30,30),D(30),CONSOR(30
     1),ITYPE1(30),ITYPE2(30),ITYPE3(30),ITYPE4(30),COEF1(30
     2),COEF2(30),COEF3(30),COEF4(30),VAR1(30),POW1(30),
     3VAR21(30),POW21(30),VAR22(30),POW22(30),VAR31(30),
     4POW31(30),VAR32(30),POW32(30),VAR33(30),POW33(30),
     5VAR4(30),TAU(30)
      COMMON/OMEG/OMEGA(30),H
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      INTEGER TYPE4
      REAL LMAT
      INTEGER VAR1,VAR21,VAR22,VAR31,VAR32,VAR33,VAR4
      DATA DEMMIN/.001/
      ST=CONSOR(I)
      XT(I)=X(I)*EXP(D(I)*H)
      IF(ABS(D(I)).LE.DEMMIN)GO TO 1
      XT(I)=XT(I)+(EXP(D(I)*H)-1.0)*ST/D(I)
      GO TO 2
C     A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C     (EXP(D)-1.0)/D  BECAUSE D IS VERY SMALL.
C     THIS SERIES EXPANSION IS MADE WHENEVER D  IS
C     LESS THAN OR EQUAL TO DEMMIN.
    1 XT(I)=XT(I)+ST*H*(1.0+D(I)*H/2.0+((D(I)*H)**2)/6.0)
    2 CONTINUE
      DO 3 K=1,N
      XK=OMEGA(K)-D(I)
      IF(ABS(XK).LE.DEMMIN ) GO TO 4
      XT(I)=XT(I)+ LMAT(I,K)*EXP(D(I)*H)*(EXP( XK*H)-1.0)*X(
     1K)/XK
      GO TO 3
C     A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C     (EXP(XK)-1.0)/XK BECAUSE XK IS VERY SMALL.
C     THIS SERIES EXPANSION IS MADE WHENEVER XK IS
C     LESS THAN OR EQUAL TO DEMMIN.
    4 XT(I)=XT(I)+LMAT(I,K)*EXP(D(I)*H)*X(K)*H*(1.0+XK*H/2.0
     1+((XK*H)**2)/6.0)
    3 CONTINUE
      IF(ITYPE1(I).LE.0)GO TO 20
C     WE HAVE TERMS OF THE FORM C(X(I)**A)
      NO=ITYPE1(I)
      IF(IFLAG.EQ.1) ISTEP1=ISTEP1-NO
      DO 11 K=1,NO
      ISTEP1=ISTEP1+1
      XK=POW1(ISTEP1)*OMEGA(VAR1(ISTEP1))-D(I)
      IF(ABS(XK).LE.DEMMIN)GO TO 12
      XT(I)=XT(I)+COEF1(ISTEP1)*EXP(D(I)*H)*(X(VAR1(ISTEP1))
     1**POW1(ISTEP1))*(EXP(XK*H)-1.0)/XK
      GO TO 11
C     A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C     (EXP(XK)-1.0)/XK BECAUSE XK IS VERY SMALL.
C     THIS SERIES EXPANSION IS MADE WHENEVER XK IS
```

```
C     LESS THAN OR EQUAL TO DEMMIN.
   12 CONTINUE
      XT(I)=XT(I)+COEF1(ISTEP1)*EXP(D(I)*H)*(X(VAR1(ISTEP1))
     1**POW1(ISTEP1))*H*(1.0+XK*H/2.0+((XK*H)**2)/6.0)
   11 CONTINUE
   20 IF(ITYPE2(I).LE.0)GO TO 30
C     WE HAVE TERMS OF THE FORM C(X(I)**A)(X(J)**B)
      NO=ITYPE2(I)
      IF(IFLAG.EQ.1) ISTEP2=ISTEP2-NO
      DO 21 K=1,NO
      ISTEP2=ISTEP2+1
      XK=POW21(ISTEP2)*OMEGA(VAR21(ISTEP2))+POW22(ISTEP2)*OM
     1EGA(VAR22(ISTEP2))-D(I)
      IF(ABS(XK).LE.DEMMIN)GO TO 22
      XT(I)=XT(I)+COEF2(ISTEP2)*EXP(D(I)*H)*(X(VAR21(ISTEP2)
     1)**POW21(ISTEP2))*(X(VAR22(ISTEP2))**POW22(ISTEP2))*(E
     2XP(XK*H)-1.0)/XK
      GO TO 21
C     A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C     (EXP(XK)-1.0)/XK BECAUSE XK IS VERY SMALL.
C     THIS SERIES EXPANSION IS MADE WHENEVER XK IS
C     LESS THAN OR EQUAL TO DEMMIN.
   22 CONTINUE
      XT(I)=XT(I)+COEF2(ISTEP2)*EXP(D(I)*H)*(X(VAR21(ISTEP2)
     1)**POW21(ISTEP2))*(X(VAR22(ISTEP2))**POW22(ISTEP2))*H*
     2(1.0+XK*H/2.0+((XK*H)**2)/6.0)
   21 CONTINUE
   30 IF(ITYPE3(I).LE.0)GO TO 40
C     WE HAVE TERMS OF THE FORM C(X(I)**A)(X(J)**B)(X(K)**D)
      NO=ITYPE3(I)
      IF(IFLAG.EQ.1) ISTEP3=ISTEP3-NO
      DO 31 K=1,NO
      ISTEP3=ISTEP3+1
      XK=POW31(ISTEP3)*OMEGA(VAR31(ISTEP3))+POW32(ISTEP3)*OM
     1EGA(VAR32(ISTEP3))+POW33(ISTEP3)*OMEGA(VAR33(ISTEP3))-
     2D(I)
      IF(ABS(XK).LE.DEMMIN)GO TO 32
      XT(I)=XT(I)+COEF3(ISTEP3)*EXP(D(I)*H)*(X(VAR31(ISTEP3)
     1)**POW31(ISTEP3))*(X(VAR32(ISTEP3))**POW32(ISTEP3))*(X
     2(VAR33(ISTEP3))**POW33(ISTEP3))*(EXP(XK*H)-1.0)/XK
      GO TO 31
C     A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C     (EXP(XK)-1.0)/XK BECAUSE XK IS VERY SMALL.
C     THIS SERIES EXPANSION IS MADE WHENEVER XK IS
C     LESS THAN OR EQUAL TO DEMMIN.
   32 CONTINUE
      XT(I)=XT(I)+COEF3(ISTEP3)*EXP(D(I)*H)*(X(VAR31(ISTEP3)
     1)**POW31(ISTEP3))*(X(VAR32(ISTEP3))**POW32(ISTEP3))*(X
     2(VAR33(ISTEP3))**POW33(ISTEP3))*H*(1.0+XK*H/2.0+((XK*H
     3)**2)/6.0)
   31 CONTINUE
```

```
   40 IF(ITYPE4(I).LE.0)GO TO 50
C        WE HAVE TERMS OF THE FORM C(X(I))@ TIME=T-TAU
C        THAT IS # WE HAVE PURE TIME DELAYS
         NO=ITYPE4(I)
         IF(IFLAG.EQ.1) ISTEP4=ISTEP4-NO
         DO 41 K=1,NO
         ISTEP4=ISTEP4+1
         IF(ABS(D(I)).LE.DEMMIN) GO TO 42
         XT(I)=XT(I)+COEF4(ISTEP4)*(EXP(D(I)*H)-1.0)*XL(ISTEP4)
        2/D(I)
         GO TO 41
C        A SERIES EXPANSION IS TO BE MADE FOR THE TERM
C        (EXP(D)-1.0)/D  BECAUSE D IS VERY SMALL.
C        THIS SERIES EXPANSION IS MADE WHENEVER D  IS
C        LESS THAN OR EQUAL TO DEMMIN.
   42 CONTINUE
         XT(I)=XT(I)+COEF4(ISTEP4)*H*(1.0+D(I)*H/2.0+((D(I)*H)*
        1*2)/6.0)*XL(ISTEP4)
   41 CONTINUE
   50 CONTINUE
         RETURN
         END
```

```
      SUBROUTINE OMGA1
      COMMON/MATRIX/X(30),XT(30),LMAT(30,30),D(30),CONSOR(30
     1),ITYPE1(30),ITYPE2(30),ITYPE3(30),ITYPE4(30),COEF1(30
     2),COEF2(30),COEF3(30),COEF4(30),VAR1(30),POW1(30),
     3VAR21(30),POW21(30),VAR22(30),POW22(30),VAR31(30),
     4POW31(30),VAR32(30),POW32(30),VAR33(30),POW33(30),
     5VAR4(30),TAU(30)
      COMMON/OMEG/OMEGA(30),H
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      REAL LMAT
      INTEGER VAR1,VAR21,VAR22,VAR31,VAR32,VAR33,VAR4
      INTEGER TYPE4
C     THIS SUBROUTINE ESTIMATES THE FIRST VALUE OF
C     THE OMEGA'S
      ISTEP1=0
      ISTEP2=0
      ISTEP3=0
      ISTEP4=0
      DO 11 I=1,N
      XT(I)=CONSOR(I)
      XT(I)=XT(I)+D(I)*X(I)
      DO 2 J=1,N
    2 XT(I)=LMAT(I,J)*X(J)+XT(I)
      IF(ITYPE1(I).LE.0) GO TO 4
      NO=ITYPE1(I)
      DO 3 K=1,NO
      ISTEP1=ISTEP1+1
    3 XT(I)=XT(I)+COEF1(ISTEP1)*(X(VAR1(ISTEP1))**POW1(ISTEP
     11))
    4 IF(ITYPE2(I).LE.0) GO TO 6
      NO=ITYPE2(I)
      DO 5 K=1,NO
      ISTEP2=ISTEP2+1
    5 XT(I)=XT(I)+COEF2(ISTEP2)*(X(VAR21(ISTEP2))**POW21(IST
     1EP2))*(X(VAR22(ISTEP2))**POW22(ISTEP2))
    6 IF(ITYPE3(I).LE.0) GO TO 8
      NO=ITYPE3(I)
      DO 7 K=1,NO
      ISTEP3=ISTEP3+1
    7 XT(I)=XT(I)+COEF3(ISTEP3)*(X(VAR31(ISTEP3))**POW31(IST
     1EP3))*(X(VAR32(ISTEP3))**POW32(ISTEP3))*(X(VAR33(ISTEP
     23))**POW33(ISTEP3))
    8 IF(ITYPE4(I).LE.0) GO TO 10
      NO=ITYPE4(I)
      DO 9 K=1,NO
      ISTEP4=ISTEP4+1
    9 XT(I)=XT(I)+COEF4(ISTEP4)*X(VAR4(ISTEP4))
   10 CONTINUE
   11 CONTINUE
      WRITE(6,100)
      WRITE(6,101) (XT(I),I=1,N)
```

```
      DO 1 I=1,N
      IF(ABS(XT(I)).LE.1.0E-03) OMEGA(I)=0.0
      IF(ABS(XT(I)).LE.1.0E-03) GO TO 1
      OMEGA(I)=(ALOG(1.0+DELTAT*XT(I)/X(I)))/DELTAT
    1 CONTINUE
  100 FORMAT(1H1,'  DXDT(I) INITIALLY IS  ')
  101 FORMAT(1H ,10E12.4)
      RETURN
      END
```

```
      SUBROUTINE LAG(LGCODE)
      COMMON/MATRIX/X(30),XT(30),LMAT(30,30),D(30),CONSOR(30
     1),ITYPE1(30),ITYPE2(30),ITYPE3(30),ITYPE4(30),COEF1(30
     2),COEF2(30),COEF3(30),COEF4(30),VAR1(30),POW1(30),
     3VAR21(30),POW21(30),VAR22(30),POW22(30),VAR31(30),
     4POW31(30),VAR32(30),POW32(30),VAR33(30),POW33(30),
     5VAR4(30),TAU(30)
      COMMON/OMEG/OMEGA(30),H
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      DIMENSION XLAG(10,100),TLAG(100)
      DIMENSION XXLAG(10,100),TTLAG(100)
      REAL LMAT
      INTEGER VAR1,VAR21,VAR22,VAR31,VAR32,VAR33,VAR4
      INTEGER TYPE4
C     THIS SUBROUTINE STORES AND FINDS THE LAGGED VARIABLES.
C     IF ANY TIME LAGS EXCEED APPROXIMATELY 80*DELTAT THEN
C     NL SHOULD BE INCREASED AND THE DIMENSIONS OF 100 IN
C     THIS SUBROUTINE SHOULD BE INCREASED LIKEWISE.
C     WHEN LGCODE=0, THE INITIAL LAGGED TERMS ARE SET EQUAL
C     TO THE INITIAL CONDITIONS AND LAGGED ARRAYS ARE
C     PRODUCED.
C     WHEN LGCODE=1, THE LAGGED ARRAYS ARE UPDATED.
C     WHEN LGCODE=2, THE LAGGED VARIABLES ARE FOUND AND
C     RETURNED TO THE   MAIN CODE.
C     ALL TAU'S MUST BE GREATER THAN DELTAT.
      IF(TYPE4.EQ.0) RETURN
      NL=100
      IF(LGCODE.GT.0) GO TO 1
      TLAG(1)=TO
      DO 2 K=1,TYPE4
    2 XLAG(K,1)=X(VAR4(K))
      DO 3 KK=2,N
      TLAG(KK)=TLAG(KK-1)-DELTAT
      DO 3 K=1,TYPE4
    3 XLAG(K,KK)=X(VAR4(K))
      RETURN
    1 IF(LGCODE.GT.1) GO TO 5
      NN=NL-1
      DO 12 KK=2,NL
      TTLAG(KK)=TLAG(KK-1)
      DO 12 K=1,TYPE4
   12 XXLAG(K,KK)=XLAG(K,KK-1)
      DO 6 KK=2,NL
      TLAG(KK)=TTLAG(KK)
      DO 6 K=1,TYPE4
    6 XLAG(K,KK)=XXLAG(K,KK)
      TLAG(1)=T
      DO 7 K=1,TYPE4
    7 XLAG(K,1)=X(VAR4(K))
      RETURN
    5 CONTINUE
```

```
      DO 11 IX=1,TYPE4
      TC=T+H/2.0-TAU(IX)
      DO 8 I=1,NL
      IF(TC.GE.TLAG(I)) L=I
      IF(TC.GE.TLAG(I)) GO TO 9
    8 CONTINUE
      WRITE(6,100)     VAR4(IX)
  100 FORMAT(1H ,'****** CAN NOT FIND LAGGED VARIABLE =',I4)
      CALL EXIT
    9 IF(TC.NE.TLAG(L)) GO TO 10
      XL(IX)=XLAG(IX,L)
      GO TO 11
   10 XL(IX)=XLAG(IX,L)+(XLAG(IX,L-1)-XLAG(IX,L))*(TC-TLAG(L
     1))/(TLAG(L-1)-TLAG(L))
   11 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COEFF
      COMMON/MATRIX/X(30),XT(30),LMAT(30,30),D(30),CONSOR(30
     1),ITYPE1(30),ITYPE2(30),ITYPE3(30),ITYPE4(30),COEF1(30
     2),COEF2(30),COEF3(30),COEF4(30),VAR1(30),POW1(30),
     3VAR21(30),POW21(30),VAR22(30),POW22(30),VAR31(30),
     4POW31(30),VAR32(30),POW32(30),VAR33(30),POW33(30),
     5VAR4(30),TAU(30)
      COMMON/OMEG/OMEGA(30),H
      COMMON/LUG/TYPE4,DELTAT,XL(10),TO,T,N
      REAL LMAT
      INTEGER VAR1,VAR21,VAR22,VAR31,VAR32,VAR33,VAR4
C     THIS SUBROUTINE MAY BE USED TO CHANGE ANY
C     COEFFICIENTS AT THE  BEGINNING OF EACH TIME STEP, ALSO
C     PERTURBATION MAY BE INTRODUCED INTO THE SYSTEM IN THIS
C     SUBROUTINE
      RETURN
      END
```

VITA

Maurice Manning Anderson, Jr. was born in Nashville, Tennessee on September 15, 1945. His parents are Maurice Manning Anderson, Sr. and the former Etheleen Mae Wilson. Bill, as he is known to his friends, attended elementary and secondary schools in Gallatin, Centerville, and Columbia, Tennessee. After graduating from Columbia Military Academy in 1963, he entered The University of Tennessee and was graduated with a Bachelor of Science in Nuclear Engineering in 1967.

Bill is married to the former Evalia Jean Rogers of Kingsport, Tennessee and is the father of Maurice Manning Anderson, III.