27 eip
W-7405

THE NUMERICAL SOLUTION OF A MODEL FOR REACTOR SAFETY ANALYSIS

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Martin J. Strasburger
and Harold D. Skank

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

MASTER

Date Transmitted: August 1976

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405-eng-82

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

THE NUMERICAL SOLUTION OF A MODEL
FOR NUCLEAR REACTOR SAFETY ANALYSIS

By

Martin J. Strasburger and Harold D. Skank

Ames Laboratory, ERDA

Iowa State University

Ames, Iowa 50011

Date Transmitted: August 1976

PREPARED FOR THE U.S. ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION UNDER CONTRACT NO. W-7405-eng-82

100-3974

—NOTICE—

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche \$2.25
Printed Copy \$4.50

Printed copy \$4.50

TABLE OF CONTENTS

	PAGE
ABSTRACT	iv
I. INTRODUCTION	1
II. PROBLEM STATEMENT	4
III. PROBLEM SOLUTION	12
A. AN OVERVIEW OF THE GEAR PACKAGE	12
B. DESCRIPTION OF A CALL TO INTERFACE SUBROUTINE DRIVE	14
C. THE IMPLEMENTATION OF $k_{ex}(t)$ AND THE "TRIP" MECHANISM	16
D. A NOTE ON "STIFFNESS"	19
IV. RESULTS OF THE COMPUTATION	21
APPENDIX A: DERIVATION OF CIRCUIT EQUATIONS	36
APPENDIX B: PROGRAM LISTING	47
ACKNOWLEDGEMENTS	58
REFERENCES	59

ABSTRACT

A mathematical model for the Ames Laboratory Research Reactor and proposed safety device electronics is presented, method of solution discussed, and results graphed and tabulated. The model is comprised of a nonlinear system of nineteen differential equations.

A derivation of the three ordinary differential equations which describe the safety electronics unique to this project is given. Three "accident" postulates (slow rod withdrawal, "step insertion" and "ramp insertion") are specified by initial conditions and other parameters. Solution of the model was obtained by using a modified version of "GEAR: Ordinary Differential Equation System Solver" from Lawrence Livermore Laboratory. The application of GEAR to this problem, and a note on "stiffness" of the problem is given. Results obtained from the solution of this model indicate the range of performance for the proposed safety electronics, which appears to be adequate. A FORTRAN computer program listing is given.

Distribution List

Chicago Patent Office	1
USERDA-TIC	27
Ames Laboratory Library	5
Martin J. Strasburger	15
Harold D. Skank	20
Robert J. Lambert	4
	—
Total	72

I. Introduction:

In the safety analyses of nuclear reactors one attempts to postulate credible phenomena giving rise to potentially hazardous reactor conditions (excessive neutron flux, excessive core temperature, etc.), then calculate the response of safety mechanisms designed to mitigate those consequences. By using such a technique it is possible to design a reactor core/safety instrumentation complex having known bounds for safe operation and known responses to unsafe conditions. This report developed out of the safety analysis for a new instrumentation system proposed for the Ames Laboratory Research Reactor (ALRR) at Ames, Iowa.

ALRR is a deuterium oxide (heavy water) moderated, heterogeneous core, research reactor designed to operate at a thermal power level of five megawatts. It uses 24, 95% enriched ^{235}U fuel elements and has six magnetic release, gravity inserted control rods whose vertical position relative to the reactor core may be controlled to effect control of total reactor power¹. The proposed instrumentation package includes a period trip circuit designed to monitor the reactor's period and shut the reactor down if that period becomes less than 3 seconds (see Appendix A.4). The numerical solution of the mathematical model of the interaction dynamics of this reactor core/control rod/instrumentation system complex is the subject of this report.

For the purposes at hand the ALRR core is adequately modeled by a set of sixteen coupled, linear differential equations with time varying coefficients describing the neutron flux as a function of time and reactivity (the control parameter for the reactor). These equations constitute the first sixteen of a full set of nineteen equations. The other three equations (derived in Appendix A) describe the instrumentation circuits that monitor the reactor neutron flux and trigger the release of the control rods to initiate a safe shutdown. Of these three equations, two are linear differential equations with constant coefficients while the third is a nonlinear differential equation.

Three "accident" postulates are used to determine the operating characteristics of the total system. The first of these supposes an uncontrolled rod withdrawal from a very low initial neutron flux level. This constitutes a potentially dangerous situation in that at low flux levels the circuit measuring the rate of reactor power increase functions very slowly. If reactor parameters were improperly chosen, it might be possible for the reactor to "outrun" the safety circuit with resulting damage to the reactor core. The problem is to determine if such conditions exist.

The second accident postulate supposes that the reactor is operating at full power initially, and that a fixed, step insertion of reactivity occurs. (Such a condition might be approached by sudden flooding of the central thimble.) Here, the safety circuit should detect the sudden

reactor power increase and initiate a shutdown. The problem is to determine the maximum tolerable reactivity insertion which results in safe neutron flux levels. This calculated value may then be compared to known reactor parameters to determine the effectiveness of the safety system.

The third postulate resembles the second, except that reactivity is inserted at a fixed, high rate instead of as a step. The maximum permissible rate of insertion is determined for safe operation, and that calculated rate of insertion may be compared back to known reactor parameters to determine the effectiveness of the safety system.

II. Problem Statement

The mathematical model for the reactor core/safety instrumentation complex will be presented. The well known "point reactor" model is used to represent the dynamics of the reactor core. The derivation of the ordinary differential equations (o.d.e.'s) for this model may be found in almost any text on nuclear reactors, such as Hetrick.² The model is only valid for "small" departures from criticality; other limitations include an inability to describe dynamic effects which are spatially dependent. Nevertheless, it is a reasonable model for this study.

The three o.d.e.'s used to describe the safety device are derived in Appendix A. Only the equations will be given here; the quantities represented, their meanings and units of measurement are contained in Appendix A.

The quantities to be presented in the point reactor model, and their units of measurement follow.

Independent variable: t - time (sec)

Dependent variables: $n(t)$ - neutron flux or density ($\frac{\text{neutron}}{\text{cm}^2 \cdot \text{sec}}$)
 $c_i(t)$ - density of precursor neutrons for
the i^{th} delayed neutron group ($\frac{\text{neutron}}{\text{cm}^2 \cdot \text{sec}}$)

Control function: $k_{\text{ex}}(t)$ - excess reactivity (dimensionless)

Parameter: S - source ($\frac{\text{neutron}}{\text{cm}^2 \cdot \text{sec}^2}$)

Constants (numerical values may be found in the BLOCK DATA subprogram of Appendix B):

β_i - delayed neutron fraction for the i^{th} delayed neutron group (dimensionless)

λ_i - decay constant for the i^{th} delayed neutron group (sec^{-1})

ℓ - effective prompt neutron lifetime (sec)
(depends on moderator material and core geometry.)

$$\beta_{\Sigma} \equiv \sum_{i=1}^{15} \beta_i$$

The point reactor model is given by:

$$\frac{dn(t)}{dt} = \left\{ \frac{k_{\text{ex}}(t) - \beta_{\Sigma} [1 + k_{\text{ex}}(t)]}{\ell} \right\} \cdot n(t) - \sum_{i=1}^{15} \lambda_i c_i(t) + S \quad (2.1)$$

$$\frac{dc_i(t)}{dt} = \frac{\beta_i}{\ell} [1 + k_{\text{ex}}(t)] \cdot n(t) + \lambda_i c_i(t),$$

$$i = 1, \dots, 15 \quad (2.2)$$

The safety device equations from Appendix A are:

$$\frac{de_0(t)}{dt} = -\omega_0 e_0(t) - \omega_0 A_{\text{DC}} e_I(t) \quad (2.3)$$

$$\frac{de_I(t)}{dt} = \frac{de_0(t)}{dt} - \frac{1}{C} \cdot \frac{10^{-6}}{3} n(t) + \frac{1}{C} I_C \exp(\alpha \cdot e_0(t)) \quad (2.4)$$

$$\frac{de_2(t)}{dt} = 1152 \frac{de_0(t)}{dt} - 20e_2(t) \quad (2.5)$$

The problem is to solve this system (for the neutron flux $n(t)$, $c_1(t)$, ..., $c_{15}(t)$, $e_0(t)$, $e_I(t)$, and $e_2(t)$) for each of the three "accident" postulates. An "accident" postulate is specified by the initial conditions for the system, the control function $k_{ex}(t)$, and the source S . It is desired to control the reactor so that $n(t)$ does not exceed 150% of full power.

The initial conditions for all three potentially hazardous situations are static ones. That is, they are the conditions obtained from the system of equations (2.1)-(2.5) by equating all first order derivatives to zero at $t = 0$. One obtains:

$$n(0) \cdot k_{ex}(0) = -S\lambda \quad (2.6)$$

(The $k_{ex}(0) = 0$ cases are explained below.)

$$c_i(0) = \frac{-\beta_i}{\lambda_i \ell} [1 + k_{ex}(0)] \cdot n(0) \quad (2.7)$$

$$e_0(0) = \frac{\ln\left[\frac{10^{-6} n(0)}{3 I_G}\right]}{\alpha} \quad (2.8)$$

$$e_I(0) = -\frac{e_0(0)}{A_{DC}} \quad (2.9)$$

$$e_2(0) = 0 \quad (2.10)$$

Excess reactivity $k_{ex}(t)$ is related to reactivity $k(t)$ by $k_{ex}(t) = k(t) - 1$. Criticality of the reactor is described by $k(t) = 1$, i.e., $k_{ex}(t) = 0$. The reactor is controlled by withdrawing or inserting

the control rods, which changes $k_{ex}(t)$. For this model, $k_{ex}(t)$ is at first a part of the "accident" postulate; subsequently, if a signal is given to drop the control rods for "shutdown" of the reactor, $k_{ex}(t)$ is modified as described below.

The signal for "shutdown" is indicated by one of the following conditions:

1. The "period trip" occurs; that is, the quantity $e_2(t)$ exceeds $8 \frac{1}{3}$ volts.
2. The "power trip" occurs; that is, the quantity $n(t)$ exceeds 125. (125% of full power.)

After the signal is given at some time t_a , there is a 350 millisecond electro-mechanical worst case delay time until the rods begin to drop, at time t_b . The drop is modeled by a change in excess reactivity corresponding to a straight line (approximating the real situation) passing through the point $(t_b, k_{ex}(t_b))$ and with slope $-.702/\text{sec}$. This linear decrease continues until some time t_c , at which $k_{ex}(t_c) = -.15$, corresponding to a state where the rods are fully inserted. Thereafter, $k_{ex}(t) = -.15$, $t > t_c$.

Note that the "period trip" and "power trip" conditions involve the dependent variables $n(t)$ and $e_2(t)$, which are to be determined; subsequently, upon "shutdown", $k_{ex}(t)$ will be changed, which will in turn modify the o.d.e.'s which determine $n(t)$ and $e_2(t)$.

We now may formulate the three "accident" postulates. The reader may wish to refer to the description of these given in the Introduction.

1. Rod withdrawal problem

Initial conditions are the static ones given by (2.6)-(2.10), with $k_{\text{ex}}(0) = -.15$. Hence, from (2.6), $n(0) = -S\ell/k_{\text{ex}}(0)$. We wish to solve the problem for various values of the parameter S . Since $n(t)$ is one of the main quantities of interest, we choose values of S so that $n(0)$ is "nice"; choose S so $n(0) = 3 \times 10^{-5}$, as well as 5, 10, 50 and 100 times this value.

The excess reactivity function is a slow ramp, corresponding to "worst case" rod withdrawal:

$$k_{\text{ex}}(t) = -.15 + (8.5 \times 10^{-4}) \cdot t, \quad 0 \leq t \leq t_b$$

A qualitative graph for the rod withdrawal problem appears in Fig. 2.1.

2. Step insertion problem

Initial conditions are the static ones given by (2.6)-(2.10), with $n(0) = 100$ (100% of full power) and $k_{\text{ex}}(0) = 0$. From (2.6), we must have $S = 0$.

The excess reactivity function is the discontinuous step function:

$$k_{\text{ex}}(t) = \begin{cases} 0, & t = 0 \\ .0002 \cdot M, & 0 < t \leq t_b \end{cases}$$

We wish to solve the problem for integer values of the parameter M ranging from 1 to 15.

A qualitative graph for the step insertion problem appears in Fig. 2.2.

3. Ramp insertion problem

Initial conditions are the same as for step insertion;

$n(0) = 100$, $k_{\text{ex}}(0) = 0$ and $S = 0$.

The excess reactivity function is a bounded ramp function:

$$k_{\text{ex}}(t) = \min(.0025 \cdot t \cdot d, .0218),$$

$$0 \leq t \leq t_b$$

We wish to solve the problem for suitable values of the parameter d ; d values used were .3, .5, .7, 1, 1.5, 2, 2.5 and 3.

A qualitative graph for the ramp insertion problem appears in Fig. 2.3.

Note: The bound .0218 of $k_{\text{ex}}(t)$ here was not reached for the problems solved in this study.

Fig. 2.1. Excess Reactivity $k_{ex}(t)$ for rod withdrawal
(not drawn to scale)

Fig. 2.2. Excess Reactivity $k_{ex}(t)$ for step insertion
(not drawn to scale)

Fig. 2.3. Excess Reactivity $k_{ex}(t)$ for ramp insertion
(not drawn to scale)

III. Problem Solution

A. An overview of the GEAR Package

An analytic solution of the model of Section II appears to be difficult, due to the nonlinear term in Eqn. (2.4). In light of this, a numerical solution is sought.

Numerical methods for the initial value problem

$$\vec{y}'(t) = \vec{f}(t, \vec{y}), \quad \vec{y}(a) = \vec{\eta},$$

have long been a topic of study. Henrici³ provides a clear treatment of the subject. Progress is to a point at which general purpose computer programs have been written to solve standard o.d.e. problems; this relieves the program user of many of the details necessary for efficient solution.

One such set of programs is the GEAR package, developed at Lawrence Livermore Laboratory by A. C. Hindmarsh⁴ and others. Written in FORTRAN, and based on the DIFSUB program of C. W. Gear⁵, these programs have been widely used for the automatic solution of o.d.e.'s. GEAR was used to solve the problem at hand.

A complete description of the GEAR package is given by Hindmarsh, as previously referenced. In the interest of keeping this paper somewhat self-contained, however, a brief discussion of the package will be given. Given this discussion, the reader should be able (provided he is familiar with the FORTRAN programming language) to understand the workings of the problem-solving FORTRAN programs given in Appendix B.

To use the package, one must provide the function $\vec{f}(t, \vec{y})$, the initial conditions $\vec{y}(a) = \vec{\eta}$, and a relative error tolerance ϵ that the numerical solution is to obey. One must provide the values of t at which the solution is desired to be output. A way that the speed of solution may be enhanced is to provide, in addition, an explicit means of evaluating the Jacobian, $\frac{\partial \vec{f}}{\partial \vec{y}}(t, \vec{y})$.

The methods used are the implicit Adams, and the backwards differentiation formulas, particularly as discussed by Gear, as previously referenced. Starting with an order one method and an initial step size (given by the user), the package will automatically adjust the order of the method used, as well as the step size, in such a way as to try to minimize the computational work in solving the problem.

To provide the function $\vec{f}(t, \vec{y})$ to GEAR, one writes a FORTRAN subroutine DIFFUN(N,T,Y,YDOT), which will evaluate $\vec{f}$ at $t = T$ and $\vec{y} = Y$ (an N- vector), with the result being returned in the N- vector YDOT.

One provides $\frac{\partial \vec{f}}{\partial \vec{y}}$ by writing a FORTRAN subroutine PEDERV(N,T,Y,PD,NO). Here, N, T, and Y have the same significance as before; PD is an NO x NO array, where $N \leq NO$. PEDERV computes

$$PD(i, j) = \frac{\partial f_i}{\partial y_j}, \quad 1 \leq i, j \leq N.$$

DIFFUN and PEDERV are called by programs in the GEAR package.

B. Description of a call to interface subroutine DRIVE

For most purposes, a GEAR package user need only be familiar with subroutine DRIVE, an interface program between the user and the rest of the package. In reality, the package contains seven subroutines: DRIVE, INTERP, STIFF, COSET, PSET, DEC, and SOL. A call of subroutine DRIVE will essentially accomplish the integration of the problem between output points, as explained below.

Subroutine DRIVE and its parameters are: SUBROUTINE DRIVE(N, TO, HO, YO, TOUT, EPS, MF, INDEX). N is the number of equations, 19 in this case. TO is the initial value of the independent variable t, 0 for this problem. HO is the step size to be tried on the first step. On output, HO is the step size last used. YO is a vector of length N containing the initial values of $\vec{y}$ on input; YO contains the values of the dependent variable $\vec{y}$ on output at $t = TOUT$. TOUT is the next output point for t.

On the first call to DRIVE, the problem is integrated from TO to TOUT. On a subsequent DRIVE call, the integration is done from the previous TOUT value to the one specified in the subsequent call. Thus, a problem to be integrated on $[a,b]$ with m intermediate output values of t is handled with $m + 1$ calls to DRIVE.

EPS specifies the local error tolerance. During solution by GEAR, estimates $\delta_i(t)$ of the single-step errors in the $y_i(t)$'s are made. An N-vector YMAX is also kept, where YMAX(i) is updated at each step to be the maximum of $|y_i(t)|$ and the current YMAX(i). YMAX(i) is set to

$|y_i(t_0)|$ initially, unless $y_i(t_0) = 0$, in which case $YMAX(i)$ is set to 1 initially. Now EPS: At each step, the computed solution $y_i(t)$ is required to satisfy

$$\sqrt{\frac{\sum_{i=1}^N \left(\frac{\delta_i(t)}{YMAX(i)} \right)^2}{N}} < EPS.$$

For the problem at hand, the $y_i(t_0)$'s are nonzero except for $e_2(0)$, so this is essentially a criterion which uses the root mean square norm on the estimated relative error vector.

MF inputs the method flag. For this application, MF = 21 was selected, to mean that the backwards differentiation formulas were to be used, and that the Jacobian $\frac{\partial \vec{f}}{\partial \vec{y}}$ was provided. A brief discussion of this choice of method is in Section III.D.

Finally, INDEX is an integer control parameter. INDEX = 1 on input indicates the first call to DRIVE. INDEX = 0 on output indicates that the integration was successful. Other output values of INDEX indicate errors. Conveniently, INDEX = 0 on input indicates a normal call to DRIVE which is after the first call.

Note: At this point, the reader should be able to take a brief glance at Appendix B, and understand some code: The main program makes calls to DRIVE, as well as doing associated details; DIFFUN and PEDERV have been explained previously; BLOCK DATA simply contains the initialization of the constants which are in various named COMMON blocks. Variable names in the program were chosen to correspond as closely as possible to the symbols used to describe the problem.

C. The implementation of $k_{ex}(t)$ and the "trip" mechanism.

Recall that the main purpose of solution of this model is to determine the behavior of $n(t)$ for the three types of "accident" postulate. Thus, a modest description of the implementation of the $k_{ex}(t)$ functions and associated "trip" coding is in order.

Some preliminaries are needed. To effect certain control between various subroutines, named COMMON blocks were used. Named COMMON/CONTRL/ contains: i) OPTION, which takes on values of 1, 2, or 3, depending on which accident postulate is being run, as described in Section II. OPTION is read in by the main program. ii) SIMNUM indicates which simulation number of the OPTION specified is being run. Thus, SIMNUM ranges from 1 to 5 for OPTION = 1, etc. iii) SWITCH indicates what is happening with respect to the "trip". For $0 \leq t \leq t_a$, SWITCH = 1. When $t = t_a$ (t_a is t_a in the programs), the trip has just occurred; SWITCH = 2 for the period trip and SWITCH = 3 for the power trip. Then, SWITCH = 4 for $t > t_a$. SWITCH is initialized to 1 in the main program for each simulation, and then changed appropriately, as explained below.

Named COMMON/KXINFO/E2, DENNEU, KEXO, TB communicates real values: E2 is $e_2(t)$ and DENNEU is $n(t)$ for when $t = t_a$; KEXO is $k_{ex}(0)$; and TB is t_b .

A FUNCTION subprogram KEXCES(T) was written to compute $k_{ex}(t)$. Given OPTION, SIMNUM, and SWITCH, this code is fairly straightforward. In case SWITCH is found equal to 2 or 3 in KEXCES, a message is printed, t_b is computed, and SWITCH is set to 4, to acknowledge that a trip has

occurred. Some notes are in order. First, T_A (for t_a) is only an approximation, but is sufficient for our purposes. Second, the GEAR package may sometimes "back up" and do an integration step over if desired accuracy of the solution is not achieved. Thus, KEXCES was written to be flexible enough to evaluate $k_{ex}(t)$ for some t_2 (less than some t_1), where KEXCES (t_1) has already been calculated. This is essential when T is near t_b or t_c .

Given the above routine KEXCES(T) and conventions, there must still be code which "monitors" $n(t)$ and $e_2(t)$, and signals a "trip" when appropriate, i.e., determine t_a . In the real life situation, this monitoring occurs continuously, in essence; for a computer program which models the problem using discrete variable methods, checking on $n(t)$ and $e_2(t)$ at "frequent" intervals is done instead.

One place that checking on $n(t)$ and $e_2(t)$ could be done is after each call to DRIVE in the main routine. However, there is a better place to make these checks: within subroutine DRIVE. Subroutine DRIVE will make as many calls as needed to another subroutine of GEAR previously mentioned, STIFF, which performs one step of the integration. By monitoring the values of $n(t)$ and $e_2(t)$ within subroutine DRIVE after each call to subroutine STIFF, a better approximation to t_a is obtained.

The changes made in SUBROUTINE DRIVE are listed in Appendix B. Briefly, there are tests made on $n(t)$ and $e_2(t)$ against the "trip" values, when SWITCH = 1. If either of the "trip" values is exceeded,

SWITCH is set to 2 or 3 accordingly, the $n(t)$ or $e_2(t)$ value is recorded, and the program's execution continues. The next time KEXCES is called, the new value of SWITCH will be detected and the situation handled as previously described.

Another modification made in subroutine DRIVE is to limit HMAX, the maximum allowable step size, to one half of the distance between output points specified in the main program. This is to assure some accuracy in t_a ; see the explanatory notes in subroutine KEXCES and in the code modifying DRIVE. Observe that this modification may diminish efficiency of solution.

A more elaborate coding would determine t_a more precisely; however, the t_a produced by these programs is greater than or equal to the one which would be given by a more precise method. In other words, it is possible that an earlier trip would be indicated by a more precise t_a ; the results obtained here are thus conservative in this regard.

One may note in the main program: COMMON/GEAR9/HUSED, NQUSED, NSTEP, NFE, NJE, which provides some statistical information from the package. The variables are, respectively, the last successful stepsize used, the last successful order of numerical method used, and running totals of number of steps taken, function evaluations made, and Jacobian evaluations made.

Appendix B should be understandable to the reader at this point.

D. A note on "stiffness".

A distinguishing feature of the problem of consideration is that it possesses the property of being "stiff". Stiff systems and methods of solving them efficiently ^{are} currently a topic of great interest to both numerical analysts and to people with stiff problems to solve. Gear discusses them; other general reference include a survey of Bjurel et al.⁶ Indeed, this survey notes that nuclear reactor modeling is a source of stiff problems.

Generally speaking, a system of o.d.e.'s is stiff if the eigenvalues of its Jacobian $\frac{\partial \vec{f}}{\partial \vec{y}}$ are widely separated. For the problem of this paper, the reason that the problem is stiff can be seen from physical considerations: the decay constants for the various delayed neutron groups (the λ_i 's) are widely separated.

If $\{\mu_i(t)\}_{i=1}^{19}$ denotes the eigenvalues of $\frac{\partial \vec{f}}{\partial \vec{y}}$ at some time t , then the "stiffness ratio" is defined to be $\frac{\max_i |\operatorname{Re}(\mu_i)|}{\min_i |\operatorname{Re}(\mu_i)|}$. The bigger the "stiffness ratio", the more "stiff" the problem is said to be. For informational purposes, the EISPACK⁷ codes were employed to determine the eigenvalues of the Jacobian matrix for this problem at various points in time. It was found that "stiffness ratio" for the problem was on the order of 10^{20} near $t = 0$ on the step and ramp insertions; otherwise, the ratio was around 2×10^{10} , and did not vary greatly.

The crux of the stiffness problem lies in methods of numerical solution. Conventional methods, such as Adams-Bashforth and Adams-Moulton are not appropriate, because of severe step-size restrictions imposed by requirements for absolute stability of the numerical methods. These restrictions are dictated by the large eigenvalues of $\frac{\partial \vec{f}}{\partial \vec{y}}$, even though their influence on the solution dies away. The choice of method flag (MF = 21), specifying the backward differentiation formulas with Jacobian evaluated analytically, was made because these formulas have the property (defined by Gear) of being "stiffly stable". This note is not intended to be a complete explanation of the problem, but a recognition that it exists.

IV. Results of the Computation

When a numerical method is employed to approximate the solution to a mathematical problem, one must ask how good the numbers produced are, i.e., what is the error committed? Of course, to know this answer is to know the problem's solution. In the program of Appendix B, a value of $\text{EPS} = 10^{-4}$ was used. To try to gain a rough idea of the error, a value of $\text{EPS} = 10^{-6}$ was used subsequently. For the most part, answers obtained using these two values of EPS agreed to at least 3 digits. Where disagreement occurred, it was not great; it was certainly not great in a qualitative sense. In the following tables, the $\text{EPS} = 10^{-6}$ results are given.

The results obtained are most concisely displayed in Figures 4.1 - 4.3. These graphs show neutron density $n(t)$ for the rod withdrawal, step insertion, and ramp insertion problems, respectively. Note that in all cases, either the period or power trip did occur and that $n(t)$ does not greatly exceed 150. Note that by using larger steps or steeper ramps, 150 could be greatly exceeded; what is indicated here are approximate limits on what steps and slopes produce manageable values of $n(t)$. Also note that the term "reactor power" is used synonymously with "neutron density".

Table 4.1 presents in tabular form some of the information used to produce Figures 4.1 - 4.3. Given in each line of this table are parameters for the problem run, approximately where the trip occurred and what type of trip it was, and the approximate maximum value of $n(t)$. Some of these numbers are approximations because values of the dependent variables (such as $n(t)$ and $e_2(t)$) were only printed out

at predetermined values of t ; a mechanism was not built into the program for iterating to the "exact" trip points or "exact" maximum value of $n(t)$. Tables 4.2 - 4.9 present a representative sampling of the data which was generated by the program of Appendix B and used to create Figures 4.1 - 4.3. Only pertinent digits are given.

Figure 4.1. Reactor power for the rod withdrawal problems.

Figure 4.2. Reactor power for step insertion problems #3, 5, 7, 9, 11, 13, 15.

Figure 4.3. Reactor power for the ramp insertion problems.

Table 4.1. Summary of Approximate Trip Points and $n(t)$ MaximumsRod Withdrawal

<u>Simulation No.</u>	<u>S</u>	<u>t_a (Approx.)</u>	<u>Type of Trip</u>	<u>Approx. Max. $n(t)$ (in units of 10^{-3})</u>
1	8.491×10^{-3}	179.49	Period	2.9012
2	4.25×10^{-2}	179.65	Period	15.474
3	8.491×10^{-2}	179.78	Period	32.277
4	4.25×10^{-1}	179.88	Period	167.29
5	8.491×10^{-1}	179.90	Period	337.02

Step Insertion

<u>Simulation No.</u>	<u>Step Height</u>	<u>t_a (Approx.)</u>	<u>Type of Trip</u>	<u>Approx. Max. $n(t)$</u>
1	.0002	83.36	Power	125.08
2	.0004	25.02	Power	125.25
3	.0006	.020	Period	109.23
4	.0008	.015	Period	112.64
5	.0010	.012	Period	116.24
6	.0012	.010	Period	120.08
7	.0014	.009	Period	124.11
8	.0016	.006	Period	128.43
9	.0018	.006	Period	133.04
10	.0020	.005	Period	137.94
11	.0022	.005	Period	143.18
12	.0024	.004	Period	148.72
13	.0026	.004	Period	154.63
14	.0028	.004	Period	161.12
15	.0030	.004	Period	167.83

Table 4.1. (Continued)

Ramp Insertion

<u>Simulation No.</u>	<u>Ramp Slope (in units of 10^{-3})</u>	<u>t_a (Approx.)</u>	<u>Type of Trip</u>	<u>Approx. Max. $n(t)$</u>
1	.750	1.78	Power	132.84
2	1.25	1.16	Power	137.24
3	1.75	.88	Power	141.47
4	2.50	.28	Period	123.76
5	3.75	.14	Period	126.54
6	5.00	.10	Period	133.18
7	6.25	.09	Period	142.37
8	7.50	.08	Period	152.30

Table 4.2. Rod Withdrawal. Simulation No. 1. $S = 8.491 \times 10^{-3}$

t	$n(t)$ (in units of 10^{-5})	$e_0(t)$	$e_I(t)$ (in units of 10^{-5})	$e_2(t)$	$k_{ex}(t)$
0.	3.00	-6.97	6.97	0.0000	-.1500
20.	3.38	-6.97	6.96	.01460	-.1330
40.	3.86	-6.96	6.95	.03223	-.1160
60.	4.52	-6.94	6.94	.0546	-.099
80.	5.44	-6.92	6.91	.0848	-.082
100.	6.83	-6.88	6.86	.130	-.065
120.	9.72	-6.83	6.80	.204	-.048
140.	13.91	-6.73	6.68	.356	-.031
160.	28.20	-6.55	6.43	.824	-.014
175.	99.83	-6.14	5.69	3.21	-.0013
176.47	127.69	-6.04	5.47	4.13	.0000
179.46	259.50	-5.74	4.56	8.35	.0025
				($\approx$ period trip)	
179.82	290.12	-5.68	4.38	9.33	.0012
	($\approx$ maximum)				
181.	6.88	-5.71	5.98	-1.960	-.15
200.	3.40	-6.11	6.21	-.761	-.15

Table 4.3. Rod Withdrawal. Simulation No. 3. $S = 8.491 \times 10^{-2}$

t	$n(t)$ (in units of 10^{-4})	$e_0(t)$	$e_I(t)$ (in units of 10^{-5})	$e_2(t)$	$k_{ex}(t)$
0.	3.00	-5.97	5.97	0.0	-.150
15.	3.27	-5.96	5.95	.079	-.137
30.	3.60	-5.93	5.92	.131	-.124
45.	4.01	-5.90	5.87	.169	-.112
60.	4.52	-5.85	5.82	.203	-.099
75.	5.18	-5.79	5.76	.238	-.086
90.	6.06	-5.72	5.68	.282	-.074
105.	7.30	-5.64	5.59	.340	-.061
120.	9.18	-5.54	5.48	.426	-.048
135.	12.33	-5.41	5.34	.569	-.035
150.	18.68	-5.23	5.12	.852	-.022
165.	37.56	-4.93	4.70	1.64	-.0098
176.47	127.69	-4.39	3.74	4.66	.0000
179.78	287.34	-4.03	2.87	8.33	.0028
				($\approx$ period trip)	
180.12	322.77	-3.98	2.73	8.98	.0031
	($\approx$ maximum)				
180.36	8.98	-4.25	12.82	-68.67	-.15
210.00	3.27	-5.92	5.88	-.400	-.15

Table 4.4. Rod Withdrawal. Simulation No. 5. $S = 8.491 \times 10^{-1}$

<u>t</u>	<u>n(t) (in units of 10⁻³)</u>	<u>e(t)</u>	<u>e_I(t) (in units of 10⁻⁵)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0.0	3.0	-4.98	4.98	0.0	-.15
15.0	3.27	-4.94	4.92	.151	-.137
30.0	3.60	-4.90	4.88	.168	-.124
45.0	4.01	-4.86	4.83	.187	-.112
60.0	4.52	-4.80	4.78	.210	-.099
75.0	5.18	-4.75	4.71	.242	-.086
90.0	6.06	-4.68	4.64	.282	-.074
105.0	7.30	-4.60	4.55	.340	-.061
120.0	9.18	-4.50	4.44	.425	-.048
135.0	12.33	-4.37	4.29	.568	-.035
150.0	18.68	-4.19	4.07	.848	-.022
165.0	37.56	-3.89	3.66	1.63	-.0098
176.47	127.69	-3.36	2.72	4.57	.0000
179.90	299.08	-2.99	1.82	8.34	.0029
				($\approx$ period trip)	
180.24	337.02 ($\approx$ maximum)	-2.94	1.68	9.01	.0032
180.48	9.22	-3.88	20.18	-164.04	-.15
200.00	3.44	-4.91	4.94	-.221	-.15

Table 4.5. Step Insertion. Simulation No. 2. Step Height = .0004

<u>t</u>	<u>n(t)</u>	<u>e₀(t)</u>	<u>e_I(t) (in units of 10⁻⁶)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0.0	100.00	- .475	4.75	0.0	0.0
2.0	108.20	- .441	4.05	.259	.0004
4.0	110.23	- .433	4.03	.214	.0004
6.0	112.02	- .426	4.00	.191	.0004
8.0	113.67	- .420	3.96	.174	.0004
10.0	115.21	- .414	3.91	.162	.0004
12.0	116.67	- .408	3.87	.152	.0004
14.0	118.07	- .403	3.83	.145	.0004
16.0	119.42	- .398	3.79	.138	.0004
18.0	120.72	- .394	3.75	.133	.0004
20.0	122.00	- .389	3.71	.129	.0004
22.0	123.26	- .385	3.67	.126	.0004
24.0	124.49	- .380	3.63	.122	.0004
25.00	125.10	- .378	3.61	.121	.0004
	(≈ power trip)				
25.25	125.25	- .378	3.61	.121	.0004
	(≈ maximum)				
25.50	9.04	-1.512	290.62	-358.023	-.0929
26.0	4.19	-1.789	26.30	- 6.301	-.15
28.0	3.33	-1.945	24.48	- 3.666	-.15
30.0	2.58	-2.056	24.47	- 2.846	-.15

Table 4.6. Step Insertion. Simulation No. 5. Step Height = .0010

<u>t</u>	<u>n(t)</u>	<u>e₀(t)</u>	<u>e_I(t) (in units of 10⁻⁶)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0.0	100.00	- .475	4.75	0.0	0.0
.015	102.57	- .464	- 47.66	10.81	.001
				(≈ period trip)	
.05	107.01	- .446	- 28.39	20.17	.001
.10	110.87	- .430	- 13.38	18.02	.001
.15	113.06	- .422	- 5.82	12.50	.001
.20	114.35	- .417	- 1.91	8.04	.001
.25	115.17	- .414	.15	5.14	.001
.30	115.74	- .412	1.25	3.39	.001
.35	116.16	- .410	1.85	2.38	.001
.36	116.24	- .410	1.94	2.23	.001
	(≈ maximum)				
.365	115.68	- .412	125.11	- 0.26	- .00098
.37	111.87	- .426	336.64	- 16.06	- .0045
.375	105.02	- .453	533.87	- 44.42	- .0080
.40	53.64	- .743	1214.95	-297.12	- .0256
.45	12.61	-1.37	555.90	-532.59	- .0606
.50	7.49	-1.59	282.30	-351.56	- .0958
.55	5.40	-1.73	210.75	-230.00	- .1308
.60	4.57	-1.81	29.07	-129.64	- .1500
.65	4.50	-1.82	28.39	- 52.46	- .1500
.70	4.44	-1.82	28.07	- 23.88	- .1500
.75	4.37	-1.83	27.79	- 13.20	- .1500
.80	4.32	-1.83	27.53	- 9.13	- .1500

Table 4.7. Step Insertion. Simulation No. 12. Step Height = .0024

<u>t</u>	<u>n(t)</u>	<u>e₀(t)</u>	<u>e_I(t) (in units of 10⁻⁶)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0	100.00	- .48	4.75	0.0	0.0
.005	102.19	- .47	- 139.63	10.16	.0024
				(≈ period trip)	
.05	117.86	- .40	- 77.53	49.04	.0024
.10	129.05	- .36	- 43.57	45.36	.0024
.15	136.23	- .34	- 25.76	33.09	.0024
.20	141.00	- .33	- 15.86	22.64	.0024
.25	144.31	- .32	- 10.14	15.42	.0024
.30	146.73	- .31	- 6.78	10.78	.0024
.35	148.61	- .30	- 4.75	7.89	.0024
.355	148.72	- .30	26.80	7.51	.0019
	(≈ maximum)				
.4	49.28	- .78	1375.34	-401.54	- .0297
.45	11.90	-1.39	494.75	-554.79	- .0648
.5	7.34	-1.60	271.12	-349.34	- .0999
.55	5.35	-1.74	205.49	-225.88	- .1350
.6	4.67	-1.80	29.14	-118.50	- .1500
.65	4.60	-1.81	28.70	- 48.55	- .1500
.7	4.53	-1.81	28.36	- 22.62	- .1500
.75	4.46	-1.82	28.05	- 12.90	- .1500
.8	4.40	-1.82	27.78	- 9.17	- .1500

Table 4.8. Ramp Insertion. Simulation No. 3. Ramp Slope = 1.75×10^{-3}

<u>t</u>	<u>n(t)</u>	<u>e₀(t)</u>	<u>e_I(t) (in units of 10⁻⁶)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0.0	100	- .48	4.75	0.0	0.0
.1	101.09	- .47	- 1.40	2.88	.000175
.2	103.20	- .46	- 3.22	5.01	.000350
.3	105.71	- .45	- 4.08	5.91	.000525
.4	108.47	- .44	- 4.72	6.40	.000700
.5	111.46	- .43	- 5.31	6.77	.000875
.6	114.69	- .42	- 5.91	7.12	.001050
.7	118.18	- .40	- 6.52	7.46	.001225
.8	121.94	- .39	- 7.15	7.82	.001400
	(≈ power trip)				
.88	125.18	- .38	- 7.67	8.11	.001540
.9	126.02	- .38	- 7.81	8.18	.001575
1.0	130.42	- .36	- 8.49	8.56	.001750
1.1	135.20	- .34	- 9.20	8.96	.001925
1.2	140.38	- .33	- 9.95	9.37	.002100
1.22	141.47	- .32	- 10.10	9.45	.002135
	(≈ maximum)				
1.3	18.10	-1.21	891.53	-585.30	- .049690
1.4	6.15	-1.68	228.17	-266.95	- .119715
1.5	4.69	-1.80	28.90	- 70.39	- .150000
1.6	4.55	-1.81	28.21	- 16.02	- .150000
1.7	4.42	-1.82	27.65	- 8.20	- .150000

Table 4.9. Ramp Insertion. Simulation No. 7. Ramp Slope = 6.25×10^{-3}

<u>t</u>	<u>n(t)</u>	<u>e₀(t)</u>	<u>e_I(t) (in units of 10⁻⁶)</u>	<u>e₂(t)</u>	<u>k_{ex}(t)</u>
0	100.00	- .48	4.75	0.0	0.0
.01	100.06	- .48	1.02	.26	.000063
				(≈ period trip)	
.09	103.31	- .46	- 16.19	9.20	.000562
.1	103.95	- .46	- 17.38	10.34	.000625
.15	107.69	- .44	- 21.85	15.08	.000938
.20	112.14	- .43	- 24.94	18.39	.001250
.25	117.20	- .41	- 27.45	20.76	.001562
.30	122.88	- .39	- 29.75	22.62	.001875
.35	129.23	- .36	- 32.03	24.25	.002188
.40	136.32	- .34	- 34.38	25.81	.002500
.44	142.37	- .32	29.69	26.36	.001687
	(≈ maximum)				
.45	132.65	- .35	451.83	- 11.00	- .005270
.50	26.88	-1.04	1227.17	-533.63	- .040058
.55	9.86	-1.47	374.54	-479.77	- .074845
.60	6.56	-1.65	245.38	-300.60	- .109633
.65	4.91	-1.78	191.93	-199.54	- .144420
.70	4.58	-1.81	28.77	- 90.78	- .150000
.75	4.52	-1.81	28.41	- 38.18	- .150000
.80	4.45	-1.82	28.10	- 18.64	- .150000
.85	4.39	-1.83	27.81	- 11.29	- .150000
.90	4.33	-1.83	27.55	- 8.44	- .150000

Appendix A: Derivation of Circuit Equations

1. Ebers-Moll Transistor Model

Figure A.1

Current and Voltage Conventions for the Ebers-Moll Equations

I_B = base current (amp)

I_C = collector current (amp)

I_E = emitter current (amp)

ϕ_C = collector to base voltage drop (volt)

ϕ_E = emitter to base voltage drop (volt)

Using the sign conventions shown in Figure A.1 for a PNP transistor, the Ebers-Moll equations are:

$$\alpha_N I_{EO} = \alpha_I I_{CO} \quad (A.1)$$

$$I_E = -\frac{I_{EO}}{1-\alpha_N \alpha_I} (e^{\Lambda \phi_E} - 1) + \frac{\alpha_I I_{CO}}{1-\alpha_N \alpha_I} (e^{\Lambda \phi_C} - 1) \quad (A.2)$$

$$I_C = \frac{\alpha_N I_{EO}}{1-\alpha_N \alpha_I} (e^{\Lambda \phi_E} - 1) - \frac{I_{CO}}{1-\alpha_N \alpha_I} (e^{\Lambda \phi_C} - 1) \quad (A.3)$$

where

α_N = Normal alpha, the DC common base forward current transfer ratio (dimensionless).

α_I = Inverted alpha, same as α_N but with emitter and collector interchanged (dimensionless).

I_{EO} = Emitter leakage current with base to emitter junction reverse biased, collector open (amp).

I_{CO} = Collector leakage current with collector to base junction reverse biased, emitter open (amp).

$A = \frac{q}{mKT}$, where

q = Electronic charge = 1.60×10^{-19} coulomb.

m = A dimensionless constant with a value between 1 and 2. Its value is experimentally determined.

K = Boltzmann's constant = $1.38 \times 10^{-23} \frac{\text{Joules}}{\text{°K}}$.

T = Absolute temperature, degrees Kelvin.

2. Feedback Circuit for the Logarithmic Amplifier

The feedback circuit for the logarithmic amplifier is formed by a differential connection of two transistors as shown below.

Figure A.2

Simplified Schematic of the Logarithmic Amplifier

Referring to Figure A.2, the action of amplifier A1 is such that φ_{C1} is effectively zero and $e^{\Lambda\varphi_{E1}} \gg 1$. From Ebers-Moll equation A.3,

$$I_{C1} = \frac{\alpha_N I_{EO}}{1 - \alpha_N \alpha_I} e^{\Lambda\varphi_{E1}}.$$

One may see from Figure A.2 that

$$\varphi_{E1} = V_2 + \varphi_{E2}.$$

In addition, the sole purpose of amplifier A2 is to fix

$$I_{C2} = -I_C,$$

and

$$V_2 + \varphi_{C2} = 0.$$

The latter equation results in

$$\varphi_{C2} = -V_2,$$

so that the Ebers-Moll equations for transistor 2 yields

$$I_{C2} = \frac{\alpha_N I_{EO}}{1 - \alpha_N \alpha_I} e^{\Lambda(\varphi_{E1} - V_2)} - \frac{I_{CO}}{1 - \alpha_N \alpha_I} e^{-\Lambda V_2}.$$

Using equation (A.1), this may be rewritten as

$$I_{C2} = e^{-\Lambda V_2} \left\{ \frac{\alpha_N I_{EO}}{1 - \alpha_N \alpha_I} e^{\Lambda \varphi_{E1}} - \frac{\alpha_N I_{EO}}{\alpha_I (1 - \alpha_N \alpha_I)} \right\}.$$

For sufficiently large values of φ_{E1} , the last equation reduces to

$$I_{C2} \approx \frac{\alpha_N I_{EO}}{1 - \alpha_N \alpha_I} e^{\Lambda \varphi_{E1}} e^{-\Lambda V_2},$$

and one may form the ratio

$$\frac{I_{C1}}{I_{C2}} = e^{\Lambda V_2},$$

and

$$I_{C1} = I_{C2} e^{\Lambda V_2} = -I_C e^{\Lambda V_2}.$$

If it is assumed that I_{B2} is sufficiently small, then one may write directly from Figure A.2,

$$V_2 = \frac{E_{out} R_{TC}}{R_{TC} + R_G}.$$

This in turn gives

$$I_{C1} = -I_C e^{\frac{\Delta E_{out} R_{TC}}{R_{TC} + R_G}} \quad (A.4)$$

In the above equations R_{TC} is a temperature-sensitive resistor chosen with properties appropriate to cancel the temperature dependence of Δ . The circuit values of R_{TC} and R_G are such that

$$\frac{\Delta R_{TC}}{R_{TC} + R_G} = \frac{2.313}{\text{volt}}.$$

Hence,

$$I_{C1} = -I_C e^{2.313 E_{out}}$$

3. Dynamic Model of the Amplifier

In the reactor safety system under study the circuit of the previous section is driven by a neutron sensitive chamber located adjacent to and sensing neutrons from the core of ALRR. The current from such a chamber is proportional to the incident neutron flux over a wide dynamic range, and is therefore proportional to reactor power. This element forms the input device for the circuit of Figure A.3 below.

Using the results from A.2 above the dynamic model of the logarithmic amplifier may be depicted by the following simplified schematic.

Figure A.3

Dynamic Model of Logarithmic Amplifier Circuit

For Figure A.3,

$n(t)$ = neutron flux, an independent variable
(neutrons/cm²/sec)

$i_{in}(t)$ = amplifier input current (amps)

$e_I(t)$ = amplifier input voltage (volts)

C = a stabilization capacitor (farads)

$e_0(t)$ = amplifier output voltage (volts)

$i(e_0) = \text{feedback current} = I_C e^{\alpha e_0(t)}$ (amps), where

$$\alpha = \frac{2.313}{\text{volt}}$$

$A(s)$ = the Laplace transform of the amplifier

$$\text{impulsive response} = \frac{\omega_0 A_{DC}}{s + \omega_0}$$

Referring to Figure A.3, one may write a node voltage equation at the $e_I(t)$ junction.

$$i_{in}(t) - i(e_0) + C \frac{d}{dt}(e_I(t) - e_0(t)) = 0 \quad (A.5)$$

The amplifier impulse response equation (in Laplace transform form) is

$$E_0(s) = -\frac{\omega_o A_{DC}}{s + \omega_o} E_I(s) .$$

Here, ω_o is the amplifier break frequency in radians/second; A_{DC} is the amplifier DC gain, a dimensionless quantity.

Expanding this equation gives

$$(s + \omega_o)E_0(s) = -\omega_o A_{DC} E_I(s) , \text{ and}$$

taking its Laplace inverse results in

$$\frac{de_0(t)}{dt} + \omega_o e_0(t) = -\omega_o A_{DC} e_I(t) . \quad (A.6)$$

Dividing equation (A.5) by C, and rearranging the terms of both equations (A.5) and (A.6) gives:

$$\frac{de_0(t)}{dt} = -\omega_o e_0(t) - \omega_o A_{DC} e_I(t) , \quad (A.7)$$

and

$$\frac{de_I(t)}{dt} = \frac{de_0(t)}{dt} - \frac{1}{C} i_{in}(t) + \frac{1}{C} I_C e^{\alpha e_0(t)} . \quad (A.8)$$

Equations (A.7) and (A.8) characterize the dynamic response of the logarithmic circuit. One may quickly verify the circuit's static character by equating all of the derivative terms to zero. Under these conditions, equation (A.8) reduces to

$$I_C e^{\alpha e_0} = i_{in} .$$

From this it follows that

$$\begin{aligned} \alpha e_0 &= \frac{i_{in}}{I_C} \\ \alpha e_0 &= \ln \frac{i_{in}}{I_C} \\ e_0 &= \frac{1}{\alpha} \ln \frac{i_{in}}{I_C} = \log_{10} \frac{i_{in}}{I_C} \end{aligned} \quad (A.9)$$

Here we see that, provided the circuit is dynamically stable, its static character possesses the desired logarithmic properties. Now, since $i_{in}(t)$ is proportional to $n(t)$, one may say

$$e_0 = \log_{10} \frac{k n}{I_C}.$$

The circuit values are such that $I_C = 10^{-4}$ amp. k is chosen so that when

$$n = 300,$$

$$e_0 = 0.$$

Thus

$$\log_{10} \frac{300k}{10^{-4}} = 0,$$

$$k = \frac{10^{-6}}{3},$$

and

$$i_{in}(t) = \frac{10^{-6}}{3} n(t). \quad (A.10)$$

Equation (2.3) is identical to equation (A.7). To form equation (2.4), equation (A.10) must be substituted into equation (A.8).

4. The Band-limited Derivative Circuit

In the development of mathematical models for nuclear reactors one finds that the time response of the reactor for fixed reactivity conditions is dominated by an exponential function of the form

$$P(t) = P_0 e^{t/T}, \text{ where}$$

P = reactor power

t = time

T = reactor period

Clearly, T represents the time required for the reactor power to increase by a factor of e (≈ 2.72); thus, the period, T , becomes a useful measure of the reactor's rate of power increase. Given a circuit capable of measuring the logarithm of the reactor power (such as the circuit of the previous section),

$$\ln P(t) = \ln P_0 + \frac{t}{T},$$

and

$$\frac{d}{dt} \ln P(t) = \frac{1}{T}.$$

Thus one may measure the reactor period by providing a circuit capable of generating the time derivative of the output from the logarithmic amplifier. Recall from Section II that a "period trip" occurs when $e_2(t)$ exceeds $8\frac{1}{3}$ volts. This corresponds to the reactor period being less than 3 seconds. An appropriate scaling factor for this correspondence occurs between the output of the derivative circuit and the input to the trip circuit; this is to be explained in a future report.

In any operational analog circuit, the generation of signals representing the time derivatives of other signals poses two serious problems. First, to form a true derivative, a circuit possessing unlimited bandwidth would be required. This represents a physical impossibility. Second, the frequency character of the derivative operation enhances high frequency noise present in all real circuits.

As a consequence of these two limitations, all real time-derivative circuits must be finite bandwidth approximations to the desired circuit, and errors result. These errors must be included in the calculation of the dynamic response of the reactor safety system.

The derivative circuit used in the problem at hand is an example of a class of circuits referred to as "state-variable filters." Its dynamic character may be derived by reference to the block diagram of Figure A.4.

Figure A.4

Block Diagram of a State-Variable Derivative Circuit

Using the Laplace transform notation, one may write the output of the summing stage for the above circuit as

$$E_0(s) - \omega_1 X(s) = sX(s), \quad (\text{A.11})$$

and immediately calculate

$$X(s) = \frac{E_0(s)}{s + \omega_1}.$$

Here

s = the Laplace variable

$E_0(s)$ = the Laplace transform of the circuit input voltage

ω_1 = a Laplace "break frequency"
(radians/second)

$X(s)$ = the Laplace transform of the time-varying voltage, $x(t)$.

Thus, the circuit output must be

$$E_2(s) = AsX(s) = \frac{As E_0(s)}{s + \omega_1} \quad (\text{A.12})$$

where A is a dimensionless voltage gain.

Continuing,

$$(s + \omega_1)E_2(s) = As E_0(s),$$

and now taking the Laplace inverse gives

$$\frac{de_2(t)}{dt} + \omega_1 e_2(t) = A \frac{de_0(t)}{dt}$$

For the circuit used, $\omega_1 = 20$ radians/second, and $A = 1152$.

Upon rewriting, the above equation becomes

$$\frac{de_2(t)}{dt} = 1152 \frac{de_0(t)}{dt} - 20 e_2(t). \quad (\text{A.13})$$

Equation (A.13) is the same as equation (2.5).

```

C ***** MAIN PROGRAM: REACTOR KINETICS & SAFETY DEVICE
C SIMULATION. JULY 1976. *****
C
C IMPLICIT REAL*8(A-H,K,L,O-Z)
C REAL*8 Y0(19),TPR(434),SFACTR(5),OP3FAC(8),IC
C INTEGER OPTION,SIMNUM,SWITCH,UPBND(3)
C COMMON /CON/ ADC,ALPHA,BETASM,CP,IC,L,OMEGA,S,BETA(15),LAMBDA(15)
C COMMON /KXINFO/ E2,DENNEU,KEX0,TB
C COMMON /CONTRL/ OPTION,SIMNUM,SWITCH
C COMMON /GEAR9/ HUSED,NGUSED,NSTEP,NFE,NJE
C DATA UPBND / 5, 15, 8 /
C DATA SFACTR / 1.D0, 5.D0, 10.D0, 50.D0, 100.D0 /
C DATA OP3FAC / .3D0, .5D0, .7D0, 1.D0, 1.5D0, 2.D0, 2.5D0, 3.D0 /
C
C ***** THIS MAIN PROGRAM (WHICH INCLUDES A CALL TO SUBROUTINE
C DRIVES), AND THE ACCOMPANYING SUBPROGRAMS BLOCK DATA,
C DIFFUN, PEDERV, AND KEXCES, ESSENTIALLY DESCRIBE TO
C THE GEAR PACKAGE THE SYSTEM OF ORDINARY DIFFERENTIAL
C EQUATIONS WITH INITIAL CONDITIONS WHICH MODEL THE
C NUCLEAR CONTROL ROD PROBLEM. SUBROUTINE DRIVE (WHICH,
C FOR THIS PROBLEM, HAS BEEN SLIGHTLY MODIFIED AS
C DESCRIBED BELOW) IS THE INTERFACE BETWEEN A USER AND
C THE GEAR PACKAGE, AS DOCUMENTED IN A. C. HINDMARSH,
C "GEAR: ORDINARY DIFFERENTIAL EQUATION SYSTEM SOLVER",
C LAWRENCE LIVERMORE LABORATORY, REPT. UCID - 30001,
C REV. 3, DECEMBER 1974.
C NOTATION: Y0(1) IS N
C Y0(I) IS C(I-1), FOR I = 2, ..., 16.
C Y0(17) IS EO
C Y0(18) IS EI
C Y0(19) IS E2 *****
C
C READ (5,5) OPTION
C 5 FORMAT (I1)
C IBND = UPBND(OPTION)
C DO 230 SIMNUM = 1, IBND

```

```

GO TO ( 10, 70, 120), OPTION
C
C ***** OPTION 1 INITIALIZATIONS *****
C
10 WRITE (6,15)
15 FORMAT ('1OPTION = 1 ; ROD WITHDRAWAL PROBLEM.  EXCESS ',
X      'REACTIVITY FUNCTION IS A GRADUAL RAMP' / ' INITIALLY',
X      ' N(T) IS "SMALL". (SEE Y0(1) BELOW)' )
C
IF (SIMNUM .GT. 1) GO TO 50
NPTS = 434
DO 20 I = 1,177
20   TPR(I) = I - 1.00
TPR(178) = .1500 / 8.5D-4
DO 30 I = 1, 227
30   TPR(I+178) = 176.48D0 + ( I - 1 ) * .02D0
DO 40 I = 1,29
40   TPR(I+405) = 181.00 + I
C
KEX0 = -0.15D0
SFIRST = - ( KEX0 * 3.D-5 ) / L
50 S = SFIRST * SFACTR(SIMNUM)
C
WRITE (6,55) SIMNUM,S
55 FORMAT (' SIMULATION NO.',I3,'.  SOURCE S = ',D14.7)
WRITE (6,65)
65 FORMAT (' SLOPE OF RAMP (ROD WITHDRAWAL) IS 8.5D-4')
C
Y0(1) = - ( S * L ) / KEX0
GO TO 150
C
C ***** OPTION 2 INITIALIZATIONS *****
C
70 WRITE (6,75)
75 FORMAT ('1OPTION = 2 ; EXCESS REACTIVITY FUNCTION IS "STEP ',

```



```

X          'INSERTION". INITIALLY N(T) IS 100% OF FULL POWER.')
C
      NPTS = 361
      IF (SIMNUM .GT. 3) GO TO 110
      IF (SIMNUM .GT. 2) GO TO 90
      DO 80 I = 1, NPTS
80      TPR(I) = ( I - 1 ) * .25D0
      GO TO 110
90      DO 100 I = 1, NPTS
100     TPR(I) = ( I - 1 ) * .005D0
C
110     KEX0 = 0.D0
      S = 0.D0
      WRITE (6,55) SIMNUM,S
      KXSTEP = SIMNUM * .0002D0
      WRITE (6,115) KXSTEP
115     FORMAT (' STEP INSERTION AT T > 0 IS OF HEIGHT = ',D14.7)
C
      Y0(1) = 100.D0
      GO TO 150
C
C          ***** OPTION 3. INITIALIZATIONS *****
C
120     WRITE (6,125)
125     FORMAT ('1OPTION = 3 ; EXCESS REACTIVITY FUNCTION IS "RAMP ",
X          'INSERTION". INITIALLY N(T) IS 100% OF FULL POWER.')
C
      NPTS = 251
      IF (SIMNUM .GT. 1) GO TO 140
      DO 130 I = 1, NPTS
130     TPR(I) = ( I - 1 ) * .01D0
C
      KEX0 = 0.D0
      S = 0.D0
140     WRITE (6,55) SIMNUM,S

```

```

KXRAMP = OP3FAC(SIMNUM) * .0025D0
WRITE (6,145) KXRAMP
145  FORMAT (' RAMP INSERTION AT T = 0 IS OF SLOPE = ',D14.7)
C
Y0(1) = 100.D0
150  CONTINUE
C ***** SET INITIAL VALUES *****
C
DO 160 I = 1,15
160  Y0(I+1) = ( - BETA(I) / ( LAMBDA(I) * L ) )
X      * ( 1.D0 + KEX0 ) * Y0(1)
Y0(17) = DLOG( ( 1.D-6 * Y0(1) ) / ( 3.D0 * IC ) ) / ALPHA
Y0(18) = - Y0(17) / ADC
Y0(19) = 0.D0
WRITE (6,165) ( Y0(I), I = 1, 19 )
165  FORMAT ('0Y0(I), I = 1, 19 AT TIME T = 0 :',(4D25.16))
WRITE (6,175)
175  FORMAT ('0',T7,'T',T21,'N',T38,'EO',T55,'EI',T71,
X      'E2',T88,'KEX')
C
C ***** INITIALIZE PROGRAM STATISTICS COUNTERS *****
C
NSTEP = 0
NFE = 0
NJE = 0
C
C ***** PREPARE FOR CALLS TO DRIVE *****
C
SWITCH = 1
IEQ = 19
T0 = 0.D0
H0 = 1.D-4
EPS = 1.D-4
MF = 21
INDEX = 1

```

```

C
C ***** DO THE INTEGRATION FROM TPR(1) TO TPR(NPTS) IN
C NPTS - 1 PIECES, UNLESS HALTED SOONER *****
C
DO 200 I = 1, NPTS
  TOUT = TPR(I)
  IF (I.EQ. 1) GO TO 190

C
C ***** HALT INTEGRATION IN CERTAIN CASES *****
C
  IF (SWITCH .NE. 4) GO TO 180
  IF (OPTION .EQ. 1) GO TO 180
  IF (OPTION .EQ. 2 .AND. SIMNUM .LE. 2 .AND.
X    TOUT .LT. TB + 5.00) GO TO 180

C
  IF (TOUT .GT. TB + .50000) GO TO 210

C
180  CALL DRIVE(IEQ,T0,H0,Y0,TOUT,EPS,MF,INDEX)
    IF (INDEX .NE. 0) GO TO 220
190  KEX = KEXCES(TPR(I))
    WRITE (6,195) TPR(I), Y0(1), Y0(17), Y0(18), Y0(19), KEX
195  FORMAT (' ',F10.3,5D17.6)
200  CONTINUE

C
210  WRITE (6,215) NSTEP, NFE, NJE
215  FORMAT ('ONSTEP = ',I8,' NFE = ',I8,' NJE = ',I8)
    GO TO 230

C
C ***** ERROR EXIT *****
C
220  WRITE (6,225) INDEX
225  FORMAT ('OINDEX = ',I3)
230  CONTINUE
    STOP
    END

```

BLOCK DATA

IMPLICIT REAL*8(A-Z)

COMMON /CON/ ADC,ALPHA,BETASM,CP,IC,L,OMEGA,S,BETA(15),LAMBDA(15)

C

C

C

***** INITIALIZE CONSTANTS IN COMMON BLOCK CON *****

REAL*8 ADC/1.D5/, ALPHA/2.313D0/, BETASM/.750746D-2/, CP/4.7D-9/,
X IC/1.D-4/, L/5.3D-4/, OMEGA/.1257D0/

C

REAL*8 BETA / .273D-3, .7475D-3, .25675D-2, .1274D-2, .14235D-2,
X .2145D-3, .651D-3, .204D-3, .7D-4, .336D-4, .207D-4, .234D-4,
X .323D-5, .103D-5, .5D-6 /

C

REAL*8 LAMBDA / -.301D1, -.114D1, -.301D0, -.111D0, -.305D-1,
X -.124D-1, -.227D0, -.169D-1, -.481D-2, -.150D-2, -.428D-3,
X -.117D-3, -.437D-4, -.363D-5, -.626D-6 /

C

END

```

SUBROUTINE DIFFUN(IEQ,T,Y,YDOT)
IMPLICIT REAL*8(A-H,K,L,O-Z)
REAL*8 IC
DIMENSION Y(IEQ),YDOT(IEQ)
COMMON /CON/ ADC,ALPHA,BETASM,CP,IC,L,OMEGA,S,BETA(15),LAMBDA(15)

```

```

C ***** DIFFUN COMPUTES THE FUNCTION YDOT = F(Y,T), THE RIGHT
C HAND SIDE OF THE O. D. E. SYSTEM *****
C

```

```

C KEX = KEXCES(T)
C

```

```

C SUM = 0.D0

```

```

C DO 10 I = 1, 15

```

```

10 SUM = SUM + ( - LAMBDA(I) * Y(I+1) )
C

```

```

C YDOT(1) = ( ( KEX - BETASM * ( 1.D0 + KEX ) ) / L ) * Y(1)

```

```

X + SUM + S

```

```

C DO 20 I = 1, 15

```

```

20 YDOT(I+1) = ( ( BETA(I) * ( 1.D0 + KEX ) ) / L ) * Y(1)

```

```

X + LAMBDA(I) * Y(I+1)
C

```

```

C YDOT(17) = - OMEGA * Y(17) - OMEGA * ADC * Y(18)

```

```

YDOT(18) = - ( 1.D-6 / ( 3.D0 * CP ) ) * Y(1) + YDOT(17)

```

```

X + IC * DEXP(ALPHA * Y(17)) / CP

```

```

YDOT(19) = - 20.D0 * Y(19) + 1152.D0 * YDOT(17)

```

```

RETURN

```

```

END

```

```

SUBROUTINE PEDERV(IEQ,T,Y,PD,IEQ0)
IMPLICIT REAL*8(A-H,K,L,O-Z)
REAL*8 IC
DIMENSION PD(IEQ0,IEQ0),Y(IEQ0)
COMMON /CON/ ADC,ALPHA,BETASM,CP,IC,L,OMEGA,S,BETA(15),LAMBDA(15)

```

```

C ***** PEDERV COMPUTES THE PARTIAL DERIVATIVES OF F WITH
C RESPECT TO Y, PD(I,J) = D F(I) / D Y(J),
C 1 .LE. I,J .LE. IEQ *****
C

```

```

KEX = KEXCES(T)

```

```

C
DO 10 I = 1, 19
  DO 10 J = 1, 19
10   PD(I,J) = 0.D0

```

```

C
PD(1,1) = ( KEX - BETASM * ( 1.D0 + KEX ) ) / L

```

```

DO 20 I = 2, 16

```

```

  PD(I,1) = BETA(I-1) * ( 1.D0 + KEX ) / L

```

```

  PD(1,I) = - LAMBDA(I-1)

```

```

20   PD(I,I) = LAMBDA(I-1)

```

```

C
PD(17,17) = - OMEGA

```

```

PD(17,18) = - OMEGA * ADC

```

```

PD(18,1) = - 1.D-6 / ( 3.D0 * CP )

```

```

PD(18,17) = - OMEGA + IC * ALPHA * DEXP(ALPHA * Y(17)) / CP

```

```

PD(18,18) = - OMEGA * ADC

```

```

PD(19,17) = - 1152.D0 * OMEGA

```

```

PD(19,18) = - 1152.D0 * OMEGA * ADC

```

```

PD(19,19) = - 20.D0

```

```

RETURN

```

```

END

```

```

FUNCTION KEXCES(T)
  IMPLICIT REAL*8(A-H,K,L,O-Z)
  REAL*8 SLOPE1 / 8.5D-4 / , SLOPE2 / -.702D0 / , OP3FAC(8)
  INTEGER OPTION,SIMNUM,SWITCH
  COMMON /CONTRL/ OPTION,SIMNUM,SWITCH
  COMMON /KXINFO/ E2,DENNEU,KEX0,TB
  DATA OP3FAC / .3D0, .5D0, .7D0, 1.0D0, 1.5D0, 2.0D0, 2.5D0, 3.0D0 /

```

```

C ***** KEXCES COMPUTES THE EXCESS REACTIVITY AT TIME T.
C          VALUES OF "SWITCH" AND THEIR MEANINGS ARE:
C          1 : FOR 0 .LE. T .LE. TA (SWITCH IS SET TO 1 IN MAIN)
C          2 : FOR T .EQ. TA (PERIOD TRIP OCCURRED)
C          3 : FOR T .EQ. TA (POWER TRIP OCCURRED)
C          4 : FOR T .GT. TA (SWITCH IS SET TO 4 IN KEXCES)

```

```

C          KEXCES IS COMPUTED AT FIRST ASSUMING SWITCH = 1 *****

```

```

C          GO TO ( 10, 20, 30 ), OPTION

```

```

C          10 KEXCES = KEX0 + SLOPE1 * T
C             GO TO 40

```

```

C          20 KEXCES = .0002D0 * SIMNUM
C             IF ( T .EQ. 0.0D0 ) KEXCES = 0.0D0
C             GO TO 40

```

```

C          30 KEXCES = .0025D0 * T * OP3FAC(SIMNUM)
C             IF ( KEXCES .GT. .0218D0 ) KEXCES = .0218D0

```

```

C          40 GO TO ( 50, 60, 70, 90 ), SWITCH
C

```

50 RETURN

C 60 WRITE (6,65) T,E2

65 FORMAT ('0FUNCTION KEXCES HAS JUST FOUND SWITCH SET TO 2 ; ',
X 'PERIOD TRIP OCCURRED.'/' T = ',D25.16, ' E2 = ',D25.16)
GO TO 80

C 70 WRITE (6,75) T,DENNEU

75 FORMAT ('0FUNCTION KEXCES HAS JUST FOUND SWITCH SET TO 3 ; ',
X 'POWER TRIP OCCURRED.'/' T = ',D25.16, ' N(T) = ',D25.16)

C 80 TA = T

C ***** NOTE: TA WILL ONLY BE ACCURATE TO THE PRINTOUT MESH
C OR HMAX (SET IN DRIVE), WHICHEVER IS SMALLER *****
TB = TA + .350D0
SWITCH = 4
RETURN

C 90 IF (T .GT. TB) GO TO 100
RETURN

100 KEXCES = KEXCES + SLOPE2 * (T - TB)
IF (KEXCES .LT. -.15D0) KEXCES = -.15D0
RETURN
END

Modifications to SUBROUTINE DRIVE:

```
REAL*8 E2,DENNEU,KEX0,TB
INTEGER OPTION,SIMNUM,SWITCH
COMMON /KXINFO/ E2,DENNEU,KEX0,TB
COMMON /CONTRL/ OPTION,SIMNUM,SWITCH
C ***** STATEMENTS ON CARDS DRI02200, DRI02310, AND DRI02340
C IN SUBROUTINE DRIVE HAVE BEEN REPLACED BY,
C HMAX = DABS(T0-TOUT) / 2.00
C 20 HMAX = DABS(TOUT-TOUTP) / 2.00
C 25 HMAX = DABS(TOUT-TOUTP) / 2.00
C RESPECTIVELY, SO THAT HMAX IS ONE HALF OF THE CURRENT
C PRINTOUT MESH SIZE. *****

C ***** DRIVE CODE SETTING SWITCH: CODE ADDED HERE IS TO TEST
C WHETHER EITHER E2 OR N(T) EXCEEDS ITS TRIP VALUE. IF
C SO, THE EXCESS REACTIVITY FUNCTION WILL HAVE A DELAYED
C CHANGE, I. E., SHUTDOWN. INFORMATION IS CONVEYED
C THROUGH VARIABLES SWITCH, E2, AND DENNEU. *****
C
C IF ( SWITCH .NE. 1 ) GO TO 65
C IF ( Y(19,1) .GT. ( 25.00/3.00 ) ) GO TO 61
C IF ( Y(1,1) .GT. 125.00 ) GO TO 62
C GO TO 65
C
C 61 SWITCH = 2
C E2 = Y(19,1)
C GO TO 65
C
C 62 SWITCH = 3
C DENNEU = Y(1,1)
C
C 65 CONTINUE
C
```

Before the first
executable statement.

After a normal return
from SUBROUTINE STIFF.

Acknowledgements

The authors would like to express their sincere appreciation to Wanda Lembke, who has so carefully typed this manuscript. Thanks are also due Patricia Nepps for help in preparing Tables 4.1 - 4.9.

References

1. Final Hazards Summary Report for Ames Laboratory Research Reactor, Contract No. AT(11-1)-855, AMF Atomics, Greenwich, Connecticut, March, 1964.
2. Hetrick, D. L., Dynamics of Nuclear Reactors, University of Chicago Press, London and Chicago, 1971.
3. Henrici, P., Discrete Variable Methods for Ordinary Differential Equations, John Wiley and Sons, New York, 1962.
4. Hindmarsh, A. C., GEAR: Ordinary Differential Equation System Solver, Lawrence Livermore Laboratory, Rept. UCID-30001 Rev. 3, December, 1974.
5. Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice Hall, Englewood Cliffs, N. J., 1971.
6. Bjurel, G., Dahlquist, G., Lindberg, B., Linde, S., and Oden, L., "Survey of Stiff Ordinary Differential Equations", Royal Institute of Technology, Stockholm, Department of Information Processing, Computer Science Report NA70.11, 1970.
7. Smith, B. T., Boyle, J. M., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B., Matrix Eigensystem Routines - EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, Springer-Verlag, New York, 1974.
8. General Electric Transistor Manual, Sixth Edition, p. 66, Syracuse, New York, 1962.