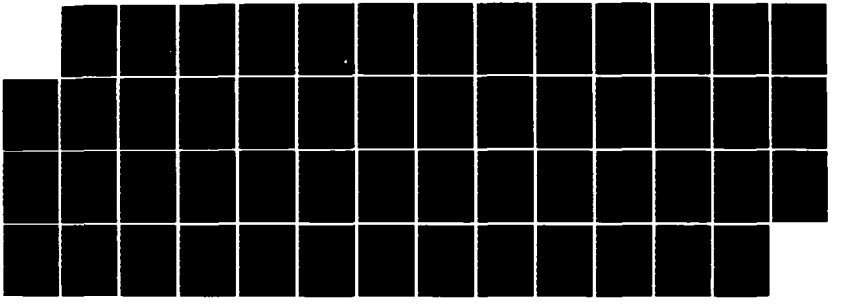
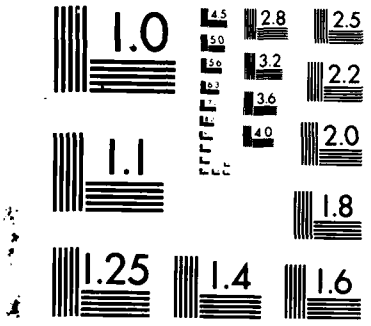
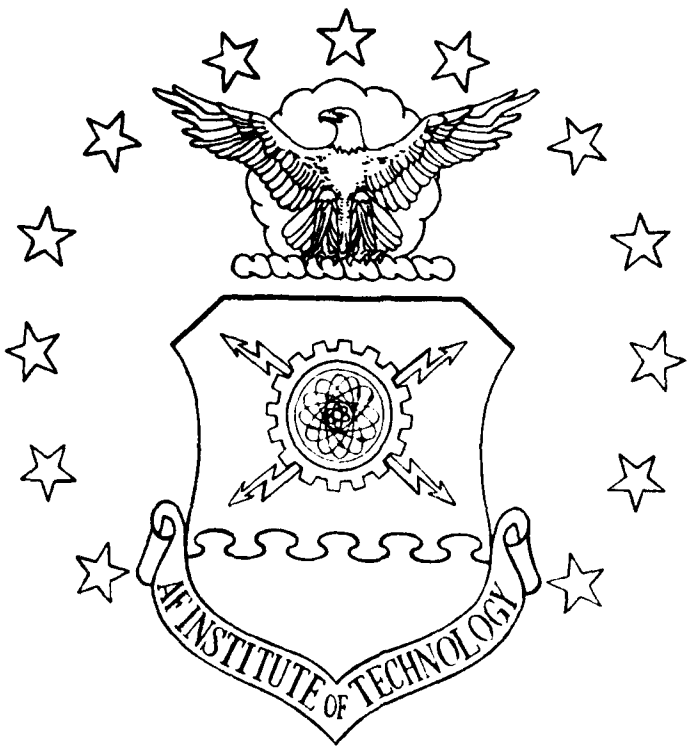
AD-A174 153 A COMPARISON STUDY OF THE EIGENVALUE METHOD FOR THE SOLUTION OF THE TRANS..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH D B GEE JAN 86 1/1

UNCLASSIFIED AFIT/GNE/ENP-86M-6 F/G 20/13 NL







A COMPARISON STUDY OF THE EIGENVALUE METHOD FOR THE SOLUTION OF THE TRANSIENT HEAT CONDUCTION EQUATION

THESIS

David B. Gee
Second Lieutenant, USAF

AFIT/GNE/ENP/86M-6

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GNE/ENP/86M

A COMPARISON STUDY OF THE EIGENVALUE METHOD FOR THE SOLUTION OF THE TRANSIENT HEAT CONDUCTION EQUATION

THESIS

David B. Gee
Second Lieutenant, USAF

AFIT/GNE/ENP/86M-6

Approved for public release; distribution unlimited

AFIT/GNE/ENP/86M-6

A COMPARISON STUDY OF THE EIGENVALUE METHOD *FOR THE SOLUTION OF* THE TRANSIENT HEAT CONDUCTION EQUATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Nuclear Engineering

David B. Gee, B. S.

Second Lieutenant, USAF

January 1986

Preface

This thesis has been written for a degree requirement. I am no mathematician and never intend to be. This study looks at numerical methods from an engineer's view, a tool to be used in solving problems. This paper has given me much needed experience in writing.

Many thanks are given to Dr. Kaplan for being so patient with me. Thanks must also go to Dr. Jones of the math department for his time in answering the many questions I had. Thanks must also be given to Captain Landry who allowed me unlimited use of the programs he ran for his study.

One person has made this all possible. For without her I am sure I would not have gotten this done. My wife, Sandra, has given me the encouragement and love needed to complete this paper. Most of all, she has been both mother and father to our two boys, giving me the time needed to get my studies done.

Table of Contents

	Page
Preface	ii
List of Figures	iv
List of Tables	v
Abstract	vi
I. Introduction	1
II. Theory	3
Implicit Method	3
Crank-Nicolson	4
Method of Lines	5
DREBS	6
DVERK	7
DGEARS	10
Eigenvalue Method	14
III. Applications	16
Error Analysis	16
Symmetric Slab Problem	17
Results	19
Full Grid Slab Problem	22
Cylindrical Problem	23
Result	29
IV. Conclusions and Recommendations	31
Conclusions	31
Recommendations	31
Appendix: Subroutines Used in the Study	33
Bibliography	39
Vita	42

List of Figures

Figure		Page
1.	Geometric Interpretation of RK Method	8
2.	Geometric Interpretation of Predictor Formula	11
3.	Geometric Interpretation of Corrector Formula	13
4.	Grid Scheme of Slab Problem	18
5.	Grid Scheme of Cylindrical Problem	28

List of Tables

Table		Page
I.	Time Comparison of Symmetric Slab Problem	20
II.	Error Comparisons of the Symmetric Slab Problem	21
III.	Time Comparison For Full Grid Problem	23
IV.	Error Comparison For Full Grid Problem	24
V.	Time Comparison For Cylindrical Problem	29
VI.	Error Comparison of Cylindrical Problem	30

Abstract

This is a comparison study of the abilities of the eigenvalue method as a numerical method in solving the transient heat conduction equation. The eigenvalue method was compared to five other numerical methods: Runge-Kutta, Gears, extrapolation, fully implicit, and Crank-Nicolson. These methods were used to solve three physical problems. The first is a two dimensional slab which takes advantage of the symmetry of the problem. The second is the same slab problem without taking advantage of the symmetry. And the third is a cylindrical problem taking full advantage of symmetry.

The scope of the study is to see which methods take less computer time while maintaining sufficient accuracy. The time it takes the computer to totally execute the *program* was used as the time comparison basis. The accuracy is a comparison of the exact solution to the numerical solution. A root mean square average of all the grid points per time step is used.

The results of the study were surprising. The accuracy of the eigenvalue method is not any better than that of the Crank-Nicolson method. The computer times show that the eigenvalue is not the fastest for short transient times. A long transient problem with nonlinear terms was not used in this study.

A COMPARISON STUDY OF THE EIGENVALUE METHOD FOR THE SOLUTION OF THE TRANSIENT HEAT CONDUCTION EQUATION

I. Introduction

In the field of nuclear engineering, the study of reactor thermal-hydraulics is coupled to the study of reactor neutronics. This is because reactor power is directly proportional to neutron population. Active core volumes are getting greater than 300,000 liters while the neutronics people are calculating the multiplication factor to four decimal place accuracy (8:634).

Solving these problems takes massive computer codes, such as COBRA, to solve just the thermalhydraulic problem. Solution of the transient heat conduction equation is only a small part of the massive problem. By solving parts of the big problem faster, the overall solution could be done faster and more accurately.

The scope of the study is limited. Density, specific heat, and thermal conductivity have all been assumed constant. These were then lumped into the single constant, thermal diffusivity. This makes the solution limited but it allows the use of dimensionless parameters. None of the problems considered had a heat source. The temperature difference used was 200° K.

The problems were investigated using Cartesian and cylindrical coordinates. Three problems were looked at. Two problems were actually

one problem solved two different ways. One way took advantage of symmetry and the other way did not. The problem was a two dimensional slab with Dirichlet boundary conditions. The third problem was a cylindrical problem with Dirichlet boundary conditions. Full advantage was taken of the symmetry in the cylindrical problem.

Six different numerical schemes were used to solve the problems: eigenvalue, Runge-Kutta, extrapolation, Gears, implicit, and Crank-Nicolson methods. The computer used was AFIT's SSC VAX 11/780. Each computer run was timed for the amount of CPU time and wall clock time necessary to execute the command.

II. Theory

Solution of the transient heat conduction equation is important in many engineering problems. It is of even greater importance for reactor design since the amount of heat a reactor produces is directly coupled to the amount of heat removed. This study looked at different numerical methods for solving the transient heat conduction equation:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \tag{1}$$

Here T is temperature, t is time, α is thermal diffusivity, and ∇^2 is the Laplacian operator.

The transient heat conduction equation was solved by three different methods: implicit, Crank-Nicolson, and the method of lines. The theory of each method will be described briefly.

Implicit Method

The implicit method (IM) is in common use among engineers. This is because of its unconditional stability. The method generates systems of simultaneous equations due to the differencing equations used. The differencing equations discretize the spatial variables with respect to the next time step using a central difference (equations (2) and (3)). A standard forward difference is used on the time derivative (equation (4)).

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j}^{k+1} - 2T_{ij}^{k+1} + T_{i-1,j}^{k-1}}{(\Delta x)^2} \tag{2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1}^{k+1} - 2T_{ij}^{k+1} + T_{i,j-1}^{k+1}}{(\Delta y)^2} \tag{3}$$

$$\frac{\partial T}{\partial t} = \frac{T_{i,j}^{k+1} - T_{i,j}^{k}}{(\Delta t)} \tag{4}$$

where k is the time step, i and j are spatial nodal points.

Plugging the above difference equations into equation (1) and solving for $T_{i,j}^{k}$:

$$T_{i,j}^{k} = (1+4C)\, T_{i,j}^{k+1} - C \left[T_{i+1,j}^{k+1} + T_{i-1,j}^{k+1} + T_{i,j+1}^{k+1} + T_{i,j-1}^{k+1} \right] \tag{5}$$

where $(\Delta x)^2 = (\Delta y)^2 \approx h^2$ and $C = \frac{\alpha(\Delta t)}{h^2}$. The above equation produces a set of simultaneous equations with errors of $0(h_t) + 0(h_{x,y})^2$.

Crank-Nicolson

A method with improved accuracy and unconditional stability is the Crank-Nicolson (CN) method. The CN method uses an averaging scheme of central time differences; see equations (6) and (7).

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{2(\Delta x)^2} \left[T_{i+1,j}^{k+1} - 2T_{i,j}^{k+1} + T_{i-1,j}^{k+1} + T_{i+1,j}^{k} - 2T_{ij}^{k} + T_{i-1,j}^{k} \right] \tag{6}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{1}{2(\Delta y)^2}\left[T^{k+1}_{i,j+1} - 2T^{k+1}_{i,j} + T^{k+1}_{i,j-1} + T^{k}_{i,j+1} - 2T^{k}_{i,j} + T^{k}_{i,j-1}\right] \tag{7}$$

The time derivative is a central difference at the $k + \frac{1}{2}$ time node. Taking equations (4), (6), and (7) and substituting them into equation (1) and putting all unknown temperatures (k+1) on the left side, results in:

$$(1+2C)\ T^{k+1}_{i,j} - \frac{1}{2}C\left[T^{k+1}_{i+1,j} + T^{k+1}_{i-1,j} + T^{k+1}_{i,j+1} + T^{k+1}_{i,j-1}\right]$$

$$= (1-2C)\ T^{k}_{i,j} + \frac{1}{2}C\left[T^{k}_{i+1,j} + T^{k}_{i-1,j} + T^{k}_{i,j+1} + T^{k}_{i,j-1}\right] \tag{8}$$

where $C = \frac{\alpha(\Delta t)}{h^2}$. Again we develop a set of simultaneous equations. The important point is the error has improved to $0(h_t)^2 + 0(h_{xy})^2$.

Method of Lines

A numerical method, uncommon for solving partial differential equations, is the methods of lines (MOL). This method changes a partial differential equation into an ordinary differential equation (ODE) of the form:

$$\frac{dy}{dt} = f(t,y) \tag{9}$$

Numerous books are available for methods of solving equation (9) (1;10;20). A formal proof of turning equation (1) into the form of

equation (9) is in either Shih and Skladany or Landry (25:411-412;19:11-15). In practice, the spatial variables are discretized leaving the time derivative alone. Here, the spatial variables will be discretized with a central difference.

$$\frac{\partial^2 T}{\partial x^2} = \frac{T^k_{i+1,j} - 2T^k_{i,j} + T^k_{i-1,j}}{(\Delta x)^2} \tag{10}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T^k_{i,j+1} - 2T^k_{i,j} + T^k_{i,j-1}}{(\Delta y)^2} \tag{11}$$

After differencing the spatial variables and substituting into equation (1), the resulting equation is

$$\left.\frac{dT}{dt}\right|_{i,j} = \frac{\alpha}{h^2}\left[T^k_{i+1,j} + T^k_{i-1,j} + T^k_{i,j+1} + T^k_{i,j-1} - 4T^k_{i,j}\right] \tag{12}$$

This is an ODE. The three IMSL (15) routines used were DREBS, DVERK, and DGEARS. They use extrapolation, Runge-Kutta, and predictor-corrector methods, respectively. All three routines were designed to solve ODE's of the form of equation (9) with an initial condition supplied.

DREBS. Extrapolation is the method used in this IMSL routine. The actual method is a modified form of a subroutine, DESUB, developed by Bulirsch and Stoer (2:10-13). It uses rational function for the extrapolation. The exact error figure is not given in the documentation. Error bounds and stability are still unanswered questions for extrapolation methods (10:101).

No documentation explaining the theory of the method was available in English. Therefore, only the basic algorithm used by Bulirsch and Stoer will be given (2:3).

$$t_{k+1} = t_k + (\Delta t) \qquad k=0,1, \ldots, L-1 \qquad (13)$$

$$\eta\left(t_1, (\Delta t)\right) = y_o + (\Delta t)\, f\,(t_o, y_o) \qquad (14)$$

$$\eta\left(t_{k+1}, (\Delta t)\right) = \eta\left(t_{k-1}, (\Delta t)\right) + 2(\Delta t)\, f\left(t_k, \eta\left(t_k, (\Delta t)\right)\right)$$

$$k=1,2, \ldots, L-1 \qquad (15)$$

$$S\left((\Delta t), t\right) = \frac{1}{2}\left[\eta\left(t_k, (\Delta t)\right) + \eta\left(t_{L-1}, (\Delta t)\right) + (\Delta t) f\left(t_L, \eta\left(t_L, (\Delta t)\right)\right)\right] \qquad (16)$$

$$T\left((\Delta t), t\right) = S\left(\frac{(\Delta t)}{2}, t\right) \qquad (17)$$

DVERK. The RK method is one of the most common methods for the solution of ordinary differential equations. It is a one step method, only requiring the information at time step k to get the information at time step k+1. Therefore the method is self starting. The RK method does not require the evaluation of any derivatives of f(t,y).

An example of a second order RK method is the improved Euler method. With the help of Figure 1, a geometric interpretation on how the method

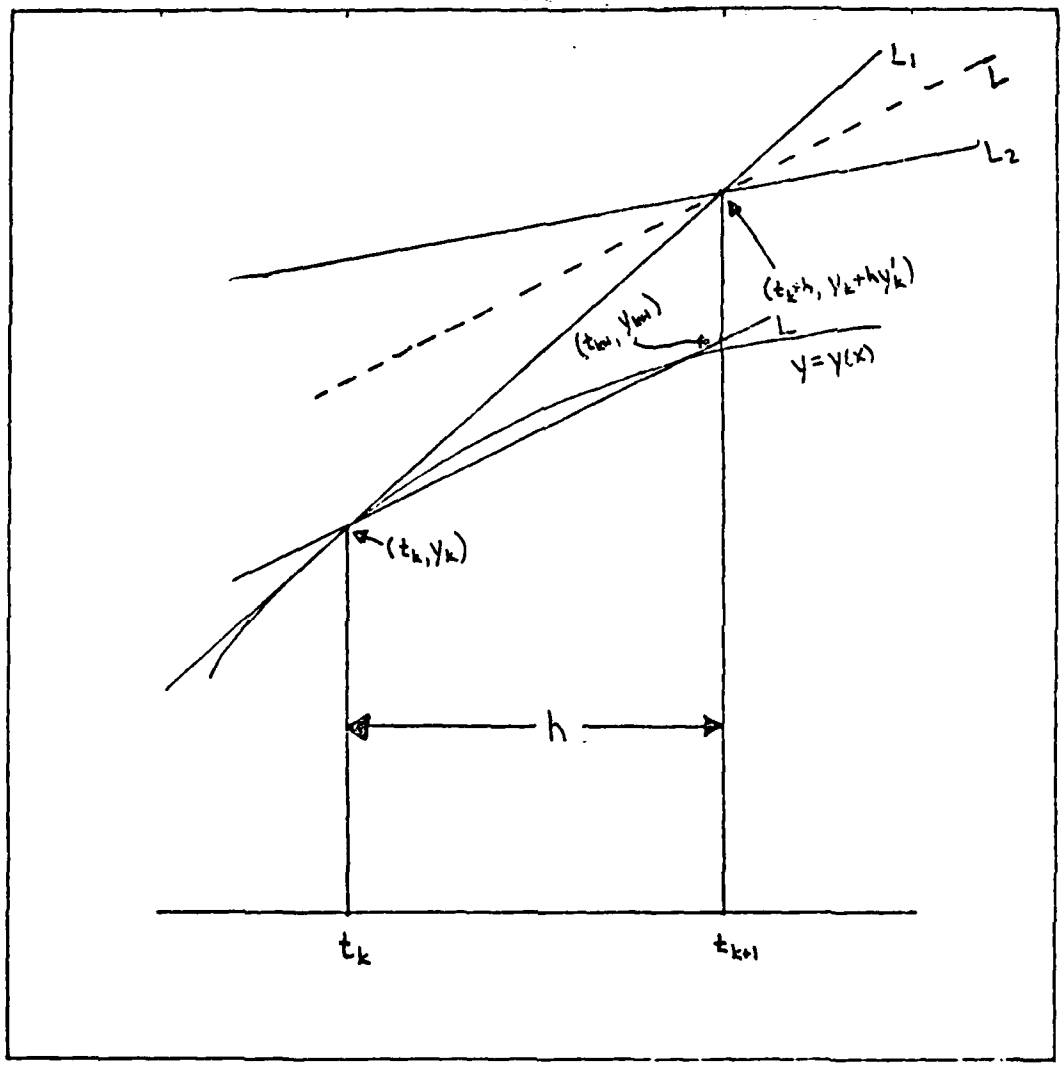


Figure 1. Geometric Interpretation of RK Method

works can be shown. What is done is an averaging of slopes. The point $(t_k + h, y_k + hy_k)$ is computed using Eulers method giving the slope of L_1. Now the slope of the curve is computed at $(t_k + h, y_k + hy_k)$, this gives the slope of L_2. The slopes of L_1 and L_2 are averaged giving the slope of $\bar{L}$. A line is drawn parallel to $\bar{L}$, L. Where L intersects the ordinate axis at $t = t_{k+1} = t_k + h$ is the next point (t,y).

The slope of $\bar{L}$ and L is given by

$$A_k = \frac{1}{2} \quad f(t_k, y_k) + f(t_k + h, y_k + hy_k) \tag{18}$$

where,

$$y_k = f(t_k, y_k) \tag{19}$$

The equation of line L is given by:

$$y = y_k + (t-t_k)\, A_k \tag{20}$$

therefore,

$$y_{k+1} = y_k + h\, A_k \tag{21}$$

Equations (18), (19) and (21) define the improved Euler method, sometimes called Heun's method (22:317-320).

This method requires only two evaluations of the function f(t,y), once at (t_k, y_k) and once at $(t_k + h, y_k + hy_k)$. For comparison, a Taylor series of the same order would require three function evaluations, f(t,y), f_x and f_y where f_x and f_y are the partial derivatives of f(t,y) with respect to the subscript.

The most common RK method is a fourth order method (18:835). What makes this method so appealing is the fact that no derivatives need to be calculated, the function itself only needs to be evaluated, and the method is self starting. The method only has partial stability though. Gears method, which is a predictor-corrector method, does have complete stability and an ability to handle stiff systems of differential equations.

DGEARS. Predictor-corrector methods are another common numerical technique for the solution of ODE's. The major draw back with these methods is that they are not self starting. A method like RK is needed to start predictor-corrector methods. Once enough points are known the predictor-corrector technique is employed for the rest of the solution. The reason predictor-corrector methods are not self starting is that they require values at prior nodal points.

For a brief explanation of the predictor-corrector formulas, a second order formula will be used. This should show how the method works and also give a comparison with the RK method just described.

First a predicted value of y_{k+1} is needed. This is done by finding the slope at (t_k, y_k) and a line L_1 is drawn through this point, as in Figure 2. Next draw a line L parallel to L_1 but through point (t_{k-1}, y_{k-1}). Where this line L crosses the value of t_{k+1} gives the first guess of $(t_{k+1}, y_{k+1}^{(0)})$. The superscript here (0) indicates that this is the first guess at y_{k+1}--a predicted value. The equation used to get $y_{k+1}^{(0)}$ is

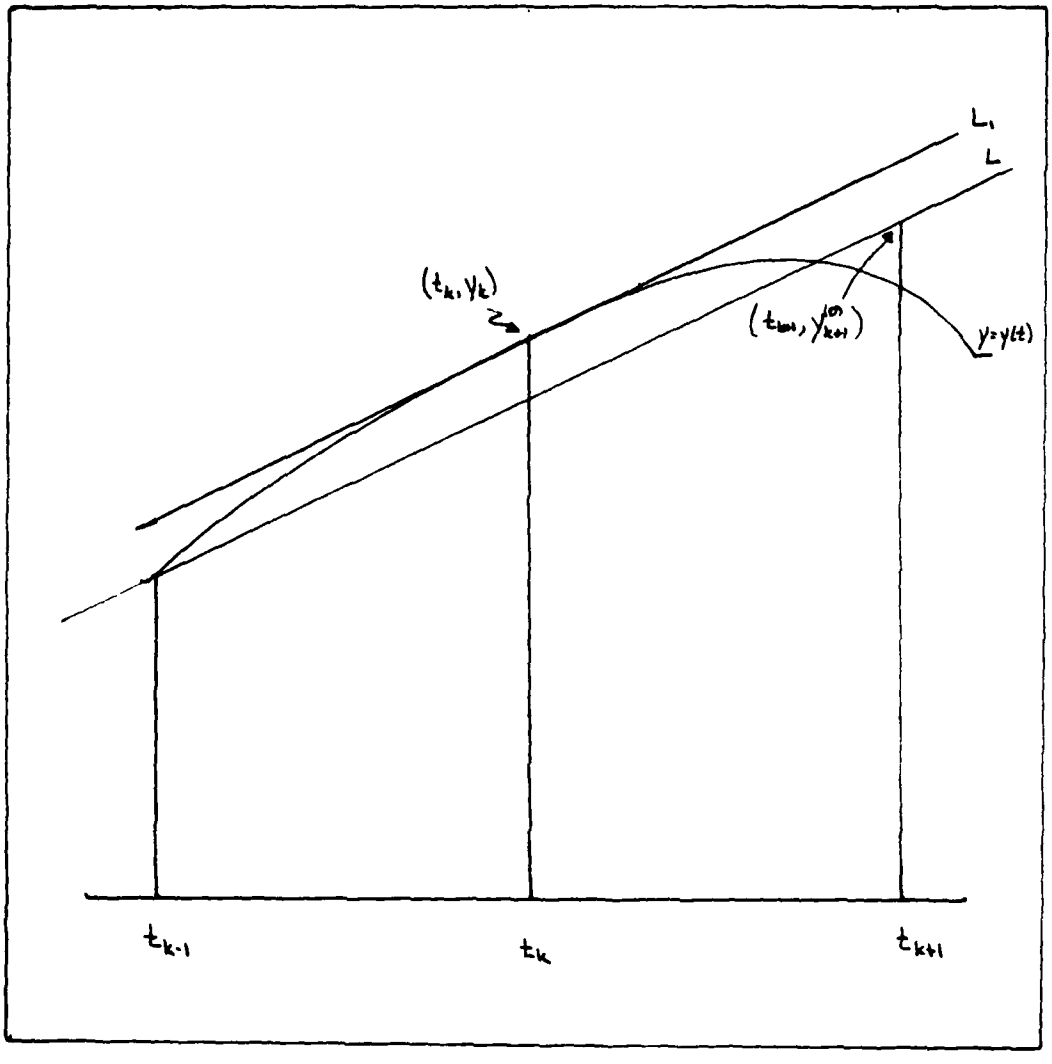


Figure 2. Geometric Interpretation of Predictor Formula

$$y_{k+1}^{(0)} = y_{k-1} + 2(\Delta t)\, f\,(t_k, y_k) \tag{22}$$

The y_{k-1} used in the above equation is why this method is not self starting.

Now it is time to correct the first value of $y_{k+1}^{(0)}$. A slope can be calculated at (t_{k+1}, y_{k+1}) which gives us line L_2 in Figure 3. Lines L_1 and L_2 are then averaged, getting line $\bar{L}$. A line L is then drawn parallel to $\bar{L}$ through the point (t_k, y_k). The intersection at $t = t_{k+1}$ will give the corrected value of y_{k+1} or $y_{k+1}^{(1)}$. With this new $y_{k+1}^{(1)}$ a better estimate of the slope of L_2 is gotten. Using this better approximation of the slope of L_2 a new estimate of y_{k+1} is gotten or $y_{k+1}^{(2)}$. In general it is given by

$$y_{k+1}^{(i)} = y_k + \frac{(\Delta t)}{2}\left[f\,(x_k, y_k) + f\;\; x_{k+1}, y_{k+1}^{(i-1)}\right] \tag{23}$$

The iterations are stopped when

$$\left| y_{k+1}^{(i)} - y_{k+1}^{(i-1)} \right| < \varepsilon \tag{24}$$

where ε is the convergence criteria of the user. It can be shown that the corrector formula does converge (22:332-333). If more than two iterations of equation (23) are required to meet the convergence criteria, then the step size is too big and should be reduced.

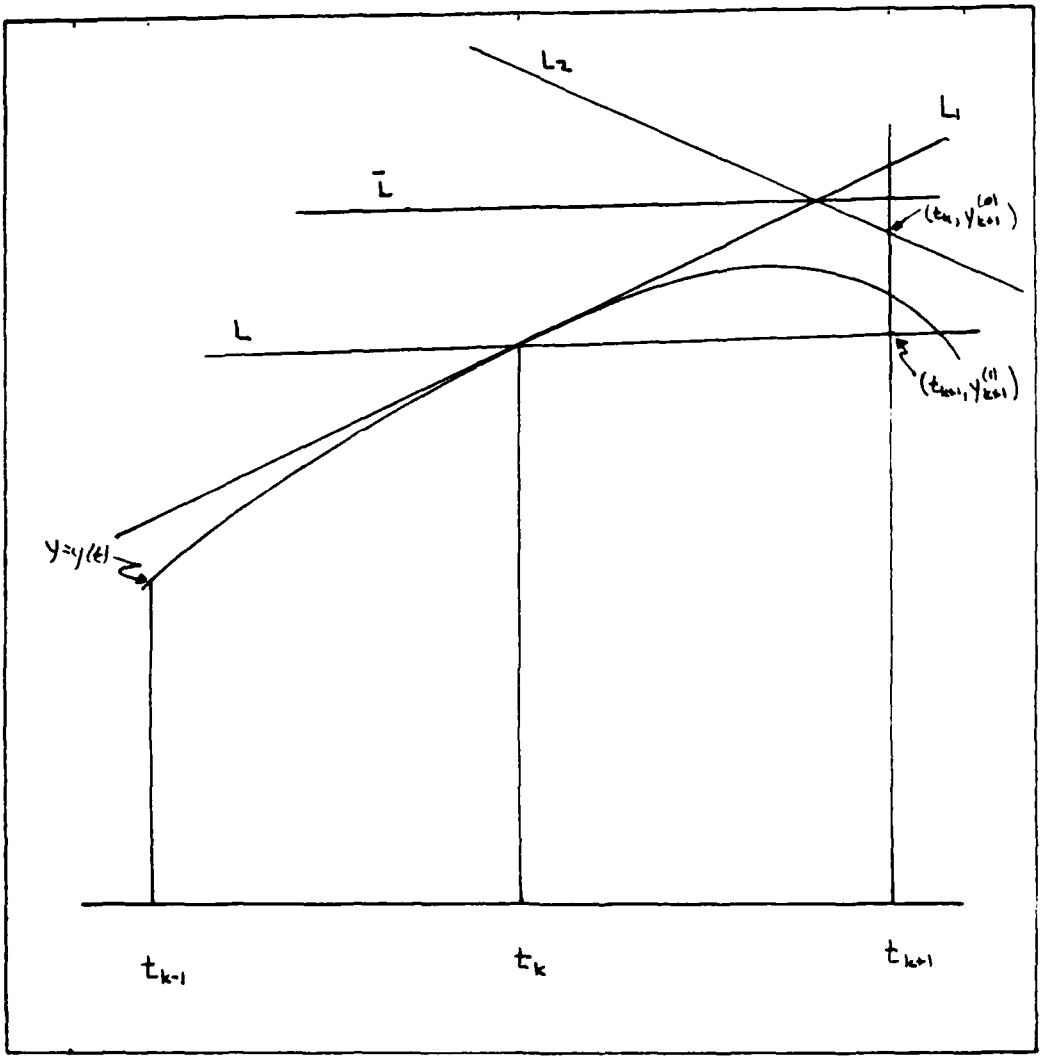


Figure 3. Geometric Interpretation of Corrector Formula

Equations (22) and (23) define a second order predictor-corrector method. There are obvious similarities between the RK and predictor-corrector methods.

Eigenvalue Method. The eigenvalue method (EM) should fall under the MOL for it also turns a partial differential equation into an ODE. A formal proof, as stated earlier, of why the EM works may be found in either Shih and Skladany or Landry.

Taking mainly from Landry: once the differential equation is in the matrix form,

$$\{\dot{\theta}\} = [D]\ \{\theta\} \tag{25}$$

Where $\{\}$ is used for an n component vector and $[\ ]$ are used for an n x n matrix. This is the matrix form of equation (9). Where the solution has the form of

$$\{\theta(t)\} = \sum_i \{c_i\} \exp(\{\lambda_i\} t) \tag{26}$$

$\{c_i\}$ are undetermined constants and $\{\lambda_i\}$ are the eigenvalues of the system of equations. The constants $\{c_i\}$ turn out to be another constant $\{a_i\}$ multiplied by the eigenvector $\{v_i\}$. Therefore, the final solution in matrix form is

$$\{T_i\} = \{T_f\} + \sum_{i,j} \{a_i\} \{v_i^j\} \exp(\{\lambda_i\} t) \tag{27}$$

where t_f are the boundary conditons, v_i^j is the jth component of the ith eigenvector associated with the ith eigenvalue. Expanded the equation looks like

$$T_i(t) = a_1 v_i^{(1)} \exp(\lambda_1 t) + \ldots + a_j v_i^{(j)} \exp(\lambda_j t) + T_f \quad (28)$$

First, one needs to set up the coefficient matrix [D] and solve it for the eigenvalues and eigenvectors. Next, one needs to solve the above equation for the coefficients $\{a_i\}$. Finally, one needs to sum all of the terms together and find the temperature T at some time t.

III. Applications

More emphasis was placed on solving the transient heat conduction equation by the MOL than by the IM or CN methods. Landry in his study and also Shih and Skladany in theirs did not put much emphasis on the MOL technique.

The basic problem looked at is a piece of material (slab or cylinder) at some initial temperature, T_o, suddenly having its outside boundaries raised instantaneously to a temperature of 200 degrees Kelvin above the initial temperature, T_o. Three problems were looked at: first, a slab was analyzed taking full advantage of symmetry; secondly, the same slab was analyzed without taking advantage of symmetry; and thirdly, a cylinder was analyzed taking full advantage of symmetry.

Error Analysis

To help evaluate the method's accuracy, the exact solution was also found for each particular problem. All the grid points, for a given time step, should be considered to get an overall error (27:1023). The exact solution is used along with the numerical solution at each grid point to get a root mean square (RMS) average value for the time step.

$$E_k = \frac{\left(\sum_{i=1}^{N} \left(u_i - u_i^* \right)^2 \right)^{\frac{1}{2}}}{N} \qquad (29)$$

where u_i is the value of the temperature at the ith nodal point, u_i^* is the exact solution at that point, and N is the total number of grid points.

In the exact solution, only 20 terms were used. This is because about 15 terms are needed to get it to converge to 10^{-10} accuracy (19:27). This accuracy is more than sufficient for engineering applications where 10^{-3} to 10^{-4} accuracy is usually the norm.

Symmetric Slab Problem

Figure 4 shows the grid system used. It has already been noted in Landry that the center of the slab should have a node also (19:18). Obviously, it is a crude assumption that the center of the slab is at the same temperature as node 20. It is hard to see why a node 21 was not used. The original authors were taking advantage of the symmetry in the problem.

The exact solution of this problem is

$$\Theta(x,y,t) = \sum_{n} \sum_{m} \frac{16}{nm\pi^2} \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi y}{L}\right) \exp\left(-\frac{\alpha t}{L^2}(n^2+m^2)\right) \tag{30}$$

The IM method used equation (5) directly, and where i is the x spatial nodes and j is the y spatial nodes. The CN method did not use equation (8) directly. The CN method solved equation (8) for $T_{i,j}^{k+1}$ first.

$$T_{i,j}^{k+1} = \frac{(1-2C)}{(1+2C)} T_{i,j}^{k} + \frac{C}{2(1+2C)} \left[T_{i+1,j}^{k+1} + T_{i-1,j}^{k+1} + T_{i,j+1}^{k+1} + T_{i,j-1}^{k+1} + T_{i+1,j}^{k} + T_{i-1,j}^{k} + T_{i,j+1}^{k} + T_{i,j-1}^{k} \right] \tag{31}$$

This equation was used in programing the CN method.

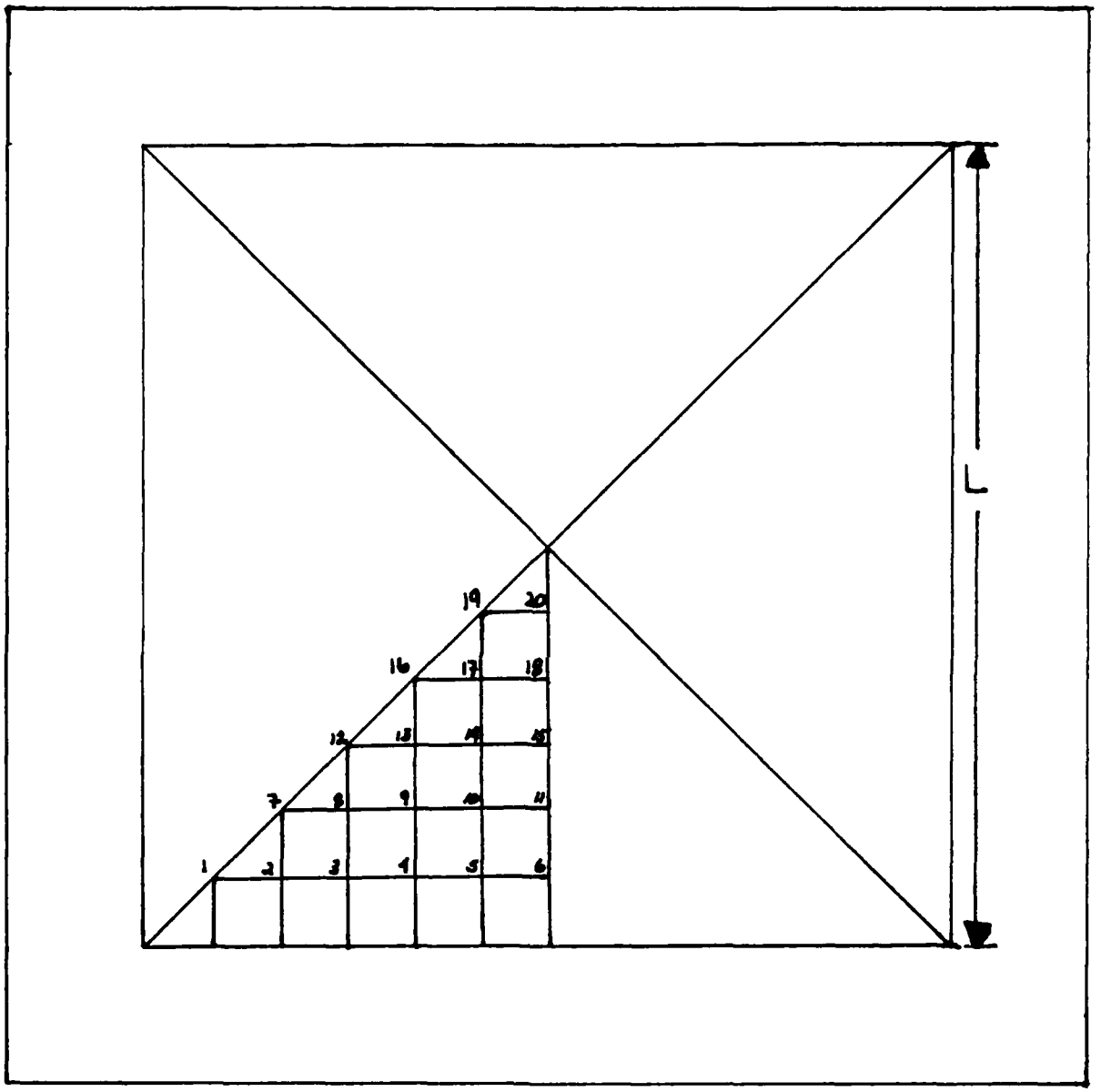


Figure 4. Grid Scheme of Slab Problem

The MOL uses equation (12) directly for the recursion formula. The difference between the IMSL routines and the EM is in the use of equation (12). The IMSL routines all need to have the first node on the boundary as the starting node. The IMSL routines all require an external subroutine be provided. This subroutine evaluates equation (12) at each nodal point (see Appendix). The EM only used equation (12) to generate the matrix [D] of equation (25).

Results. Two things are being looked at in the study: first is the time the program takes to run and second, the amount of error involved with the results.

To get the time required by the computer to execute the program the UNIX time command was used (23:473-474). The actual command used on AFIT's SSC computer was /bin/time instead of simply time. This returned *real, user, and system times all in seconds.* Real time is the wall clock time, user and system time are CPU time. The results of the timed computer runs can be seen in Table I.

The symmetric problem has shown some different results than those of Landry. The wall clock time is insignificant because the computer is a time sharing machine. The CPU time should not vary significantly, for this is the amount of time the computer spent working on the problem. Landry shows a CPU time for the EM, IM, and CN methods as 0.8, 4.2, and 6.4 respectively (19:29).

When the computer runs were timed, the whole program was timed, that is, the setting up of variables, reading in data from files, and writing the results to files. This could explain why the results of the time runs are so much greater than those of Landry.

TABLE I

Time Comparison of Symmetric Slab Problem

Method	CPU (sec)	Wall Clock (sec)
Eigenvalue	7.1	19.1
Runge-Kutta	0.9	1.9
Gears	1.8	2.9
Extrapolation	11.9	37.7
Implicit	50.6	166.9
Crank-Nicolson	50.5	157.5

The errors, as compared in Table II, can be seen to be as good as those obtained by the EM. Even the ODE solvers have errors comparable to that of the EM. These errors are shown with respect to the ten time steps, TBAR. The closer the method is to the exact solution, the smaller the value of the error.

Landry has shown that a comparison of the accuracies of the EM with the exact solution has errors of ±0.3% after the second time step (19:23). Before this time step errors, as compared with the exact solution, are considerably worse. Shih and Skladany noticed this problem and came up with an explanation for the cause (25:417).

> Since initially the heating rate $\partial T/\partial t$ and the change of the heating rate $\partial^2 T/\partial t^2$ are large, and since $\partial T/\partial t$ is related to the space derivatives by
>
> $$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$
>
> it follows that initially the value of $\nabla^4 T$ must be large because

TABLE II

Error Comparisons of the Symmetric Slab Problem

Method	TBAR									
	1	2	3	4	5	6	7	8	9	10
Eigenvalue Method (EM)	2.26	1.54	1.51	1.56	1.56	1.58	1.55	1.51	1.46	1.39
Runge-Kutta (RK)	2.26	1.54	1.51	1.56	1.58	1.58	1.55	1.51	1.46	1.39
Gears	2.26	1.54	1.51	1.56	1.58	1.58	1.55	1.51	1.46	1.4
Extrapolation	2.26	1.54	1.51	1.56	1.59	1.58	1.55	1.54	1.46	1.39
Implicit Method (IM)	2.28	1.5	2.54	0.98	0.74	0.54	0.38	0.26	0.16	0.09
Crank-Nicolson (CN)	2.06	1.90	2.14	2.16	2.3	2.8	3.6	4.56	5.61	6.62

$$\frac{\partial^2 T}{\partial t^2} = \alpha \nabla^2 \left(\frac{\partial T}{\partial t} \right) = \alpha^2 \nabla^4 T$$

Now we already know that the fourth derivative $\nabla^4 T$ is embedded in the leading truncation term in the second-order accurate finite-difference approximation.

This states that the truncation error of the initial heating rate will be large in any scheme using a central differencing approximation of the spatial variables.

Full Grid Slab Problem

Many engineering problems involve many more nodal points than just 20. Typically, more than a hundred nodal points are used in most applications. To get a better idea on how these methods compare in a big grid, the original slab problem was rerun without taking advantage of symmetry. The number of nodal points went from 20 to 121. All the methods were rerun for the full grid case except for the IM and CN methods. Problems arose in solving the set of simultaneous equations and the results were not correct.

All of the equations used for the full grid case were the same as in the symmetric case. The same IMSL routine used to solve the set of simultaneous equations in the symmetric case was also used in the full grid case.

Results. Table III shows the timed run results for all the MOL programs. Even though the IM and CN methods did not run properly, from a comparison of Landry's results of the symmetric and full grid times and the results that this study got for times, it is felt that the RK and Gears methods are faster than the IM and CN methods.

TABLE III

Time Comparison for Full Grid Problem

Method	CPU (sec)	Wall Clock (sec)
Eigenvalue	462.2	659.9
Runge-Kutta	6.9	25.0
Gears	8.4	24.0
Extrapolation	71.0	229.9

The error comparison was done for the full grid problem also. The results obtained do not agree with those of Landry. As can be seen in Table IV, the error for the EM is comparable to all the other methods (the values for the IM and CN methods came from Landry's thesis).

Cylindrical Problem

The most common nuclear engineering geometry is a cylinder. Therefore, most nuclear engineering problems are done in cylindrical coordinates. The third and final problem looked at was a solid cylinder. The differencing equations were all done in cylindrical coordinates. The first derivative of temperature with respect to the radius was set equal to zero at the center of the cylinder. The rate of change of temperature with respect to the angle Θ, was set equal to zero. The Laplacian of this problem then is

$$\nabla^2 T = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \qquad (32)$$

TABLE IV

Error Comparison for Full Grid Problem

Method	TBAR									
	1	2	3	4	5	6	7	8	9	10
Eigenvalue Method (EM)	7.14	4.52	3.64	2.95	2.31	1.78	1.35	1.01	0.74	0.53
Runge-Kutta (RK)	7.14	4.51	3.64	2.95	2.31	1.77	1.34	1.0	0.74	0.53
Gears	7.14	4.51	3.63	2.95	2.31	1.77	1.35	1.01	0.74	0.53
Extrapolation	7.14	4.51	3.64	2.95	2.31	1.77	1.34	1.0	0.74	0.53
Implicit Method (IM)	7.23	4.63	3.73	2.98	2.29	1.71	1.26	0.90	0.63	0.42
Crank-Nicolson (CN)	7.20	4.59	3.73	3.06	2.44	1.92	1.50	1.16	0.89	0.68

and when looking at points on the cylinder centerline it becomes

$$\nabla^2 T = 2\frac{\partial^2 T}{\partial r^2} + \frac{\partial^2 T}{\partial z^2} \tag{33}$$

The two in equation (33) comes from an expansion done on the middle term in equation (32). The value of r at the centerline and $\partial T/\partial r$ at the centerline are both zero giving an indeterminant value there. From an expansion of this term it can be shown that

$$\lim_{r \to 0} \frac{1}{r}\frac{\partial T}{\partial r} = \frac{\partial^2 T}{\partial r^2}$$

Therefore there is a two in equation (33).

A central difference was used on the center term of equation (32).

$$\frac{1}{r}\frac{\partial T}{\partial r} \approx \frac{1}{i(\Delta r)} \frac{T^k_{i+1,j} - T^k_{i-1,j}}{2(\Delta r)} = \frac{T^k_{i+1,j} - T^k_{i-1,j}}{2i(\Delta r)^2} \tag{34}$$

The recursion formula used in the cylindrical problem using the IM method was

$$T^i_{k,j} = (1+4C)\, T^{k+1}_{i,j} - C\left[\left(1 + \frac{1}{2i}\right) T^{k+1}_{i+1,j} + \left(1 - \frac{1}{2i}\right) T^{k+1}_{i-1,j} + T^{k+1}_{i,j+1} + T^{k+1}_{i,j-1}\right] \tag{35}$$

Then for the centerline

$$T^k_{i,j} = (1+6C)\, T^{k+1}_{i,j} - C\left[2T^{k+1}_{i+1,j} + 2T^{k+1}_{i-1,j} + T^{k+1}_{i,j+1} + T^{k+1}_{i,j-1}\right] \tag{36}$$

The formula used in the CN method was

$$(1+2C)\ T^{k+1}_{i,j} - \frac{1}{2}\,C\left[\left(1+\frac{1}{4i}\right)\ T^{k+1}_{i+1,j} + \left(1-\frac{1}{4i}\right)\ T^{k+1}_{i-1,j} + T^{k+1}_{i,j+1} + T^{k+1}_{i,j-1}\right] = (1-2C)\ T^{k}_{i,j} + \frac{1}{2}\,C\left[\left(1+\frac{1}{4i}\right)\ T^{k}_{i+1,j} + \left(1-\frac{1}{4i}\right) T^{k}_{i-1,j} + T^{k}_{i,j+1} + T^{k}_{i,j-1}\right] \tag{37}$$

Then when evaluating the problem on the centerline

$$(1+3C)\ T^{k+1}_{i,j} - \frac{1}{2}\,C\left[2T^{k+1}_{i+1,j} + 2T^{k+1}_{i-1,j} + T^{k+1}_{i,j+1} + T^{k+1}_{i,j-1}\right] = (1-3C)\ T^{k}_{i,j} + \frac{1}{2}\,C\left[2T^{k}_{i+1,j} + 2T^{k}_{i-1,j} + T^{k}_{i,j+1} + T^{k}_{i,j-1}\right] \tag{38}$$

The basic formula to solve the problem by the MOL is

$$\left.\frac{dT}{dt}\right|_{i,j} = \alpha\left[\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta r)^2} + \frac{T_{i+1,j} - T_{i-1,j}}{2i(\Delta r)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta z)^2}\right] \tag{39}$$

and when evaluating on the centerline

$$\left.\frac{dT}{dt}\right|_{i,j} = \alpha\left[2\left(\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta r)^2}\right) + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta z)^2}\right] \tag{40}$$

Full advantage was taken of symmetry. Figure 5 shows the grid scheme used in this study. When i=5 , then equations (36), (39), and (40) were used.

The exact solution of this problem used was (21:165)

$$\Theta(r,z,t) = \sum_{n}\sum_{m} A_n A_m J_o\left(\mu_n \frac{r}{R}\right) \cos\left(\mu_m \frac{z}{H}\right) \exp\left(-\left(\mu_n^2 + \mu_m^2 K_1^2\right) F_o\right) \tag{41}$$

where

$$A_n = \frac{2}{\mu_n J_1(\mu_n)}$$

$$A_m = (-1)^{m+1} \frac{2}{\mu_m}$$

$$\mu_m = (2m-1)\frac{\pi}{2}$$

$$K_1 = \frac{R}{H}$$

$$F_o = \frac{\alpha t}{R^2}$$

and μ_n are the roots of the Bessel function of the first kind of zero order.

This exact solution was found to have some problems. The values of the summation should be between zero and one. Some of the values were greater than one causing erroneous values in the solution (i.e., negative

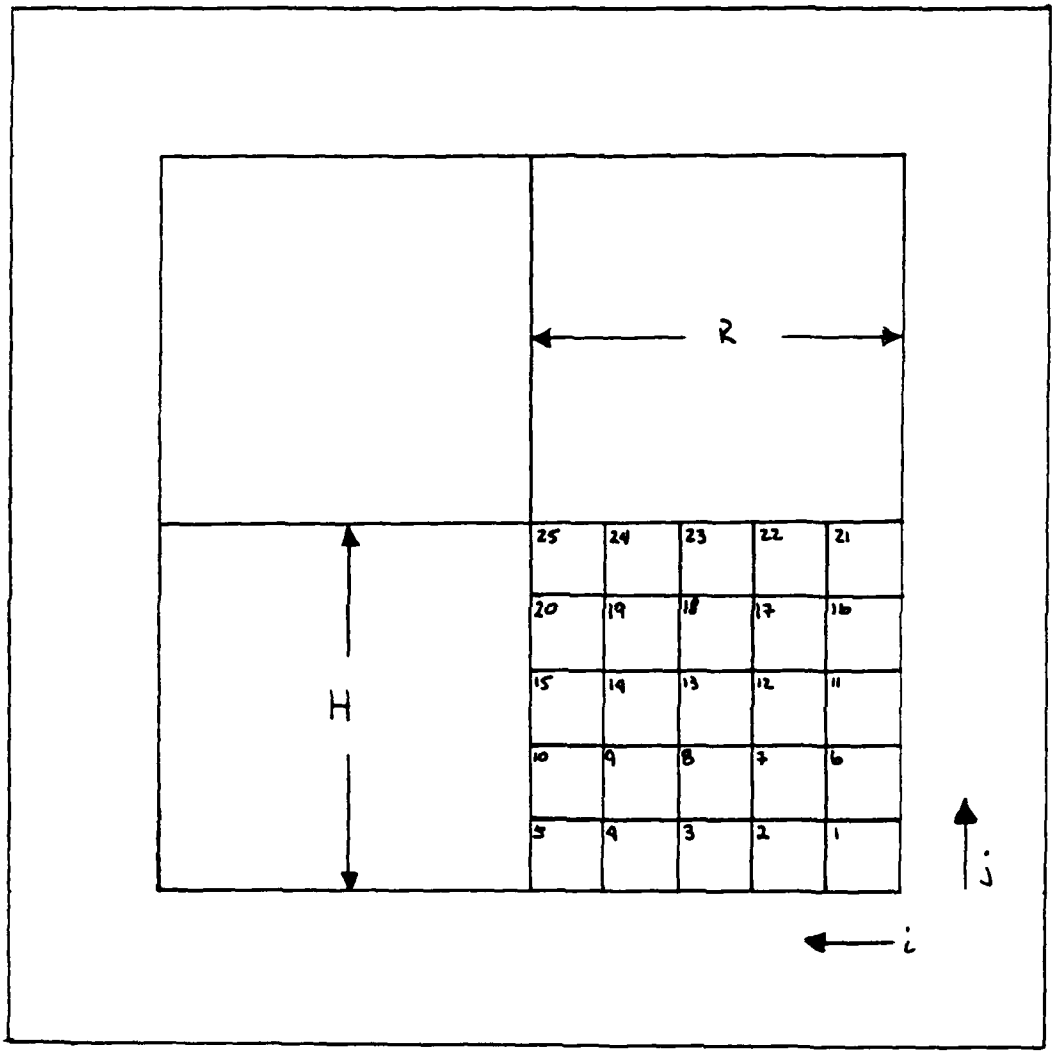


Figure 5. Grid Scheme of Cylindrical Problem

TABLE V

Time Comparison for Cylindrical Problem

Method	CPU (sec)	Wall Clock (sec)
Eigenvalue Method	7.4	31.4
Runge-Kutta	1.6	2.2
Gears	2.7	4.5
Extrapolation	15.8	83.1
Implicit Method	67.9	232.6
Crank-Nicolson	85.8	323.8

temperatures). All negative temperatures were set to T_o assuming no temperature change at that nodal point.

Result. The computer time needed to run this problem is found in Table V. As before, the RK method was the fastest.

The error comparison can be found in Table VI. The results shown here are consistent with the results for the previous two problems. That is, the EM does not show to have significantly better accuracy than the other methods (19:31).

The explanation for this is in the differencing scheme. The partial differential equation was turned into an ODE by the use of a central difference on the spatial variables. That central difference has truncation error of $0(h)^2$. The IMSL subroutines demonstrate this fact in the accuracies being almost identical with the EM for all three problems examined.

TABLE VI

Error Comparison of Cylindrical Problem

Method	TBAR									
	1	2	3	4	5	6	7	8	9	10
Eigenvalue Method (EM)	25.16	34.38	44.04	51.5	55.0	55.53	54.07	51.37	47.96	44.21
Runge-Kutta (RK)	25.16	34.38	44.04	51.5	55.0	55.53	54.07	51.37	47.97	44.23
Gears	25.16	34.38	44.04	51.5	55.0	55.53	54.07	51.37	47.96	44.21
Extrapolation	25.16	34.38	44.04	51.5	55.0	55.53	54.07	51.37	47.97	44.22
Implicit Method (IM)	25.24	34.48	44.11	51.57	55.08	55.62	54.16	51.46	48.05	44.30
Crank-Nicolson (CN)	25.16	34.38	44.04	51.5	55.01	55.53	54.07	51.37	47.97	44.22

IV. Conclusions and Recommendations

Conclusions

The EM is not as accurate as Landry had said (19:44). The results obtained have indicated that the EM is as accurate as the CN or IM methods. This is believed to be connected to the Taylor series differencing used on the spatial variables. The results have shown that the ODE solvers used from IMSL are as accurate as the EM even though the IMSL routines use other methods with truncation errors to solve the ODE.

The EM is probably the easiest method to program. Once the [D] matrix of equation (25) is set up, it can be loaded into the program via an input file. Once the eigenvalues and eigenvectors are found, the number of time steps that is taken can be changed to whatever is needed or even made an input variable.

As far as the fastest routines, the EM is not as fast as the RK method. The RK method is the fastest but not as easy to program. The problems run here have the evaluation of the derivative being very inexpensive in computer time. This tends to make the RK method better (14:617).

Overall the MOL is the better method over CN and IM. The programs are easier to write because a canned subroutine is available to do all the work. No less accuracy was encountered when using the IMSL ODE solvers.

Recommendations

One thing that has not been done yet is to use the EM on a nonlinear

problem. An example would be to have heat loss by radiation. The example in Shih and Skladany required the eigenvalues and eigenvectors to be found each and every time step. Finding eigenvalues and eigenvectors is another area being looked into heavily (19:46). For all the problems presented here the eigenvalues and eigenvectors were found only once for all ten time steps.

Another thing would be to time the EM for long transient times (i.e., 50 or more time steps), then compare the amount of computer time needed with that of the other MOL techniques.

If the problem does not have any nonlinear terms, then the eigenvalues and eigenvectors only need to be found once. Where the MOL requires the evaluation of the derivatives more than once per time step, some interesting results could be derived from this test.

Finally, a problem more like those found in nuclear engineering, consider a cylinder with a heat source in the center surrounded by cladding. The first problem here is to find the exact solution. This is a coupled partial differential equation, which solving for the exact solution cannot be done by standard methods.

Appendix: Subroutines Used in the Study

This appendix contains the subroutines used in the IMSL routines for this study. This is provided for the reader to help them better evaluate the routines used.

Any questions concerning the IMSL routines should be directed to the company given in the bibliography.

```
      SUBROUTINE FCN1 (N,X,Y,YPRIME)
      INTEGER N
      REAL Y(N),YPRIME(N),X
      YPRIME(1) = 0.0
      YPRIME(2) = ((2*Y(1)) - (4*Y(2)) + (2*Y(3)))
      YPRIME(3) = (Y(1) + Y(2) + Y(4) + Y(8) - (4*Y(3)))
      YPRIME(4) = (Y(1) + Y(3) + Y(5) + Y(9) - (4*Y(4)))
      YPRIME(5) = (Y(1) + Y(4) + Y(6) + Y(10) - (4*Y(5)))
      YPRIME(6) = (Y(1) + Y(5) + Y(7) + Y(11) - (4*Y(6)))
      YPRIME(7) = (Y(1) + (2*Y(6)) + Y(12) - (4*Y(7)))
      YPRIME(8) = ((2*Y(3)) + (2*Y(9)) - (4*Y(8)))
      YPRIME(9) = (Y(4) + Y(8) + Y(10) + Y(13) - (4*Y(9)))
      YPRIME(10) = (Y(5) + Y(9) + Y(11) + Y(14) - (4*Y(10)))
      YPRIME(11) = (Y(6) + Y(10) + Y(12) + Y(15) -(4*Y(11)))
      YPRIME(12) = (Y(7) + (2*Y(11)) + Y(16) - (4*Y(12)))
      YPRIME(13) = ((2*Y(9)) + (2*Y(14)) - (4*Y(13)))
      YPRIME(14) = (Y(10) + Y(13) + Y(15) + Y(17) - (4*Y(14)))
      YPRIME(15) = (Y(11) + Y(14) + Y(16) + Y(18) - (4*Y(15)))
      YPRIME(16) = (Y(12) + (2*Y(15)) + Y(19) - (4*Y(16)))
      YPRIME(17) = ((2*Y(14)) + (2*Y(18)) - (4*Y(17)))
      YPRIME(18) = (Y(15) + Y(17) + Y(19) + Y(20) - (4*Y(18)))
      YPRIME(19) = (Y(16) + (2*Y(18)) + Y(21) - (4*Y(19)))
      YPRIME(20) = ((2*Y(18)) + (2*Y(21)) - (4*Y(20)))
      YPRIME(21) = Y(19) + (2*Y(20)) - (3*Y(21))
      RETURN
      END
```

```
      SUBROUTINE FCN1 (N,X,Y,YPRIME)
      INTEGER N
      REAL Y(N),YPRIME(N),X
C
       YPRIME(1) = 0.0
       YPRIME(2) = Y(3) + 2.*Y(1) + Y(12) - 4.*Y(2)
       YPRIME( 3)=Y( 4)+Y( 2)+Y(14)-4.*Y( 3)+Y(1)
       YPRIME( 4)=Y( 5)+Y( 3)+Y(15)-4.*Y( 4)+Y(1)
       YPRIME( 5)=Y( 6)+Y( 4)+Y(16)-4.*Y( 5)+Y(1)
       YPRIME( 6)=Y( 7)+Y( 5)+Y(17)-4.*Y( 6)+Y(1)
       YPRIME( 7)=Y( 8)+Y( 6)+Y(18)-4.*Y( 7)+Y(1)
       YPRIME( 8)=Y( 9)+Y( 7)+Y(19)-4.*Y( 8)+Y(1)
       YPRIME( 9)=Y(10)+Y( 8)+Y(20)-4.*Y( 9)+Y(1)
       YPRIME(10)=Y(11)+Y( 9)+Y(21)-4.*Y(10)+Y(1)
       YPRIME(11)=Y(12)+Y(10)+Y(22)-4.*Y(11)+Y(1)
       YPRIME(12)=Y(1)+Y(11)+Y(23)+Y(1)-4.*Y(12)
       YPRIME( 13)=Y( 14)+Y( 24)+Y(  2)-4.*Y( 13)+Y(1)
       YPRIME( 14)=Y( 15)+Y( 13)-4.*Y( 14)+Y( 25)+Y(  3)
       YPRIME( 15)=Y( 16)+Y( 14)-4.*Y( 15)+Y( 26)+Y(  4)
       YPRIME( 16)=Y( 17)+Y( 15)-4.*Y( 16)+Y( 27)+Y(  5)
       YPRIME( 17)=Y( 18)+Y( 16)-4.*Y( 17)+Y( 28)+Y(  6)
       YPRIME( 18)=Y( 19)+Y( 17)-4.*Y( 18)+Y( 29)+Y(  7)
       YPRIME( 19)=Y( 20)+Y( 18)-4.*Y( 19)+Y( 30)+Y(  8)
       YPRIME( 20)=Y( 21)+Y( 19)-4.*Y( 20)+Y( 31)+Y(  9)
       YPRIME( 21)=Y( 22)+Y( 20)-4.*Y( 21)+Y( 32)+Y( 10)
       YPRIME( 22)=Y( 23)+Y( 21)-4.*Y( 22)+Y( 33)+Y( 11)
       YPRIME( 23)=Y( 22)+Y( 34)+Y( 12)-4.*Y( 23)+Y(1)
       YPRIME( 24)=Y( 25)+Y( 35)+Y( 13)-4.*Y( 24)+Y(1)
       YPRIME( 25)=Y( 26)+Y( 24)-4.*Y( 25)+Y( 36)+Y( 14)
       YPRIME( 26)=Y( 27)+Y( 25)-4.*Y( 26)+Y( 37)+Y( 15)
       YPRIME( 27)=Y( 28)+Y( 26)-4.*Y( 27)+Y( 38)+Y( 16)
       YPRIME( 28)=Y( 29)+Y( 27)-4.*Y( 28)+Y( 39)+Y( 17)
       YPRIME( 29)=Y( 30)+Y( 28)-4.*Y( 29)+Y( 40)+Y( 18)
       YPRIME( 30)=Y( 31)+Y( 29)-4.*Y( 30)+Y( 41)+Y( 19)
       YPRIME( 31)=Y( 32)+Y( 30)-4.*Y( 31)+Y( 42)+Y( 20)
       YPRIME( 32)=Y( 33)+Y( 31)-4.*Y( 32)+Y( 43)+Y( 21)
       YPRIME( 33)=Y( 34)+Y( 32)-4.*Y( 33)+Y( 44)+Y( 22)
       YPRIME( 34)=Y( 33)+Y( 45)+Y( 23)-4.*Y( 34)+Y(1)
       YPRIME( 35)=Y( 36)+Y( 46)+Y( 24)-4.*Y( 35)+Y(1)
       YPRIME( 36)=Y( 37)+Y( 35)-4.*Y( 36)+Y( 47)+Y( 25)
       YPRIME( 37)=Y( 38)+Y( 36)-4.*Y( 37)+Y( 48)+Y( 26)
       YPRIME( 38)=Y( 39)+Y( 37)-4.*Y( 38)+Y( 49)+Y( 27)
       YPRIME( 39)=Y( 40)+Y( 38)-4.*Y( 39)+Y( 50)+Y( 28)
       YPRIME( 40)=Y( 41)+Y( 39)-4.*Y( 40)+Y( 51)+Y( 29)
       YPRIME( 41)=Y( 42)+Y( 40)-4.*Y( 41)+Y( 52)+Y( 30)
       YPRIME( 42)=Y( 43)+Y( 41)-4.*Y( 42)+Y( 53)+Y( 31)
       YPRIME( 43)=Y( 44)+Y( 42)-4.*Y( 43)+Y( 54)+Y( 32)
       YPRIME( 44)=Y( 45)+Y( 43)-4.*Y( 44)+Y( 55)+Y( 33)
       YPRIME( 45)=Y( 44)+Y( 56)+Y( 34)-4.*Y( 45)+Y(1)
       YPRIME( 46)=Y( 47)+Y( 37)+Y( 35)-4.*Y( 46)+Y(1)
       YPRIME( 47)=Y( 48)+Y( 46)-4.*Y( 47)+Y( 58)+Y( 36)
       YPRIME( 48)=Y( 49)+Y( 47)-4.*Y( 48)+Y( 59)+Y( 37)
       YPRIME( 49)=Y( 50)+Y( 48)-4.*Y( 49)+Y( 60)+Y( 38)
       YPRIME( 50)=Y( 51)+Y( 49)-4.*Y( 50)+Y( 61)+Y( 39)
       YPRIME( 51)=Y( 52)+Y( 50)-4.*Y( 51)+Y( 62)+Y( 40)
```

```
YPRIME( 52)=Y( 53)+Y( 51)-4.*Y( 52)+Y( 63)+Y( 41)
YPRIME( 53)=Y( 54)+Y( 52)-4.*Y( 53)+Y( 64)+Y( 42)
YPRIME( 54)=Y( 55)+Y( 53)-4.*Y( 54)+Y( 65)+Y( 43)
YPRIME( 55)=Y( 56)+Y( 54)-4.*Y( 55)+Y( 66)+Y( 44)
YPRIME( 56)=Y( 55)+Y( 67)+Y( 45)-4.*Y( 56)+Y(1)
YPRIME( 57)=Y( 58)+Y( 68)+Y( 46)-4.*Y( 57)+Y(1)
YPRIME( 58)=Y( 59)+Y( 57)-4.*Y( 58)+Y( 69)+Y( 47)
YPRIME( 59)=Y( 60)+Y( 58)-4.*Y( 59)+Y( 70)+Y( 48)
YPRIME( 60)=Y( 61)+Y( 59)-4.*Y( 60)+Y( 71)+Y( 49)
YPRIME( 61)=Y( 62)+Y( 60)-4.*Y( 61)+Y( 72)+Y( 50)
YPRIME( 62)=Y( 63)+Y( 61)-4.*Y( 62)+Y( 73)+Y( 51)
YPRIME( 63)=Y( 64)+Y( 62)-4.*Y( 63)+Y( 74)+Y( 52)
YPRIME( 64)=Y( 65)+Y( 63)-4.*Y( 64)+Y( 75)+Y( 53)
YPRIME( 65)=Y( 66)+Y( 64)-4.*Y( 65)+Y( 76)+Y( 54)
YPRIME( 66)=Y( 67)+Y( 65)-4.*Y( 66)+Y( 77)+Y( 55)
YPRIME( 67)=Y( 66)+Y( 78)+Y( 56)-4.*Y( 67)+Y(1)
YPRIME( 68)=Y( 69)+Y( 79)+Y( 57)-4.*Y( 68)+Y(1)
YPRIME( 69)=Y( 70)+Y( 68)-4.*Y( 69)+Y( 80)+Y( 58)
YPRIME( 70)=Y( 71)+Y( 69)-4.*Y( 70)+Y( 81)+Y( 59)
YPRIME( 71)=Y( 72)+Y( 70)-4.*Y( 71)+Y( 82)+Y( 60)
YPRIME( 72)=Y( 73)+Y( 71)-4.*Y( 72)+Y( 83)+Y( 61)
YPRIME( 73)=Y( 74)+Y( 72)-4.*Y( 73)+Y( 84)+Y( 62)
YPRIME( 74)=Y( 75)+Y( 73)-4.*Y( 74)+Y( 85)+Y( 63)
YPRIME( 75)=Y( 76)+Y( 74)-4.*Y( 75)+Y( 86)+Y( 64)
YPRIME( 76)=Y( 77)+Y( 75)-4.*Y( 76)+Y( 87)+Y( 65)
YPRIME( 77)=Y( 78)+Y( 76)-4.*Y( 77)+Y( 88)+Y( 66)
YPRIME( 78)=Y( 77)+Y( 89)+Y( 67)-4.*Y( 78)+Y(1)
YPRIME( 79)=Y( 80)+Y( 90)+Y( 68)-4.*Y( 79)+Y(1)
YPRIME( 80)=Y( 81)+Y( 79)-4.*Y( 80)+Y( 91)+Y( 69)
YPRIME( 81)=Y( 82)+Y( 80)-4.*Y( 81)+Y( 92)+Y( 70)
YPRIME( 82)=Y( 83)+Y( 81)-4.*Y( 82)+Y( 93)+Y( 71)
YPRIME( 83)=Y( 84)+Y( 82)-4.*Y( 83)+Y( 94)+Y( 72)
YPRIME( 84)=Y( 85)+Y( 83)-4.*Y( 84)+Y( 95)+Y( 73)
YPRIME( 85)=Y( 86)+Y( 84)-4.*Y( 85)+Y( 96)+Y( 74)
YPRIME( 86)=Y( 87)+Y( 85)-4.*Y( 86)+Y( 97)+Y( 75)
YPRIME( 87)=Y( 88)+Y( 86)-4.*Y( 87)+Y( 98)+Y( 76)
YPRIME( 88)=Y( 89)+Y( 87)-4.*Y( 88)+Y( 99)+Y( 77)
YPRIME( 89)=Y( 89)+Y(100)+Y( 78)-4.*Y( 89)+Y(1)
YPRIME( 90)=Y( 91)+Y(101)+Y( 79)-4.*Y( 90)+Y(1)
YPRIME( 91)=Y( 92)+Y( 90)-4.*Y( 91)+Y(102)+Y( 80)
YPRIME( 92)=Y( 93)+Y( 91)-4.*Y( 92)+Y(103)+Y( 81)
YPRIME( 93)=Y( 94)+Y( 92)-4.*Y( 93)+Y(104)+Y( 82)
YPRIME( 94)=Y( 95)+Y( 93)-4.*Y( 94)+Y(105)+Y( 83)
YPRIME( 95)=Y( 96)+Y( 94)-4.*Y( 95)+Y(106)+Y( 84)
YPRIME( 96)=Y( 97)+Y( 95)-4.*Y( 96)+Y(107)+Y( 85)
YPRIME( 97)=Y( 98)+Y( 96)-4.*Y( 97)+Y(108)+Y( 86)
YPRIME( 98)=Y( 99)+Y( 97)-4.*Y( 98)+Y(109)+Y( 87)
YPRIME( 99)=Y(100)+Y( 98)-4.*Y( 99)+Y(110)+Y( 88)
YPRIME(100)=Y( 99)+Y(111)+Y( 89)-4.*Y(100)+Y(1)
YPRIME(101)=Y(102)+Y(112)+Y( 90)-4.*Y(101)+Y(1)
YPRIME(102)=Y(103)+Y(101)-4.*Y(102)+Y(113)+Y( 91)
YPRIME(103)=Y(104)+Y(102)-4.*Y(103)+Y(114)+Y( 92)
YPRIME(104)=Y(105)+Y(103)-4.*Y(104)+Y(115)+Y( 93)
YPRIME(105)=Y(106)+Y(104)-4.*Y(105)+Y(116)+Y( 94)
YPRIME(106)=Y(107)+Y(105)-4.*Y(106)+Y(117)+Y( 95)
```

```
      YPRIME(107)=Y(108)+Y(106)-4.*Y(107)+Y(118)+Y( 96)
      YPRIME(108)=Y(109)+Y(107)-4.*Y(108)+Y(119)+Y( 97)
      YPRIME(109)=Y(110)+Y(108)-4.*Y(109)+Y(120)+Y( 98)
      YPRIME(110)=Y(111)+Y(109)-4.*Y(110)+Y(121)+Y( 99)
      YPRIME(111)=Y(119)+Y(122)+Y(109)-4.*Y(111)+Y(1)
      YPRIME(112)=Y(113)+Y(1)+Y(1)+Y(101)-4.*Y(112)
      YPRIME(113)=Y(114)+Y(112)-4.*Y(113)+Y(102)+Y(1)
      YPRIME(114)=Y(115)+Y(113)-4.*Y(114)+Y(103)+Y(1)
      YPRIME(115)=Y(116)+Y(114)-4.*Y(115)+Y(104)+Y(1)
      YPRIME(116)=Y(117)+Y(115)-4.*Y(116)+Y(105)+Y(1)
      YPRIME(117)=Y(118)+Y(116)-4.*Y(117)+Y(106)+Y(1)
      YPRIME(118)=Y(119)+Y(117)-4.*Y(118)+Y(107)+Y(1)
      YPRIME(119)=Y(120)+Y(118)-4.*Y(119)+Y(108)+Y(1)
      YPRIME(120)=Y(121)+Y(119)-4.*Y(120)+Y(109)+Y(1)
      YPRIME(121)=Y(122)+Y(120)-4.*Y(121)+Y(110)+Y(1)
      YPRIME(121)=Y(1)+Y(121)+Y(1)+Y(111)-4.*Y(122)
C
      RETURN
      END
```

```
      SUBROUTINE FCN1 (N,X,Y,YPRIME)
      INTEGER N
      REAL Y(N),YPRIME(N),X
C
C     YPRIME OF CENTER LINE IS ZERO
C
      C4 = 1. + 1./8.
      C4B = 1. - 1./8.
      C3 = 1. + 1./6.
      C3B = 1. - 1./6.
      C2 = 1.25
      C2B = 0.75
      C1 = 1.5
      C1B = 0.5
C
      YPRIME(1) = 0.0
C
      YPRIME(2) = C1*Y(3) - 4.*Y(2) + Y(7) + C1*Y(1)
      YPRIME(3) = C2*Y(4) + C2B*Y(2) - 4.*Y(3) + Y(8) + Y(1)
      YPRIME(4) = C3*Y(5) + C3B*Y(3) - 4.*Y(4) + Y(9) + Y(1)
      YPRIME(5) = C4*Y(6) + C4B*Y(4) - 4.*Y(5) + Y(10) + Y(1)
      YPRIME(6) = 2.*Y(5) - 4.*Y(6) + Y(11) + Y(1)
C
      YPRIME(7) = C1*Y(8) + C1B*Y(1) - 4.*Y(7) + Y(12) + Y(2)
      YPRIME(8) = C2*Y(9) + C2B*Y(7) - 4.*Y(8) + Y(13) + Y(3)
      YPRIME(9) = C3*Y(10) + C3B*Y(8) - 4.*Y(9) + Y(14) + Y(4)
      YPRIME(10) = C4*Y(11) + C4B*Y(9) - 4.*Y(10) + Y(15) + Y(5)
      YPRIME(11)= 2.*Y(10) - 4.*Y(11) + Y(16) + Y(6)
C
      YPRIME(12) = C1*Y(13) + C1B*Y(1)  - 4.*Y(12) + Y(17)+ Y(7)
      YPRIME(13) = C2*Y(14) + C2B*Y(12) - 4.*Y(13) + Y(18) + Y(8)
      YPRIME(14) = C3*Y(15) + C3B*Y(13) - 4.*Y(14) + Y(19) + Y(9)
      YPRIME(15) = C4*Y(16) + C4B*Y(14) - 4.*Y(15) + Y(20) + Y(10)
      YPRIME(16) = 2.*Y(15) - 4.*Y(16) + Y(21) + Y(11)
C
      YPRIME(17) = C1*Y(18) + C1B*Y(1)  - 4.*Y(17) + Y(22) + Y(12)
      YPRIME(18) = C2*Y(19) + C2B*Y(17) - 4.*Y(18) + Y(23) + Y(13)
      YPRIME(19) = C3*Y(20) + C3B*Y(18) - 4.*Y(19) + Y(24) + Y(14)
      YPRIME(20) = C4*Y(21) + C4B*Y(19) - 4.*Y(20) + Y(25) + Y(15)
      YPRIME(21) = 2.*Y(20) - 4.*Y(21) + Y(26) + Y(16)
C
      YPRIME(22) = C1*Y(23) + C1B*Y(1)  - 4.*Y(22) + Y(17) + Y(17)
      YPRIME(23) = C2*Y(24) + C2B*Y(22) - 4.*Y(23) + 2.*Y(18)
      YPRIME(24) = C3*Y(25) + C3B*Y(23) - 4.*Y(24) + 2.*Y(19)
      YPRIME(25) = C4*Y(26) + C4B*Y(24) - 4.*Y(25) + 2.*Y(20)
      YPRIME(26) = 2.*Y(25) - 4.*Y(26) + 2.*Y(21)
C
      RETURN
      END
```

Bibliography

1. Brauer, Fred and John A. Nohel. Ordinary Differential Equations: A First Course. New York: W.A. Benjamin, Inc., 1967.

2. Builrsch R. and J. Stoen. "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods." Numerische Mathematik, Band 8 Heft 1, 1-13, (1966).

3. Canale, R.P. and S.C. Chapra. Numerical Methods for Engineers with Personnel Computer Applications. New York: McGraw-Hill 509-533, (1985).

4. Carslaw, H.S. and J.C. Jaeger. Conduction of Heat in Solids. London: Oxford University Press, 1959.

5. Churchill, Ruel V. Fourier Series and Boundary Value Problems. New York: McGraw-Hill Book Company, 1963.

6. Clark, M. and K.F. Hansen. Numerical Methods of Reactor Analysis. New York: Academic Press, 1964.

7. Crank, J. and P. Nicolson. "A Practical Method for Numerical Evaluations of Solutions of Partial Differential Equations of the Heat Conduction Type," *Proceedings of the Cambridge Philosophical Society*, 43: 50-67 (1947).

8. Duderstadt, James J. and Louis J. Hamilton. Nuclear Reactor Analysis. New York: John Wiley and Sons, Inc., 1976.

9. Fried, Isaac. Numerical Solution of Differential Equations. New York: Academic Press, Inc., 1979.

10. Gear, C. William. Numerical Initial Value Problems in Ordinary Differential Equations. Englewood Cliffs NJ: Prentice-Hall, Inc., 1971.

11. Henrici, Peter. Discrete Variable Methods in Ordinary Differential Equations. New York: John Wiley and Sons, Inc., 1962.

12. Hicks, J.S. and J. Wei. "Numerical Solution of Parabolic Partial Differential Equations With Two-Point Boundry Conditions by Use of the Method of Lines," Journal of the Association for Computing Machinery, 14 (3): 549-562 (1967).

13. Hopkins, T.R. and R. Wait, "A Comparison of Galerkin, Collocation, and the Method of Lines for Partial Differential Equations," International Journal for Numerical Methods in Engineering, 12; 1081-1107 (1978).

14. Hull, T.E. and others. "Comparing Numerical Methods for Ordinary Differential Equations," SIAM Journal Numerical Analysis, 9 (4): 603-637 (1972).

15. IMSL LIB-0009. IMSL Library, Reference Manual (Ninth Edition), Computer Library for FORTRAN subroutines. Houston, IMSL, Inc. 1982.

16. Jennings, Walter. First Course in Numerical Methods. Toronto: MacMillan Company, 1964.

17. Ketter, Robert L. and Sherwood P. Prawel, Jr. Modern Methods of Engineering Computation. New York: McGraw-Hill Book Company, 1969.

18. Kreyszig, E. Advanced Engineering Mathmatics (Fifth Edition). New York: John Wiley and Sons, Inc., 1983.

19. Landry, Capt D.W. Evaluation of the Eigenvalue Method in the Solution of Transient Heat Conduction Problems. MS thesis. School of Engineering. Air Force Institute of Technology (AU). Wright-Patterson AFB OH, January, 1985.

20. Leighton, Walter. Ordinary Differential Equations (Third Edition). Belmont CA: Wadsworth Publishing Company, Inc., 1970.

21. Luikov, A.V. Analytical Heat Diffusion Theory. New York: Academic Press, Inc., 1968.

22. McCracken, Daniel D. and William S. Dorn. Numerical Methods and Fortran Programming with Applications in Engineering and Science. New York: John Wiley and Sons, Inc., 1964.

23. McGilton, H. and R. Morgan. Introducing the Unix System. New York: McGraw-Hill Book Co., 1983.

24. Scarborough, J. B. Numerical Mathmatical Analysis. New York: John Hopkins Press, 1962.

25. Shih, T.M. and J.T. Skladany. "An Eigenvalue Method for Solving Transient Heat Conduction Problems." Numerical Heat Transfer, 6: 409-422 (1983).

26. Szidarovszky, Ferenc and Sidney Yakowitz. Principles and Procedures of Numerical Analysis. New York: Plenum Press, 1978.

27. Towler, B.F. and R.Y.K. Yang. "On Comparing the Accuracy of Some Finite-Difference Methods for Parabolic Partial Differential Equations," International Journal for Numerical Methods in Engineering, 14: 1021-1035 (1979).

28. Zafarullah, A. "Application of the Method of Lines to Parabolic Partial Differential Equations With Error Estimates," Journal of the Association for Computing Machinery, 17 (2): 294-302 (1970).

Vita

Second Lieutenant David B. Gee was born 29 July 1959 in Sedalia, Missouri. He graduated from Lake Oswego High School in Lake Oswego, Oregon in June 1977. He spent three years in the U.S. Navy and was nuclear plant mechanical operator onboard the U.S.S. Enterprise. He received an honorable discharge to enter the Air Force ROTC program in October 1980. He graduated from Oregon State University receiving a Bachelor of Science degree in Nuclear Engineering in June 1984. He was called to active duty in August 1984 to continue his education at the School of Engineering, Air Force Institute of Technology.

Permanent address: P.O. Box 111

Cornell, WI 54732

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GNE/ENP/86M-6	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENP	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification): see box 19	
12. PERSONAL AUTHOR(S): David B. Gee B.S. 2Lt., USAF	
13a. TYPE OF REPORT: MS Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14. DATE OF REPORT (Yr., Mo., Day): 1986 January; 15. PAGE COUNT: 49	
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Numerical Analysis, Numerical Methods, Method of Lines, Eigenvalue Method, Heat Transfer
12	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A Comparison Study of the Eigenvalue Method for the Solution of the Transient Heat Conduction Equation

Thesis Chairman: Dr. Bernard Kaplan

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 9 May 86
Dean for Research and Professional Development
Air Force Institute of Technology (AFC)
Wright-Patterson AFB OH 45433

Next Page.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. Bernard Kaplan	22b. TELEPHONE NUMBER (Include Area Code): 513-255-2012/4877; 22c. OFFICE SYMBOL: AFIT/ENP

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

This is a comparison study of the abilities of the eigenvalue method as a numerical method in solving the transient heat conduction equation. The eigenvalue method was compared to five other numerical methods; Runge-Kutta, Gears, extrapolation, fully implicit, and Crank-Nicolson. These methods were used to solve three physical problems. The first is a two dimensional slab which takes advantage of the symmetry of the problem. The second is a the same slab problem without taking advantage of the symmetry. And the third is a cylindrical problem taking full advantage of symmetry.

The scope of the study is to see which methods take less computer time while maintaining sufficient accuracy. The time it takes the computer to totally execute the program was used as the time comparison basis. The accuracy is a comparison of the exact solution to the numerical solution. A root mean square average of all the grid points per time step is used.

The results of the study were surprising. The accuracy of the eigenvalue method is not any better than that of the Crank-Nicolson method. The computer times show that the eigenvalue is not the fastest for short transient times. A long transient problem with nonlinear terms was not used in this study.

END

1-87

DTIC