To the Graduate Council:

I am submitting herewith a thesis written by John Paul Travis entitled "A Numerical Study of Diagonally Implicit Runge Kutta Methods For a Parabolic Partial Differential Equation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Lawrence A. Bales

Lawrence A. Bales, Major Professor

We have read this thesis
and recommend its acceptance:

Steven M. Serbin

Ohannes Karakashian

Accepted for the Council:

C W Minkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature John Travis

Date August 7, 1925

A NUMERICAL STUDY OF DIAGONALLY IMPLICIT RUNGE KUTTA METHODS FOR A PARABOLIC PARTIAL DIFFERENTIAL EQUATION

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

John Paul Travis

August 1985

ACKNOWLEDGEMENTS

The author is very thankful to his major professor, Dr. Larry Bales, for his helpfulness and support as well as for his friendship.

Further, the author is grateful to Drs. O. Karakashian, L. Bales, and S. Serbin for the giving of their time and patience for the completion of this project. Also, the author is especially grateful to Dr. S. Serbin who has been helpful in many ways over the past three years.

Finally, the author wishes to thank his parents, Dr. and Mrs. James L. Travis, for their support and encouragement throughout the years even when it wasn't appreciated.

ABSTRACT

In this thesis, we describe numerical experiments designed to test the performance of diagonally implicit Runge-Kutta methods for approximately solving a parabolic partial differential equation.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this paper, we describe the results of numerical experiments designed to test the performance of a Galerkin-Runge-Kutta method in solving a parabolic partial differential equation. In particular, we concern ourselves with diagonally implicit Runge-Kutta (DIRK) methods [1] and investigate the computational rates of convergence of these methods.

For certain partial differential equations in which the forcing term $f(x,t)$ is zero and the coefficients are independent of t, theoretical rates of convergence are known [2]. Crouzeix in [3] has considered the case in which the coefficients depend on t and the case of the non-zero forcing term. These theoretical results will be compared later to our computational results. Then, we will consider several problems for which these rates are not known. The goal of this study is to test these DIRK methods for several problems and generate a sufficient amount of computational data upon which one may obtain a reasonable prediction for the theoretical rate of convergence of the approximation in general problems.

CHAPTER 2

THE RUNGE-KUTTA METHODS AND THE GALERKIN APPROXIMATION

In this section, we describe the diagonally implicit Runge-Kutta (DIRK) methods and the finite element formulation of the problem. First, to motivate these methods, consider the differential equation $y'(t)=f(t,y(t))$. Integrating both sides of this equation from $t=t_n$ to $t=t_n+\tau k$ gives us the equivalent form

$$(2.1)\qquad y(t_n+\tau k) = y(t_n) + \int_{t_n}^{t_n+\tau k} f(t,y(t))dt,$$

using the Fundamental Theorem of Calculus to evaluate the left hand side. With the transformation $s=(t-t_n)/k$ we can rewrite this as

$$y(t_n+\tau k) = y(t_n) + k\int_0^{\tau} f(t_n+sk,y(t_n+sk))ds.$$

In particular, with $\tau=1$, since $t_{n+1}=t_n+k$, we have

$$y(t_{n+1}) = y(t_n) + k\int_0^1 f(t_n+sk,y(t_n+sk)ds\ .$$

Now, by evaluating the above integral using a q-stage numerical quadrature formula leads us to define the approximation y_{n+1} to $y(t_{n+1})$ by

$$y_{n+1} = y_n + k\sum_{j=1}^{q} b_j f(t_{n,j},y_{n,j})\ ,$$

where $t_{n,j}=t_n+k\tau_j$ and $y_{n,j}$ is an approximation to $y(t_{n,j})$. Similarly, using (2.1) and evaluating the integral with a q-stage numerical quadrature formula for each $\tau=\tau_j$ leads us to generate the stages $\{y_{n,i}\}$ from the relation

$$y_{n,i} = y_n + k\sum_{j=1}^{q} a_{ij}f(t_{n,j},y_{n,j}).$$

We see that the stages $\{y_{n,i}\}$ and thus y_{n+1} are obtained from a coupled set of q nonlinear equations, and that the entire procedure for one step is completely characterized by the tableau

$$(2.2)\qquad \begin{array}{ccc|c} a_{11} & \cdots & a_{1q} & \tau_1 \\ \vdots & & \vdots & \vdots \\ a_{q1} & \cdots & a_{qq} & \tau_q \\ \hline b_1 & \cdots & b_q & \end{array} \;=\; \begin{array}{c|c} A & T \\ \hline b & \end{array}$$

where the integer $q\geq 1$ is the number of intermediate stages.

<u>DEFINITION 2.3.</u> We say that the method is diagonally implicit if A is triangular and all of the diagonal elements are equal. In general, the method is implicit if no permutation of rows can reduce A to triangular form with zero diagonal elements.

<u>NOTE 2.4.</u> We will henceforth assume that A in (2.2) is lower triangular with all diagonal elements equal. This assumption, as we will later see, gives us considerable computational advantage as well as insuring numerical stability of the resulting methods.

So, consider the test problem:

$$
(2.5)\quad \begin{cases}
\frac{\partial}{\partial t}u(x,t) = -L(x,t)u(x,t) + f(x,t), \; a < x < b, \; t_0 < t \le t^*, \\
L(x,t)u(x,t) = -\frac{\partial}{\partial x}(a(x,t)\frac{\partial}{\partial x}u(x,t)) + b(x,t)u(x,t) \\
u(x,0) = v(x) \qquad , \; a \le x \le b \\
u(a,t) = u(b,t) = 0, \quad 0 \le t \le t^* \quad ,
\end{cases}
$$

where a,b,and t^* are real numbers, $v(x)$ is a given function on $[a,b]$, and $a(x,t)>0$, $b(x,t)\ge 0$ are continuously differentiable functions defined on $[a,b]\times[0,t^*]$.

Let M be a positive integer and let $h=\frac{(b-a)}{M+1}$ be the space step size. Further, define $x_i=a+ih$, $i=0,\ldots,M+1$. At each time step, we solve a boundary value problem in the space variable x. Indeed, if S_h is the subspace of cubic splines which vanish at a and b, the semidiscrete approximation to $u(x,t)$ is defined by

$$
(2.6)\qquad u_h(t) = \sum_{i=0}^{M+1} \alpha_i(t)\phi_i(x) \quad ,
$$

where M is the number of interior mesh nodes and $\{\phi_i\}_{i=0}^{M+1}$ is a basis for C^2 cubics. The semidiscrete approximation satisfies the equation

$$
(2.7)\qquad G\alpha'(t) = -S(t)\alpha(t)+F(t) \quad ,
$$

where G is the "Gram Matrix," $G_{ij}=\int_a^b \phi_i(x)\phi_j(x)dx$, S the "Stiffness Matrix,"

$$S_{ij} = \int_a^b a(x,t)\phi_i'(x)\phi_j'(x)dx + \int_a^b b(x,t)\phi_i(x)\phi_j(x)dx$$

and F is the L_2 projection of f(t) onto S_h, $(F(t))_i = \int_a^b f(x,t)\phi_i(x)dx$.

Note in the above that time is still a continuous variable while the spacial variable has been discretized. Hence, we get the initial-value problem in R^{M+2}:

$$(2.8)\quad \begin{cases} G\alpha'(t) = -S(t)\alpha(t)+F(t) \\ \alpha_j(t_0) = \int_a^b v(x)\phi_j(x)dx \ , \ j=0,\dots,M+1 \ , \end{cases}$$

or on S_h

$$(2.9)\quad \begin{cases} (u_h(t))_t = -L_h(t)u_h(t) + Pf(t) \ . \\ u_h(0) = Pv \quad , \text{ the } L_2 \text{ projection of } v(x) \text{ onto the subspace } S_h. \end{cases}$$

Now, toward solving this initial value problem in time, let N be a positive integer and let $k=\frac{(t^*-t_0)}{N}$ be the time step size, with $t^n=t_0+nk$, n=0,...,N. Define

$$(2.10)\quad \begin{cases} U_h^n = \sum_{j=0}^{M+1} \alpha_j^n \phi_j(x) \ , \\ U_h^{n,i} = \sum_{j=0}^{M+1} \alpha_j^{n,i} \phi_j(x) \ , \\ U_h^{n,i} = U_h^n + k \sum_{j=1}^{q} a_{ij}(-L_h(t^{n,j})U^{n,j}+Pf(t^{n,j})), \quad i=1,\dots,q \ . \end{cases}$$

Using our basis functions and above, we get the block system

$$\left[\begin{pmatrix} G & & 0 \\ & \ddots & \\ 0 & & G \end{pmatrix} + k\begin{pmatrix} a_{11}S(t^{n,1}) & \cdots & a_{1q}S(t^{n,q}) \\ \vdots & & \vdots \\ a_{q1}S(t^{n,1}) & \cdots & a_{qq}S(t^{n,q}) \end{pmatrix}\right]\begin{bmatrix} \alpha^{n,1} \\ \vdots \\ \alpha^{n,q} \end{bmatrix} = \begin{bmatrix} G\alpha^{n} \\ \vdots \\ G\alpha^{n} \end{bmatrix} + \begin{bmatrix} B_1 \\ \vdots \\ B_q \end{bmatrix} \tag{2.11}$$

where $B_i = k\sum_{j=1}^{q} a_{ij}F(t^{n,j})$.

So, given $U_h^n \approx u(t^n)$, we obtain the fully discrete approximation $U_h^{n+1} \approx u(t^{n+1})$ in two steps. First, compute the q intermediate stages $\{U_h^{n,i}\}_{i=1}^{q}$ where $t^{n,i} = t^n + k\tau_i$, $i=1,\dots,q$, using (2.10). Note, since our methods are diagonally implicit, we may easily solve for $U^{n,i}$ by forward solving the block system above. For $i=1,\dots,q$, this involves only factoring matrices of the form $G + ka_{ii}S(t^{n,i})$. Finally, we combine these to get U_h^{n+1} :

$$U_h^{n+1} = U_h^n + k \sum_{i=1}^{q} b_i(-L_h(t^{n,i})U^{n,i} + Pf(t^{n,i})). \tag{2.12}$$

CHAPTER 3

RESULTS OF NUMERICAL EXPERIMENTS

The linear system (2.11) was solved using a preconditioned conjugate gradient algorithm with a convergence tolerence of 1E-20. The preconditioner used is equal to the actual left-hand side determined at time zero and is not changed during each run. This seems to work very well if the left-hand side changes little with respect to time. Indeed, if the coefficients a(x,t) and b(x,t) are independent of time, then the preconditioner equals the actual left-hand side for all time. The factorization of this preconditioner was calculated once using the IMSL routine LEQ1PB on the IBM 3081 CPU at the University of Tennessee in Knoxville. Throughout the program, computations were performed using double precision arithmetic.

Finally, for each run corresponding to an increasing number N and M of nodes, two error approximations were calculated via the formulas:

$$(3.1)\qquad \varepsilon_2 = \left\{ \int_a^b (U^N - u(x,t^*))^2 dx \right\}^{\frac{1}{2}}$$
$$= \left\{ \int_a^b u(x,t^*)^2 dx - 2\alpha^t P u + \alpha^t G \alpha \right\}^{\frac{1}{2}} ,$$

with the integral computed using a composite 5-point Gauss quadrature rule, and

$$(3.2)\qquad \varepsilon_r = \left\{ \sum_{i=1}^{M} (U_i^N - u(x_i,t^*))^2 / \sum_{i=1}^{M} u(x_i,t^*)^2 \right\}^{\frac{1}{2}} .$$

Comparing ε^1 at $N=N_1$ to ε^2 at $N=N_2$ using the formula

$$(3.3) \qquad p \approx \frac{\ln(\varepsilon^1/\varepsilon^2)}{\ln(N_2/N_1)}$$

gives the rate of convergence p. Rates of convergence using both (3.1) and (3.2) are listed in the results.

In the Appendix, a table exhibits the parameters of the DIRKs used. At the end of this section, several tables are exhibited showing the values for a(x,t), b(x,t), and u(x,t) that were chosen along with the resulting f(x,t) and v(x). Computational results are given with the two rates of convergence shown for N and M ranging by tens from 10 to 100 and with q = 1, 2, 3. The last two columns in each table give the computed rates of convergence via (3.3) which as N (and M) become large, should be approaching some limiting value.

SUMMARY OF RESULTS.

Problem 1. (Table 1). Here, the coefficients a(x,t) and b(x,t) are constant and the forcing term is zero. In [2], optimal rates of convergence of q+1 have been shown for these q×q DIRK methods. The results generated agree with this fact.

Problem 2. (Table 2). This problem differs from problem 1 in that the forcing term is now non-zero but is a function of x alone. Again, it seems that optimal rates of convergence are obtained. Note, due to the size of the L_2 error for the 3×3 DIRK, roundoff in the square root function gives meaningless results for N and M large.

TABLE 1

FORCING TERM ZERO

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.1402D-02	0.2587D-01	0.0	0.0
0.1571	0.1571	0.3500D-03	0.6462D-02	0.2002D+01	0.2001D+01
0.1047	0.1047	0.1555D-03	0.2871D-02	0.2001D+01	0.2000D+01
0.0785	0.0785	0.8747D-04	0.1615D-02	0.2000D+01	0.2000D+01
0.0628	0.0628	0.5598D-04	0.1034D-02	0.2000D+01	0.2000D+01
0.0524	0.0524	0.3888D-04	0.7178D-03	0.2000D+01	0.2000D+01
0.0449	0.0449	0.2856D-04	0.5273D-03	0.2000D+01	0.2000D+01
0.0393	0.0393	0.2187D-04	0.4037D-03	0.2000D+01	0.2000D+01
0.0349	0.0349	0.1728D-04	0.3190D-03	0.2000D+01	0.2000D+01
0.0314	0.0314	0.1399D-04	0.2584D-03	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.3142	0.3142	0.3541D-03	0.6524D-02	0.0	0.0
0.1571	0.1571	0.5059D-04	0.9332D-03	0.2807D+01	0.2805D+01
0.1047	0.1047	0.1575D-04	0.2906D-03	0.2878D+01	0.2877D+01
0.0785	0.0785	0.6818D-05	0.1258D-03	0.2910D+01	0.2910D+01
0.0628	0.0628	0.3546D-05	0.6546D-04	0.2929D+01	0.2929D+01
0.0524	0.0524	0.2074D-05	0.3829D-04	0.2941D+01	0.2941D+01
0.0449	0.0449	0.1316D-05	0.2430D-04	0.2950D+01	0.2950D+01
0.0393	0.0393	0.8871D-06	0.1638D-04	0.2956D+01	0.2956D+01
0.0349	0.0349	0.6259D-06	0.1155D-04	0.2961D+01	0.2961D+01
0.0314	0.0314	0.4579D-06	0.8454D-05	0.2965D+01	0.2965D+01
			3X3 DIRK		
0.3142	0.3142	0.1330D-03	0.2442D-02	0.0	0.0
0.1571	0.1571	0.1153D-04	0.2120D-03	0.3528D+01	0.3526D+01
0.1047	0.1047	0.2573D-05	0.4734D-04	0.3699D+01	0.3698D+01
0.0785	0.0785	0.8678D-06	0.1597D-04	0.3778D+01	0.3777D+01
0.0628	0.0628	0.3697D-06	0.6804D-05	0.3824D+01	0.3823D+01
0.0524	0.0524	0.1830D-06	0.3370D-05	0.3855D+01	0.3854D+01
0.0449	0.0449	0.1007D-06	0.1854D-05	0.3880D+01	0.3875D+01
0.0393	0.0393	0.5974D-07	0.1103D-05	0.3906D+01	0.3891D+01
0.0349	0.0349	0.3760D-07	0.6965D-06	0.3931D+01	0.3903D+01
0.0314	0.0314	0.2453D-07	0.4612D-06	0.4052D+01	0.3913D+01

Problem 1: $a(x,t)=1$ $b(x,t)=0$ $v(x)=\sin(x)$

$u(x,t)=e^{-t}\sin(x)$ $f(x,t)=0$

$B= 3.14159$ $T^{*}= 3.14159$

TABLE 2

FORCING TERM IS A FUNCTION OF X

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.1402D-02	0.1059D-02	0.0	0.0
0.1571	0.1571	0.3500D-03	0.2669D-03	0.2002D+01	0.1988D+01
0.1047	0.1047	0.1555D-03	0.1188D-03	0.2001D+01	0.1996D+01
0.0785	0.0785	0.8747D-04	0.6685D-04	0.2000D+01	0.1998D+01
0.0628	0.0628	0.5598D-04	0.4279D-04	0.2000D+01	0.1999D+01
0.0524	0.0524	0.3888D-04	0.2972D-04	0.2000D+01	0.1999D+01
0.0449	0.0449	0.2856D-04	0.2184D-04	0.2000D+01	0.1999D+01
0.0393	0.0393	0.2187D-04	0.1672D-04	0.2000D+01	0.2000D+01
0.0349	0.0349	0.1728D-04	0.1321D-04	0.2000D+01	0.2000D+01
0.0314	0.0314	0.1399D-04	0.1070D-04	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.3142	0.3142	0.3543D-03	0.2570D-03	0.0	0.0
0.1571	0.1571	0.5059D-04	0.3784D-04	0.2808D+01	0.2764D+01
0.1047	0.1047	0.1575D-04	0.1188D-04	0.2878D+01	0.2858D+01
0.0785	0.0785	0.6818D-05	0.5162D-05	0.2910D+01	0.2897D+01
0.0628	0.0628	0.3546D-05	0.2691D-05	0.2930D+01	0.2919D+01
0.0524	0.0524	0.2073D-05	0.1576D-05	0.2943D+01	0.2934D+01
0.0449	0.0449	0.1316D-05	0.1001D-05	0.2951D+01	0.2943D+01
0.0393	0.0393	0.8842D-06	0.6752D-06	0.2976D+01	0.2951D+01
0.0349	0.0349	0.6237D-06	0.4766D-06	0.2963D+01	0.2956D+01
0.0314	0.0314	0.4544D-06	0.3489D-06	0.3006D+01	0.2961D+01
			3X3 DIRK		
0.3142	0.3142	0.1336D-03	0.8795D-04	0.0	0.0
0.1571	0.1571	0.1155D-04	0.7968D-05	0.3532D+01	0.3464D+01
0.1047	0.1047	0.2577D-05	0.1801D-05	0.3700D+01	0.3668D+01
0.0785	0.0785	0.8675D-06	0.6108D-06	0.3785D+01	0.3758D+01
0.0628	0.0628	0.3656D-06	0.2611D-06	0.3872D+01	0.3809D+01
0.0524	0.0524	0.1731D-06	0.1296D-06	0.4100D+01	0.3842D+01
0.0449	0.0449	0.8297D-07	0.7141D-07	0.4772D+01	0.3865D+01
0.0393	0.0393	0.4470D-07	0.4252D-07	0.4631D+01	0.3883D+01
0.0349	0.0349	0.4470D-07	0.2687D-07	0.0	0.3896D+01
0.0314	0.0314	0.5771D-07	0.1781D-07	-0.2424D+01	0.3907D+01

Problem 2: $a(x,t)=1$ $b(x,t)=0$ $v(x)=2\sin(x)$

$u(x,t)=(e^{-t}+1)\sin(x)$ $f(x,t)=\sin(x)$

$B= 3.14159$ $T^{*}= 3.14159$

Problem 3. (Table 3). Next, we again choose constant coefficients but let the forcing term be a non-zero function of both x and t. Further, the exact solution here is just a polynomial in x and t. Note that the 3×3 DIRK now deviates from the results of Problems 1 and 2 in that 4th order convergence is not indicated.

Problem 4. (Table 4). Here, b(x,t) is non-zero but still constant, the forcing term is non-zero in both x and t, the initial value v(x) is zero and the exact solution is again a polynomial in x and t. The rates of convergence have greatly diminished especially in the 3×3 case.

Problem 5. (Table 5). Now, the exact solution as well as the forcing term are both rather complicated but the coefficients are still constant. Definitely, both the 2×2 and the 3×3 DIRKs indicate 3rd order.

Problem 6. (Table 6). This problem is similar to 5 except b(x,t)=x. Notice that similar results hold as in Problem 5.

Problem 7. (Table 7). The forcing term is again homogeneous but contrary to the given condition that b(x,t) be non-negative, the coefficient b(x,t) is negative for all t > 0. Also, a(x,t) here is a function of time. Regardless, optimal convergence is achieved as in Problem 1.

Problem 8. (Table 8). Now, b(x,t)=0, a(x,t) is a function of time only, and the forcing term is a function of both t and x. In this case, it is not clear what the rate of convergence should be.

Problem 9. (Table 9). Finally, we have a case where a(x,t) is both a function of t and x. In this case, the elliptic operator L(x,t) defined in (2.4) does not commute with itself at different time levels. Also, the forcing term is non-zero. In this case, the 3×3 DIRK is again only 3rd order.

TABLE 3

FORCING TERM IS A FUNCTION OF X AND T

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.1000	0.3142	0.4503D-02	0.2632D-02	0.0	0.0
0.0500	0.1571	0.1126D-02	0.6582D-03	0.2000D+01	0.2000D+01
0.0333	0.1047	0.5005D-03	0.2925D-03	0.2000D+01	0.2000D+01
0.0250	0.0785	0.2816D-03	0.1646D-03	0.2000D+01	0.2000D+01
0.0200	0.0628	0.1802D-03	0.1053D-03	0.2000D+01	0.2000D+01
0.0167	0.0524	0.1251D-03	0.7314D-04	0.2000D+01	0.2000D+01
0.0143	0.0449	0.9194D-04	0.5373D-04	0.2000D+01	0.2000D+01
0.0125	0.0393	0.7039D-04	0.4114D-04	0.2000D+01	0.2000D+01
0.0111	0.0349	0.5562D-04	0.3250D-04	0.2000D+01	0.2000D+01
0.0100	0.0314	0.4505D-04	0.2633D-04	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.1000	0.3142	0.2312D-02	0.1351D-02	0.0	0.0
0.0500	0.1571	0.4701D-03	0.2747D-03	0.2298D+01	0.2298D+01
0.0333	0.1047	0.1761D-03	0.1029D-03	0.2422D+01	0.2422D+01
0.0250	0.0785	0.8559D-04	0.5002D-04	0.2507D+01	0.2507D+01
0.0200	0.0628	0.4824D-04	0.2819D-04	0.2570D+01	0.2570D+01
0.0167	0.0524	0.2993D-04	0.1749D-04	0.2618D+01	0.2618D+01
0.0143	0.0449	0.1988D-04	0.1162D-04	0.2656D+01	0.2656D+01
0.0125	0.0393	0.1389D-04	0.8115D-05	0.2686D+01	0.2686D+01
0.0111	0.0349	0.1009D-04	0.5896D-05	0.2712D+01	0.2712D+01
0.0100	0.0314	0.7564D-05	0.4421D-05	0.2734D+01	0.2733D+01
			3X3 DIRK		
0.1000	0.3142	0.1666D-02	0.9735D-03	0.0	0.0
0.0500	0.1571	0.2979D-03	0.1741D-03	0.2483D+01	0.2483D+01
0.0333	0.1047	0.9931D-04	0.5804D-04	0.2710D+01	0.2710D+01
0.0250	0.0785	0.4344D-04	0.2539D-04	0.2874D+01	0.2874D+01
0.0200	0.0628	0.2225D-04	0.1300D-04	0.2999D+01	0.2999D+01
0.0167	0.0524	0.1265D-04	0.7396D-05	0.3095D+01	0.3095D+01
0.0143	0.0449	0.7762D-05	0.4536D-05	0.3171D+01	0.3171D+01
0.0125	0.0393	0.5042D-05	0.2947D-05	0.3231D+01	0.3230D+01
0.0111	0.0349	0.3427D-05	0.2003D-05	0.3278D+01	0.3278D+01
0.0100	0.0314	0.2416D-05	0.1413D-05	0.3318D+01	0.3315D+01

Problem 3: $a(x,t)=1$ $b(x,t)=0$ $v(x)=x^2(x-1)$

$u(x,t)=x(1-x)(t^2-x)$ $f(x,t)=2(t(x-x^2)+t^2+1-3x)$

$B= 1.00000$ $T^*= 3.14159$

TABLE 4

INITIAL VALUE ZERO

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.1000	0.3142	0.2393D-02	0.2485D-02	0.0	0.0
0.0500	0.1571	0.6019D-03	0.6249D-03	0.1991D+01	0.1992D+01
0.0333	0.1047	0.2675D-03	0.2778D-03	0.2000D+01	0.2000D+01
0.0250	0.0785	0.1505D-03	0.1562D-03	0.2000D+01	0.2000D+01
0.0200	0.0628	0.9632D-04	0.1000D-03	0.2000D+01	0.2000D+01
0.0167	0.0524	0.6689D-04	0.6944D-04	0.2000D+01	0.2000D+01
0.0143	0.0449	0.4914D-04	0.5102D-04	0.2000D+01	0.2000D+01
0.0125	0.0393	0.3762D-04	0.3906D-04	0.2000D+01	0.2000D+01
0.0111	0.0349	0.2973D-04	0.3086D-04	0.2000D+01	0.2000D+01
0.0100	0.0314	0.2408D-04	0.2500D-04	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.1000	0.3142	0.1294D-02	0.1343D-02	0.0	0.0
0.0500	0.1571	0.2732D-03	0.2836D-03	0.2244D+01	0.2244D+01
0.0333	0.1047	0.1056D-03	0.1097D-03	0.2343D+01	0.2343D+01
0.0250	0.0785	0.5281D-04	0.5483D-04	0.2410D+01	0.2410D+01
0.0200	0.0628	0.3050D-04	0.3167D-04	0.2459D+01	0.2459D+01
0.0167	0.0524	0.1935D-04	0.2009D-04	0.2497D+01	0.2497D+01
0.0143	0.0449	0.1311D-04	0.1361D-04	0.2527D+01	0.2527D+01
0.0125	0.0393	0.9322D-05	0.9678D-05	0.2552D+01	0.2552D+01
0.0111	0.0349	0.6884D-05	0.7148D-05	0.2573D+01	0.2573D+01
0.0100	0.0314	0.5239D-05	0.5440D-05	0.2592D+01	0.2591D+01
			3X3 DIRK		
0.1000	0.3142	0.9596D-03	0.9962D-03	0.0	0.0
0.0500	0.1571	0.1847D-03	0.1917D-03	0.2378D+01	0.2377D+01
0.0333	0.1047	0.6613D-04	0.6866D-04	0.2533D+01	0.2533D+01
0.0250	0.0785	0.3102D-04	0.3221D-04	0.2631D+01	0.2631D+01
0.0200	0.0628	0.1699D-04	0.1764D-04	0.2698D+01	0.2698D+01
0.0167	0.0524	0.1030D-04	0.1069D-04	0.2747D+01	0.2747D+01
0.0143	0.0449	0.6703D-05	0.6959D-05	0.2784D+01	0.2784D+01
0.0125	0.0393	0.4602D-05	0.4778D-05	0.2816D+01	0.2815D+01
0.0111	0.0349	0.3292D-05	0.3419D-05	0.2844D+01	0.2843D+01
0.0100	0.0314	0.2433D-05	0.2527D-05	0.2871D+01	0.2868D+01

Problem 4: $a(x,t)=1$ $b(x,t)=1$ $v(x)=0$

$u(x,t)=x^2(1-x)t^2$ $f(x,t)=2t(x^2-x^3)+t^2(x^2-x^3+6x-2)$

$B= 1.00000$ $T^*= 3.14159$

TABLE 5

FORCING TERM COMPLICATED

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.1119D-02	0.1396D-01	0.0	0.0
0.1571	0.1571	0.2716D-03	0.3238D-02	0.2043D+01	0.2108D+01
0.1047	0.1047	0.1204D-03	0.1420D-02	0.2006D+01	0.2033D+01
0.0785	0.0785	0.6771D-04	0.7952D-03	0.2002D+01	0.2015D+01
0.0628	0.0628	0.4332D-04	0.5079D-03	0.2001D+01	0.2009D+01
0.0524	0.0524	0.3008D-04	0.3523D-03	0.2000D+01	0.2006D+01
0.0449	0.0449	0.2210D-04	0.2587D-03	0.2000D+01	0.2004D+01
0.0393	0.0393	0.1692D-04	0.1980D-03	0.2000D+01	0.2003D+01
0.0349	0.0349	0.1337D-04	0.1564D-03	0.2000D+01	0.2002D+01
0.0314	0.0314	0.1083D-04	0.1267D-03	0.2000D+01	0.2002D+01
			2X2 DIRK		
0.3142	0.3142	0.4807D-03	0.6121D-02	0.0	0.0
0.1571	0.1571	0.6261D-04	0.7915D-03	0.2941D+01	0.2951D+01
0.1047	0.1047	0.1941D-04	0.2377D-03	0.2888D+01	0.2967D+01
0.0785	0.0785	0.8455D-05	0.1017D-03	0.2889D+01	0.2952D+01
0.0628	0.0628	0.4435D-05	0.5277D-04	0.2892D+01	0.2939D+01
0.0524	0.0524	0.2616D-05	0.3093D-04	0.2894D+01	0.2930D+01
0.0449	0.0449	0.1674D-05	0.1971D-04	0.2896D+01	0.2924D+01
0.0393	0.0393	0.1137D-05	0.1334D-04	0.2898D+01	0.2920D+01
0.0349	0.0349	0.8081D-06	0.9465D-05	0.2899D+01	0.2916D+01
0.0314	0.0314	0.5954D-06	0.6963D-05	0.2900D+01	0.2914D+01
			3X3 DIRK		
0.3142	0.3142	0.3073D-03	0.3780D-02	0.0	0.0
0.1571	0.1571	0.2533D-04	0.3491D-03	0.3601D+01	0.3437D+01
0.1047	0.1047	0.6121D-05	0.7852D-04	0.3502D+01	0.3680D+01
0.0785	0.0785	0.2297D-05	0.2690D-04	0.3406D+01	0.3724D+01
0.0628	0.0628	0.1102D-05	0.1191D-04	0.3294D+01	0.3652D+01
0.0524	0.0524	0.6151D-06	0.6277D-05	0.3196D+01	0.3513D+01
0.0449	0.0449	0.3802D-06	0.3742D-05	0.3121D+01	0.3354D+01
0.0393	0.0393	0.2524D-06	0.2437D-05	0.3069D+01	0.3212D+01
0.0349	0.0349	0.1765D-06	0.1691D-05	0.3037D+01	0.3102D+01
0.0314	0.0314	0.1284D-06	0.1230D-05	0.3019D+01	0.3024D+01

Problem 5: $a(x,t)=1$ $b(x,t)=0$ $v(x)=\sin(x)$

$u(x,t)=e^{-tx}\sin(x)$ $f(x,t)=e^{-tx}((1-x-t^2)\sin(x)+2t\cos(x))$

$B= 3.14159$ $T^{*}= 3.14159$

TABLE 6

COEFFICIENT B IS FUNCTION OF X

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.6984D-03	0.8891D-02	0.0	0.0
0.1571	0.1571	0.1649D-03	0.1999D-02	0.2082D+01	0.2153D+01
0.1047	0.1047	0.7317D-04	0.8692D-03	0.2004D+01	0.2053D+01
0.0785	0.0785	0.4116D-04	0.4855D-03	0.1999D+01	0.2025D+01
0.0628	0.0628	0.2635D-04	0.3098D-03	0.1999D+01	0.2014D+01
0.0524	0.0524	0.1830D-04	0.2148D-03	0.1999D+01	0.2009D+01
0.0449	0.0449	0.1345D-04	0.1576D-03	0.1999D+01	0.2006D+01
0.0393	0.0393	0.1030D-04	0.1206D-03	0.1999D+01	0.2005D+01
0.0349	0.0349	0.8136D-05	0.9526D-04	0.1999D+01	0.2003D+01
0.0314	0.0314	0.6591D-05	0.7714D-04	0.2000D+01	0.2003D+01
			2X2 DIRK		
0.3142	0.3142	0.3520D-03	0.4486D-02	0.0	0.0
0.1571	0.1571	0.4500D-04	0.5898D-03	0.2968D+01	0.2927D+01
0.1047	0.1047	0.1442D-04	0.1797D-03	0.2807D+01	0.2932D+01
0.0785	0.0785	0.6447D-05	0.7811D-04	0.2797D+01	0.2895D+01
0.0628	0.0628	0.3450D-05	0.4115D-04	0.2802D+01	0.2872D+01
0.0524	0.0524	0.2067D-05	0.2443D-04	0.2809D+01	0.2860D+01
0.0449	0.0449	0.1339D-05	0.1574D-04	0.2816D+01	0.2853D+01
0.0393	0.0393	0.9188D-06	0.1076D-04	0.2822D+01	0.2850D+01
0.0349	0.0349	0.6586D-06	0.7690D-05	0.2827D+01	0.2848D+01
0.0314	0.0314	0.4887D-06	0.5697D-05	0.2832D+01	0.2848D+01
			3X3 DIRK		
0.3142	0.3142	0.2891D-03	0.3515D-02	0.0	0.0
0.1571	0.1571	0.2548D-04	0.3515D-03	0.3504D+01	0.3322D+01
0.1047	0.1047	0.6440D-05	0.8250D-04	0.3393D+01	0.3575D+01
0.0785	0.0785	0.2468D-05	0.2912D-04	0.3334D+01	0.3620D+01
0.0628	0.0628	0.1192D-05	0.1313D-04	0.3263D+01	0.3568D+01
0.0524	0.0524	0.6654D-06	0.6982D-05	0.3195D+01	0.3465D+01
0.0449	0.0449	0.4100D-06	0.4167D-05	0.3141D+01	0.3348D+01
0.0393	0.0393	0.2710D-06	0.2703D-05	0.3101D+01	0.3241D+01
0.0349	0.0349	0.1886D-06	0.1865D-05	0.3075D+01	0.3153D+01
0.0314	0.0314	0.1367D-06	0.1347D-05	0.3059D+01	0.3089D+01

Problem 6: $a(x,t)=1$ $b(x,t)=x$ $v(x)=\sin(x)$

$u(x,t)=e^{-tx}\sin(x)$ $f(x,t)=e^{-tx}((1-t^2)\sin(x)+2t\cos(x))$

$B= 3.14159$ $T^*= 3.14159$

TABLE 7

COEFFICIENTS NONCONSTANT BUT FORCING TERM ZERO

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.1402D-02	0.2587D-01	0.0	0.0
0.1571	0.1571	0.3500D-03	0.6462D-02	0.2002D+01	0.2001D+01
0.1047	0.1047	0.1555D-03	0.2871D-02	0.2001D+01	0.2000D+01
0.0785	0.0785	0.8747D-04	0.1615D-02	0.2000D+01	0.2000D+01
0.0628	0.0628	0.5598D-04	0.1034D-02	0.2000D+01	0.2000D+01
0.0524	0.0524	0.3888D-04	0.7178D-03	0.2000D+01	0.2000D+01
0.0449	0.0449	0.2856D-04	0.5273D-03	0.2000D+01	0.2000D+01
0.0393	0.0393	0.2187D-04	0.4037D-03	0.2000D+01	0.2000D+01
0.0349	0.0349	0.1728D-04	0.3190D-03	0.2000D+01	0.2000D+01
0.0314	0.0314	0.1399D-04	0.2584D-03	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.3142	0.3142	0.3541D-03	0.6524D-02	0.0	0.0
0.1571	0.1571	0.5059D-04	0.9332D-03	0.2807D+01	0.2805D+01
0.1047	0.1047	0.1575D-04	0.2906D-03	0.2878D+01	0.2877D+01
0.0785	0.0785	0.6818D-05	0.1258D-03	0.2910D+01	0.2910D+01
0.0628	0.0628	0.3546D-05	0.6546D-04	0.2929D+01	0.2929D+01
0.0524	0.0524	0.2074D-05	0.3829D-04	0.2941D+01	0.2941D+01
0.0449	0.0449	0.1316D-05	0.2430D-04	0.2950D+01	0.2950D+01
0.0393	0.0393	0.8871D-06	0.1638D-04	0.2956D+01	0.2956D+01
0.0349	0.0349	0.6259D-06	0.1155D-04	0.2961D+01	0.2961D+01
0.0314	0.0314	0.4579D-06	0.8454D-05	0.2965D+01	0.2965D+01
			3X3 DIRK		
0.3142	0.3142	0.1330D-03	0.2442D-02	0.0	0.0
0.1571	0.1571	0.1153D-04	0.2120D-03	0.3528D+01	0.3526D+01
0.1047	0.1047	0.2573D-05	0.4734D-04	0.3699D+01	0.3698D+01
0.0785	0.0785	0.8678D-06	0.1597D-04	0.3778D+01	0.3777D+01
0.0628	0.0628	0.3697D-06	0.6804D-05	0.3824D+01	0.3823D+01
0.0524	0.0524	0.1830D-06	0.3370D-05	0.3855D+01	0.3854D+01
0.0449	0.0449	0.1007D-06	0.1854D-05	0.3879D+01	0.3875D+01
0.0393	0.0393	0.5976D-07	0.1103D-05	0.3906D+01	0.3891D+01
0.0349	0.0349	0.3754D-07	0.6965D-06	0.3946D+01	0.3903D+01
0.0314	0.0314	0.2457D-07	0.4612D-06	0.4024D+01	0.3913D+01

Problem 7: $a(x,t)=1+t$ $b(x,t)=-t$ $v(x)=\sin(x)$

$u(x,t)=e^{-t}\sin(x)$ $f(x,t)=0$

$B= 3.14159$ $T^{*}= 3.14159$

TABLE 8

COEFFICIENT A IS FUNCTION OF T

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.8159D-03	0.1505D-01	0.0	0.0
0.1571	0.1571	0.2062D-03	0.3807D-02	0.1984D+01	0.1983D+01
0.1047	0.1047	0.9184D-04	0.1695D-02	0.1995D+01	0.1995D+01
0.0785	0.0785	0.5169D-04	0.9544D-03	0.1998D+01	0.1997D+01
0.0628	0.0628	0.3309D-04	0.6110D-03	0.1999D+01	0.1998D+01
0.0524	0.0524	0.2299D-04	0.4244D-03	0.1999D+01	0.1999D+01
0.0449	0.0449	0.1689D-04	0.3118D-03	0.1999D+01	0.1999D+01
0.0393	0.0393	0.1293D-04	0.2388D-03	0.1999D+01	0.1999D+01
0.0349	0.0349	0.1022D-04	0.1887D-03	0.2000D+01	0.2000D+01
0.0314	0.0314	0.8277D-05	0.1528D-03	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.3142	0.3142	0.3632D-03	0.6692D-02	0.0	0.0
0.1571	0.1571	0.6310D-04	0.1164D-02	0.2525D+01	0.2523D+01
0.1047	0.1047	0.2148D-04	0.3965D-03	0.2657D+01	0.2657D+01
0.0785	0.0785	0.9795D-05	0.1808D-03	0.2730D+01	0.2730D+01
0.0628	0.0628	0.5271D-05	0.9729D-04	0.2777D+01	0.2777D+01
0.0524	0.0524	0.3158D-05	0.5829D-04	0.2810D+01	0.2810D+01
0.0449	0.0449	0.2040D-05	0.3766D-04	0.2834D+01	0.2834D+01
0.0393	0.0393	0.1394D-05	0.2573D-04	0.2853D+01	0.2853D+01
0.0349	0.0349	0.9941D-06	0.1835D-04	0.2868D+01	0.2868D+01
0.0314	0.0314	0.7339D-06	0.1355D-04	0.2881D+01	0.2880D+01
			3X3 DIRK		
0.3142	0.3142	0.2327D-03	0.4283D-02	0.0	0.0
0.1571	0.1571	0.3064D-04	0.5648D-03	0.2925D+01	0.2923D+01
0.1047	0.1047	0.8392D-05	0.1548D-03	0.3194D+01	0.3193D+01
0.0785	0.0785	0.3198D-05	0.5900D-04	0.3353D+01	0.3353D+01
0.0628	0.0628	0.1478D-05	0.2726D-04	0.3460D+01	0.3460D+01
0.0524	0.0524	0.7754D-06	0.1431D-04	0.3537D+01	0.3536D+01
0.0449	0.0449	0.4455D-06	0.8221D-05	0.3594D+01	0.3594D+01
0.0393	0.0393	0.2740D-06	0.5057D-05	0.3640D+01	0.3639D+01
0.0349	0.0349	0.1777D-06	0.3280D-05	0.3677D+01	0.3675D+01
0.0314	0.0314	0.1202D-06	0.2220D-05	0.3708D+01	0.3704D+01

Problem 8: $a(x,t)=1+t$ $b(x,t)=0$ $v(x)=\sin(x)$

$u(x,t)=e^{-t}\sin(x)$ $f(x,t)=te^{-t}\sin(x)$

$B= 3.14159$ $T^{*}= 3.14159$

TABLE 9

COEFFICIENT A IS FUNCTION OF BOTH T AND X

DELX	DELT	L2 ERROR	REL. ERROR	L2 RATE	REL RATE
			1X1 DIRK		
0.3142	0.3142	0.7878D-03	0.1453D-01	0.0	0.0
0.1571	0.1571	0.1991D-03	0.3675D-02	0.1984D+01	0.1983D+01
0.1047	0.1047	0.8866D-04	0.1637D-02	0.1995D+01	0.1995D+01
0.0785	0.0785	0.4990D-04	0.9214D-03	0.1998D+01	0.1997D+01
0.0628	0.0628	0.3195D-04	0.5899D-03	0.1999D+01	0.1998D+01
0.0524	0.0524	0.2219D-04	0.4097D-03	0.1999D+01	0.1999D+01
0.0449	0.0449	0.1630D-04	0.3010D-03	0.1999D+01	0.1999D+01
0.0393	0.0393	0.1248D-04	0.2305D-03	0.1999D+01	0.1999D+01
0.0349	0.0349	0.9865D-05	0.1821D-03	0.2000D+01	0.2000D+01
0.0314	0.0314	0.7991D-05	0.1475D-03	0.2000D+01	0.2000D+01
			2X2 DIRK		
0.3142	0.3142	0.3687D-03	0.6794D-02	0.0	0.0
0.1571	0.1571	0.6732D-04	0.1242D-02	0.2453D+01	0.2452D+01
0.1047	0.1047	0.2380D-04	0.4393D-03	0.2564D+01	0.2563D+01
0.0785	0.0785	0.1119D-04	0.2065D-03	0.2624D+01	0.2624D+01
0.0628	0.0628	0.6174D-05	0.1140D-03	0.2664D+01	0.2664D+01
0.0524	0.0524	0.3778D-05	0.6975D-04	0.2693D+01	0.2693D+01
0.0449	0.0449	0.2486D-05	0.4589D-04	0.2716D+01	0.2716D+01
0.0393	0.0393	0.1725D-05	0.3185D-04	0.2735D+01	0.2735D+01
0.0349	0.0349	0.1248D-05	0.2303D-04	0.2751D+01	0.2751D+01
0.0314	0.0314	0.9322D-06	0.1721D-04	0.2765D+01	0.2765D+01
			3X3 DIRK		
0.3142	0.3142	0.2488D-03	0.4580D-02	0.0	0.0
0.1571	0.1571	0.3724D-04	0.6867D-03	0.2740D+01	0.2737D+01
0.1047	0.1047	0.1155D-04	0.2132D-03	0.2886D+01	0.2885D+01
0.0785	0.0785	0.4942D-05	0.9121D-04	0.2952D+01	0.2951D+01
0.0628	0.0628	0.2533D-05	0.4675D-04	0.2996D+01	0.2995D+01
0.0524	0.0524	0.1457D-05	0.2689D-04	0.3034D+01	0.3034D+01
0.0449	0.0449	0.9073D-06	0.1675D-04	0.3071D+01	0.3071D+01
0.0393	0.0393	0.5993D-06	0.1106D-04	0.3106D+01	0.3106D+01
0.0349	0.0349	0.4140D-06	0.7643D-05	0.3140D+01	0.3140D+01
0.0314	0.0314	0.2964D-06	0.5472D-05	0.3172D+01	0.3171D+01

Problem 9: $a(x,t)=1+xt$ $b(x,t)=0$ $v(x)=\sin(x)$

$u(x,t)=e^{-t}\sin(x)$ $f(x,t)=te^{-t}(x\sin(x)-\cos(x))$

B= 3.14149 T*= 3.14159

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Alexander. Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s. SIAM J. Numer. Anal. Vol. 14, No. 6, Dec. 1977, pp 1006-1021.

[2] G.A. Baker, J.H. Bramble and V. Thomee. Single Step Galerkin Approximations for Parabolic Problems. Math Comp., 31(1977),pp 818-847.

[3] M. Crouzeix. Sur l'Approximation des quations Differentienlles Operationnelles Lineaires par des Methods de Runge-Kutta. Thesis, University of Paris V1, 1975.

[4] O.A. Karakashian. On Runge-Kutta Methods for Parabolic Problems with Time Dependent Coefficients, preprint.

APPENDIX

RUNGE KUTTA TABLEAU VALUES

$q = 1$, (Crank-Nicholson)

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline 1 & \end{array}$$

$q = 2$, $\beta = (1+3^{-\frac{1}{2}})/2$

$$\begin{array}{cc|c} \beta & 0 & \beta \\ 1-2\beta & \beta & 1-\beta \\ \hline 1/2 & 1/2 & \end{array}$$

$q = 3$, $\beta = 3^{-\frac{1}{2}}\cos(\pi/18)+1/2$, $b = 1/(6(2\beta-1)^2)$

$$\begin{array}{ccc|c} \beta & 0 & 0 & \beta \\ 1/2-\beta & \beta & 0 & 1/2 \\ 2\beta & 1-4\beta & \beta & \beta \\ \hline b & 1-2b & b & \end{array}$$

PROGRAM LISTING

```
C THIS PROGRAM BY JOHN TRAVIS, UT MATH DEPT.
C In this program, we solve a version of the heat equation in one time
C and one space variable.  The space term is solved using a Galerkin
C type method with cubic splines.  The time term is solved using DIRK
C Runge-Kutta methods.  Output is the L2 error and a relative error at
C the final time step.  Then, for different mesh sizes, convergence of
C the method is calculated.  Note, for each different set of functions
C given, the program computes the approx. solution using 3 different
C Runge-Kutta methods and mesh sizes ranging by tens from 10 to 100.
C
C Variable list (only for major global variables )
C    GRAM   - Contains the "Gram Matrix".
C    STIFF  - Contains the "Stiffness Matrix" and the left-hand
C                  side when solving the systems of equations.
C    PREC   - Contains the preconditioner used in the Conjugate
C                  Gradient solver.
C    BVEC   - Contains the right-hand side and temp storage.
C    X      - Contains the node values along the x-axis from a to b.
C    DELX   - Contains the step sizes between x-nodes.
C    STEP   - Contains the Quadrature points.
C    COEF   - Contains the Quadrature coefficients.
C    F      - Contains the basis function values for Cubic splines
C                  which have support over only 4 intervals.  Columns
C                  1-4 for 1st interval, columns 5-8 for middle interval
C                  and columns 9-12 for last interval.
C    ALPHAN - Contains the coefficients at the current time step for
C                  the function approximation.
C    ALPHA  - Contains the coefficients for intermediate values.
C    TAB    - Defines the Runge-Kutta method.  A table of constants.
C    C1,C2  - Contain certain coefficients needed for Runge-Kutta step.
C    N      - Number of interior nodes in both space.
C    NQ     - Number of "stages" in the Runge-Kutta method.
C    NK     - Number of steps in time.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(10),B(10)
      COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
      COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
      COMMON /A3/ F(10,12)
      COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
      COMMON /A5/ SOLN(102),GRAM(102,4),GU(102),ALPHAN(102)
     2    ,ALPHA(102,3),ALPHAT(102)
      COMMON /A6/ TAB(4,4),C1(4),C2(3,3),NQ,NQP1
      EXTERNAL FNF,FNY,FNV,CALCS,CALCG,QUAD,QUADG
      CALL ENDVAL(XINIT,XFINAL,TINIT,TFINAL)
```

```
      WRITE(6,12)
 12   FORMAT(2X,'|',62('-'),'|')
      WRITE(6,11) XINIT,XFINAL,TFINAL
 11   FORMAT(2X,'|',8X,'XA=',F11.5,', XB=',F11.5,', T*=',F11.5,8X,'|')
C Mesh points and quadrature coefficients.
      STEP(1)=.0469100770306680D0
      STEP(2)=.2307653449471584D0
      STEP(3)=.5D0
      STEP(4)=.7692346550528416D0
      STEP(5)=.9530899229693320D0
      COEF(1)=.2369268850561888D0
      COEF(2)=.4786286704993665D0
      COEF(3)=.5688888888888892D0
      COEF(4)=.4786286704993662D0
      COEF(5)=.2369268850561891D0
      CALL LOADPH
      DO 1000 NQ=1,3
      NPREV=0
      EPREV1=0.0D0
      EPREV2=0.0D0
      WRITE(6,5)
 5    FORMAT(2X,'|',62('-'),'|')
      WRITE(6,6) NQ,NQ
 6    FORMAT(2X,'|',27X,I1,'X',I1,' DIRK',27X,'|')
      WRITE(6,9)
 9    FORMAT(2X,'| DELX   DELT   L2 ERROR    REL. ERROR    L2 RATE
     2REL RATE |')
      DO 1000 ILOOP=1,10
      N=10*ILOOP-1
      NK=10*ILOOP
      NKP1=NK+1
      NQP1=NQ+1
      RN=DFLOAT(N)
      RNP1=RN+1.0D0
      NM1=N-1
      NM2=N-2
      NP1=N+1
      NP2=N+2
      XINT=(XFINAL-XINIT)/RNP1
      DO 10 I=1,NP2
10      X(I)=XINIT+XINT*DFLOAT(I-1)
      DO 15 I=1,NP1
        IP1=I+1
15      DELX(I)=X(IP1)-X(I)
      TIME=0.0D0
      DELK=(TFINAL-TINIT)/DFLOAT(NK)
      CALL TABLE(DELK)
      CALL LOADS(QUADG,CALCG,TINIT)
```

```
      DO 20 J=1,4
        DO 20 I=1,NP2
 20       GRAM(I,J)=STIFF(I,J)
      CALL LOADB(FNV,TINIT)
      DO 23 I=1,NP2
 23     GU(I)=BVEC(I)
      CALL LEQ1PB(STIFF,NP2,3,102,BVEC,102,1,IDGT,D1,D2,IER)
      CALL LOADS(QUAD,CALCS,TINIT)
      DO 25 I=1,NP2
        ALPHAN(I)=BVEC(I)
        DO 25 J=1,4
 25       STIFF(I,J)=GRAM(I,J)+TAB(1,1)*STIFF(I,J)
      CALL LUDAPB(STIFF,NP2,3,102,PREC,102,D1,D2,IER)
C
      DO 200 K=1,NK
        CALL TSTEP(TIME)
        CALL VMULQF(GRAM,NP2,3,102,ALPHAN,1,102,GU,102)
        TIME=TIME+DELK
 200  CONTINUE
      CALL ERRORS(DELK,TFINAL,XINT,EPREV1,EPREV2,NPREV)
 1000 CONTINUE
      WRITE(6,299)
 299  FORMAT(2X,'|',62('-'),'|')
      STOP
      END
C
C  This function evaluates STIFF(I,J).
      FUNCTION QUAD(A,B,H)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(10),B(10)
      COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
      T=0.D0
      HSQ=H*H
      DO 1510 J=1,5
        JP5=J+5
        TEMP=FA(J)*A(JP5)*B(JP5)/HSQ+FB(J)*A(J)*B(J)
 1510   T=T+TEMP*COEF(J)
      QUAD=T*H/2.0D0
      RETURN
      END
C
C  This function evaluates BVEC(I).
      FUNCTION QUADB(A,H)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(10)
      COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
      T=0.0D0
      DO 1610 J=1,5
```

```
 1610     T=T+COEF(J)*FF(J)*A(J)
        QUADB=T*H/2.0D0
        RETURN
        END
C
C   This function evaluates GRAM(I,J).
        FUNCTION QUADG(A,B,H)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION A(10),B(10)
        COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
        T=0.0D0
        DO 1710 J=1,5
 1710     T=T+A(J)*B(J)*COEF(J)
        QUADG=T*H/2.0D0
        RETURN
        END
C
C  This function determines function values for Stiffness matrix.
        FUNCTION CALCS(H,XI,TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
        DO 10 J=1,5
          XP=XI+H*STEP(J)
          FA(J)=FNA(XP,TIME)
 10       FB(J)=FNB(XP,TIME)
        CALCS=1.0D0
        RETURN
        END
C
C  This function is a placebo.
        FUNCTION CALCG(H,XI,TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        CALCG=1.0D0
        RETURN
        END
C
C  This subroutine calculates the L2 and relative errors.
        SUBROUTINE ERRORS(DELK,TIME,XINT,EPREV1,EPREV2,NPREV)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(10)
        COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
        COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
        COMMON /A3/ F(10,12)
        COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
        COMMON /A5/ SOLN(102),GRAM(102,4),GU(102),ALPHAN(102)
     2    ,ALPHA(102,3),ALPHAT(102)
        COMMON /A6/ TAB(4,4),C1(4),C2(3,3),NQ,NQP1
        EXTERNAL FNY
```

```
      SUM=0.0D0
      SUM1=0.0D0
      SUM2=0.0D0
      FNORM=0.0D0
      CALL LOADB(FNY,TIME)
      DO 85 I=1,NP1
        DO 80 J=1,5
          XP=X(I)+DELX(I)*STEP(J)
          FF(J)=FNY(XP,TIME)
 80       A(J)=FF(J)
 85     SUM=SUM+QUADB(A,DELX(I))
      DO 90 I=1,NP2
        SUM=SUM+ALPHAN(I)*GU(I)
        SUM1=SUM1+ALPHAN(I)*BVEC(I)
        FY=FNY(X(I),TIME)
        SUM2=SUM2+(SOLN(I)-FY)**2
 90     FNORM=FNORM+FY*FY
      ERROR1=DSQRT(SUM2/FNORM)
      ERR2P=DABS(SUM-2.0D0*SUM1)
      ERROR2=DSQRT(ERR2P)
      RED1=0.0D0
      RED2=0.0D0
      IF(NPREV.EQ.0) GOTO 99
      TEMP=DLOG( DFLOAT(NK)/DFLOAT(NPREV) )
      RED1=DLOG(EPREV1/ERROR1)/TEMP
      RED2=DLOG(EPREV2/ERROR2)/TEMP
 99   WRITE(6,101) DELX(1),DELK,ERROR2,ERROR1,RED2,RED1
      NPREV=NK
      EPREV1=ERROR1
      EPREV2=ERROR2
 101  FORMAT(2X,'|',2F7.4,4(D12.4),'|')
      RETURN
      END
C
C  This subroutine performs the Conjugate-Gradient method.
C  Routine uses a preconditioner equal to the actual stiffness matrix
C  initial time.  This is factored once and stored in PREC.
      SUBROUTINE PCGM
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
      COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
      COMMON /A5/ SOLN(102),GRAM(102,4),GU(102),ALPHAN(102)
     2    ,ALPHA(102,3),ALPHAT(102)
      DIMENSION R(102),D(102),XN(102),AR(102),AD(102)
      CALL VMULQF(STIFF,NP2,3,102,ALPHAN,1,102,XN,102)
      DO 10 I=1,NP2
        R(I)=BVEC(I)-XN(I)
 10     XN(I)=ALPHAN(I)
```

```
      CALL LUELPB(PREC,R,NP2,3,102,D)
      DELTA=0.0D0
      DO 20 I=1,NP2
20      DELTA=DELTA+R(I)*D(I)
      ITER=0
      CALL VMULQF(STIFF,NP2,3,102,D,1,102,AD,102)
      ALPHAG=0.0D0
      DO 30 I=1,NP2
30      ALPHAG=ALPHAG+D(I)*AD(I)
      ALPHAG=DELTA/ALPHAG
      SUM=0.0D0
      DO 40 I=1,NP2
        S1=R(I)-ALPHAG*AD(I)
        SUM=SUM+S1*S1
40      XN(I)=XN(I)+ALPHAG*D(I)
      IF(SUM.LT.1E-20) GOTO 105
100   CONTINUE
        DO 50 I=1,NP2
50        R(I)=R(I)-ALPHAG*AD(I)
        CALL LUELPB(PREC,R,NP2,3,102,AR)
        DELTAN=0.0D0
        DO 60 I=1,NP2
60        DELTAN=DELTAN+R(I)*AR(I)
        BETA=DELTAN/DELTA
        DELTA=DELTAN
        DO 70 I=1,NP2
70        D(I)=AR(I)+BETA*D(I)
        CALL VMULQF(STIFF,NP2,3,102,D,1,102,AD,102)
        ALPHAG=0.0D0
        DO 75 I=1,NP2
75        ALPHAG=ALPHAG+D(I)*AD(I)
        ALPHAG=DELTA/ALPHAG
        DO 77 I=1,NP2
77        XN(I)=XN(I)+ALPHAG*D(I)
        ITER=ITER+1
      IF((DABS(DELTA).GT.1E-20).AND.(ITER.LT.20)) GOTO 100
105   DO 110 I=1,NP2
110     BVEC(I)=XN(I)
      RETURN
      END
C
C  This routine sets up initial tableau and method coefficients.
      SUBROUTINE TABLE(DELK)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AI(4,4)
      COMMON /A6/ TAB(4,4),C1(4),C2(3,3),NQ,NQP1
      GO TO (1,2,3),NQ
 1    TAB(1,1)=0.5D0
```

```
      TAB(1,2)=0.5D0
      TAB(2,1)=1.0D0
      GOTO 5
2     BETA=(1.0D0+1.0D0/DSQRT(3.0D0))/2.0D0
      TAB(1,1)=BETA
      TAB(1,3)=BETA
      TAB(2,1)=(1.0D0-2.0D0*BETA)
      TAB(2,2)=BETA
      TAB(2,3)=(1.0D0-BETA)
      TAB(3,1)=0.5D0
      TAB(3,2)=0.5D0
      C2(2,1)=TAB(2,1)/TAB(1,1)
      GOTO 5
3     PI=4.0D0*DATAN(1.0D0)
      BETA=DCOS(PI/18.0D0)/DSQRT(3.0D0)+0.5D0
      TAB(1,1)=BETA
      TAB(1,4)=BETA
      TAB(2,1)=(0.5D0-BETA)
      TAB(2,2)=BETA
      TAB(2,4)=0.5D0
      TAB(3,1)=2.0D0*BETA
      TAB(3,2)=(1.0D0-4.0D0*BETA)
      TAB(3,3)=BETA
      TAB(3,4)=(1.0D0-BETA)
      TAB(4,1)=1.0D0/(6.0D0*(2.0D0*BETA-1.0D0)**2)
      TAB(4,2)=1.0D0-2.0D0*TAB(4,1)
      TAB(4,3)=TAB(4,1)
      BSQ=BETA*BETA
      C2(2,1)=TAB(2,1)/TAB(1,1)
      C2(3,1)=(TAB(3,1)*TAB(2,2)-TAB(3,2)*TAB(2,1))/(TAB(1,1)*TAB(2,2))
      C2(3,2)=TAB(3,2)/TAB(2,2)
5     CONTINUE
      AI(1,1)=1.0D0/TAB(1,1)
      GOTO (30,20,10),NQ
10    AI(3,1)=(TAB(2,1)*TAB(3,2)-TAB(3,1)*TAB(3,3))/(TAB(3,3)**3)
      AI(3,2)=-1.0D0*TAB(3,2)/(TAB(3,3)*TAB(3,3))
      AI(3,3)=AI(1,1)
20    AI(2,1)=-1.0D0*TAB(2,1)/(TAB(2,2)*TAB(2,2))
      AI(2,2)=AI(1,1)
30    DO 40 I=1,NQ
        SUM=0.0D0
        DO 35 J=I,NQ
35        SUM=SUM+AI(J,I)*TAB(NQP1,J)
        IP1=I+1
40      C1(IP1)=SUM
      SUM=0.0D0
      DO 50 I=2,NQP1
50      SUM=SUM+C1(I)
```

```
      C1(1)=1.0D0-SUM
      DO 60 I=1,NQ
       DO 60 J=1,NQP1
60        TAB(I,J)=DELK*TAB(I,J)
      RETURN
      END
C
C  This routine evaluates U(N+1) given U(N).
      SUBROUTINE TSTEP(TIME)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
      COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
      COMMON /A3/ F(10,12)
      COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
      COMMON /A5/ SOLN(102),GRAM(102,4),GU(102),ALPHAN(102)
     2   ,ALPHA(102,3),ALPHAT(102)
      COMMON /A6/ TAB(4,4),C1(4),C2(3,3),NQ,NQP1
      EXTERNAL FNF,CALCS,QUAD
      DO 80 I1=1,NQ
        I1M1=I1-1
        T1=TIME+TAB(I1,NQP1)
        CALL LOADS(QUAD,CALCS,T1)
        DO 50 I=1,NP2
50        ALPHAT(I)=0.0D0
        IF(I1.EQ.1) GOTO 75
        DO 60 J=1,NP2
          SUM=0.0D0
          DO 65 KK=1,I1M1
65          SUM=SUM+C2(I1,KK)*(ALPHA(J,KK)-ALPHAN(J))
60        BVEC(J)=SUM
        CALL VMULQF(GRAM,NP2,3,102,BVEC,1,102,ALPHAT,102)
75      CALL LOADB(FNF,T1)
        DO 70 I=1,NP2
          BVEC(I)=GU(I)+TAB(I1,I1)*BVEC(I)+ALPHAT(I)
          DO 70 J=1,4
70          STIFF(I,J)=GRAM(I,J)+TAB(I1,I1)*STIFF(I,J)
        CALL PCGM
        DO 80 I=1,NP2
          ALPHA(I,I1)=BVEC(I)
80    CONTINUE
      DO 90 I=1,NP2
        SUM=0.0D0
        DO 85 J=1,NQ
          JP1=J+1
85        SUM=SUM+C1(JP1)*ALPHA(I,J)
90      ALPHAN(I)=C1(1)*ALPHAN(I)+SUM
      DO 95 I=2,NP1
        IP1=I+1
```

```
         IM1=I-1
 95      SOLN(I)=(ALPHAN(IP1)+ALPHAN(IM1))/24.0D0+ALPHAN(I)/6.0D0
       SOLN(1)=0.0D0
       SOLN(NP2)=0.0D0
       RETURN
       END
C
C   This routine loads the basis elements.  Note, symmetry.
       SUBROUTINE LOADPH
       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
       COMMON /A3/ F(10,12)
       DO 20 J=1,5
         JP5=J+5
         P2=       STEP(J)
         P1= 1.0D0+STEP(J)
         F(J,1)=((2.0D0-P1)**3)/24.0D0
         F(J,2)=((2.0D0-P2)**3)/24.0D0-((1.0D0-P2)**3)/6.0D0
         F(JP5,1)=-1.0D0*((2.0D0-P1)**2)/8.0D0
 20      F(JP5,2)=((1.0D0-P2)**2)/2.0D0-((2.0D0-P2)**2)/8.0D0
       DO 30 J=1,5
         JP5=J+5
         J1=6-J
         J2=11-J
         F(J1,3)=F(J,2)
         F(J1,4)=F(J,1)
         F(J2,3)=-1.0D0*F(JP5,2)
 30      F(J2,4)=-1.0D0*F(JP5,1)
       DO 40 J=1,5
         JP5=J+5
         F(J,5)=F(J,1)
         F(J,6)=F(J,2)-4.0D0*F(J,1)
         F(J,7)=F(J,3)-F(J,1)
         F(J,8)=F(J,4)
         F(JP5,5)=F(JP5,1)
         F(JP5,6)=F(JP5,2)-4.0D0*F(JP5,1)
         F(JP5,7)=F(JP5,3)-F(JP5,1)
         F(JP5,8)=F(JP5,4)
         F(J,9)=F(J,1)
         F(J,10)=F(J,2)-F(J,4)
         F(J,11)=F(J,3)-4.0D0*F(J,4)
         F(J,12)=F(J,4)
         F(JP5,9)=F(JP5,1)
         F(JP5,10)=F(JP5,2)-F(JP5,4)
         F(JP5,11)=F(JP5,3)-4.0D0*F(JP5,4)
 40      F(JP5,12)=F(JP5,4)
       RETURN
```

```
      END
C
C  This routine calculates both the stiffness matrix and the
C  gram matrix dependent upon the supplied function from the main
C  program through external parameters.
      SUBROUTINE LOADS(Q,CAL,TIME)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(10),B(10)
      COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
      COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
      COMMON /A3/ F(10,12)
      COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
      DO 10 J=1,4
        DO 10 I=1,NP2
10        STIFF(I,J)=0.0D0
      H=DELX(1)
      R=CAL(H,X(1),TIME)
      DO 23 K=1,3
        KP5=K+5
        CALL LOADV(KP5,A)
        CALL LOADV(KP5,B)
 23     STIFF(K,4)=Q(A,B,H)
      DO 24 K=2,3
        KP5=K+5
        KP4=K+4
        CALL LOADV(KP4,A)
        CALL LOADV(KP5,B)
 24     STIFF(K,3)=Q(A,B,H)
      CALL LOADV(6,A)
      CALL LOADV(8,B)
      STIFF(3,2)=Q(A,B,H)
      DO 30 I=2,N
        IM2=I-2
        H=DELX(I)
        R=CAL(H,X(I),TIME)
        DO 30 IBAND=1,4
        ICOL=5-IBAND
        DO 30 K=IBAND,4
          KP=K+1-IBAND
            CALL LOADV(KP,A)
            CALL LOADV(K,B)
            IROW=IM2+K
            STIFF(IROW,ICOL)=STIFF(IROW,ICOL)+Q(A,B,H)
 30   CONTINUE
      H=DELX(NP1)
      R=CAL(H,X(NP1),TIME)
      DO 37 K=1,3
        KP8=K+8
```

```
          CALL LOADV(KP8,A)
          CALL LOADV(KP8,B)
          IROW=NM1+K
 37       STIFF(IROW,4)=STIFF(IROW,4)+Q(A,B,H)
        DO 38 K=2,3
          KP7=K+7
          KP8=K+8
          CALL LOADV(KP7,A)
          CALL LOADV(KP8,B)
          IROW=NM1+K
 38       STIFF(IROW,3)=STIFF(IROW,3)+Q(A,B,H)
        CALL LOADV(9,A)
        CALL LOADV(11,B)
        STIFF(NP2,2)=STIFF(NP2,2)+Q(A,B,H)
        RETURN
        END
C
C  This subroutine calculates the right hand side, BVEC.
        SUBROUTINE LOADB(FUN,TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION A(10)
        COMMON /A1/ STIFF(102,4),PREC(102,4),BVEC(102),X(102),DELX(102)
        COMMON /A2/ FA(5),FB(5),FF(5),STEP(5),COEF(5)
        COMMON /A3/ F(10,12)
        COMMON /A4/ NM2,NM1,N,NP1,NP2,NK,NKP1
        DO 10 I=1,NP2
 10       BVEC(I)=0.0D0
        H=DELX(1)
        DO 20 J=1,5
          XP=X(1)+H*STEP(J)
 20       FF(J)=FUN(XP,TIME)
        DO 25 K=1,3
          KP5=K+5
          CALL LOADV(KP5,A)
 25       BVEC(K)=QUADB(A,H)
        DO 30 I=2,N
          IM2=I-2
          H=DELX(I)
          DO 35 J=1,5
            XP=X(I)+H*STEP(J)
 35         FF(J)=FUN(XP,TIME)
          DO 30 K=1,4
            CALL LOADV(K,A)
            IROW=IM2+K
            BVEC(IROW)=BVEC(IROW)+QUADB(A,H)
 30     CONTINUE
        H=DELX(NP1)
        DO 40 J=1,5
```

```
          XP=X(NP1)+H*STEP(J)
 40       FF(J)=FUN(XP,TIME)
        DO 45 K=1,3
          KP8=K+8
          CALL LOADV(KP8,A)
          IROW=NM1+K
 45       BVEC(IROW)=BVEC(IROW)+QUADB(A,H)
        RETURN
        END
C
C   This subroutine transfers data from matrix F to vectors A and B.
        SUBROUTINE LOADV(KK,AA)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION AA(10)
        COMMON /A3/ F(10,12)
        DO 3100 J=1,10
 3100     AA(J)=F(J,KK)
        RETURN
        END
C
C  This function is fiven in D.E. as A(X,T).
        FUNCTION FNA(X,T)
        IMPLICIT  REAL*8 (A-H,O-Z)
        FNA=1.0D0+T*X
        RETURN
        END
C
C  This function is given in D.E. as B(X,T)
        FUNCTION FNB(X,T)
        IMPLICIT REAL*8 (A-H,O-Z)
        FNB=0.0D0
        RETURN
        END
C
C  This function gives V(X)=U(X,0), the initial value.
        FUNCTION FNV(X,T)
        IMPLICIT REAL*8(A-H,O-Z)
        FNV=DSIN(X)
        RETURN
        END
C
C  This function is fiven in D.E. as F(X,T).
        FUNCTION FNF(X,T)
        IMPLICIT REAL*8 (A-H,O-Z)
        FNF=T*DEXP(-1.0D0*T)*(X*DSIN(X)-DCOS(X))
        RETURN
```

```
      END
C
C  This function evaluates the exact solution.
      FUNCTION FNY(X,T)
      IMPLICIT REAL*8 (A-H,O-Z)
      FNY=DEXP(-1.0D0*T)*DSIN(X)
      RETURN
      END
C
C  This subroutine gives the endpoint parameters for P.D.E.
      SUBROUTINE ENDVAL(XINIT,XFINAL,TINIT,TFINAL)
      IMPLICIT REAL*8(A-H,O-Z)
      PI=4.0D0*DATAN(1.0D0)
      XINIT=0.0D0
      XFINAL=PI
      TINIT=0.0D0
      TFINAL=PI
      RETURN
      END
```

VITA

John Paul Travis was born in New Orleans, Louisiana, on May 11, 1960. He attended elementary school in Blue Mountain, Mississippi, and graduated from Blue Mountain High School in May, 1978. The following August, he entered Mississippi College in Clinton, Mississippi, and in May 1982 received a Bachelor of Arts degree in Mathematics. In the fall of 1982, he accepted a teaching assistantship at The University of Tennessee, Knoxville, and began study toward a Master's degree. This degree was awarded in August, 1985.

The author is a nominee member of the American Mathematical Society as well as a member of the Society of Industrial and Applied Mathematics. Mr. Travis plans to pursue his doctoral degree at The University of Tennessee after graduation.