

**A Comparative Computational Study of the
Numerical Integration of Parabolic PDEs
by Finite Differences, Finite Elements
and Finite Volumes**

W. Schiesser and A. Yücel

Superconducting Super Collider Laboratory*
2550 Beckleymeade Ave.
Dallas, TX 75237

November 1994

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

Index

1 Introduction	1
2 The Test Problems	1
3 Finite Differences	5
3.1 Dirichlet Boundary Conditions	5
3.2 Neumann Boundary Conditions	12
4 Finite Elements	19
4.1 Dirichlet Boundary Conditions	24
4.2 Neumann Boundary Conditions	31
5 Finite Volumes	41
5.1 Dirichlet Boundary Conditions	45
5.2 Neumann Boundary Conditions	50
6 Boundary Conditions of the Third Type and Conductances	55
6.1 Finite Differences	55
6.2 Finite Volumes	62
7 Summary	67
<i>References</i>	68
<i>Appendix 1: Derivation of the Analytical Solution for the Parabolic Test Problem with Neumann Boundary Conditions</i>	69
<i>Appendix 2: Main Program for the Numerical Solution of the Test Problems via Subroutine LSODES</i>	71

(1) Introduction

This study was undertaken to compare the features and performance of the three principal numerical approximations to partial differential equations (PDEs), finite differences, finite elements and finite volumes. We made this comparative study by applying these three approximations to a single parabolic test problem, Fourier's second law (the heat conduction equation) with Dirichlet and Neumann boundary conditions, and boundary conditions of the third type. The results of this study should in general provide some insight into the characteristics and performance of the three methods of approximation to PDEs which are first order in time and second order in space. The implementation of the three approximations is done consistently within the numerical method of lines format so that results can be compared on this consistent basis. Also, the derivation of the approximating ODEs is done at a tutorial level, and the method of lines programs to solve the approximating ODEs are provided in the body of the discussion or in appendices so that the reader will know exactly how we performed the calculations. All of the software to produce the solutions we discuss is available from the authors.

(2) The Test Problems

We now consider the finite difference, finite element and finite volume solutions of Fourier's second law (the one dimensional heat conduction equation)

$$u_t = Du_{xx} \tag{1}$$

with the initial condition

$$u(x,0) = g(x) \tag{2}$$

and either Dirichlet bounditions

$$u(x_1,t) = f_1(t), u(x_N,t) = f_N(t) \tag{3}(4)$$

or Neumann boundary conditions

$$u_x(x_1,t) = h_1(t), u_x(x_N,t) = h_N(t) \tag{5}(6)$$

(boundary conditions of the third type will be considered subsequently).

In particular, we use the special cases $x_1 = 0$, $x_N = 1$ with

Case 1:

$$u(x,0) = \sin(\pi x) \quad (7)$$

$$u(x_1,t) = 0, u(x_N,t) = 0 \quad (8)(9)$$

Case 2:

$$u(x,0) = \begin{cases} 0, & 0 \leq x < 0.5 \\ 1, & 0.5 < x \leq 1 \end{cases} \quad (10)$$

$$u_x(x_1,t) = 0, u_x(x_N,t) = 0 \quad (11)(12)$$

Thus, Case 1 involves Dirichlet boundary conditions and Case 2 involves Neumann boundary conditions. For Case 1, the exact solution is simply

$$u(x,t) = e^{-\pi^2 t} \sin(\pi x) \quad (13)$$

which is used to compute the error in the three numerical solutions (from the three methods of approximation).

The solution to Case 2 is derived in Appendix 1 as

$$u(x,t) = \sum_{n=0}^{\infty} C_n e^{-\lambda_n^2 t} \cos(\lambda_n x) \quad (14)$$

where $\lambda_0 = 0$, $C_0 = 1/2$, $\lambda_n = n\pi$, $C_n = (\frac{2}{n\pi})\sin(0.5n\pi)$, $n = 1, 2, 3, \dots$. A function SERIES to calculate $u(x,t)$ from equation (14) is included with the programs for the finite difference, element and volume numerical solutions of equations (1), (10), (11) and (12) for comparison of these numerical solutions with the analytical solution (equation (14)). Also, we can compute the total energy of the system, which should remain constant because of the "insulated boundary conditions", equations (11) and (12), and observe the approach to the steady state solution $u(\infty,t) = 1/2$.

For each of the three spatial approximations (finite differences, elements and volumes), we use a uniform spatial grid of 50 sections, and the same ODE integrator with the same error tolerances, so that differences in the computed solutions result only from the three spatial approximations. We now consider each approximation applied to the two test problems. Generally, the approach is to discretize

the spatial derivative, u_{xx} , in equation (1), by one of the three approximations, then integrate the resulting system of ODEs with respect to t . The initial conditions for the ODEs are set in subroutine INITIAL, the ODE derivatives are computed in subroutine DERV and the solution is printed in subroutine PRINT. These three subroutines are in Program 1 for the first test problem (Case 1, equations (1), (7), (8) and (9))

SUBROUTINE INITIAL

```

C...
C... THE TEST PROBLEM FOR DIRICHLET BOUNDARY CONDITIONS IS
C...
C...      U   = D*U                               (1)
C...      T     XX
C...
C...      U(X,0) = SIN(PI*X/L)                   (2)
C...
C...      U(0,T) = 0, U(L,T) = 0                 (3)(4)
C...
C... WITH THE EXACT SOLUTION
C...
C...      UE(X,T) = EXP(-(PI**2/L**2)*T)SIN(PI*X/L) (5)
C...
C... THE FOLLOWING CODE COMPUTES A NUMERICAL SOLUTION TO EQUATIONS
C... (1) TO (4) BY
C...
C...      (1) FINITE DIFFERENCES
C...
C...          NORUN = 1, THREE POINT DIFFERENCES
C...
C...          NORUN = 2, FIVE POINT DIFFERENCES
C...
C...      (2) FINITE ELEMENTS
C...
C...          NORUN = 3, LINEAR FINITE ELEMENTS (HAT FUNCTIONS)
C...
C...      (3) FINITE VOLUMES
C...
C...          NORUN = 4, EQUAL VOLUMES
C...
C... IN EACH CASE, A 51 POINT GRID IS USED. THE NUMERICAL SOLUTION,
C... THE EXACT SOLUTION, EQUATION (5), AND THE DIFFERENCES BETWEEN
C... THE TWO SOLUTIONS ARE PRINTED SO THE RELATIVE ACCURACIES OF THE
C... THREE APPROXIMATIONS CAN BE ASSESSED.
C...
C... COMMON/T/      T, NSTOP, NORUN
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE DIFFERENCES
C... IF((NORUN.EQ.1).OR.(NORUN.EQ.2))CALL INIT1
C...

```

```

C... SOLUTION OF EQUATIONS (1) TO (4) BY LINEAR FINITE ELEMENTS (HAT
C... FUNCTIONS)
      IF(NORUN.EQ.3)CALL INIT2
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE VOLUMES
      IF(NORUN.EQ.4)CALL INIT3
      RETURN
      END

      SUBROUTINE DERV
      COMMON/T/          T,  NSTOP,  NORUN
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE DIFFERENCES
      IF((NORUN.EQ.1).OR.(NORUN.EQ.2))CALL DERV1
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY LINEAR FINITE ELEMENTS (HAT
C... FUNCTIONS)
      IF(NORUN.EQ.3)CALL DERV2
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE VOLUMES
      IF(NORUN.EQ.4)CALL DERV3
      RETURN
      END

      SUBROUTINE PRINT(NI,NO)
      COMMON/T/          T,  NSTOP,  NORUN
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE DIFFERENCES
      IF((NORUN.EQ.1).OR.(NORUN.EQ.2))CALL PRINT1(NI,NO)
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY LINEAR FINITE ELEMENTS (HAT
C... FUNCTIONS)
      IF(NORUN.EQ.3)CALL PRINT2(NI,NO)
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE VOLUMES
      IF(NORUN.EQ.4)CALL PRINT3(NI,NO)
      RETURN
      END

```

Program 1: Subroutines INITAL, DERV and PRINT for Case 1 Test Problem

Note that these three subroutines merely call other subroutines for the implementation of the Case 1 test problem. Thus, INITAL calls (1) INIT1 to implement initial condition (7) (or (2) in the comments) with finite differences ($NORUN = 1$ for three point approximations and $NORUN = 2$ for five point approximations), (2) INIT2 for initial condition (7) with finite elements and (3) INIT3 for initial condition (7) with finite volumes. Similarly, DERV calls DERV1, DERV2 and DERV3 to

compute the ODE derivatives resulting from finite differences, elements and volumes, respectively, applied to equation (1) with boundary conditions (8) and (9). Finally, PRINT calls PRINT1, PRINT2 and PRINT3 to print the solutions from finite differences, elements and volumes, respectively. Subroutines INITIAL, DERV and PRINT for the Case 2 test problem, equations (1), (10), (11) and (12) are essentially identical. We now proceed to a discussion of the numerical methods and subroutines for each of the approximations.

(3) Finite Differences

Solutions to the two test problems are computed by three point and five point finite differences approximations that are discussed in detail elsewhere [Schiesser (1)]. Therefore, we consider only the details of the computer program.

(3.1) Dirichlet Boundary Conditions

Subroutines INIT1, DERV1 and PRINT1 are listed in Program 2 (Case 1 test problem, equations (1) and (7) to (9) with finite differences)

```

SUBROUTINE INIT1
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE DIFFERENCES
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N), PI
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... PI
PI=4.0*ATAN(1.0)
C...
C... LENGTH
L=1.0
C...
C... GRID SPACING
DX=L/FLOAT(N-1)
C...
C... INITIAL CONDITION (2)
DO 1 I=1,N
X(I)=FLOAT(I-1)*DX

```

```

      U(I)=SIN(PI*X(I)/L)
1     CONTINUE
C...
C...  INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/
      CALL DERV
      IP=0
      RETURN
      END

      SUBROUTINE DERV1
C...
C...  SUBROUTINE DERV1 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C...  ODES IN THE FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
C...
      PARAMETER (N=51)
      COMMON/T/      T,  NSTOP,  NORUN
1         /Y/      U(N)
2         /F/      UT(N)
3         /XG/     L,    DX,    X(N),    PI
4         /I/      IP
C...
C...  TYPE SELECTED VARIABLES AS REAL
      REAL L
C...
C...  DEFINE ARRAYS FOR THE SPATIAL DERIVATIVES
      REAL UX(N), UXX(N)
C...
C...  BOUNDARY CONDITION (3)
      U(1)=0.
      UT(1)=0.
      NL=1
C...
C...  BOUNDARY CONDITION (4)
      U(N)=0
      UT(N)=0.
      NU=1
C...
C...  U BY THREE POINT DIFFERENCES
      XX
      IF(NORUN.EQ.1)CALL DSS042(0.,L,N,U,UX,UXX,NL,NU)
C...
C...  U BY FIVE POINT DIFFERENCES
      XX
      IF(NORUN.EQ.2)CALL DSS044(0.,L,N,U,UX,UXX,NL,NU)
C...
C...  EQUATION (1)
      DO 1 I=2,N-1
          UT(I)=UXX(I)
1     CONTINUE
      RETURN
      END

```

```

SUBROUTINE PRINT1(NI,NO)
C...
C... SUBROUTINE PRINT1 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N), PI
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L, UE(N), DIFF(N)
C...
C... CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C... NUMERICAL AND EXACT SOLUTIONS
DO 1 I=1,N,10
UE(I)=EXACT(I)
DIFF(I)=U(I)-UE(I)
1 CONTINUE
C...
C... PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
WRITE(NO,2)T,(U(I),I=1,N,10),
1 (UE(I),I=1,N,10),
2 (DIFF(I),I=1,N,10)
2 FORMAT(' T = ',F6.2,/,15X,' X=0',5X,'X=0.2',5X,'X=0.4',5X,
1 'X=0.6',5X,'X=0.8',5X,' X=1',/,
2 ' U(X,T)',6F10.6,/,
3 ' UE(X,T)',6F10.6,/,
4 ' DIFF(X,T)',6F10.6,/)
RETURN
END

REAL FUNCTION EXACT(I)
C...
C... FUNCTION EXACT COMPUTES THE EXACT SOLUTION TO EQUATIONS (1) TO
C... TO (4), I.E., EQUATION (5), AT GRID INDEX I
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N), PI
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... EXACT SOLUTION AT X AND T

```

```

EXACT=EXP((-PI**2/L**2)*T)*SIN(PI*X(I)/L)
RETURN
END

```

Program 2: Solution of Case 1 Test Problem by Finite Differences

We can note the following details concerning Program 2:

- (1) Initial condition (7) is defined over a 51 point grid in INIT1.
- (2) The derivative u_{xx} is computed by subroutine DSS042 in DERV1 for NORUN = 1 (three point approximations) and by subroutine DSS044 for NORUN = 2 (five point approximations). The details of the differentiation formulas in DSS042 and DSS044, and the significance of the arguments of these subroutines are discussed elsewhere [Schiesser (1)].
- (3) PRINT1 computes the exact solution from equation (13) and the difference between the numerical and exact solutions. It then prints the two solutions and their difference.

The solution is printed at $t = 0, 0.1, 0.2, \dots, 0.5$ (PRINT1 is called at these times), as specified by the data in Program 3

```

FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
0.      0.5      0.1      0.0000001
  51
FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
0.      0.5      0.1      0.0000001
  51
FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
0.      0.5      0.1      0.0000001
  51
FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
0.      0.5      0.1      0.0000001
  50
END OF RUNS

```

Program 3: Data Used by Program 2

These data are for four runs. The first and second runs (NORUN = 1 and 2 in Program 1) are for finite differences as implemented in INIT1, DERV1 and PRINT1. The third run is for finite elements

as implemented in INIT2, DERV2 and PRINT2 (NORUN = 3), and the fourth run is for finite volumes as implemented in INIT3, DERV3 and PRINT3 (NORUN =3). Note that for each run (each set of data), time runs from 0 to 0.5 in steps of 0.1 (e.g., PRINT1 is called every 0.1 time units to print the numerical and exact solutions). 51 ODEs are specified (for the 51 point spatial grid) and the error tolerance of the ODE integration is 0.0000001. This stringent error tolerance, used by ODE integrator LSODES which is called by the main program in Appendix 2, was selected to insure that the ODE integration did not contribute significantly to the numerical solution; thus, any error in the numerical solutions can be attributed to the spatial approximation of equation (1).

The output from Programs 1, 2 and 3 (in combination with the main program of Appendix 2 and ODE integrator LSODES) is given in Table 1 for the first and second runs

```

RUN NO. - 1 FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
FINAL T - 0.500E+00
PRINT T - 0.100E+00
NUMBER OF DIFFERENTIAL EQUATIONS - 51
INTEGRATION ALGORITHM - LSODES
MAXIMUM INTEGRATION ERROR - 0.100E-06

T = 0.00
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1
U(X,T) 0.000000 0.587785 0.951057 0.951057 0.587785 0.000000
UE(X,T) 0.000000 0.587785 0.951057 0.951057 0.587785 0.000000
DIFF(X,T) 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

T = 0.10
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1
U(X,T) 0.000000 0.219144 0.354582 0.354582 0.219144 0.000000
UE(X,T) 0.000000 0.219072 0.354466 0.354466 0.219072 0.000000
DIFF(X,T) 0.000000 0.000072 0.000116 0.000116 0.000072 0.000000

T = 0.20
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1
U(X,T) 0.000000 0.081703 0.132198 0.132198 0.081703 0.000000
UE(X,T) 0.000000 0.081650 0.132112 0.132112 0.081650 0.000000
DIFF(X,T) 0.000000 0.000053 0.000086 0.000086 0.000053 0.000000

T = 0.30

```

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.030461	0.049287	0.049287	0.030461	0.000000
UE(X,T)	0.000000	0.030432	0.049239	0.049239	0.030432	0.000000
DIFF(X,T)	0.000000	0.000030	0.000048	0.000048	0.000030	0.000000

T = 0.40

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.011357	0.018376	0.018376	0.011357	0.000000
UE(X,T)	0.000000	0.011342	0.018352	0.018352	0.011342	0.000000
DIFF(X,T)	0.000000	0.000015	0.000024	0.000024	0.000015	0.000000

T = 0.50

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.004234	0.006851	0.006851	0.004234	0.000000
UE(X,T)	0.000000	0.004227	0.006840	0.006840	0.004227	0.000000
DIFF(X,T)	0.000000	0.000007	0.000011	0.000011	0.000007	0.000000

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.126E-01
LAST ORDER OF THE METHOD	5
TOTAL NUMBER OF STEPS TAKEN	108
NUMBER OF FUNCTION EVALUATIONS	161
NUMBER OF JACOBIAN EVALUATIONS	2

RUN NO. - 2 FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)

INITIAL T - 0.000E+00

FINAL T - 0.500E+00

PRINT T - 0.100E+00

NUMBER OF DIFFERENTIAL EQUATIONS - 51

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-06

T = 0.00

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.587785	0.951057	0.951057	0.587785	0.000000
UE(X,T)	0.000000	0.587785	0.951057	0.951057	0.587785	0.000000
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

T = 0.10						
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.219071	0.354465	0.354465	0.219071	0.000000
UE(X,T)	0.000000	0.219072	0.354466	0.354466	0.219072	0.000000
DIFF(X,T)	0.000000	-0.000001	-0.000001	-0.000001	-0.000001	0.000000
T = 0.20						
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.081650	0.132113	0.132113	0.081650	0.000000
UE(X,T)	0.000000	0.081650	0.132112	0.132112	0.081650	0.000000
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
T = 0.30						
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.030432	0.049240	0.049240	0.030432	0.000000
UE(X,T)	0.000000	0.030432	0.049239	0.049239	0.030432	0.000000
DIFF(X,T)	0.000000	0.000000	0.000001	0.000001	0.000000	0.000000
T = 0.40						
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.011342	0.018352	0.018352	0.011342	0.000000
UE(X,T)	0.000000	0.011342	0.018352	0.018352	0.011342	0.000000
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
T = 0.50						
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.004227	0.006840	0.006840	0.004227	0.000000
UE(X,T)	0.000000	0.004227	0.006840	0.006840	0.004227	0.000000
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.232E-02
LAST ORDER OF THE METHOD	4
TOTAL NUMBER OF STEPS TAKEN	179
NUMBER OF FUNCTION EVALUATIONS	324
NUMBER OF JACOBIAN EVALUATIONS	7

Table 1: Output from Programs 1, 2 and 3

We note the following points about this output:

- (1) The numerical solution has better agreement with the exact solution for the five point

approximations (of DSS044) than for the three point approximations (of DSS042), as expected.

(2) The computational effort for LSODES to integrate the 51 ODEs was modest (as reflected in the computational statistics). This contrasts with the explicit ODE integrator RKF45 which produced solutions with reasonable effort for the finite differences (NORUN = 1 and 2), but failed because of excessive computational effort for the finite elements, apparently because of stiffness of the ODEs (NORUN = 3).

(3) The calculations were performed on a 32 bit computer for which single precision Fortran gave acceptable results.

(4) Parenthetically, the solutions were essentially identical for an ODE error tolerance of 0.000001, indicating that the ODE integration error was practically negligible.

Also, as a point that can be better appreciated when the finite element solution is discussed next, the programming of equations (1) and (7) to (9) was quite straightforward (v. Program 2).

(3.2) Neumann Boundary Conditions

The coding of the Case 2 test problem, equations (1) and (10) to (12), is given in Program 4

```
SUBROUTINE INIT1
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE DIFFERENCES
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... LENGTH
L=1.0
C...
C... GRID SPACING
DX=L/FLOAT(N-1)
C...
C... INITIAL CONDITION (2)
```

```

DO 1 I=1,N
  X(I)=FLOAT(I-1)*DX
  IF(I.LT.(N+1)/2)U(I)=0.
  IF(I.GT.(N+1)/2)U(I)=1.
  IF(I.EQ.(N+1)/2)U(I)=0.5
1  CONTINUE
C...
C...  INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/
CALL DERV
IP=0
RETURN
END

SUBROUTINE DERV1
C...
C...  SUBROUTINE DERV1 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C...  ODES IN THE FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
C...
PARAMETER (N=51)
COMMON/T/      T,  NSTOP,  NORUN
1  /Y/      U(N)
2  /F/      UT(N)
3  /XG/     L,    DX,    X(N)
4  /I/      IP

C...
C...  TYPE SELECTED VARIABLES AS REAL
REAL L

C...
C...  DEFINE ARRAYS FOR THE SPATIAL DERIVATIVES
REAL UX(N), UXX(N)

C...
C...  BOUNDARY CONDITION (3)
UX(1)=0.
NL=2

C...
C...  BOUNDARY CONDITION (4)
UX(N)=0.
NU=2

C...
C...  U BY THREE POINT DIFFERENCES
C...  XX
IF(NORUN.EQ.1)CALL DSS042(0.,L,N,U,UX,UXX,NL,NU)

C...
C...  U BY FIVE POINT DIFFERENCES
C...  XX
IF(NORUN.EQ.2)CALL DSS044(0.,L,N,U,UX,UXX,NL,NU)

C...
C...  EQUATION (1)
DO 1 I=1,N
  UT(I)=UXX(I)
1  CONTINUE

```

```

RETURN
END

SUBROUTINE PRINT1(NI,NO)
C...
C... SUBROUTINE PRINT1 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L, UE(N), DIFF(N)
C...
C... COMPUTE THE TOTAL ENERGY (RHO = CP = 1)
CALL DERV1
ENERGY=0.
DO 1 I=2,N
ENERGY=ENERGY+0.5*(U(I)+U(I-1))*DX
1 CONTINUE
C...
C... CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C... NUMERICAL AND EXACT SOLUTIONS
DO 3 I=1,N,10
UE(I)=SERIES(I)
DIFF(I)=U(I)-UE(I)
3 CONTINUE
C...
C... PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
WRITE(NO,2)T,ENERGY,(U(I),I=1,N,10),
1 (UE(I),I=1,N,10),
2 (DIFF(I),I=1,N,10)
2 FORMAT(' T = ',F6.2,' ENERGY = ',F9.5,
1 /,14X,' X=0',5X,'X=0.2',5X,'X=0.4',
2 5X,'X=0.6',5X,'X=0.8',5X,'X=1.0'/,
3 ' U(X,T)',6F10.6,/,
4 ' UE(X,T)',6F10.6,/,
5 ' DIFF(X,T)',6F10.6,/)
RETURN
END

REAL FUNCTION SERIES(I)
C...
C... PROGRAM SERIES EVALUATES THE ANALYTICAL SOLUTION TO EQUATIONS
C... (1), (10), (11) AND (12), I.E., EQUATION (14)
C...
PARAMETER (M=51)

```

```

COMMON/T/      T,  NSTOP,  NORUN
1      /Y/      U(M)
2      /F/      UT(M)
3      /XG/     L,    DX,    X(M)
4      /I/      IP

C...
C...  TYPE SELECTED VARIABLES AS REAL
      REAL L

C...
C...  DEFINE MAXIMUM NUMBER OF TERMS IN THE SERIES (THIS IS SET TO A
C...  HIGH VALUE TO ACHIEVE CONVERGENCE TO THREE FIGURES FOR T = 0;
C...  BEYOND T = 0, THE SERIES CONVERGES RAPIDLY)
      PARAMETER (NS=1000)

C...
C...  PI
      PI=4.0*ATAN(1.0)

C...
C...  EVALUATE SERIES SOLUTION
      SUM=0.5
      SIGN=1.0
      DO 1 N=1,NS,2
          EN=FLOAT(N)*PI
          PN=-(EN**2)*T
          SIGN=-1.0*SIGN
          CN=SIGN*2.0/(FLOAT(N)*PI)
          SUM=SUM+CN*EXP(PN)*COS(EN*X(I))

C...
C...  AVOID UNDERFLOW OF THE EXP FUNCTION
      IF(PN.LT.-100.0)GO TO 2
1      CONTINUE

C...
C...  SUMMATION OF SERIES IS COMPLETE
2      SERIES=SUM
      END

```

Program 4: Solution of Case 2 Test Problem by Finite Differences

Note that subroutine PRINT1 computes the total energy of the solution, E, as

$$E = \int_0^1 H(u) dx \quad (15)$$

where H(u) is the enthalpy per unit volume which is a function of temperature u, and can be computed as

$$H(u) = \int_0^u \rho(u) C_p(u) du \quad (16)$$

For equation (1), $D = k/(\rho C_p) = 1$ and therefore we take $\rho = 1$ and $C_p = 1$ in equation (16). Thus, $H(u) = u$, which, when substituted in equation (15) gives

$$E = \int_0^1 u \, dx \quad (17)$$

The integral in equation (17) is evaluated numerically by the trapezoidal rule in subroutine PRINT1 over the 51 point grid in x each time PRINT1 is called. The energy (ENERGY in the code) is then printed with the numerical solution.

Since we are interested in the steady state numerical solution (recall that from equation (14), $u(x, \infty) = 0.5$), the solutions were computed to $t = 1$ (rather 0.5 as specified by the data in Program 3). Abbreviated output from Program 4 is listed in Table 2 (for the first and second runs)

```

RUN NO. - 1 FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
FINAL T - 0.100E+01
PRINT T - 0.100E+00
NUMBER OF DIFFERENTIAL EQUATIONS - 51
INTEGRATION ALGORITHM - LSODES
MAXIMUM INTEGRATION ERROR - 0.100E-06

T = 0.00 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
UE(X,T) 0.000318 0.000393 0.001030 0.998971 0.999607 0.999682
DIFF(X,T) -0.000318 -0.000393 -0.001030 0.001029 0.000393 0.000318

T = 0.10 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.262758 0.308032 0.426654 0.573347 0.691968 0.737242
UE(X,T) 0.262756 0.308033 0.426655 0.573345 0.691967 0.737244
DIFF(X,T) 0.000001 -0.000001 -0.000001 0.000001 0.000001 -0.000002

T = 0.20 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.411539 0.428433 0.472664 0.527336 0.571567 0.588461
UE(X,T) 0.411566 0.428456 0.472673 0.527327 0.571544 0.588434
DIFF(X,T) -0.000027 -0.000023 -0.000009 0.000009 0.000023 0.000027

```

T = 0.30 ENERGY = 0.50000

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0
U(X,T)	0.467019	0.473318	0.489808	0.510192	0.526682	0.532981
UE(X,T)	0.467040	0.473335	0.489815	0.510185	0.526665	0.532960
DIFF(X,T)	-0.000021	-0.000017	-0.000007	0.000006	0.000017	0.000021

T = 0.40 ENERGY = 0.50000

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0
U(X,T)	0.487703	0.490052	0.496200	0.503800	0.509948	0.512296
UE(X,T)	0.487716	0.490062	0.496204	0.503796	0.509938	0.512284
DIFF(X,T)	-0.000012	-0.000010	-0.000004	0.000004	0.000010	0.000012

T = 0.50 ENERGY = 0.50000

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0
U(X,T)	0.495415	0.496291	0.498583	0.501417	0.503709	0.504585
UE(X,T)	0.495421	0.496296	0.498585	0.501415	0.503704	0.504578
DIFF(X,T)	-0.000006	-0.000005	-0.000002	0.000002	0.000005	0.000006

T = 1.00 ENERGY = 0.50000

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0
U(X,T)	0.499967	0.499973	0.499989	0.500010	0.500026	0.500033
UE(X,T)	0.499967	0.499973	0.499990	0.500010	0.500027	0.500033
DIFF(X,T)	-0.000001	0.000000	0.000000	0.000000	0.000000	0.000000

COMPUTATIONAL STATISTICS

LAST STEP SIZE 0.173E-01

LAST ORDER OF THE METHOD 3

TOTAL NUMBER OF STEPS TAKEN 299

NUMBER OF FUNCTION EVALUATIONS 414

NUMBER OF JACOBIAN EVALUATIONS 6

RUN NO. - 2 FINITE DIFFERENCE SOLUTION OF EQUATIONS (1) TO (4)

INITIAL T - 0.000E+00

FINAL T - 0.100E+01

PRINT T - 0.100E+00

NUMBER OF DIFFERENTIAL EQUATIONS - 51

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-06

T = 0.00 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
 UE(X,T) 0.000318 0.000393 0.001030 0.998971 0.999607 0.999682
 DIFF(X,T) -0.000318 -0.000393 -0.001030 0.001029 0.000393 0.000318

T = 0.10 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.262834 0.308095 0.426678 0.573321 0.691904 0.737167
 UE(X,T) 0.262756 0.308033 0.426655 0.573345 0.691967 0.737244
 DIFF(X,T) 0.000078 0.000062 0.000023 -0.000024 -0.000063 -0.000077

T = 0.20 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.411595 0.428478 0.472681 0.527318 0.571520 0.588404
 UE(X,T) 0.411566 0.428456 0.472673 0.527327 0.571544 0.588434
 DIFF(X,T) 0.000029 0.000023 0.000008 -0.000010 -0.000024 -0.000029

T = 0.30 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.467050 0.473343 0.489817 0.510181 0.526655 0.532948
 UE(X,T) 0.467040 0.473335 0.489815 0.510185 0.526665 0.532960
 DIFF(X,T) 0.000010 0.000008 0.000002 -0.000004 -0.000010 -0.000012

T = 0.40 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.487720 0.490064 0.496204 0.503793 0.509933 0.512278
 UE(X,T) 0.487716 0.490062 0.496204 0.503796 0.509938 0.512284
 DIFF(X,T) 0.000004 0.000003 0.000000 -0.000003 -0.000005 -0.000006

T = 0.50 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.495424 0.496297 0.498585 0.501412 0.503700 0.504574
 UE(X,T) 0.495421 0.496296 0.498585 0.501415 0.503704 0.504578
 DIFF(X,T) 0.000002 0.000001 0.000000 -0.000003 -0.000004 -0.000004

 .
 .
 .
 T = 1.00 ENERGY = 0.50000
 X=0 X=0.2 X=0.4 X=0.6 X=0.8 X=1.0
 U(X,T) 0.499965 0.499971 0.499987 0.500008 0.500025 0.500032
 UE(X,T) 0.499967 0.499973 0.499990 0.500010 0.500027 0.500033
 DIFF(X,T) -0.000002 -0.000003 -0.000002 -0.000002 -0.000002 -0.000001

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.366E-02
LAST ORDER OF THE METHOD	1
TOTAL NUMBER OF STEPS TAKEN	646
NUMBER OF FUNCTION EVALUATIONS	1913
NUMBER OF JACOBIAN EVALUATIONS	105

Table 2: Output from Programs 1 and 4

Note that energy is constant to five figures, and the steady state solution $u(x,\infty) = 0.5$ is closely approximated at $t = 1$. Also, the analytical solution implemented in function SERIES does not satisfy initial condition (10) to more than about three figures, even with 1000 terms used in the series of equation (14). However, this series solution converges rapidly for $t > 0$, and therefore the comparison between the numerical and analytical solutions is reliable for $t > 0$.

In summary, the three and five point finite difference approximations produced accurate solutions with straightforward programming. We now consider the finite element solution of the two test problems in the same format so that a comparison of the numerical solutions with the preceding finite difference solutions can be made.

(4) Finite Elements

To develop the finite element algorithm, we start with a separated solution of the form

$$u(x,t) \approx \sum_{i=1}^N c_i(t)\phi_i(x) \quad (18)$$

The derivatives in equation (1) are then

$$u_t(x,t) \approx \sum_{i=1}^N c_i'(t)\phi_i(x) \quad (19)$$

$$u_{xx}(x,t) \approx \sum_{i=1}^N c_i(t)\phi_i''(x) \quad (20)$$

where ($'$) denotes differentiation (there is no confusion about this differentiation since $c_i(t)$ and $\phi_i(x)$ are functions of only one variable). If equations (19) and (20) are substituted in equation (1), we have

$$\sum_{i=1}^N c_i'(t)\phi_i(x) \approx D \sum_{i=1}^N c_i(t)\phi_i''(x) \quad (21)$$

Equation (18) will probably, at best, be only an approximate solution, and therefore equation (1) will not be truly satisfied. Thus we arrange equation (21) as

$$\sum_{i=1}^N c_i'(t)\phi_i(x) - D \sum_{i=1}^N c_i(t)\phi_i''(x) = R(x,t) \quad (22)$$

where $R(x,t)$ is a *residual*, which ideally is zero for all x and t (if we could find the exact solution to equation (1)). Since equation (18) (or (21)) will only be approximate, we attempt to minimize $R(x,t)$ in equation (22) in some fashion. Generally, this is done by forming the integral

$$\int_{x_1}^{x_N} w(x)R(x,t)dx = 0 \quad (23)$$

where $w(x)$ is a *weighting function* selected by the analyst. Thus the use of equation (23) is called the *method of weighted residuals* (as suggested by the integrand in equation (23)). In other words, we are *making the residual, $R(x,t)$, orthogonal to the weighting function, $w(x)$* .

In order to obtain $c_i'(t)$, $i = 1, 2, \dots, N$, we multiply equation (22) by $\phi_j(x)$ and integrate from x_1 to x_N

$$\int_{x_1}^{x_N} \phi_j(x) \sum_{i=1}^N c_i'(t)\phi_i(x)dx - D \int_{x_1}^{x_N} \phi_j(x) \sum_{i=1}^N c_i(t)\phi_i''(x)dx = \int_{x_1}^{x_N} \phi_j(x)R(x,t)dx = 0$$

or

$$\sum_{i=1}^N c_i'(t) \int_{x_1}^{x_N} \phi_j(x)\phi_i(x)dx - D \sum_{i=1}^N c_i(t) \int_{x_1}^{x_N} \phi_j(x)\phi_i''(x)dx = \int_{x_1}^{x_N} \phi_j(x)R(x,t)dx = 0 \quad (24)$$

where we have interchanged the order of integration and summation.

In order to evaluate the integrals of equation (24), we must select some basis (shape) functions. Our choice is the linear finite element (hat or chapeau function) depicted in Figure 1.a, with first and

second derivatives depicted in Figures 1.b and 1.c

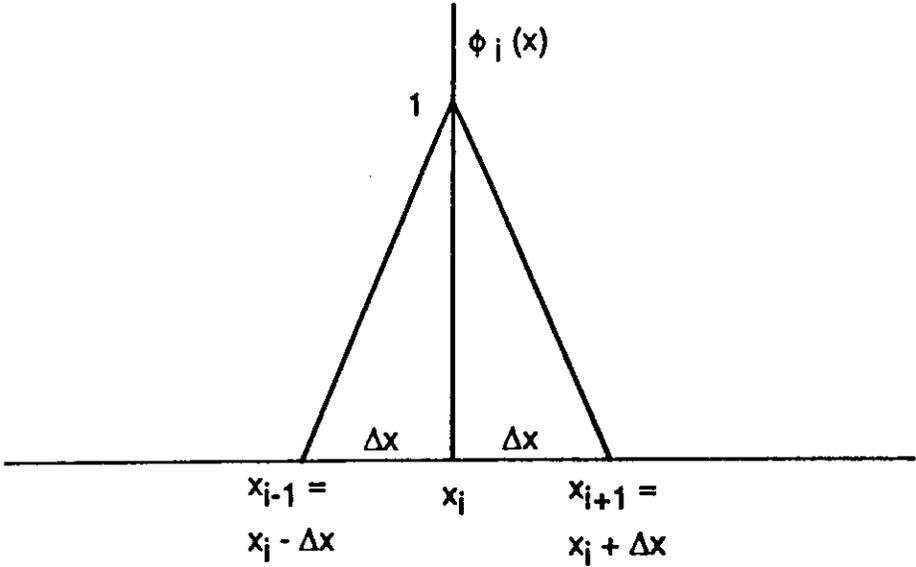


Fig. 1.a: Linear Finite Element

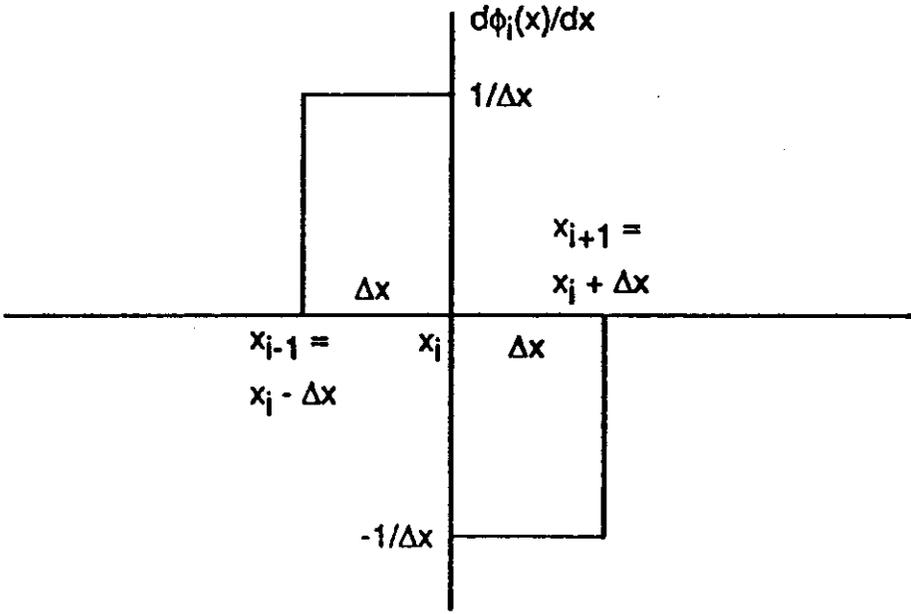


Fig. 1.b: Derivative of Linear Finite Element

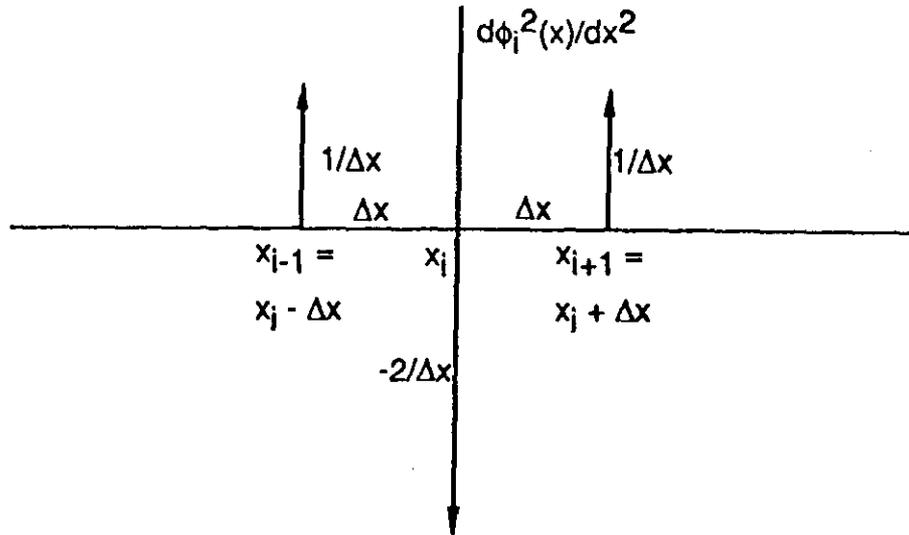


Fig. 1.c: Second Derivative of Linear Finite Element

We now must consider three cases for each of the two LHS integrals in equation (24). For the first integral

$$(1) \int_{x_1}^{x_N} \phi_j(x)\phi_i(x)dx$$

$$(1.1) \quad i = j$$

$$\int_{x_1}^{x_N} \phi_i(x)\phi_i(x)dx = (2/3)\Delta x$$

$$(1.2) \quad i = j+1 \text{ (or } j = i-1)$$

$$\int_{x_1}^{x_N} \phi_i(x)\phi_{i+1}(x)dx = (1/6)\Delta x$$

$$(1.3) \quad i = j-1 \text{ (or } j = i+1)$$

$$\int_{x_1}^{x_N} \phi_i(x)\phi_{i-1}(x)dx = (1/6)\Delta x$$

We have stated here just the final results of the integrations; the details are given else where [Carcagno, et al (2)].

$$(2) \quad \int_{x_1}^{x_N} \phi_j(x)\phi_i''(x)dx$$

$$(2.1) \quad i = j$$

$$\begin{aligned} \int_{x_1}^{x_N} \phi_i(x)\phi_i''(x)dx &= \int_{x_{i-1}}^{x_{i+1}} \left\{ \phi_i(x)(1/\Delta x)\delta(x_i-\Delta x) - 2\phi_i(x)(1/\Delta x)\delta(x_i) + \phi_i(x)(1/\Delta x)\delta(x_i+\Delta x) \right\} dx \\ &= \phi_i(x_i-\Delta x)(1/\Delta x) - 2\phi_i(x_i)(1/\Delta x) + \phi_i(x_i+\Delta x)(1/\Delta x) = -2/\Delta x \end{aligned}$$

$$(2.2) \quad i = j+1 \text{ (or } j = i-1)$$

$$\int_{x_1}^{x_N} \phi_i(x)\phi_{i+1}''(x)dx = \int_{x_i}^{x_{i+1}} \left\{ \phi_i(x)(1/\Delta x)\delta(x_i) - 2\phi_i(x)(1/\Delta x)\delta(x_i+\Delta x) \right\} dx = \phi_i(x_i)(1/\Delta x) = 1/\Delta x$$

$$(2.3) \quad i = j-1 \text{ (or } j = i+1)$$

$$\int_{x_1}^{x_N} \phi_i(x) \phi_{i-1}''(x) dx = \int_{x_{i-1}}^{x_i} \left\{ -2\phi_i(x)(1/\Delta x)\delta(x_i - \Delta x) + \phi_i(x)(1/\Delta x)\delta(x_i) \right\} dx = \phi_i(x_i)(1/\Delta x) = 1/\Delta x$$

Here we have made use of the delta function properties of Figure 1.c.

Thus, equation (24) becomes

$$(\Delta x/6)c_{i+1}' + (4\Delta x/6)c_i' + (\Delta x/6)c_{i-1}' = D(1/\Delta x) \left\{ c_{i+1} - 2c_i + c_{i-1} \right\}$$

or

$$(1/6)c_{i-1}' + (4/6)c_i' + (1/6)c_{i+1}' = D \left\{ \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} \right\} \quad (25)$$

Equation (25) is the (local) Galerkin approximation to the one dimensional heat conduction, equation (1), based on the linear finite element of Figure 1.a. We note in particular that the RHS is just the well known second order central difference approximation for the second derivative u_{xx} . However, the LHS is a weighted sum of three time derivatives.

(4.1) Dirichlet Boundary Conditions

If we now consider the Dirichlet boundary conditions, equations (8) and (9), we also have

$$u(x_1, t) = c_1(t) = f_1(t), \quad u(x_N, t) = c_N(t) = f_N(t) \quad (26)(27)$$

which follows from equation (18) with $\phi_1(x_1) = 1$, $\phi_i(x_1) = 0$, $i = 2, 3, \dots, N$ and $\phi_N(x_N) = 1$, $\phi_i(x_N) = 0$, $i = 1, 2, \dots, N-1$. Then from equations (26) and (27),

$$c_1'(t) = f_1'(t), \quad c_N'(t) = f_N'(t) \quad (28)(29)$$

Thus, the full set of (global) equations for $c_1(t)$ to $c_N(t)$ becomes

$$c_1'(t) = f_1'(t)$$

$$(1/6)c_1' + (4/6)c_2' + (1/6)c_3' = D \left\{ \frac{c_3 - 2c_2 + c_1}{\Delta x^2} \right\}$$

$$\begin{aligned}
(1/6)c_2' + (4/6)c_3' + (1/6)c_4' &= D \left\{ \frac{c_4 - 2c_3 + c_2}{\Delta x^2} \right\} \\
&\vdots \\
(1/6)c_{N-3}' + (4/6)c_{N-2}' + (1/6)c_{N-1}' &= D \left\{ \frac{c_{N-1} - 2c_{N-2} + c_{N-3}}{\Delta x^2} \right\} \\
(1/6)c_{N-2}' + (4/6)c_{N-1}' + (1/6)c_N' &= D \left\{ \frac{c_N - 2c_{N-1} + c_{N-2}}{\Delta x^2} \right\} \\
c_N'(t) &= f_N'(t)
\end{aligned} \tag{30}$$

which can be integrated to obtain $c_1', c_2', \dots, c_{N-1}', c_N'$ subject to the initial conditions

$$c_1(0) = g(x_1), c_2(0) = g(x_2), \dots, c_{N-1}(0) = g(x_{N-1}), c_N(0) = g(x_N) \tag{31}$$

Initial condition (31) follows from equations (2) and (18).

For the special case of $f_1(t) = C_1$ and $f_N(t) = C_N$ where C_1 and C_N are given constants, and therefore $c_1'(t) = 0$ and $c_N'(t) = 0$, equations (30) can be stated as

$$\begin{aligned}
(4/6)c_2' + (1/6)c_3' &= D \left\{ \frac{c_3 - 2c_2 + C_1}{\Delta x^2} \right\} \\
(1/6)c_2' + (4/6)c_3' + (1/6)c_4' &= D \left\{ \frac{c_4 - 2c_3 + c_2}{\Delta x^2} \right\} \\
&\vdots \\
(1/6)c_{N-3}' + (4/6)c_{N-2}' + (1/6)c_{N-1}' &= D \left\{ \frac{c_{N-1} - 2c_{N-2} + c_{N-3}}{\Delta x^2} \right\} \\
(1/6)c_{N-2}' + (4/6)c_{N-1}' &= D \left\{ \frac{C_N - 2c_{N-1} + c_{N-2}}{\Delta x^2} \right\}
\end{aligned} \tag{32}$$

In other words, we integrate only $N-2$ ODEs for $c_2', c_3', \dots, c_{N-2}', c_{N-1}'$ since c_1 and c_N are given by the boundary conditions $c_1(t) = C_1$ and $c_N(t) = C_N$.

Subroutines INIT2, DERV2 and PRINT2 for the solutions of equations (31) and (32), which

are called by Program 1, are listed in Program 5

```
      SUBROUTINE INIT2
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY LINEAR FINITE ELEMENTS
C...
      PARAMETER (N=51)
      COMMON/T/      T,  NSTOP,  NORUN
1      /Y/      U(N)
2      /F/      UT(N)
3      /XG/      L,      DX,      X(N),      PI
4      /I/      IP
5      /FE/      AL(N),  BM(N),  CU(N),  BRHS(N)
C...
C... TYPE SELECTED VARIABLES AS REAL
      REAL L
C...
C... PI
      PI=4.0*ATAN(1.0)
C...
C... LENGTH
      L=1.0
C...
C... GRID SPACING
      DX=L/FLOAT(N-1)
C...
C... SET UP THE COEFFICIENT MATRIX IN BAND STORAGE MODE. THIS IS
C... DONE ONLY ONCE SINCE THE COEFFICIENT MATRIX IS CONSTANT
C...
C... LOWER DIAGONAL
      DO 2 I=1,N
          IF(I.EQ.1)THEN
              AL(1)=0.
          ELSE
+          IF(I.EQ.N)THEN
              AL(N)=0.
          ELSE
              AL(I)=1.0/6.0
          END IF
2      CONTINUE
C...
C... MAIN DIAGONAL
      DO 3 I=1,N
          IF(I.EQ.1)THEN
              BM(1)=1.0
          ELSE
+          IF(I.EQ.N)THEN
              BM(N)=1.0
          ELSE
              BM(I)=4.0/6.0
```

```

        END IF
3      CONTINUE
C...
C...  UPPER DIAGONAL
      DO 4 I=1,N
        IF(I.EQ.1)THEN
          CU(1)=0.
        ELSE
+      IF(I.EQ.N)THEN
          CU(N)=0.
        ELSE
          CU(I)=1.0/6.0
        END IF
4      CONTINUE
C...
C...  INITIAL CONDITION (2)
      DO 1 I=1,N
        X(I)=FLOAT(I-1)*DX
        U(I)=SIN(PI*X(I)/L)
1      CONTINUE
C...
C...  INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/
      CALL DERV
      IP=0
      RETURN
      END

```

SUBROUTINE DERV2

```

C...
C...  SUBROUTINE DERV2 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C...  ODES IN THE FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
C...
      PARAMETER (N=51)
      COMMON/T/      T,  NSTOP,  NORUN
1      /Y/      U(N)
2      /F/      UT(N)
3      /XG/     L,      DX,      X(N),      PI
4      /I/      IP
5      /FE/     AL(N),  BM(N),  CU(N),  BRHS(N)
C...
C...  TYPE SELECTED VARIABLES AS REAL
      REAL L
C...
C...  BOUNDARY CONDITION (3)
      U(1)=0.
C...
C...  BOUNDARY CONDITION (4)
      U(N)=0.
C...
C...  RIGHT HAND SIDE VECTOR (WITH D = 1)
      DO 1 I=1,N

```

```

        IF(I.EQ.1)THEN
            BRHS(1)=0.
        ELSE
+       IF(I.EQ.N)THEN
            BRHS(N)=0.
        ELSE
            BRHS(I)=(1.0/DX**2)*(U(I+1)-2.0*U(I)+U(I-1))
        END IF
1      CONTINUE
C...
C... SOLVE THE LINEAR ALGEBRAIC EQUATIONS BY SUBROUTINE TRIDAG, WHICH
C... RETURNS THE DERIVATIVE VECTOR UT IN COMMON/F/
      CALL TRIDAG(AL,BM,CU,BRHS,UT,N)
      RETURN
      END

      SUBROUTINE PRINT2(NI,NO)
C...
C... SUBROUTINE PRINT2 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
C...
      PARAMETER (N=51)
      COMMON/T/      T,  NSTOP,  NORUN
1      /Y/      U(N)
2      /F/      UT(N)
3      /XG/      L,    DX,    X(N),    PI
4      /I/      IP
5      /FE/      AL(N), BM(N), CU(N), BRHS(N)
C...
C... TYPE SELECTED VARIABLES AS REAL
      REAL L, UE(N), DIFF(N)
C...
C... PRINT THE COEFFICIENT MATRIX FOR VERIFICATION
      IP=IP+1
      IF(IP.EQ.1)THEN
          WRITE(NO,3)
3         FORMAT(/,' COEFFICIENT MATRIX',/)
          DO 4 I=1,N
              WRITE(NO,5)I,AL(I),BM(I),CU(I)
5              FORMAT(I5,3F12.4)
4         CONTINUE
          WRITE(NO,6)
6         FORMAT(//)
      END IF
C...
C... CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C... NUMERICAL AND EXACT SOLUTIONS
      DO 1 I=1,N,10
          UE(I)=EXACT(I)
          DIFF(I)=U(I)-UE(I)
1      CONTINUE

```

```

C...
C... PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
      WRITE(NO,2)T, (U(I), I=1,N,10),
1         (UE(I), I=1,N,10),
2         (DIFF(I), I=1,N,10)
2  FORMAT('      T = ',F6.2,/,15X,'      X=0',5X,'X=0.2',5X,'X=0.4',5X,
1         'X=0.6',5X,'X=0.8',5X,'      X=1',/,
2         '      U(X,T)',6F10.6,/,
3         '      UE(X,T)',6F10.6,/,
4         '      DIFF(X,T)',6F10.6,/)
      RETURN
      END

```

Program 5: Solution of Case 1 Test Problem by Finite Elements

The following points should be noted about Program 5:

(1) The LHS tridiagonal coefficient matrix in equations (32) is set numerically in subroutine INIT2. Then initial condition (31) (or (7)) is implemented in DO loop 1.

(2) The RHS vector of equations (32) is computed in subroutine DERV2 after boundary conditions (8) and (9) are implemented. A call to the tridiagonal solver TRIDAG then computes the derivative vector $c_1'(t)$, $c_2'(t)$, ..., $c_{N-1}'(t)$, $c_N'(t)$ (in array UT) which then goes to the ODE integrator through COMMON/F/.

(3) Subroutine PRINT2 prints the numerical solution at $t = 0, 0.1, 0.2, \dots, 0.5$ (according to the third set of data in Program 3).

The output from Program 5 is listed in Table 3

```

RUN NO. - 3 FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
      FINAL T - 0.500E+00
      PRINT T - 0.100E+00
NUMBER OF DIFFERENTIAL EQUATIONS - 51
INTEGRATION ALGORITHM - LSODES

```

MAXIMUM INTEGRATION ERROR - 0.100E-06

COEFFICIENT MATRIX

1	0.0000	1.0000	0.0000
2	0.1667	0.6667	0.1667
3	0.1667	0.6667	0.1667
4	0.1667	0.6667	0.1667
5	0.1667	0.6667	0.1667
6	0.1667	0.6667	0.1667
7	0.1667	0.6667	0.1667
8	0.1667	0.6667	0.1667
9	0.1667	0.6667	0.1667
10	0.1667	0.6667	0.1667
.	.	.	.
45	0.1667	0.6667	0.1667
46	0.1667	0.6667	0.1667
47	0.1667	0.6667	0.1667
48	0.1667	0.6667	0.1667
49	0.1667	0.6667	0.1667
50	0.1667	0.6667	0.1667
51	0.0000	1.0000	0.0000

T = 0.00

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.587785	0.951057	0.951057	0.587785	0.000000
UE(X,T)	0.000000	0.587785	0.951057	0.951057	0.587785	0.000000
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

T = 0.10

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.219002	0.354352	0.354352	0.219002	0.000000
UE(X,T)	0.000000	0.219072	0.354466	0.354466	0.219072	0.000000
DIFF(X,T)	0.000000	-0.000070	-0.000114	-0.000114	-0.000071	0.000000

T = 0.20

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.081597	0.132027	0.132027	0.081597	0.000000
UE(X,T)	0.000000	0.081650	0.132112	0.132112	0.081650	0.000000
DIFF(X,T)	0.000000	-0.000053	-0.000086	-0.000086	-0.000053	0.000000

T = 0.30

	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.030402	0.049192	0.049192	0.030402	0.000000
UE(X,T)	0.000000	0.030432	0.049239	0.049239	0.030432	0.000000
DIFF(X,T)	0.000000	-0.000030	-0.000048	-0.000048	-0.000030	0.000000

T =	0.40						
		X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.011327	0.018328	0.018328	0.011327	0.000000	
UE(X,T)	0.000000	0.011342	0.018352	0.018352	0.011342	0.000000	
DIFF(X,T)	0.000000	-0.000015	-0.000024	-0.000024	-0.000015	0.000000	
T =	0.50						
		X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.004220	0.006829	0.006829	0.004220	0.000000	
UE(X,T)	0.000000	0.004227	0.006840	0.006840	0.004227	0.000000	
DIFF(X,T)	0.000000	-0.000007	-0.000011	-0.000011	-0.000007	0.000000	

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.112E-01
LAST ORDER OF THE METHOD	4
TOTAL NUMBER OF STEPS TAKEN	137
NUMBER OF FUNCTION EVALUATIONS	220
NUMBER OF JACOBIAN EVALUATIONS	3

Table 3: Output from Programs 1 and 5

The solution has better than 0.1% accuracy. Also, Program 5 was first executed with explicit ODE integrator RKF45, which reported excessive work (the number of calls to DERV2 exceeded the default set in RKF45). We therefore switched to the implicit ODE integrator LSODES, which is called in the main program of Appendix 2.

(4.2) Neumann Boundary Conditions

For the case of Neumann boundary conditions (11) and (12), we first write the original PDE, equation (1), in residual form

$$u_t - Du_{xx} = R(x,t)$$

(recall that u will only be approximate so that in general, for some x and t , $R(x,t) \neq 0$). Then we substitute this residual in the Galerkin integral

$$\int_{x_1}^{x_N} w(x)R(x,t)dx = \int_{x_1}^{x_N} w(x) \left\{ u_t - Du_{xx} \right\} dx = \int_{x_1}^{x_N} \phi_i(x) \left\{ u_t - Du_{xx} \right\} dx = 0 \quad (33)$$

We can now consider two integrals. The first involves u_t

$$\int_{x_1}^{x_N} \phi_i(x)u_t dx = (\Delta x/6)c_{i+1}' + (4\Delta x/6)c_i' + (\Delta x/6)c_{i-1}' \quad (34)$$

where we have made use of the previous results of equations (25) to (30). The second integral involves u_{xx}

$$\begin{aligned} -D \int_{x_1}^{x_N} \phi_i(x)u_{xx} dx &= -D \left\{ \phi_i(x)u_x \Big|_{x_1}^{x_N} - \int_{x_1}^{x_N} \phi_i'(x)u_x dx \right\} \\ &= -D \left\{ u_x(x_N,t) - \int_{x_1}^{x_N} \phi_i'(x) \sum_{i=1}^N c_i(t)\phi_i'(x) dx \right\} \quad (\text{for } i = N) \\ &= -D \left\{ -u_x(x_1,t) - \int_{x_1}^{x_N} \phi_i'(x) \sum_{i=1}^N c_i(t)\phi_i'(x) dx \right\} \quad (\text{for } i = 1) \\ &= -D \left\{ - \int_{x_1}^{x_N} \phi_i'(x) \sum_{i=1}^N c_i(t)\phi_i'(x) dx \right\} \quad (\text{for } i \neq 1 \text{ or } N) \end{aligned}$$

If we consider first the last result (for $i \neq 1$ or N), three integrals must be considered (for the hat function of Figure 1.a)

$$\begin{aligned} c_i \int_{x_{i-1}}^{x_{i+1}} \phi_i'(x)\phi_i'(x) dx + c_{i-1} \int_{x_{i-1}}^{x_i} \phi_i'(x)\phi_{i-1}'(x) dx + c_{i+1} \int_{x_i}^{x_{i+1}} \phi_i'(x)\phi_{i+1}'(x) dx \\ = c_i \left(\int_{x_i}^{x_{i+1}} \phi_i'(x)\phi_i'(x) dx + \int_{x_{i-1}}^{x_i} \phi_i'(x)\phi_i'(x) dx \right) \end{aligned}$$

$$\begin{aligned}
& + c_{i-1} \int_{x_{i-1}}^{x_i} \phi_i'(x) \phi_{i-1}'(x) dx + c_{i+1} \int_{x_i}^{x_{i+1}} \phi_i'(x) \phi_{i+1}'(x) dx \\
& = c_i \left(\int_{x_i}^{x_{i+1}} (-1/\Delta x)^2 dx + \int_{x_{i-1}}^{x_i} (1/\Delta x)^2 dx \right) \\
& + c_{i-1} \int_{x_{i-1}}^{x_i} (1/\Delta x)(-1/\Delta x) dx + c_{i+1} \int_{x_i}^{x_{i+1}} (-1/\Delta x)(1/\Delta x) dx \\
& = c_i \left((-1/\Delta x)^2 x \Big|_{x_i}^{x_{i+1}} + (1/\Delta x)^2 x \Big|_{x_{i-1}}^{x_i} \right) \\
& - c_{i-1} (1/\Delta x)^2 x \Big|_{x_{i-1}}^{x_i} - c_{i+1} (1/\Delta x)^2 x \Big|_{x_i}^{x_{i+1}} \Big\} \\
& = c_i \left((1/\Delta x) + (1/\Delta x) \right) - c_{i-1} (1/\Delta x) - c_{i+1} (1/\Delta x)
\end{aligned}$$

Then, combining the two integrals (for u_t and $-Du_{xx}$), we have as the approximation of equation (1) at the interior grid points (for $i \neq 1$ or N)

$$\begin{aligned}
& (\Delta x/6)c_{i+1}' + (4\Delta x/6)c_i' + (\Delta x/6)c_{i-1}' \\
& = -D \left\{ c_i \left((1/\Delta x) + (1/\Delta x) \right) - c_{i-1} (1/\Delta x) - c_{i+1} (1/\Delta x) \right\} \\
\text{or} \\
& (1/6)c_{i+1}' + (4/6)c_i' + (1/6)c_{i-1}' = D \left\{ \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} \right\}
\end{aligned}$$

We now consider the preceding results at the boundaries. First, at the left boundary, $x = x_1$,

$$-D \left\{ -u_x(x_1, t) - \int_{x_1}^{x_N} \phi_1'(x) \sum_{i=1}^N c_i(t) \phi_i'(x) dx \right\} \quad (\text{for } i = 1)$$

$$= -D \left\{ -u_x(x_1, t) - \int_{x_1}^{x_2} \phi_1'(x) \sum_{i=1}^N c_i(t) \phi_i'(x) dx \right\}$$

The integration in x gives

$$\begin{aligned} & c_1 \int_{x_1}^{x_2} \phi_1'(x) \phi_1'(x) dx + c_2 \int_{x_1}^{x_2} \phi_1'(x) \phi_2'(x) dx \\ &= c_1 \int_{x_1}^{x_2} (-1/\Delta x)^2 dx + c_2 \int_{x_1}^{x_2} (-1/\Delta x)(1/\Delta x) dx \\ &= c_1 (-1/\Delta x)^2 x \Big|_{x_1}^{x_2} - c_2 (1/\Delta x)^2 x \Big|_{x_1}^{x_2} \\ &= c_1 (1/\Delta x) - c_2 (1/\Delta x) \end{aligned}$$

The approximation to equation (1) at $x = x_1$ is then

$$\begin{aligned} & (2\Delta x/6)c_1' + (\Delta x/6)c_2' \\ &= D \left\{ -h_1(t) - c_1(1/\Delta x) + c_2(1/\Delta x) \right\} \end{aligned}$$

or

$$(2/6)c_1' + (1/6)c_2' = -Dh_1(t)/\Delta x + D \left\{ \frac{c_2 - c_1}{\Delta x^2} \right\}$$

where we have used boundary condition (5).

Similarly, for the boundary at $x = x_N$, we have

$$-D \left\{ u_x(x_N, t) - \int_{x_1}^{x_N} \phi_1'(x) \sum_{i=1}^N c_i(t) \phi_i'(x) dx \right\} \quad (\text{for } i = N)$$

$$= -D \left\{ u_x(x_N, t) \cdot \int_{x_{N-1}}^{x_N} \phi_1'(x) \sum_{i=1}^N c_i(t) \phi_i'(x) dx \right\}$$

The integration in x gives

$$\begin{aligned} & c_N \int_{x_{N-1}}^{x_N} \phi_N'(x) \phi_N'(x) dx + c_{N-1} \int_{x_{N-1}}^{x_N} \phi_N'(x) \phi_{N-1}'(x) dx \\ &= c_N \int_{x_{N-1}}^{x_N} (1/\Delta x)^2 dx + c_{N-1} \int_{x_{N-1}}^{x_N} (1/\Delta x)(-1/\Delta x) dx \\ &= c_N (1/\Delta x)^2 x \Big|_{x_{N-1}}^{x_N} - c_{N-1} (1/\Delta x)^2 x \Big|_{x_{N-1}}^{x_N} \\ &= c_N (1/\Delta x) - c_{N-1} (1/\Delta x) \end{aligned}$$

The approximation to equation (1) at $x = x_N$ is then

$$\begin{aligned} & (2\Delta x/6)c_N' + (\Delta x/6)c_{N-1}' \\ &= D \left\{ h_N(t) + c_{N-1}(1/\Delta x) - c_N(1/\Delta x) \right\} \end{aligned}$$

or

$$(2/6)c_N' + (1/6)c_{N-1}' = Dh_N(t)/\Delta x + D \left\{ \frac{c_{N-1} - c_N}{\Delta x^2} \right\}$$

where we have made use of boundary condition (6).

The complete set of ODEs for Neumann boundary conditions (5) and (6) is therefore

$$(2/6)c_1' + (1/6)c_2' = -Dh_1(t)/\Delta x + D \left\{ \frac{c_2 - c_1}{\Delta x^2} \right\}$$

$$(1/6)c_1' + (4/6)c_2' + (1/6)c_3' = D \left\{ \frac{c_3 - 2c_2 + c_1}{\Delta x^2} \right\}$$

$$(1/6)c_2' + (4/6)c_3' + (1/6)c_4' = D \left\{ \frac{c_4 - 2c_3 + c_2}{\Delta x^2} \right\}$$

.

(35)

$$(1/6)c_{N-2}' + (4/6)c_{N-1}' + (1/6)c_N' = D \left\{ \frac{c_N - 2c_{N-1} + c_{N-2}}{\Delta x^2} \right\}$$

$$(1/6)c_{N-1}' + (2/6)c_N' = Dh_N(t)/\Delta x + D \left\{ \frac{c_{N-1} - c_N}{\Delta x^2} \right\}$$

This is a system of N ODEs which can be integrated to obtain $c_1(t), c_2(t), \dots, c_{N-1}(t), c_N(t)$ subject to the initial conditions

$$c_1(0) = g(x_1), c_2(0) = g(x_2), \dots, c_{N-1}(0) = g(x_{N-1}), c_N(0) = g(x_N) \quad (36)$$

Subroutines INIT2, DERV2 and PRINT2 for the integration of equations (35) and (36) are listed in Program 6

SUBROUTINE INIT2

```

C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY LINEAR FINITE ELEMENTS (HAT
C... FUNCTIONS)
C...
PARAMETER (N=51)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)
4 /I/ IP
5 /FE/ AL(N), BM(N), CU(N), BRHS(N)
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... LENGTH
L=1.0
C...
C... GRID SPACING
DX=L/FLOAT(N-1)
C...
C... SET UP THE COEFFICIENT MATRIX IN BAND STORAGE MODE. THIS IS
C... DONE ONLY ONCE SINCE THE COEFFICIENT MATRIX IS CONSTANT

```

```

C...
C... LOWER DIAGONAL
      DO 2 I=1,N
        IF(I.EQ.1)THEN
          AL(1)=0.
        ELSE
+       IF(I.EQ.N)THEN
          AL(N)=1.0/6.0
        ELSE
          AL(I)=1.0/6.0
        END IF
2      CONTINUE
C...
C... MAIN DIAGONAL
      DO 3 I=1,N
        IF(I.EQ.1)THEN
          BM(1)=2.0/6.0
        ELSE
+       IF(I.EQ.N)THEN
          BM(N)=2.0/6.0
        ELSE
          BM(I)=4.0/6.0
        END IF
3      CONTINUE
C...
C... UPPER DIAGONAL
      DO 4 I=1,N
        IF(I.EQ.1)THEN
          CU(1)=1.0/6.0
        ELSE
+       IF(I.EQ.N)THEN
          CU(N)=0.
        ELSE
          CU(I)=1.0/6.0
        END IF
4      CONTINUE
C...
C... INITIAL CONDITION (2)
      DO 1 I=1,N
        X(I)=FLOAT(I-1)*DX
        IF(I.LT.(N+1)/2)U(I)=0.
        IF(I.GT.(N+1)/2)U(I)=1.
        IF(I.EQ.(N+1)/2)U(I)=0.5
1      CONTINUE
C...
C... INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/
      CALL DERV
      IP=0
      RETURN
      END

```

```

SUBROUTINE DERV2
C...
C... SUBROUTINE DERV2 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C... ODES IN THE FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
C...
PARAMETER (N=51)
COMMON/T/      T,  NSTOP,  NORUN

1      /Y/      U(N)
2      /F/      UT(N)
3      /XG/     L,      DX,   X(N)
4      /I/      IP
5      /FE/     AL(N),  BM(N), CU(N), BRHS(N)
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... BOUNDARY CONDITION (3)
H1=0.
C...
C... BOUNDARY CONDITION (4)
HN=0.
C...
C... RIGHT HAND SIDE VECTOR (WITH D = 1)
DO 1 I=1,N
  IF(I.EQ.1)THEN
    BRHS(I)=-H1/DX+(U(I+1)-U(I))/DX**2
  ELSE
+   IF(I.EQ.N)THEN
    BRHS(I)= HN/DX+(U(I-1)-U(I))/DX**2
  ELSE
    BRHS(I)=(U(I+1)-2.0*U(I)+U(I-1))/DX**2
  END IF
1 CONTINUE
C...
C... SOLVE THE LINEAR ALGEBRAIC EQUATIONS BY SUBROUTINE TRIDAG, WHICH
C... RETURNS THE DERIVATIVE VECTOR UT IN COMMON/F/
CALL TRIDAG(AL,BM,CU,BRHS,UT,N)
RETURN
END

SUBROUTINE PRINT2(NI,NO)
C...
C... SUBROUTINE PRINT2 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
C...
PARAMETER (N=51)
COMMON/T/      T,  NSTOP,  NORUN
1      /Y/      U(N)
2      /F/      UT(N)
3      /XG/     L,      DX,   X(N)

```

```

4      /I/      IP
5      /FE/    AL(N), BM(N), CU(N), BRHS(N)
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L, UE(N), DIFF(N)
C...
C... PRINT THE COEFFICIENT MATRIX FOR VERIFICATION
IP=IP+1
IF(IP.EQ.1)THEN
WRITE(NO,13)
13     FORMAT(/,' COEFFICIENT MATRIX',/)
DO 14 I=1,N
WRITE(NO,15)I,AL(I),BM(I),CU(I)
15     FORMAT(I5,3F12.4)
14     CONTINUE
WRITE(NO,16)
16     FORMAT(//)
END IF
C...
C... COMPUTE THE TOTAL ENERGY (RHO = CP = 1)
CALL DERV2
ENERGY=0.
DO 1 I=2,N
ENERGY=ENERGY+0.5*(U(I)+U(I-1))*DX
1     CONTINUE
C...
C... CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C... NUMERICAL AND EXACT SOLUTIONS
DO 3 I=1,N,10
UE(I)=SERIES(I)
DIFF(I)=U(I)-UE(I)
3     CONTINUE
C...
C... PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
WRITE(NO,2)T,ENERGY,(U(I),I=1,N,10),
1      (UE(I),I=1,N,10),
2      (DIFF(I),I=1,N,10)
2     FORMAT(' T = ',F6.2,' ENERGY = ',F9.5,
1      /,14X,' X=0',5X,'X=0.2',5X,'X=0.4',
2      5X,'X=0.6',5X,'X=0.8',5X,'X=1.0'/,
3      ' U(X,T)',6F10.6,/,
4      ' UE(X,T)',6F10.6,/,
5      ' DIFF(X,T)',6F10.6,/)
RETURN
END

```

Program 6: Solution of Case 2 Test Problem by Finite Elements

The only essential difference between Programs 5 and 6 is that in the latter, initial condition (10)

(rather than (7)) is implemented in INIT2, and boundary conditions (5) and (6) (or (11) and (12) rather than (3) and (4) (or (8) and (9)) are implemented in DERV2. Also, the total energy is again calculated in PRINT2 (as was done in Program 4).

The output from Program 6 is listed in Table 4 (the weighting coefficients of equations (35) are not printed since they are the same as in Table 3)

```

RUN NO. - 3 FINITE ELEMENT SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
FINAL T - 0.100E+01
PRINT T - 0.100E+00
NUMBER OF DIFFERENTIAL EQUATIONS - 51
INTEGRATION ALGORITHM - LSODES
MAXIMUM INTEGRATION ERROR - 0.100E-06

COEFFICIENT MATRIX
(deleted)

T = 0.00 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
UE(X,T) 0.000318 0.000393 0.001030 0.998971 0.999607 0.999682
DIFF(X,T) -0.000318 -0.000393 -0.001030 0.001029 0.000393 0.000318

T = 0.10 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.262911 0.308158 0.426703 0.573297 0.691842 0.737090
UE(X,T) 0.262756 0.308033 0.426655 0.573345 0.691967 0.737244
DIFF(X,T) 0.000154 0.000125 0.000048 -0.000048 -0.000125 -0.000154

T = 0.20 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.411652 0.428525 0.472699 0.527301 0.571475 0.588348
UE(X,T) 0.411566 0.428456 0.472673 0.527327 0.571544 0.588434
DIFF(X,T) 0.000085 0.000069 0.000027 -0.000026 -0.000069 -0.000085

T = 0.30 ENERGY = 0.50000
      X=0      X=0.2      X=0.4      X=0.6      X=0.8      X=1.0
U(X,T) 0.467082 0.473369 0.489828 0.510172 0.526631 0.532918
UE(X,T) 0.467040 0.473335 0.489815 0.510185 0.526665 0.532960

```

```

DIFF(X,T)  0.000042  0.000034  0.000013 -0.000013 -0.000034 -0.000042
    T =    0.40  ENERGY =    0.50000
           X=0    X=0.2    X=0.4    X=0.6    X=0.8    X=1.0
    U(X,T)  0.487735  0.490077  0.496210  0.503790  0.509923  0.512266
    UE(X,T)  0.487716  0.490062  0.496204  0.503796  0.509938  0.512284
DIFF(X,T)  0.000019  0.000016  0.000006 -0.000006 -0.000015 -0.000019
    T =    0.50  ENERGY =    0.50000
           X=0    X=0.2    X=0.4    X=0.6    X=0.8    X=1.0
    U(X,T)  0.495430  0.496303  0.498588  0.501413  0.503698  0.504570
    UE(X,T)  0.495421  0.496296  0.498585  0.501415  0.503704  0.504578
DIFF(X,T)  0.000009  0.000007  0.000003 -0.000002 -0.000006 -0.000008
    .
    .
    .
    T =    1.00  ENERGY =    0.50000
           X=0    X=0.2    X=0.4    X=0.6    X=0.8    X=1.0
    U(X,T)  0.499967  0.499974  0.499990  0.500010  0.500027  0.500033
    UE(X,T)  0.499967  0.499973  0.499990  0.500010  0.500027  0.500033
DIFF(X,T)  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000

```

COMPUTATIONAL STATISTICS

```

LAST STEP SIZE                0.150E-01
LAST ORDER OF THE METHOD      3
TOTAL NUMBER OF STEPS TAKEN  347
NUMBER OF FUNCTION EVALUATIONS 644
NUMBER OF JACOBIAN EVALUATIONS 7

```

Table 4: Output from Programs 1 and 6

Again, as in Table 2, the energy is conserved to five figures. We now proceed to the third method of approximation, finite volumes.

(5) Finite Volumes

The essential idea of the method of finite volumes is the application of one or more conservation principles to a *control volume* [Patankar (3), Lick (4)]. This idea is illustrated in Figure 2 where a spatial grid in x is represented in terms of a series of control volumes which are arranged for

Dirichlet boundary conditions with temperatures u_0 and u_L specified (u represents temperature in the subsequent analysis), or Neumann boundary conditions with fluxes q_0 and q_N specified (which in turn define the gradients or first order spatial derivatives at $x = 0$ and $x = 1$)

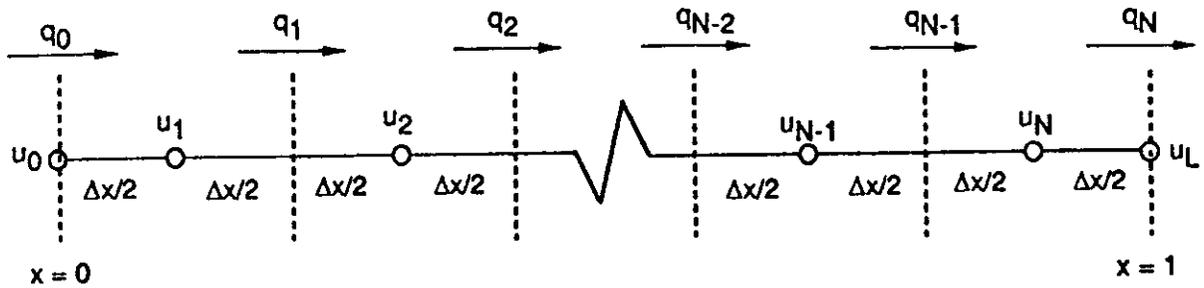


Fig. 2: Finite Volume Spatial Grid

An energy balance written for the volume with temperature u_2 gives

$$V_2 \rho_2 C_{p2} \frac{du_2}{dt} = A_1 q_1 - A_2 q_2 \quad (37)$$

where

- u_2 temperature of control volume 2
- t time
- V_2 volume of control volume 2
- ρ_2 density of material in control volume 2
- C_{p2} specific heat of material in control volume 2
- q_1 heat flux due to temperature difference $u_1 - u_2$
- A_1 area across which q_1 flows
- q_2 heat flux due to temperature difference $u_2 - u_3$
- A_2 area across which q_2 flows

If we take fluxes q_1 and q_2 as

$$q_1 = k_{1,2} \left(\frac{u_1 - u_2}{\Delta x_1/2 + \Delta x_2/2} \right), \quad q_2 = k_{2,3} \left(\frac{u_2 - u_3}{\Delta x_2/2 + \Delta x_3/2} \right) \quad (38)(39)$$

where

$k_{1,2}$ thermal conductivity as a function of u_1 and u_2 (to be defined)

$k_{2,3}$ thermal conductivity as a function of u_2 and u_3 (to be defined)

Δx_1 lengths indicated in Figure

Δx_2

Δx_3

then substitution of equations (38) and (39) in equation (37) gives an ODE for the temperature u_2 . However, in a computer code, it is often more convenient to compute the fluxes separately, e.g., from equations (38) and (39), then use them in the energy balance, e.g., equation (37).

We can now consider an interface temperature, u_i , at the left face of volume V_2 . Then, if we assume continuity in the heat flux q_1 across the left face

$$q_1 = k_{1,2} \left(\frac{u_1 - u_2}{\Delta x_1/2 + \Delta x_2/2} \right) = k_1 \left(\frac{u_1 - u_i}{\Delta x_1/2} \right) = k_2 \left(\frac{u_i - u_2}{\Delta x_2/2} \right) \quad (40)$$

u_i obtained from the last of equations (40) is

$$\left\{ \left(\frac{k_1}{\Delta x_1/2} \right) + \left(\frac{k_2}{\Delta x_2/2} \right) \right\} u_i = \left(\frac{k_1}{\Delta x_1/2} \right) u_1 + \left(\frac{k_2}{\Delta x_2/2} \right) u_2$$

or

$$u_i = \frac{\left(\frac{k_1}{\Delta x_1/2} \right) u_1 + \left(\frac{k_2}{\Delta x_2/2} \right) u_2}{\left(\frac{k_1}{\Delta x_1/2} \right) + \left(\frac{k_2}{\Delta x_2/2} \right)} \quad (41)$$

Then, we can form the temperature difference $u_i - u_2$ (after cancellation of the common factor of 2 in equation (41))

$$u_i - u_2 = \frac{\left(\frac{k_1}{\Delta x_1} \right) u_1 + \left(\frac{k_2}{\Delta x_2} \right) u_2}{\left(\frac{k_1}{\Delta x_1} \right) + \left(\frac{k_2}{\Delta x_2} \right)} - u_2$$

or

$$u_i - u_2 = \frac{\left(\frac{k_1}{\Delta x_1}\right)u_1 + \left(\frac{k_2}{\Delta x_2}\right)u_2 - u_2 \left\{ \left(\frac{k_1}{\Delta x_1}\right) + \left(\frac{k_2}{\Delta x_2}\right) \right\}}{\left(\frac{k_1}{\Delta x_1}\right) + \left(\frac{k_2}{\Delta x_2}\right)}$$

$$= \frac{\left(\frac{k_1}{\Delta x_1}\right)(u_1 - u_2)}{\left(\frac{k_1}{\Delta x_1}\right) + \left(\frac{k_2}{\Delta x_2}\right)}$$

Then, from equation (40),

$$q_1 = k_{1,2} \left(\frac{u_1 - u_2}{\Delta x_1/2 + \Delta x_2/2} \right) = k_2 \left(\frac{u_i - u_2}{\Delta x_2/2} \right)$$

$$= \frac{2k_1 k_2 / (\Delta x_1 \Delta x_2)}{\left(\frac{k_1}{\Delta x_1}\right) + \left(\frac{k_2}{\Delta x_2}\right)} (u_1 - u_2) = \frac{2}{\left(\frac{\Delta x_1}{k_1}\right) + \left(\frac{\Delta x_2}{k_2}\right)} (u_1 - u_2)$$

or

$$k_{1,2} = \frac{(\Delta x_1 + \Delta x_2)}{\left(\frac{\Delta x_1}{k_1}\right) + \left(\frac{\Delta x_2}{k_2}\right)} \quad (42)$$

which is the result reported by Patankar [3, eq. (4.9)]. For the special case $\Delta x_1 = \Delta x_2 = \Delta x$, equation (42) reduces to

$$k_{1,2} = \frac{2k_1 k_2}{k_1 + k_2} \quad (43)$$

so that $k_{1,2}$ is the *harmonic mean* of k_1 and k_2 .

We can now generalize equations (37), (38), (39) and (42) to

$$V_i \rho_i C_{pi} \frac{du_i}{dt} = A_{i-1} q_{i-1} - A_i q_i \quad (44)$$

$$q_{i-1} = k_{i-1,i} \left(\frac{u_{i-1} - u_i}{\Delta x_{i-1}/2 + \Delta x_i/2} \right), \quad q_i = k_{i,i+1} \left(\frac{u_i - u_{i+1}}{\Delta x_i/2 + \Delta x_{i+1}/2} \right) \quad (45)(46)$$

$$k_{i-1,i} = \frac{(\Delta x_{i-1} + \Delta x_i)}{\left(\frac{\Delta x_{i-1}}{k_{i-1}}\right) + \left(\frac{\Delta x_i}{k_i}\right)} \quad (47)$$

where "i" is now a grid index and does not denote an interface. By writing equation (44) in terms of the grid index i, it is possible to vary the properties as a function of u_i , i.e., nonlinear characteristics can be included by programming ρ_i , C_{pi} , $k_{i-1,i}$ and $k_{i,i+1}$ as a function of u_i . Also, the geometric parameters can be varied with i, i.e., Δx_i , V_i , A_{i-1} and A_i as a function of i, which could be used to implement equation (44) in various coordinate systems, e.g., cylindrical or spherical coordinates.

(5.1) Dirichlet Boundary Conditions

Subroutines INIT3, DERV3 and PRINT3 for the application of equations (44) to (47) to equations (1), (7), (8) and (9) are listed in Program 7

```

SUBROUTINE INIT3
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE VOLUMES
C...
PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N), PI
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... PI
PI=4.0*ATAN(1.0)
C...
C... LENGTH
L=1.0
C...
C... GRID SPACING
DX=L/FLOAT(N)
C...
C... INITIAL CONDITION (2)
DO 1 I=1,N
X(I)=FLOAT(I-1)*DX+0.5*DX
U(I)=SIN(PI*X(I)/L)
1 CONTINUE
C...
C... INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/

```

```

CALL DERV
IP=0
RETURN
END

```

```

SUBROUTINE DERV3

```

```

C...
C... SUBROUTINE DERV3 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C... ODES IN THE FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
C...

```

```

PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N), PI
4 /I/ IP

```

```

C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L

```

```

C...
C... DEFINE ARRAY FOR THE FLUXES
REAL Q(0:N)

```

```

C...
C... BOUNDARY CONDITION (3)
UO=0.

```

```

C...
C... BOUNDARY CONDITION (B4)
UL=0.

```

```

C...
C... COMPUTE THE FLUXES (K/(RHO*CP) = 1)
Q(0)=(UO-U(1))/(0.5*DX)
DO 2 I=1,N-1
    Q(I)=(U(I)-U(I+1))/DX
2 CONTINUE
Q(N)=(U(N)-UL)/(0.5*DX)

```

```

C...
C... EQUATION (1)
DO 1 I=1,N
    UT(I)=(Q(I-1)-Q(I))/DX
1 CONTINUE
RETURN
END

```

```

SUBROUTINE PRINT3(NI,NO)

```

```

C...
C... SUBROUTINE PRINT3 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
C...

```

```

PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)

```

```

2      /F/  UT(N)
3      /XG/ L,    DX,  X(N),  PI
4      /I/  IP
C...
C...  TYPE SELECTED VARIABLES AS REAL
      REAL L, UE(N), DIFF(N)
C...
C...  CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C...  NUMERICAL AND EXACT SOLUTIONS
      DO 1 I=1,21,5
          UE(I)=EXACT(I)
          DIFF(I)=U(I)-UE(I)
1      CONTINUE
C...
C...  PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
      UAVG=(U(N/2)+U(N/2+1))/2.0
      WRITE(NO,2)T,(U(I),I=1,21,5),UAVG,
1          (UE(I),I=1,21,5),
2          (DIFF(I),I=1,21,5)
2      FORMAT('  T = ',F6.2,
1          /,14X,'X=0.01',4X,'X=0.11',4X,'X=0.21',
2          4X,'X=0.31',4X,'X=0.41',4X,'X=0.50'/,
3          '    U(X,T)',6F10.6,/,
4          '    UE(X,T)',5F10.6,/,
5          '    DIFF(X,T)',5F10.6,/)
      RETURN
      END

```

Program 7: Solution of Case 1 Test Problem by Finite Volumes

The programming is essentially self explanatory. Note in particular the programming of the fluxes in DO loop 2 and the finite volume (energy balance) equation in DO loop 1 of subroutine DERV3, in accordance with equations (44) to (47).

The output from Program 7 is listed in Table 5

```

RUN NO. - 4 FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
FINAL T - 0.500E+00
PRINT T - 0.100E+00
NUMBER OF DIFFERENTIAL EQUATIONS - 50
INTEGRATION ALGORITHM - LSODES

```

MAXIMUM INTEGRATION ERROR - 0.100E-06

T =	0.00					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.031411	0.338738	0.612907	0.827081	0.960294	0.999507
UE(X,T)	0.031411	0.338738	0.612907	0.827081	0.960294	
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	

T =	0.10					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.011711	0.126292	0.228510	0.308360	0.358026	0.372646
UE(X,T)	0.011707	0.126250	0.228435	0.308259	0.357909	
DIFF(X,T)	0.000004	0.000041	0.000075	0.000101	0.000117	

T =	0.20					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.004366	0.047085	0.085195	0.114965	0.133482	0.138933
UE(X,T)	0.004363	0.047054	0.085140	0.114891	0.133395	
DIFF(X,T)	0.000003	0.000031	0.000055	0.000075	0.000087	

T =	0.30					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.001628	0.017555	0.031763	0.042862	0.049766	0.051798
UE(X,T)	0.001626	0.017538	0.031732	0.042821	0.049718	
DIFF(X,T)	0.000002	0.000017	0.000031	0.000042	0.000048	

T =	0.40					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.000607	0.006545	0.011842	0.015980	0.018554	0.019312
UE(X,T)	0.000606	0.006536	0.011827	0.015960	0.018530	
DIFF(X,T)	0.000001	0.000008	0.000015	0.000021	0.000024	

T =	0.50					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.000226	0.002440	0.004415	0.005958	0.006917	0.007200
UE(X,T)	0.000226	0.002436	0.004408	0.005948	0.006906	
DIFF(X,T)	0.000000	0.000004	0.000007	0.000009	0.000011	

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.112E-01
LAST ORDER OF THE METHOD	4
TOTAL NUMBER OF STEPS TAKEN	110
NUMBER OF FUNCTION EVALUATIONS	158
NUMBER OF JACOBIAN EVALUATIONS	2

Table 5: Output from Programs 1 and 7

The comparison with the preceding solutions (Tables 1 and 3) is complicated somewhat by the fact that the temperatures in the grid of Figure 2 are not at the same values of x as for the finite difference and finite elements solutions. For example, $u(\Delta x/2,t)$ is the first (left-most) temperature ($U(1)$ in Program 7) and $u(1-\Delta x/2,t)$ is the last temperature ($U(N)$ in Program 7). Thus, the surface temperatures are not available for comparison with the preceding solutions.

However, we can make a comparison, keeping in mind the "offset" of $\Delta x/2$. From Tables 1, 3 and 5, we have

Three point finite differences (from Table 1)

T =	0.10					
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.219071	0.354465	0.354465	0.219071	0.000000
UE(X,T)	0.000000	0.219072	0.354466	0.354466	0.219072	0.000000
DIFF(X,T)	0.000000	-0.000001	-0.000001	-0.000001	-0.000001	0.000000

Finite elements (from Table 3)

T =	0.10					
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1
U(X,T)	0.000000	0.219002	0.354352	0.354352	0.219002	0.000000
UE(X,T)	0.000000	0.219072	0.354466	0.354466	0.219072	0.000000
DIFF(X,T)	0.000000	-0.000070	-0.000114	-0.000114	-0.000071	0.000000

Finite volumes (from Table 5)

T =	0.10					
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.011711	0.126292	0.228510	0.308360	0.358026	0.372646
UE(X,T)	0.011707	0.126250	0.228435	0.308259	0.357909	
DIFF(X,T)	0.000004	0.000041	0.000075	0.000101	0.000117	

We can now make a linear interpolation of the solution from Table 5 over the interval $\Delta x/2$ for comparison with the solutions from Table 1 and 3. For example, extrapolation from $x = 0.01$ to $x = 0$ gives

$$u(0,0.1) \approx u(0.01,0.1) - \left\{ \left(u(0.11,0.1) - u(0.01,0.1) \right) / (0.11 - 0.01) \right\} 0.01$$

$$= 0.011711 - (0.126292 - 0.011711)0.1 = 0.011711 - 0.011458 = 0.000253$$

and the exact value is $u(0,0.1) = 0$ (from boundary condition (8)). Similarly, an interpolation from $x = 0.21$ to $x = 0.2$ gives

$$u(0.2,0.1) \approx u(0.21,0.1) - \left\{ \left(u(0.21,0.1) - u(0.11,0.1) \right) / (0.21 - 0.11) \right\} 0.01$$

$$= 0.228510 - (0.228510 - 0.126292)0.1 = 0.228510 - 0.102218 = 0.218882$$

and the exact value is $u(0.2,0.1) = 0.219072$.

(5.2) Neumann Boundary Conditions

Subroutines INIT3, DERV3 and PRINT3 for the finite volume solution of equations (1), (10), (11) and (12) is listed in Program 8

```

SUBROUTINE INIT3
C...
C... SOLUTION OF EQUATIONS (1) TO (4) BY FINITE VOLUMES
C...
PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)
4 /I/ IP
C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... LENGTH
L=1.0
C...
C... GRID SPACING
DX=L/FLOAT(N)
C...
C... INITIAL CONDITION (2)
DO 1 I=1,N
X(I)=FLOAT(I-1)*DX+0.5*DX
IF(I.LE.(N/2))U(I)=0.
IF(I.GT.(N/2))U(I)=1.0
1 CONTINUE
C...
C... INITIALIZE THE TEMPORAL DERIVATIVES IN COMMON/F/

```

```

CALL DERV
IP=0
RETURN
END

```

```

SUBROUTINE DERV3

```

```

C...
C... SUBROUTINE DERV3 IS CALLED BY THE ODE INTEGRATOR TO DEFINE THE
C... ODES IN THE FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
C...

```

```

PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)
4 /I/ IP

```

```

C...
C... TYPE SELECTED VARIABLES AS REAL
REAL L
C...
C... DEFINE ARRAY FOR THE FLUXES
REAL Q(0:N)
C...
C... BOUNDARY CONDITION (3)
Q(0)=0.
C...
C... BOUNDARY CONDITION (4)
Q(N)=0.
C...
C... COMPUTE THE FLUXES (K/(RHO*CP) = 1)
DO 2 I=1,N-1
    Q(I)=(U(I)-U(I+1))/DX
2 CONTINUE
C...
C... EQUATION (1)
DO 1 I=1,N
    UT(I)=(Q(I-1)-Q(I))/DX
1 CONTINUE
RETURN
END

```

```

SUBROUTINE PRINT3(NI,NO)

```

```

C...
C... SUBROUTINE PRINT3 IS CALLED BY THE MAIN PROGRAM TO PRINT THE
C... FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
C...

```

```

PARAMETER (N=50)
COMMON/T/ T, NSTOP, NORUN
1 /Y/ U(N)
2 /F/ UT(N)
3 /XG/ L, DX, X(N)

```

```

4      /I/      IP
C...
C...  TYPE SELECTED VARIABLES AS REAL
      REAL L, UE(N), DIFF(N)
C...
C...  COMPUTE THE TOTAL ENERGY (RHO = CP = 1)
      CALL DERV3
      ENERGY=0.
      DO 1 I=1,N
        ENERGY=ENERGY+U(I)*DX
1     CONTINUE
C...
C...  CALCULATE THE EXACT SOLUTION, AND THE DIFFERENCE BETWEEN THE
C...  NUMERICAL AND EXACT SOLUTIONS
      DO 3 I=1,21,5
        UE(I)=SERIES(I)
        DIFF(I)=U(I)-UE(I)
3     CONTINUE
C...
C...  PRINT THE NUMERICAL AND EXACT SOLUTIONS TO EQUATIONS (1) TO (4)
      UAVG=(U(N/2)+U(N/2+1))/2.0
      WRITE(NO,2)T,ENERGY,(U(I),I=1,21,5),UAVG,
1         (UE(I),I=1,21,5),
2         (DIFF(I),I=1,21,5)
2     FORMAT('      T = ',F6.2,'      ENERGY = ',F9.5,
1         /,14X,'X=0.01',4X,'X=0.11',4X,'X=0.21',
2         4X,'X=0.31',4X,'X=0.41',4X,'X=0.50' /,
3         '      U(X,T)',6F10.6,/,
4         '      UE(X,T)',5F10.6,/,
5         '      DIFF(X,T)',5F10.6,/)
      RETURN
      END

```

Program 8: Solution of Case 2 Test Problem by Finite Volumes

Program 8 follows closely the details of Program 7. The only essential difference is the use of boundary conditions (11) and (12) rather than (8) and (9) in subroutine DERV3.

The output from Program 8 is listed in Table 6

```

RUN NO. - 4 FINITE VOLUME SOLUTION OF EQUATIONS (1) TO (4)
INITIAL T - 0.000E+00
FINAL T - 0.100E+01
PRINT T - 0.100E+00

```

NUMBER OF DIFFERENTIAL EQUATIONS - 50

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-06

T =	0.00	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000
UE(X,T)	0.000319	0.000338	0.000403	0.000566	0.001141	
DIFF(X,T)	-0.000319	-0.000338	-0.000403	-0.000566	-0.001141	

T =	0.10	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.262758	0.276660	0.312414	0.366538	0.433748	0.500000
UE(X,T)	0.262873	0.276769	0.312506	0.366604	0.433781	
DIFF(X,T)	-0.000115	-0.000109	-0.000092	-0.000066	-0.000033	

T =	0.20	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.411538	0.416726	0.430067	0.450252	0.475308	0.500000
UE(X,T)	0.411610	0.416795	0.430124	0.450293	0.475328	
DIFF(X,T)	-0.000072	-0.000068	-0.000057	-0.000041	-0.000020	

T =	0.30	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.467018	0.468952	0.473926	0.481452	0.490794	0.500000
UE(X,T)	0.467056	0.468989	0.473957	0.481474	0.490804	
DIFF(X,T)	-0.000039	-0.000036	-0.000030	-0.000022	-0.000011	

T =	0.40	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.487703	0.488424	0.490279	0.493085	0.496568	0.500000
UE(X,T)	0.487722	0.488442	0.490293	0.493095	0.496573	
DIFF(X,T)	-0.000019	-0.000018	-0.000015	-0.000011	-0.000005	

T =	0.50	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.495415	0.495684	0.496375	0.497422	0.498720	0.500000
UE(X,T)	0.495424	0.495692	0.496382	0.497427	0.498723	
DIFF(X,T)	-0.000009	-0.000008	-0.000007	-0.000005	-0.000002	

.
.
.

T =	1.00	ENERGY = 0.50000				
	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.499967	0.499969	0.499974	0.499981	0.499991	0.500000
UE(X,T)	0.499967	0.499969	0.499974	0.499981	0.499991	
DIFF(X,T)	0.000000	0.000000	0.000000	0.000000	0.000000	

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.227E-01
LAST ORDER OF THE METHOD	3
TOTAL NUMBER OF STEPS TAKEN	315
NUMBER OF FUNCTION EVALUATIONS	419
NUMBER OF JACOBIAN EVALUATIONS	6

Table 6: Output from Programs 1 and 8

The numerical solutions at $t = 0.1$ for the three methods of approximation are

Three point finite differences (from Table 2)

T =	0.10	ENERGY =	0.50000				
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0	
U(X,T)	0.262758	0.308032	0.426654	0.573347	0.691968	0.737242	
UE(X,T)	0.262756	0.308033	0.426655	0.573345	0.691967	0.737244	
DIFF(X,T)	0.000001	-0.000001	-0.000001	0.000001	0.000001	-0.000002	

Five point finite differences (from Table 2)

T =	0.10	ENERGY =	0.50000				
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0	
U(X,T)	0.262834	0.308095	0.426678	0.573321	0.691904	0.737167	
UE(X,T)	0.262756	0.308033	0.426655	0.573345	0.691967	0.737244	
DIFF(X,T)	0.000078	0.000062	0.000023	-0.000024	-0.000063	-0.000077	

Finite elements (from Table 4)

T =	0.10	ENERGY =	0.50000				
	X=0	X=0.2	X=0.4	X=0.6	X=0.8	X=1.0	
U(X,T)	0.262911	0.308158	0.426703	0.573297	0.691842	0.737090	
UE(X,T)	0.262756	0.308033	0.426655	0.573345	0.691967	0.737244	
DIFF(X,T)	0.000154	0.000125	0.000048	-0.000048	-0.000125	-0.000154	

Finite volumes (from Table 6)

T =	0.10	ENERGY =	0.50000
-----	------	----------	---------

	X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50
U(X,T)	0.262758	0.276660	0.312414	0.366538	0.433748	0.500000
UE(X,T)	0.262873	0.276769	0.312506	0.366604	0.433781	
DIFF(X,T)	-0.000115	-0.000109	-0.000092	-0.000066	-0.000033	

The three point finite difference solution, $u(0,0.1) = 0.262758$, the five point finite difference solution, $u(0,0.1) = 0.262834$, and the finite element solution $u(0,0.1) = 0.262911$ agree with the finite volume solution $u(0.01,0.1) = 0.262758$ to better than three figures because of the Neumann boundary condition $u_x(0,0.1) = 0$ (equation (11)), i.e., this zero slope condition means that the solution changes very little in the neighborhood of $x = 0$.

Thus, we conclude that for the Case 1 and Case 2 test problems of section (2), the three methods give essentially the same numerical solutions, and the choice of a solution, at least on the basis of these test problems, is largely determined by the convenience of implementation. We now conclude this discussion of approximations for parabolic PDEs with a test problem having boundary conditions of the third type.

(6) Boundary Conditions of the Third Type and Conductances

When two different materials come into contact in heat conduction and diffusion problems modeled by parabolic PDEs, the usual procedure for modeling the common boundary between the materials is to use a conductance (also termed a contact resistance, heat transfer coefficient or mass transfer coefficient). This procedure leads to a boundary condition of the third type, which is a combination of a Dirichlet boundary condition (since the PDE dependent variable appears in the boundary condition) and a Neumann boundary condition (since the first order spatial derivative of the dependent variable appears in the boundary condition). We now illustrate how boundary conditions of the third type can be implemented in a finite difference or a finite volume solution. We do not include a finite element solution because of the relative complexity of this method, as illustrated by the two preceding test problems (and by Programs 5 and 6).

The problem to be considered is again equation (1) with initial condition (2) (or, specifically, equation (10)) and the boundary condition

$$k \frac{\partial u(0.5,t)}{\partial x} = C \{ u(0.5+,t) - u(0.5-,t) \} \quad (48)$$

Equation (48) states that the flux at $x = 0.5$, $k \frac{\partial u(0.5,t)}{\partial x}$, is proportional to the difference in temperatures just to the right and left of $x = 0.5$, $C \{ u(0.5+,t) - u(0.5-,t) \}$. The proportionality

constant, C , is the *conductance*. We now consider two approaches to the solution of equation (1) with this boundary condition.

(6.1) Finite Differences

In using finite differences, we consider two domains, one to the left of $x = 0.5$ with temperature $u_1(x,t)$, and the other to the right of $x = 0.5$ with temperature $u_2(x,t)$. The complete test problem is then

$$u_{1t} = u_{1xx}, \quad u_{2t} = u_{2xx} \quad (49)(50)$$

$$u_1(x,0) = 0, \quad u_2(x,0) = 1 \quad (51)(52)$$

$$u_{1x}(0,t) = 0, \quad u_{2x}(1,t) = 0, \quad (53)(54)$$

$$k_1 u_{1x}(0.5,t) = C_{12} \{u_2(0.5,t) - u_1(0.5,t)\}, \quad k_2 u_{2x}(0.5,t) = C_{12} \{u_2(0.5,t) - u_1(0.5,t)\} \quad (55)(56)$$

where k_1 and k_2 are the thermal conductivities for the sections to the left and to the right of $x = 0.5$, respectively, and C_{12} is the conductance at $x = 0.5$.

Equations (49) to (56) are implemented in Program 9

```

SUBROUTINE INITAL
PARAMETER(NX=26)
COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U1(NX),      U2(NX)
2      /F/      U1T(NX),      U2T(NX)
3      /S/      U1X(NX),      U1XX(NX),      U2X(NX),      U2XX(NX),
4      /R/      X1(NX),      X2(NX),      XL,      DX,
5      K1,      K2,      C12
6      /I/      IP
REAL          K1,      K2
C...
C... HALF LENGTH
        XL=0.5
C...
C... GRID INCREMENT
        DX=XL/FLOAT(NX-1)
C...
C... GRID POINTS
        DO 1 I=1,NX
          X1(I)= FLOAT(I-1)*DX
          X2(I)=XL+FLOAT(I-1)*DX
1      CONTINUE

```

```

C...
C... CONDUCTIVITIES
      K1=1.0
      K2=1.0
C...
C... CONDUCTANCE
      C12=0.1
C...
C... INITIAL CONDITION
      U10=0.
      U20=1.0
      DO 2 I=1,NX
          U1(I)=U10
          U2(I)=U20
2      CONTINUE
C...
C... INITIALIZE COUNTER FOR PRINTED AND PLOTTED SOLUTION
      IP=0
      RETURN
      END

      SUBROUTINE DERV
      PARAMETER(NX=26)
      COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U1(NX),      U2(NX)
2      /F/      U1T(NX),      U2T(NX)
3      /S/      U1X(NX),      U1XX(NX),      U2X(NX),      U2XX(NX),
4      /R/      X1(NX),      X2(NX),      XL,      DX,
5      /I/      K1,      K2,      C12
6      REAL      K1,      K2
C...
C... BOUNDARY CONDITION AT X = 0
      NL=2
      U1X(1)=0.0E0
C...
C... BOUNDARY CONDITION AT X = 0.5 FOR U1
      NU=2
      U1X(NX)=(C12/K1)*(U2(1)-U1(NX))
C...
C... DERIVATIVE U1
      XX
C...      IF(NORUN.EQ.1)CALL DSS042(0.,XL,NX,U1,U1X,U1XX,NL,NU)
C...      IF(NORUN.EQ.2)CALL DSS044(0.,XL,NX,U1,U1X,U1XX,NL,NU)
C...
C... BOUNDARY AT X = 0.5 FOR U2
      NL=2
      U2X(1)=(K1/K2)*U1X(NX)
C...
C... BOUNDARY CONDITION AT X = 1
      NU=2

```

```

      U2X(NX)=0.0E0
C...
C...  DERIVATIVE U2
C...      XX
      IF(NORUN.EQ.1)CALL DSS042(XL,1.0,NX,U2,U2X,U2XX,NL,NU)
      IF(NORUN.EQ.2)CALL DSS044(XL,1.0,NX,U2,U2X,U2XX,NL,NU)
C...
C...  PDE FOR U1
      DO 1 I=1,NX
          U1T(I)=K1*U1XX(I)
1      CONTINUE
C...
C...  PDE FOR U2
      DO 2 I=1,NX
          U2T(I)=K2*U2XX(I)
2      CONTINUE
      RETURN
      END

      SUBROUTINE PRINT(NI,NO)
      PARAMETER(NX=26)
      COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U1(NX),      U2(NX)
2      /F/      U1T(NX),      U2T(NX)
3      /S/      U1X(NX),      U1XX(NX),      U2X(NX),      U2XX(NX),
4      /R/      X1(NX),      X2(NX),      XL,      DX,
5      /I/      K1,      K2,      C12
6      REAL      K1,      K2
C...
C...  PRINT A HEADING FOR THE SOLUTION
      IP=IP+1
      IF(IP.EQ.1)THEN
          WRITE(NO,1)
1      FORMAT(5X,'t'
2      ,1X,' U1(0,t)',1X,' U1(0.3,t)',1X,' U1(0.5,t)'
          ,1X,' U2(0.5,t)',1X,' U2(0.7,t)',1X,' U2(1.0,t)')
          END IF
C...
C...  COMPUTE THE TOTAL ENERGY
      CALL DERV
      ENERGY=0.
C...
C...  ENERGY FOR U1
      DO 2 I=2,NX
          ENERGY=ENERGY+0.5*(U1(I-1)+U1(I))*DX
2      CONTINUE
C...
C...  ENERGY FOR U2
      DO 3 I=2,NX
          ENERGY=ENERGY+0.5*(U2(I-1)+U2(I))*DX

```

```

3      CONTINUE
C...
C...  PRINT THE NUMERICAL SOLUTION
      WRITE(NO,4)T,U1(1),U1(16),U1(26),
      +      U2(1),U2(11),U2(26),
      +      ENERGY
4      FORMAT(F6.1,6F11.5,/,,' ENERGY = ',F8.5,/)
      RETURN
      END

```

```

BOUNDARY CONDITION OF THE THIRD TYPE VIA SECOND ORDER FINITE DIFFERENCES
0.      25.0      2.5
      52      0.00001
BOUNDARY CONDITION OF THE THIRD TYPE VIA FOURTH ORDER FINITE DIFFERENCES
0.      25.0      2.5
      52      0.00001
END OF RUNS

```

Program 9: Solution of Two Region Problem by Finite Differences

The following points should be noted about Program 9:

- (1) Two arrays in COMMON/Y/, U1(NX) and U2(NX) (NX = 26), represent dependent variables u_1 and u_2 in equations (49) to (56). The temporal derivatives are in U1T(NX) and U2T(NX) in COMMON/F/, representing u_{1t} and u_{2t} , respectively. The spatial derivatives are in U1X(NX), U1XX(NX), U2X(NX) and U2XX(NX) in COMMON/S/, representing u_{1x} , u_{1xx} , u_{2x} and u_{2xx} , respectively.
- (2) Initial conditions (51) and (52) are implemented in DO loop 2 in INITAL.
- (3) u_{1xx} and u_{2xx} are computed by calls to subroutine DSS042 (for NORUN = 1) or DSS044 (for NORUN = 2).
- (4) Boundary conditions (53) to (55) are implemented before the calls to DSS042 or DSS044. Also, a ratio of boundary conditions (55) to (56) is used as the fourth boundary condition, i.e.,

$$u_{2x}(0.5,t) = (k_1/k_2)u_{1x}(0.5,t)$$

This choice of a fourth boundary condition is arbitrary, and equation (56) could have been used.

(5) The total energy of the system is computed in PRINT before the numerical solution is printed.

Abbreviated output from Program 9 is listed in Table 7

```

RUN NO. - 1 BOUNDARY CONDITION OF THE THIRD TYPE VIA SECOND ORDER
           FINITE DIFFERENCES

INITIAL T - 0.000E+00
FINAL T - 0.250E+02
PRINT T - 0.250E+01

NUMBER OF DIFFERENTIAL EQUATIONS - 52

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-04

    t      U1(0,t)  U1(0.3,t)  U1(0.5,t)  U2(0.5,t)  U2(0.7,t)  U2(1.0,t)
  0.0      0.00000  0.00000    0.00000    1.00000    1.00000    1.00000
ENERGY = 0.50000

    2.5      0.30693  0.31028    0.31619    0.68381    0.68972    0.69307
ENERGY = 0.50000

    5.0      0.42663  0.42791    0.43016    0.56984    0.57209    0.57336
ENERGY = 0.50000

    7.5      0.47213  0.47261    0.47347    0.52652    0.52738    0.52786
ENERGY = 0.50000

   10.0      0.48941  0.48960    0.48991    0.51007    0.51039    0.51058
ENERGY = 0.49999

           .
           .
           .
   25.0      0.49995  0.49996    0.49996    0.50001    0.50001    0.50001
ENERGY = 0.49998

COMPUTATIONAL STATISTICS

LAST STEP SIZE                0.175E+01

LAST ORDER OF THE METHOD                3

```

TOTAL NUMBER OF STEPS TAKEN 95
 NUMBER OF FUNCTION EVALUATIONS 135
 NUMBER OF JACOBIAN EVALUATIONS 4

RUN NO. - 2 BOUNDARY CONDITION OF THE THIRD TYPE VIA FOURTH ORDER
 FINITE DIFFERENCES

INITIAL T - 0.000E+00

FINAL T - 0.250E+02

PRINT T - 0.250E+01

NUMBER OF DIFFERENTIAL EQUATIONS - 52

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-04

t	U1(0,t)	U1(0.3,t)	U1(0.5,t)	U2(0.5,t)	U2(0.7,t)	U2(1.0,t)
0.0	0.00000	0.00000	0.00000	1.00000	1.00000	1.00000
ENERGY = 0.50000						
2.5	0.30693	0.31028	0.31619	0.68382	0.68974	0.69309
ENERGY = 0.50001						
5.0	0.42660	0.42788	0.43013	0.56990	0.57215	0.57342
ENERGY = 0.50001						
7.5	0.47208	0.47257	0.47342	0.52659	0.52744	0.52793
ENERGY = 0.50000						
10.0	0.48938	0.48956	0.48989	0.51013	0.51046	0.51065
ENERGY = 0.50001						
	.				.	
	.				.	
	.				.	
25.0	0.50000	0.50000	0.50000	0.50002	0.50002	0.50002
ENERGY = 0.50001						

COMPUTATIONAL STATISTICS

LAST STEP SIZE 0.150E+01

LAST ORDER OF THE METHOD 2

TOTAL NUMBER OF STEPS TAKEN	107
NUMBER OF FUNCTION EVALUATIONS	166
NUMBER OF JACOBIAN EVALUATIONS	5

Table 7: Output from Program 9

The two solutions (for NORUN = 1 and 2) are in close agreement, and the energy is conserved to about five figures.

(6.2) Finite Volumes

A second approach to this problem with a conductance at $x = 0.5$ is based on finite volumes, and in particular, equations (44) to (47). To include the conductance, we modify equation (47) to

$$C = \frac{k_{i-1,i}}{(\Delta x_{i-1} + \Delta x_i)} = \frac{1}{\left(\frac{\Delta x_{i-1}}{k_{i-1}}\right) + \frac{1}{C_v} + \left(\frac{\Delta x_i}{k_i}\right)} \quad (57)$$

or for a uniform grid with spacing Δx ,

$$C = \frac{k_{i-1,i}}{2\Delta x} = \frac{1}{\left(\frac{\Delta x}{k_{i-1}}\right) + \frac{1}{C_v} + \left(\frac{\Delta x}{k_i}\right)} \quad (58)$$

If Δx is now defined as the distance between two adjacent temperatures (i.e., twice the Δx in equation (58), see Figure 3), equation (58) becomes

$$C = \frac{k_{i-1,i}}{\Delta x} = \frac{1}{\left(\frac{0.5\Delta x}{k_{i-1}}\right) + \frac{1}{C_v} + \left(\frac{0.5\Delta x}{k_i}\right)} \quad (59)$$

Equation (59) is used in subroutine DERV in Program 10, along with equations (44), (45) and (46), which for the present formulation with a uniform grid, are

$$V\rho_i C_{pi} \frac{du_i}{dt} = A_{i-1}q_{i-1} - A_i q_i \quad (60)$$

$$q_{i-1} = C(u_{i-1} - u_i), \quad q_i = C(u_i - u_{i+1}) \quad (61)(62)$$

where C is given by equation (59) (with the $\frac{1}{C_v}$ term removed if $x \neq 0.5$). Note that the denominator of equation (59) is just the sum of three thermal resistances in series, i.e., $(\frac{0.5\Delta x}{k_{i-1}}) + \frac{1}{C_v} + (\frac{0.5\Delta x}{k_i})$

(and we use the idea that a reciprocal resistance is a conductance, or in other words, equations (61) and (62) are a thermal analog of Ohm's law). In Program 10, we take $\rho_i = C_{pi} = k_{i-1} = k_i = A_{i-1} = A_i = A = 1$, $V = A\Delta x$, $C_v = 0.1$

```

SUBROUTINE INITAL
PARAMETER(NX=50)
COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U(NX)
2      /F/      UT(NX)
3      /R/      XG(NX),      XL,      DX,
4      KV,      CV,      C1,      C2
5      /I/      IP
REAL          KV
C...
C...  FULL LENGTH
      XL=1.0
C...
C...  GRID INCREMENT
      DX=XL/FLOAT(NX)
C...
C...  GRID POINTS
      DO 1 I=1,NX
      XG(I)=FLOAT(I-1)*DX
1      CONTINUE
C...
C...  CONDUCTIVITY
      KV=1.0
C...
C...  CONDUCTANCE
      CV=0.1
C...
C...  CONSTANTS FOR COMPUTING HEAT FLUXES ON A UNIFORM GRID
      C1=1.0/((0.5*DX/KV)+(0.5*DX/KV))
      C2=1.0/((0.5*DX/KV)+(1.0/CV)+(0.5*DX/KV))
C...
C...  INITIAL CONDITION
      U0=1.
      DO 2 I=1,NX
      IF(I.LE.25)U(I)=0.
      IF(I.GT.25)U(I)=U0
2      CONTINUE
C...

```

```

C... INITIALIZE COUNTER FOR PRINTED AND PLOTTED SOLUTION
IP=0
RETURN
END

```

```

SUBROUTINE DERV
PARAMETER(NX=50)
COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U(NX)
2      /F/      UT(NX)
3      /R/      XG(NX),  XL,      DX,
4      KV,      CV,      C1,      C2
5      /I/      IP
REAL          KV

```

```

C...
C... DEFINE ARRAY FOR THE FLUXES
REAL Q(0:NX)
C...
C... BOUNDARY CONDITION AT X = 0
Q(0)=0.
C...
C... BOUNDARY CONDITION AT X = 1
Q(NX)=0.
C...
C... COMPUTE THE FLUXES (K/(RHO*CP) = 1)
DO 2 I=1,NX-1
  Q(I)=C1*(U(I)-U(I+1))
2 CONTINUE
C...
C... INCLUDE CONDUCTANCE
Q(25)=C2*(U(25)-U(26))
C...
C... EQUATION (1)
DO 1 I=1,NX
  UT(I)=(Q(I-1)-Q(I))/DX
1 CONTINUE
RETURN
END

```

```

SUBROUTINE PRINT(NI,NO)
PARAMETER(NX=50)
COMMON/T/      T,      NSTOP,      NORUN
1      /Y/      U(NX)
2      /F/      UT(NX)
3      /R/      XG(NX),  XL,      DX,
4      KV,      CV,      C1,      C2
5      /I/      IP
REAL          KC

```

```

C...
C... COMPUTE THE TOTAL ENERGY
CALL DERV

```

```

ENERGY=0.
  DO 2 I=1,NX
    ENERGY=ENERGY+U(I)*DX
2    CONTINUE
C...
C... PRINT THE NUMERICAL SOLUTION TO EQUATIONS (1) TO (4) AND THE TOTAL
C... ENERGY
      UAVG=(U(NX/2)+U(NX/2+1))/2.0
      WRITE(NO,4)T,ENERGY,U(1),U(6),U(11),U(16),U(21),UAVG
4    FORMAT('      T = ',F6.2,' ENERGY = ',F9.5,
1    /,4X,'X=0.01',4X,'X=0.11',4X,'X=0.21',
2    4X,'X=0.31',4X,'X=0.41',4X,'X=0.50',/,
3    6F10.6,/)
      RETURN
      END

```

Program 10: Solution of Two Region Problem by Finite Volumes

The code in Program 10 follows directly from equations (59) to (62). There are a few points we should note:

(1) Only one dependent variable is used in contrast with Program 10, i.e., $U(NX)$ in COMMON/Y/, with the temporal derivative $UT(NX)$ in COMMON/F/. Also, the grid now has 50 points rather than the 52 in Program 9 because of the characteristics of the grid in Figure 2. This change in the number of grid points is also reflected in the third line of data.

(2) Spatial derivatives are not contained in arrays as in Program 10 since spatial derivatives do not appear explicitly in equations (59) to (62).

(3) The conductance used in subroutine DERV is $C2$ when $x = 0.5$ and $C1$ when $x \neq 0.5$, in accordance with equation (59), with ($C2$) or without ($C1$) the term $\frac{1}{C_V}$.

The output from Program 10 is listed in Table 8

```

RUN NO. - 1 BOUNDARY CONDITION OF THE THIRD TYPE VIA FINITE VOLUMES
INITIAL T - 0.000E+00
FINAL T - 0.250E+02
PRINT T - 0.250E+01

```

NUMBER OF DIFFERENTIAL EQUATIONS - 50

INTEGRATION ALGORITHM - LSODES

MAXIMUM INTEGRATION ERROR - 0.100E-04

T =	0.00	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	
T =	2.50	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.306937	0.307385	0.308578	0.310512	0.313179	0.500000	
T =	5.00	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.426644	0.426814	0.427268	0.428003	0.429016	0.500000	
T =	7.50	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.472116	0.472182	0.472359	0.472630	0.473015	0.499996	
T =	10.00	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.489398	0.489422	0.489488	0.489594	0.489741	0.500000	
	:				:	
	:				:	
	:				:	
T =	25.00	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.499970	0.499970	0.499970	0.499970	0.499971	0.500000	

COMPUTATIONAL STATISTICS

LAST STEP SIZE	0.113E+01
LAST ORDER OF THE METHOD	4
TOTAL NUMBER OF STEPS TAKEN	97
NUMBER OF FUNCTION EVALUATIONS	135
NUMBER OF JACOBIAN EVALUATIONS	4

Table 8: Output from Program 10

The conservation of energy of the solution in Table 8 is to five figures. Also, the following comparison

of the finite difference and finite volume solutions at $t = 2.5$ indicates that the agreement is good.

Second order finite differences (from Table 7)

2.5	0.30693	0.31028	0.31619	0.68381	0.68972	0.69307
ENERGY =	0.50000					

Fourth order finite differences (from Table 7)

2.5	0.30693	0.31028	0.31619	0.68382	0.68974	0.69309
ENERGY =	0.50001					

Finite volumes (from Table 8)

T =	2.50	ENERGY =	0.50000			
X=0.01	X=0.11	X=0.21	X=0.31	X=0.41	X=0.50	
0.306937	0.307385	0.308578	0.310512	0.313179	0.500000	

The finite volume solution, e.g., $u(\Delta x/2, 2.5) = u(0.01, 2.5) = 0.306937$, and the finite difference solution is $u(0, 2.5) = 0.30693$. The close agreement indicates that the grid in Figure 2 works well for the method of finite volumes for Dirichlet and Neumann boundary conditions (from the results of Tables 5 and 6) and for boundary conditions of the third type (from Table 8).

(7) Summary

Through the use of a series of basic test problems, we have demonstrated the three principal methods of numerical approximation of PDEs: finite differences, finite elements and finite elements. The presentation is tutorial, and is intended to also illustrate the details of programming these three approximations when applied to PDEs that are first order in time and second order in space, i.e., parabolic PDEs. All of the programs discussed previously, and that produced the numerical results, are available from the authors (send inquiries to WES).

Generally, we conclude that the three methods are equivalent, with the possible exception that the method of finite elements, as used in the context of the test problems, was relatively more difficult to use, e.g., the analytical or numerical evaluation of the Galerkin integrals was required, and this must be done for each PDE problem that has new terms in the PDEs. Of course, finite elements are widely used, and they would possibly be selected for other problems in preference to finite differences or finite volumes.

We also offer the opinion that any numerical approximation of Fourier's second law should

obey energy conservation (or, perhaps more generally, satisfy invariance conditions based on conservation principles). However, we regard this as only a necessary condition for a numerical approximation. In other words, a numerical approximation could conceivably exhibit conservation of energy and still produce an inaccurate solution. This can perhaps be appreciated by the fact that the conservation of energy is a global property (typically computed by spatial integration) and therefore does not insure point-to-point accuracy of the numerical solution.

References

- (1) Schiesser, W. E., The Numerical Method of Lines Integration of Partial Differential Equations, Academic Press, San Diego, 1991
- (2) Carcagno, R. H. , J. A. Demko, W. E. Schiesser and A. Yücel, "A Numerical Investigation of Spatial Approximations for Strongly Convective Flows", internal report, SSC Cryogenics Department, July, 1991
- (3) Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, New York, 1980
- (4) Lick, W. J., Difference Equations from Differential Equations, Springer-Verlag, Berlin, 1989

Appendix 1

Derivation of the Analytical Solution for the Parabolic Test Problem with Neumann Boundary Conditions

The test problem, equations (1), (10), (11) and (12), is restated here

$$u_t = Du_{xx} \quad (\text{a.1})$$

$$u(x,0) = \begin{cases} 0, & 0 \leq x < 0.5 \\ 1, & 0.5 < x \leq 1 \end{cases} \quad (\text{a.2})$$

$$u_x(x_1,t) = 0, u_x(x_N,t) = 0 \quad (\text{a.3})(\text{a.4})$$

The solution to equations (a.1), (a.3) and (a.4) is easily derived (with $D = 1$, $x_1 = 0$, $x_N = 1$) as

$$u(x,t) = Ce^{-\lambda_n^2 t} \cos(\lambda_n x) \quad (\text{a.5})$$

where the eigenvalues are $\lambda_n = n\pi$, $n = 0, 1, 2, \dots$ and C is a constant to be determined. Since equation (a.1) is linear, we can take a superposition of solutions of the form of equation (a.5) to satisfy the initial condition, equation (a.2)

$$u(x,t) = \sum_{n=0}^{\infty} C_n e^{-\lambda_n^2 t} \cos(\lambda_n x) \quad (\text{a.6})$$

We now follow the well known procedure of evaluating the Fourier coefficients, C_n , by multiplying equation (a.6) by $\cos(\lambda_m x)$ and integrating (with $t = 0$ since we are considering initial condition (a.2))

$$\int_{0.5}^1 1 \cos(\lambda_m x) dx = \int_0^1 \cos(\lambda_m x) \sum_{n=0}^{\infty} C_n \cos(\lambda_n x) dx \quad (\text{a.7})$$

Interchanging the orders of integration and summation in the RHS of equation (a.7), and using the orthogonality property

$$\int_0^1 \cos(\lambda_m x) \cos(\lambda_n x) dx = 0, m \neq n$$

$$\int_0^1 \cos^2(\lambda_n x) dx = \int_0^1 (1/2) \{1 + \cos(2\lambda_n x)\} dx = (1/2) \left\{ x + \left(\frac{1}{2\lambda_n} \right) \sin(2\lambda_n x) \right\} \Big|_0^1 = 1/2 \quad (\text{a.8})$$

The LHS integral of equation (a.7) is (with $\lambda_n = \lambda_m \neq 0$)

$$\int_{0.5}^1 \cos(\lambda_n x) dx = \left(\frac{1}{\lambda_n} \right) \sin(\lambda_n x) \Big|_{0.5}^1 = \left(\frac{-1}{n\pi} \right) \sin(0.5n\pi)$$

Thus

$$\begin{aligned} C_n &= \left(\frac{-2}{n\pi} \right) \sin(0.5n\pi), \quad n = 1, 2, \dots \\ &= -2/\pi, \quad 2/(3\pi), \quad -2/(5\pi), \quad 2/(7\pi), \dots \end{aligned} \quad (\text{a.9})$$

For the case $\lambda_n = 0$ ($n = 0$) equation (a.7) becomes

$$\int_{0.5}^1 1 \, dx = C_0 \int_0^1 1^2 \, dx$$

or

$$C_0 = 1/2 \quad (\text{a.10})$$

Equations (a.6), (a.9) and (a.10) are then the complete solution, which are also stated as equation (14).

Appendix 2

Main Program for the Numerical Solution
of the Test Problems via Subroutine LSODES

```
PROGRAM LSODES2
C...
C... THE FOLLOWING PROGRAM INTEGRATES A SYSTEM OF ODES, DEFINED BY
C... THE USER-SUPPLIED SUBROUTINES INITIAL, DERV AND PRINT, PLUS DATA.
C... THE SYSTEM OF ODES PROGRAMMED IN SUBROUTINE DERV IS INTEGRATED
C... BY LSODES (THE SPARSE MATRIX SOLVER IN ODEPACK, A LIBRARY OF
C... INTEGRATORS DEVELOPED BY ALAN C. HINDMARSH, LAWRENCE LIVERMORE
C... NATIONAL LABORATORY).
C...
C... THE MODEL INITIAL CONDITIONS ARE SET IN SUBROUTINE INITIAL, AND
C... THE MODEL DERIVATIVES ARE PROGRAMMED IN SUBROUTINE DERV. THE
C... NUMERICAL SOLUTION IS PRINTED AND PLOTTED IN SUBROUTINE PRINT.
C...
C... THE FOLLOWING CODING IS FOR 250 ORDINARY DIFFERENTIAL EQUATIONS.
C... IF MORE ODES ARE TO BE INTEGRATED, ALL OF THE 250*S SHOULD BE
C... CHANGED TO THE REQUIRED NUMBER
COMMON/T/          T,      NSTOP,      NORUN
1      /Y/          Y(250)
2      /F/          F(250)
C...
C... COMMON AREA TO PROVIDE THE INPUT/OUTPUT UNIT NUMBERS TO OTHER
C... SUBROUTINES
COMMON/IO/         NI,      NO
C...
C... ABSOLUTE DIMENSIONING OF THE ARRAYS REQUIRED BY LSODES. ARRAY
C... WORK IS SIZED BY THE FOLLOWING FORMULAS TAKEN DIRECTLY FROM THE
C... LSODES DOCUMENTATION:
C...
C RWORK = REAL WORK ARRAY OF LENGTH AT LEAST..
C          20 + 16*NEQ          FOR MF = 10,
C          20 + (2 + 1./LENRAT)*NNZ + (11 + 9./LENRAT)*NEQ
C                               FOR MF = 121 OR 222,
C
C WHERE..
C NNZ      = THE NUMBER OF NONZERO ELEMENTS IN THE SPARSE
C           JACOBIAN (IF THIS IS UNKNOWN, USE AN ESTIMATE), AND
C LENRAT = THE REAL TO INTEGER WORDLENGTH RATIO (USUALLY 1 IN
C           SINGLE PRECISION AND 2 IN DOUBLE PRECISION).
C MF      = METHOD FLAG. STANDARD VALUES ARE..
C          10 FOR NONSTIFF (ADAMS) METHOD, NO JACOBIAN USED.
C          121 FOR STIFF (BDF) METHOD, USER-SUPPLIED SPARSE JACOBIAN.
C          222 FOR STIFF METHOD, INTERNALLY GENERATED SPARSE JACOBIAN.
C...
```

```

C...  THUS, FOR MF = 222, NEQ (= NEQN IN THE SUBSEQUENT CODING) = 250,
C...  NHZ = 1250, LENRAT = 1, 20+(2+1./1.)*1250+(11+9./1.)*250 = 9000
C...  (APPROXIMATELY)
C...  DIMENSION YV(250), RWORK(10000), IWORK(30)
C...
C...  EXTERNAL THE DERIVATIVE AND ODE JACOBIAN MATRIX ROUTINES CALLED BY
C...  LSODES
C...  EXTERNAL FCN, JAC
C...
C...  ARRAY FOR THE TITLE (FIRST LINE OF DATA), CHARACTERS  END OF RUNS
C...  CHARACTER TITLE(20)*4, ENDRUN(3)*4
C...
C...  VARIABLE FOR THE TYPE OF ERROR CRITERION
C...  CHARACTER*3 ABSREL
C...
C...  DEFINE THE CHARACTERS  END OF RUNS
C...  DATA ENDRUN/'END ', 'OF R', 'UNS '/
C...
C...  DEFINE THE INPUT/OUTPUT UNIT NUMBERS
C...  NI=5
C...  NO=6
C...
C...  OPEN INPUT AND OUTPUT FILES
C...  OPEN(NI,FILE='DATA', STATUS='OLD')
C...  OPEN(NO,FILE='OUTPUT',STATUS='NEW')
C...
C...  INITIALIZE THE RUN COUNTER
C...  NORUN=0
C...
C...  BEGIN A RUN
C...  1  NORUN=NORUN+1
C...
C...  INITIALIZE THE RUN TERMINATION VARIABLE
C...  NSTOP=0
C...
C...  READ THE FIRST LINE OF DATA
C...  READ(NI,1000,END=999)(TITLE(I),I=1,20)
C...
C...  TEST FOR  END OF RUNS  IN THE DATA
C...  DO 2 I=1,3
C...  IF(TITLE(I).NE.ENDRUN(I))GO TO 3
C...  2  CONTINUE
C...
C...  AN END OF RUNS HAS BEEN READ, SO TERMINATE EXECUTION
C...  999  STOP
C...
C...  READ THE SECOND LINE OF DATA
C...  3  READ(NI,1001,END=999)TO,TF,TP
C...
C...  READ THE THIRD LINE OF DATA
C...  READ(NI,1002,END=999)NEQN,ERROR

```

```

C...
C... PRINT A DATA SUMMARY
      WRITE(NO,1003)NORUN,(TITLE(I),I=1,20),
      1          TO,TF,TP,
      2          NEQN,ERROR
C...
C... INITIALIZE TIME
      T=TO
C...
C... SET THE INITIAL CONDITIONS
      CALL INITAL
C...
C... PRINT THE INITIAL CONDITIONS
      CALL PRINT(NI,NO)
C...
C... SET THE INITIAL CONDITIONS FOR SUBROUTINE LSODES
      DO 5 I=1,NEQN
      YV(I)=Y(I)
5     CONTINUE
C...
C... SET THE PARAMETERS FOR SUBROUTINE LSODES
      TV=TO
      ITOL=1
      RTOL=ERROR
      ATOL=ERROR
      LRW=10000
      LIW=30
      IOPT=1
      ITASK=1
      ISTATE=1
C...
C... ONE NONSTIFF OPTION, MF = 10, AND ONE STIFF OPTION, MF = 222,
C... ARE PROGRAMMED HERE. SEVERAL OTHER OPTIONS ARE AVAILABLE FOR
C... LSODES WHICH SHOULD BE CONSIDERED WHEN RUNNING PROBLEMS THAT
C... LARGER AND MORE COMPLEX THAN THE RELATIVELY SMALL TEST PROBLEM
C... IN SUBROUTINES INITAL, DERV AND PRINT
C...
C...      (1) NORUN = 1
C...
C...      IMPLICIT ADAMS METHOD, FUNCTIONAL ITERATION
C...      IF(NORUN.EQ.1)THEN
C...      MF=10
C...
C...      (2) NORUN = 2
C...
C...      IMPLICIT ADAMS METHOD, CHORD ITERATION WITH AN INTERNALLY
C...      GENERATED (DIFFERENCE QUOTIENT) SPARSE JACOBIAN
C...      ELSE IF(NORUN.EQ.2)THEN
C...      MF=222
C...      END IF
C...

```

```

C... CALL SUBROUTINE LSODES TO COVER ONE PRINT INTERVAL
4   TOUT=TV+TP
C...
C... REDEFINE THE SPARSITY STRUCTURE
C... ISTATE=3
      CALL LSODES(FCN,NEQN,YV,TV,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,
1     IOPT,RWORK,LRW,IWORK,LIW,JAC,MF)
C...
C... PRINT THE SOLUTION
      T=TV
      DO 6 I=1,NEQN
        Y(I)=YV(I)
6     CONTINUE
      CALL PRINT(NI,NO)
C...
C... TEST FOR AN ERROR CONDITION
      IF(ISTATE.LT.0)THEN
C...
C... PRINT A MESSAGE INDICATING AN ERROR CONDITION
      WRITE(NO,1004)ISTATE
C...
C... GO ON TO THE NEXT RUN
      GO TO 1
      END IF
C...
C... CHECK FOR A RUN TERMINATION
      IF(NSTOP.NE.0)GO TO 1
C...
C... CHECK FOR THE END OF THE RUN
      IF(TV.LT.(TF-0.5*TP))GO TO 4
C...
C... THE CURRENT RUN IS COMPLETE, SO PRINT THE COMPUTATIONAL STAT-
C... ISTICS FOR LSODES AND GO ON TO THE NEXT RUN
      WRITE(NO,8)RWORK(11),IWORK(14),IWORK(11),IWORK(12),IWORK(13)
8     FORMAT(1H ,//,' COMPUTATIONAL STATISTICS' ,//,
1     ' LAST STEP SIZE ', E10.3,//,
2     ' LAST ORDER OF THE METHOD ', I10,//,
3     ' TOTAL NUMBER OF STEPS TAKEN ', I10,//,
4     ' NUMBER OF FUNCTION EVALUATIONS ', I10,//,
5     ' NUMBER OF JACOBIAN EVALUATIONS ', I10,/)
      GO TO 1
C...
C... *****
C...
C... FORMATS
C...
1000 FORMAT(20A4)
1001 FORMAT(3E10.0)
1002 FORMAT(I5,20X,E10.0)
1003 FORMAT(1H1,
1     ' RUN NO. - ',I3,2X,20A4,//,

```

```

2 ' INITIAL T - ',E10.3,/,
3 '   FINAL T - ',E10.3,/,
4 '   PRINT T - ',E10.3,/,
5 ' NUMBER OF DIFFERENTIAL EQUATIONS - ',I4,/,
6 ' INTEGRATION ALGORITHM - LSODES ',/,
7 ' MAXIMUM INTEGRATION ERROR - ',E10.3,/,
8 1H1)
1004 FORMAT(1H ,/, ' ISTATE = ',I3,/,
1 ' INDICATING AN INTEGRATION ERROR, SO THE CURRENT RUN' ,/,
2 ' IS TERMINATED. PLEASE REFER TO THE DOCUMENTATION FOR' ,/,
3 ' SUBROUTINE',/,25X,'LSODES',/,
4 ' FOR AN EXPLANATION OF THESE ERROR INDICATORS' )
END

SUBROUTINE FCN(NEQN,TV,YV,YDOT)
C...
C... SUBROUTINE FCN IS AN INTERFACE ROUTINE BETWEEN SUBROUTINES LSODES
C... AND DERV
C...
COMMON/T/          T,      NSTOP,      NORUN
1   /Y/          Y(1)
2   /F/          F(1)
C...
C... VARIABLE DIMENSION THE DEPENDENT AND DERIVATIVE ARRAYS
DIMENSION YV(NEQN), YDOT(NEQN)
C...
C... TRANSFER THE INDEPENDENT VARIABLE, DEPENDENT VARIABLE VECTOR
C... FOR USE IN SUBROUTINE DERV
T=TV
DO 1 I=1,NEQN
Y(I)=YV(I)
1 CONTINUE
C...
C... EVALUATE THE DERIVATIVE VECTOR
CALL DERV
C...
C... TRANSFER THE DERIVATIVE VECTOR FOR USE BY SUBROUTINE LSODES
DO 2 I=1,NEQN
YDOT(I)=F(I)
2 CONTINUE
RETURN
END

SUBROUTINE JAC(NEQ,T,Y,J,IAN,JAN,PDJ)
C...
C... SUBROUTINE JAC IS CALLED ONLY IF AN OPTION OF LSODES IS SELECTED
C... FOR WHICH THE USER MUST SUPPLY THE ANALYTICAL JACOBIAN MATRIX OF
C... THE ODE SYSTEM. THE PROGRAMMING OF THE ANALYTICAL JACOBIAN IN
C... SUBROUTINE JAC IS DESCRIBED IN DETAIL IN THE DOCUMENTATION COMMENTS
C... COMMENTS AT THE BEGINNING OF SUBROUTINE LSODES.
C...

```

RETURN
END