P.R.L.

TECHNICAL NOTE

81-24 TN ~~80.05~~

NUMERICAL SOLUTION OF LINEAR PARABOLIC PARTIAL DIFFERENTIAL EQUATION WITH INFINITE RANGE FOR SPACE VARIABLE

By

N. Hussain, S. K. Rao and P. Sharma

1980

PHYSICAL RESEARCH LABORATORY
AHMEDABAD-9

NUMERICAL SOLUTION OF LINEAR PARABOLIC PARTIAL DIFFERENTIAL EQUATION WITH INFINITE RANGE FOR SPACE VARIABLE

N. Hussain, S.K. Rao+ and P. Sharma
Physical Research Laboratory
Ahmedabad 380009

Abstract

Linear parabolic partial differential equation is encountered in many areas of science. Some of the well-known examples are heat conduction in solids, dispersion of tracers in geophysical reservoirs, movement of groundwater and moisture in soils, dispersal of pollutants in water and air etc.

We have used the well-known Crank-Nicholson Scheme in solving a general linear parabolic partial differential equation with infinite range for space variable. A new feature of the algorithm developed by us is the transformation of the space variable to make the spatial grid equally spaced between zero and infinity. This procedure saves efforts as well as computer time.

+Present address: Dept. of Mathematics
Indian Institute of Technology
Bombay 400 076

NUMERICAL SOLUTION OF LINEAR PARABOLIC PARTIAL DIFFERENTIAL EQUATION WITH INFINITE RANGE FOR SPACE VARIABLE

N. Hussain, S.K. Rao[+] and P. Sharma
Physical Research Laboratory
Ahmedabad 380 009

This report deals with the development of finite difference method of solving a linear parabolic partial differential equation of the form:

$$\frac{\delta u}{\delta t} = f_1(t,y)\frac{\delta^2 u}{\delta y^2} + f_2(t,y)\frac{\delta u}{\delta y} + f_3(t,y)\,u + f_4(t,y) \quad \ldots (1)$$

with the initial and boundary conditions as:

$$\left.\begin{aligned} u(o,t) &= g_1(t) \\ u(\infty,t) &= g_2(t) \\ u(y,0) &= g_3(y) \end{aligned}\right\} \quad \ldots (2)$$

where f_1, f_2, f_3, f_4 are specified functions of t and y; g_1, g_2 are given functions of t only and g_3 is a given function of y.

For finite difference methods, it is known that step size in the spatial direction has to be smaller when the gradients are higher. Thus step size should be smaller year $y = 0$ and has to be made progressively larger as $y \rightarrow \infty$. Such a choice of grid making, however, creates practical difficulties. To obviate the above difficulty, we introduce the transformation

---

[+]Present address: Department of Mathematics
Indian Institute of Technology
Bombay 400 076

$$z = \exp(-y^{\beta}) \quad \ldots\ldots \quad (3)$$

which brings the range of spatial variable z between 1 and 0 corresponding to the range of y variable between 0 and ∞. This transformation has the additional advantage of uniform spacing of the interval (o, 1) in the z-variable.

The grid points in the z variable are 1, (n-1)h, (n-2)h, ----------2h, h, 0; where h = 1/n (zero corresponds to ∞ of the original spatial variable y). The choice of β can be made in such a way that

$$y = (-\ln h)^{1/\beta}$$

is of the order of infinity for the problem in question.

From (3)

$$z' = \frac{dz}{dy} = z\ (-\beta y^{\beta-1}) \quad \ldots\ldots \quad (4)$$

$$z'' = \frac{d^2 z}{dy^2} = z\ y^{\beta-1}\left[\beta y^{\beta-1} - (\beta-1)/y\right. \quad \ldots\ldots \quad (5)$$

Also

$$\frac{\partial u}{\partial y} = \frac{\partial u}{\partial z}\frac{dz}{dy} = \frac{\partial u}{\partial z} z' \quad \ldots\ldots \quad (6)$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial z}\frac{d^2 z}{dy^2} + \frac{\partial^2 u}{\partial z^2}\left(\frac{dz}{dy}\right)^2 = \frac{\partial u}{\partial z} z'' + \frac{\partial u^2}{\partial z^2} z'^2 \quad (7)$$

using (6) and (7) the given parabolic differential equation, (1), takes the form

$$\frac{\partial u}{\partial t} = f_1(t,z)\left[z'^2 \frac{\partial^2 u}{\partial z^2} + z'' \frac{\partial u}{\partial z}\right] + f_2(t,z)\ z' \frac{\partial u}{\partial z}$$

$$+ f_3 u + f_4$$

or, $$\frac{\partial u}{\partial t} = f_1\ z'^2 \frac{\partial^2 u}{\partial z^2} + (f_1 z'' + f_2 z')\frac{\partial u}{\partial z}$$

$$+ f_3 u + f_4 \quad \ldots\ldots \quad (8)$$

with the transformed initial and boundary conditions corresponding to (2) as:

$$\left.\begin{aligned} u(1, t) &= g_1(t) \\ u(0, t) &= g_2(t) \\ u(z, 0) &= g_3(z) \end{aligned}\right\} \quad \ldots\ldots (9)$$

Applying the Crank-Nicholson scheme, the finite difference at the i^{th} space interval and for the time interval between j and j + 1 is given as:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \left[(E_i)_{i+1}^{j+1} + (E_i)^j \right] /2 \quad \ldots\ldots (10)$$

where

$$t_j = j.\Delta t \quad \ldots\ldots (11)$$

$$E_i = f_{1i} \; z_i'^2 \; \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + (f_{1i} z_i'' + f_{2i} z_i')$$

$$\frac{u_{i+1} - u_{i-1}}{2h} + f_{3i} \; u_i + f_{4i} \quad \ldots\ldots (12)$$

In (12) $f_{1i} = f_1(t, z_i)$ and likewise f_{2i}, f_{3i} and f_{4i}.

Writing out explicitly the terms using (11) and (12), the finite-difference equation (10) takes the form

$$A_{i-1} \; u_i^{j+1} + B_{i-1} \; u_{i+1}^{j+1} + F_{i-1} \; u_{i-1}^{j+1} = d_{i-1};$$

$$i = 2, 3, \text{---} n \quad \ldots\ldots (13)$$

where

$$
\left.\begin{aligned}
A_{i-1} &= \frac{2}{\Delta t} + \frac{2\, f_{1i}^{j+1}}{h^2} z_i'^{\,2}\, f_{3i}^{j+1} \\
B_{i-1} &= -f_{1i}^{j+1}\, z_i'^{\,2}/h^2 - (f_{1i}^{j+1}\, z_i'' + f_{2i}^{j+1}\, z_i')/2h \\
F_{i-1} &= -f_{1i}^{j+1}\, \frac{z_i'^{\,2}}{h^2} + (f_{1i}^{j+1}\, z_i'' + f_{2i}^{j+1}\, z_i')/2h \\
d_{i-1} &= (E_i)^j + f_{4i}^{j+1} + \frac{2}{\Delta t}\, u_i^j
\end{aligned}\right\} \quad \ldots. \; (14)
$$

Equation (13) represents (n-1) simultaneous linear equations in (n-1) unknowns at $(j+1)^{th}$ level of the variable t, namely

$$
\begin{bmatrix}
A_1 & B_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
F_1 & A_2 & B_2 & 0 & \cdots & 0 & 0 & 0 \\
0 & F_2 & A_3 & B_3 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & F_{n-3} & A_{n-2} & B_{n-2} \\
0 & 0 & 0 & 0 & \cdots & 0 & F_{n-2} & A_{n-1}
\end{bmatrix}
\begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_n \end{bmatrix}
=
\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{n-1} \end{bmatrix}
$$

$$
; \quad i = 1, 2 \;\cdots\; n \qquad \ldots. \; (15)
$$

The boundary conditions will alter the first and last component of the right hand side of (15) as:

$$
\left.\begin{aligned}
d_1 &= d_1 - F_1\, u_1 \\
d_{n-1} &= d_{n-1} - B_{n-1}\, u_{n+1}
\end{aligned}\right\} \quad \ldots. \; (16)
$$

where u_1 and u_{n+1} are known from the given boundary conditions.

For t = 0, the profile of u is given (i.e., initial condition), the profile at $t = \Delta t$ can be obtained by solving these simultaneous equations. We have solved this tridiagonal system of linear equations by the Gaussian Elimination Method (Subroutine "TRID", Appendix 1). This subroutine returns the solution in variables F_1 to F_{n-1}. The values of u at the required points of the space variable y has been done using the nearest four points in the solution, (two above and two below) by fitting a third degree polynomial.

It may be noted that in some physical problems the initial condition

$$u(y, 0) = g(y)$$

may be the form

$$u(y, \infty) = g(y)$$

In such cases the transformation $T = 1/t$ will bring the problem to the above formulation.

AN ALGORITHM FOR THE FINITE DIFFERENCE SCHEME

In order to illustrate the above method we give an algorithm (Appendix 1) for solving the second order partial differential equation:

$$\frac{\partial C(x,t)}{\partial t} = \frac{D\partial^2 C}{\partial x^2} - \frac{u\partial C}{\partial x} \quad \text{.... (17)}$$

For the initial and boundary conditions:

$$\left.\begin{aligned} C(o,t) &= C_o \sin \omega t \text{ ------(a)} \\ C(\infty,t) &= o \text{ ------(b)} \\ C(x,o) &= o \text{ ------(c)} \end{aligned}\right) \quad \text{.... (18)}$$

Equations (17) and (18) describe the behaviour of a tracer in a dispersive groundwater system. 'C' represents the tracer concentration (dimension, ML^{-3}) at a time 't' at a distance 'x' from the recharge boundary (x=o) of the aquifer. Thus it is assumed that the tracer is injected continuously into the aquifer at the recharge boundary, the concentration varying with time sinusoidally as given by equation 18(a). The tracer flows with the water with a velocity 'u' and during the flow the tracer gets dispersed into the aquifer, the disperson coefficient being D (dimension, $L^2\ T^{-1}$). It is customary to use non-dimensional distance parameter xu/D (say 'y' for brevity) and the dispersion time constant, D/u^2 (dimension, T^{-1}). With these notations (17) becomes

$$\frac{\partial C(y,t)}{\partial t} = \frac{1}{(D/u^2)}\frac{\partial^2 C(y,t)}{\partial y^2} - \frac{1}{(D/u^2)}\frac{\partial C(y,t)}{\partial y} \quad \ldots. \ (19)$$

with the transformed initial and boundary conditions as:

$$C(y,o) = 0;\ C(o,t) = C_o \sin \omega t;\ C(\infty,0) = 0 \quad \ldots. \ (20)$$

On comparing (19) and (20) with the general linear parabolic partial differential equation (1) and the initial and boundary conditions (2) respectively, one obtains

$$\left.\begin{aligned} f_1(t,y) &= 1/(D/u^2) \\ f_2(t,y) &= -1/(D/u^2) \\ f_3(t,y) &= 0 \\ f_4(t,y) &= 0 \end{aligned}\right\} \quad \ldots. \ (21)$$

$$\left.\begin{aligned} g_1(t) &= C_o \sin \omega t \\ g_2(t) &= 0 \\ g_3(y) &= 0 \end{aligned}\right) \quad \ldots\ldots (22)$$

Equation (19) has been solved by the Crank-Nicholson Method for the given initial and boundary conditions (18) by appropriately defining (using equations 21 and 22) f_1, f_2, f_3m f_4, g_1, g_2 and g_3 in FUNCTION Subprograms F1, F2, F3, F4, G1, G2, and G3 given in the algorithm (Appendix 1). In order to get the normalized concentrations (i.e. $C(y,t)/C_o$) we put $C_o = 1$ wherever it appears in (21) and (22).

In Fig. 1 the continuous curves A and B give $C(y,t)/C_o$ vs. t at $xu/D = 5$ and 10 respectively, obtained from the method given above. The value of D/u^2 taken for both the curves is 100 years and that of $\omega = 2\pi/T = 2\pi/1000\ yr^{-1}$ (i.e. period of the sine wave as 1000 years, see inset in Fig. 1).

We have also solved equation (17) for the initial and boundary conditions given in (18) analytically; Appendix 2 gives the analytical solution. The values obtained from the analytical solution for the same values of parameters D/u^2 and xu/D are shown by 'dots' in Fig. 1. It is seen that the numerical solution agrees closely with the analytical solution.

Table 1 gives a comparison of the values of tracer concentration obtained from the numerical and analytical methods at $xu/D = 5$ and for various values of time (normalised with the period of the sine wave). The error is the difference of the solutions expressed as a percentage of the analytical value.

t/T	Finite Difference Solution	Analytical Solution	Difference	Percentage Error
0.25	0.067171	0.067277	-0.000106	-1.57
0.50	0.459601	0.459920	-0.000319	-0.07
0.75	0.341715	0.341093	+0.000622	0.18
1.0	-0.270253	-0.270703	+0.000450	-0.17
1.5	0.305339	0.305933	-0.000594	-0.19
2.0	-0.298452	-0.298945	+0.000493	-0.16

The Crank-Nicholson Implicit Method has advantage over the "Explicit" methods in that the former is unconditionally stable and convergent for all finite values of step size of both space and time variables. In addition for a linear problem, the resulting tridiagonal system of equations can be conveniently solved saving a lot of computer time.

Acknowledgements

We thank Shri Avinash Khare for help in obtaining the analytical solution of dispersion convection equation given in Appendix 2.

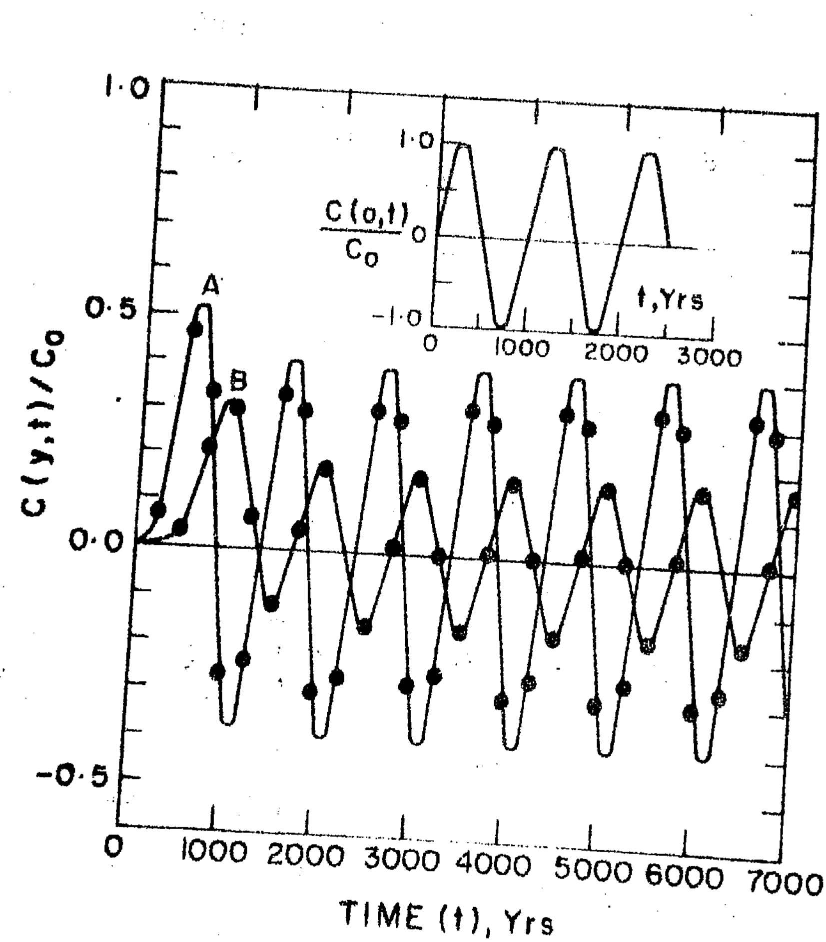

Fig. 1: Normalized radiotracer concentration, $C(y,t)/C_o$, in an aquifer having sinusoidal tracer concentration input at the recharge boundary (y=o) as shown in the inset. The value of D/u^2 is 100 years; the value of xu/D is 5 for curve A and 10 for curve B. Dots represent values obtained from the analytical solution.

APPENDIX 1

```
*************
C     ************************************************************************
C     PROGRAM SOLVES A GENERAL LINEAR PARTIAL DIFFERENTIAL
C     PARABOLIC EQUATION
C     DU/DT = F1(T,Y)*D2U/DY2 + F2(T,Y)*DU/DY+F3(T,Y)*U+F4(Y)
C     WHERE DU/DT IS PARTIAL DERIVATIVE OF U WITH RESPECT TO T
C     DU/DY IS PARTIAL DERIVATIVE OF U WITH RESPECT TO Y AND
C     D2U/DY2 IS PARTIAL SECOND DERIVATIVE OF U WITH RESPECT TO Y
C     WITH THE INITIAL AND BOUNDARY CONDITIONS
C     ************************************************************************
C     U(0,T) = G1(T)
C     U(INFINITY,T) = G2(T) , AND
C     U(Y,0) = G3(Y)
C     ************************************************************************
C     F1,F2,F3,F4 ARE SPECIFIED FUNCTIONS OF T AND Y: G1 AND G2 ARE
C     GIVEN FUNCTIONS OF T AND G3 IS FUNCTION OF Y
C     FUNCTIONS F1,F2,F3,F4,G1,G2 AND G3 GIVEN HERE ARE FOR THE CASE
C     DISCUSSED ON PAGE 7 OF THE TEXT ( SEE FIG. 1)
C     THE USER HAS TO DEFINE THESE FUNCTIONS  AS PER NEED
C     ************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4DENS
      DIMENSION Y(1000),TIMES(1000),Z(1000),DZ(1000),D2Z(1000),C(1000),
     CA(1000),B(1000),D(1000),E(1000),UU(1000),YC(100),DENS(100,200)
      COMMON NN,NH,LM,N1,N,K1,K2,K3,K4
      IP=0
      LM=0
      N=100
      N1=N+1
      NN=N-1
C     ************************************************************************
C     IF K1=1 : T=1/T VALUES WILL BE TAKEN FOR PRINTING AS WELL AS
C     PLOTTING
C     IF K1=0  : NO INVERSE TRANSFORMATION BUT LINEAR TIME VALUES WILL
C     BE TAKEN FOR PRINTING AND PLOTTING
C     IF K2=1  : LOG10 VALUES OF X AND Y WILL BE TAKEN FOR PRINTING
C     AND PLOTTING
C     IF K2=0  : X AND Y VALUES AS SUCH TAKEN (LINEAR PLOT)
C     IF K3=1  : PLOTS REQUIRED : K3=0 : NO PLOTS REQUIRED
C     IF K4 = 1: X AND Y VALUES AT INTERPOLATED 'YC' VALUES ARE PRINTED
C     ************************************************************************
      READ 200,K1,K2,K3,K4
  200 FORMAT(I3)
C     ************************************************************************
C     TIMEF=FINAL TIME
C     ************************************************************************
      READ 100, TIMEF
  100 FORMAT(F14.8)
C     ************************************************************************
C     NH=TOTAL NUMBER OF Y VALUES AT WHICH RESULTS ARE REQUIRED
C     ************************************************************************
```

```
      READ 200,NH
C     ******************************************************************
C     YS=STARTING Y(SPACE VARIABLE); YF=FINAL Y ; YC'S=VALUES OF Y'S AT
C     WHICH RESULTS REQUIRED
C     ******************************************************************
      READ 100, YS,YF,(YC(I),I=1,NH)
      IER=10
      H=1./N
      BETA =DLOG10(-DLOG(H))/DLOG10(YF)
      PRINT 300, N,BETA,TIMEF,YF
  300 FORMAT(1X,'N = ',I3,5X,'BETA = ',D14.6,5X,'FINAL TIME =',E14.6,
     C5X,'Y-FINAL = ',E14.6,//)
      R=-H
      Z(1)=1.D00
      DO 400 I=2,N
      Z(I)=Z(I-1)-H
      Y(I)=(-DLOG(Z(I)))**(1./BETA)
      DZ(I)=-BETA*(Y(I)**(BETA-1.))*Z(I)
  400 D2Z(I)=DZ(I)*((BETA-1.)/Y(I)-BETA*Y(I)**(BETA-1.))
      TIME=0.D00
      CALL INIT(C,Y,N1)
      IDT=TIMEF/1000.+0.1
      DT=IDT
  500 TIME=TIME+DT
      DO 600 I=2,N
      DZZ=DZ(I)
      D2ZZ=D2Z(I)
      YI=Y(I)
      F1TY=F1(TIME,YI)
      F2TY=F2(TIME,YI)
      A(I-1)=2./DT+2.*DZZ*DZZ/R/ R*F1TY-F3(TIME,YI)
      T1=F1TY*DZZ*DZZ/R/R
      T2 = ( F1TY*D2ZZ + F2TY*DZZ) /2./ R
      B(I-1)=-T1-T2
      E(I-1) = -T1 +T2
      F1ITY=F1(TIME-DT,YI)
      F2ITY=F2(TIME-DT,YI)
      F3ITY=F3(TIME-DT,YI)
      F4ITY=F4(TIME-DT,YI)
      D(I-1)=F1ITY*DZZ*DZZ/R/ R*(C(I+1)-2.*C(I)
     1+C(I-1))+(C(I+1)-C(I-1))/2./R*(F1ITY *D2ZZ+F2ITY*DZZ)
     2 +F3ITY*C(I)+F4ITY+F4(TIME,YI)+2./DT*C(I)
  600 CONTINUE
      D(1)=D(1)-E(1)*G1(TIME)
      D(N-1)=D(N-1)-B(N-1)*G2(TIME)
      DO 700 I=1,NN
  700 E(I)=E(I+1)
  800 CALL TRID(NN,E,B,A,D)
      C(1)=G1(TIME)
      C(N1)=G2(TIME)
      DO 900 I=2,N
```

```
  500 C(I)=E(I-1)
      IP=IP+1
      IF(IP.NE.IPR) GO TO 500
      IP=0
      LN=LN+1
      CALL INTPLN(Y,YC,C,DENS,BETA)
      TIMES(LN)=TIME
      IF(TIME.GT.TIMEF) GO TO 1000
      GO TO 500
 1000 CALL WRITE(YC,DENS,TIMES)
      STOP
      END

C
      SUBROUTINE INTPLN(Y,YC,C,DENS,BETA)
C     ************************************************************
C     THE SUBROUTINE INTERPOLATES 'DENS' VALUES AT GIVEN YC'S USING A
C     THIRD DEGREE POLYNOMIAL
C     ************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL * 4 DENS
      REAL * 4 HSS
      DIMENSION Y(1000),YC(100),C(1000),DENS(100,200)
      COMMON NM,NH,LN,N1,N
      DO 6 J=1,NH
      DO 8 I=1,N1
      IF(YC(J).EQ.0.D00) GO TO 5
      IF(YC(J).GT.Y(I)) GO TO 8
      M=I
      GO TO 7
    8 CONTINUE
      M=N1
    7 IF(M.LT.3) M=3
      IF(M.GT.N1-2) M=N1-1
      X=DEXP(-Y(M-2)**BETA)
      D1=C(M-1)-C(M-2)
      D2=C(M)-2.*C(M-1)+C(M-2)
      D3= C(M+1)-3.*C(M)+3.*C(M-1)-C(M-2)
      XX=DEXP(-YC(J)**BETA)
      UU=XX-X
      HSS=C(M-2)+UU*D1+UU*(UU-1.)/2.*D2+UU*(UU-1.)*(UU-2.)/6.*D3
      DENS(J,LN)=HSS
      GO TO 6
    5 DENS(J,LN)=C(1)
    6 CONTINUE
      RETURN
      END
C

      SUBROUTINE INIT (C,Y,N1)
```

```
C*********************************************************************
C     THE SUBROUTINE INITIALIZES THE GIVEN FUNCTION USING ITS VALUE
C     DEFINED FOR THE BOUNDARY CONDITION AT TIME=0
C*********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(1000),  Y(1000)
      DO 3 I= 1,N1
    3 C(I)=G3(Y(I))
      RETURN
      END

      SUBROUTINE TRIDI(N,C,B,A,D)
C*********************************************************************
C     THE SUBROUTINE SOLVES THE TRIDIAGONAL MATRIX BY GAUSSIAN
C     ELIMINATION METHOD
C*********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(1), B(1),A(1),D(1)
      NN=N-1
      DO10I=1,NN
      D(I)=D(I)/A(I)
      B(I)=B(I)/A(I)
      T1=D(I)
      T2=B(I)
      IF(DABS(T1).LT.1.D-30)T1=0.D0
      IF(DABS(T2).LT.1.D-30)T2=0.D0
      D(I+1)=D(I+1)-T1*C(I)
   10 A(I+1)=A(I+1)-T2*C(I)
      IF(DABS(D(N)).LT.1.D-30)D(N)=0.D0
      C(N)=D(N)/A(N)
      DO20I=1,NN
      C(N-I)=D(N-I)-B(N-I)*C(N-I+1)
      IF(DABS(C(N-I)).LT.1.D-30)C(N-I)=0.D0
   20 CONTINUE
      RETURN
      END

      SUBROUTINE WRITE(YC,CONS,TIMES)
C*********************************************************************
C     OUTPUT SUBROUTINE : IT CAN PLOT AND/OR PRINT AS SPECIFIED
C     BY THE USER THROUGH THE VALUE OF K3 IN THE MAIN PROGRAM
C*********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 CONS
      DIMENSION YC(1000),CONS(100,200),TIMES(1000)
      COMMON NN,NH,LM,N1,N,K1,K2,K3,K4
      YMAX=1.D-50
      YMIN=1.D 50
      XMIN=TIMES(1)
```

```
      YMAX=TIMES(LN)
      DC 2 J=1,NH
      DC 1 I=1,LN
      IF(DENS(J,I).GT.YMAX) YMAX=DENS(J,I)
    1 IF(DENS(J,I).LT.YMIN)YMIN=DENS(J,I)
    2 CCNTINUE
      IF(K2.EC.1) XXMAX=1./XMIN
      IF(K2.EC.1) XXMIN=1./XMAX
      IF(K2.EC.1) XMAX=XXMAX
      IF(K2.EC.1) XMIN=XXMIN
      IF(K1.EC.1) XMIN=DLCG10(XMIN)
      IF(K1.EC.1) XMAX=DLCG10(XMAX)
      IF(K1.EC.1) YMAX=DLCG10(YMAX)
      IF(K1.EC.1.AND.YMIN.EC.0.D00) GC TC 20
      IF(K1.EC.1.AND.YMIN.NE.0.D00) YMIN=DLCG10(YMIN)
   1C HX=XMAX-XMIN
      HY=YMAX-YMIN
      WRITE(6,3) XMIN,XMAX,YMIN,YMAX
    3 FCRMAT(1X,'XMIN = ',E12.6,10X,'XMAX = ',E12.6,/,1X,'YMIN = '
     C,E12.6,10X,'YMAX = ',E12.6,/,1X,'YC,S - ')
      IF(K3.NE.1) GC TC 40
      CALL PLCT(7)
      CALL PLCT(1,XMIN,XMAX,10.,HX,YMIN,YMAX,10.,HY)
      DC 5 J=1,NH
      DC 4 I=1,LN
      XX=TIMES(I)
      YY=DENS(J,I)
      IF(K2.EC.1) XX=1./TIMES(I)
      IF(K1.EC.1) XX=DLCG10(XX)
      IF(K1.EC.1) YY=DLCG10(YY)
      CALL PLCT(90,XX,YY)
    4 CCNTINUE
      CALL PLCT(99)
      WRITE(6,6) YC(J)
    6 FCRMAT(10X,E12.6)
    5 CCNTINUE
      IF(K4.NE.1) GC TC 11
   4C WRITE(6,25) (YC(I),I=1,NH)
   25 FCRMAT(1X,'TIME PARAMETER',T20,'VALUE CF FUNCTION AT INTERPOLATED
     C YC,S',//,1X,T19, 8(1X,E12.6))
      DC 3C I=1,LN
      IF(K2.EC.1) TIMES(I)=1./TIMES(I)
      IF(K1.EC.1) TIMES(I)=DLCG10(TIMES(I))
      DC 45 J=1,NH
   45 IF(K1.EC.1)  DENS(J,I)=DLCG10(DENS(J,I))
      WRITE(6,35) TIMES(I),(DENS(J,I),J=1,NH)
   35 FCRMAT(1X,E12.6,T19,8(1X,E12.6))
   3C CCNTINUE
      GC TC 11
   2C WRITE(6,15)
   15 FCRMAT(1X,'DLCG10(YMIN) CANNCT BE TAKEN SINCE ARGUMENT = 0.0')
```

```
11 RETURN
   END

   FUNCTION F1(TIME,YI)
   IMPLICIT REAL*8(A-H,O-Z)
   F1=1.D-02
   RETURN
   END

   FUNCTION F2(TIME,YI)
   IMPLICIT REAL*8(A-H,O-Z)
   F2=-1.D-02
   RETURN
   END

   FUNCTION F3(TIME,YI)
   IMPLICIT REAL*8(A-H,O-Z)
   F3=0.D00
   RETURN
   END

   FUNCTION F4(TIME,YI)
   IMPLICIT REAL*8(A-H,O-Z)
   F4=0.D00
   RETURN
   END

   FUNCTION G1(TIME)
   IMPLICIT REAL*8(A-H,O-Z)
   G1 = DSIN( 2* 3.14159265  / 1000. * TIME )
   RETURN
   END

   FUNCTION G2(TIME)
   IMPLICIT REAL*8(A-H,O-Z)
   G2=0.D00
   RETURN
   END

   FUNCTION G3(TIME)
   IMPLICIT REAL*8(A-H,O-Z)
   G3=0.D00
   RETURN
   END
```

APPENDIX 2

The solution of

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - u\frac{\partial c}{\partial x} \qquad (1)$$

subject to the initial and boundary conditions

$$C(x, 0) = 0; \quad C(0, t) = C_o \operatorname{Sin} \omega t; \quad C(\infty, t) = 0 \qquad (2)$$

is given by

$$C(x, t) = \frac{C_o}{2} \exp\left\{\frac{ux}{2D} - \mu t\right\}\Big[\exp(\varphi_1) \sin\varphi_2$$

$$\{1 - \operatorname{Re}\{\operatorname{erf}(z_+)\} + \exp(\varphi_3) \sin\varphi_4\{1 - \operatorname{Re}\{\operatorname{erf}(z_-)\}\}$$

$$-\{\exp(\varphi_1) \cos\varphi_2 \operatorname{Im}\{\operatorname{erf}(z_+)\} + \exp(\varphi_3) \cos\varphi_4$$

$$\operatorname{Im}\{\operatorname{erf}(z_-)\}\}\Big] \qquad (3)$$

where

$$\left.\begin{aligned}
\varphi_1 &= tr\cos\theta + x(r/D)^{1/2}\cos\theta/2 \\
\varphi_2 &= tr\sin\theta + x(r/D)^{1/2}\sin\theta/2 \\
\varphi_3 &= tr\cos\theta - x(r/D)^{1/2}\cos\theta/2 \\
\varphi_4 &= tr\sin\theta - x(r/D)^{1/2}\sin\theta/2
\end{aligned}\right\} \qquad (4)$$

$$r = (\mu^2 + \omega^2)^{1/2}; \quad \tan\theta = \omega/\mu; \quad \mu = u^2/4D$$

Infinite series approximation for complex error function is

$$\mathrm{erf}\,(x+iy) = \mathrm{erf}\,(x) + \frac{e^{-x^2}}{2\pi x}\,[(1-\cos 2xy)$$

$$+ i \sin 2xy] + \frac{2}{\pi} e^{-x^2} \sum_{n=1}^{\infty} \frac{e^{-n^2/4}}{n^2+4x^2} [f_n(x,y)$$

$$+ i\, g_n(x,y)] + \varepsilon(x,y)$$

where

$$f_n(x,y) = 2x - 2x \cosh ny \cos 2xy + n \sinh ny \sin 2xy$$

$$g_n(x,y) = 2x \cosh ny \sin 2xy + n \sinh ny \cos 2xy$$

and $\varepsilon(x,y) \simeq 10^{-16} \,|\, \mathrm{erf}\,(x+iy)\,|$

where $z_{\pm}$ are defined as

$$\left.\begin{aligned} z_{\pm} &= x/2\,(Dt)^{1/2} \pm (rt)^{1/2} \cos\theta/2 \pm iy \\ \text{and } y &= (rt)^{1/2} \sin\theta/2 \end{aligned}\right\} \qquad (5)$$

Various terms appearing in equations (3) - (5) may be expressed in terms of a dimensionless distance parameter, (xu/D), and the dispersion time constant, (D/u^2). For example,

(i) $ux/2D - \mu t$ appearing in eqn.(3) may be written as
$ux/2D - t/(4D/u^2)$

(ii) $\varphi_1 = tr\cos\theta + x\,(r/D)^{1/2} \cos\theta/2$

$$= t\mu(1+\omega^2/\mu^2)^{1/2} + (xu/2D)\,(1+\omega^2/\mu^2)^{1/4}$$

and similar expressions for φ_2, φ_3 and φ_4.

(iii) $z_+ = (xu/2D)\ (tD/u^2)^{-1/2} + (t/(2D/u^2))^{1/2}$

$$\{1 + \omega^2/(4D/u^2)^2\}\ \cos\ \theta/2$$

$$+ i\{(t/(4D/u^2))^{1/2}\ \{1 + \omega^2/(4D/u^2)^2\}^{1/4}\ \sin\ \theta/2.$$