Faculty of Civil Engineering
Belgrade
Master Study
COMPUTATIONAL ENGINEERING

Faculty of Civil Engineering and Architecture
Niš
Doctoral Study

LECTURES

LESSON XI

11. Partial Differential Equations - PDE

11.1. Introduction

Partial differential equations (PDEs) arise in all fields of engineering and science. Most real physical processes are governed by partial differential equations. In many cases, simplifying approximations are made to reduce the governing PDEs to ordinary differential equations (ODEs) or even algebraic equations. However, because of the ever increasing requirements for more accurate modelling of physical processes, engineers and scientists are more and more required to solve the actual PDEs that govern the physical problem being investigated. Physical problems are governed by many different PDEs. A few problems are governed by a single first-order PDE. Numerous problems are governed by a system of first order PDEs. Some problems are governed by a single second-order PDE, and numerous problems are governed by a system of second-order PDEs. A few problems are governed by fourth order PDEs. The two most frequent types of physical problems described by PDEs are equilibrium and propagation problems.

The classification of PDEs is most easily explained for a single second order linear PDE of form

$$A\frac{\partial^2 u}{\partial x^2}+B\frac{\partial^2 u}{\partial x\partial y}+C\frac{\partial^2 u}{\partial y^2}+D\frac{\partial u}{\partial x}+E\frac{\partial u}{\partial y}+Fu=G, \tag{11.1.1}$$

where A, B, C, D, E, F, G are given functions which are continuous in area S of plane xOy. The area S is usually defined as inside part of some curve Γ. Of course, the area S can be as finite as well as infinite. Typical problem is finding two times continuous differentiable solution $(x, y) \to u(x, y)$ which satisfies equation (11.1.1) and some conditions on curve (contour) Γ.

Linear PDEs of second order can be classified as eliptic, parabolic and hyperbolic, depending on the sign of the discriminant $B^2 - 4AC$ in given area S, as follows:

1^0 $B^2 - 4AC < 0$ Elliptic
2^0 $B^2 - 4AC = 0$ Parabolic
3^0 $B^2 - 4AC < 0$ Hyperbolic

The terminology elliptic, parabolic, and hyperbolic chosen to classify PDEs reflects the analogy between the form of the discriminant, $B^2 - 4AC$, for PDEs and the form of the discriminant, $B^2 - 4AC$, which classifies conic sections, described by the general second-order algebraic equation

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0,$$

where we for negative, zero, and positive value of discriminant have ellipse, parabola, and hyperbola, respectively. It is easy to check that the Laplace equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0, \tag{11.1.2}$$

is of elliptic type, heat conduction equation

$$\frac{\partial u}{\partial t} - a^2 \frac{\partial^2 u}{\partial x^2} = 0, \tag{11.1.3}$$

is of parabolic type, and wave equation

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2 \partial y} = 0 \tag{11.1.4}$$

of hyperbolic type. In this chapter we will show one way for numerical solution of PDEs, for Laplace and wave equation by grid method. In the similar way can be solve heat conduction equation, what we leave to the reader.

11.2. Grid method

Grid method or difference method, or finite-difference grid method, is basic method for solution of equations of mathematical physics (partial equations which appear in physics and science)

Let be given linear PDE

$$\mathbf{L}u = f \tag{11.2.1}$$

and let in area $\mathbf{D}$, which is bounded by curve $\Gamma(\mathbf{D} = int\ \Gamma)$, look for such its solution on curve Γ that satisfies given boundary condition

$$\mathbf{K}u = \Psi \quad ((x, y) \in \Gamma). \tag{11.2.2}$$

In application of grid method, at first, one should chose discrete set of points D_h, which belongs to area $\overline{D}(= D \cup \Gamma)$, called grid. Most frequently, in applications is for grid taken family of parallel straight lines $x_i = x_0 + ih,\ y_j = y_0 + jl\ (i, j = 0, \pm 1, \pm 2, \ldots)$. Intersection points of these families are called nodes of grid, and h and l are steps of grid. Two nodes of grid are called neighbored if the distance between them along x and y axes is one step only. If all four neighbor nodes of some node belong to area $\overline{D}$, then this node is called interior or inner; in counterpart node of grid D_h is called boundary node. In addition to rectangular grids, in practice are also used other grid shapes.

Grid method consists of approximation of equations (11.2.1) and (11.2.2) using corresponding difference equations. Namely, we can approximate operator $\mathbf{L}$ by difference operator very simple, by substituting derivative with corresponding differences in inner nodes of grid. Thereby are used the following formulas

$$\begin{aligned}
\frac{\partial u(x_i, y_j)}{\partial x} &\cong \frac{u_{i+1,j} - u_{i,j}}{h} \\
\frac{\partial u(x_i, y_j)}{\partial y} &\cong \frac{u_{i+1,j} - u_{i-1,j}}{2h} \\
\frac{\partial^2 u(x_i, y_j)}{\partial x^2} &\cong \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \quad \text{etc.}
\end{aligned}$$

Formulas for partial derivatives in variable y are absolutely symmetric. Approximation of contour conditions can be in some cases very complicated problem, what depends on form of operator K and contour Γ. At so known contour conditions of first kind, where $Ku = u$, one practical way for approximation was proposed by L. Collatz and comprises of the following:

Let the closest point from contour Γ to boundary node A be point B and let their distance be δ (see Fig. 11.2.1).

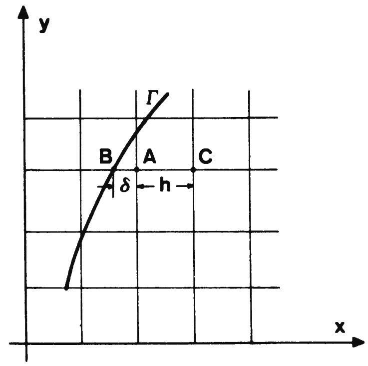

Figure 11.2.1

Based on function values in points B and C, we get by linear interpolation

$$u(A) \cong \frac{h\Psi(B) + \delta u(C)}{h+\delta}.$$

Approximation of boundary condition (11.2.2) in this case comprises of defining equations of above form for every boundary node.

The equations obtained by approximation of equation (11.2.1) and boundary condition (11.2.2) form system of linear equations, by which solution are obtained numerical solutions of given problem.

In further consideration we will give two basic examples.

11.3. Laplace equation

Let it be needfully to find solution of Laplace equation

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad ((x,y) \in D),$$

which on the contour of square $D = \{(x,y) | 0 < x < 1,\ 0 < y < 1\}$ fulfills given condition $u(x,y) = \Psi(x,y)$ $((x,y) \in \Gamma)$. Let's chose the grid in D_h at which is $l = h = \dfrac{1}{N-1}$, so that grid nods are points $(x_i, y_i) = ((i-1)h,\ (j-1)l)$ $(i,j = 1,\ldots,N)$. The standard difference approximation scheme for solving Laplace equation is of form

$$\frac{1}{h^2} u_{i+1,j} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j} = 0,$$

or

$$u_{i,j} = \frac{1}{4} u_{i,j+1} + u_{i,j-1} + u_{i-1,j} + u_{i+1,j}).$$

Taking $i,j = 2,\ldots,N-1$ in last equality we get the system of $(N-2)^2$ linear equations. For solving of this system usually is used method of simple iterations, or, even more simpler, Gauss-Seidel method.

The corresponding program for solving of problem in consideration is of form

```
C==================================================
C  RESAVANJE LAPLACE-OVE JEDNACINA
C==================================================
      DIMENSION U(25,25)
      OPEN(8,FILE='LAPLACE.IN')
```

```
      OPEN(5,FILE='LAPLACE.OUT')
      READ(8,4)N
 4    FORMAT(I2)
      M=N-1
      READ(8,1)(U(1,J),J=1,N),(U(N,J),J=1,N),
     1(U(I,1),I=2,M),(U(I,N),I=2,M)
 1    FORMAT(8F10.0)
      DO 10 I=2,M
      DO 10 J=2,M
 10   U(I,J)=0.
      IMAX=0
 20   WRITE(*,5)
  5   FORMAT(5X,'UNETI MAKSIMALNI BROJ ITERACIJA'/
     110X, '(ZA MAX=0 => KRAJ)')
      READ(*,4)MAX
      IF(MAX.EQ.0) GOTO 100
      DO 30 ITER=1,MAX
      DO 30 I=2,M
      DO 30 J=2,M
 30   U(I,J)=(U(I,J+1)+U(I,J-1)+U(I-1,J)+U(I+1,J))/4.
      IMAX=IMAX+MAX
      WRITE(5,65) IMAX,(J,J=1,N)
 65   FORMAT(//26X,'BROJ ITERACIJA JE',I3//17X,
     14(5X,'J=',I2))
      DO 60 I=1,N
 60   WRITE(5,66) I,(U(I,J),J=1,N)
 66   FORMAT(13X,'I =',I2,6F10.4)
      GO TO 20
100   CLOSE(8)
      CLOSE(5)
      STOP
      END
```

For solving system of linear equations we used Gauss-Seidel method with initial conditions $u_{i,j} = 0$ $(i, j = 2, \ldots, N-1)$, whereby one can control number of iterations on input. For `N=4` and boundary conditions

$$u_{11} = 0,\ u_{1,2} = 30,\ u_{13} = 60,\ u_{1,4} = 90,$$
$$u_{41} = 180,\ u_{4,2} = 120,\ u_{43} = 60,\ u_{4,4} = 0,$$
$$u_{21} = 60,\ u_{3,1} = 120,\ u_{24} = 60,\ u_{3,4} = 30,$$

the following results are obtained:

```
              BROJ ITERACIJA JE  2
          J= 1      J= 2      J= 3      J= 4
  I = 1     .0000   30.0000   60.0000   90.0000
  I = 2   60.0000   47.8125   53.9063   60.0000
  I = 3  120.0000   83.9063   56.9531   30.0000
  I = 4  180.0000  120.0000   60.0000     .0000
              BROJ ITERACIJA JE  7
          J= 1      J= 2      J= 3      J= 4
  I = 1     .0000   30.0000   60.0000   90.0000
  I = 2   60.0000   59.9881   59.9940   60.0000
  I = 3  120.0000   89.9940   59.9970   30.0000
  I = 4  180.0000  120.0000   60.0000     .0000
              BROJ ITERACIJA JE  9
          J= 1      J= 2      J= 3      J= 4
  I = 1     .0000   30.0000   60.0000   90.0000
  I = 2   60.0000   59.9993   59.9996   60.0000
  I = 3  120.0000   89.9996   59.9998   30.0000
  I = 4  180.0000  120.0000   60.0000     .0000
              BROJ ITERACIJA JE 10
          J= 1      J= 2      J= 3      J= 4
  I = 1     .0000   30.0000   60.0000   90.0000
  I = 2   60.0000   59.9998   59.9999   60.0000
  I = 3  120.0000   89.9999   60.0000   30.0000
  I = 4  180.0000  120.0000   60.0000     .0000
              BROJ ITERACIJA JE 21
```

	J= 1	J= 2	J= 3	J= 4
I = 1	.0000	30.0000	60.0000	90.0000
I = 2	60.0000	60.0000	60.0000	60.0000
I = 3	120.0000	90.0000	60.0000	30.0000
I = 4	180.0000	120.0000	60.0000	.0000

11.4. Wave equation

Consider wave equation

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{a^2}\cdot\frac{\partial^2 u}{\partial x^2} \tag{11.4.1}$$

with initial conditions

$$u(x,0) = f(x),\ u_1(x,0) = g(x) \qquad (0 < x < h) \tag{11.4.2}$$

and boundary conditions

$$u(0,t) = \Phi(t),\ u(b,t) = \Psi(t) \qquad (t \geq 0). \tag{11.4.3}$$

Using finite differences, the equation (11.4.1) can be approximated by

$$u_{i+1,j} - 2u_{i,j} + u_{i-1,j} = \frac{1}{r^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}), \tag{11.4.4}$$

where $r = a\frac{1}{h}$ (h and l are steps along x and t axes respectively, and $u_{i,j} \cong u(x_i, t_j)$. Based on first equality in (11.4.2) we have

$$u_{i,0} = f(x_i) = f_i. \tag{11.4.5}$$

By introducing fictive layer $j = -1$, second initial condition in (11.4.2) can simple be approximated using

$$u_i(x_i, 0) = g(x_i) = g_i \cong \frac{u_{i,1} - u_{i,-1}}{2l}. \tag{11.4.6}$$

If we put in (11.4.4) $j = 0$ we get

$$f_{i+1} - 2f_i + f_{i-1} - \frac{1}{r^2}(u_{i,1} - 2f_i + u_{i,-1}) = 0,$$

wherefrom, in regard to (11.4.6) it follows

$$u_{i,1} = lg_i + f_i + \frac{1}{2}r^2(f_{i+1} - 2f_i + f_{i-1}),$$

i.e.

$$u_{i,1} = lg_i + (1 - r^2)f_i + \frac{1}{2}r^2(f_{i+1} + f_{i-1}). \tag{11.4.7}$$

On the other hand, from (11.4.4) it follows

$$u_{i,j+1} = \frac{1}{r^2}(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1} + 2(\frac{1}{r^2} - 1)u_{i,j}. \tag{11.4.8}$$

If we put $h = b/N$ and $x_i = (i-1)h$ $(i = 1, 2, \ldots, N+1)$, due to boundary conditions (11.4.3) we have

$$u_{1,j} = \Phi_j,\ u_{N+1,j} = \Psi(t_j) = \Psi_j, \tag{11.4.9}$$

where $j = 0, 1, \ldots$. For determining of solution inside of rectangle $P = \{(x,t)|0 < x < b, 0 < t < T_{max}\}$, maximal value of index j is integer part of T_{max}/l i.e. $j_{max} = M = [T_{max}/l]$.

Based on equalities (11.4.5), (11.4.7), (11.4.8), (11.4.9) the approximate solutions of given problem in grid nodes of rectangle P, are simple to obtain. This algorithm is coded in the following program.

```
C==================================================
C RESAVANJE PARCIJALNE DIF. JED. HIPERBOLICNOG TIPA
C==================================================
      DIMENSION U(3,9)
      OPEN(8,FILE='TALAS.IN')
      OPEN(5,FILE='TALAS.OUT')
      READ (8,5)N,A,B,R,TMAX
 5    FORMAT(I2,4F5.2)
      N1=N+1
      WRITE (5,10) (I,I=1,N1)
 10   FORMAT(10X,1HJ,<N+1>(4X,'U(',I1,',J)')/)
      H=B/FLOAT(N)
      EL=R*H/A
      M=TMAX/EL
      T=0.
      DO 15 K=1,2
      U(K,1)=FF(T,B,3)
      U(K,N1)=FF(T,B,4)
 15   T=T+EL
      X=0.
      R2=R*R
      DO 20 I=2,N
      X=X+H
      U(1,I)=FF(X,B,1)
 20   U(2,I)=EL*FF(X,B,2)+(1.-R)*U(1,I)
      DO 25 I=2,N
 25   U(2,I)=U(2,I)+R2/2.*(U(1,I+1)+U(1,I-1))
      J=0
 30   WRITE(5,35)J,(U(1,I),I=1,N1)
 35   FORMAT(7X,I5,<N1>F10.4)
      IF(J.EQ.M)GO TO 50
      J=J+1
      U(3,1)=FF(T,B,3)
      U(3,N1)=FF(T,B,4)
      DO 40 I=2,N
 40   U(3,I)=(U(2,I+1)+U(2,I-1))/R2-U(1,I)-2.
     1*(1./R2-1.)*U(2,I)
      T=T+EL
      DO 45 I=1,N1
      U(1,I)=U(2,I)
 45   U(2,I)=U(3,I)
      GO TO 30
 50   CLOSE(5)
      CLOSE(5)
      STOP
      END
```

Note that the values of solution in three successive layers $j-1, j, j+1$, are stored in first, second, and third row of matrix U, respectively. .

Functions f, g, Φ, Ψ are defined by function subroutine `FF` for `I=1,2,3,4`, respectively.

In considered case for $a = 2$, $b = 4$, $T_{max} = 6$, $f(x) = x(4-x)$, $g(x) = 0$, $\Phi(t) = 0$, $\Psi(t) = 0$, $N = 4$, and $r = 1$, subroutine `FF` and corresponding output listing with result have the following form:

```
      FUNCTION FF(X,B,I)
      GO TO(10,20,30,40),I
 10   FF=X*(B-X)
      RETURN
 20   FF=0.
      RETURN
```

```
30    FF=0.
      RETURN
40    FF=0.
      RETURN
      END
```

```
J    U(1,J)    U(2,J)    U(3,J)    U(4,J)    U(5,J)
 0     .0000    3.0000    4.0000    3.0000     .0000
 1     .0000    2.0000    3.0000    2.0000     .0000
 2     .0000     .0000     .0000     .0000     .0000
 3     .0000   -2.0000   -3.0000   -2.0000     .0000
 4     .0000   -3.0000   -4.0000   -3.0000     .0000
 5     .0000   -2.0000   -3.0000   -2.0000     .0000
 6     .0000     .0000     .0000     .0000     .0000
 7     .0000    2.0000    3.0000    2.0000     .0000
 8     .0000    3.0000    4.0000    3.0000     .0000
 9     .0000    2.0000    3.0000    2.0000     .0000
10     .0000     .0000     .0000     .0000     .0000
11     .0000   -2.0000   -3.0000   -2.0000     .0000
12     .0000   -3.0000   -4.0000   -3.0000     .0000
```

11.5. Packages for PDEs

Elliptic PDEs govern equilibrium problem, which have no preferred paths of information propagation. The domain of dependence and range of influence of every point is the entire closed solution domain. Such problems are solved numerically by relaxation methods. Finite difference methods, as typified by five-point method, yield a system of finite difference equations, called the system equations, which have to be solved by relaxation methods. The successive-over-relaxation method (SOR)method is generally method of choice. The multigrid method (Brandt, 1977) shows the best potential for rapid convergence. Nonlinear PDEs yield nonlinear finite difference equations (FDE). System of nonlinear FDEs can be very difficult to solve. The multigrid method can be applied directly to nonlinear PDEs. Three-dimensional PDEs are approximated simply by including the finite difference approximations of the spatial derivatives in the third direction. The relaxation techniques used to solve two-dimensional problems generally can be used to solve three-dimensional problems, at the expense of a considerable increase of computational time.

Parabolic PDEs govern propagation problems which have an infinite physical information propagation speed. They are usually solved numerically by marching method. Explicit finite difference methods, like FTCS (Forward-Time Centered-Space method, see [3], pp. 633-635) are conditionally stable and require relatively small step size in the marching direction to satisfy the stability criteria. Implicit methods, like BTCS (Backward-Time Centered-Space method, see [3], pp. 635-637) are unconditionally stable. The marching step size is restricted by accuracy requirements, not stability requirements. For accurate solution of transient problems, the marching step-size for implicit methods cannot be very much larger than the stable step size for explicit methods. Consequently, explicit methods are generally preferred for obtaining accurate transient solutions. Asymptotic steady state solutions can be obtained very efficiently by BTCS method with a large marching step size. Nonlinear PDEs can be solved directly by explicit methods. When solved by implicit methods, system of nonlinear FDEs must be solved. Multidimensional problems can be solved directly by explicit methods. When solved by implicit methods, large banded systems of FDEs result.

Hyperbolic PDEs govern propagation problems, which have a finite physical information propagation speed. They are solved numerically by marching method. Explicit finite difference methods are conditionally stable and require a relatively small step size in marching direction to satisfy the stability criteria. Implicit methods, as typified by the BTCS method, are unconditionally stable. The marching step size is restricted by accuracy requirements, not stability requirements. For accurate solution of transient problems, explicit methods are recommended. When steady state solutions are to be obtained as the asymptotic solution in time of an appropriate unsteady propagation problem, BTCS with a large step size is recommended.

Nonlinear PDEs can be solved directly by explicit methods. When solved by implicit methods, system of nonlinear FDEs must be solved. Multidimensional problems can be solved directly by explicit methods. When solved by implicit methods, large banded systems of FDEs result.

Numerous libraries and software packages are available for integrating the Laplace and Poisson equations, diffusion type (i.e. parabolic) and convection type (i.e. hyperbolic) PDEs. Many work stations and main frame computers have such libraries attached to their operating systems.

Many commercial software packages contain routines for integrating Laplace and Poisson equations. Due to the wide variety of elliptic, parabolic, and hyperbolic PDEs governing physical problems, many PDE solvers (programs) have been developed.

The book *Numerical Recipes* ([7]) contains a lot of algorithms for integrating PDEs. For some of them is given programming code in `Fortran` (available also in `C`). Survey of methods for solving different classes of PDEs accompanied with algorithms, from which some are codded, is given in book *Numerical Methods for Engineers and Scientists* ([3], Chapter 9, 10 and 11).

Bibliography (Cited references and further reading)

[1] Milovanović, G.V., *Numerical Analysis III*, Naučna knjiga, Beograd, 1988 (Serbian).
[2] Milovanović, G.V., *Numerical Analysis I*, Naučna knjiga, Beograd, 1988 (Serbian).
[3] Hoffman, J.D., *Numerical Methods for Engineers and Scientists.* Taylor & Francis, Boca Raton-London-New York-Singapore, 2001.
[4] Milovanović, G.V. and Djordjević, Dj.R., *Programiranje numeričkih metoda na FORTRAN jeziku.* Institut za dokumentaciju zaštite na radu "Edvard Kardelj", Niš, 1981 (Serbian).
[5] Milovanović, G.V., *Numerical Analysis II*, Naučna knjiga, Beograd, 1988 (Serbian).
[6] Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis*, Springer, New York, 1980.
[7] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recepies - The Art of Scientific Computing.* Cambridge University Press, 1989.
[8] Milovanović, G.V. and Kovačević, M.A., *Zbirka rešenih zadataka iz numeričke analize.* Naučna knjiga, Beograd, 1985. (Serbian).
[9] Ralston,A., *A First Course in Numerical Analysis.* McGraw-Hill, New York, 1965.
[10] Hildebrand, F.B., *Introduction to Numerical Analysis.* Mc.Graw-Hill, New York, 1974.
[11] Acton, F.S., *Numerical Methods That Work* (corrected edition). Mathematical Association of America, Washington, D.C., 1990.
[12] Abramowitz, M., and Stegun, I.A., *Handbook of Mathematical Functions.* National Bureau of Standards, Applied Mathematics Series, Washington, 1964 (reprinted 1968 by Dover Publications, New York).
[13] Rice, J.R., *Numerical Methods, Software, and Analysis.* McGraw-Hill, New York, 1983.
[14] Forsythe, G.E., Malcolm, M.A., and Moler, C.B., *Computer Methods for Mathematical Computations.* Englewood Cliffs, Prentice-Hall, NJ, 1977.

[15] Kahaner, D., Moler, C., and Nash, S., 1989, *Numerical Methods and Software.* Englewood Cliffs, Prentice Hall, NJ, 1989.

[16] Hamming, R.W., *Numerical Methods for Engineers and Scientists.* Dover, New York, 1962 (reprinted 1986).

[17] Ferziger, J.H., *Numerical Methods for Engineering Applications.* Stanford University, John Willey & Sons, Inc., New York, 1998

[18] Pearson, C.E., *Numerical Methods in Engineering and Science.* University of Washington, Van Nostrand Reinhold Company, New York, 1986.

[19] Stephenson, G. and Radmore, P.M., *Advanced Mathematical Methods for Engineering and Science Students.* Imperial College London, University College, London Cambridge Univ. Press, 1999

[20] Ames, W.F., *Numerical Methods for Partial Differential Equations.* 2nd ed. Academic Press, New York, 1977.

[21] Richtmyer, R.D. and Morton, K.W., *Difference Methods for Initial Value Problems.* 2nd ed., Wiley-Interscience, New York, 1967.

[22] Mitchell, A.R., and Griffiths, D.F. , *The Finite Difference Method in Partial Differential Equations.*, Wiley, New York, 1980.

[23] *IMSL Math/Library Users Manual* , IMSL Inc., 2500 City West Boulevard, Houston TX 77042

[24] *NAG Fortran Library*, Numerical Algorithms Group, 256 Banbury Road, Oxford OX27DE, U.K., Chapter F02.