NAG Toolbox for MATLAB

Chapter Introduction

D03 – Partial Differential Equations

Contents

1 Scope of the Chapter

This chapter is concerned with the numerical solution of partial differential equations.

2 Background to the Problems

The definition of a partial differential equation problem includes not only the equation itself but also the domain of interest and appropriate subsidiary conditions. Indeed, partial differential equations are usually classified as elliptic, hyperbolic or parabolic according to the form of the equation **and** the form of the subsidiary conditions which must be assigned to produce a well-posed problem. The functions in this chapter will often call upon functions from other chapters, such as Chapter F04 (Simultaneous Linear Equations) and Chapter D02 (Ordinary Differential Equations). Other chapters also contain relevant functions, in particular Chapter D06 (Mesh Generation) and Chapter F11 (Large Scale Linear Systems).

The classification of partial differential equations is easily described in the case of **linear** equations of the **second order** in two independent variables, i.e., equations of the form

$$au_{xx} + 2bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0, \tag{1}$$

where a, b, c, d, e, f and g are functions of x and y only. Equation (1) is called elliptic, hyperbolic or parabolic according to whether $ac - b^2$ is positive, negative or zero, respectively. Useful definitions of the concepts of elliptic, hyperbolic and parabolic character can also be given for differential equations in more than two independent variables, for systems and for nonlinear differential equations.

For **elliptic** equations, of which Laplace's equation

$$u_{xx} + u_{yy} = 0 \tag{2}$$

is the simplest example of second order, the subsidiary conditions take the form of **boundary** conditions, i.e., conditions which provide information about the solution at all points of a **closed** boundary. For example, if equation (2) holds in a plane domain D bounded by a contour C, a solution u may be sought subject to the condition

$$u = f \qquad \text{on} \qquad C, \tag{3}$$

where f is a given function. The condition (3) is known as a Dirichlet boundary condition. Equally common is the Neumann boundary condition

$$u' = g \qquad \text{on} \qquad C, \tag{4}$$

which is one form of a more general condition

$$u' + fu = g \qquad \text{on} \qquad C, \tag{5}$$

where u' denotes the derivative of u normal to the contour C, and f and g are given functions. Provided that f and g satisfy certain restrictions, condition (5) yields a well-posed **boundary-value problem** for Laplace's equation. In the case of the Neumann problem, one further piece of information, e.g., the value of u at a particular point, is necessary for uniqueness of the solution. Boundary conditions similar to the above are applicable to more general second-order elliptic equations, whilst two such conditions are required for equations of fourth order.

For **hyperbolic** equations, the wave equation

$$u_{tt} - u_{xx} = 0 \tag{6}$$

is the simplest example of second order. It is equivalent to a first-order system

$$u_t - v_x = 0, \qquad v_t - u_x = 0. \tag{7}$$

The subsidiary conditions may take the form of **initial** conditions, i.e., conditions which provide information about the solution at points on a suitable **open** boundary. For example, if equation (6) is satisfied for $t > 0$, a solution u may be sought such that

$$u(x, 0) = f(x), \qquad u_t(x, 0) = g(x), \tag{8}$$

where f and g are given functions. This is an example of an **initial value problem**, sometimes known as Cauchy's problem.

For **parabolic** equations, of which the heat conduction equation

$$u_t - u_{xx} = 0 \tag{9}$$

is the simplest example, the subsidiary conditions always include some of **initial** type and may also include some of **boundary** type. For example, if equation (9) is satisfied for $t > 0$ and $0 < x < 1$, a solution u may be sought such that

$$u(x,0) = f(x), \qquad 0 < x < 1, \tag{10}$$

and

$$u(0,t) = 0, \qquad u(1,t) = 1, \qquad t > 0. \tag{11}$$

This is an example of a mixed **initial/boundary-value problem**.

For all types of partial differential equations, finite difference methods (see Mitchell and Griffiths 1980) and finite element methods (see Wait and Mitchell 1985) are the most common means of solution and such methods obviously feature prominently either in this chapter or in the companion NAG Finite Element Library. Some of the utility functions in this chapter are concerned with the solution of the large sparse systems of equations which arise from finite difference and finite element methods. Further functions for this purpose are provided in Chapter F11.

Alternative methods of solution are often suitable for special classes of problems. For example, the method of characteristics is the most common for hyperbolic equations involving time and one space dimension (see Smith 1985). The method of lines (see Mikhlin and Smolitsky 1967) may be used to reduce a parabolic equation to a (stiff) system of ordinary differential equations, which may be solved by means of functions from Chapter D02 (Ordinary Differential Equations). Similarly, integral equation or boundary element methods (see Jaswon and Symm 1977) are frequently used for elliptic equations. Typically, in the latter case, the solution of a boundary-value problem is represented in terms of certain boundary functions by an integral expression which satisfies the differential equation throughout the relevant domain. The boundary functions are obtained by applying the given boundary conditions to this representation. Implementation of this method necessitates discretization of only the boundary of the domain, the dimensionality of the problem thus being effectively reduced by one. The boundary conditions yield a full system of simultaneous equations, as opposed to the sparse systems yielded by finite difference and finite element methods, but the full system is usually of much lower order. Solution of this system yields the boundary functions, from which the solution of the problem may be obtained, by quadrature, as and where required.

3 Recommendations on Choice and Use of Available Functions

The choice of function will depend first of all upon the type of partial differential equation to be solved. At present no special allowances are made for problems with boundary singularities such as may arise at corners of domains or at points where boundary conditions change. For such problems results should be treated with caution. The choice of function may also depend on whether or not it is to be used in a multithreaded environment.

You may wish to construct your own partial differential equation solution software for problems not solvable by the functions described in Section 3.1 to Section 3.7 below. In such cases you can employ appropriate functions from the Linear Algebra Chapters to solve the resulting linear systems; see Section 3.9 for further details.

3.1 Elliptic Equations

The function d03ea solves Laplace's equation in two dimensions, equation (2), by an integral equation method. This function is applicable to an arbitrary domain bounded internally or externally by one or more closed contours, when the value of either the unknown function u or its normal derivative u' is given at each point of the boundary.

The functions d03eb and d03ec solve a system of simultaneous algebraic equations of five-point and seven-point molecule form (see Mikhlin and Smolitsky 1967) on two-dimensional and three-dimensional topologically-rectangular meshes respectively, using Stone's Strongly Implicit Procedure (SIP). These

functions, which make repeated calls of the utility functions d03ua and d03ub respectively, may be used to solve any boundary-value problem whose finite difference representation takes the appropriate form.

The function d03ed solves a system of seven-point difference equations in a rectangular grid (in two dimensions), using the multigrid iterative method. The equations are supplied by you, and the seven-point form allows cross-derivative terms to be represented (see Mitchell and Griffiths 1980). The method is particularly efficient for large systems of equations with diagonal dominance and should be preferred to d03eb whenever it is appropriate for the solution of the problem.

The function d03ee discretizes a second-order equation on a two-dimensional rectangular region using finite differences and a seven-point molecule. The function allows for cross-derivative terms, Dirichlet, Neumann or mixed boundary conditions, and either central or upwind differences. The resulting seven-diagonal difference equations are in a form suitable for passing directly to the multigrid function d03ed, although other solution methods could just as easily be used.

The function d03fa, based on the function HW3CRT from FISHPACK (see Swarztrauber and Sweet 1979), solves the Helmholtz equation in a three-dimensional cuboidal region, with any combination of Dirichlet, Neumann or periodic boundary conditions. The method used is based on the fast Fourier transform algorithm, and is likely to be particularly efficient on vector-processing machines.

3.2 Hyperbolic Equations

See Section 3.7.

3.3 Parabolic Equations

There are five functions available for solving general parabolic equations in one space dimension:

d03pc,

d03pd,

d03ph,

d03pj,

d03pp.

Equations may include nonlinear terms but the true derivative u_t should occur linearly and equations should usually contain a second-order space derivative u_{xx}. There are certain restrictions on the coefficients to try to ensure that the problems posed can be solved by the above functions.

The method of solution is to discretize the space derivatives using finite differences or collocation, and to solve the resulting system of ordinary differential equations using a 'stiff' solver.

d03pc and d03pd can solve a system of parabolic equations of the form

$$\sum_{j=1}^{n} P_{ij}(x,t,U,U_x)\frac{\partial U_j}{\partial t} + Q_i(x,t,U,U_x) = x^{-m}\frac{\partial}{\partial x}(x^m R_i(x,t,U,U_x)),$$

where $i = 1,2,\ldots,n$, $a \leq x \leq b$, $t \geq t_0$.

The parameter m allows the function to handle different co-ordinate systems easily (Cartesian, cylindrical polars and spherical polars). d03pc uses a finite differences spatial discretization and d03pd uses a collocation spatial discretization.

d03ph and d03pj are similar to d03pc and d03pd respectively, except that they provide scope for coupled differential-algebraic systems. This extended functionality allows for the solution of more complex and more general problems, e.g., periodic boundary conditions and integro-differential equations.

d03pp is similar to d03ph but allows remeshing to take place in the spatial direction. This facility can be very useful when the nature of the solution in the spatial direction varies considerably over time.

For parabolic systems in two space dimensions see Section 3.6.

3.4 Black–Scholes Equations

d03nc solves the Black–Scholes equation

$$\frac{\partial f}{\partial t} + (r-q)S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2} = rf$$

$$S_{\min} < S < S_{\max}, \qquad t_{\min} < t < t_{\max},$$

for the value f of a European or American, put or call stock option. The parameters r, q and σ may each be either constant or time-dependent. The values of the Greeks are also returned.

In certain cases an analytic solution of the Black–Scholes equation is available. In these cases the solution may be computed by d03nd.

3.5 First-order Systems in One Space Dimension

There are three functions available for solving systems of first-order partial differential equations:

d03pe,

d03pk,

d03pr.

Equations may include nonlinear terms but the time derivative should occur linearly. There are certain restrictions on the coefficients to ensure that the problems posed can be solved by the above functions.

The method of solution is to discretize the space derivatives using the Keller box scheme and to solve the resulting system of ordinary differential equations using a 'stiff' solver.

d03pe is designed to solve a system of the form

$$\sum_{j=1}^{n} P_{ij}(x,t,U,U_x)\frac{\partial U_j}{\partial t} + Q_i(x,t,U,U_x) = 0,$$

where $i = 1,2,\ldots,n$, $a \le x \le b$, $t \ge t_0$.

d03pk is similar to d03pe except that it provides scope for coupled differential algebraic systems. This extended functionality allows for the solution of more complex problems.

d03pr is similar to d03pk but allows remeshing to take place in the spatial direction. This facility can be very useful when the nature of the solution in the spatial direction varies considerably over time.

d03pe, d03pk or d03pr may also be used to solve systems of higher or mixed order partial differential equations which have been reduced to first-order. Note that in general these functions are unsuitable for hyperbolic first-order equations, for which an appropriate upwind discretization scheme should be used (see Section 3.7 for example).

3.6 Second-order Systems in Two Space Dimensions

There are two functions available for solving nonlinear second-order time-dependent systems in two space dimensions:

d03ra,

d03rb.

These functions are formally applicable to the general nonlinear system

$$F_j\left(t,x,y,u,u_t,u_x,u_y,u_{xx},u_{xy},u_{yy}\right) = 0,$$

where $j = 1,2,\ldots,$ **npde**, $(x,y) \in \Omega$, $t_0 \le t \le t_{\text{out}}$. However, they should not be used to solve purely hyperbolic systems, or time-independent problems.

d03ra solves the nonlinear system in a rectangular domain, while d03rb solves in a rectilinear region, i.e., a domain bounded by perpendicular straight lines.

Both functions use the method of lines and solve the resulting system of ordinary differential equations using a backward differentiation formula (BDF) method, modified Newton method, and BiCGSTAB iterative linear solver. Local uniform grid refinement is used to improve accuracy.

Utility functions d03ry and d03rz may be used in conjunction with d03rb to check the user-supplied initial mesh, and extract mesh co-ordinate data.

3.7 Convection-diffusion Systems

There are three functions available for solving systems of convection-diffusion equations with optional source terms:

d03pf,

d03pl,

d03ps.

Equations may include nonlinear terms but the time derivative should occur linearly. There are certain restrictions on the coefficients to ensure that the problems posed can be solved by the above functions, in particular the system must be posed in conservative form (see below). The functions may also be used to solve hyperbolic convection-only systems.

Convection terms are discretized using an upwind scheme involving a numerical flux function based on the solution of a Riemann problem at each mesh point (see LeVeque 1990); and diffusion and source terms are discretized using central differences. The resulting system of ordinary differential equations is solved using a 'stiff' solver. In the case of Euler equations for a perfect gas various approximate and exact Riemann solvers are provided in d03pu, d03pv, d03pw and d03px. These functions may be used in conjunction with d03pf, d03pl and d03ps.

d03pf is designed to solve systems of the form

$$\sum_{j=1}^{n} P_{ij}(x,t,U)\frac{\partial U_j}{\partial t} + \frac{\partial}{\partial x}F_i(x,t,U) = C_i(x,t,U)\frac{\partial}{\partial x}D_i(x,t,U,U_x) + S_i(x,t,U),$$

or hyperbolic convection-only systems of the form

$$\sum_{j=1}^{n} P_{ij}(x,t,U)\frac{\partial U_j}{\partial t} + \frac{\partial F_i(x,t,U)}{\partial x} = 0,$$

where $i = 1,2,\ldots,n$, $a \leq x \leq b$, $t \geq t_0$.

d03pl is similar to d03pf except that it provides scope for coupled differential algebraic systems. This extended functionality allows for the solution of more complex problems.

d03ps is similar to d03pl but allows remeshing to take place in the spatial direction. This facility can be very useful when the nature of the solution in the spatial direction varies considerably over time.

3.8 Automatic Mesh Generation

The function d03ma places a triangular mesh over a given two-dimensional region. The region may have any shape and may include holes. It may also be used in conjunction with functions from the NAG Finite Element Library. A wider range of mesh generation functions are available in Chapter D06.

3.9 Utility Functions

d03ua (d03ub) calculates, by the Strongly Implicit Procedure, an approximate correction to a current estimate of the solution of a system of simultaneous algebraic equations for which the iterative update matrix is of five (seven) point molecule form on a two- (three-) dimensional topologically-rectangular mesh.

Functions are available in the Linear Algebra Chapters for the direct and iterative solution of linear equations. Here we point to some of the functions that may be of use in solving the linear systems that arise from finite difference or finite element approximations to partial differential equation solutions. Chapters F01, F04 and F11 should be consulted for further information and for the appropriate function

documents. Decision trees for the solution of linear systems are given in Section 4 in the F04 Chapter Introduction.

The following functions allow the direct solution of symmetric positive-definite systems:

Band	f07hd and f07he
Variable band (skyline)	f01mc and f04mc
Tridiagonal	f07ja, f07jd and f07je
Sparse	f11ja* and f11jb

(* the description of f11jb explains how f11ja should be called to obtain a direct method) and the following functions allow the iterative solution of symmetric positive-definite and symmetric-indefinite systems:

Sparse	f11gd, f11ge, f11gf, f11ja, f11jc and f11je

The latter two functions above are black box functions which include Incomplete Cholesky, SSOR or Jacobi preconditioning.

The following functions allow the direct solution of nonsymmetric systems:

Band	f07bd and f07be
Almost block-diagonal	f01lh and f04lh
Tridiagonal	f01le, f04le or f07ca
Sparse	f01br (and f01bs) and f04ax

and the following functions allow the iterative solution of nonsymmetric systems:

Sparse	f11bd, f11be, f11bf, f11da, f11dc and f11de

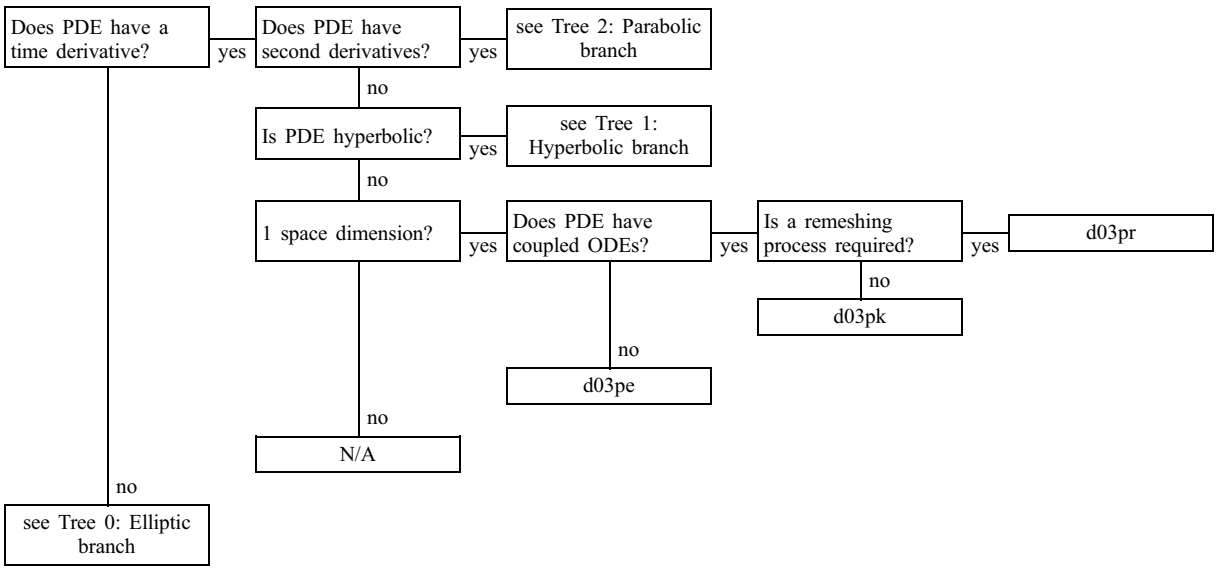
The latter two functions above are black box functions which include incomplete LU, SSOR and Jacobi preconditioning.

The functions d03pz and d03py use linear interpolation to compute the solution to a parabolic problem and its first derivative at the user-specified points. d03pz may be used in conjunction with d03pc, d03pe, d03ph, d03pk, d03pp and d03pr. d03py may be used in conjunction with d03pd and d03pj.

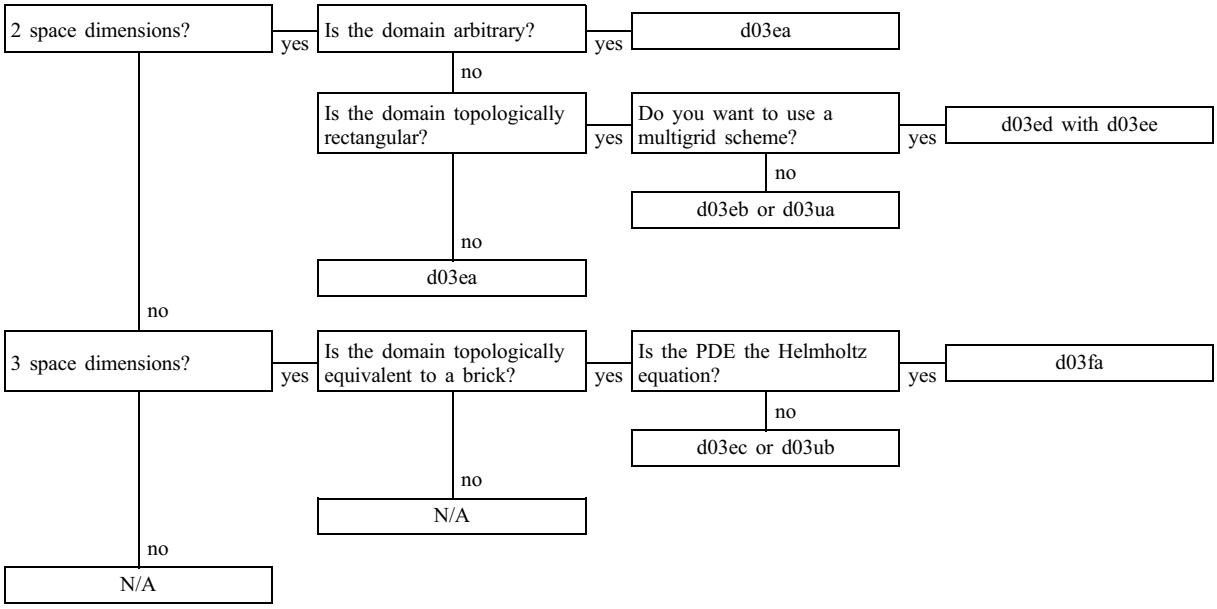
d03ry and d03rz are utility functions for use in conjunction with d03rb. They can be called to check the user-specified initial mesh and to extract mesh co-ordinate data.

4 Decision Trees

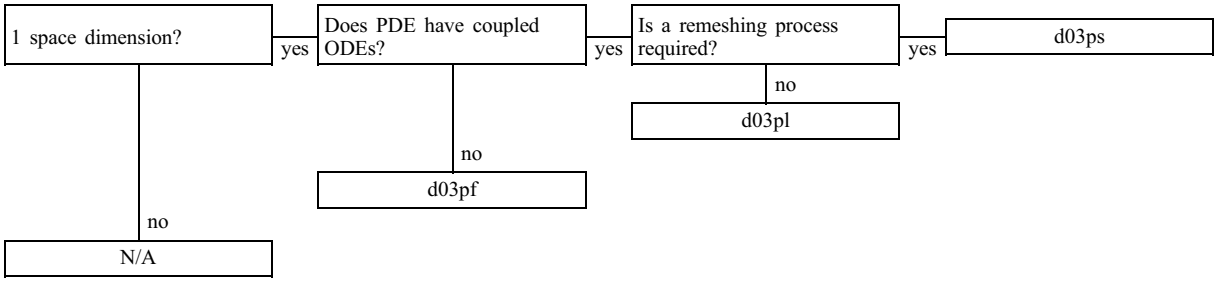
Tree -1



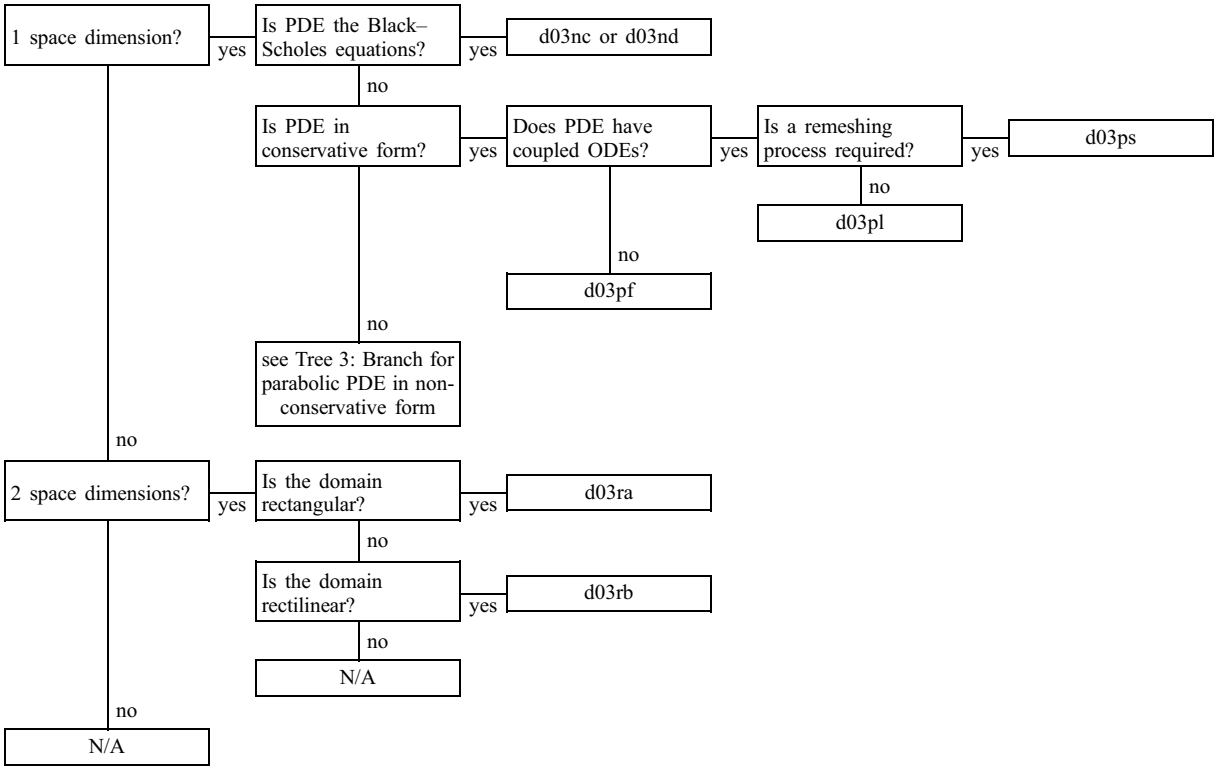
Tree 0: Elliptic branch



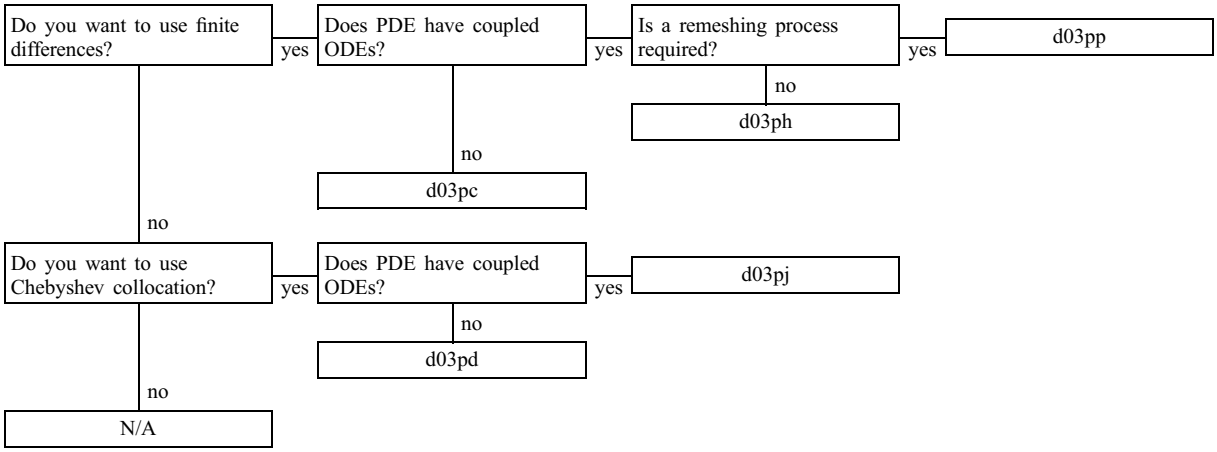
Tree 1: Hyperbolic branch



Tree 2: Parabolic branch



Tree 3: Branch for parabolic PDE in non-conservative form



5 Index

6 References

Ames W F 1977 *Nonlinear Partial Differential Equations in Engineering* (2nd Edition) Academic Press

Berzins M 1990 Developments in the NAG Library software for parabolic equations *Scientific Software Systems* (ed J C Mason and M G Cox) 59–72 Chapman and Hall

Jaswon M A and Symm G T 1977 *Integral Equation Methods in Potential Theory and Elastostatics* Academic Press

LeVeque R J 1990 *Numerical Methods for Conservation Laws* Birkhäuser Verlag

Mikhlin S G and Smolitsky K L 1967 *Approximate Methods for the Solution of Differential and Integral Equations* Elsevier

Mitchell A R and Griffiths D F 1980 *The Finite Difference Method in Partial Differential Equations* Wiley

Pennington S V and Berzins M 1994 New NAG Library software for first-order partial differential equations *ACM Trans. Math. Softw.* **20** 63–99

Richtmyer R D and Morton K W 1967 *Difference Methods for Initial-value Problems* (2nd Edition) Interscience

Smith G D 1985 *Numerical Solution of Partial Differential Equations: Finite Difference Methods* (3rd Edition) Oxford University Press

Swarztrauber P N and Sweet R A 1979 Efficient Fortran subprograms for the solution of separable elliptic partial differential equations *ACM Trans. Math. Software* **5** 352–364

Wait R and Mitchell A R 1985 *Finite Element Analysis and Application* Wiley