fficient PDECOL code', Technical
tatistics and Computing Science,

abolic problems. In 'Numerical
Hubbard, editor), Academic

540. PDECOL: General colloca-
CM Trans. Math. Softw. 5,

he solution of almost block diago-
Gaussian points with monomial
., submitted.
lgorithm XXX: ABDPACK and
f almost block diagonal linear sys-
nts with monomial basis functions.

nation for solving certain linear

Developments in the NAG library software for parabolic equations

M. Berzins
School of Computer Studies,
Leeds University, UK

Abstract The NAG Library parabolic partial differential equation (p.d.e.) sub-chapter D03P has recently been revised to make use of the successful SPRINT Leeds University/Shell Research Software and so offers a range of different space discretization methods that can be applied to a common problem class of parabolic-elliptic systems of p.d.e.s with coupled differential-algebraic equations. Three significant advances over the existing software in D03P are the wide class of problems that can be solved, the spatial remeshing routines that are available and the modular structure which allows a wide range of NAG time integration and linear algebra routines to be used with all the spatial discretization routines. The improvements that these new routines offer over the existing D03P routines are illustrated by a number of example problems. The future requirements of software in this area are considered.
Key Words Parabolic-elliptic equations, Method of lines, NAG Library software.

1.0 Introduction

The work of Dew and Walsh (1981) in developing the present D03P parabolic equations software was instrumental in the development of the Leeds University/Shell Research SPRINT software of Berzins, Dew

1989

and Furzeland (1988). Both packages are based on the successful method of lines approach, see Gaffney (1982) for time dependent p.d.e. problems. Although the existing D03P routines have stood up well to the test of time, the wider range of discretization methods in SPRINT, the spatial remeshing facilities and the broader class of problems solved by SPRINT will greatly improve the flexibility of the chapter.

The development of the SPRINT software resulted in three new spatial discretization codes together with routines for adaptive spatial grids and specialized time integration. The main routines that are available in SPRINT are a new finite-difference spatial discretization method developed by Skeel and Berzins (1987) and the novel variable order Chebyshev collocation method developed by Berzins and Dew (1987). These methods both treat second order equations directly while the Keller (1970) box scheme code developed by Furzeland treats systems of first-order equations. Both the box scheme and the finite difference scheme have a spatial remeshing scheme in which mesh movement is specified by a user supplied monitor function.

In common with the existing software, the user interface has both high and low levels. The high level interfaces provide easy access to routines for systems of p.d.e.s (rather like the present D03PBF routine). In contrast the low-level routines provide a wide range of facilities including a choice of linear algebra, time integration routines and spatial remeshing as well as allowing the user to solve non-standard problems consisting of coupled p.d.e./o.d.e. problems. The low level interface is itself called by the high level routines. All the routines in the low-level interface call a very general time integration driver which accesses the reverse communication driver for the D02N subchapter. This structure provides a clean interface between the spatial discretization and time integration parts of the software, (see Berzins, Brankin and Gladwell (1987)). Furthermore other integrators (see Berzins and Furzeland (1985)) developed for the solution of the large systems of o.d.e.s that arise when partial differential equations are discretized in space have been added.

2.0 P.D.E. Problem Class

The master equation format for the software has been selected to fit two types of problems. The first type of problems consist of systems of p.d.e. problems, similar to the Dew and Walsh problem class (1981), and correspond to the high level easier-to-use routines. The master equation for the p.d.e.s is

$$\sum_{p=1}^{NPDE} P_{j,p}(x,t,\underline{U},\underline{U}_x)\frac{\partial U_p}{\partial t} + Q_j(x,t,\underline{U},\underline{U}_x) \qquad (1)$$

$$= x^{-m}\frac{\partial}{\partial x}\left(x^m R_j(x,t,\underline{U},\underline{U}_x)\right)\ j=1,\ldots,NPDE\ (x,t)\in[A,B]x(0,t_e]$$

where m denotes the space geomtry type. The vector $\underline{U}$ is defined by

$$\underline{U}(x,t) = [U_1(x,t),\ldots,U_{NPDE}(x,t)]^T \qquad (2)$$

The function R() can be thought of as a flux, e.g. $K\frac{\partial U}{\partial x}$, and is used al in the definition of the boundary conditions. For each p.d.e. the boundary conditions have the master equation form:

$$\beta_j(t)R(x,t,\underline{U},\underline{U}_x) = \gamma_j(x,t,\underline{U},\underline{U}_x,\underline{U}_t),\ j=1,\ldots,NPDE \text{ at } x=A \text{ and } x=B. \qquad (3)$$

The initial conditions for $\underline{U}(X,t,)$ are assumed to be supplied by the user in the form

$$\underline{U}(x,0)=[K_1(x),\ldots,K_{NPDE}(x)]^T \qquad (4)$$

2.1 Coupled O.D.E. and P.D.E. Problems

The second class of problems is that of coupled p.d.e.s. and o.d.e.s. The p.d.e. variables are denoted as above and the o.d.e. variables are denoted by a vector $\underline{V}$ of length NV. The o.d.e.s may be coupled to the p.d.e.s at various space points $\underline{\xi}$ which may or may not be equal to some of the spatial mesh points.

The master equation is

$$\sum_{p=1}^{NPDE} P_{j,p}(x,t,\underline{U},\underline{U}_x,\underline{V})\frac{\partial U_p}{\partial t} + Q_j(x,t,\underline{U},\underline{U}_x,\underline{V},\dot{\underline{V}}) \qquad (5)$$

$$= x^{-m}\frac{\partial}{\partial x}\left(x^m R_j(x,t,\underline{U},\underline{U}_x,\underline{V})\right),\ j-1,\ldots,NPDE\ (x,t)\in[A,B]x(0,t_e]$$

where m denotes the space geometry type.

The boundary conditions have a similar master equation form to that in equation (3) above at x=A and x=B

$$\beta_j(t)R(x,t,\underline{U},\underline{U}_x,\underline{V}) = \gamma_j(x,t,\underline{U},U_x, \quad \underline{V},\dot{\underline{V}}),j=1,\ldots,NPDE. \tag{6}$$

The master equation for the coupled o.d.e.s is

$$\underline{F}(\underline{V}, \dot{\underline{V}}, \underline{\xi}, \underline{U}^*, \underline{U}^*_x, \underline{R}^*, \underline{U}^*_t, \underline{U}^*_{xt}) = 0 \tag{7}$$

The arrays

$$\underline{U}^*, \underline{U}^*_x, \underline{R}^*, \underline{U}^*_t, \underline{U}^*_{xt}$$

hold the solution, flux and derivative values at the coupling points $\underline{\xi}$. The initial conditions for $\underline{U}(X,t)$ and $\underline{V}(t)$ are assumed to be known vectors which can be supplied by the user in a subroutine.

2.2 First Order P.D.E. Systems

The underlying assumption in the above problem classes is that the p.d.e. involves second order space derivatives. However, many systems of p.d.e.s can be written more naturally as systems of first-order p.d.e.s, e.g. see Section 3.4 below. In this case the master equation for the p.d.e.s is

$$\sum_{p=1}^{NPDE} P_{j,p}(x,t,\underline{U},\underline{U}_x,\underline{V}) \frac{\partial U_p}{\partial t} + Q(x,t,\underline{U},\underline{U}_x,\underline{V},\dot{\underline{V}}) = 0 \tag{8}$$

$$, \; j=1,\ldots,NPDE \; (x,t)\varepsilon[A,B]x(0,t_e]$$

The evaluation of the p.d.e. functions P and Q is performed at the mid-points of the user supplied mesh.

2.3 Boundary Conditions for First Order Systems

The main difference between first-order systems of p.d.e.s and second-order systems of p.d.e.s is that in the first-order case each p.d.e. component only needs one boundary condition rather than two as in the second-order case. Furthermore the user must choose which component has a boundary condition at the left or right boundary with some care.

The general rule is that if the characteristic direction of the ith solution component at the (say) left boundary points into the interior then the boundary condition for the ith component should be specified at the left-hand boundary. In other words at x=A the boundary conditions must have the general form

$$\sum_{P=1}^{NPDE} E^{L}_{j,p}(x,t,\underline{U},\underline{U}_x,\underline{V})\frac{\partial U_p}{\partial t} + S^{L}_{j}(x,t,\underline{U},\underline{U}_x,\underline{V},\dot{\underline{V}})=0, j=1,\ldots,NLEFT \qquad (9)$$

and at x=B

$$\sum_{p=1}^{NPDE} E^{R}_{j,p}(x,t,\underline{U},\underline{U}_x,\underline{V})\frac{\partial U_p}{\partial t} + S^{R}_{j}(x,t,\underline{U},\underline{U}_x,\underline{V},\dot{\underline{V}})=0, j=1,\ldots,NRIGHT \qquad (10)$$

where NLEFT + NRIGHT = NPDE, the number of p.d.e.s in the system. Non-local boundary conditions which cannot be separated into left and right boundary conditions can be included by making use of the coupled o.d.e./p.d.e. facility, as in Section 3.2 below.

2.4 Coupled O.D.E. Equations

The master equation for the coupled o.d.e.s is exactly the same as in equation (7) above except that for first order systems the flux $\underline{R}^*$ is undefined and so the problem class is

$$\underline{F}\ (\underline{V},\ \dot{\underline{V}},\ \underline{\xi},\ \underline{U}^*_x,\ \underline{U}^*_t,\ \underline{U}^*_{xt}) = 0$$

3.0 Example Problems

The problem class outlined above allows many interesting problems to be solved such as moving boundary problems by co-ordinate transformation and p.d.e. problems with non-local boundary conditions. Several interesting examples of this type of problem are described by Schryer (1984). In this section we shall present four examples of such problems.

3.1 A Fourth Order P.D.E., Zaturska, Drazin and Banks (1988)

The first problem provides an example of a p.d.e. problem that can be solved very easily using the high-level p.d.e. interface. The problem consists of a fourth order p.d.e. which can be written as a pair of

second order elliptic parabolic p.d.e.s for U(x,t) and V(x,t).

$$0 = \frac{\partial^2 U}{\partial x^2} - V \tag{11}$$

$$\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2} + V\frac{\partial U}{\partial x} - \frac{\partial V}{\partial x}U \tag{12}$$

The boundary conditions are given by

$\frac{\partial U}{\partial x} = 0$ at $x = -1$ and at $x = 1$, $U(-1,t) = 1$ and $U(1,t) = -1$

The absence of boundary conditions for V(x,t) does not pose any difficulties provided that the derivative flux boundary conditions are assigned to the first p.d.e. (equation (11)) which has the correct flux ($\frac{\partial U}{\partial x}$). The conditions of U(x,t) at the boundaries are then assigned to the second p.d.e. by setting $\beta_2(t)=0$ in equation (3) and placing the Dirichlet boundary conditions on U(x,t) in the function $\gamma_2(\ldots)$.

3.2 Periodic Boundary Conditions, Schryer (1984)

The next example shows how the coupled p.d.e./o.d.e. problem class allows problems with non-local boundary conditions to be solved. The problem consists of a quasi-linear heat equation with periodic boundary conditions:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + g(x,t) \tag{13}$$

The boundary conditions are given by

$$\frac{\partial U}{\partial x}(-\pi,t) = \frac{\partial U}{\partial x}(\pi,t) \text{ and } U(-\pi,t) = U(\pi,t).$$

The boundary conditions are modified by introducing an o.d.e. variable $v_1(t)$ which is equal to the common flux at the boundaries:

$$\frac{\partial U}{\partial x}(-\pi,t) = v_1(t) \text{ and } \frac{\partial U}{\partial x}(\pi,t) = v_1(t)$$

The extra coupled algebraic equation needed to define all the variables is

$$U(-\pi,t) - U(\pi,t) = 0.$$

The spatial coupling points are thus defined by $\xi = [-\pi,\pi]^T$. Another example which also results in an algebraic coupled equation which does not contain the coupled o.d.e. variable is given by Berzins and Furzeland (1986).

3.3 An Integro-Differential Equation, Schryer (1984)

Let U(x,t) be defined by the integro-differential equation:

$$\frac{\partial U}{\partial t} = p \frac{\partial^2 U}{\partial x^2} + \int_0^1 c(x,y)U(y,t)) \, dy \quad (x,t) \in [0,1]x(0,1]. \tag{14}$$

with given boundary and initial conditions. In practice we need to be able to evaluate the p.d.e., defining functions one mesh point at a time, whereas the integral involves values across the entire spatial range. The problem can be cast as a coupled p.d.e./o.d.e. problem in the following way. Assume that the spatial mesh is defined by

$$A = x_1 < x_2 < x_3 < \ldots < x_N = B$$

and that the approximate solution has the form

$$U(x,t) = \sum_{i=1}^{N} \Phi_i(x) \, U_i(t)$$

where $U_i(t)$ is the nodal solution value at x_i and $\Phi_i(x)$ is the spatial basis function defined by $\Phi_i(x_j) = \delta_{ij}$ where δ is the Kronecker δ. We now define the functions $c_i(x)$ by

$$c_i(x) = \int_0^1 c(x,y) \; \Phi_i(y) \, dy, \; i=1,\ldots,N$$

and the N coupled o.d.e. variables $v_i(t)$ by

$$v_i(t) - U_i(t) = 0, i = 1,\ldots,N. \tag{15}$$

The integral can now be approximated in terms of the o.d.e. variables $v_i(t)$ and the functions $c_i(x)$ and a suitable quadrature rule such as the trapezoidal rule with weights w_i, i=1,...,N may be used to give

$$\sum_{i=1}^{N} w_i \, c_i(x) \, v_i(t) \approx \int_0^1 c(x,y) \, U(y,t)) \, dy$$

and this may be substituted in the integro-differential equation

$$\frac{\partial U}{\partial t} = p\,\frac{\partial 2_U}{\partial x^2} + \sum_{i=1}^{N} w_i\, c_i(x)\, v_i(t),\ (x,t)\,\varepsilon\,[0,1]x(0,1].$$

In this case the coupled o.d.e. system consists of the N equations (15), and the coupling points space vector $\underline{\xi}$ consists of all the spatial mesh points $\underline{\xi} = [x_i,\ldots,x_N]^T$ as the coupled o.d.e.s (equations (15)) involve p.d.e. values at all these points. The vector $\underline{v}(t)$ is made up of the N coupled o.d.e. variables in the following way

$$\underline{v}(t) = [v_1(t),\ v_2(t),\ldots,v_N(t)]^T$$

where $v_i(t)$ is defined by equation (15).

3.4 A First Order System of Differential Equations

Consider the boundary layer equations

$$\frac{\partial u}{\partial t} + \frac{\partial v}{\partial x} = 0$$

$$u\,\frac{\partial u}{\partial t} = -v\,\frac{\partial u}{\partial x} + \frac{\partial w}{\partial x} \qquad (16)$$

$$w = \frac{\partial u}{\partial x}$$

with boundary conditions

$u(0,t) = v(0,t) = 0$ at $x = 0$ and $u(0,t) = 1$ at $x = \infty$

and the initial conditions

$v(x,t) = 0,\ u(x,t) = 1$ for $x > 0$

A non-standard feature of the above problem is the semi-infinite spatial range and the boundary condition at inifinity. Some of the different possible ways of treating p.d.e. problems on infinite domain are described by Grosch and Orszag (1977). It is possible that if, as x tends to infinity, the solution rapidly approaches some constant

ferential equation

,1].

of the N equations
nsists of all the
oled o.d.e.s (equations
. The vector $\underline{v}(t)$ is
e following way

ions

(16)

$x = \infty$

the semi-infinite
finity. Some of the
lems on infinite domains
s possible that if,
roaches some constant

value then the infinite domain may be mapped onto a finite one. However these mappings may not be effective in some cases (see Grosch and Orszag (1977)) and it may be necessary to approximate infinity by a large positive value, x_∞. This is the approach adopted for the above problem. However it should be noted that the steady-state solution to the above problem is dependent on the value of x_∞ and is given by

$$v(x,t) = 0,\ u(x,t) = x/x_\infty,\ w(x,t) = 1/x_\infty \text{ for } t > 5$$

This shows that some care must be taken when using the approach.

4.0 The New D03P Software

Three aspects of the software will now be considered in detail. These are the user interface, the spatial discretization methods and the time integration interface.

4.1 User Interface

The general user interface is very similar for each of the three discretisation methods in that the user writes a driver program which calls the library subroutines and the user also writes subroutines to describe the p.d.e. in terms of the master equation format in Section 2 above. In the case of the high-level interface these names are fixed whereas in the case of the low-level interface these names can vary. In the case of p.d.e. problems the user supplies three subroutines:

(i) One routine to evaluate the functions P(...), Q(...) and R(...) in the p.d.e. equation (1),

(ii) One routine to evaluate the function $\beta(...)$ and $\gamma(...)$ in the boundary conditions equation (3), and

(iii) One routine to supply the initial conditions, equation (4).

In the case of coupled p.d.e./o.d.e. problems the user must also supply:

(iv) A routine to evaluate the residual of the coupled o.d.e. system, equation (7).

and, in the case when remeshing is used, the user must supply

(v) A routine to describe the form of the remeshing monitor function which is used (see Furzeland (1985)) to calculate the new mesh.

It should be noted that the last parameters of each of the usersupplied problem definition routines is the integer IRES. This parameter can be reset, so as to force the integrator to retake the step to avoid unphysical solution values or so as to stop integration completely.

4.2 Spatial Discretization Routines

The new finite difference routine developed by Skeel and Berzins (1987) is an improvement over the existing routine as it has increased accuracy for polar problems and problems with material interfaces. In order to accurately model the solutions of problems which involve some kind of travelling wave it is necessary for the spatial mesh points to follow the wave. The remesh module that is based on the work of Furzeland (1985) allows the mesh to be changed to discrete time levels according to a criterion supplied by the user and based on the current solution profile. This criterion consists of a remeshing monitor routine which has access to solution and derivative values at the mesh points and which returns the value of a function which should be equi-distributed by the new mesh. Although this method is not a complete solution to this class of problems, in that it is not designed to cope with shock wave problems, it has been very successful in dealing with flame propagation problems in which steep but smooth wave fronts are generated.

In order to solve p.d.e. problems with smooth solutions it may be more efficient to use spatial discretization methods with greater accuracy than the second-order method of Skeel and Berzins (1987). It is for this reason that the CO collocation method of Berzins and Dew (1987) has been implemented in SPRINT and in the D03P software. This method is applicable to a broad class of problems and offers a family of high-order formulae based on Chebyshev polynomials from which the user can select the order of approximating polynomial to be used. The piecewise polynomial has C^0 continuity at each breakpoint. The lowest degree of polynomial that can be used is one (linear polynomial), however the power of the collocation method lies in its ability to use high-order polynomials, up to degree fifty.

ach of the usersupplied
. This parameter can be
he step to avoid
gration completely.

Skeel and Berzins (1987)
it has increased accuracy
nterfaces. In order to
n involve some kind of
mesh points to follow
he work of Furzeland
e time levels according
n the current solution
g monitor routine which
the mesh points and
uld be equi-distributed
complete solution to
ed to cope with shock
aling with flame prop-
fronts are generated.
solutions it may be
thods with greater
and Berzins (1987). It
od of Berzins and Dew
e D03P software. This
ns and offers a family
omials from which the
nomial to be used. The
reakpoint. The lowest
inear polynomial), how-
n its ability to use

The discretization method employed for the many systems of p.d.e.s which can be written more naturally as systems of first-order p.d.e.s is based on the well-known box scheme of Keller (1970). However, within the method of lines framework, the box scheme is applied in the space variable only and the time integration performed by using the D02N integrators. The same spatial remeshing option as for the finite difference routines is also available.

4.3 Time Integration

One of the main differences between the old and new chapters is that the new chapter calls the D02N sub-chapter routines to perform the time integration, rather than using its own integrator. The D02N sub-chapter, also developed from SPRINT, is specifically intended to be used with p.d.e. method of lines software (see Berzins and Gladwell (1987)) and solves differential-algebraic equations of the form

$$A(t,\underline{Y})\dot{\underline{Y}} = \underline{G}(t,\underline{Y}(t)) \qquad (17)$$

where $A(t,\underline{Y})$ may be a singular matrix. The default integrator is that based on the backward differentiation formulae of Gear (see Berzins (1986)) while the default linear algebra routines are the banded matrix routines. The new integrators are more efficient, treat systems of differential algebraic equations better and allow different linear algebra modules to be selected for the different discretization methods.

The main implication of the problem class solved by the D02N routines on the p.d.e. problem class is that the discretization method must generate a differential-algebraic system of the form of equation (17). This means that the p.d.e. functions $Q(\ldots)$, $\gamma_j(\ldots)$ and $F(\ldots)$ defined by equations (5), (6) and (7) must be linear in any of the time derivatives present in their argument lists. This means that there should be no multiplication of one time derivative by another and that there should be no functions of time derivatives present in the functions

The following convention is used by the D03P routines introduced above in discretizing systems of p.d.e.s that are coupled to o.d.e.s. The o.d.e. solution vector, say $\underline{Y}(t)$, that is passed to the D02N sub-routines is ordered as follows. Suppose that a system of NPDE partial

differential equations has NV coupled o.d.e.s, and that NPTS spatial mesh points are used in the discretization of the p.d.e. The p.d.e./o.d.e. solution component that is stored in a given component of the solution vector used by DO3P, $\underline{Y}(t)$ is defined by:

$Y_k(t)=U_i(x_j,t)$ where k=NPDEx(j-1)+i,i=1,...,NPDE, j=1,...,NPTS and

$Y_1(t)=V_m$ where 1=NPDExNPTS+m,m=1,...,NV.

In other words the p.d.e. components are stored column-wise i.e. all components at mesh point 1, then mesh point 2, etc. and all the p.d.e. solution values are stored before the coupled o.d.e. components.

The most efficient linear algebra routines to be used in conjunction with the method of lines and DO3P are the banded matrix routines. The above convention used by the discretization routines in ordering the o.d.e. solution vector means that banded matrix routines CANNOT in general be used with coupled o.d.e./p.d.e. problems. In the case of such problems the sparse matrix techniques are probably the most efficient anyway. Moreover in the case of the CO collocation method when only two or three breakpoints are employed, the o.d.e. system integrated in time will have a structure that makes the full matrix routines the most appropriate.

The integrators in DO2 are not specifically designed for the low accuracy solution of large systems of o.d.e.s that arise when the method of lines is applied to parabolic equations. The SPRINT integrator STHETB of Berzins and Furzeland (1985) is based on the well-known theta method and on existing theta method codes developed at both Leeds University and Shell Research. The code can be selected from the low level DO3P interfaces and is tuned for the solution of parabolic p.d.e.s with the option to use function iteration, where possible, to improve efficiency.

The form of the solution produced by the DO3P software consists of p.d.e. values at spatial mesh points. Solution values at non-mesh points can be generated by calling one of two spatial interpolation routines. A simple routine based on linear interpolation is available for the finite difference routines and a Chebyshev polynomial routine is available for use with the CO collocation generated solution.

nd that NPTS spatial
e p.d.e. The p.d.e./
ven component of the
:

, j=1,...,NPTS and

column-wise i.e. all
tc. and all the p.d.e.
d.e. components.
be used in conjunction
matrix routines. The
ines in ordering the
routines CANNOT in
ems. In the case of
robably the most ef-
collocation method when
o.d.e. system integrated
matrix routines the

esigned for the low
at arise when the
s. The SPRINT integrator
on the well-known theta
ped at both Leeds
selected from the low
ion of parabolic
n, where possible,

software consists of
values at non-mesh
atial interpolation
rpolation is available
ev polynomial routine
erated solution.

5.0 Summary

Table 1 shows the range of routines in the revised D03P chapter for the solution of parabolic-elliptic equations with coupled algebraic equations. This enables many non-standard problems to be solved provided that they can be defined as a coupled p.d.e./o.d.e. system.

Table 1 Summary of D03P Subroutines

Method Type	High Level Module	Low Level Module	Remeshing Available
Finite Diff.	D03PCF	D03PHF	Yes
CO Collocation	D03PDF	D03PJF	No
Keller Box	D03PEF	D03PLF	No

Although there is no software in D03P for the solution of p.d.e. problems in two space dimensions it is planned to produce such software in the near future. The main disadvantage of the present software is that there is no attempt to estimate or to control the space error by remeshing. The required solution is a robust global error tracking procedure which includes both space and time error components and has a strategy for balancing these errors. Berzins and Dew (1987) have implemented such algorithms, but more research is needed before NAG Library software of this type becomes a reality. This research is now underway.

References

Berzins, M. and Furzeland, R.M. (1985), A type-insensitive method for the solution of stiff and non-stiff differential equations. Leeds University, Department of Computer Studies, Report No 204

Berzins, M. (1986), A C^1 interpolant for codes based upon backward differentiation formulae. Applied Numerical Analysis, Vol 2 pp 109-118

Berzins, M. and Furzeland, R.M. (1986), A user's manual for SPRINT - a versatile software package for solving systems of algebraic ordinary and partial differential equations : part 2 partial differential equations, TNER. 86.050, Thornton Research Centre, Chester

Berzins, M. and Dew, P.M. (1987), CO Chebyshev methods for parabolic p.d.e.s. I.M.A. Journal of Numerical Analysis, Vol 7, pp 15-37

Berzins, M. Brankin, R. and Gladwell, I. (1987), The design of stiff integrators in the NAG library, University of Manchester, Department of Mathematics Report, 7, see also SIGNUM Newsletter (1988)

Berzins, M., Dew, P.M. and Furzeland, R.M. (1988), Developing software for time-dependent problems using the method of lines and differential algebraic integrators, in press with Appl. Num. Math.

Dew, P.M. and Walsh, J.E. (1981) A set of library routines for solving parabolic equations in one space variable. ACM Trans. on Math. Soft., Vol 7, No 3, pp 295-314

Furzeland, R.M. (1985), The construction of adaptive space meshes, TNER,85,022, Shell Research Ltd., Thornton Research Centre, P.O. Box 1, Chester CH1 3SH

Gaffney, P. (1982), Using the method of lines technique to solve boundary value p.d.e.s. In 'Scientific Computation', ed R. Stepleman, IMACS/North Holland.

Grosh, C. and Orszag, S.A. (1977), Numerical solution of problems in unbounded regions: co-ordinate transforms. Journal of Computational Physics, Vol 25, pp 273-296

Keller, H.B. (1970), A new difference scheme for parabolic equations, Numerical Solution of Partial Differential Equations, Vol 2, (J. Bramble, ed) Academic Press, New York

Schryer, N.L. (1984), Partial differential equations in one space variable, Computing Science Technical Report No 115, 1984, AT and T Laboratories, Murray Hill, New Jersey 07974

Skeel, R. D. and Berzins, M. (1987), Spatial discretization methods for parabolic p.d.e.s, paper submitted to SIAM Journ. of Sci. and Stat. Comp.

Zaturska, N.B., Drazin, P.G. and Banks, W.H.H. (1988), On the flow of a viscous fluid driven along a channel by suction at porous walls. Fluid Dynamics Research, Vol 4, 1988, in press