THE METHOD OF LINES - THEORY, SOFTWARE AND SOME APPLICATIONS

N.G. Barton

In a recent review of software for the numerical solution of partial differential equations (PDEs), Machura & Sweet [11] assert that the method of lines (MOL) is the most popular technique for solving systems of time-dependent PDEs. The MOL is a direct and practical consequence of extensive research on the numerical solution of ordinary diffential equations. A number of robust and user-friendly software interfaces are now available to implement the method, and this note describes the application of one such interface (PDEONE, Sincovec & Madsen, [15]) to three consulting problems. The results confirm that the MOL is a valuable consequence of past mathematical development.

1. INTRODUCTION

The numerical solution of partial differential equations (PDEs) is a bread-and-butter issue in applied mathematics and most branches of the physical and engineering sciences. Many of the PDEs which require a numerical solution are (possibly systems of) parabolic PDEs for which the method of lines (MOL) is ideally suited. The MOL is based on semi-discretization: the conversion of PDEs into a set of ordinary differential equations (ODEs). There are many ways available for this, and the utility of all the methods derives from highly developed and robust software for the numerical solution of ODEs. Thus the MOL - a widely used and obviously practical

technique - is a direct product of mathematical research in two areas: the formalism underlying the semi-discretization, and the theoretical and practical development of ODE integrators. This marriage of results from past research leads to an approach which is quite different to numerical schemes based on traditional finite differences. More recently, numerical analysts have provided some direct theoretical support for users of the MOL by examining the truncation error associated with various forms of semi-discretization (see, for example, [3,4,8,9,17]).

Machura & Sweet ([11], Section 12) give a list of 9 packages which have been developed since the mid 1970s to act as an interface between the PDE requiring solution and the ODE integrators. The application of PDEONE, one of the earliest of these software interfaces, to three consulting problems is described in the present contribution. The three problems investigated are quite representative of parabolic PDEs: the first is posed by a standard one-dimensional non-linear PDE with a non-linear boundary condition, the second problem is also non-linear and involves a system of one-dimensional PDEs, whilst the third involves a two-dimensional PDE. The results for the third problem also show that the MOL can be used as a building block in solving more complicated problems. In summary, the MOL as implemented using PDEONE and GEARB (a standard ODE integrator modified for systems with a banded Jacobian) is highly successful and deserving a place in every applied mathematician's tool-bag.

2. THE METHOD OF LINES

For the purposes of exposition, consider the application of the MOL to the single quasi-linear parabolic PDE

$$u_t = f(x,t,u,u_x,u_{xx}), \quad 0 < x < 1, \quad t > 0 \tag{2.1a}$$

under boundary conditions

(2.1b) $$\alpha u + \beta u_x = \gamma \quad \text{at} \quad x = 0,\ 1$$

and the initial condition

(2.1c) $$u(x,0) = g(x).$$

This problem may be discretized on the mesh

$$x_i = ih, \quad i = 0,1,2,\ldots,N+1, \quad h = 1/(N+1)$$

so that if $v_i(t)$ is an approximation to $u(x_i,t)$ and the spatial derivatives in (2.1a) are replaced by second order central difference approximations, the $v_i(t)$ satisfy the system of ODEs

(2.2a) $$\frac{dv_i}{dt} = f\left[x_i,t,v_i,\frac{v_{i+1}-v_{i-1}}{2h},\frac{v_{i-1}-2v_i+v_{i+1}}{h^2}\right]$$

$$= F_i(t,v_{i-1},v_i,v_{i+1})$$

for $i = 1,2,\ldots,N$. ODEs for $v_0(t)$ and $v_{N+1}(t)$ may be obtained by using (2.1b) and, if necessary, one-sided difference approximations for u_x, u_{xx} at the boundaries x=0,1. This gives the additional ODEs

(2.2b) $$\frac{dv_0}{dt} = F_0(t,v_0,v_1,v_2), \qquad \frac{dv_{N+1}}{dt} = F_{N+1}(t,v_{N-1},v_N,v_{N+1}).$$

The system of ODEs (2.2a,2.2b) has a banded structure and can be efficiently solved subject to the initial condition

(2.2c) $$v_i(0) = g(x_i)$$

using a standard ODE integrator modified for banded systems. The usual ODE package for this task appears to be GEARB (Hindmarsh, [10]). There are, of course, many embellishments on the above theme. For example, higher order or upwind differencing could be used. Also, the spatial variation could be described by writing

$$u(x,t) = \sum_{i=0}^{N+1} C_i(t) b_i(x) \tag{2.3}$$

where the $\{b_i(x)\}_{i=0}^{N+1}$ form a suitable basis (e.g. Fourier series, orthogonal polynomials or B-splines). ODEs for the $\{C_i(t)\}_{i=0}^{N+1}$ are then obtained by arithmetizing the spatial variation using some form of the method of weighted residuals (e.g. by collocation, or the Galerkin or least squares methods).

A survey of nine software interfaces written since 1975 for the method of lines is given by Machura & Sweet ([11], section 12). The present work is concerned with one of the earliest and possibly the best known of these - PDEONE (Sincovec & Madsen [15]). PDEONE follows the basic structure for discretization outlined in this section as applied to the system of NPDE coupled parabolic PDEs

$$\frac{\partial u_k}{\partial t} = f_k\left(t,x,\underset{\sim}{u},\frac{\partial \underset{\sim}{u}}{\partial x},x^{-c}\frac{\partial}{\partial x}\left(x^{c}\underset{\sim}{D}_k\cdot\frac{\partial \underset{\sim}{u}}{\partial x}\right)\right) \tag{2.4a}$$

$$(a < x < b, \quad t > t_0, \quad k = 1,2, \ldots, NPDE),$$

$$\underset{\sim}{u} = [u_1 \; u_2 \; \ldots \; u_{NPDE}]^T,$$

subject to boundary conditions

$$\alpha_k u_k + \beta_k \frac{\partial u_k}{\partial x} = \gamma_k \quad \text{at} \quad x = a,b, \quad t > t_0, \quad k = 1,2,\ldots,NPDE, \tag{2.4b}$$

and initial condition

$$\underset{\sim}{u}(x,0) = \underset{\sim}{g}(x), \quad a \leqslant x \leqslant b. \tag{2.4c}$$

The constant c in (2.4a) is 0,1 or 2 depending on whether the problem is posed in Cartesian, cylindrical or spherical co-ordinates respectively. A complete algorithm for PDEONE consisting of only 153 lines of code is freely available [15]. The user merely has to write a driver program and

three subroutines defining the boundary conditions (2.4b), the diffusion coefficients $\underset{\sim}{D}_k$ and the function $\underset{\sim}{f}$ in (2.4a). PDEONE then performs the discretization - on a variable mesh if desired - and GEARB solves the resulting ODE initial value problem with automatic timestep control to achieve a specified accuracy. Both PDEONE and GEARB are written in FORTRAN and the computations described subsequently in this note were performed in single precision on a VAX 11/750.

3. TRANSMISSION LINE PROBLEM

The first problem discussed is the quasi-linear parabolic PDE

$$\gamma(\theta)c(\theta)\frac{\partial\theta}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(\lambda(\theta)r\frac{\partial\theta}{\partial r}\right) + Q(r,\theta) \tag{3.1a}$$

for the temperature $\theta(r,t)$ in a solid metal conductor carrying an electric current. Here the temperature dependent coefficients are given by

$$\begin{aligned} \lambda(\theta) &= \lambda_0(1 + \kappa\theta) \quad \text{(thermal conductivity, W m}^{-1}\text{ K}^{-1}\text{)}, \\ \gamma(\theta) &= \gamma_0(1 + \delta\theta) \quad \text{(density, kg m}^{-3}\text{)}, \\ c(\theta) &= c_0(1 + \beta\theta) \quad \text{(specific heat, J kg}^{-1}\text{ K)}, \end{aligned} \tag{3.1b}$$

and the problem is to be solved subject to the initial and boundary conditions

$$\theta(r,0) = 0, \tag{3.1c}$$

$$\theta_r(0,t) = 0, \qquad \theta_r(a,t) = -h\{\theta(a,t)\}^q. \tag{3.1d}$$

The heating function $Q(r,\theta)$ is defined by (Dwight, [7], p.159)

$$Q(r,\theta) = |J(r,\theta)|^2\rho(\theta) \tag{3.1e}$$

where

$$J(r,\theta) = \frac{Im}{2\pi a}\,\frac{\text{ber mr} + i\,\text{bei mr}}{\text{bei' ma} - i\,\text{ber' ma}} \tag{3.1f}$$

and I is the total current (A), a is the outer radius (m), $m = \{2\pi f\mu\mu_0/\rho(\theta)\}^{1/2}$ (m^{-1}), f is the frequency (Hz), $\rho(\theta) = \rho_0(1 + \alpha\theta)$ is the temperature dependent resistivity (Ωm), μ is the relative permeability of the conductor (dimensionless), μ_0 is the permeability of free space (Ω sec m^{-1}), and {ber, bei} are the Kelvin functions (Abramowitz & Stegun, [1], section 9.9).

The problem (3.1) required a numerical solution for a number of different conductors for various values of the parameters h and q in the boundary condition (3.1d). (This boundary condition is an empirical model of heat dissipation by convection under various ambient wind conditions as in Morgan, [14]). A separation of variables solution of problem (3.1) is possible if the temperature dependence is neglected in the coefficients (3.1b) and the heating (3.1c) and if the index q in (3.1d) is unity. This analytic solution served as one check on the program. A further check on the complete program with all temperature dependent terms retained was that the heat input should balance the surface dissipation in the ultimate steady state. In all applications, this balance was accurate to within 0.2%.

Once the heating function $Q(\theta,t)$ was understood and coded, it took less than a day to design and implement a numerical solution of problem (3.1) using PDEONE. As expected, the execution speed of the program depended on the method chosen in the ODE integrator GEARB : with the best setting for the method flag (MF), the MOL solution computed the initial temperature rise 40 times faster than a forward time centre space (FTCS) solution of comparable accuracy; whilst the worst setting for MF gave an execution speed about 4 times slower than the FTCS solution.

The execution speeds are summarized in Table 1, and typical values of

the surface temperature are plotted in Figure 1 for various values of the index q in the boundary condition (3.1d). It should be noted that the execution speed of the MOL solution for a given accuracy depends on the automatic timestep control available in GEARB, and the timesteps become bigger once the initial heating is over. This feature is not available in the FTCS solution.

Table 1. Comparison of c.p.u. time for the numerical solution of problem (3.1) for the first 60 seconds heating in an aluminium conductor. Properties and constants as in the legend for Figure 1; ϵ is the error control setting on GEARB, MF denotes the method flag in GEARB.

method	c.p.u. time (s)	timestep	ϵ	comments
FTCS	126.5	0.01		timestep for stability
MF=10	510.2	0.006-0.032	10^{-1}	less accurate than FTCS
MF=12	12.0	0.287-3.20	10^{-6}	superior accuracy to FTCS
MF=12	3.0	3.98-16.9	10^{-4}	superior accuracy to FTCS
MF=13	53.4	0.114-1.01	10^{-5}	comparable accuracy to FTCS
MF=20	502.0	0.005-0.034	10^{-1}	less accurate than FTCS
MF=22	3.5	3.99-9.83	10^{-4}	superior accuracy to FTCS
MF=23	24.4	0.037-4.32	10^{-5}	comparable accuracy to FTCS

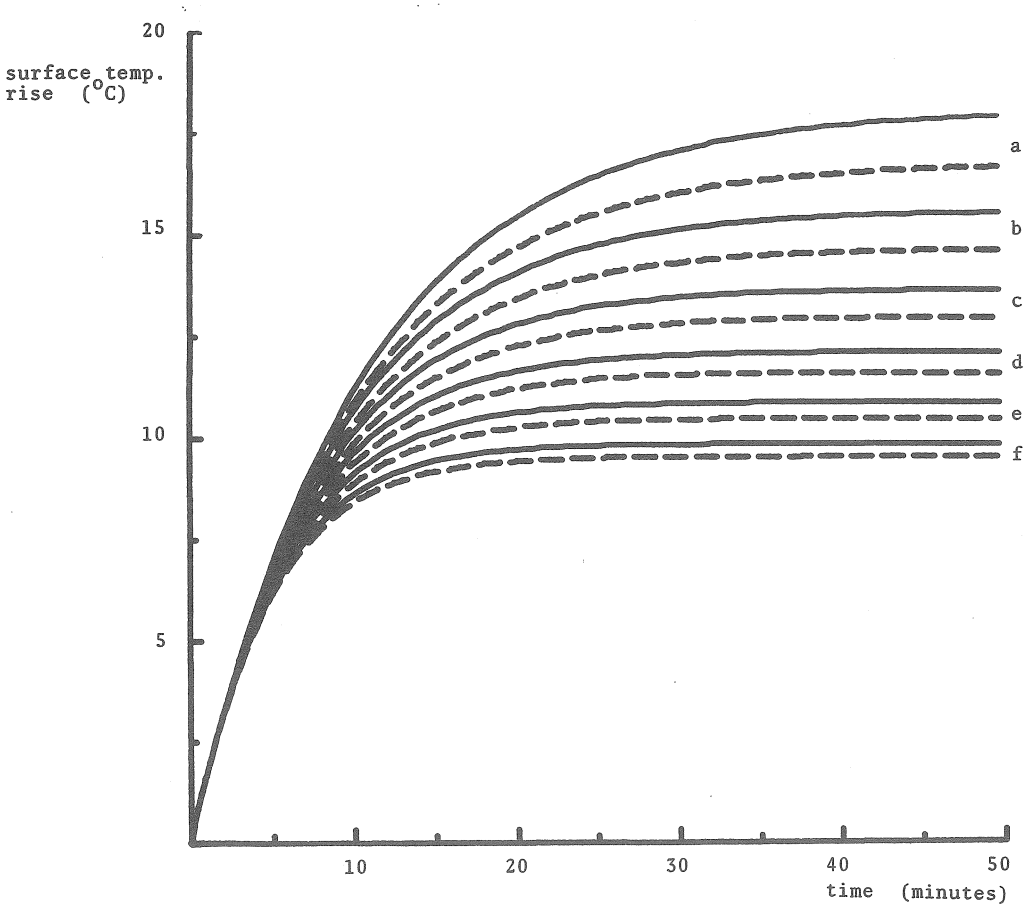

Figure 1. The surface temperature as a function of time in an aluminium conductor of radius 0.01m carrying a current of 500 A; number of radial points = 7; the constants (mksA units) are $\lambda_0 = 236$, $\kappa = 0.17\text{x}10^{-3}$, $\gamma_0 = 2.7\text{x}10^{3}$, $\delta = 0.72\text{x}10^{-4}$, $c_0 = 860$, $\beta = 0.38\text{x}10^{-3}$, $\alpha = 0.445\text{x}10^{-2}$, $\rho_0 = 0.263\text{x}10^{-7}$, h = 20; the dashed lines show the results for temperature independent coefficients ($\kappa = \delta = \beta = \alpha = 0$); (a) q = 1.00, (b) q = 1.05, (c) q = 1.10, (d) q = 1.15, (e) q = 1.20 (f) q = 1.25.

4. SOLUTE TRANSPORT PROBLEM

The second problem discussed is posed by the coupled dimensionless

quasi-linear parabolic PDEs

$$(\theta + \rho\frac{\partial s_1}{\partial c_1})\frac{\partial c_1}{\partial t} + \rho\frac{\partial s_1 \partial c_2}{\partial c_2 \partial t} = \frac{\partial^2 c_1}{\partial x^2} - \frac{\partial c_1}{\partial x}, \tag{4.1a}$$

$$\rho\frac{\partial s_2 \partial c_1}{\partial c_1 \partial t} + (\theta + \rho\frac{\partial s_2}{\partial c_2})\frac{\partial c_2}{\partial t} = D_2\frac{\partial^2 c_2}{\partial x^2} - \frac{\partial c_2}{\partial x}, \tag{4.1b}$$

which describe the concentration of two species of ions in a column of reactive soil through which water is flowing at a steady rate. In the equations, θ and D_2 are constants characteristic of the process, and $\rho s_1(c_1,c_2)$ and $\rho s_2(c_1,c_2)$ are instantaneous adsorption isotherms describing the amount of the ions adsorbed onto soil particles as a function of concentration. A description of the process is given by Charbeneau [6] and Barnes & Aylmore [2] reduce the general problem to the form given by (4.1a,b).

In the present application, the adsorption isotherms are taken to be

$$\rho s_1 = 0.1\, c_1\{1 + c_2 - \frac{1.25}{c_1+c_2}\}, \tag{4.2a}$$

$$\rho s_2 = -0.1\, c_2\{1 - c_1 - \frac{1.25}{c_1+c_2}\}, \tag{4.2b}$$

the boundary conditions are set to be

$$c_1(0,t) = 0.25, \quad c_2(0,t) = 0.75, \tag{4.2c}$$

$$c_{1_x}(x_\infty,t) = 0, \quad c_{2_x}(x_\infty,t) = 0 \tag{4.2d}$$

(where the subscript x denotes a partial derivative), and the PDEs are to be solved subject to the initial conditions

$$c_1(x,0) = 0.25, \quad c_2(x,0) = 0.25. \tag{4.2e}$$

Equations (4.1a,b) need to be cast in the form of (2.4a) prior to using PDEONE, and this is achieved by forming suitable linear combinations of (4.1a,b). Thus, the PDEs to be solved are found to be

$$\frac{\partial C_1}{\partial t} = \left[e_{11} + g_{11}\frac{\partial C_1}{\partial x} + g_{12}\frac{\partial C_2}{\partial x}\right]\frac{\partial C_1}{\partial x} + \left[e_{12} + g_{12}\frac{\partial C_1}{\partial x} + g_{13}\frac{\partial C_2}{\partial x}\right]\frac{\partial C_2}{\partial x} + \frac{\partial}{\partial x}\left[d_{11}\frac{\partial C_1}{\partial x} + d_{12}\frac{\partial C_2}{\partial x}\right], \tag{4.2f}$$

$$\frac{\partial C_2}{\partial t} = \left[e_{21} + g_{21}\frac{\partial C_1}{\partial x} + g_{22}\frac{\partial C_2}{\partial x}\right]\frac{\partial C_1}{\partial x} + \left[e_{22} + g_{22}\frac{\partial C_1}{\partial x} + g_{23}\frac{\partial C_2}{\partial x}\right]\frac{\partial C_2}{\partial x} + \frac{\partial}{\partial x}\left[d_{21}\frac{\partial C_1}{\partial x} + d_{22}\frac{\partial C_2}{\partial x}\right] \tag{4.2g}$$

where the coefficient functions $\underset{\sim}{e}$, $\underset{\sim}{g}$, $\underset{\sim}{d}$ involve lengthy but straightforward derivatives of S_1 and S_2 and the details are omitted.

The problem posed by equations (4.2) was solved numerically using PDE-ONE and GEARB. It took about two days to perform the preliminary analysis to arrive at equations (4.2f,g), and then another couple of days to write, test and debug the program. A test on the program was afforded by considering the associated problem for $C_0 = C_1 + C_2$ under the definitions (4.2a,b) for the adsorption isotherms: if D_2 is unity, C_0 satisfies

$$\alpha\frac{\partial C_0}{\partial t} = \frac{\partial^2 C_0}{\partial x^2} - \frac{\partial C_0}{\partial x} \qquad (\alpha \text{ constant}, \ 0 < x < x_\infty, \ t > 0), \tag{4.3a}$$

with

$$C_0(0,t) = 1.0, \quad C_{0_x}(x_\infty,t) = 0, \tag{4.3b}$$

$$C_0(x,0) = 0.5. \tag{4.3c}$$

This problem for C_0 possesses an analytic solution (van Genuchten & Alves, [18]) which was used to check the numerical method.

Experimentally, measurements of C_1 and C_2 would be made at various times at fixed $x = L \leq x_\infty$ in the belief that the results are not sensitive to the choice of x_∞ in (4.2d). These measurements are called breakthrough curves. The computer program is required to simulate the experiments, in

particular to examine the effects of varying the adsorption isotherms and the boundary and initial conditions. In practice, it was found the results were quite insensitive to the value chosen for x_∞ and breakthrough curves for $x_\infty = 1.6L$ are shown in Figure 2. The execution speed of the program again depended on the choice of the method flag in GEARB as shown in Table 2.

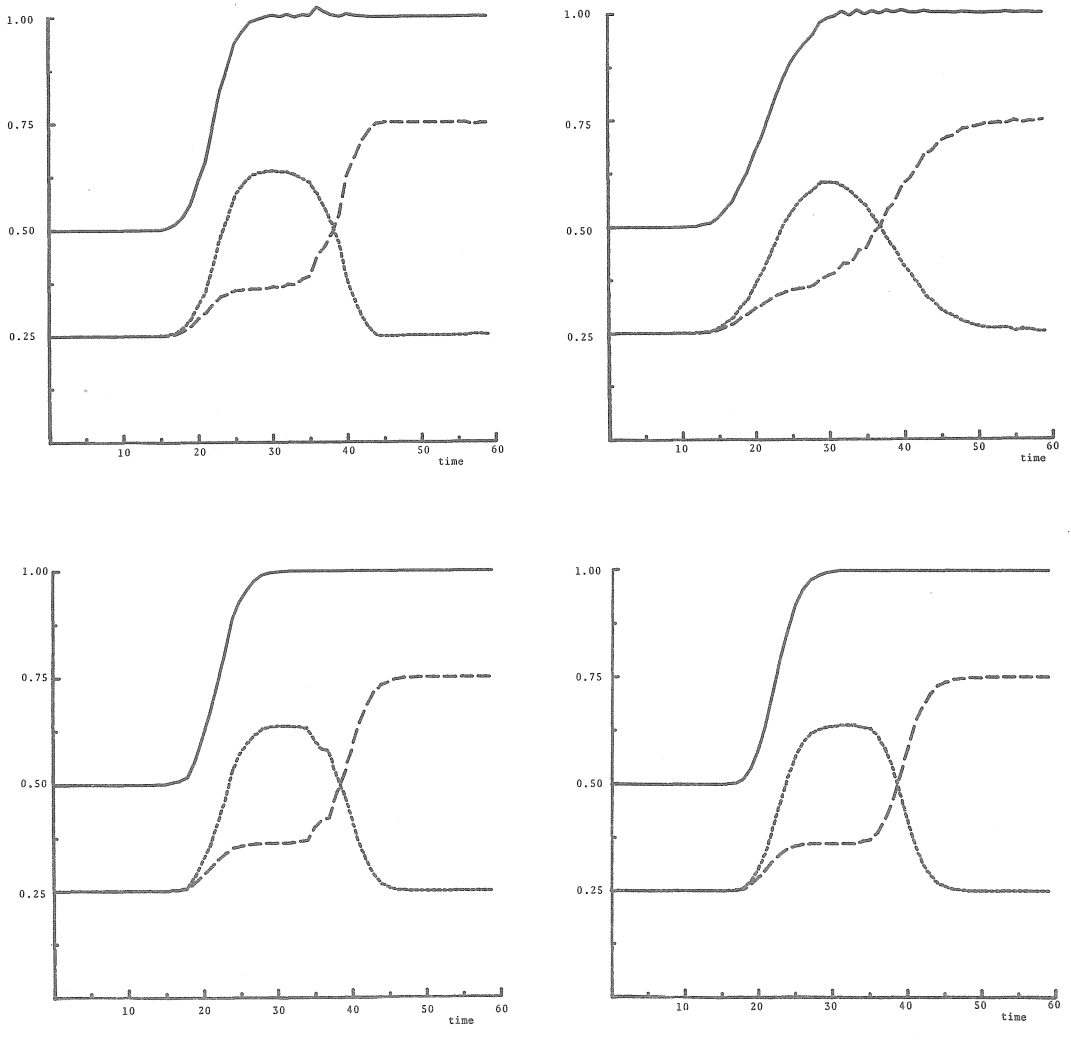

Figure 2. Caption on p.12.

Figure 2. Breakthrough curves for C_1, C_2 and $C_0 = C_1 + C_2$ at $x = L$ for the solutions of problem (4.1) subject to conditions (4.2a-e), $\epsilon = 10^{-6}$, $\theta = 0.25$, $D_2 = 1.0$, $L = 150$, $x_\infty = 240$: ..... C_1, - - - - C_2, ——— C_0. Top left, NPTS=61, c.p.u. time 179 sec; top right, NPTS=61, first derivative in PDEONE calculated using upwind differences, c.p.u. time 206 sec; bottom left, NPTS=121, c.p.u. time 565 sec; bottom right, NPTS=241, c.p.u. time 1623 sec.

Table 2. Comparison of c.p.u. time taken for the first 5 seconds of a test numerical solution of problem (4.1) using 16 points and with $x_\infty = L = 7.5$, ϵ (error tolerance in GEARB) is 10^{-3}, $\theta = 0.25$, $D_2 = 1$, MF denotes the method flag in GEARB.

non-stiff options			stiff options		
MF	c.p.u. time (sec)	timestep	MF	c.p.u. time (sec)	timestep
10	112.6	0.006-0.009	20	117.9	0.006-0.055
12	15.8	0.110-0.323	22	21.5	0.101-0.368
13	76.9	0.005-0.129	23	124.7	0.012-0.041

5. COMPUTATIONAL SHEAR DISPERSION

The final problem investigated is the parabolic PDE in time and two space dimensions

$$\frac{\partial C}{\partial t} + \frac{1}{2}(1-12y^2)\frac{\partial C}{\partial x} = P^{-2}\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}, \tag{5.1a}$$

subject to the boundary conditions

$$\frac{\partial C}{\partial y} = 0 \quad \text{at } y=\pm 1/2, \tag{5.1b}$$

$$C, \frac{\partial C}{\partial x} \to 0 \text{ as } |x| \to \infty \tag{5.1c}$$

and the initial condition

(5.1d) $$C(x,y,0) = \delta(x).$$

These non-dimensional equations describe the dispersion of a passive contaminant instantaneously injected into laminar flow between two flat plates at $y=\pm 1/2$. The co-ordinate x is measured in a frame moving at the discharge speed of the flow, and P is the Péclet number of the flow. This problem is a well known one in fluid mechanics and further references and the large time asymptotic behaviour have been given by Smith [16].

Problem (5.1) cannot be solved by PDEONE as it stands. Rather the Fourier transform

$$\hat{C}(\lambda,y,t) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} e^{i\lambda x} C(x,y,t)\,dx$$

is taken, and the symmetry of the solution about y=0 is used to derive coupled PDEs for the real and imaginary parts of $\hat{C}=\hat{C}_R+i\hat{C}_I$ in $0<y<1/2$, that is,

(5.2a) $$\frac{\partial \hat{C}_R}{\partial t} = -\frac{\lambda}{2}(1-12y^2)\hat{C}_I - \frac{\lambda^2}{P^2}\hat{C}_R + \frac{\partial^2 \hat{C}_R}{\partial y^2},$$

(5.2b) $$\frac{\partial \hat{C}_I}{\partial t} = \frac{\lambda}{2}(1-12y^2)\hat{C}_R - \frac{\lambda^2}{P^2}\hat{C}_I + \frac{\partial^2 \hat{C}_I}{\partial y^2}$$

subject to

(5.2c) $$\frac{\partial \hat{C}}{\partial y} = 0 \text{ at } y=0, 1/2,$$

(5.2d) $$\hat{C}(\lambda,y,0) = (2\pi)^{-1/2} \text{ for all } \lambda.$$

This problem (5.2) was solved numerically NLAM times for λ values between 0 and λ_{max} inclusive, and the solution C(x,y,t) was obtained by a numerical inversion of the Fourier transform. It was an easy matter to implement the numerical procedures and results for low values of the Péclet number were

available on the day the project was started. The high Péclet number cases were more difficult computationally since (5.2) then had to be solved for ever-larger values of λ. Nevertheless, the quality of results available can be seen from Figure 3 which is a plot of the cross-sectional average $\bar{C}$ for P=1000 and at various times after injection of contaminant.

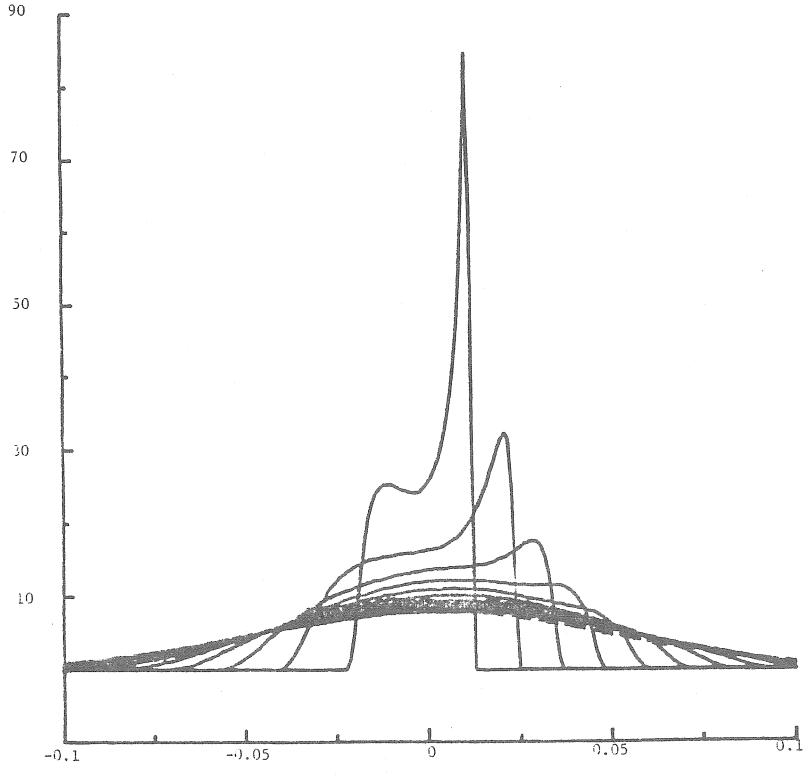

Figure 3. Plots of $\bar{C}$ for t=0.025(0.025)0.25 for the problem (5.1) with P=1000. Settings for GEARB and PDEONE: MF=12, $\epsilon=10^{-6}$, NPTS=21, NPDE=2; also NLAM=301 and λ_{max}=4500 (except for t=0.025 for which NLAM=501 and λ_{max}=7500 were used). The dashed line shows the (Gaussian) asymptotic profile at t=0.025.

The results displayed in Figure 3 required about three hours c.p.u. time on the VAX 11/750, and this is not inordinately large given the nature of the computation and the quality of the results. Two checks were performed on the numerical solutions. First, the mass of contaminant was conserved to better than 0.1% during the computation; and second, the numerical solution agreed well with an asymptotic solution [16]. Moreover, the numerical solutions do not suffer obviously from discretization errors or Gibbs ripples.

6. DISCUSSION

Sincovec & Madsen [15] begin their general comments on PDEONE with "We have been using the numerical method of lines approach for solving PDEs for some time and in general have found the method to be quite powerful, reasonably efficient from a computer time point of view, and extremely versatile and easy to implement for most problems. ... When our interface is used with one of the recently developed stiff ODE integrators, we feel that one has a reasonably robust piece of software for obtaining numerical solutions for fairly broad classes of problems." My experiences with PDEONE supports these claims.

The number of packages available for the MOL has increased greatly since 1975 when Sincovec & Madsen wrote "Measuring the efficiency and effectiveness of general purpose software is extremely difficult if not impossible... Comparable software against which comparisons could be made is not readily (if at all) available". Since then, a number of packages based on the MOL have been introduced. These have been summarized by Machura & Sweet [11] who rank PDEONE as the second lowest in complexity and power of the nine mentioned. Other packages include PDETWO (Melgaard &

Sincovec, [13]) for problems with two space dimensions, PDECOL (Madsen & Sincovec, [12]) where the spatial approximation is by B-splines, and FORSIM VI (Carver et al., [5]) with options for more sophisticated (e.g. upwind) differencing. These packages are designed to solve a different class of problems from PDEONE and a detailed comparison of their relative efficiency has not been attempted.

For computing efficiency, it is essential to exploit the structure of the system of ODEs by using an ODE integrator designed for banded systems. For robustness, it is also essential that the ODE integrator has a stiff option (even though the fastest execution speeds for the problems described above were obtained using non-stiff methods). On the efficiency of PDEONE, Sincovec & Madsen remark that "General purpose software such as PDEONE is typically aimed at reducing the 'human time' spent rather than computer time. Obviously, for any specific problem, one could write a more efficient program for its solution." Based on the author's experience with the transmission line problem, numerical solutions using PDEONE/GEARB are to be strongly preferred over explicit FTCS solutions. Moreover, in the transmission line problem, as with many problems, it was impractical to develop an implicit method of solution which would permit longer timesteps. For such cases, it is hardly worthwhile trying to develop a better method of solution than PDEONE/GEARB. Finally, the hybrid approach to the shear dispersion example discussed in Section 5 shows again the flexibility of the MOL and the possibility of using it as a building block in solving complicated problems.

The author would like to thank Dr J.H. Knight for his generous assistance with many aspects of this work.

REFERENCES

[1] M. Abramowitz and I.A. Stegun, Handbook of mathematical functions, Dover, New York, 1965.

[2] C.J. Barnes and L.A.G. Aylmore, A theoretical treatment of the effects of ionic and non-ionic competitive adsorption during solute transport in soils, Aust. J. Soil. Res. to appear.

[3] M. Bieterman and I. Babuska, The finite element method for parabolic equations, I. A posteriori error estimation, Numer. Math. 40 (1982a) 339-371.

[4] M. Bieterman and I. Babuska, The finite element method for parabolic equations, II. A posteriori error estimation and adaptive approach, Numer. Math. 40 (1982b) 373-406.

[5] M. Carver et al., The FORSIM VI simulation package for the automated solution of arbitrarily defined partial differential and/or ordinary differential equation systems. Rep. AECL 5821, Chalk River Nuclear Laboratories, Ontario, Canada, 1978.

[6] R.J. Charbeneau, Groundwater contaminant transport with adsorpton and ion exchange chemistry: method of characteristics for the case without dispersion, Water Resources Res. 17 (1981) 705-713.

[7] H.B. Dwight, Electrical coils and conductors, McGraw-Hill, New York, 1945.

[8] G. Fairweather and A.V. Saylor, On the application of extrapolation, deferred correction and defect correction to discrete-time Galerkin methods for parabolic problems, IMA J. Num. Anal. 3 (1983) 173-192.

[9] D. Gottlieb and S.A. Orszag, Numerical analysis of spectral methods, SIAM, Philaelphia, 1977.

[10] A.C. Hindmarsh, GEARB : solution of ordinary differential equations having banded Jacobian, Rep. UCID-30059, Lawrence Livermore Lab., Livermore, Calif., 1977.

[11] M. Machura and R.A. Sweet, A survey of software for partial differential equations, ACM Trans. Math. Software 6 (1980) 461-488.

[12] N.K. Madsen and R.F. Sincovec, PDECOL, general software for partial differential equations, ACM Trans Math Software 5 (1979) 326-351.

[13] D.K. Melgaard and R.F. Sincovec, General software for two-dimensional nonlinear partial differential equations, ACM Trans Math Software 7 (1981) 106-125.

[14] V.T. Morgan, Rating of bare overhead conductors for continuous currents, Proc. IEE 114 (1967) 1473-1482.

[15] R.F. Sincovec and N.K. Madsen, Software for nonlinear partial differential equations, ACM Trans Math. Software 3 (1975) 232-260; PDE-ONE, solutions of systems of partial differential equations, ACM Trans Math. Software 3 (1975) 261-263.

[16] R. Smith, A delay-diffusion description for contaminant dispersion, J. Fluid Mech. 105 (1981) 469-486.

[17] V. Thomée and L. B. Wahlbin, Maximum-norm stability and error estimates in Galerkin methods for parabolic equations in one space variable, Numer. Math. 41 (1983) 345-371.

[18] M. Th. van Genuchten and W.J. Alves, Analytical solutions of the one-dimensional convective-dispersive solute transport equation, Tech. Bull. 1661, U.S. Dept. Agric. Res. Service, 1981.

Division of Mathematics and Statistics
C.S.I.R.O.
PO Box 218
LINDFIELD NSW 2070