Int. J. Engng Ed. Vol. 10, No. 2, pp. 208–214, 1994
Printed in Great Britain.

0742–0269/91 $3.00+0.00

Method of Characteristics Using an Electronic Spreadsheet*

C. Y. LAM
School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 2263

This paper describes an educational spreadsheet program to solve the hyperbolic one-dimensional wave equation by the method of characteristics. The program was developed using the Lotus 1-2-3 spreadsheet software, but does not require users to have any knowledge of the spreadsheet. Apart from being fully interactive, a powerful interactive graphics feature has also been incorporated which can be used to execute the 'what-if?' analysis graphically. These features are illustrated by a numerical example.

INTRODUCTION

OVER THE years, apart from the original commercial intention of electronic spreadsheets, such software has also been applied to various scientific and engineering problems [1–7]. Recently, the author [6, 7] successfully demonstrated the suitability of spreadsheet programs in computing the solutions of partial differential equations. As far as the learning effectiveness of the numerical methods is concerned, those spreadsheet programs have been shown to offer greater educational value compared to traditional computer programs in a language such as FORTRAN. Their main advantages over the traditional programs are: (i) they are ready to run, so that keying-in, compiling and debugging are not required; (ii) they are interactive and friendly with substantial error and help messages; (iii) intermediate iterates are available for iterative methods; and (iv) they are able easily to execute interactive graphics for 'what-if?' analysis. The numerical methods implemented in those spreadsheet programs are the Gauss–Seidel method, the successive over-relaxation method and the alternating direction implicit method for the elliptic equation; the explicit forward time central space method and the implicit Crank–Nicolson method for the parabolic equation; and the explicit central difference method for the hyperbolic equation. However, all of these methods belong to the class of finite-difference method.

In view of the success of the earlier work [6, 7], further work has been pursued using the same spreadsheet approach. A spreadsheet program capable of solving the hyperbolic one-dimensional wave equation by the method of characteristics (MOC), which is not a finite-difference method, has been developed. In addition to all the features and advantages of its predecessors, the interactive graphics feature, which was not implemented in its predecessors, has also been implemented in the current program so that it can be used without spreadsheet knowledge. As the MOC is classical and commonly taught in courses related to the numerical solutions of partial differential equations in various disciplines, the current spreadsheet programs, together with those developed previously [6,7], form a useful set of programs that can be used in such courses as tools to aid the students to grasp the concepts of the numerical methods.

PROBLEM FORMULATION

The hyperbolic one-dimensional wave equation is

$$\frac{\partial^2 u}{\partial^2 t} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{1}$$

where c (= constant) is the wave propagation speed, $u = u(x, t)$ is the dependent variable, t and x are the time and spatial coordinates, respectively. To compute the numerical solution, two Dirichlet boundary conditions, $u(0, t)$ and $u(L, t)$, and two initial conditions, $u(x, 0)$ and $\partial u(x, 0)/\partial t$, are specified.

The MOC implemented in the current spreadsheet program can be found in many standard texts (e.g. [8–11]). The basic numerical steps and expressions used in the current spreadsheet program are briefly described here.

For the one-dimensional wave equation, the C− and C+ characteristics are straight lines with constant slopes $\pm 1/c$, respectively. For a constant step size $\mathrm{d}x$ in x, the step size $\mathrm{d}t$ in t is thus $\mathrm{d}x/2c$ and the computational domain is discretized by a characteristics mesh as shown in Fig. 1.

* Paper accepted 15 November 1993.

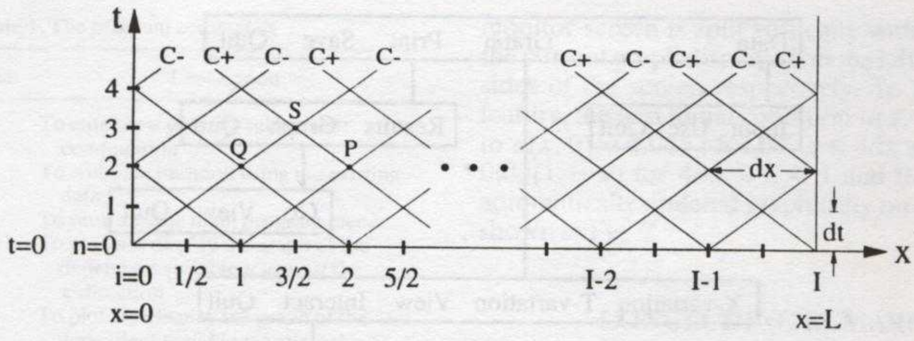


Fig. 1. The characteristics mesh for the one-dimensional wave equation.

For this mesh, the locations of the mesh points are

$$x^n_{i+1/2} = \frac{1}{2}(x^n_i + x^n_{i+1}) \quad i = 0, 1, 2, \ldots; n = 1, 3, \ldots \tag{2a}$$

$$x^n_i = x^0_i \quad i = 0, 1, 2, \ldots; n = 2, 4, \ldots \tag{2b}$$

$$t^n_i = \frac{n\mathrm{d}x}{2c} \quad n = 1, 2, \ldots \tag{2c}$$

where the subscript and superscript denote the x- and t-stations, respectively, as indicated in Fig. 1.

The numerical solutions at all mesh points can be obtained by a marching procedure starting from the mesh points on the x-axis at which $u(x, 0)$, $\partial u(x, 0)/\partial x$ and $\partial u(x, 0)/\partial t$ are known. The values of $u(x, 0)$ and $\partial u(x, 0)/\partial t$ are actually the two prescribed initial conditions, and the values of $\partial u(x, 0)/\partial x$ are obtained by numerical differentiation of the first initial condition $u(x, 0)$. For the numerical differentiation, the program uses the three-point forward and backward differences at $(0, 0)$ and $(L, 0)$, respectively, and the three-point central difference at other mesh points on the x-axis. During each marching step, the ordinary differential equations originated from equation (1) along the characteristics are integrated.

When the C− and C+ characteristics intersect at S, this yields:

$$(u_x)_S = \frac{1}{2}\left\{(u_x)_Q + (u_x)_P + \frac{1}{c}[-(u_t)_Q + (u_t)_P]\right\} \tag{3a}$$

$$(u_t)_S = \frac{1}{2}\{c[-(u_x)_Q + (u_x)_P] + (u_t)_Q + (u_t)_P\} \tag{3b}$$

and

$$u_S = \frac{u_Q + u_P}{2} + \frac{\mathrm{d}x}{8}\left\{(u_x)_Q - (u_x)_P + \frac{1}{c}[2(u_t)_S + (u_t)_Q + (u_t)_P]\right\} \tag{3c}$$

where P, Q and S are mesh points as shown in Fig. 1; the x and t subscripts denote the first partial derivatives with respect to x and t, respectively.

When the C+ characteristics from a mesh point P hit $x = 0$ at S:

$$(u_x)_S = (u_x)_P - \frac{1}{c}[(u_t)_S - (u_t)_P] \tag{4a}$$

$$(u_t)_S = \frac{\mathrm{d}u(0, t)}{\mathrm{d}t} \tag{4b}$$

and the value of u_S follows directly from the boundary condition $u(0, t)$.

When the C− characteristics from a mesh point Q hit $x = L$ at S:

$$(u_x)_S = (u_x)_Q + \frac{1}{c}[(u_t)_S - (u_t)_Q] \tag{5a}$$

and

$$(u_t)_S = \frac{\mathrm{d}u(L, t)}{\mathrm{d}t} \tag{5b}$$

and the value of u_S follows directly from the boundary condition $u(L, t)$.

THE SPREADSHEET PROGRAMS

Like its predecessor, the current spreadsheet program was developed using the popular spreadsheet software Lotus 1-2-3, release 3.1 [12]. The entire program was written in macro commands in order to facilitate user friendliness, interactiveness and the advanced interactive graphics feature.

The spreadsheet program developed is menu driven. The program menu tree is shown in Fig. 2 and the corresponding program commands are described in Table 1. The menu structure and commands are very close to those for the program developed previously [6, 7]. The additional feature that is now built into the program is the interactive graphics, under the program command Graph Interact, that enhances user-friendliness in using this feature, which allows 'what-if?' analysis to be done graphically. All tasks can be done within the program menu. The way of running the program is identical to that of its predecessors, which was described in [6, 7]. The friendliness, interactiveness and extensive error and help messages make the program easily accessible to users with no spreadsheet knowledge. The only necessary requirements from the user are to run the Lotus 1-2-3 program by keying-in 123 followed by the Enter key under

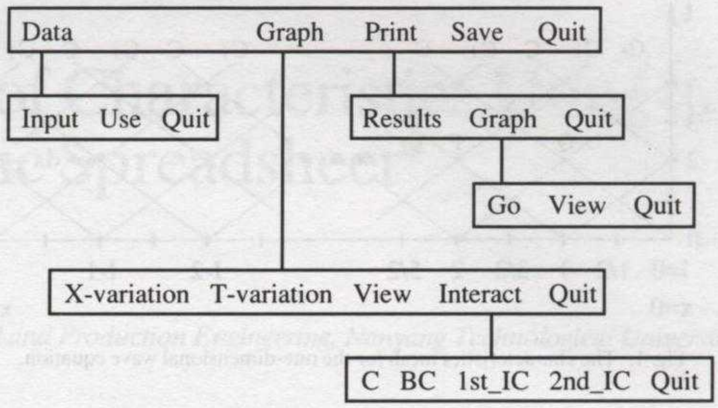


Fig. 2. The program menu tree.

the DOS prompt, to use the 1-2-3 command/File Retrieve to load the spreadsheet program into memory and to use the 1-2-3 command/Quit which ends 1-2-3 and returns to DOS. As a result, effort can be focused on the numerical aspects of MOC without distraction.

Example

Consider the problem of finding the deflection $u(x, t)$ governed by equation (1) of an elastic string of length $L = 1$ m and $c = 1$ m/s. The string is held fixed at both ends and released from rest with an initial deflection $u(x, 0) = \sin(\pi x)/15$.

Using the program command Data Input, the data c, L, I (the number of x-intervals, taken to be 5), the two prescribed boundary conditions $u(0, t)$ and $u(L, t)$ and the two prescribed initial conditions $u(x, 0)$ and $u_t(x, 0)$ are entered as shown in Fig. 3. The values of dx and dt in Fig. 3 are computed by the program. It should be noted that in Fig. 3 and the latter figures, the worksheet areas for displaying the values of the boundary conditions, the initial conditions, $\partial u(x, 0)/\partial x$ and the computed results are set automatically by the program. Also, these values are shown up to $x = 0.5$ for this example due to the limited resolution supported by the monitor screen. After the data are entered, the $\partial u(x, 0)/\partial x$ values that are required to start the marching procedure are computed automatically, as shown in Fig. 4. The numerical results for 15 t-steps are obtained and shown in Fig. 5.

Upon invoking the program commands Graph X-variation and Graph T-variation and following through a series of questions presented by the program, which are subsequently answered, the computed results are displayed graphically on the screen as shown in Fig. 6. It shows that the period of vibration of the string is about 2 s.

Using the program command Graph Interact, the interactive graphics mode is activated and the

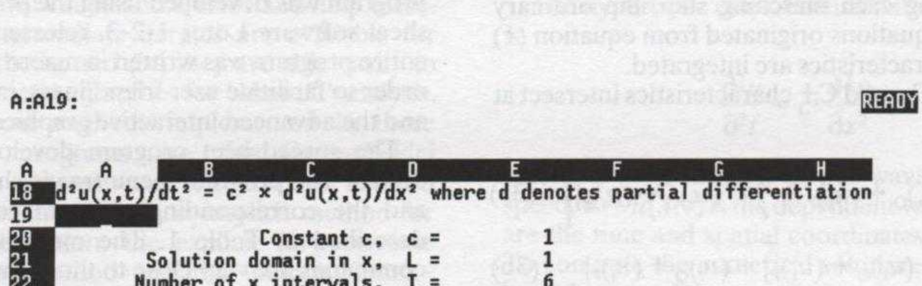


Fig. 3. The input data.

Table 1. The program commands

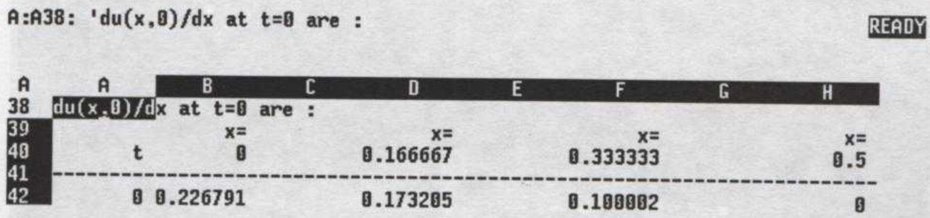
Program command	Description
Data Input	To enter new data for subsequent computation
Data Use	To continue iteration using the existing data
Data Quit	To return to the main program menu
Graph X-variation	To plot and display the graph of the dependent variable u against the x-direction
Graph T-variation	To plot and display the graph of the dependent variable u against the t-direction
Graph View	To display the current graph as and when the user desires
Graph Interact	To invoke the interactive graphics mode for graphical 'what-if?' analysis
Graph Interact [*list*]	To change a parameter in [*list*] in interactive graphics mode
Graph Interact Quit	To clear the interactive graphics mode and return to the previous menu
Graph Quit	To return to the main program menu
Print Results	To send the input data and the computed results to a printer
Print Graph Go	To send the current graph to a printer
Print Graph View	To display the current graph. This allows the user to view the graph before printing
Print Graph Quit	To return to the previous program menu
Print Quit	To return to the main program menu
Save	To save the spreadsheet program with the existing data and results in a file
Quit	To return to 1-2-3's READY mode

monitor screen is split vertically with the text and the current graph displayed on the left and the right sides of the screen respectively. To illustrate this feature, the first initial condition $u(x, 0)$ is changed to $u(x, 0) = 0.015x$ for $0 \leqslant x \leqslant 4dx$ and $u(x, 0) = 0.03(1 - x)$ for $4dx \leqslant x \leqslant 1$ and the results are automatically updated graphically on the screen as shown in Fig. 7.

CONCLUDING REMARKS

A spreadsheet program has been developed to solve the one-dimensional wave equation by the MOC. The program uses constant step sizes and constant Dirichlet boundary conditions and is compatible with the Lotus 1-2-3 release 3 or above. It is menu driven, user friendly and interactive. Automatic error detection and extensive error and help messages have been incorporated. The program can therefore be used without much spreadsheet knowledge. The built-in powerful interactive graphics feature allows numerical experiments to be performed graphically with ease. These features, which are difficult or impossible to achieve by the traditional programs, allow students to concentrate on the numerical aspects of the MOC and therefore help them to understand the numerical method in a shorter time.

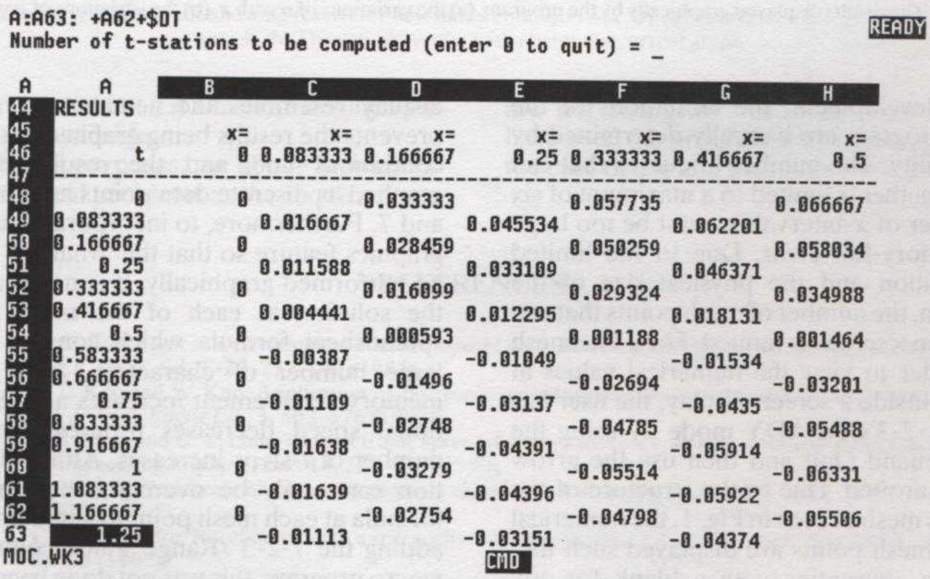
Since the Lotus 1-2-3 spreadsheet software was



A:A38: 'du(x,0)/dx at t=0 are : READY

A	A	B	C	D	E	F	G	H
38	du(x,0)/dx at t=0 are :							
39		x=		x=		x=		x=
40	t	0		0.166667		0.333333		0.5
41	------							
42	0	0.226791		0.173205		0.100002		0

Fig. 4. The values of $\partial u(x, 0)/\partial x$ as computed automatically.

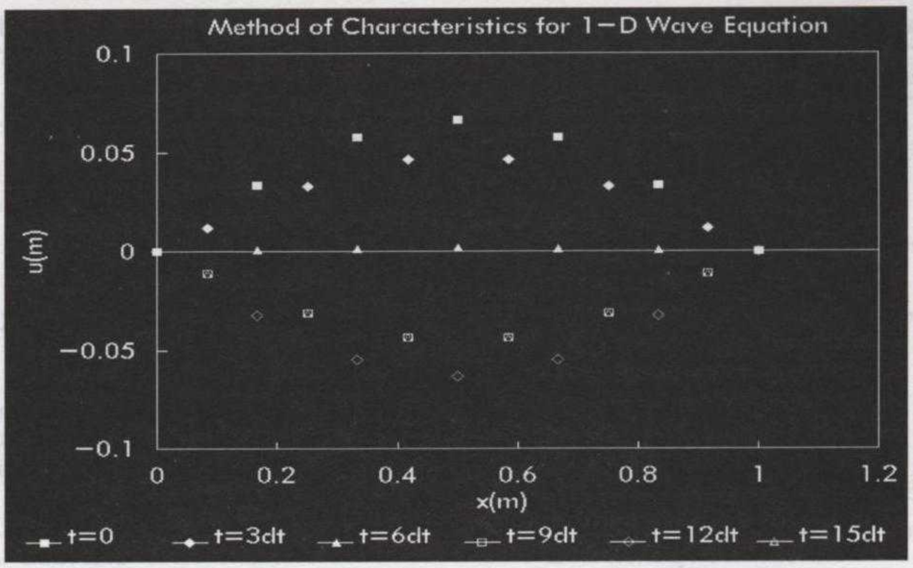


A:A63: +A62+$DT READY
Number of t-stations to be computed (enter 0 to quit) = _

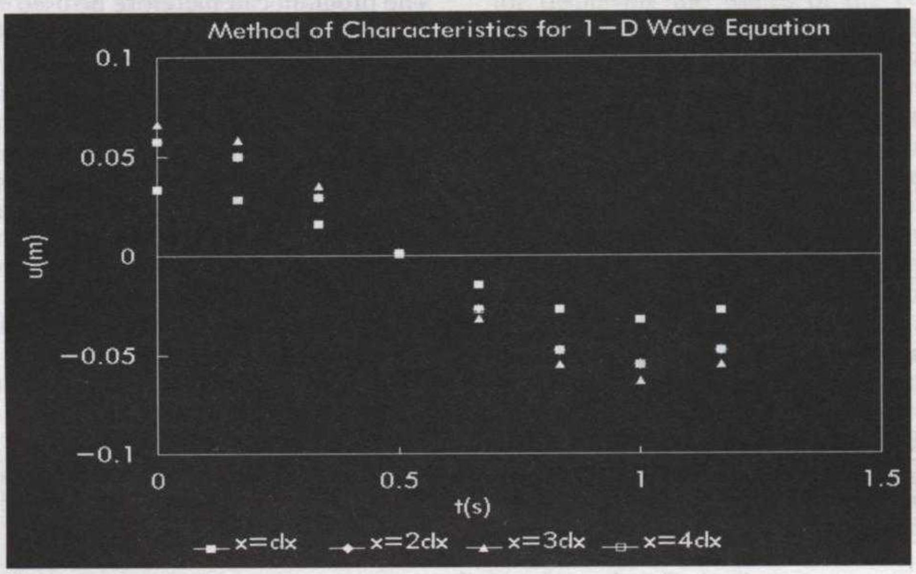
A	A	B	C	D	E	F	G	H
44	RESULTS							
45		x=	x=	x=	x=	x=	x=	x=
46	t	0	0.083333	0.166667	0.25	0.333333	0.416667	0.5
47	------							
48	0	0		0.033333		0.057735		0.066667
49	0.083333		0.016667		0.045534		0.062201	
50	0.166667	0		0.028459		0.050259		0.058034
51	0.25		0.011588		0.033109		0.046371	
52	0.333333	0		0.016099		0.029324		0.034988
53	0.416667		0.004441		0.012295		0.018131	
54	0.5	0		0.000593		0.001188		0.001464
55	0.583333		-0.00387		-0.01049		-0.01534	
56	0.666667	0		-0.01496		-0.02694		-0.03201
57	0.75		-0.01109		-0.03137		-0.0435	
58	0.833333	0		-0.02748		-0.04785		-0.05488
59	0.916667		-0.01638		-0.04391		-0.05914	
60	1	0		-0.03279		-0.05514		-0.06331
61	1.083333		-0.01639		-0.04396		-0.05922	
62	1.166667	0		-0.02754		-0.04798		-0.05506
63	1.25		-0.01113		-0.03151		-0.04374	

MOC.WK3 CMD

Fig. 5. The results of the one-dimensional wave equation.



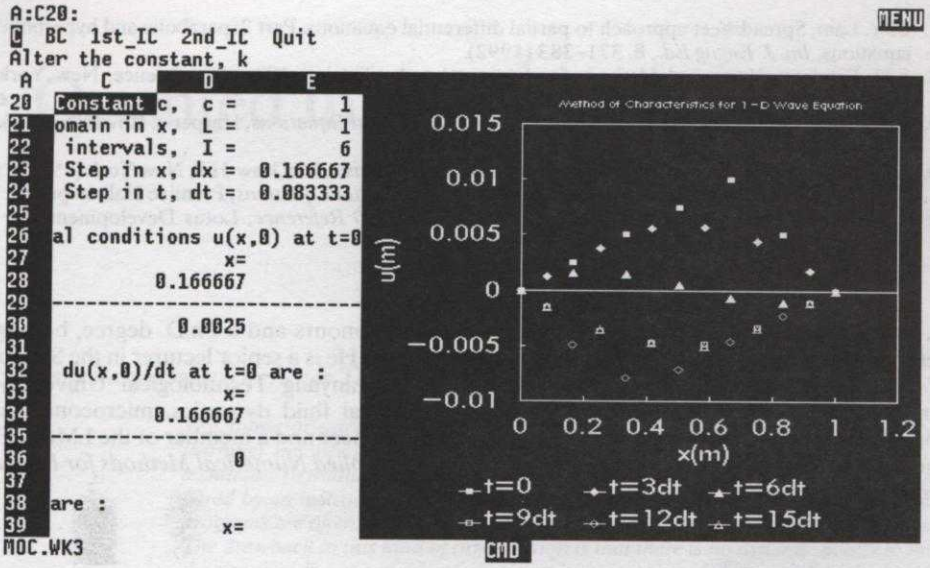
(a)



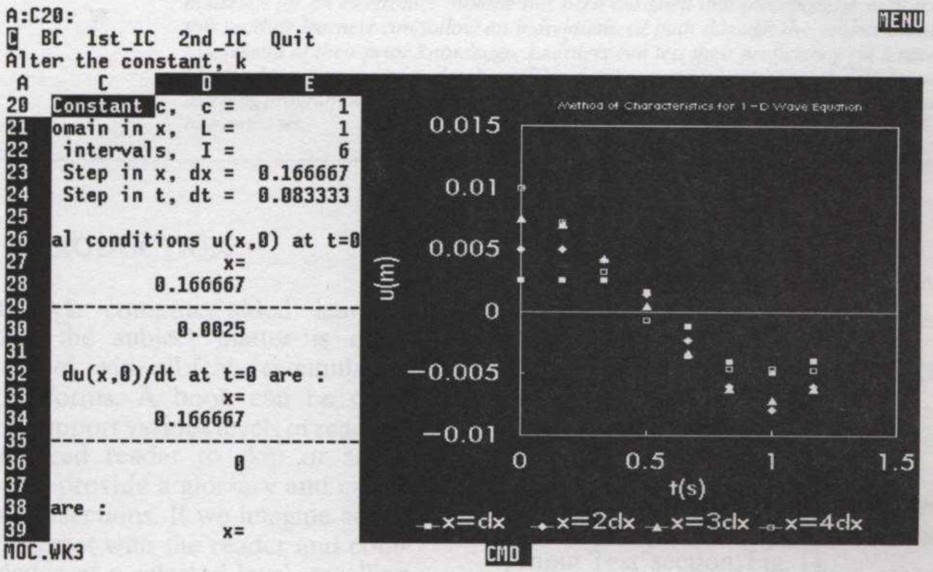
(b)

Fig. 6. The results displayed graphically by the program. (a) the variations of u with x. (b) the variations of u with t.

used in the development, the limitations of the spreadsheet program are basically determined by 1-2-3's capability. The number of curves that can be graphed together is limited to a maximum of six and the number of x-intervals cannot be too large to avoid memory-full error. Due to the limited display resolution and the physical size of the monitor screen, the number of mesh points that can be displayed on a screen is limited. For a fine mesh system, in order to view the numerical values at mesh points outside a screen display, the user has to return to 1-2-3's READY mode by using the program command Quit and then use the arrow keys to move around. Due to the structure of the characteristics mesh as seen in Fig. 1, the numerical results at the mesh points are displayed such that the values are alternating with a blank for any values of x and t as shown in Fig. 5. While this display resembles the actual mesh structure, it prevents the results being graphed in the form of a continuous line, and the results can only be graphed as discrete data points as shown in Figs 6 and 7. Furthermore, to incorporate the interactive graphics feature so that the 'what-if?' analysis can be performed graphically, it is necessary to retain the solution at each of the mesh points as a spreadsheet formula which consists of a rather large number of characters. As a result, the memory requirement increases and the computational speed decreases progressively when the number of t-steps increases. Although this limitation can easily be overcome by converting the formula at each mesh point to a numerical value by adding the 1-2-3 /Range Value command to the macro program, this was not done because it would disable the useful 'what-if?' analysis, while the



(a)



(b)

Fig. 7. The interactive graphics mode showing results for new initial condition $u(x, 0)$. (a) The variations of u with x being the current graph. (b) The variations of u with t being the current graph.

limitation can be alleviated if a faster machine with more memory is available.

For educational purposes, the above limitations do not pose any real problems as the requirements are not stringent. Feedback from students who have used the program are promising, especially in the ease of use and the interactive graphics feature.

REFERENCES

1. N. D. Rao, Typical applications of microcomputer spreadsheets to electrical engineering problems, *IEEE Trans. Ed.*, **E-27**, 237–242 (1984).
2. Y. L. Kuo and W. Kuo, Application of electronic spreadsheets to linear and integer programming, *Int. J. Appl. Engng Ed.*, **3**, 563–576 (1987).
3. S. K. Chan and C. Y. Lam, The electronic spreadsheet as a tool for course co-ordination in a school of engineering, *Comput. Ed.*, **14**, 231–238 (1990).
4. R. Kari, Spreadsheets in advanced physical chemistry, *J. Comput. Math. Sci. Teach.*, **10**, 39–48 (1990).
5. C. Pinter-Lucke, Rootfinding with a spreadsheet in pre-calculus, *J. Comput. Math. Sci. Teach.*, **11**, 85–93 (1992).
6. C. Y. Lam, Spreadsheet approach to partial differential equations. Part 1: elliptic equations, *Int. J. Engng Ed.*, **8**, 278–287 (1992).

7. C. Y. Lam, Spreadsheet approach to partial differential equations. Part 2: parabolic and hyperbolic equations, *Int. J. Engng Ed.*, **8**, 371–383 (1992).
8. J. H. Ferziger, *Numerical Methods for Engineering Application*, Wiley-Interscience, New York (1981).
9. P. DuChateau and D. Zachmann, *Applied Partial Differential Equations*, Harper & Row, New York (1989).
10. J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, McGraw-Hill, New York (1992).
11. C. Y. Lam, *Applied Numerical Methods for Partial Differential Equations*, Prentice Hall, in press.
12. Lotus Development Corporation, *Lotus 1-2-3 Release 3.0 Reference*, Lotus Development Corporation, Cambridge (1989).

C. Y. Lam holds a B.Sc. (Eng) degree with first-class honours and a Ph.D. degree, both in aeronautical engineering, from the University of London. He is a senior lecturer in the School of Mechanical and Production Engineering at the Nanyang Technological University, Singapore. His research interests include computational fluid dynamics, microcomputer applications and anthropometry. He is a Chartered Engineer and a member of the I.Mech.E. and AIAA. He is currently authoring a book entitled *Applied Numerical Methods for Partial Differential Equations*.