Int. J. Engng Ed. Vol. 11, No. 2, pp. 125–135, 1995
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Spreadsheet Implementation of the Lax–Wendroff Method*

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

A spreadsheet program to solve the hyperbolic one-dimensional wave equation by the Lax-Wendroff one-step method has been successfully developed by using the Lotus 1-2-3 spreadsheet package. The users do not require spreadsheet knowledge in order to use the program. The program is fully interactive and provides some unique features, such as a powerful interactive graphics capability, that are not normally achievable by conventional programming. These features are illustrated by a numerical example.

INTRODUCTION

ELECTRONIC spreadsheets, originally intended for commercial applications, have been applied to various scientific and engineering problems [1–6]. Lam [4–6] has demonstrated the application of spreadsheets in solving partial differential equations of the elliptic, parabolic and hyperbolic types. Spreadsheet programs have been proven to offer greater educational value in enhancing the learning effectiveness as compared to the conventional programs in high-level languages that are normally used in classrooms. The main advantages over the conventional programs are: (a) ready-to-run, so that keying-in, compiling and debugging are not required; (b) interactiveness and user-friendliness with substantial error and help messages; (c) intermediate iterates are available for iterative methods; and (d) able to execute interactive graphics for 'What-if?' analysis easily. Due to their proven advantages, these spreadsheet programs have been incorporated into a recent text [7] written by the author.

All previous spreadsheet programs [4–6] apply various numerical methods to second-order partial differential equations directly. To extend the spreadsheet approach to systems of first-order equations, the explicit Lax–Wendroff method has been implemented in a spreadsheet to solve the second-order one-dimensional wave equation. The Lax–Wendroff method is chosen because it is often included in related courses due to its ease of use, its second-order accuracy and because it forms the basis of other better methods. The one-dimensional wave equation is chosen for the current development because it can be easily broken into a pair of first-order equations and most people have some physical understanding of the string vibration problem which is discussed in a numerical example and is modelled by the one-dimensional wave equation. All the features and advantages of the previous programs, including the interactive graphics feature, are retained in the current spreadsheet program. Like its predecessors, it can thus be used without spreadsheet knowledge. It can be adopted in related courses as a better alternative tool (other than the conventional computer programs) to aid the students to understand the numerical aspects of the Lax–Wendroff method.

* Paper accepted 2 December 1994.

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 temporal 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 prescribed. The solution domain is shown in Fig. 1.

The Lax–Wendroff method is an explicit time-marching technique of second-order accuracy in time and space for solving systems of first-order partial differential equations. It is well-documented in many standard texts [8–10]. Therefore, only the basic numerical expressions and steps used in the current spreadsheet program are briefly described.

When the Lax–Wendroff method is used to solve the one-dimensional wave equation (1), the second-order equation is first split into a pair of coupled first-order equations as follows:

$$\frac{\partial p}{\partial t} = c \frac{\partial q}{\partial x} \tag{2a}$$

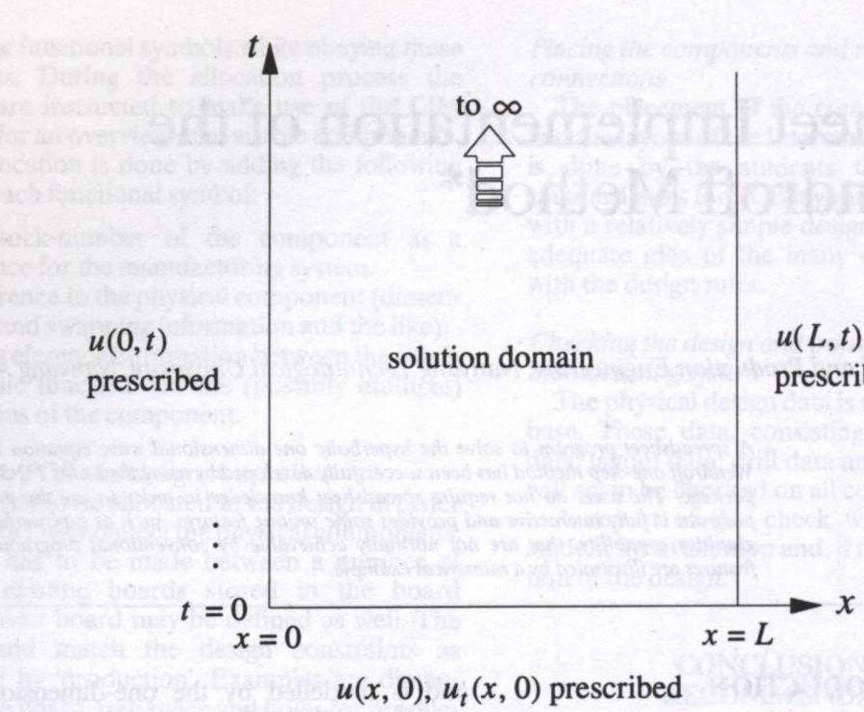


Fig. 1. The solution domain.

$$\frac{\partial q}{\partial t} = c\frac{\partial p}{\partial x} \tag{2b}$$

where

$$p(x, t) = c\frac{\partial u}{\partial x} \text{ and } q(x, t) = \frac{\partial u}{\partial t}. \tag{3}$$

Using Taylor series to expand $p(x, t)$ and $q(x, t)$ in time, combining with (2), and using central difference for the derivative terms yields

$$p_i^{n+1} = p_i^n + \frac{R}{2}(q_{i+1}^n - q_{i-1}^n) + \frac{R^2}{2}(p_{i+1}^n - 2p_i^n + p_{i-1}^n) \tag{4a}$$

and

$$q_i^{n+1} = q_i^n + \frac{R}{2}(p_{i+1}^n - p_{i-1}^n) + \frac{R^2}{2}(q_{i+1}^n - 2q_i^n + q_{i-1}^n) \tag{4b}$$

where R is the Courant number given by

$$R = \frac{c\Delta t}{\Delta x} \tag{5}$$

and the subscripts and superscripts denote the x- and t-stations respectively.

Using Taylor series to expand $u(x, t)$ in time, combining with (1) and (3), and using central difference for the derivative terms yields

$$u_i^{n+1} = u_i^n + \Delta t\, q_i^n + \frac{R\Delta t}{4}(p_{i+1}^n - p_{i-1}^n). \tag{6}$$

In (4) and (6), the third and higher derivative terms are truncated in the series. If the appropriate boundary conditions and initial conditions of p and q are known, (4) can be used to generate p and q in a time-marching manner. Figure 2 shows the required boundary conditions and initial conditions together with the grid system in the solution domain. Once p and q are known, the solution u of the one-dimensional wave equation then follows from (6).

In the first marching step from $n = 0$ to $n = 1$, for I x-intervals,

$$p_i^0 = c\frac{\partial u_i^0}{\partial x} \; (i = 0, 1, \ldots, I)$$

which are obtained by differentiating the prescribed initial condition u_i^0 with respect to x numerically. To be consistent with the second-order accuracy of the Lax–Wendroff method, the three-point forward difference, central difference and three-point backward difference are used for $i = 0$, $i = 1, \ldots, I - 1$ and $i = I$ respectively. Also,

$$q_i^0 = \frac{\partial u_i^0}{\partial t} \; (i = 0, 1, \ldots, I)$$

are obtained directly from the prescribed initial condition $\partial u_i^0/\partial t$. Using these in (6) and (4) yields u_i^1, p_i^1 and q_i^1 for $i = 1, \ldots, I - 1$. To obtain $p_i^1[= c(\partial u_i^1/\partial x)]$ for $i = 0$ and $i = I$, we again use three-point forward difference and three-point backward difference respectively to differentiate u_i^1 with respect to x numerically with the values u_0^1 and u_I^1 from the prescribed boundary conditions. To obtain $q_i^1[= \partial u_i^1/\partial t]$ for $i = 0$ and $i = I$, we numerically differentiate the prescribed boundary conditions u_0^n and u_I^n with respect to t at $n = 1$. In the present program, these boundary conditions are taken to be constants and thus $q_i^1 = 0$ for $i = 0$ and i

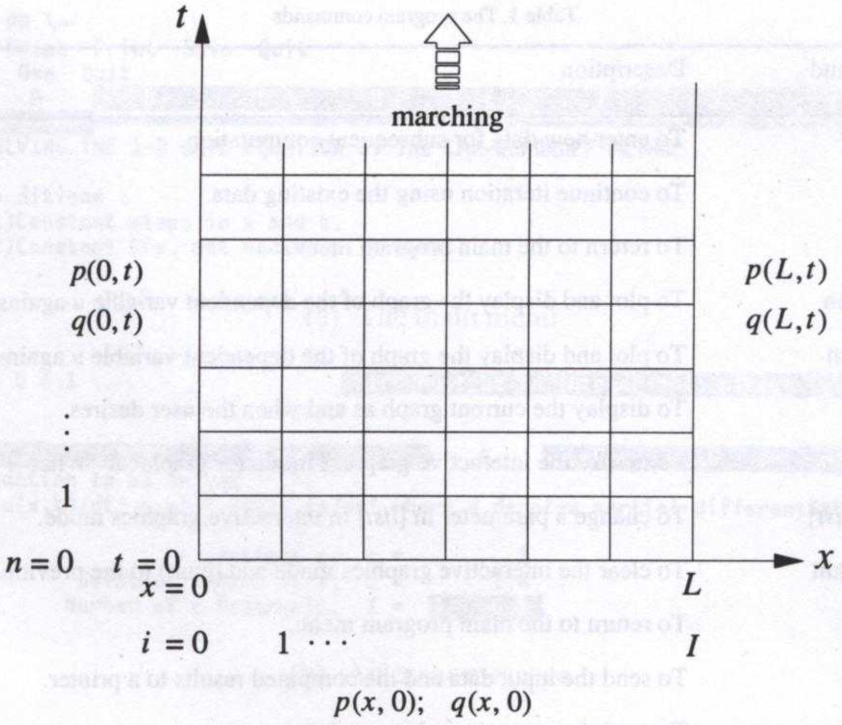


Fig. 2. The grid system and the boundary and initial conditions.

$= I$. The marching procedure can be continued to generate u_i^n at all interior grid points.

The Lax–Wendroff method is conditionally stable and the stability requirement [8–10] is

$$R = \frac{c\Delta t}{\Delta x} \leqslant 1$$

known as the Courant condition.

THE SPREADSHEET PROGRAM

The current spreadsheet program was developed by using the popular spreadsheet software Lotus 1-2-3 Release 3.1 [11]. 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. 3 and the corresponding program commands are described in Table 1. The menu structure and commands are the same as those for the programs developed previously [4–6]. The way of running the program is identical to that of its predecessors which was described in [4–6]. Due to the user-friendliness, interactiveness and the provision of extensive error and help messages, the program

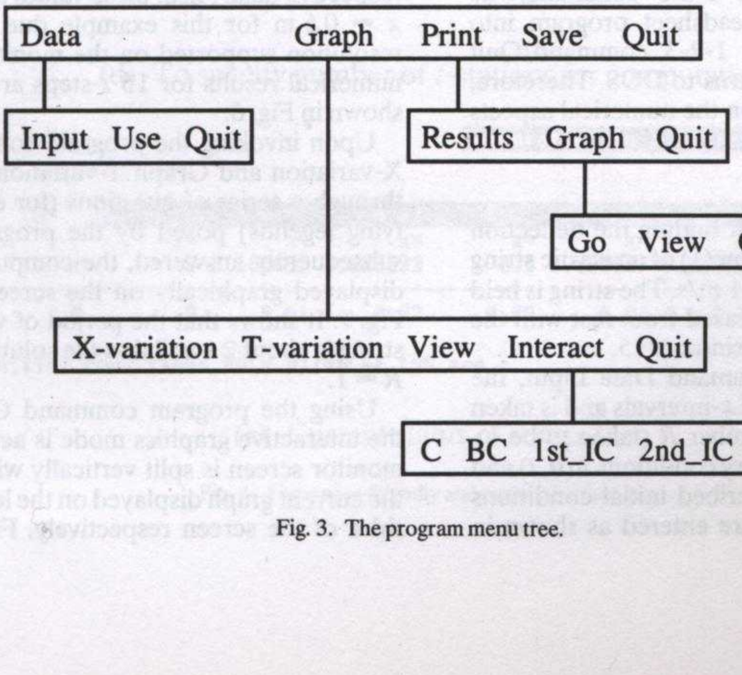


Fig. 3. The program menu tree.

Table 1. The program commands

Program Command	Description
Data Input	To enter new data for subsequent computation.
Date 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.

can be employed by users without spreadsheet knowledge. Figure 4 shows some of these messages as displayed on the top of the monitor screen (in 1-2-3's control panel). The only requirements from the user are to run the Lotus 1-2-3 program by keying-in 123 followed by the Enter key under 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. Therefore, the user can concentrate on the numerical aspects of the Lax–Wendroff method easily.

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 the 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 and is taken to be 12), the Courant number R (taken to be 1), the two prescribed boundary conditions $u(0, t)$ and $u(L, t)$ and the two prescribed initial conditions $u(x, 0)$ and $\partial u(x, 0)/\partial t$ are entered as shown in Fig. 5. The values of dx and dt are computed by the program. In Fig. 5 and the latter figures, the worksheet areas for displaying the values of the boundary conditions, the initial conditions and the computed results are set and displayed automatically by the program after confirmation of the respective data. Also, these values are shown up to $x = 0.5$ m for this example due to the limited resolution supported on the monitor screen. The numerical results for 15 t-steps are obtained and shown in Fig. 6.

Upon invoking the program commands Graph X-variation and Graph T-variation and following through a series of questions (for example, specifying legends) posed by the program which are subsequently answered, the computed results are displayed graphically on the screen as shown in Fig. 7. It shows that the period of vibration of the string is about 2 s and that the solution is stable for $R = 1$.

Using the program command Graph Interact, the interactive graphics mode is activated and the monitor screen is split vertically with the text and the current graph displayed on the left and the right sides of the screen respectively. Figures 8 and 9

```
A:A1: PR \=                                                              MENU
Data  Graph  Print  Save  Quit
Input  Use  Quit
 A     A        B        C        D        E        F        G        H
1    ==========================================================================
2    SOLVING THE 1-D WAVE EQUATION BY THE LAX-WENDROFF METHOD
3
4    Conditions :
5    (1)Constant steps in x and t.
6    (2)Constant BCs, not necessarily equal.
```

(a) The main menu

```
A:E17: U 6.1                          Entry must be an integer > 1. please re-enter !

 A     A        B        C        D        E        F        G        H
12   Equation to be solved :
13   d²u(x,t)/dt² = c² × d²u(x,t)/dx² where d denotes partial differentiation
14
15                        Constant c,  c =         1
16              Solution domain in x,  L =       1.4
17             Number of x intervals,  I =       6.1
```

(b) An error message

```
A:H26: U 0                                                              READY
Any changes for u(x,0)?  <Y/N>: _

 A     A        B        C        D        E        F        G        H
22   Boundary and initial conditions u(x,0) at t=0 are :
23                  x=       x=       x=       x=       x=       x=       x=
24         t         0 0.166667 0.333333      0.5 0.666667 0.833333        1
25   --------------------------------------------------------------------------
26         0         0      0.1      0.2      0.3      0.4      0.2        0
27
```

(c) Option to amend $u(x, 0)$

```
A:A38: U 0                                                              READY
Number of t-stations to be computed (enter 0 to quit) = _

 A     A        B        C        D        E        F        G        H
22   Boundary and initial conditions u(x,0) at t=0 are :
23                  x=       x=       x=       x=       x=       x=       x=
24         t         0 0.166667 0.333333      0.5 0.666667 0.833333        1
25   --------------------------------------------------------------------------
26         0         0     0.01     0.02     0.03     0.04     0.02        0
27
```

(d) To specify number of t-stations for computation

```
A:A36: U "t                           Computation in progress. please WAIT !

 A     A        B        C        D        E        F        G        H
23                  x=       x=       x=       x=       x=       x=       x=
24         t         0 0.166667 0.333333      0.5 0.666667 0.833333        1
25   --------------------------------------------------------------------------
26         0         0     0.01     0.02     0.03     0.04     0.02        0
27
28   Initial conditions du(x,0)/dt at t=0 are :
```

(e) Current status of computer

Fig. 4. Some user-friendly and interactive features.

```
A:B35: U "x=                                                           READY
Enter value of t (an INTEGER, in multiple of dt) for FIRST variation: _

 A      A        B        C        D        E        F        G        H
34 RESULTS
35                 x=       x=       x=       x=       x=       x=       x=
36        t         0 0.166667 0.333333      0.5 0.666667 0.833333        1
37 ------------------------------------------------------------------------
38        0         0     0.01     0.02     0.03     0.04     0.02        0
39 0.166667         0     0.01     0.02  0.02625   0.0325  0.01625        0
```

(f) To enter value of *t* for the first *x*-variation

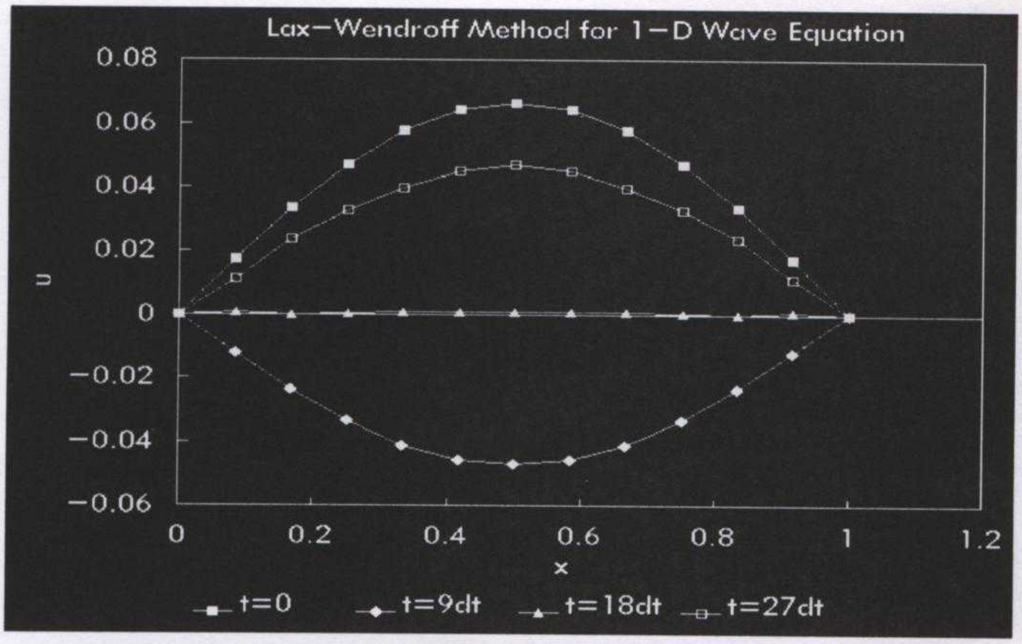
Fig. 4 (*cont.*)

```
A:A32: U 0                                                         READY

 A      A        B        C        D        E        F        G        H
15                    Constant c,  c =        1
16          Solution domain in x,  L =        1
17           Number of x intervals,  I =     12
18                Courant number,  R =        1
19                  Step in x, dx =  0.083333
20                  Step in t, dt =  0.083333
21
22 Boundary and initial conditions u(x,0) at t=0 are :
23                 x=       x=       x=       x=       x=       x=       x=
24        t         0 0.083333 0.166667     0.25 0.333333 0.416667      0.5
25 ------------------------------------------------------------------------
26        0         0 0.017255 0.033333  0.04714 0.057735 0.064395 0.066667
27
28 Initial conditions du(x,0)/dt at t=0 are :
29                 x=       x=       x=       x=       x=       x=       x=
30        t         0 0.083333 0.166667     0.25 0.333333 0.416667      0.5
31 ------------------------------------------------------------------------
32        0         0        0        0        0        0        0        0
33
34 RESULTS
LAX_WEN.WK3
```

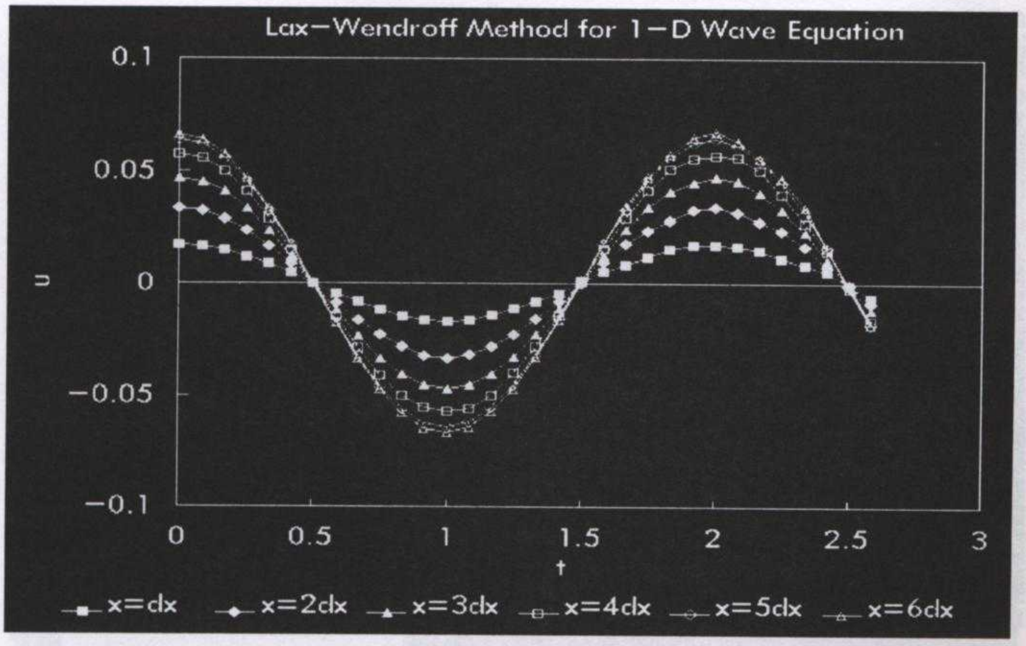
Fig. 5. The input data (dx and dt computed by program).

```
A:A53: U +A52+$DT                                                  READY
Number of t-stations to be computed (enter 0 to quit) = _

 A      A        B        C        D        E        F        G        H
34 RESULTS
35                 x=       x=       x=       x=       x=       x=       x=
36        t         0 0.083333 0.166667     0.25 0.333333 0.416667      0.5
37 ------------------------------------------------------------------------
38        0         0 0.017255 0.033333  0.04714 0.057735 0.064395 0.066667
39 0.083333         0  0.01653 0.032217 0.045562 0.055801 0.062238 0.064434
40 0.166667         0 0.014451 0.028832 0.040879 0.050066 0.055841 0.057811
41     0.25         0 0.011763 0.023222 0.033337  0.04092  0.04564  0.04725
42 0.333333         0 0.008532 0.016321 0.023264 0.028913  0.03233  0.03347
43 0.416667         0 0.004392 0.008591 0.011897 0.014675 0.016744 0.017411
44      0.5         0 -0.00025  0.00002  3.3E-06 -0.00027 -0.00024 0.000019
45 0.583333         0 -0.00457 -0.00878 -0.01215 -0.01491 -0.01699 -0.01789
46 0.666667         0  -0.0084 -0.01674  -0.0237 -0.02887 -0.03256 -0.03401
47     0.75         0 -0.01196 -0.02336 -0.03346 -0.04135 -0.04588 -0.04723
48 0.833333         0 -0.01483  -0.0287 -0.04101 -0.05047 -0.05602 -0.05775
49 0.916667         0  -0.0165  -0.0325 -0.04571 -0.05568 -0.06234  -0.0648
50        1         0 -0.01723 -0.03358 -0.04717 -0.05758 -0.06446 -0.06693
51 1.083333         0 -0.01663 -0.03197 -0.04545 -0.05595 -0.06217 -0.06412
52 1.166667         0 -0.01431 -0.02856 -0.04075 -0.05004  -0.0556  -0.0574
53     1.25         0 -0.01136 -0.02319 -0.03315 -0.04041 -0.04527 -0.04709
LAX_WEN.WK3                            CMD
```

Fig. 6. The numerical results.



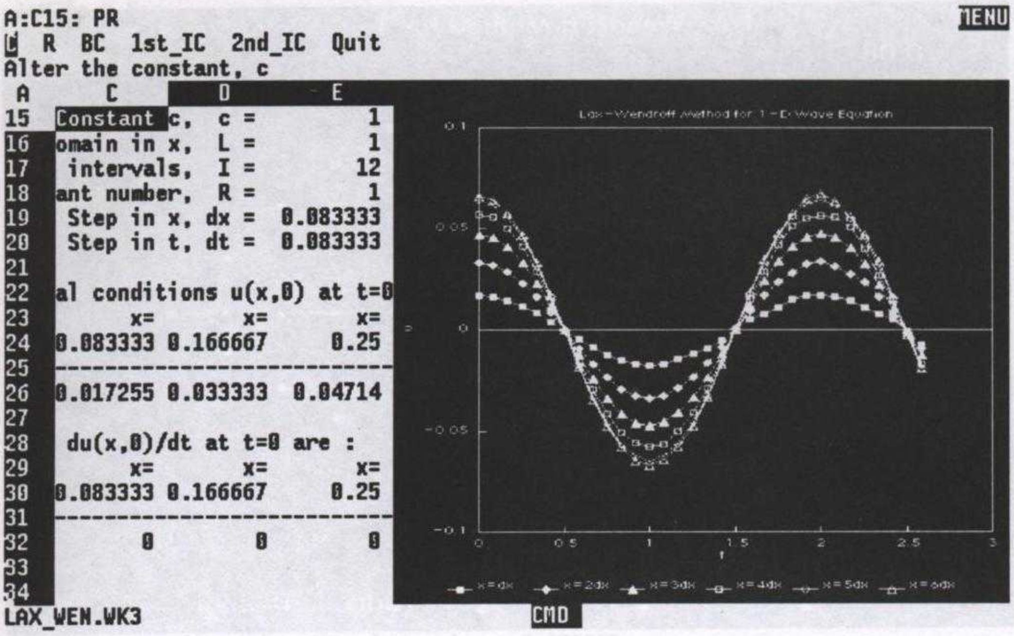
(a) Variations of u with x, $dt = 0.083333$ s



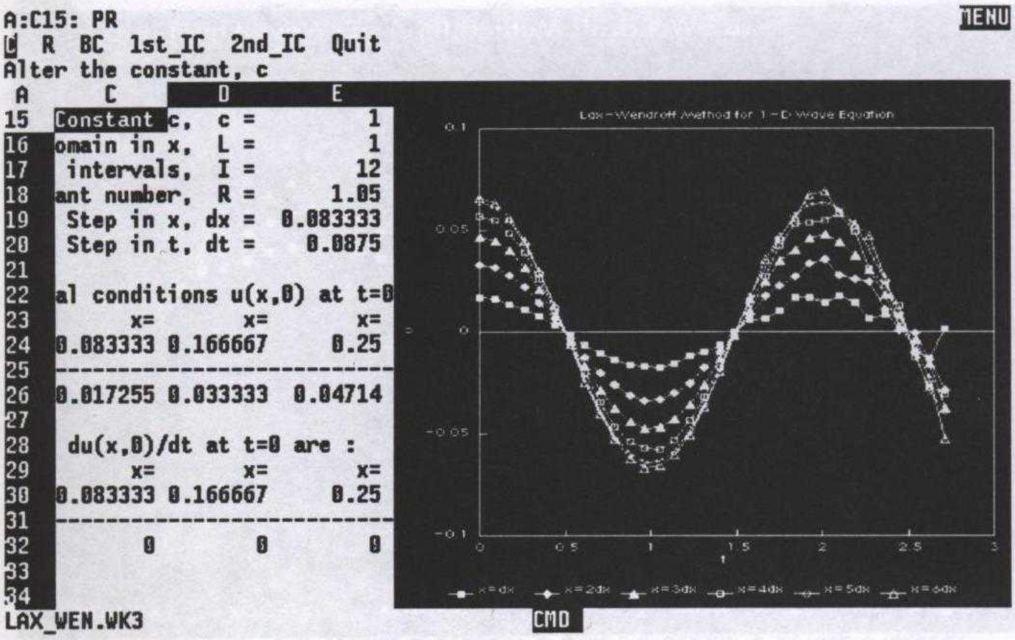
(b) Variations of u with t, $dx = 0.08333$ m

Fig. 7. The results graphed by the GRAPH commands.

show the screen displays in the interactive graphics mode after the results are graphed by the Graph T-variation and Graph X-variation commands respectively. In these figures, the Courant number R is changed to 1.05 and 1.1 and the results are automatically updated graphically on the screen. This is the graphical 'What-if?' analysis. It can be seen in Figures 8 and 9 that the solution is unstable for $R > 1$. As R increases, the instability swamps the solution at earlier time and it also becomes more severe. This information can be visualised by the user almost instantly as he or she proceeds in the interactive graphics mode. This feature thus enhances the learning effectiveness. Apart from R, the effects of changing other parameter such as the initial conditions can also be visualised by using the interactive graphics feature.



(a) $R = 1$

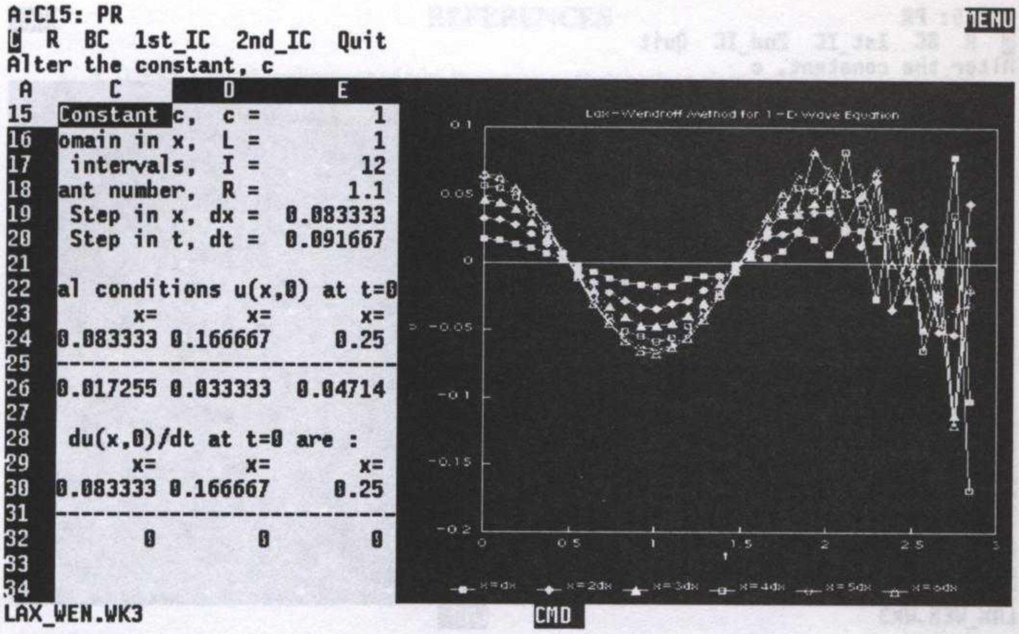


(b) $R = 1.05$

Fig. 8. The interactive graphics mode showing the t-variations for different R.

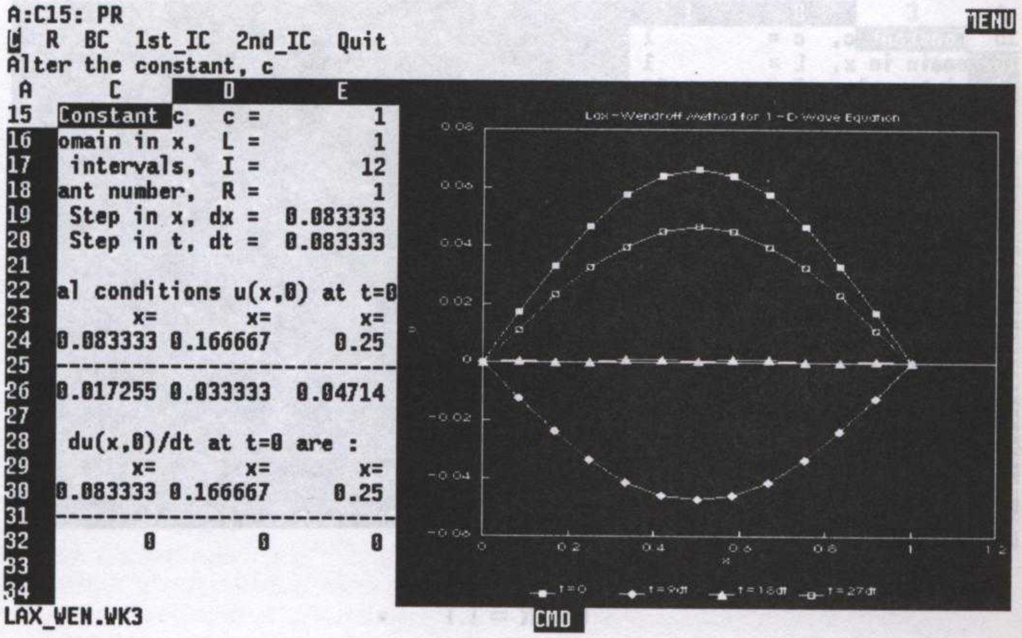
CONCLUDING REMARKS

A spreadsheet program has been developed to solve the one-dimensional wave equation by the Lax–Wendroff method. The program employs 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, extensive error and help messages have been incorporated. The program can therefore be used without much spreadsheet knowledge. The built-in power interactive graphics feature allows numerical experiments to be done graphically with ease. These features, which are not normally achievable by conventional programs, allow the students to concentrate on the numerical aspects of the Lax–Wendroff method and, therefore, help



(c) $R = 1.1$

Fig. 8. (*cont.*)

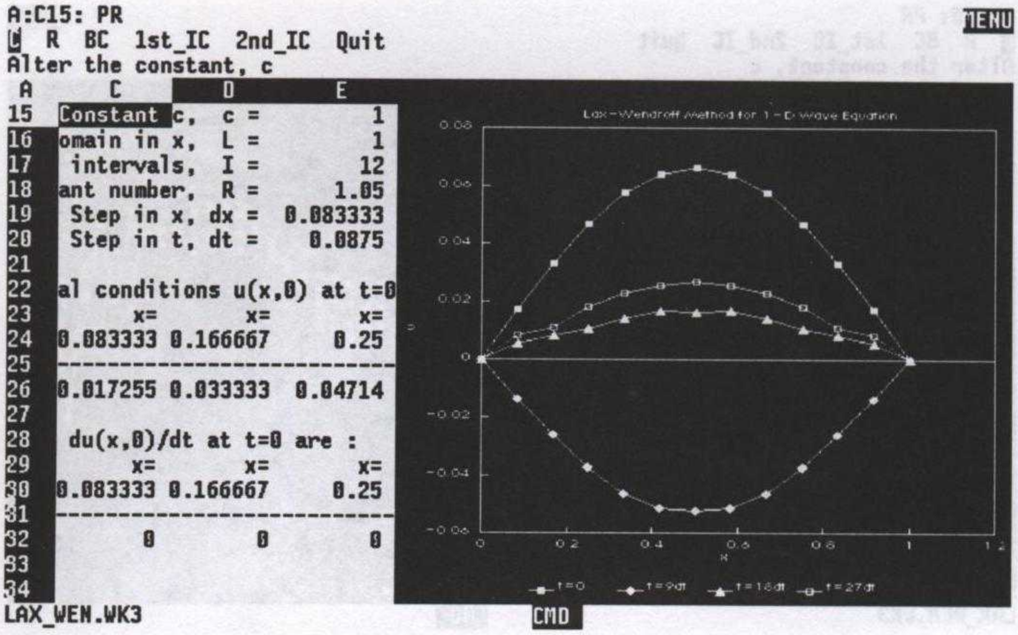


(a) $R = 1$

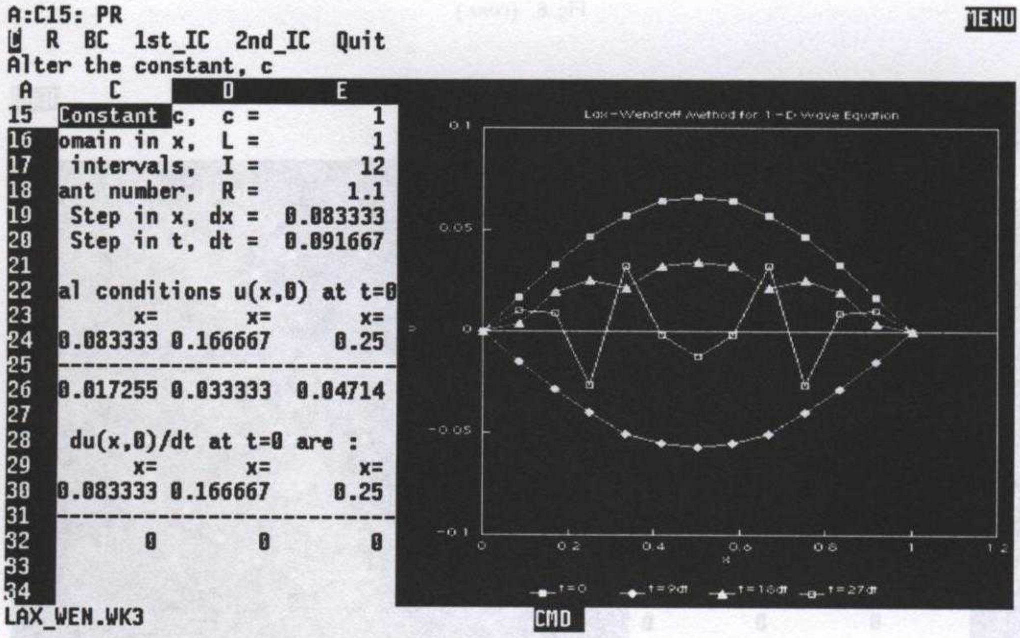
Fig. 9. The interactive graphics mode showing the x-variations for different R.

in stimulating their interest in exploring the method.

The limitations of the spreadsheet program are primarily governed by the capabilities of the Lotus 1-2-3 package. The number of curves that can be graphed together is limited to a maximum of six and the number of grid points 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 grid points that can be displayed on a screen is limited. For a fine grid system, in order to view the numerical values at some grid points outside a screen display, the user has to return to 1-2-3's READY mode by using the



(b) $R = 1.05$



(c) $R = 1.1$

Fig. 9 (*cont.*)

program command Quit and then use the arrow keys to move around. In order to incorporate the interactive graphics feature, the solution at each of the grid points is maintained as a spreadsheet formula which consists of a rather large number of characters. As a result, the memory requirement increases progressively when the number of t-steps increases. Although this limitation can easily be overcome by converting the formula at each grid point to a numerical value by incorporating the 1-2-3/Range Value command in the macro program, this was not done because it would disable the useful graphical 'What-if?' analysis while the limitation can be alleviated if more memory is available.

For educational applications, the above limitations are not significant as the requirements are not stringent. With the advancement in software and hardware technology, it is anticipated that these limitations will become less severe.

REFERENCES

1. N. D. Rao, Typical applications of microcomputer spreadsheets to electrical engineering problems, *IEEE Trans. Educ.*, E-**27**, 237–242 (1984).
2. S. K. Chan and C. Y. Lam, The electronic spreadsheet as a tool for course co-ordination in a school of engineering, *Comput. Educ.*, **14**, 231–238 (1990).
3. R. Kari, Spreadsheets in advanced physical chemistry, *J. Comput. Math. Sci. Teaching*, **10**, 39–48 (1990).
4. C. Y. Lam, Spreadsheet approach to partial differentiation equations. Part 1: elliptic equations, *Int. J. Engng Education*, **8**, 278–287 (1992).
5. C. Y. Lam, Spreadsheet approach to partial differential equations. Part 2: parabolic and hyperbolic equations, *Int. J. Engng Education*, **8**, 371–383 (1992).
6. C. Y. Lam, Method of characteristics using an electronic spreadsheet, *Int. J. Engng Education*, **10**, 208–214 (1994).
7. C. Y. Lam, *Applied Numerical Methods for Partial Differential Equations*. Prentice-Hall, Singapore (1994).
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. 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 current interests include computational fluid dynamics, mathematical modelling, microcomputer applications and anthropometry. He is a Chartered Engineer and a member of the IMechE and AIAA. He has written a textbook entitled *Applied Numerical Methods for Partial Differential Equations*.