Int. J. Engng Ed. Vol. 8, No. 5, pp. 371–383, 1992
Printed in Great Britain.
0742–0269/91 $3.00+0.00

Spreadsheet Approach to Partial Differential Equations. Part 2: Parabolic and Hyperbolic Equations*

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

As a companion paper to Part 1 [1], three other interactive spreadsheet programs developed by using the Lotus 1-2-3 spreadsheet software to solve the parabolic one-dimensional diffusion equation and the hyperbolic one-dimensional wave equation are presented. For the diffusion equation, the explicit forward time central space method and the implicit Crank–Nicholson method have been implemented in the first two programs. For the wave equation, the explicit central difference method has been implemented in the third program. The advanced interactive graphics capability is demonstrated in two of these programs employing the explicit methods. It is found that this feature is very useful in performing the 'What-if' analysis graphically. These programs have been tested in the classroom and favourable comments have been obtained.

INTRODUCTION

THE solution of the elliptic Laplace equation by using the spreadsheet approach has been presented in Part 1 [1]. Apart from the elliptic equation, the parabolic and hyperbolic equations are also the core discussion topics in engineering and science subjects related to partial differential equations. Generally, the parabolic one-dimensional diffusion equation and the hyperbolic one-dimensional wave equation are used to illustrate the numerical methods in the teaching of these subjects. The reasons are that these equations are simple and that they have a wide variety of physical applications; for example, the heat conduction (diffusion), the nuclear diffusion and the unsteady fluid flow problems for the diffusion equation and the vibration, the electricity transmission and the supersonic flow problems for the wave equation. The finite difference methods that are usually included in teaching are the explicit forward time central space (FTCS) method and the Crank–Nicolson (CN) method for the diffusion equation and the explicit central difference (CD) method for the wave equation. These methods are based on a marching technique in the time direction. Here, three programs developed to implement these finite difference methods by using the spreadsheet approach are described. The objective is again to provide the students with better alternative tools (other than the traditional computer programs), which can be used to enhance their understanding of the numerical methods concerned. The usefulness of the advanced interactive graphics feature provided by the Lotus 1-2-3 [2], which is impossible to achieve by the traditional approach, is explored and is found to be extremely helpful in viewing the effects of varying the input parameters in graphical form automatically for the explicit methods.

* Paper accepted 3 August 1992.

PROBLEM FORMULATION

The finite difference methods used in the spreadsheet programs can be found in many standard texts, for example [3–5].

The one-dimensional diffusion equation

The parabolic one-dimensional diffusion equation is

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2} \qquad (1)$$

where c is a constant and c^2 is the diffusivity, u is the dependent variable, t and x are the time and spatial coordinates respectively.

To solve the diffusion equation by finite difference methods, the physical t–x domain is discretized and the grid points are labelled as shown in Fig. 1. To obtain the numerical solution, two Dirichlet boundary conditions ($u(0, t)$ and $u(L, t)$, $t \geqslant$) and one initial condition ($u(x, 0)$, $0 < x < L$) are specified. The solution procedure is based on the marching technique in the t-direction.

The FTCS method. This method is also known as the explicit Euler method. By using the forward difference and the central difference representations for the time and spatial derivatives, respect-

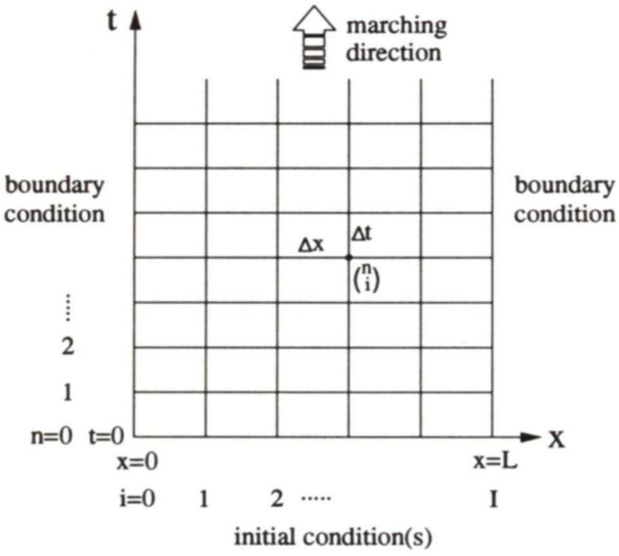


Fig. 1. The discretized computational domain

ively, the finite difference equation for equation (1) is

$$u_i^{n+1} = F(u_{i-1}^n + u_{i+1}^n) + (1-2F)u_i^n \quad (2)$$

where $F = c^2\Delta t/(\Delta x)^2$, known as the grid Fourier number, and the subscripts and the superscripts denote the numbers of the x- and t-grid points respectively. In this explicit method, $F \leqslant 1/2$ for stability.

The CN method. In this method, the spatial derivative is replaced by the mean of the central differences at the two consecutive t-grid points (n and $n+1$) and the time derivative is replaced by the central difference in the middle between these two grid points. The finite difference equation for equation (1) is

$$-Fu_{i-1}^{n+1} + (2+2F)u_i^{n+1} - Fu_{i+1}^{n+1} = Fu_{i-1}^n + (2-2F)u_i^n + Fu_{i+1}^n \quad (3)$$

where all parameters are the same as those defined for equation (2).

Applying this at the interior grid points $i = 1, \ldots, I-1$ for marching one t-step from n to $n+1$ yields

$$\begin{bmatrix} 2+2F & -F & 0 & 0 & 0 & \ldots & 0 & 0 \\ -F & 2+2F & -F & 0 & 0 & \ldots & 0 & 0 \\ 0 & -F & 2+2F & -F & 0 & \ldots & 0 & 0 \\ 0 & 0 & -F & 2+2F & -F & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \ldots & -F & 2+2F \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \cdot \\ \cdot \\ u_{I-1}^{n+1} \end{bmatrix} = \begin{bmatrix} Fu_0^n + (2-2F)u_1^n + Fu_2^n + Fu_0^{n+1} \\ Fu_1^n + (2-2F)u_2^n + Fu_3^n \\ Fu_2^n + (2-2F)u_3^n + Fu_4^n \\ Fu_3^n + (2-2F)u_4^n + Fu_5^n \\ \cdot \\ \cdot \\ Fu_{I-2}^n + (2-2F)u_{I-1}^n + Fu_I^n + Fu_I^{n+1} \end{bmatrix} \quad (4)$$

where I is the number of x-intervals.

This implicit method is unconditionally stable.

The one-dimensional wave equation

The hyperbolic one-dimensional wave equation is

$$\frac{\partial^2 u}{\partial^2 t} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (5)$$

where c (= constant) is the wave propagation speed, the other parameters are the same as those defined for equation (1).

In applying the finite difference method, the x–t domain is discretized and the grid points are labelled in the same manner as for the diffusion equation shown in Fig. 1. However, to obtain the numerical solution, two Dirichlet boundary conditions (u_0^n and u_I^n, $n \geqslant 0$) and two initial conditions (u_i^0 for $0 < i < I$ and $\partial u_i^0/\partial t$ for $0 \leqslant i \leqslant I$) are specified. Similar to that of the diffusion equation, the solution procedure is based on marching in the t-direction.

The CD method. Using the central differences for both derivatives, the finite difference equation for the wave equation is

$$u_i^{n+1} = -u_i^{n-1} + R^2(u_{i+1}^n + u_{i-1}^n) + 2(1-R^2)u_i^n \quad (6a)$$

where $R = c\Delta t/\Delta x$, known as the Courant number, and the subscripts and superscripts denote the numbers of the x- and t-grid points respectively.

Equation (6a) is used for $n > 0$. For the first time step, $n = 0$ and u_i^{-1} in equation (6a) is replaced by that obtained from the initial condition $\partial u_i^0/\partial t$ using the central difference and the equation becomes

$$u_i^1 = \frac{R^2}{2}(u_{i+1}^0 + u_{i-1}^0) + (1-R^2)u_i^0 + (\Delta t)\frac{\partial u_i^0}{\partial t} \quad (6b)$$

where $\partial u_i^0/\partial t = \partial u(x_i, 0)/\partial t$ is prescribed.

This method is explicit and $R \leqslant 1$ for stability.

THE SPREADSHEET PROGRAMS

The programs developed are menu driven with a common menu almost identical to that for the programs for solving Laplace equation as presented in Part 1 (see fig. 3 and table 1 in [1]). The only differences are that the Graph T-variation command replaces the Graph Y-variation command because t is now one of the independent variables and that the Data Use command will continue the calculations at more t-steps (rather than the iteration) using the existing data since the finite difference methods used here are based on the time marching procedure. The same user-friendly and interactive features as those described in Part 1 [1] are retained. The only difference is in the procedure of using these programs due to the time marching procedure. In this case, after the marching of a t-step is completed, the user will be requested to indicate whether the marching is to be continued for the next t-step. If the response is negative, the program will return to the main menu.

In the programs, the step sizes Δx and Δt are computed automatically based on the input data L, I, F and R.

The interactive graphics feature. The Lotus 1-2-3 Release 3.1 offers a useful interactive graphics capability which is ideal in studying the effects on the solution graphically when an input parameter is changed—the so-called 'What-if' analysis. This is done by using the two simple 1-2-3 commands /Worksheet Window Graph and /Worksheet Window Clear. First, the program command Quit must be used to return to 1-2-3's READY mode after a solution has been obtained and a graph has been created by the program command Graph X-variation or Graph T-variation. To set the interactive graphics mode, the arrow keys are used to move the 1-2-3 cursor to a column which can be any one apart from the leftmost column shown on the screen. Upon invoking the 1-2-3's /Worksheet Window Graph command, the monitor screen will be split vertically at that column and the current graph will be displayed on the right side of the screen. Some of the input data can then be altered manually (by moving the 1-2-3 cursor to the respective location(s) followed by keying in the new value(s) and hitting the Enter key) and the numerical results as well as the current graph will be updated automatically. The interactive graphics mode can be cleared by using the 1-2-3's /Worksheet Window Clear command.

The interactive graphics mode as explained can be used only when the spreadsheet cells storing the solution contain formulae which depend algebraically on the contents of the other cells storing the input data so that the contents of the former cells as well as the graph will be updated accordingly when the contents of the latter cells are changed. For the programs employing the FTCS method for the diffusion equation and the CD method for the wave equations, any input data except the number of x-intervals I can be altered manually after a solution has been obtained. The value of I cannot be altered in this manner because new x-grid points are required for a different value of I and the program can only generate new x-grid points within the command Data Use. For the CN method implemented for the diffusion equation, the interactive graphics mode cannot be used since each t-step marching is done by solving equation (4) using matrix operations in a way that the contents of the cells storing the solution are numerical values.

All programs assume constant Dirichlet boundary conditions. The boundary conditions are required to be entered only once at the starting step corresponding to $t = 0$ and they will be used for the time marching in subsequent t-steps. However, for the FTCS method and the CD method, variable Dirichlet boundary conditions can also be used as will be discussed later. The initial condition is entered at each interior grid point corresponding to $t = 0$ accordingly.

The spreadsheet program FTCS.WK3

This spreadsheet program employs the FTCS method to solve the diffusion equation. To march one t-step, the program assigns equation (2) at the new grid points and the numerical results are obtained. This is done by the 1-2-3's /Copy command programmed in the macros.

Example. Consider the unsteady one-dimensional heat conduction problem of a thin metal rod perfectly insulated laterally of length $L = 1$ m, $c = 1.1$ m/s$^{0.5}$ and initial temperature

```
A:H20: 10                                                              READY
Any changes for the boundary or initial conditions?  <Y/N> : _

A      A         B         C         D         E         F         G         H
1  =======================================================================
2  SOLVING THE 1-D DIFFUSION EQUATION BY THE FTCS METHOD
3  Conditions : constant steps in x and t, constant B.C.'s (may not equal)
4  [© C Y Lam, Nanyang Technological University]
5  =======================================================================
6  Equation to be solved :
7  du(x,t)/dt = c² × d²u(x,t)/dx² where d denotes partial differentiation
8
9                     Constant c,  c =        1.1
10          Solution domain in x,  L =          1
11         Number of x intervals,  I =          6
12           Grid Fourier number,  F =        0.3
13                 Step in x, dx =  0.166667
14                 Step in t, dt =  0.006887
15
16 RESULTS
17                 x=        x=        x=        x=        x=        x=        x=
18        t         0  0.166667  0.333333       0.5  0.666667  0.833333         1
19 -----------------------------------------------------------------------
20        0        30         0         0         0         0         0        10
FTCS.WK3                                 CMD
```

Fig. 2. The input data in FTCS.WK3

$u(x, 0) = 0$°C. At $t = 0$ s, the temperature at one end $u(0, t)$ is suddenly brought to 30°C and the temperature at the other end $u(1, t)$ is brought to 10°C.

Using the program command Data Input, the data are entered as shown in Fig. 2 in which $I = 6$ and $F = 0.3$. In this figure, the x-grid points have been overlaid on the working sheet A automatically before the boundary and initial conditions are entered along the first row corresponding to $t = 0$. The results for 15 t-steps are obtained as shown in Fig. 3(a) and the variations of u with x are graphed as shown in Fig. 3(b).

Figure 4 shows the screen in interactive graphics mode invoked by the 1-2-3's command /Worksheet Window Graph with the cursor moved to column D. It can be seen that the screen is divided vertically with the current graph (variations with x in this case) displayed on the right. To study the effects of varying the grid Fourier number F, the content of the cell with address A:E12 which contains F is changed manually and the numerical results as well as the graph are updated automatically as shown in Fig. 5(a–d) for $F = 0.5$, 0.51, 0.55 and 0.6 respectively. These figures clearly verify that the FTCS method is unstable for $F > 0.5$.

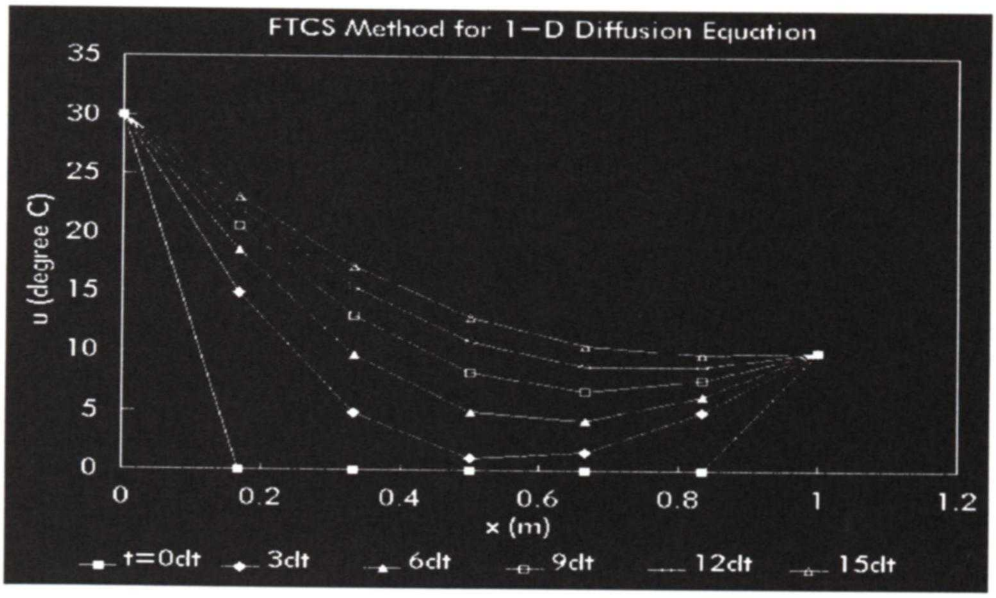
The 'What-if' analysis is also done by varying the other parameters. Figure 6 shows the results by changing the boundary condition $u(0, t)$ to 10°C and Fig. 7 shows the results by changing the initial condition $u(x, 0)$ to $-20 \sin(\pi x)$. In Fig. 7, the new initial condition is entered as 1-2-3 formulaes

A:A35: +A34+$DT
Proceed another t-step ? <Y/N> : _ READY

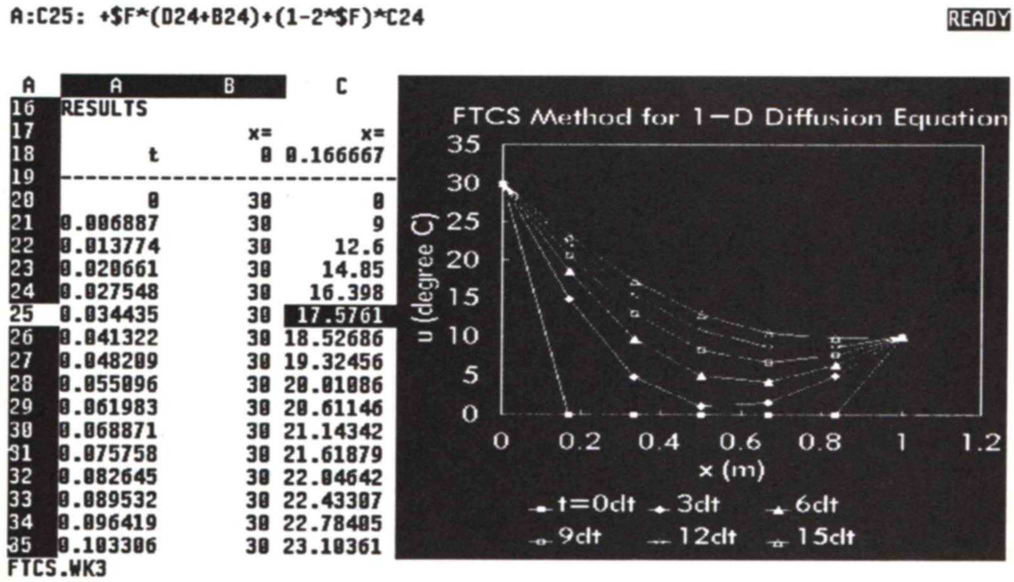
A	A	B	C	D	E	F	G	H
16	RESULTS							
17		x=	x=	x=	x=	x=	x=	x=
18	t	0	0.166667	0.333333	0.5	0.666667	0.833333	1
19	----------							
20	0	30	0	0	0	0	0	10
21	0.006887	30	9	0	0	0	3	10
22	0.013774	30	12.6	2.7	0	0.9	4.2	10
23	0.020661	30	14.85	4.86	1.08	1.62	4.95	10
24	0.027548	30	16.398	6.723	2.376	2.457	5.466	10
25	0.034435	30	17.5761	8.3214	3.7044	3.3354	5.9235	10
26	0.041322	30	18.52686	9.71271	4.9788	4.22253	6.37002	10
27	0.048209	30	19.32456	10.93678	6.172092	5.093658	6.814767	10
28	0.055096	30	20.01086	12.02371	7.277969	5.933521	7.254004	10
29	0.061983	30	20.61146	12.99613	8.298356	6.733	7.681658	10
30	0.068871	30	21.14342	13.8714	9.238082	7.487204	8.092563	10
31	0.075758	30	21.61879	14.66301	10.10281	8.194075	8.483187	10
32	0.082645	30	22.04642	15.38168	10.89825	8.85343	8.851497	10
33	0.089532	30	22.43307	16.03607	11.62983	9.466296	9.196628	10
34	0.096419	30	22.78405	16.6333	12.30264	10.03446	9.51854	10
35	0.103306	30	23.10361	17.17933	12.92139	10.56014	9.817753	10

FTCS.WK3 CMD

Fig. 3. The solution of the one-dimensional heat conduction problem for $I = 6$ and $F = 0.3$ using FTCS.WK3
(a) The solution for 15 t-steps



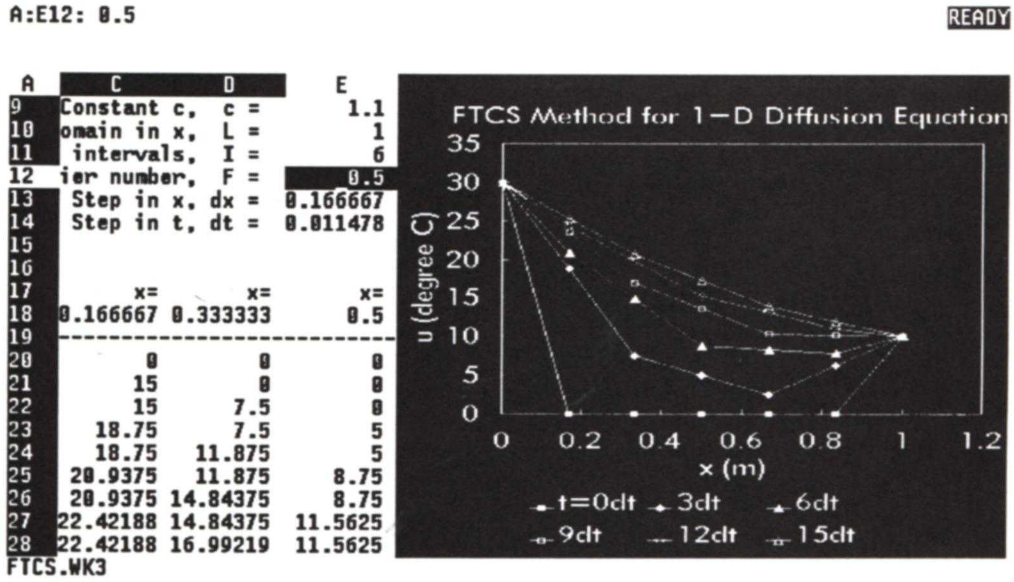
(b) Variations of u with x as displayed on the screen


A:C25: +$F*(D24+B24)+(1-2*$F)*C24 READY

	A	B	C
16	RESULTS		
17		x=	x=
18	t	0	0.166667
19	------------	------------	------------
20	0	30	0
21	0.006887	30	9
22	0.013774	30	12.6
23	0.020661	30	14.85
24	0.027548	30	16.398
25	0.034435	30	17.5761
26	0.041322	30	18.52686
27	0.048209	30	19.32456
28	0.055096	30	20.01086
29	0.061983	30	20.61146
30	0.068871	30	21.14342
31	0.075758	30	21.61879
32	0.082645	30	22.04642
33	0.089532	30	22.43307
34	0.096419	30	22.78405
35	0.103306	30	23.10361

FTCS.WK3

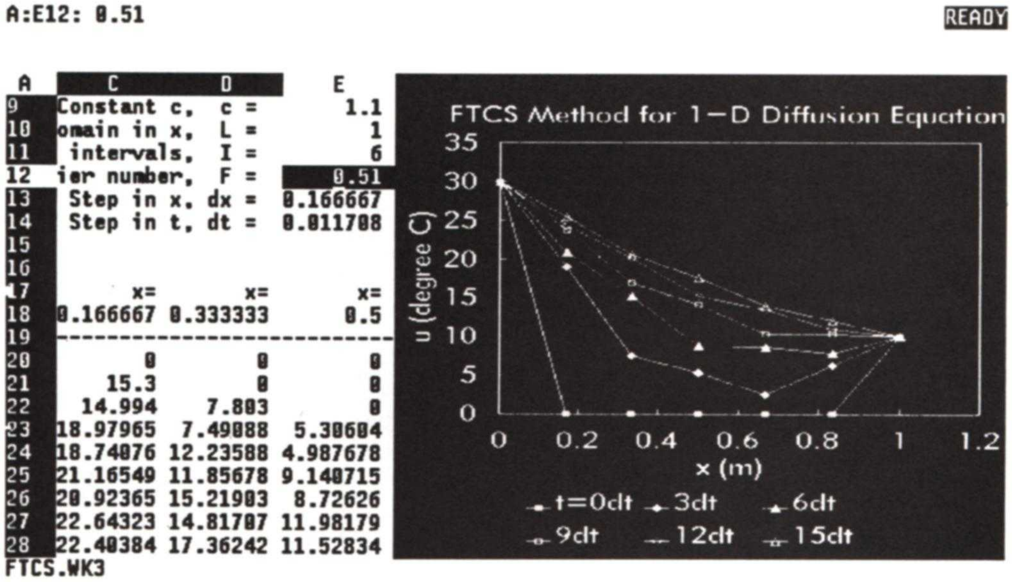
Fig. 4. The interactive graphics screen for $I = 6$ and $F = 0.3$ using FTCS.WK3


A:E12: 0.5 READY

	C	D	E
9	Constant c,	c =	1.1
10	omain in x,	L =	1
11	intervals,	I =	6
12	ier number,	F =	0.5
13	Step in x,	dx =	0.166667
14	Step in t,	dt =	0.011478
15			
16			
17	x=	x=	x=
18	0.166667	0.333333	0.5
19	------------	------------	------------
20	0	0	0
21	15	0	0
22	15	7.5	0
23	18.75	7.5	5
24	18.75	11.875	5
25	20.9375	11.875	8.75
26	20.9375	14.84375	8.75
27	22.42188	14.84375	11.5625
28	22.42188	16.99219	11.5625

FTCS.WK3

Fig. 5. The effects of varying F using FTCS.WK3 ($I = 6$)
(a) $F = 0.5$


A:E12: 0.51 READY

	C	D	E
9	Constant c,	c =	1.1
10	omain in x,	L =	1
11	intervals,	I =	6
12	ier number,	F =	0.51
13	Step in x,	dx =	0.166667
14	Step in t,	dt =	0.011708
15			
16			
17	x=	x=	x=
18	0.166667	0.333333	0.5
19	------------	------------	------------
20	0	0	0
21	15.3	0	0
22	14.994	7.803	0
23	18.97965	7.49088	5.30604
24	18.74076	12.23588	4.987678
25	21.16549	11.85678	9.140715
26	20.92365	15.21903	8.72626
27	22.64323	14.81707	11.98179
28	22.40384	17.36242	11.52834

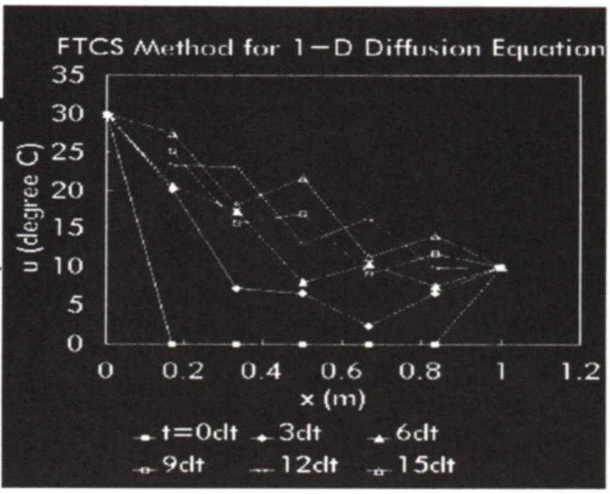
FTCS.WK3

(b) $F = 0.51$.

A:E12: 0.55 READY

A	C	D	E
9	Constant c,	c =	1.1
10	omain in x,	L =	1
11	intervals,	I =	6
12	ier number,	F =	0.55
13	Step in x,	dx =	0.166667
14	Step in t,	dt =	0.012626
15			
16			
17	x=	x=	x=
18	0.166667	0.333333	0.5
19	----------	----------	----------
20	0	0	0
21	16.5	0	0
22	14.85	9.075	0
23	20.00625	7.26	6.655
24	18.49238	13.93769	4.6585
25	22.31649	11.33921	11.09721
26	20.50492	17.24362	8.010956
27	23.9335	13.95937	14.40762
28	21.7843	19.69168	10.3583

FTCS.WK3

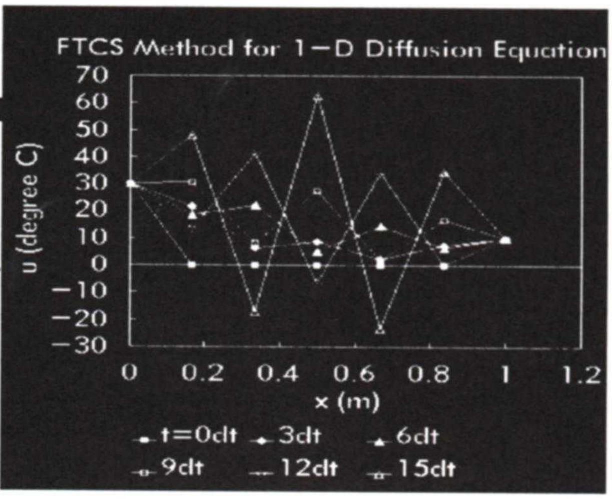


(c) $F = 0.55$

A:E12: 0.6 READY

A	C	D	E
9	Constant c,	c =	1.1
10	omain in x,	L =	1
11	intervals,	I =	6
12	ier number,	F =	0.6
13	Step in x,	dx =	0.166667
14	Step in t,	dt =	0.013774
15			
16			
17	x=	x=	x=
18	0.166667	0.333333	0.5
19	----------	----------	----------
20	0	0	0
21	18	0	0
22	14.4	10.8	0
23	21.6	6.48	8.64
24	17.568	16.848	3.456
25	24.5952	9.2448	14.8608
26	18.62784	21.82464	4.8384
27	27.36922	9.714816	20.72218
28	18.35505	26.91187	3.942605

FTCS.WK3



(d) $F = 0.6$

A:B20: 10 READY

A	A	B	C
7	du(x,t)/dt =	c² ×	d²u(x,t)/
8			
9			Constant
10		Solution	domain in
11		Number of	x interval
12		Grid	Fourier numbe
13			Step in
14			Step in
15			
16	RESULTS		
17		x=	x=
18	t	0	0.166667
19	----------	----------	----------
20	0	10	0
21	0.006887	10	3
22	0.013774	10	4.2
23	0.020661	10	4.95
24	0.027548	10	5.466
25	0.034435	10	5.8749
26	0.041322	10	6.22422

FTCS.WK3

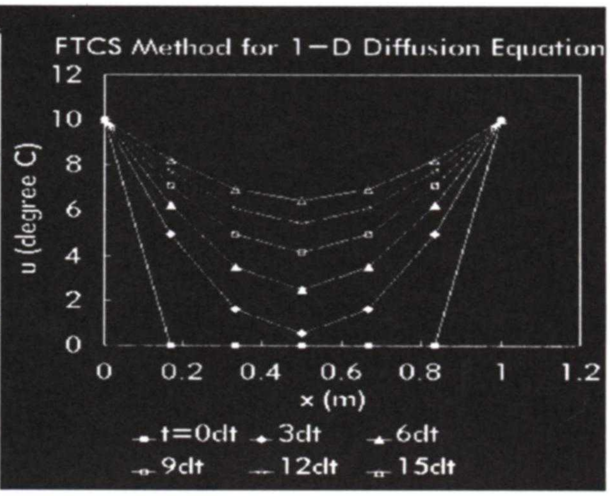


Fig. 6. The effects of changing the boundary conditions to $u(0, t) = 10$ using FTCS.WK3 ($I = 6$, $F = 0.3$)

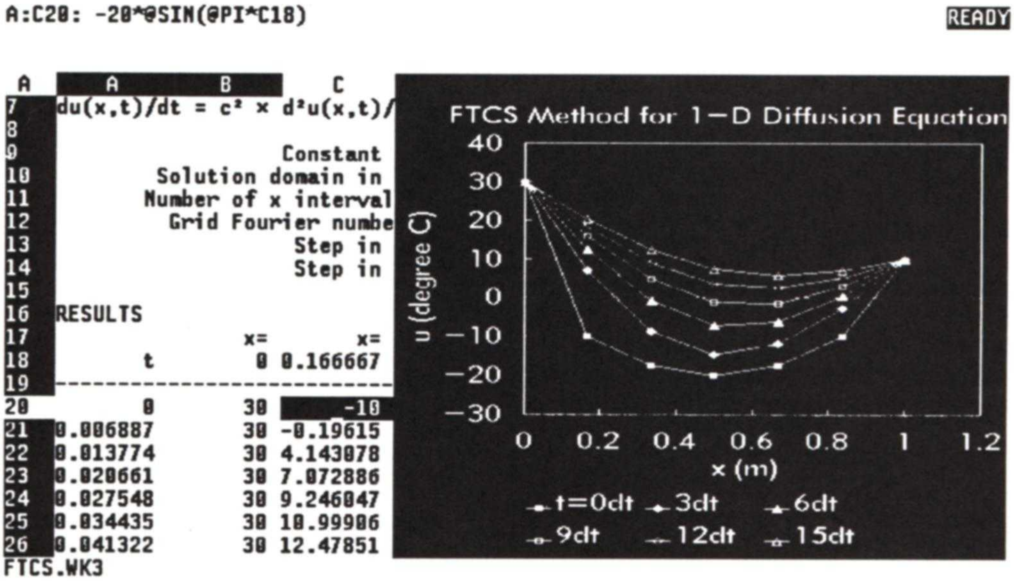


Fig. 7. The effects of changing the initial condition to $u(x, 0) = -20\sin(\pi x)$ using FTCS.WK3 ($I = 6$, $F = 0.3$)

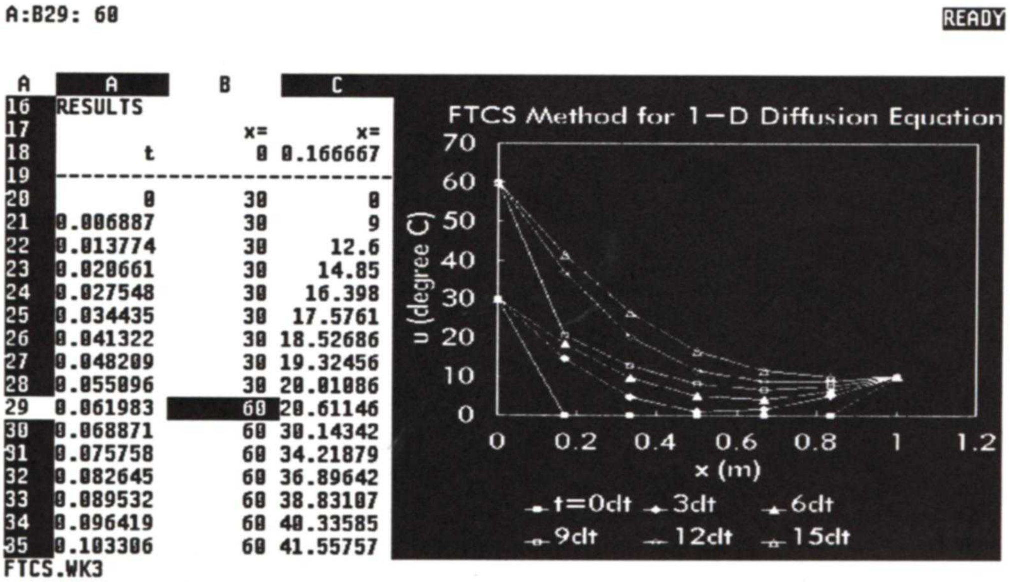


Fig. 8. The effects of changing to a variable boundary condition $u(0, t) = 30$ for $t < 9\Delta t$, $u(0, t) = 60$ for $t \geqslant 9\Delta t$ using FTCS.WK3 ($I = 6$, $F = 0.3$)

at the corresponding x-grid points using the built-in mathematical functions of 1-2-3. The alternative way is to directly enter the pre-calculated numerical values of the initial conditions at the x-grid points. By using this technique, problems with variable boundary conditions can be easily treated. Figure 8 displays the results when the original boundary condition $u(0, t)$ is suddenly increased from 30 to 60°C at $t = 9\Delta t$.

The spreadsheet program CN.WK3

This spreadsheet program employs the CN method to solve the diffusion equation. To march one t-step, equation (4) is solved by the matrix inversion technique. This is done by the 1-2-3's /Data Matrix Invert and /Data Matrix Multiply commands programmed in the macros.

Example. Consider the same problem solved previously.

The input data are the same as those shown in Fig. 2 for FTCS.WK3. In this case, the computed results for 15 t-steps and the variations of u with x are obtained as shown in Fig. 9. For CN.WK3, the interactive graphics mode cannot be used as explained previously. However, the results for $F = 0.6$ are obtained as shown in Fig. 10 which does not exhibit any numerical instability.

The spreadsheet program CD.WK3

This spreadsheet program employs the explicit CD method to solve the wave equation. To march one t-step, the program assigns equation (6a) or (6b) accordingly to the interior grid points at the end of the t-step and the numerical results are obtained. This is done by the 1-2-3's /Copy command programmed in the macros.

Example. Consider the problem of finding the deflection $u(x, t)$ 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 $u(x, 0) = \sin(\pi x)/15$.

The input data c, L, I, R, the boundary conditions and the initial conditions are entered as shown in Fig. 11. The computed results for 15 t-steps are obtained as shown in Fig. 12(a), the variation of u with x and the variation of u with t are shown in Fig. 12(b–c). From Fig. 12(c), it is easily seen that the period of the vibration is 2 s.

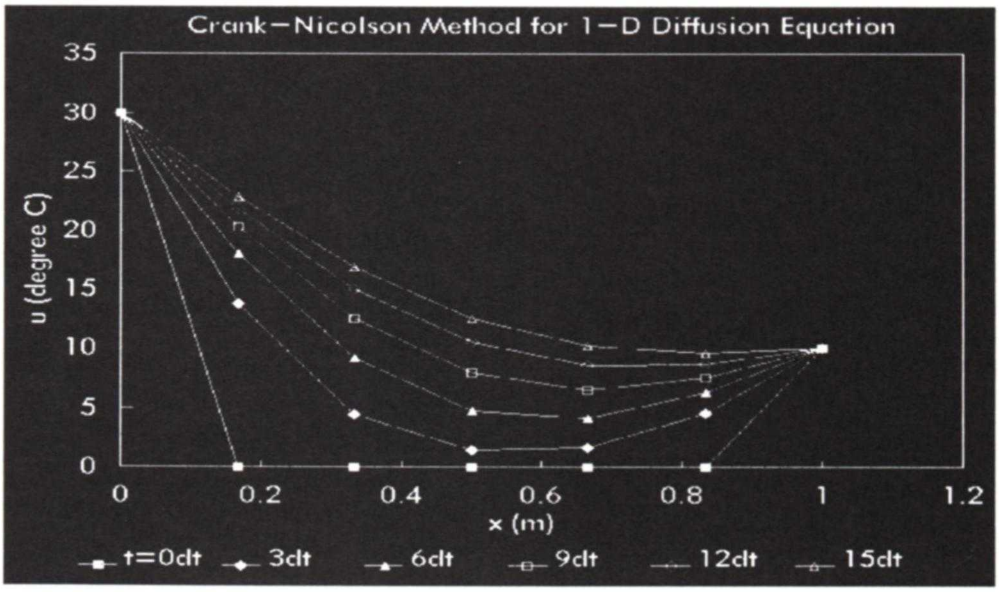
The interactive graphics mode is used to perform the 'What-if' analysis by following the same procedure as that described for FTCS.WK3. The effects of varying the Courant number R are shown in Figs 13 and 14. These figures clearly demonstrate that the CD method is unstable for $R > 1$. Figure 15 displays the results by changing the initial condition

A:A35: +A34+DT READY
Proceed another t-step ? <Y/N> : _

A	A	B	C	D	E	F	G	H
16	Results							
17		x=	x=	x=	x=	x=	x=	x=
18	t	0	0.166667	0.333333	0.5	0.666667	0.833333	1
19	-----	-----	-----	-----	-----	-----	-----	-----
20	0	30	0	0	0	0	0	10
21	0.006887	30	7.018219	0.824565	0.128008	0.284834	2.340558	10
22	0.013774	30	11.09949	2.619332	0.60719	0.936215	3.708883	10
23	0.020661	30	13.72261	4.512353	1.454215	1.701745	4.609163	10
24	0.027548	30	15.55581	6.265784	2.513184	2.514704	5.276063	10
25	0.034435	30	16.92614	7.833681	3.656446	3.346841	5.824981	10
26	0.041322	30	18.00547	9.225091	4.805544	4.179018	6.312589	10
27	0.048209	30	18.8899	10.46183	5.917891	4.99657	6.7655	10
28	0.055096	30	19.63623	11.56599	6.97268	5.788673	7.195105	10
29	0.061983	30	20.27982	12.5567	7.961357	6.54789	7.605429	10
30	0.068871	30	20.84373	13.44978	8.881953	7.269554	7.997244	10
31	0.075758	30	21.3437	14.25818	9.735896	7.951132	8.370134	10
32	0.082645	30	21.79091	14.9925	10.52627	8.59166	8.723471	10
33	0.089532	30	22.19364	15.66146	11.25686	9.191289	9.056825	10
34	0.096419	30	22.55817	16.27237	11.9317	9.750942	9.370086	10
35	0.103306	30	22.88945	16.83136	12.55477	10.27205	9.663468	10

CN.WK3 CMD

Fig. 9. The solution of the one-dimensional heat conduction problem for $I = 6$ and $F = 0.3$ using CN.WK3
(a) The solution for 15 t-steps



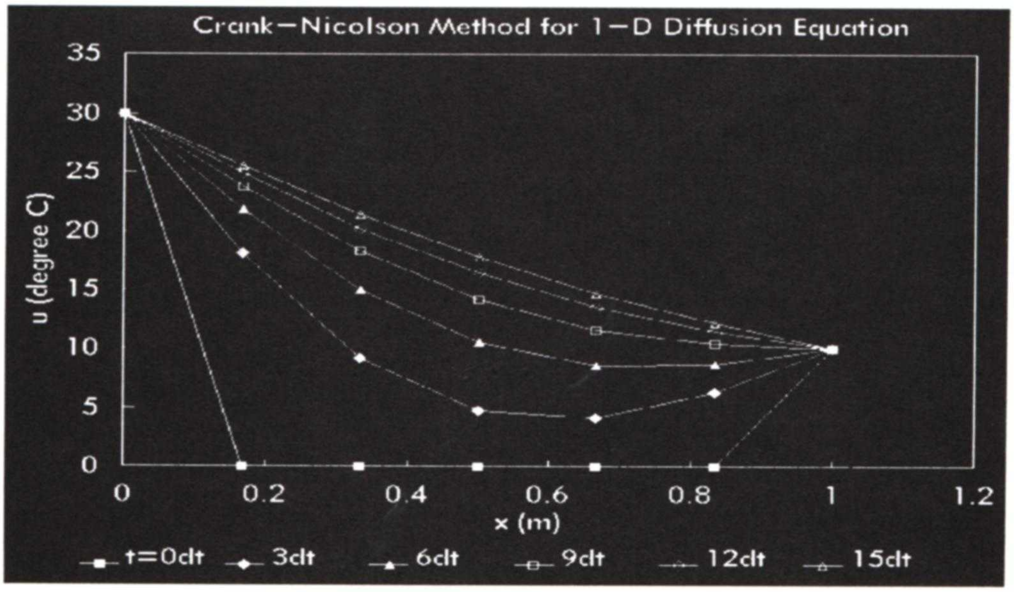
(b) Variations of u with x as displayed on the screen

A:A35: +A34+DT MENU
Data Graph Print Save Quit
Input Use Quit

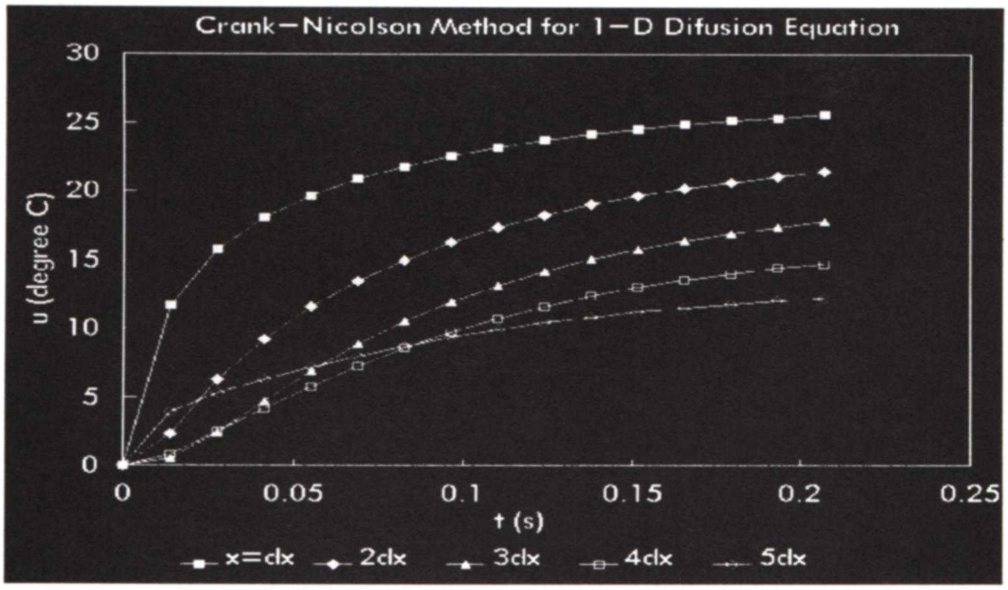
	A	B	C	D	E	F	G	H
16	Results							
17		x=	x=	x=	x=	x=	x=	x=
18	t	0	0.166667	0.333333	0.5	0.666667	0.833333	1
19	-----	-----	-----	-----	-----	-----	-----	-----
20	0	30	0	0	0	0	0	10
21	0.013774	30	11.6814	2.300797	0.58952	0.843307	3.90812	10
22	0.027548	30	15.77958	6.281755	2.384465	2.505263	5.354887	10
23	0.041322	30	18.10979	9.264369	4.760458	4.158436	6.338165	10
24	0.055096	30	19.68991	11.60212	6.965303	5.776082	7.197264	10
25	0.068871	30	20.87523	13.47924	8.889326	7.265231	7.994562	10
26	0.082645	30	21.81174	15.01638	10.53876	8.593457	8.722144	10
27	0.096419	30	22.57328	16.29212	11.94569	9.7567	9.371191	10
28	0.110193	30	23.20317	17.36043	13.14399	10.76443	9.94051	10
29	0.123967	30	23.72968	18.26029	14.16415	11.63168	10.4344	10
30	0.137741	30	24.17269	19.02115	15.03256	12.37497	10.85985	10
31	0.151515	30	24.54702	19.66604	15.77175	13.01041	11.22472	10
32	0.165289	30	24.86417	20.21349	16.40094	13.55276	11.53677	10
33	0.179063	30	25.13332	20.67867	16.93651	14.01521	11.80319	10
34	0.192837	30	25.362	21.07421	17.39238	14.40926	12.03038	10
35	0.206612	30	25.55641	21.41065	17.78041	14.74491	12.224	10

CN.WK3 CMD

Fig. 10. The solution of the one-dimensional heat conduction problem for $I = 6$ and $F = 0.6$ using CN.WK3.
(a) The solution for 15 t-steps



(b) Variations of u with x as displayed on the screen



(c) Variations of u with t as displayed on the screen

```
A:H20: 0                                                              READY
Any changes for u(x,0) ?  <Y/N> :  _

A      A        B        C        D        E        F        G         H
1  ======================================================================
2  SOLVING THE 1-D WAVE EQUATION BY THE EXPLICIT CENTRAL DIFFERENCE METHOD
3  Conditions : constant steps in x and t, constant B.C.'s (may not equal)
4  [© C Y Lam, Nanyang Technological University]
5  ======================================================================
6  Equation to be solved :
7  d²u(x,t)/dt² = c² x d²u(x,t)/dx² where d denotes partial differentiation
8
9                   Constant c,  c =          1
10          Solution domain in x,  L =          1
11         Number of x intervals,  I =          6
12               Courant number,  R =          1
13                  Step in x, dx =  0.166667
14                  Step in t, dt =  0.166667
15
16 Boundary and initial conditions u(x,0) at t=0 are :
17                    x=       x=       x=       x=       x=       x=       x=
18         t           0 0.166667 0.333333      0.5 0.666667 0.833333        1
19 ----------------------------------------------------------------------
20         0           0 0.033333 0.057735 0.066667 0.057735 0.033333        0
CD.WK3                                 CMD
```

Fig. 11. The input data in CD.WK3

(a) The input data and the boundary and initial conditions $u(x, 0)$

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

A      A        B        C        D        E        F        G         H
22 Initial conditions du(x,0)/dt 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        0        0        0        0        0
27
```

(b) The initial condition $\partial u(x, 0)/\partial t$

$u(x, 0)$ to $u(x, 0) = 0.015x$ for $0 \leqslant x \leqslant 4\Delta x$ and $u(x, 0) = 0.03(1 - x)$ for $4\Delta x \leqslant x \leqslant 1$. The other initial and boundary conditions can also be altered to study the effects of the changing.

CONCLUSIONS

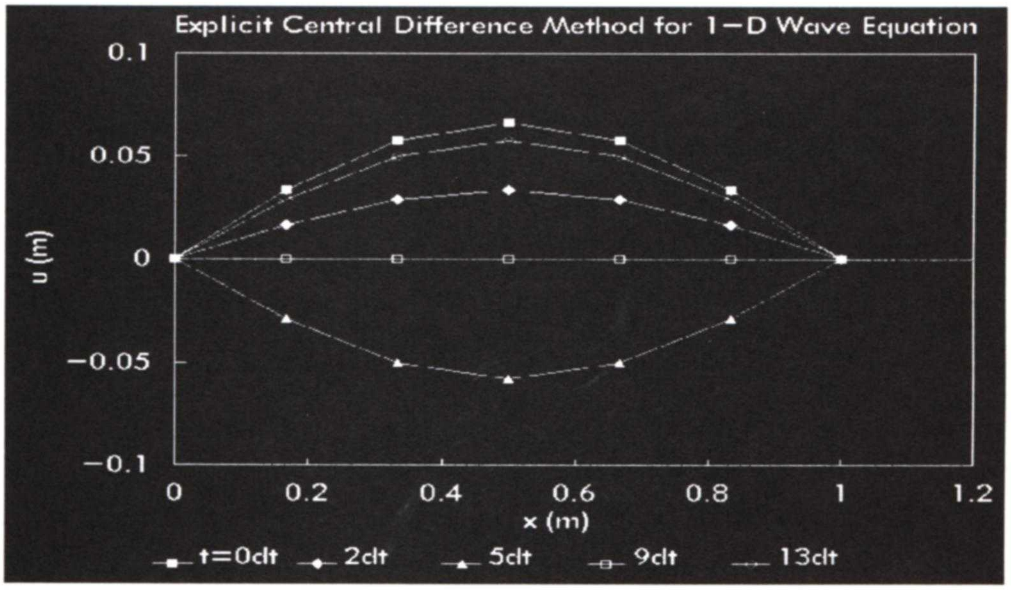
Three spreadsheet programs have been developed to solve the one-dimensional diffusion equation by the explicit FTCS method and the implicit CN method, and the one-dimensional wave equation by the explicit CD method. These programs are menu-driven, user-friendly and interactive. Little spreadsheet knowledge is required to use these programs. The programs use constant Dirichlet boundary conditions. However, variable boundary conditions can also be used if they are entered manually. The powerful interactive graphics feature allows the user to carry out a series of numerical experiments easily by varying different input parameters and to view the effects automatically in graphical form. These features, which are difficult or impossible to obtain by the traditional programming approach, certainly help the students to improve their understanding of the numerical methods. The limitations of these spreadsheet programs are similar to those of the programs developed for Laplace equation as discussed in Part 1 [1]. They include a maximum of six curves that can be graphed together and the number of x-grid points cannot be too large to avoid memory full error. However, the limitations do not present any problems for educational purposes as the grid systems need not be very fine. Classroom use of these programs has resulted favourable comments by the students, especially in the ease of use and the interactive graphics feature. The spreadsheet approach can be extended to other finite difference methods, other problems subject to Neumann boundary conditions and other parabolic and hyperbolic equations. Work in this direction is being pursued.

A:A47: +A46+$DT READY
Proceed another t-step ? <Y/N> : _

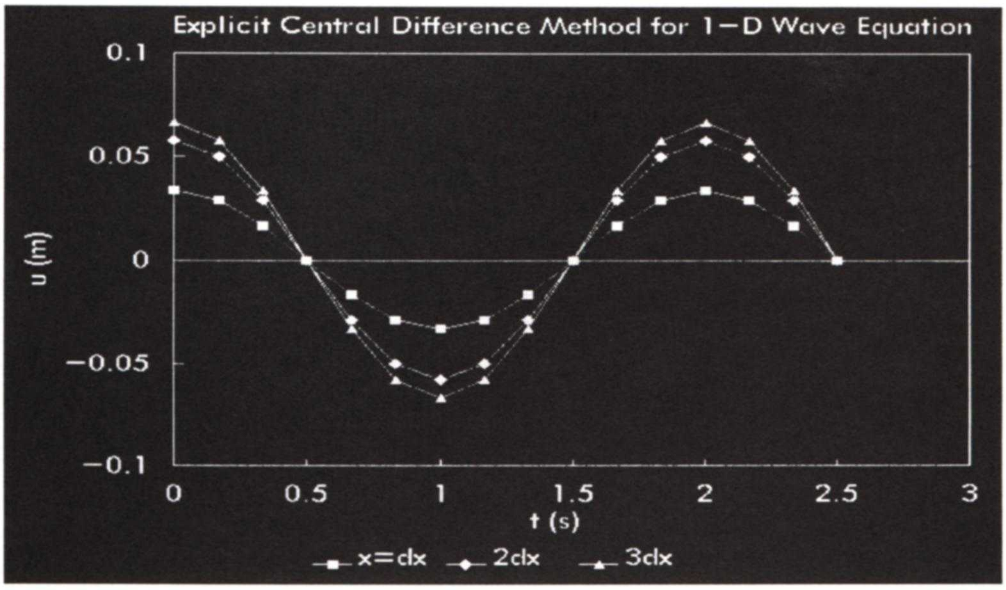
	A	B	C	D	E	F	G	H
28	RESULTS							
29		x=	x=	x=	x=	x=	x=	x=
30	t	0	0.166667	0.333333	0.5	0.666667	0.833333	1
31	-------	-------	-------	-------	-------	-------	-------	-------
32	0	0	0.033333	0.057735	0.066667	0.057735	0.033333	0
33	0.166667	0	0.028868	0.05	0.057735	0.05	0.028868	0
34	0.333333	0	0.016667	0.028868	0.033333	0.028868	0.016667	0
35	0.5	0	-3.4E-21	-6.8E-21	-3.4E-21	1.0E-20	0	0
36	0.666667	0	-0.01667	-0.02887	-0.03333	-0.02887	-0.01667	0
37	0.833333	0	-0.02887	-0.05	-0.05774	-0.05	-0.02887	0
38	1	0	-0.03333	-0.05774	-0.06667	-0.05774	-0.03333	0
39	1.166667	0	-0.02887	-0.05	-0.05774	-0.05	-0.02887	0
40	1.333333	0	-0.01667	-0.02887	-0.03333	-0.02887	-0.01667	0
41	1.5	0	-3.4E-21	3.4E-21	-3.4E-21	2.0E-20	3.4E-21	0
42	1.666667	0	0.016667	0.028868	0.033333	0.028868	0.016667	0
43	1.833333	0	0.028868	0.05	0.057735	0.05	0.028868	0
44	2	0	0.033333	0.057735	0.066667	0.057735	0.033333	0
45	2.166667	0	0.028868	0.05	0.057735	0.05	0.028868	0
46	2.333333	0	0.016667	0.028868	0.033333	0.028868	0.016667	0
47	2.5	0	-1.0E-20	-2.7E-20	3.4E-21	-1.0E-20	6.8E-21	0

CD.WK3 CMD

Fig. 12. The solution of the one-dimensional heat conduction problem for $I = 6$ and $R = 1$ using CD.WK3
(a) The solution for 15 t-steps



(b) Variations of u with x as displayed on the screen



(c) Variations of u with t as displayed on the screen

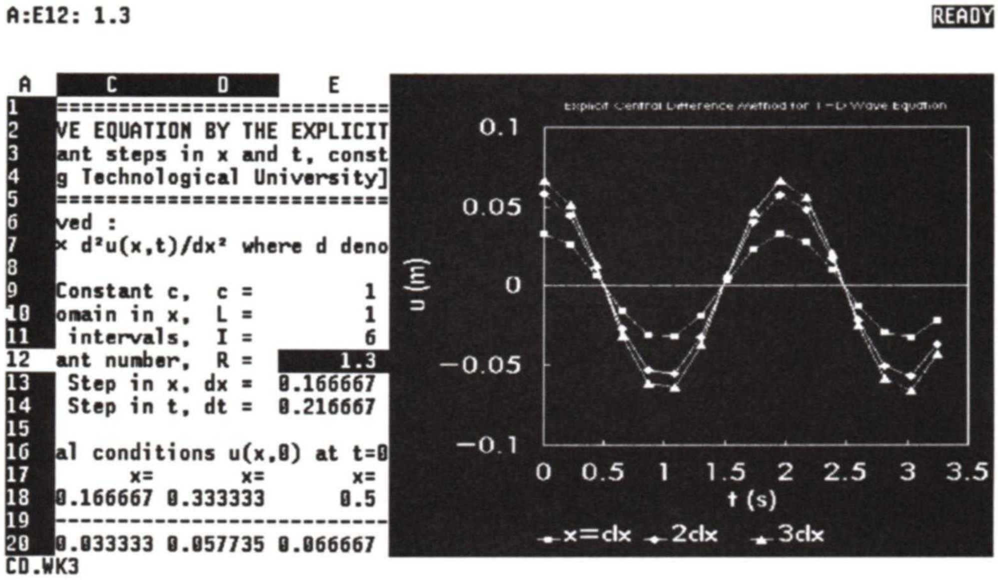
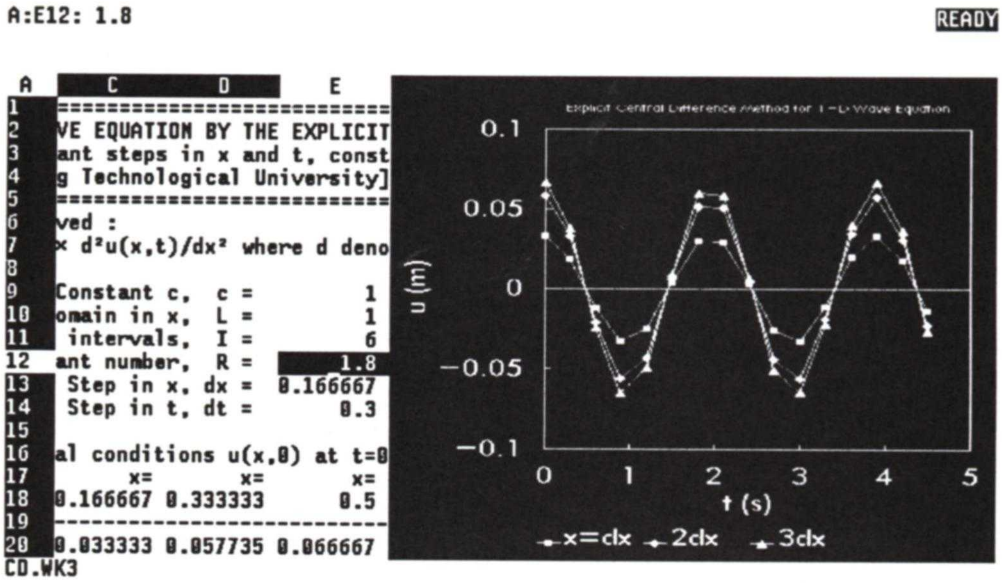
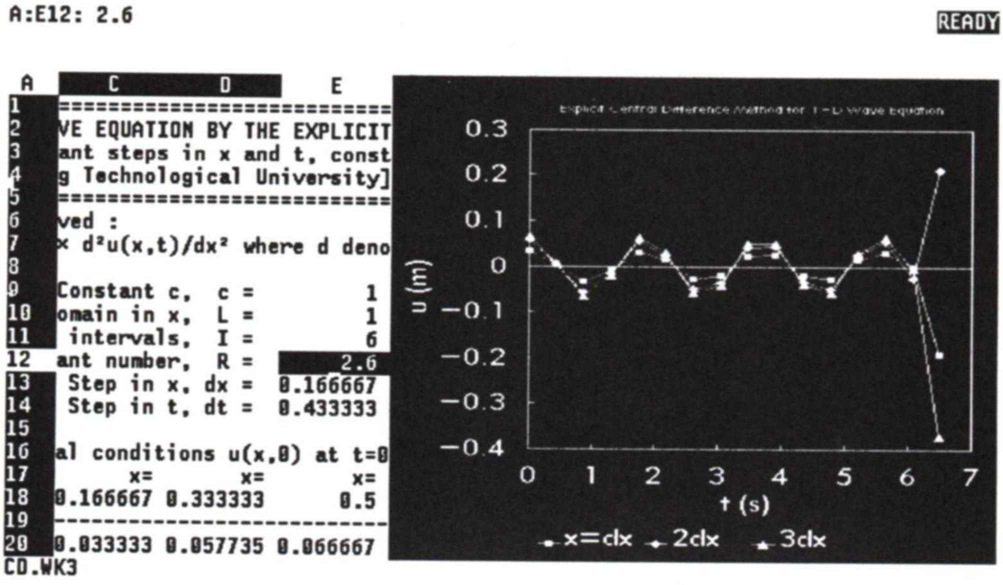


Fig. 13. The effects of varying R using CD.WK3

(a) $R = 1.3$



(b) $R = 1.8$



(c) $R = 2.6$

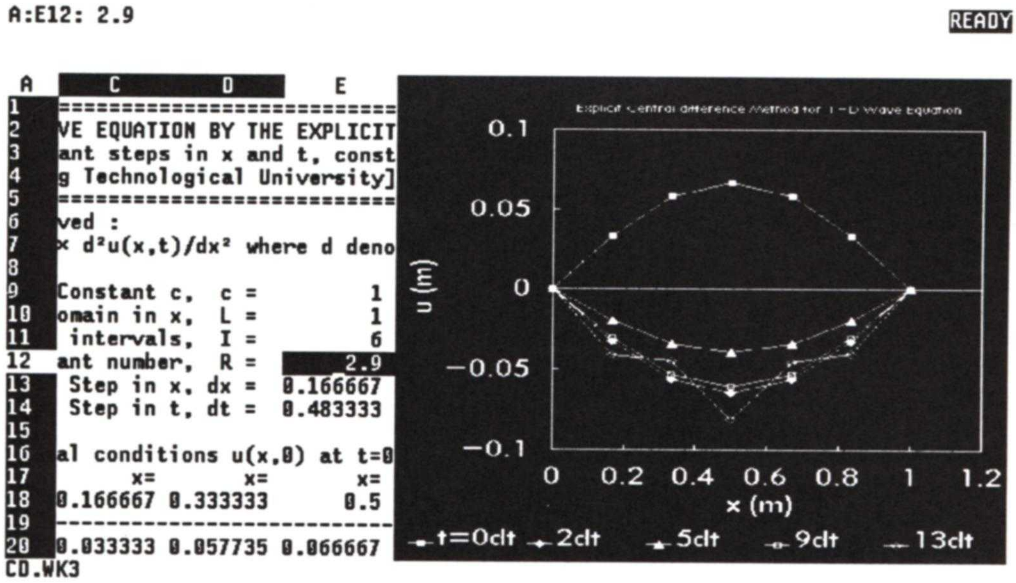


Fig. 14. The results for $R = 2.9$ using CD.WK3

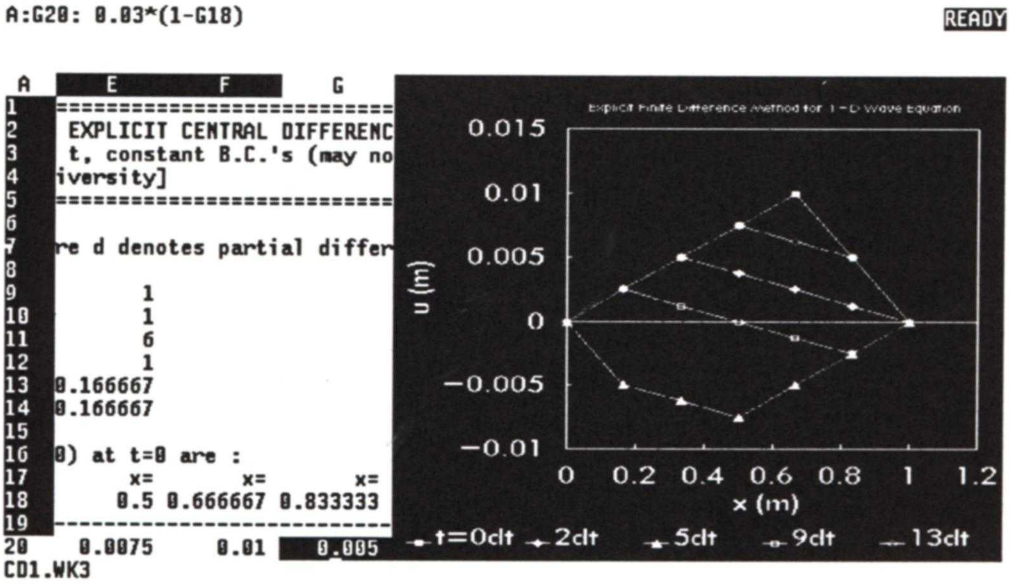


Fig. 15. The effects of changing the initial condition $u(x, 0)$ to $u(x, 0) = 0.015x$ for $0 \leqslant x \leqslant 4\Delta x$, $u(x, 0) = 0.03(1 - x)$ for $4\Delta x \leqslant x \leqslant 1$ using CD.WK3

REFERENCES

1. C. Y. Lam, Spreadsheet approach to partial differential equations. Part 1: Elliptic equation, *Int. J. Engng Educ.*, **8**, 278–287 (1992).
2. Lotus Development Corporation, *Lotus 1-2-3 Release 3.0 Reference*. Lotus Development Corporation, Cambridge (1989).
3. J. H. Ferziger, *Numerical Methods for Engineering Application*. Wiley-Interscience, New York (1981).
4. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes—The Art of Scientific Computing (Fortran Version)*. Cambridge University Press, Cambridge (1989).
5. J. D. Hoffman, *Numerical Methods for Engineers and Scientists*. McGraw-Hill, New York (1992).