Afyon Kocatepe Üniversitesi
FEN BİLİMLERİ DERGİSİ

4 (1-2)
135-145

Afyon Kocatepe University
JOURNAL OF SCIENCE

STUDY ON 9 POINT FTCS TWO-LEVEL FINITE-DIFFERENCE METHODS WITH LOD PROCEDURE FOR TWO-DIMENSIONAL DIFFUSION EQUATION

Mustafa GÜLSU

Muğla Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü, MUĞLA

ABSTRACT

Finite-difference techniques based on Explicit method and 9-point forward time centered space (FTCS) method for one dimensional diffusion are used to solve the two-dimensional time dependent diffusion equation with boundary condition. In these cases locally one-dimensional (LOD) techniques are used to extend the one-dimensional techniques to solve the two-dimensional problem. The results of numerical testing shows that these schemes uses less central processor (CPU) time than the fully implicit schemes.

Key Words: Finite Difference, LOD Method, Explicit Method, FTCS.

İKİ BOYUTLU DİFFUZYON DENKLEMİ İÇİN LOD YÖNTEMİ İLE AÇIK SONLU FARK METODLARI ÜZERİNE

ÖZET

Bu çalışmada bir boyutlu diffüzyon denklemi için Açık yöntem ve 9-Nokta ileri fark yöntemini temel alan sonlu fark teknikleri, iki boyutlu zaman bağımlı diffüzyon denklemini çözmek için kullanıldı. Yerel bir boyut (LOD) yöntemi iki boyutlu diffüzyon denklemini çözmek için genişletildi. Nümerik sonuçlar ile bu yöntemin kapalı yöntemlere göre daha az zaman (CPU) harcadığı gösterildi.

Anahtar Kelimeler: Sonlu Farklar, LOD Yöntemi, Açık Yöntem, İleri Fark Yöntemi,

1. INTRODUCTION

The constant-coefficient two-dimensional diffusion equation, namely

$$\frac{\partial u}{\partial t}=\alpha_x\frac{\partial^2 u}{\partial x^2}+\alpha_y\frac{\partial^2 u}{\partial y^2},\ 0\leq x\leq M\ ,\ 0\leq y\leq N\ ,\ 0\leq t\leq T \tag{1.1}$$

where α_x and α_y are the coefficients of diffusion in the x and y directions respectively, has many applications to practical problems, including the flow of groundwater, and the diffusion of heat through solids. For many years the standart explicit two-level finite difference method for solving (1.1) was the classical explicit forward-time centred-space method described in Noye B. J. and Hayman K. J. [4].

Recent improvements include the efficient alternating group explicit method of Dehghan M. [2]. The present article investigate the development of a fourth-order accurate two-level explicit finite difference method for solving (1.1) subject to Drichlet boundary condition. In particular locally one dimensional (LOD) methods and 9-point forward time centered space methods are investigated.

For convenience, a method which uses a computational molecule that involves m_1 grid points from time level (n+1) and m_2 grid points from time level n is denoted as an (m_1,m_2) methods. Also, the grid point $(i\Delta x,j\Delta y,n\Delta t)$ $i=0,1,2,...I$, $j=0,1,2,...,J$, $n=0,1,2,...K$ where $\Delta x=M/I$, $\Delta y=N/J$, $\Delta t=T/K$, is referred to as the (i,j,n) grid point. At this point the partial differential equation (PDE) (1.1) is discretised to give the aproximating finite difference equation (FDE)

$$u_{i,j}^{n+1}=\sum_l\sum_m a_{l,m}u_{i+l,j+m}^{\ n} \tag{1.2}$$

The coefficient $a_{l,m}$ are functions of the non dimensional diffusion numbers

$$r_x=\alpha_x\frac{\Delta t}{(\Delta x)^2}\ ,\ r_y=\alpha_y\frac{\Delta t}{(\Delta y)^2} \tag{1.3}$$

Theoratical comparisons of the order of convergence of various finite-difference methods are based on the leading error terms in their modified equivalent partial differential equations (MEPDE) which have the general form

$$\frac{\partial u}{\partial t}-\alpha_x\frac{\partial^2 u}{\partial x^2}-\alpha_y\frac{\partial^2 u}{\partial t^2}+\sum_{p=3}^{\infty}\sum_{q=0}^{p}C_{p,q}\frac{\partial^p}{\partial x^{p-q}\partial y^q}=0 \tag{1.4}$$

where the $C_{p,q}$ are coefficients of errors term. Given that (1.2) is consistent with the two-dimensional diffusion equation (1.1) which requires that

$$Lim_{\Delta x,\Delta y,\Delta t\to 0}C_{p,q}=0 \text{ for } p\geq 0, \tag{1.5}$$

the error coefficient $C_{p,q}$ in the MEPDE can be written in the form;

$$C_{p,q}=\begin{cases}\dfrac{2\alpha_x(\Delta x)^{p-2}}{p!}\Gamma_{p,q}(r_x,r_y)..................q=0\\ \dfrac{2\alpha_y(\Delta y)^{p-2}}{p!}\Gamma_{p,q}(r_x,r_y)..................q=p\\ \dfrac{4\alpha_x(\Delta x)^{p-q-2}(\Delta y)^q}{(p-q)!q!}\Gamma_{p,q}(r_x,r_y).......otherwise\end{cases} \tag{1.6}$$

It can be seen from (1.4) that the error term associated with the coefficients $C_{p,q}$ are of the order (p-2) in Δx and Δy. The order of accuracy of an FDE which approximately solves (1.1) is the smallest order of any error term present in the corresponding MEPDE. Hence if the leading error term in the MEPDE is $C_{P,q}$ for any q=0,1,2,...,P then the FDE is order (P-2) accurate.

In the following the time-stepping stability of the FDE (1.2) is established by means of the von Neumann method.

In order to verify theoretical predictions, numerical test were carried out on a two dimensional time-dependent diffusion equation:

$$\frac{\partial u}{\partial t}=\alpha_x\frac{\partial^2 u}{\partial x^2}+\alpha_y\frac{\partial^2 u}{\partial y^2} \tag{1.7}$$

u(x,y,0)=f(x) = exp(x+y) , $0\leq x,y\leq 1$ (1.8)

u(0,y,t)=g_0(y,t)= exp(y+2t) , $0\leq t\leq T, 0\leq y\leq 1$

u(1,y,t)= g_1(y,t)= exp(1+y+2t), $0\leq t\leq T, 0\leq y\leq 1$

u(x,1,t)= h_1(x,t)= exp(1+x+2t), $0\leq t\leq T, 0\leq x\leq 1$

u(x,0,t)= h_0(x,t)= exp(x+2t), $0\leq t\leq T, 0\leq x\leq 1$

2. LOD METHODS

Partial Differential Equation (1.1) can be solved by splitting it into two one-dimensional equation

$$\frac{1}{2}\frac{\partial u}{\partial t} = \alpha_x \frac{\partial^2 u}{\partial x^2}$$
$$\frac{1}{2}\frac{\partial u}{\partial t} = \alpha_y \frac{\partial^2 u}{\partial y^2} \tag{2.1}$$

rather than discretising the complete two-dimensional diffusion equation to give an approximating finite-difference equation based on a two-dimensional computational molecule. Each of these equations is then solved over half of the time step used for the complete two-dimensional equation using techniques for the one dimensional problems. This is advantageous since accurate and stable techniques for one -dimensional diffusion are much easier to develop and use than single step methods for two-dimensional diffusion equation.

Commencing with the initial condition for each n=0,1,2,...,K the process of stepping from time t_n to t_{n+1} is carried out in two stages. In the first stage , in advancing from t_n=nk to the time $t_{n+\frac{1}{2}} = (t_n + \frac{k}{2})$, the partial differential equation

$$\frac{1}{2}\frac{\partial u}{\partial t} = \alpha_x \frac{\partial^2 u}{\partial x^2} \tag{2.2}$$

is solved numerically at the spatial points (x_i, y_j), i=1,2,...,I-1 for each j=0,1,...,J.

Commencing with previously computed values $u_{i,j}^{\ n}$ i,j=1,2,...,M-1 and boundary values:

$$u_{0,j}^{\ n} = g_0(y_j, t_n) \ , \ j=0,1,2,...,J \tag{2.3}$$
$$u_{M,j}^{\ n} = g_1(y_j, t_n) \ , \ j=0,1,2,...,J$$

results in the set of approximate values $u_{i,j}^{n+\frac{1}{2}}$, i=1,2,...,I-1 , j=0,1,...,J being found at the intermediate time $t_{n+\frac{1}{2}}$. Then in advancing from the time $t_{n+\frac{1}{2}}$ to $t_{n+1}=(t_n+k)$ the equation:

$$\frac{1}{2}\frac{\partial u}{\partial t}=\alpha_y\frac{\partial^2 u}{\partial y^2} \tag{2.4}$$

is solved numericaly at the spatial points (x_i,y_j) , commencing with initial values
$u_{i,j}^{n+\frac{1}{2}}$, i=1,2,...,I-1 , j=1,2,...,J-1 and using as boundary values $u_{i,0}^{n+\frac{1}{2}}$ and $u_{i,M}^{n+\frac{1}{2}}$ i=1,2,...,I-1. Not that the boundary conditions (1.8) are not used at the intermediate time $t_{n+\frac{1}{2}}$. This is because in the time interval t_n to $t_{n+\frac{1}{2}}$, the process of diffusion in the x-direction has been applied with a diffusion coefficient which is twice that in the original equation (1.1) as can be seen by rearranging in the form

$$\frac{\partial u}{\partial t}=2\alpha_x\frac{\partial^2 u}{\partial x^2} \tag{2.5}$$

Not that the values of $u_{i,j}^{n+\frac{1}{2}}$ i=1,2,...,I , j=1,2,...,J are not approximate solutions to the original problem.

Let's running the LOD process using explicit method for which the correct two-stage procedure is:

$$\frac{u_{i,j}^{n+\frac{1}{2}}-u_{i,j}^{n}}{k}=\frac{1}{h^2}\left\{u_{i-1,j}^{n}-2u_{i,j}^{n}+u_{i+1,j}^{n}\right\} \tag{2.6}$$

for each j=0,1,2,...,J apply

$$u_{i,j}^{n+\frac{1}{2}}=r_x\left(u_{i-1,j}^{n}+u_{i+1,j}^{n}\right)+(1-2r_x)u_{i,j}^{n} \tag{2.7}$$

for each i=1,2,...,I-1 then for each i=1,2,...,I-1 apply

$$u_{i,j}^{n+1} = r_y\left(u_{i,j-1}^{n+\frac{1}{2}} + u_{i,j+1}^{n+\frac{1}{2}}\right) + (1-2r_y)u_{i,j}^{n+\frac{1}{2}} \quad (2.8)$$

for each j=1,2,...,J-1.

These are von Neumann stable for $0 < r_x \leq \frac{1}{2}$, $0 < r_y \leq \frac{1}{2}$.

If $r_x=r_y=r^*=1/6$ the results obtained should be fourth-order accurate and if $r_x=r_y=r^*=1/2$ the results should be second –order accurate.

However, when known boundary values $u_{i,0}^{n+\frac{1}{2}}$, $u_{i,j}^{n+\frac{1}{2}}$, i=0,1,...,I for the complete problem computed using (1.4) are used instead of those calculated using (2.7) the result shown in Figure1. indicate that this LOD procedure has produced only second order results, for s*=1/6 the slope of the line of best fit which gives an estimate of the order of convergence of the error, is 1.82 and for r*=1/2 it is 2.02.

The correct boundary values to be used along y=0 and y=N at the intermediate time level for the second half-time step are those obtained from the boundary values at the previous time t_n by applying the one-dimensional finite-difference equation being used elsewhere in the interior of the region. Note that end-points values along x=0 and x=M at the intermediate time level are not required in the second stage.

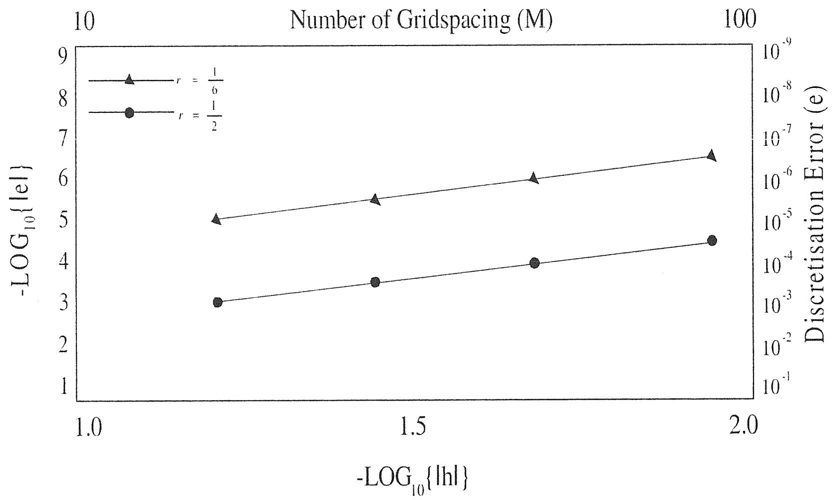

Figure 1. Relation between error u and gridspacing for exp.method with using (1.4)

The numerical result obtained with this procedure are shown in Figure 2. It is clear that the errors when r*=1/6 are now of order fourth. In fact, the slope of the line of best fit for r*=1/6 is 4.01 while that for r*=1/2 is 2.02. This clearly shows that the correct treatment of the boundaries at the intermediate time level for any time-splitting procedure is very important in the generation of the final solution.

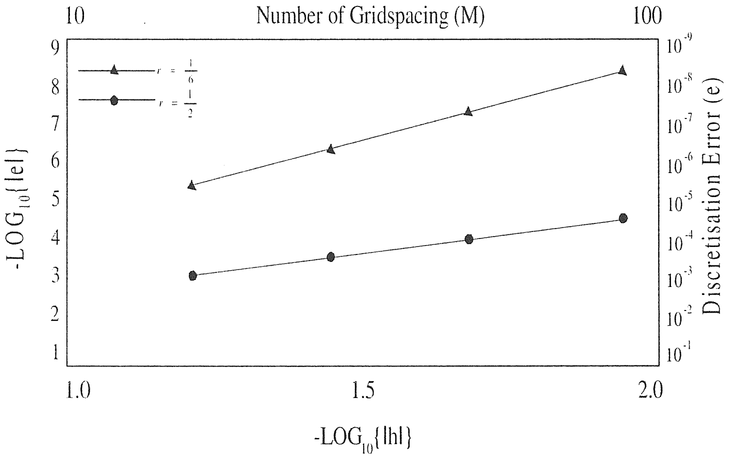

Figure 2. Relation between error u and gridspacing for exp. method with using (2.7)

3. THE (1,9) FTCS EQUATION

The 9-point fully explicit finite- difference method uses a forward-difference approximation for the time derivative and following weighted approximation for the derivatives. Using the finite –difference approximation

$$\frac{\partial^2 u}{\partial x^2} \approx \varphi\{[CS(i,j-1,n)]+[CS(i,j+1,n)]+(1-2\varphi)[CS(i,j,n)] \quad (3.1)$$

$$\frac{\partial^2 u}{\partial y^2} \approx \gamma\{[CS(i-1,j,n)]+[CS(i+1,j,n)]+(1-2\gamma)[CS(i,j,n)] \quad (3.2)$$

in which φ and γ are weights which depend on r_x and r_y and CS is used to denote the second-order centred-difference approximation in the appropriate direction about the grid point.This differencing yields the explicit (1,9) FTCS difference equation:

$$u_{i,j}^{n+1} = \{\varphi r_x + \gamma r_y\}(u_{i-1,j-1}^{\;n} + u_{i+1,j-1}^{\;n} + u_{i-1,j+1}^{\;n} + u_{i+1,j+1}^{\;n}) \qquad (3.3)$$

$$+\{r_y-2(\varphi r_x+\gamma r_y)\}(u_{i,j-1}{}^n+u_{i,j+1}{}^n)$$
$$+\{r_x-2(\varphi r_x+\gamma r_y)\}(u_{i-1,j}{}^n+u_{i+1,j}{}^n)$$
$$+\{1-2(r_x+r_y)+4(\varphi r_x+\gamma r_y)\}u_{i,j}{}^n$$

These approximation can be written as follows:

$$\frac{\partial u}{\partial t}=\frac{u_{i,j}{}^{n+1}-u_{i,j}{}^n}{k} \quad (3.4)$$

$$\frac{\partial^2 u}{\partial x^2}=\frac{r_x}{2}\left(\frac{(u_{i+1,j-1}{}^n-2u_{i,j-1}{}^n+u_{i-1,j-1}{}^n)}{(\Delta x)^2}+\frac{(u_{i+1,j+1}{}^n-2u_{i,j+1}{}^n+u_{i-1,j+1}{}^n)}{(\Delta x)^2}\right)+$$
$$+(1\text{-}r_x)\left(\frac{u_{i+1,j}{}^n-2u_{i,j}{}^n+u_{i-1,j}{}^n}{(\Delta x)^2}\right) \quad (3.5)$$

$$\frac{\partial^2 u}{\partial y^2}=\frac{r_y}{2}\left(\frac{(u_{i-1,j+1}{}^n-2u_{i-1,j}{}^n+u_{i-1,j-1}{}^n)}{(\Delta y)^2}+\frac{(u_{i+1,j+1}{}^n-2u_{i+1,j}{}^n+u_{i+1,j-1}{}^n)}{(\Delta y)^2}\right)+$$
$$+(1\text{-}r_y)\left(\frac{u_{i,j+1}{}^n-2u_{i,j}{}^n+u_{i,j-1}{}^n}{(\Delta y)^2}\right) \quad (3.6)$$

These approximations produce the following finite-difference equation:

$$u_{i,j}{}^{n+1}=r_xr_y(u_{i-1,j-1}{}^n+u_{i-1,j+1}{}^n+u_{i+1,j-1}{}^n+u_{i+1,j+1}{}^n)+r_y(1-2r_x)$$
$$(u^n{}_{i,j-1}+u^n{}_{i,j+1})+r_x(1-2r_y)(u_{i-1,j}{}^n+u_{i+1,j}{}^n)+(1-2r_x)(1-2r_y)u_{i,j}{}^n \quad (3.7)$$

If $r_x=r_y=r$, then (3.7) becomes

$$u_{i,j}{}^{n+1}=r^2(u_{i-1,j-1}{}^n+u_{i-1,j+1}{}^n+u_{i+1,j-1}{}^n+u_{i+1,j+1}{}^n)+r(1-2r)$$
$$(u_{i,j-1}{}^n+u_{i,j+1}{}^n)+r(1-2r)(u_{i-1,j}{}^n+u_{i+1,j}{}^n)+(1-2r)^2u_{i,j}{}^n \quad (3.8)$$

The range of stability corresponding to (3.8) is $0<r\le 1/2$

In Table1 the results are shown for $u_{i,j}^{\ n}$ with Δx=Δy=h=0.05 and r=1/2 at T=1.0 using explicit method and the (1,9) FTCS formula with given boundary values everywhere.

When the absolute value of the error;

$$e_{i,j}^{\ n} = u(ih, jk, nk) - u_{i,j}^{\ n} \tag{3.9}$$

at the point (0.5,0.5) at time T=1.0 was graphed against h on a logarithmic scale for various of r it was found that the slope of lines was always close to 2 for 9-point FTCS formula. These results illustrate the theoretical orders of accuracy evident from the modified equivalent equation.

The numerical results obtained with the fourth-order one dimensional equation are shown in Figure1. The slopes of the lines of best fit to the results are very close to 4 for each r. These results indicate that this fourth-order technique is much more accurate then the second-order method based on the FTCS formula. For example with r=1/2 and Δx=0.5 the error produced by the second –order method was about 10^{-4} while that produced by the fourth-order technique was about 10^{-7}.

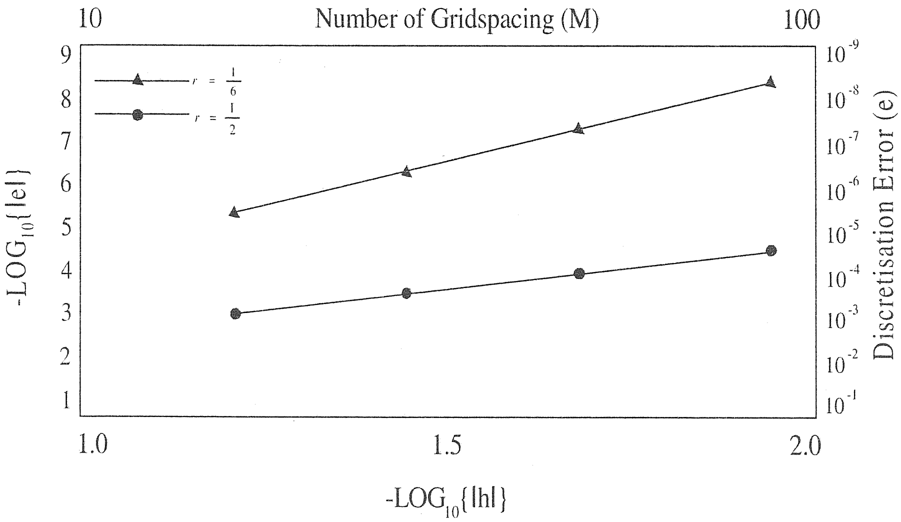

Figure 3. Relation between error u and gridspacing for 9-point FTCS method

Table 1. Results for u with T=1.0 , h=0.05, r=1/2

x	y	Exp-Method	(1,9)FTCS Method	Exp-Error	(1,9)-Error	Analitical Solutions
0.1	0.1	9.024880145	9.025067499	0.00133354	0.000054	9.025013499
0.2	0.2	11.02277750	11.02316638	0.00039888	0.000100	11.02317638
0.3	0.3	13.46303642	13.46364604	0.00070162	0.000092	13.46373804
0.4	0.4	16.44367357	16.44447677	0.00097320	0.000170	16.44464777
0.5	0.5	20.08438023	20.08531692	0.00115669	0.000220	20.08553692
0.6	0.6	24.53132433	24.53227020	0.00120587	0.000260	24.53253020
0.7	0.7	29.96301156	29.96386005	0.00108849	0.000240	29.96410005
0.8	0.8	36.59744019	36.59804444	0.00079425	0.000190	36.59823444
0.9	0.9	44.70082306	44.70109849	0.00036143	0.000086	44.70118449

Overall, it can be seen that LOD techniques provide an effective solution to the two-dimensional problem. However, it must be kept in mind that proper treatment is required with the LOD procedure to obtain the correct values to be used on the boundary at intermediate time levels.

4.CONCLUSION

In this paper time-split finite difference method have been used to solve the two-dimensional constant coefficient diffusion equation with given boundary values. Using the Explicit method for one-dimensional diffusion equation in a LOD procedure with special treatment on the boundaries at the intermediate time level gave fourth-order accuracy. Without the special boundary treatment at the intermediate time levels high-order methods used at interior grid points in an LOD procedure only produce low-order results.

A comparison with the implicit scheme for the test problem clearly demonstrates that this technique are computationally superior. The numerical test obtained by using these methods give acceptable results. Also (1,9) FTCS method produced second-order results. It used more CPU time than the fourth-order LOD procedure to get results of the same accuracy.

REFERENCES

1. Ames W. F., Numerical Methods for Partial Differential Equations, Academic Press, New York, (1992).
2. Dehghan M., Implicit Locally One-Dimensional Methods for Two-Dimensional Diffusion with a Non-Local Boundary Condition, Math. And Computers in Simulation, 49(4), 331-349, (1999).
3. Herceg D., Cvetkovic L., On a Numerical Differentiation, SIAM J. Numer. Anal., 23 (3), 686-691, (1986).
4. Noye B. J., Hayman K. J., Explicit Two-Level Finite –Difference Methods for The Two-Dimensional Diffusion Equation, Intern.J.Computer Math., 42, 223-236, (1991).
5. Smith G. D., Numerical Solution of Partial Differantial Equations, Oxford Univ. Press, Oxford, (1985).
6. Vanaja V., Kellogg R. B., Iterative Methods for a Forward-Backward Heat Equation, SIAM J. Numer. Anal., 27 (3), 622-635, (1990).