Finite Difference Methods

Kumail Raza,* **Sir Abdul Wali Khan**†
Faculty of Institute of Numerical Sciences, KUST Institute, Tapaee 26000, KPK, Pakistan.

1 INTRODUCTION TO NUMERICAL METHOD OF PDE,s

2 Forward Difference method

In this chapter we establish some theoretical discussion of few finite difference scheme namely FTCS scheme, BTCS scheme and Theta method for numerically solving parabolic partial differential equations. Here we take heat equation with Dirichlet boundary conditions for simplicity. We also discuss the local truncation error, consistency, stability and convergence of the FTCS scheme, BTCS scheme and Theta method with example.

2.1 FTCS

Let us consider the heat equation

$$u_t = u_{xx}, \qquad 0 < x < 1, \qquad t > 0 \tag{2.1}$$

We subdivide the space (x) and time (t) domains uniformly. The space grid size is $\Delta x = \frac{1}{N}$ for some integer N and time step is Δt (unspecified). Using forward difference approximation for u_t and central difference approximation for u_{xx} where $x_j = j\Delta x$ and $t_{n=n\Delta t}.(x_0, t_0) = (0, 0)$ which give the following approximation of PDE at (x_j, t_n).
Note $(\Delta t, k)$ And $(\Delta x, h)$ Time and Space variables and (r, λ).

$$u_t = \frac{F_t u_j^n}{\Delta t} = \frac{u_j^{n+1} + u_j^n}{\Delta t} \quad and \quad u_{xx} = \frac{\delta x^2 u_j^n}{\Delta x^2} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \tag{2.2}$$

The Eq(1.1) becomes,

$$\frac{u_j^{n+1} + u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \tag{2.3}$$

At all interior points for $(j = 1 : N - 1)$ and $(n \geq 0)$. Further this can be simplified as
Where

$$r, \lambda = \frac{\Delta t}{\Delta t^2}$$

Where r, λ is the mesh ratio and very important quantity, later we will discuss it. The scheme gives u_j^{n+1} for $, j = 1, 2, \ldots, N$ in terms of solution components at a time t_n, and hence progresses one full time level after another. The stencil for FTCS scheme is shown in Figure (2.1)

*kumailraza050gmail.com (corresponding author)
†−walikhanmashwani@gmail.co.in

2.2 FTCS Algorithm

(1) Choose N, r and find Δx and Δt,
(2) Use the initial condition to find the value at the $t = 0$,

$$u_j^0 = f(x) Where j = 1, 2, \ldots, N-1 and set n \geq 0,$$

(3) Use FTCS scheme to find values of the solution at the interior nodes,
(4) Use boundary conditions to find boundary value at time t_{n+1}

$$u_0^{n+1} = g(t_{n+1}), u_N^{n+1} = h(t_{n+1}).$$

The FTCS scheme is explicit, it does not require any algebraic equations to be solved to calculate the solution at time t_{n+1} from a solution at time t_n.

2.3 Manual Example

We use FTCS scheme to solve the given heat equation.

$$u_t = u_{xx}, \quad 0 < x < 1, \quad t > 0 \tag{2.4}$$

With boundary conditions

$$u(0,t) = 0 = u(1,t), \quad 0 \leq t, \tag{2.5}$$

And initial condition

$$u(x,0) = sin(\pi x), \quad 0 \leq x \leq 1, \tag{2.6}$$

Use the step size($N = 3$), ($r, \lambda = 0.45$) up to two time level.
Solution:

$$\Delta x = \frac{1}{N} = \frac{1}{3} \tag{2.7}$$

Since

$$\Delta t = r * \Delta x^2 \quad so \quad \Delta t = 0.05. \tag{2.8}$$

Initial condition

$$u(x,0) = sin(\pi x), \quad 0 \leq x \leq 1, \tag{2.9}$$

$u_j^0 = sin(\pi x_j), \quad j = 0,1,2,3 \quad Since \quad x_j = j\Delta x, \quad j = 0,1,2,3$
$u_0^0 = 0, \qquad x_0 = 0$
$u_1^0 = 0.8660 \qquad x_1 = \frac{1}{3}$
$u_2^0 = 0.8660 \qquad x_2 = \frac{2}{3}$
$u_3^0 = 0 \qquad x_3 = 1$
Boundary conditions
$u(0,t) = 0 \qquad t \geq 0$ That is
$u_0^{n+1} = 0, \qquad n \geq 0$
And
$u(1,t) = 0 \qquad t \geq 0$
$u_3^{n+1} = 0, \qquad n \geq 0$
Since the FTCS scheme is given by

Table 1: In below Table 3.2 we can see the result from left to right at each time level

Time	$x=0$	$x=\frac{1}{3}$	$x=\frac{2}{3}$	$x=1$
$t=0$	0	0.8660	0.8660	0
$t=0.05$	0	0.4763	0.4763	0
$t=0.1$	0	0.2620	0.2620	0

2.4 1^{st} time level (means $n=0$)

$$u_j^1 = ru_{j-1}^0 + (1-2r)u_j^0 + ru_{j+1}^0, \quad j=1,2$$

For $j=1$

$u_1^1 = ru_0^0 + (1-2r)u_1^0 + ru_2^0$

$u_1^1 = (0.45*0) + (1-2*0.45)*0.8660 + (0.45*0.8660)$

$u_1^1 = 0.4763$

2.5 2^{st} time level (means $n=1$)

The scheme is

$u_j^2 = ru_{j-1}^1 + (1-2r)u_j^1 + ru_{j+1}^1, \quad j=1,2$

For j=1

$u_1^2 = ru_0^1 + (1-2r)u_1^1 + ru_2^1$

$u_1^2 = (0.45*0) + (1-2*0.45)*0.4763 + (0.45*0.4763)$

$u_1^1 = 0.2620$

For j=2

$u_2^2 = ru_1^1 + (1-2r)u_2^1 + ru_3^1$

$u_2^2 = (0.45*0.4763) + (1-2*0.45)*0.4763 + (0.45*0)$

$u_2^2 = 0.2620$

2.6 Local Truncation Error (LTE)

The questions arise that how can one tell that whether an approximation of $u_t = u_{xx}$ is any good? The PDE corresponds to the operator equation $Lu=0$, where L is the differential operator

$$L = \frac{\partial}{\partial t} - \frac{\partial^2}{\partial t^2}$$

(so that $L = u_t - u_{xx}$, and suppose that L_Δ is a finite difference approximation of L with time mesh size Δt and space mesh sizes Δx. For example, the operator L_Δ for the FTCS scheme is

$$L_\Delta = \frac{F_t}{\delta t} = \frac{\delta^2 x}{\Delta x^2}$$

so that

$$L_\Delta u_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n - u_{j-1}^n}{\Delta x^2}$$

We want the solution u_j^n of the difference scheme to approximate the exact solution u of the PDE at the point (x_j, t_n) with the approximate getting better as the mesh take smaller. For this to happen we need L_Δ to "look like" L as $\Delta s \to 0$ and $\Delta x \to 0$. The LTE tells us how well L_Δ close to L.

2.7 Stability Analysis for FTCS scheme

We have seen for FTCS scheme that if we take x and t are small enough then the FTCS approximation operator L_Δ look like differential operatorL, that the LTE approaches to zero means that the FTCS scheme gives better result for $u_t = u_{xx}$. Is this enough guarantee that the approximate solution looks like the exact solution.
Consider the example

$$u_t = u_{xx}, \quad 0 < x < 1, \quad t > 0$$

With boundary conditions

$$u(0,t) = 0 = u(1,t), \quad \forall 0 \le t$$

, And initial condition

$$u(x,0) = sin(\pi x), \quad 0 \le x \le 1$$

, We compare the exact solution which is $u(x,t) = e^(- \pi^2 t)\ sin\pi x$ with the approximate solution obtained by FTCS scheme. Mesh plot of the exact solution and the approximate solutions calculated using $r = 0.45$ and $N = 2,4$ and 10 Figure 2.2 show that exact solution of that given problem, while Figure $2.3, 2.4$ and 2.5 show that approximate solution at $N = 2,4$ and 10 respectively, and the different time levels. put graphes

3 BTCS Scheme

Let us consider the heat equation

$$u_t = u_{xx}, \quad 0 < x < 1, \quad t > 0,$$

with initial and boundary conditions

$$u(x,0) = f(x), \quad 0 \le x \le 1$$

$$u(0,t) = g_0(t), \quad u(1,t) = g_1(t), \quad t \ge 0,$$

we subdivide the space (x) and time (t) domains uniformly. The space grid size is $\Delta x = \frac{1}{N}$ for some integer N and time step δt (unspecified). Using backward difference approximation for u_t and central difference approximation for u_{xx} where $x_j = j\Delta x$ and $t_n = n\Delta t$ which give the following approximation of PDE at (x_j, t_{n+1}).

$$u_t = \frac{B_t u_j^{n+1}}{\Delta t} = \frac{u_j^{n+1} - u_j^n}{\Delta t} \quad and \quad u_{xx} = \frac{\delta_x^2 u_j^{n+1}}{\Delta x^2} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2},$$

The equation (12) becomes

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2},$$

At all interior point for $j = 1 : N - 1$ and $n \ge 0$. Further this can be simplified as

$$u_j^n = -ru_{j-1}^{n+1} + (1 + 2r)u_j^{n+1} - ru_{j+1}^{n+1}, \quad j = 1, 2, ..., N - 1, \quad n \ge 0,$$

$$where \frac{\Delta t}{\Delta x^2}.$$

Equation (13) is called BTCS scheme for $u_t = u_{xx}$. This scheme is implicit because it involves more than one component of u^{n+1}.
The difference between FTCS and BTCS is that FTCS scheme involves $\delta_x^2 u_j^n$ (i.e at the old time level, $t = t_n$) and BTCS involves $\delta_x^2 u_j^{n+1}$ (i.e at new time level, $t = t_{n+1}$).

Diagram

3.1 Matrix form of BTCS Scheme

Note that the BTCS scheme is contrast to FTCS scheme, we have now three unknown in equation (13).Since this equation hold for $j = 1, 2, ..., N-1$ so we have $N-1$ unknown. Now we write these $N-1$ equation in a matrix form for finding these unknown.
since we have $u(x, 0) = f(x), \quad 0 \leq x \leq L.$
so we have

$$u_j^0 = f(x), \quad \forall j = 1, 2, ..., N,$$

Then we generate the next t-row by using equation (13)

3.2 $(\pi = 0, 1^{st}$ time level)

$n \times n$ matrix

$$\begin{bmatrix} 1+2r & -r & 0 & ... & ... \\ -r & 1+2r & -r & 0 & ... \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & 0 & -r & 1+2r \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_3^1 \\ \vdots \\ u_{N-1}^1 \end{bmatrix} \tag{3.1}$$