



A Generalize Formula for Parabolic Partial Differential Equation (PPDE) Using Matrix Analysis

Md.Sahidul Islam¹, Md. Reduanul Alam²

¹Department of Electronics and Communication Engineering
East West University

Aftabnagar, Dhaka-1212, Bangladesh

²Department of Mathematics

University of Chittagong

Chittagong-4338, Bangladesh

Abstract: In this paper we consider the generalize formula for Parabolic Partial Differential Equations (PPDE) using matrix analysis. Actually it is an improvement of generalize θ scheme. The generalize θ scheme combines three formula (one explicit and two implicit). Our proposed generalize formula combines five formulae (two explicit and three implicit).

Keywords: explicit and implicit method; toeplitz matrix; tridiagonal linear system; convex combination

I. Introduction

Different numerical methods are used for solving Parabolic Partial Differential Equation (PPDE). The numerical solution of PPDE in finite difference method can be solved by using explicit and implicit techniques.

Consider the parabolic partial differential equation with initial condition $u(x, 0) = f(x)$ and the boundary conditions $u(0, t) = u_a(t)$ and $u(L, t) = u_b(t)$

$$(1.1) \quad \frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$$

We can solve the PPDE via matrix [1] analysis. Let us rewrite the most general PPDE in matrix form. We observe that the decomposition of x, t ($1D + t$) plane consists of two stages; first, spatial decomposition (setting $x = a + ih$), and second, temporal decomposition (setting $t_j = jk$).

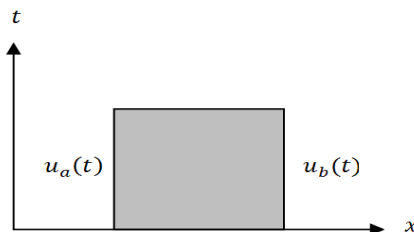


Figure 1: Solution domain for initial value problem

Next we evaluate the partial differential at an arbitrary grid point $x = x_i$. Replace the derivative on the right hand of the equation by its second-order central difference approximation. We denote v_i be the semi discrete approximation to $u(x_i, t_j)$, the function $u \approx v_i$,

$$(1.2) \quad \frac{dv_i}{dt} = D \frac{1}{(\Delta x)^2} \{v_{i-1}(t) - 2v_i(t) + v_{i+1}(t)\}$$

For $1 \leq i \leq N - 1$, these equations are supplemented by the boundary conditions $v_0(t) = u_a(t)$ and $v_N(t) = u_b(t)$. The initial value problem for the $v_i(t)$ can be easily be written in the matrix form. Let-

$$\underline{v}(t) = [v_1(t) \ v_2(t) \ v_3(t) \ \cdots \ v_{N-1}(t)]^T; \quad \underline{f} = [f(x_1) \ f(x_2) \ f(x_3) \ \cdots \ f(x_{N-1})]^T;$$

$$\underline{b}(t) = [-u_a(t) \ 0 \ 0 \ \cdots \ -u_b(t)]^T$$

$$\text{and } \underline{A} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \vdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

In most cases, $\mathbf{A}$ is called Toeplitz matrix. The order of $\mathbf{A}$, N_h depends on the space decomposition. With these definitions, equation (1.2) becomes-

$$(1.3) \quad \frac{dv_i}{dt} = -\frac{D}{(\Delta x)^2} [\mathbf{A}\underline{v} + \underline{b}(t)] \quad \underline{v}(0) = \underline{f}$$

To solve this PPDE (1.3), we use different finite difference explicit and implicit techniques. In this study we have proposed a generalize formula which combines five different finite difference methods.

II. Explicit Method

A. Schmidt method

For *Schmidt* [1], [5] method evaluate the equation (1.3) at the time level, $t = t_j$ and use first forward difference approximation for the time derivative, we get-

$$\frac{1}{(\Delta t)} [\underline{w}^{(j+1)} - \underline{w}^{(j)}] = -\frac{D}{(\Delta x)^2} [\mathbf{A}\underline{w}^{(j)} + \underline{b}(t)] \text{ and } \underline{w}^{(0)} = \underline{f}; \text{ where } \underline{w} = [w_1^{(j)} \quad w_2^{(j)} \quad w_3^{(j)} \quad \dots \quad w_{N-1}^{(j)}]^T$$

Solving the previous equation for $\underline{w}^{(j+1)}$ yields-

$$(1.4) \quad \underline{w}^{(j+1)} = (I - \lambda \mathbf{A})\underline{w}^{(j)} - \lambda \underline{b}^{(j)}; \text{ where, } \lambda = \frac{D\Delta t}{(\Delta x)^2}$$

This is an explicit equation for computing the approximate solution at one time level from the values of the solution at the previous time level with $\underline{w}^{(0)}$ known from initial condition. We can march forward one increment of Δt at a time. The matrix $(I - \lambda \mathbf{A})$ is called evolution matrix for the numerical method.

B. Richardson's method

In the *Richardson's* [1], [7] method we evaluate the equation at the time level $t = t_j$ and we use central difference approximation for the time derivative. Then equation (1.3) becomes

$$\frac{1}{2(\Delta t)} [\underline{w}^{(j+1)} - \underline{w}^{(j-1)}] = -\frac{D}{(\Delta x)^2} [\mathbf{A}\underline{w}^{(j)} + \underline{b}^{(j)}]$$

which gives-

$$(1.5) \quad \underline{w}^{(j+1)} = \underline{w}^{(j-1)} - \lambda [\mathbf{A}\underline{w}^{(j)} + \underline{b}^{(j)}]; \text{ where, } \lambda = \frac{2D\Delta t}{(\Delta x)^2}$$

This is another explicit formula. This formula has a special starting procedure. To use this formula we need the solution at time level $t = t_{j-1}$ which we get from *Schmidt method*.

III. Implicit Method

A. O'Brien Method

This method [1], [6] is obtained evaluating (1.3) at time level $(j+1)$ -th, $t = t_{j+1}$ and using the first order forward difference approximation for the time derivative. We get-

$$\frac{1}{(\Delta t)} [\underline{w}^{(j+1)} - \underline{w}^{(j)}] = -\frac{D}{(\Delta x)^2} [\mathbf{A}\underline{w}^{(j+1)} + \underline{b}^{(j+1)}]$$

Re-arranging terms in the evolution equation, we get

$$(1.6) \quad (I + \lambda \mathbf{A})\underline{w}^{(j+1)} = \underline{w}^{(j)} - \lambda \underline{b}^{(j+1)}; \text{ where } \lambda = \frac{D\Delta t}{(\Delta x)^2}$$

We see that O'Brien method defines $\underline{w}^{(j+1)}$ implicitly in term of $\underline{w}^{(j)}$. Since $I + \lambda \mathbf{A}$ is a tridiagonal matrix, computing $\underline{w}^{(j+1)}$ from the equation (1.5) requires only about twice as many algebraic operation as computing $\underline{w}^{(j+1)}$ (1.6).

B. Crank-Nicolson method

In Crank-Nicolson [1], [6] method the space derivative is replaced by the average of the matrix in (1.3) taken along $(j+1)$ and j -th level and time derivative is replaced by forward difference this yield

$$[\underline{w}^{(j+1)} - \underline{w}^{(j)}] = -\frac{D(\Delta t)}{2(\Delta x)^2} [\mathbf{A}\underline{w}^{(j)} + \underline{b}^{(j)} + \mathbf{A}\underline{w}^{(j+1)} + \underline{b}^{(j+1)}]$$

Then we have

$$(1.7) \quad (I + \lambda \mathbf{A})\underline{w}^{(j+1)} = (I - \lambda \mathbf{A})\underline{w}^{(j)} - \lambda(\underline{b}^{(j+1)} + \underline{b}^{(j)}); \text{ where } \lambda = \frac{D(\Delta t)}{2(\Delta x)^2}$$

This is another implicit formula requiring the solution of a Tridiagonal linear system at each time step.

C. Burnett method

In Burnett method [1], [2], first forward difference for the time derivative and the space derivative is replaced by the average [Multiplying the matrix in (1.3) taken along $(j+1)$ by 2 and adding it with the matrix in (1.3) taken along j -th level] of $(j+1)$ and j -th level evaluates (1.3). We have-

$$[\underline{w}^{(j+1)} - \underline{w}^{(j)}] = -\frac{D(\Delta t)}{3(\Delta x)^2} [\mathbf{A}\underline{w}^{(j)} + \underline{b}^{(j)} + 2\{\mathbf{A}\underline{w}^{(j+1)} + \underline{b}^{(j+1)}\}]$$

