Comparison of Analytical Methods to Finite Difference Methods for the One Dimensional Heat Equation

Alex Taylor and Vrushaly Shinglot *
Department of Mathematical Sciences
The University of Texas at Dallas

October 21, 2018

Abstract

In this review paper we present several methods for solving the one-dimensional partial differential heat equation both numerically and analytically. We obtain analytical solutions of this equation using separation of variables, and then we compare the solutions to numerical solutions derived from finite difference schemes. We then analyze the error between the various solutions theoretically and graphically, and develop criteria for assessing the stability of solutions.

1 Introduction

This paper was written for an undergraduate audience and is intended as a basic survey of several important methods for solving partial differential equations such as the heat equation. In the first section we recall basic preliminary definitions and notations from calculus such as the definition of partial derivative, boundary conditions for PDEs, etc., and we introduce the one dimensional heat equation.

1.1 Derivatives and Partial Derivatives

We will use several definitions of the derivative in the following sections.

Definition 1.1. Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function of x. The standard **forward difference derivative** of f is defined by the limit

$$f'(x) = \lim_{h\to 0} \frac{1}{h}(f(x+h) - f(x)).$$

Equivalently, the **backward difference derivative** of f is defined by

$$f'(x) = \lim_{h\to 0} \frac{1}{h}(f(x) - f(x-h)).$$

Definition 1.2. Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function of x. The **central difference derivative** of f is defined by the limit

$$f'(x) = \lim_{h\to 0} \frac{1}{2h}(f(x+h) - f(x-h)).$$

*This paper was written as a part of the Directed Research Program at The University of Texas at Dallas. We thank the mathematics department for providing space for us to work and Joseph Burnett for organizing the program.

One can check that the forward and backward difference derivatives coincide whenever f is differentiable. Using this fact, it is easy to see that f is differentiable if and only if the central difference derivative exists and is equal to the forward difference derivative. Beginning with the forward difference quotient,

$$\begin{aligned}
f'(x) &= \lim_{h\to 0} \frac{1}{h}(f(x+h) - f(x)) \\
&= \lim_{h\to 0} \frac{1}{h}(f(x+h) + f(x-h) - f(x-h) - f(x)) \\
&= \lim_{h\to 0} \frac{1}{h}(f(x+h) - f(x-h)) - \lim_{h\to 0} \frac{1}{h}(f(x) - f(x-h)) \\
&= \lim_{h\to 0} \frac{1}{h}(f(x+h) - f(x-h)) - f'(x).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
2f'(x) &= \lim_{h\to 0} \frac{1}{h}(f(x+h) - f(x-h)) \\
f'(x) &= \lim_{h\to 0} \frac{1}{2h}(f(x+h) - f(x-h))
\end{aligned}$$

These definitions can be immediately extended to functions of two or more variables.

Definition 1.3. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a function of two variables x and y. The standard **forward difference partial derivative** of f with respect to x is defined by

$$\frac{\partial f}{\partial x} = \lim_{h\to 0} \frac{1}{h}(f(x+h, y) - f(x, y)),$$

and similarly the partial derivative of f with respect to y is

$$\frac{\partial f}{\partial y} = \lim_{h\to 0} \frac{1}{h}(f(x, y+h) - f(x, y)).$$

We also define the backward and central difference partial derivatives analogously to the above definitions for functions of a single variable.

In the next few sections we will also use the following notation to denote the derivative of u with respect to x at the point x_i,

$$u'(x_i) = \left.\frac{du}{dx}\right|_{x_i},$$

and we will denote the partial derivative of u with respect to x at the point (x_i, t_j) by

$$\frac{\partial u}{\partial x}(x_i, t_j) = \left.\frac{\partial u}{\partial x}\right|_{x_i, t_j}.$$

1.2 Partial Differential Equations

A **partial differential equation (PDE)** is an equation relating a function of several independent variables and its partial derivatives. An **analytical solution** to a PDE is a function that satisfies the PDE at every point in the domain of the equation. A **numerical solution** to a PDE is a function that *approximately* satisfies the PDE at a discrete set of points in the domain of the equation. In this paper we will be concerned with the **heat equation**

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2},$$

where we restrict the domain of the function u to the region satisfying

$$0 < x < l, \quad \text{and} \quad t > 0.$$

This is the heat equation with a single spatial dimension. It describes the flow of heat across a one-dimensional object with length l. The function u measures the temperature in the object, and the constant α is the thermal diffusivity constant that depends upon the material of the object, which is squared by convention.

2 Finite Differences

In many real world problems, partial differential equations are not solvable analytically. In such cases we can still approximate a solution to the equation using numerical methods. In this section we derive finite difference schemes, the Crank-Nicolson algorithm, and a fully implicit algorithm to obtain approximate solutions to partial differential equations. In the next section we use these techniques to obtain solutions to the heat equation and then compare them to analytical solutions.

2.1 Forward Difference

We will approximate the solution $u(x,t)$ to the PDE by replacing the partial derivatives with difference quotients, discretizing the domain of u, and solving the resulting set of algebraic equations. The end result will be a discrete mesh of approximate values of u. We write $u_{i,j}$ to denote $u(x_i, t_j)$, the solution evaluated at the mesh points x_i and t_j, which is the approximate numerical solution obtained by solving the finite difference equations. We denote the spatial step size or increment by h, and the temporal step size or increment by k. We will begin by fixing t so that u is a function of only one independent variable x. Then from the definition of the derivative,

$$\begin{aligned} \left.\frac{du}{dx}\right|_{x_i} &\approx \frac{1}{h}(u(x_i+h) - u(x_i)), \\ &= \frac{1}{h}(u_{i+1} - u_i), \end{aligned}$$

and for the second derivative we have,

$$\begin{aligned} \left.\frac{d^2u}{dx^2}\right|_{x_i} &= \lim_{h\to 0} \frac{1}{h}\left(\left.\frac{du}{dx}\right|_{x_i+h} - \left.\frac{du}{dx}\right|_{x_i} \right) \\ &\approx \frac{1}{h}\left(\frac{1}{h}(u_{i+2} - u_{i+1}) - \frac{1}{h}(u_{i+1} - u_i) \right) \\ &= \frac{1}{h^2}\left(u_{i+2} - 2u_{i+1} - u_i\right). \end{aligned}$$

Now allowing t to range over all time steps,

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x_i, t_j} &\approx \frac{1}{h}(u(x_i+h, t_j) - u(x_i, t_j)) \\ &= \frac{1}{h}(u_{i+1,j} - u_{i,j}). \end{aligned}$$

Similarly, the second order partial derivative is

$$\begin{aligned}
\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i,t_j} &= \lim_{h\to 0}\frac{1}{h}\left(\left.\frac{\partial u}{\partial x}\right|_{x_{i+h},t_j} - \left.\frac{\partial u}{\partial x}\right|_{x_i,t_j}\right) \\
&\approx \frac{1}{h}\left(\frac{1}{h}(u_{i+2,j}-u_{i+1,j}) - \frac{1}{h}(u_{i+1,j}-u_{i,j})\right) \\
&= \frac{1}{h^2}(u_{i+2,j}-2u_{i+1,j}+u_{i,j}). \\
&= \frac{1}{h^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j}).
\end{aligned}$$

This is a *forward difference* scheme because at each iteration we use the next spatial step to determine approximate values of the solution u.

2.2 Backward Difference

Instead of approximating the derivatives of u using the nodes x_i and x_{i+1} for a forward difference scheme, we can use the nodes x_i and x_{i-1} for a backward difference scheme. Analogously to the previous process, we have

$$\begin{aligned}
\left.\frac{du}{dx}\right|_{x_i} &= \lim_{h\to 0}\frac{1}{h}(u(x_i)-u(x_i-h)) \\
&\approx \frac{1}{h}(u(x_i)-u(x_i-h)) \\
&= \frac{1}{h}(u_i-u_{i-1}).
\end{aligned}$$

And therefore

$$\begin{aligned}
\left.\frac{\partial u}{\partial x}\right|_{x_i,t_j} &\approx \frac{1}{h}(u(x_i,t_j)-u(x_i-h,t_j)) \\
&= \frac{1}{h}(u_{i,j}-u_{i-1,j}).
\end{aligned}$$

2.3 Central Difference

Instead of approximating the derivatives of u using the nodes x_i and x_{i+1} as we did in the forward difference method, we can use the nodes x_{i-1} and x_{i+1}. As before, we begin by fixing t and approximating the univariate derivative of u with

$$\begin{aligned}
\left.\frac{du}{dx}\right|_{x_i} &= \lim_{h\to 0}\frac{1}{2h}(u(x_i+h)-u(x_i-h)) \\
&\approx \frac{1}{2h}(u(x_i+h)-u(x_i-h)) \\
&= \frac{1}{2h}(u_{i+1}-u_{i-1}).
\end{aligned}$$

Now allowing t to vary, we have

$$\begin{aligned}
\left.\frac{\partial u}{\partial x}\right|_{x_i,t_j} &\approx \frac{1}{2h}(u(x_i+h,t_j)-u(x_i-h,t_j)) \\
&= \frac{1}{2h}(u_{i+1,j}-u_{i-1,j}).
\end{aligned}$$

3 The Heat Equation

In this section we derive an analytical solution to the heat equation using the method of separation of variables, and then obtain approximate solutions using the previously derived numerical methods. We then compare these solutions and analyze the error.

3.1 Analytical Solution

We will derive a solution to the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad \text{where} \quad 0 < x < l, \quad t > 0, \tag{1}$$

subject to the boundary conditions

$$u(0,t) = 0 \quad \text{and} \quad u(1,t) = 0,$$

and the initial condition

$$u(x,0) = \sin(\pi x).$$

We begin by assuming that a solution is of the form

$$u(x,t) = X(x)T(t),$$

for some functions X and T. Substituting these into (1) yields

$$X(x)T'(t) = X''(x)T(t),$$

and after dividing by X(x)T(t) we have

$$\frac{X''}{X} = \frac{T'}{T}.$$

Now since the left-hand side is independent of t and the right-hand side is independent of x, they must both be constant. Thus, we write

$$\frac{X''}{X} = \frac{T'}{T} = -k^2,$$

for some constant k. We have separated our partial differential equation into two ordinary differential equations. The first equation is

$$X'' + k^2 X = 0,$$

which is a second-order, linear, homogeneous ODE that can be solved by making the substitution $X = e^{rx}$.

$$r^2 e^{rx} + k^2 e^{rx} = 0 \implies r^2 + k^2 = 0.$$

Hence, we have $r = \pm ik$ and therefore

$$X = c_1 e^{ikx} + c_2 e^{-ikx} = c_1 \cos(kx) + c_2 \sin(kx).$$

Now the second equation

$$T' + k^2 T = 0,$$

is a first-order, linear, homogeneous ODE with solution

$$T = c_3 e^{-k^2 t}.$$

Combining these two separated solutions, we have

$$u(x,t) = c_3 e^{-k^2 t}(c_1 \cos(kx) + c_2 \sin(kx)).$$

Now the boundary and initial conditions give us the equations

(i) $u(0,t) = c_3 e^{-k^2 t}(c_1) = 0$

(ii) $u(1,t) = c_3 e^{-k^2 t}(c_1 \cos(k) + c_2 \sin(k)) = 0$

(iii) $u(x,0) = c_3(c_1 \cos(kx) + c_2 \sin(kx) = \sin(\pi x)$

Equation (i) requires either $c_3 = 0$ or $c_1 = 0$. Since $c_3 = 0$ is a trivial solution, we let $c_1 = 0$. Then equation (ii) yields

$$c_3 e^{-k^2 t}(c_1 \sin(k)) = 0 \implies c_1 \sin(k) = 0,$$

and again $c_3 = 0$ would give us a trivial solution so we have $\sin(k) = 0$. Thus, choosing $k = \pi$ and $c_3 = c_2 = 1$ yields the particular solution

$$u(x,t) = e^{-\pi^2 t} \cos(\pi x).$$

3.2 Forward Time Centered Space Scheme

We would like to obtain an approximate solution to the heat equation from above.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad \text{for} \quad 0 < x < 1 \quad \text{and} \quad t > 0, \tag{2}$$

given the boundary conditions

$$u(0,t) = 0, \; u(1,t) = 0, \quad \text{and} \quad u(x,0) = \sin(\pi x).$$

We will use the forward difference approximation for the temporal and spatial derivatives. This is called the forward time centered space (FTCS) scheme. Substituting the difference quotients in for the partial derivatives yields

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}),$$

and therefore

$$\begin{aligned} u_{i,j+1} &= u_{i,j} + \frac{k}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}). \\ &= a u_{i+1,j} + (1-2a)u_{i,j} + a u_{i-1,j}, \end{aligned}$$

where $a = \frac{k}{h^2}$. Using this formula with $k = .02$ and $h = .2$, we have for the node $(i,j) = (1,0)$,

$$\begin{aligned} u_{1,1} &= \frac{1}{2}u_{2,0} + \left(1 - 2 \cdot \frac{1}{2}\right) u_{1,0} + \frac{1}{2}u_{0,0} \\ &= \frac{1}{2}u(.04,0) + \frac{1}{2}u(0,0) \\ &= \frac{1}{2}\sin(\pi(.4)) + 0 \\ &\approx .4755. \end{aligned}$$

We can follow this procedure with $i = 1, 2, ...$ and $j = 0, 1, 2, ...$ to obtain the following approximate values of u at each mesh point (i, j).

(i,j)	$u_{i,j}$	Formula	Value
$(1,0)$	$u_{1,0}$	$\frac{1}{2}u_{2,0} + \frac{1}{2}u_{0,0}$	.4755
$(2,0)$	$u_{2,0}$	$\frac{1}{2}u_{3,0} + \frac{1}{2}u_{1,0}$	.7694
$(3,0)$	$u_{3,0}$	$\frac{1}{2}u_{4,0} + \frac{1}{2}u_{2,0}$	.7694
$(4,0)$	$u_{4,0}$	$\frac{1}{2}u_{5,0} + \frac{1}{2}u_{3,0}$	.4755
$(1,1)$	$u_{1,1}$	$\frac{1}{2}u_{2,1} + \frac{1}{2}u_{0,1}$	.7694
$(2,1)$	$u_{2,1}$	$\frac{1}{2}u_{3,1} + \frac{1}{2}u_{1,1}$	1.245
$(3,1)$	$u_{3,1}$	$\frac{1}{2}u_{4,1} + \frac{1}{2}u_{2,1}$	1.245

A benefit of the finite difference method is that it is very easy to implement in a programming language such as MATLAB. The figure below gives a plot of the two solutions using two time steps and constant spatial and temporal increments.

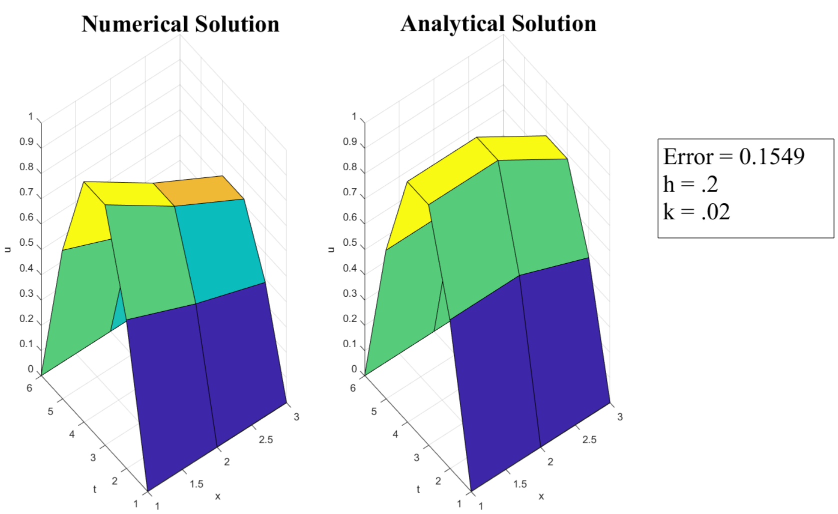


Figure 1: Analytical and FTCS Solution for Heat Equation

The figure below is a plot of the two solutions using smaller increments of $h = .05$ and $k = .005$.

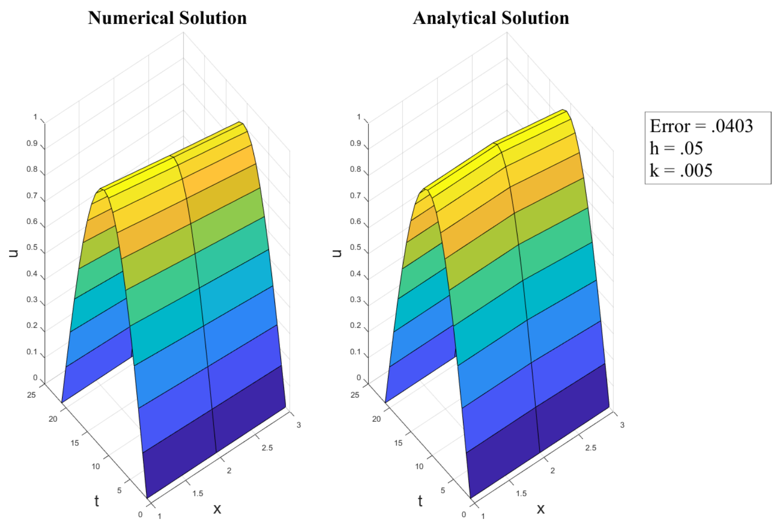


Figure 2: Analytical and FTCS Solution for Heat Equation with smaller steps

As h and k decrease we obtain better approximations. We will show this theoretically in the next section.

3.3 Truncation Error of FTCS Scheme

We would like to have a way of measuring the error between the analytical solution and the numerical solution obtained from the finite difference approximation. Intuitively, this error should decrease as h and k get smaller.

We start with the FTCS difference equation

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}),$$

i.e.,

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) - \frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = 0.$$

Let U denote the exact analytical solution of the heat equation, and $U_{i,j} = U(x_i, t_j)$ the value of the analytical solution at step (i, j). Then we define the **truncation error** at (i, j) by

$$T_{i,j} = \frac{1}{k}(U_{i,j+1} - U_{i,j}) - \frac{1}{h^2}(U_{i+1,j} - 2U_{i,j} + U_{i-1,j}). \tag{3}$$

The Taylor series expansion of $U_{i,j}$ with respect to t about the zero is

$$U_{i,j+1} = U_{i,j} + k\frac{\partial U}{\partial t} + \frac{k^2}{2!}\frac{\partial^2 U}{\partial t^2} + \frac{k^3}{3!}\frac{\partial^3 U}{\partial t^3} + \cdots$$

where each derivative is evaluated at (x_i, t_j). Similarly,

$$U_{i+1,j} = U_{i,j} + h\frac{\partial U}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 U}{\partial x^2} + \frac{h^3}{3!}\frac{\partial^3 U}{\partial x^3} + \cdots,$$

where each derivative is evaluated at (x_i, t_j), and

$$U_{i-1,j} = U_{i,j} - h\frac{\partial U}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 U}{\partial x^2} - \frac{h^3}{3!}\frac{\partial^3 U}{\partial x^3} + \cdots ,$$

now plugging each of these into (3), we obtain

$$T_{i,j} = \left(\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2}\right) + \left(\frac{k}{2}\frac{\partial^2 U}{\partial t^2} + \frac{k^2}{3!}\frac{\partial^3 U}{\partial t^3} + \cdots\right) + \left(\frac{1}{12}\frac{\partial^4 U}{\partial x^4} + \frac{h^2}{360}\frac{\partial^6 U}{\partial x^6} + \cdots\right)$$

We choose the lowest powers of h and k to determine the order of the truncation error $\mathcal{O}(T_{i,j})$. Therefore, the order of $T_{i,j}$ is

$$\mathcal{O}\left(T_{i,j}\right) = \mathcal{O}(1) + \mathcal{O}(k) + \mathcal{O}(h^2),$$

and k is itself of order h^2 because by definition $k = ah^2$, so

$$\mathcal{O}\left(T_{i,j}\right) = \mathcal{O}(h^2).$$

We will now verify this graphically by plotting the solution of the PDE using decreasing increments and then plotting the error. The following plot compares the analytical to the numerical solution using decreasing spatial increments starting at $h = .5$.

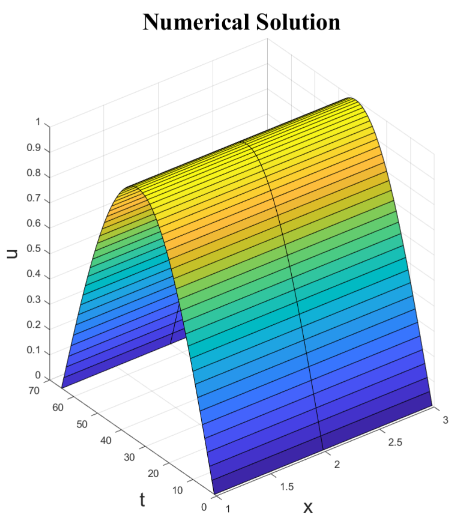


Figure 3: Analytical and FTCS Solutions of the heat equation with decreasing increments

Now since the truncation error is of order h^2, we write

$$\begin{aligned} T_{i,j} &= ch^2 \quad \text{for some constant } c \\ \log(T_{i,j}) &= \log(ch^2) \\ &= \log c + 2\log h, \end{aligned}$$

hence, when we plot the log of the error we obtain a line with slope 2.

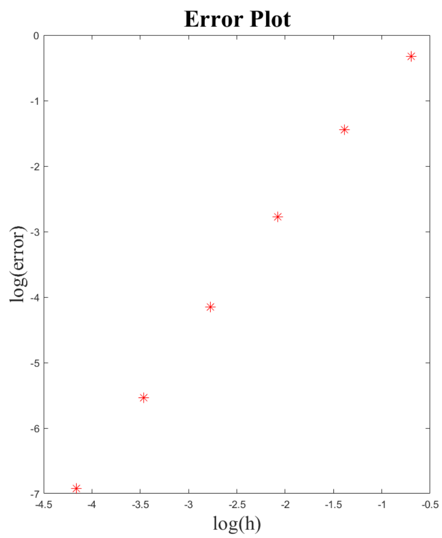


Figure 4: Error plot of the FTCS solution of the heat equation

3.4 Error and Stability of FTCS Scheme

Consider the general heat equation

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{where} \quad 0 < x < l, \quad t > 0, \tag{4}$$

given boundary conditions as in Section 1.2. We define the **discretization error** of the finite difference scheme as the difference between the exact analytical solution to the PDE and the exact solution to the finite difference equations used to approximate the PDE. We say that a numerical solution to a PDE is **stable** if the error remains bounded as the number of steps increases. We will determine a condition for which the approximate solution to (4) obtained by the FTCS scheme is stable. Recall the FTCS algorithm

$$u_{i,j+1} = au_{i-1,j} + (1-2a)u_{i,j} + au_{i+1,j}.$$

Substituting $i = 1, 2, \ldots, n-1$ into the algorithm yields the following system of equations:

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ \vdots \\ u_{n-1,j+1} \end{bmatrix} = \begin{bmatrix} 1-2a & a & 0 & \cdots & 0 \\ a & 1-2a & a & & \vdots \\ 0 & a & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & a \\ 0 & \cdots & 0 & a & 1-2a \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ \vdots \\ u_{n-1,j} \end{bmatrix},$$

which we will rewrite as

$$U_{j+1} = AU_j, \tag{5}$$

where U_j denotes the vector consisting of approximate values of the solution to (4) at time step j and spatial steps $i = 1, \ldots, n-1$. Now substituting $j = 0$ and $j = 1$ into (5), we have

$$U_1 = AU_0 \quad \text{and} \quad U_2 = AU_1 = A^2U_0,$$

so in general for the mth time step $j = m$,

$$U_m = A^m U_0,$$

where

$$U_0 = \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ \vdots \\ u_{n-1,0} \end{bmatrix}$$

is the vector of initial values. Let U_j^* denote the vector consisting of values of the exact, analytical solution to (4) at time step j and spatial steps $i = 1, \ldots, n-1$. Then analogously to equation (5) we have

$$U_m^* = A^m U_0^*.$$

Then the error vector at the initial time step $j = 0$ is

$$e_0 = U_0^* - U_0,$$

and at the mth time step,

$$\begin{aligned} e_m &= U_m^* - U_m \\ &= A^m U_0^* - A^m U_0 \\ &= A^m(U_0^* - U_0) \\ &= A^m e_0. \end{aligned}$$

Notice that the matrix A above is a tridiagonal matrix with constant diagonals. Such matrices are called **Toeplitz**, and there is a closed form solution for their eigenvalues (see [3]). The eigenvalues of A are

$$\begin{aligned} \lambda_k &= (1-2a) + 2\sqrt{a^2}\cos\left(\frac{k\pi}{(n-1)+1}\right) \\ &= 1 - 2a + 2a\cos\left(\frac{k\pi}{n}\right) \\ &= 1 - 2a\left(1 - \cos\left(\frac{k\pi}{n}\right)\right) \\ &= 1 - 4a\sin^2\left(\frac{k\pi}{2n}\right), \end{aligned}$$

for $k = 1, 2, \ldots, n-1$. Since each eigenvalue is distinct, the corresponding eigenvectors are linearly independent and therefore span $\mathbb{R}^{n-1}$. Since $e_0 \in \mathbb{R}^{n-1}$, we can write

$$e_0 = \sum_{r=1}^{n-1} c_r v_r,$$

where each v_r is the eigenvector of A corresponding to λ_r and each c_r is a scalar. Now using this in the equation for e_m above, we have

$$\begin{aligned} e_m &= A^m e_0 \\ &= A^m \sum_{r=1}^{n-1} c_r v_r \\ &= \sum_{r=1}^{n-1} c_r A^m v_r \\ &= \sum_{r=1}^{n-1} c_r \lambda_r^m v_r \end{aligned}$$

and therefore

$$\begin{aligned} \|e_m\| &= \left\| \sum_{r=1}^{n-1} c_r \lambda_r^m v_r \right\| \\ &\leq \sum_{r=1}^{n-1} \|c_r \lambda_r^m v_r\| \\ &= \sum_{r=1}^{n-1} |c_r| |\lambda_r|^m \|v_r\| . \end{aligned}$$

Hence, from the above inequality we see that the error will remain bounded if $|\lambda_r| \leq 1$ for each r, that is, if

$$\begin{aligned} -1 &\leq \lambda_r \leq 1 \\ -1 &\leq 1 - 4a\sin^2\left(\frac{r\pi}{2n}\right) \leq 1 \\ -2 &\leq -4a\sin^2\left(\frac{r\pi}{2n}\right) \leq 0 \\ 0 &\leq a \leq \frac{1}{2\sin^2\left(\frac{r\pi}{2n}\right)}. \end{aligned}$$

Since a is positive by definition, and $\sin^2 x \leq 1$, the requirement for stability is

$$a \leq \frac{1}{2} \implies k \leq \frac{h^2}{2}.$$

So we can always ensure the stability of our solution by choosing a sufficiently small time step.

3.5 Crank-Nicolson Algorithm

Although computationally efficient, one drawback of the FTCS algorithm is that it is valid for only small values of k. The Crank-Nicolson algorithm on the other hand is valid for all values of h and k. We will employ this algorithm to solve the following heat equation

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{where} \quad 0 < x < 1, t > 0, \tag{6}$$

given the boundary conditions

$$u(0,t) = 0, \quad \text{and} \quad u(1,t) = 0,$$

and the initial condition

$$u(x, 0) = \sin(\pi x).$$

The Crank-Nicolson algorithm is a finite differences scheme that approximates the left-hand side of (6) with the backward time difference and the right-hand side of (6) with the average of the central difference evaluated at the current and previous time step. Thus, (6) becomes

$$\frac{1}{k}\left(u_i^j - u_i^{j-1}\right) = \frac{1}{2h^2}\left[\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right) + \left(u_{i+1}^{j-1} - 2u_i^{j-1} + u_{i-1}^{j-1}\right)\right],$$

and rearranging so that values of u at time step j are on the left and values at time step $j-1$ are on the right, we have

$$-au_{i+1,j} + 2u_{i,j} + 2au_{i,j} - au_{i-1,j} = au_{i+1,j-1} + 2u_{i,j-1} - 2au_{i,j-1} + au_{i-1,j-1}$$

where $a = \frac{k}{h^2}$. After collecting terms, the equation becomes

$$-au_{i+1,j} + 2(1+a)u_{i,j} - au_{i-1,j} = au_{i+1,j-1} + 2(1-a)u_{i,j-1} + au_{i-1,j-1}.$$

Finally, after shifting the time index we have the Crank-Nicolson algorithm

$$-au_{i+1,j+1} + 2(1+a)u_{i,j+1} - au_{i-1,j+1} = au_{i+1,j} + 2(1-a)u_{i,j} + au_{i-1,j}. \tag{7}$$

Evidently, at each time step we will need to solve a system of equations for the values of u at the subsequent time step. For example, let $h = .25$ be our spatial step and $k = .03125$ be our temporal step. Then $a = \frac{1}{2}$, so (7) becomes

$$-\frac{1}{2}u_{i+1,j+1} + 3u_{i,j+1} - \frac{1}{2}u_{i-1,j+1} = \frac{1}{2}u_{i+1,j} + u_{i,j} + \frac{1}{2}u_{i-1,j}. \tag{8}$$

The first four values of u at the initial time $t = 0$ are given by:

1. $u_{0,0} = u(0, 0) = \sin(\pi \cdot 0) = 0.$
2. $u_{1,0} = u(.25, 0) = \sin(.25(\pi)) \approx .7071.$
3. $u_{2,0} = u(.5, 0) = \sin(.5(\pi)) = 1.$
4. $u_{3,0} = u(.75, 0) = \sin(.75(\pi)) \approx .7071.$

Now after applying the first time step $j = 0$ to (8), we have

$$-\frac{1}{2}u_{i+1,1} + 3u_{i,1} - \frac{1}{2}u_{i-1,1} = \frac{1}{2}u_{i+1,0} + u_{i,0} + \frac{1}{2}u_{i-1,0}. \tag{9}$$

Substituting $i = 1$ into (8) gives us the equation

$$-\frac{1}{2}u_{2,1} + 3u_{1,1} - \frac{1}{2}u_{0,1} = \frac{1}{2}u_{2,0} + u_{1,0} + \frac{1}{2}u_{0,0},$$

which simplifies to

$$-\frac{1}{2}u_{2,1} + 3u_{1,1} \approx 1.207.$$

Similarly, the analogous equations for $i = 2$ and $i = 3$ are given in the table below.
Thus, the three equations can be combined into the matrix equation

$$\begin{bmatrix} 3 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 3 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 3 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} \approx \begin{bmatrix} 1.207 \\ 1.707 \\ 1.207 \end{bmatrix},$$

(i,j)	Equation
$(1,0)$	$-\frac{1}{2}u_{2,1} + 3u_{1,1} \approx 1.207$
$(2,0)$	$-\frac{1}{2}u_{3,1} + 3u_{2,1} - \frac{1}{2}u_{1,1} \approx 1.707$
$(3,0)$	$3u_{3,1} - \frac{1}{2}u_{2,1} \approx 1.207$

and solving this equation yields the approximate solution

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} \approx \begin{bmatrix} 0.5265 \\ 0.7445 \\ 0.5265 \end{bmatrix}.$$

Repeating this process at each node, we obtain a mesh of approximate values of u. Implementing the algorithm in MATLAB and plotting the analytical solution for comparison gives us the following two plots.

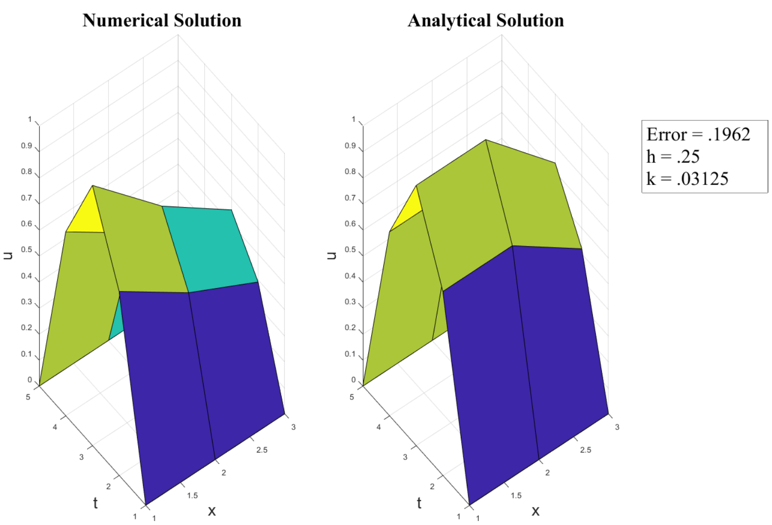


Figure 5: Analytical and Crank-Nicolson Solution for Heat Equation

Now plotting the two solutions using smaller increments:

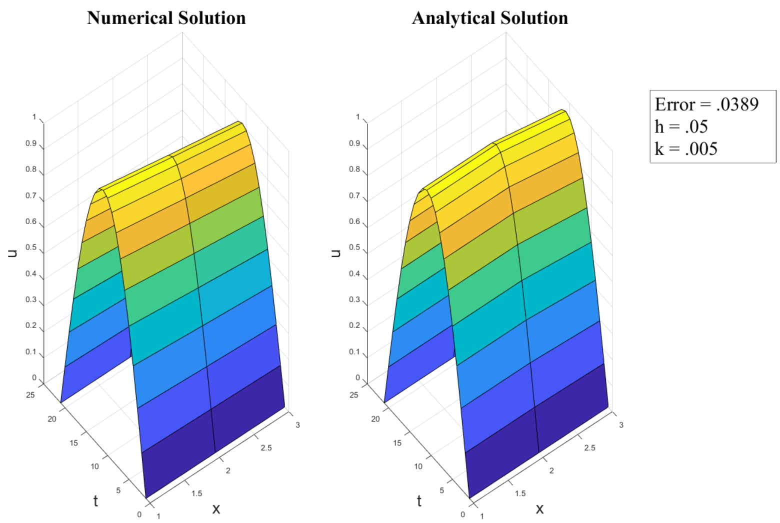
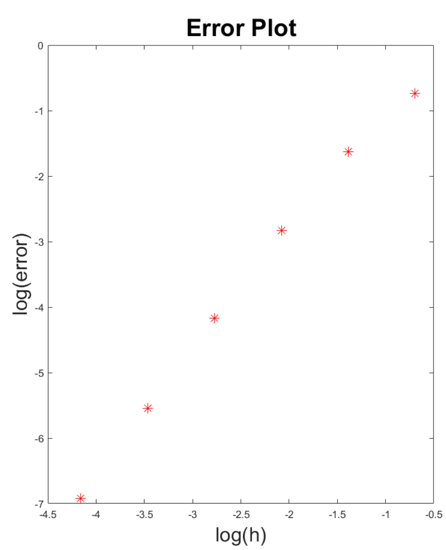


Figure 6: Analytical and Crank-Nicolson Solution for Heat Equation with smaller steps



3.6 Error and Stability of Crank-Nicolson Algorithm

We motivated the Crank-Nicolson algorithm above by suggesting that it is valid for any step size, unlike the FTCS scheme. We will now demonstrate this the same way we showed that FTCS is valid for $a \leq \frac{1}{2}$. Inserting $i = 1, 2, \ldots, n-1$ into the Crank-Nicolson algorithm (7), we obtain the system of equations

$$\begin{bmatrix} 2(a+1) & -a & 0 & \cdots & 0 \\ -a & 2(a+1) & -a & & \vdots \\ 0 & -a & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -a \\ 0 & \cdots & 0 & -a & 2(a+1) \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{n-1,j+1} \end{bmatrix} = \begin{bmatrix} 2(1-a) & a & 0 & \cdots & 0 \\ a & 2(1-a) & a & & \vdots \\ 0 & a & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & a \\ 0 & \cdots & 0 & a & 2(1-a) \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots u_{n-1,j} \end{bmatrix}$$

which we will rewrite as

$$BU_{j+1} = CU_j.$$

Let $A = B^{-1}C$. Then the equation becomes

$$U_{j+1} = AU_j,$$

and as in Section 3.3,

$$U_m = A^m U_0,$$

where m denotes the mth time step and U_0 is the vector of initial values. Letting U_j^* again denote the vector consisting of values of the analytical solution to the heat equation at time step j, the error at the mth time step is given by

$$\begin{aligned} e_m &= U_m^* - U_m \\ &= A^m U_0^* - A^m U_0 \\ &= A^m (U_0^* - U_0) \\ &= A^m e_0. \end{aligned}$$

Since matrices B and C are both Toeplitz, their eigenvalues are, respectively,

$$\begin{aligned} \lambda_k^B &= 2(a+1) + 2\sqrt{a^2}\cos\left(\frac{k\pi}{(n-1)+1}\right) \\ &= 2a + 2 + 2a\cos\left(\frac{k\pi}{n}\right) \\ &= 2 + 2a\left(1 + \cos\left(\frac{k\pi}{n}\right)\right) \\ &= 2 + 4a\sin^2\left(\frac{k\pi}{2n}\right) \end{aligned}$$

$$\begin{aligned} \lambda_k^C &= 2(1-a) + 2\sqrt{a^2}\cos\left(\frac{k\pi}{n}\right) \\ &= 2 - 2a + 2a\cos\left(\frac{k\pi}{n}\right) \\ &= 2 - 2a\left(1 - \cos\left(\frac{k\pi}{n}\right)\right) \\ &= 2 - 4a\sin^2\left(\frac{k\pi}{2n}\right) \end{aligned}$$

Therefore, the eigenvalues of A are

$$\lambda_k = \frac{2 - 4a\sin^2\left(\frac{k\pi}{2n}\right)}{2 + 4a\sin^2\left(\frac{k\pi}{2n}\right)}, \quad k = 1, 2, \ldots, n-1.$$

Since these eigenvalues are all distinct, the corresponding $n-1$ eigenvectors $v_1, \ldots, v_{n-1}$ are linearly independent and therefore span $\mathbb{R}^{n-1}$. Hence, we have

$$e_0 = \sum_{r=1}^{n-1} c_r v_r,$$

and

$$\begin{aligned} e_m &= A^m e_0 \\ &= A^m \sum_{r=1}^{n-1} c_r v_r \\ &= \sum_{r=1}^{n-1} c_r A^m v_r \\ &= \sum_{r=1}^{n-1} c_r \lambda_r^m v_r, \end{aligned}$$

which gives us

$$\|e_m\| = \sum_{r=1}^{n-1} |c_r||\lambda_r|^m \|v_r\|,$$

so for $\|e_m\|$ to remain bounded, we must have $|\lambda_r| \leq 1$. Therefore,

$$\begin{aligned} -1 &\leq \lambda_r \leq 1 \\ -1 &\leq \frac{2 - 4a\sin^2\left(\frac{r\pi}{2n}\right)}{2 + 4a\sin^2\left(\frac{r\pi}{2n}\right)} \leq 1 \\ -1 - 2a\sin^2\left(\frac{r\pi}{2n}\right) &\leq 1 - 2a\sin^2\left(\frac{r\pi}{2n}\right) \leq 1 + 2a\sin^2\left(\frac{r\pi}{2n}\right) \\ -1 &\leq 1 \leq 1 + 4a\sin^2\left(\frac{r\pi}{2n}\right) \end{aligned}$$

so we have

$$0 \leq 4a\sin^2\left(\frac{r\pi}{2n}\right),$$

which is always true because $a > 0$. Therefore, the Crank-Nicolson algorithm is stable for any combination of step sizes.

3.7 Truncation Error of Crank-Nicolson Algorithm

We will now quantify the truncation error obtained when solving the finite difference scheme (7). In this section, we use the matrix exponential of the differential operator D, that is, the linear map defined by

$$D^k f(x) = f^{(k)}(x).$$

From Taylor's theorem, we have

$$\begin{aligned}
f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x)\cdots \\
&= f(x) + hDf(x) + \frac{h^2}{2!}D^2f(x) + \frac{h^3}{3!}D^3f(x)\cdots \\
&= \left(1 + hD + \frac{1}{2!}(hD)^2 + \frac{1}{3!}(hD)^3\right)f(x) \\
&= e^{hD}f(x)
\end{aligned}$$

and similarly,

$$\begin{aligned}
f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \cdots \\
&= f(x) + hDf(x) + \frac{h^2}{2!}D^2f(x) + \cdots \\
&= \left(1 - hD + \frac{1}{2!}(hD)^2 - \frac{1}{3!}(hD)^3\right)f(x) \\
&= e^{-hD}f(x)
\end{aligned}$$

Now we apply these matrix exponentials to the Crank-Nicolson algorithm (7), where on the left hand side we obtain

$$\begin{aligned}
\frac{1}{k}(u_{i,j+1} - u_{i,j}) &= \frac{1}{k}(u(x_i, t_{j+1}) - u(x_i, t_j)) \\
&= \frac{1}{k}(u(x_i, t_j + k) - u(x_i, t_j)) \\
&= \frac{1}{k}\left(e^{kDt}u - u\right) \\
&= \frac{1}{k}\left(e^{kDt} - 1\right)u,
\end{aligned}$$

where $u = u(x_i, t_j)$. As for the right-hand side,

$$\begin{aligned}
u_{i+1,j} - 2u_{i,j} + u_{i-1,j} &= u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j) \\
&= e^{hDx_i}u - 2u + e^{-hDx_i}u \\
&= (e^{hDx_i} - 2te^{-hDx_i})u,
\end{aligned}$$

and

$$\begin{aligned}
u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} &= u(x_i + h, t_j + k) - 2u(x_i, t_j + k) + u(x_i - h, t_j + k) \\
&= \left(e^{hDx+kDt}u - 2e^{kDt}u + e^{-hDx+kDt}u\right) \\
&= \left(e^{hDx+kDt} - 2e^{kDt} + e^{-hDx+kDt}\right)u.
\end{aligned}$$

Now substituting these into the algorithm, we have

$$\begin{aligned}
\frac{1}{k}\left(e^{kDt} - 1\right)u &= \frac{1}{2h^2}\left(e^{hDx+kDt} - 2e^{kDt} + e^{-hDx+kDt} + e^{hDx} - 2 + e^{-hDx}\right)u \\
&= \frac{1}{2h^2}\left(e^{kDt}\left(e^{hDx} - 2 + e^{-hDx}\right) + \left(e^{hDx} - 2 + e^{-hDx}\right)\right)u \\
&= \frac{1}{2h^2}\left(e^{hDx} - 2 + e^{-hDx}\right)\left(e^{kDt} + 1\right)u
\end{aligned}$$

and therefore,

$$\left(\frac{1}{k}\left(e^{kDt}-1\right)-\frac{1}{2h^2}\left(e^{hDx}-2+e^{-hDx}\right)\left(e^{kDt}+1\right)\right)u=0,$$

which we will rewrite as

$$\mathcal{D}u=0.$$

Now we write the heat equation in terms of partial differential operators,

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}\implies D_t u=D_x^2 u,$$

which yields

$$\begin{aligned}(D_t-D_x^2)u &= 0\\ Lu &= 0.\end{aligned}$$

Hence, the truncation error of the Crank-Nicolson algorithm is given by

$$\begin{aligned}T_{i,j} &= (\mathcal{D}-L)u\\ &= \left(\frac{1}{k}\left(e^{kDt}-1\right)-\frac{1}{2h^2}\left(e^{hDx}-2+e^{-hDx}\right)\left(e^{kDt}+1\right)-\left(D_t-D_x^2\right)\right)u.\end{aligned}$$

This yields a truncation error on the order of

$$\mathcal{O}(T_{i,j})=\mathcal{O}(k^2)+\mathcal{O}(h^2).$$

We can verify this graphically by plotting the Crank-Nicolson solution against the analytical solution using decreasing increments. The error should again decrease as h and k decrease.

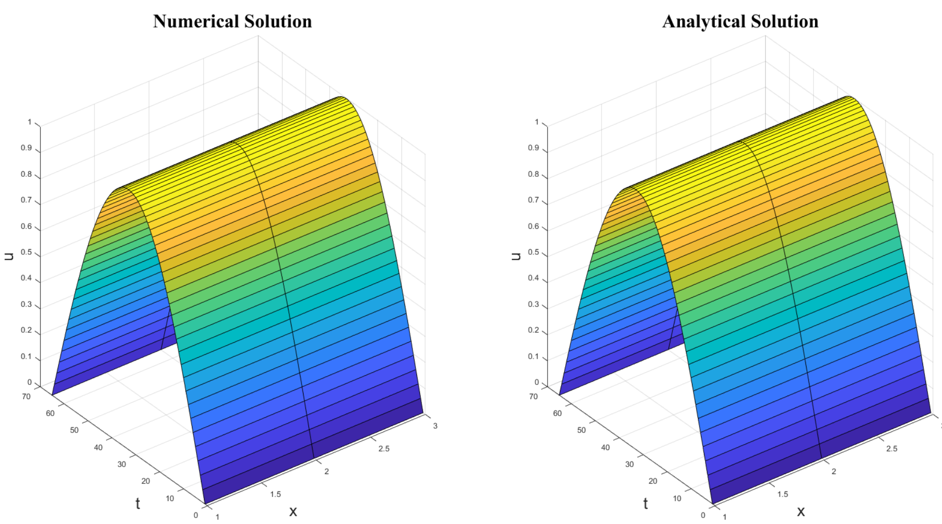


Figure 7: Analytical and Crank-Nicolson Solutions for Heat Equation with decreasing steps

Plotting the log of the error confirms our theoretical result, as it decreases linearly.

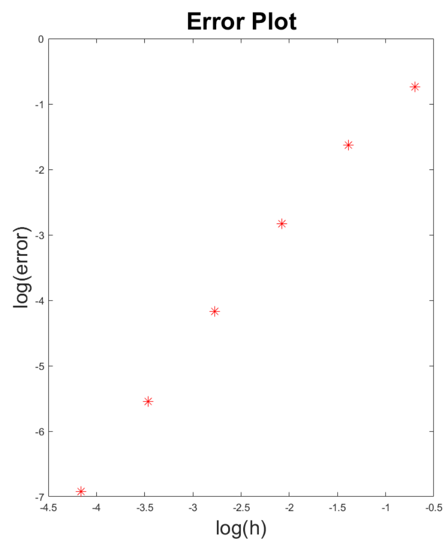


Figure 8: Error Plot of Crank-Nicolson Solution

3.8 Fully Implicit Algorithm

The next method we look at is a fully implicit algorithm that uses the backward difference at time $j+1$ to approximate the time derivative and the central difference to approximate the spatial derivative. Again, we solve the heat equation

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{where} \quad 0 < x < 1, t > 0, \tag{10}$$

given the boundary conditions

$$u(0,t) = 0, \quad \text{and} \quad u(1,t) = 0,$$

and the initial condition

$$u(x,0) = \sin(\pi x).$$

Substituting the aforementioned difference quotients into equation (10) yields

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{1}{h^2}(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}).$$

Now after solving for $u_{i,j}$ we have the fully implicit algorithm

$$u_{i,j} = -au_{i+1,j+1} + (2a+1)u_{i,j+1} - au_{i-1,j+1}, \tag{11}$$

where $a = \frac{k}{h^2}$. We will solve (11) using the spatial step $h = .25$ and time step $k = .0625$ so that $a = 1$. Then (11) becomes

$$u_{i,j} = -u_{i+1,j+1} + 3u_{i,j+1} - u_{i-1,j+1}. \tag{12}$$

From the boundary conditions, the first few values of u at the initial time step $j = 0$ are:

1. $u_{0,0} = u(0,0) = \sin(\pi \cdot 0) = 0.$
2. $u_{1,0} = u(.25,0) = \sin(.25(\pi)) \approx .7071.$
3. $u_{2,0} = u(.5,0) = \sin(.5(\pi)) = 1.$

4. $u_{3,0} = u(.75, 0) = \sin(.75(\pi)) = .7071.$

Now substituting $i = 1$ and $j = 0$ into (12) gives us the equation

$$u_{1,0} = -u_{2,1} + 3u_{1,1} - u_{0,1},$$

and after substituting $u_{1,0} \approx .7071$ and $u_{0,1} = 0$, we have

$$3u_{1,1} - u_{2,1} \approx .7071.$$

We follow this procedure for the subsequent nodes $i = 2, 3$ to obtain the equations:

(i, j)	Equation
$(1, 0)$	$3u_{1,1} - u_{2,1} \approx .7071$
$(2, 0)$	$-u_{3,1} + 3u_{2,1} - u_{1,1} = 1$
$(3, 0)$	$3u_{3,1} - u_{2,1} \approx .7071$

These equations can be written in matrix form as

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} \approx \begin{bmatrix} .7071 \\ 1 \\ .7071 \end{bmatrix}$$

Solving this equation yields the approximate solution

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} \approx \begin{bmatrix} .4459 \\ .6306 \\ .4459 \end{bmatrix}$$

Implementing this algorithm in MATLAB, we obtain a mesh of approximate values of u and the following plot.

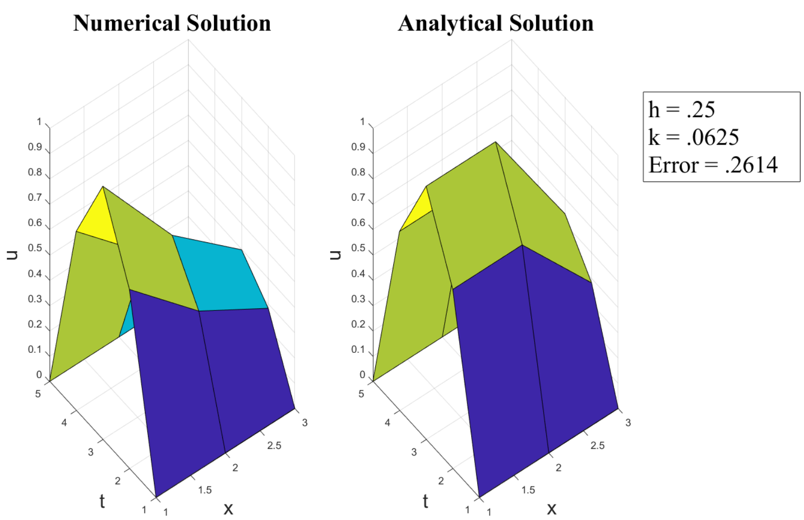


Figure 9: Analytical and Fully Implicit Solutions for Heat Equation

Using smaller increments of of $h = .05$ and $k = .005$, the fully implicit algorithm generates the following plot.

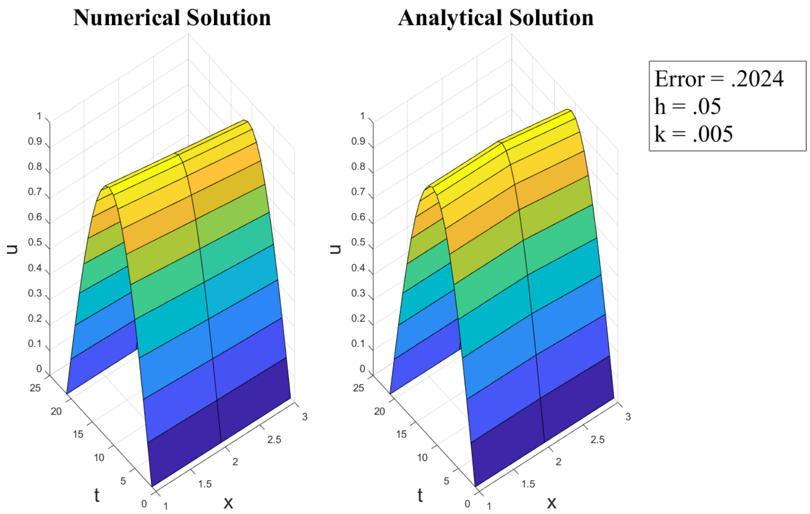
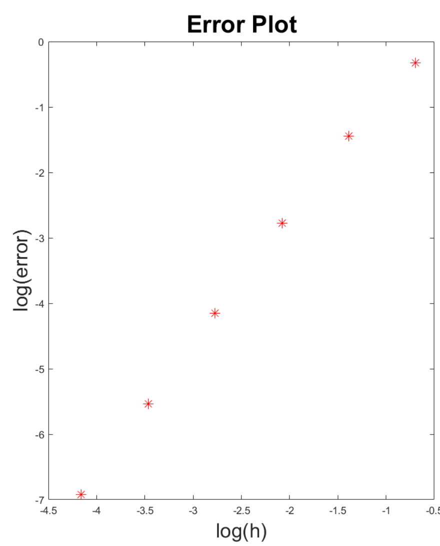


Figure 10: Analytical and Fully Implicit Solutions for Heat Equation with small steps



3.9 Error and Stability of Fully Implicit Algorithm

In this section we make use of complex numbers, so we will temporarily adopt the subscripts r and s for spatial and temporal nodes, respectively. The fully implicit scheme is given by

$$u_{r,s} = -au_{r+1,s+1} + (2a+1)u_{r,s+1} - au_{r-1,s+1}. \tag{13}$$

We make the substitution

$$u_{r,s} = e^{i\beta rh} z^s,$$

for some $\beta \in \mathbb{R}$ and $z \neq 0$. From this substitution, the method remains stable if $|z| \leq 1$. Now (12) becomes

$$e^{i\beta rh} z^s = -ae^{i\beta(r+1)h} z^{s+1} + (2a+1)e^{i\beta rh} z^{s+1} - ae^{i\beta(r-1)h} z^{s+1}.$$

After dividing out common exponents, this equation simplifies to

$$\begin{aligned} 1 &= -ae^{i\beta h} z + (2a+1)z - ae^{-i\beta h} z \\ &= -2az \left(\frac{e^{i\beta h} + e^{-i\beta h}}{2} \right) + (2a+1)z \\ &= -2az \cos(\beta h) - (2a+1)z \\ &= -4az \sin^2 \left(\frac{\beta h}{2} \right) - (2a+1)z, \end{aligned}$$

and after solving for z we have

$$z = \frac{1}{1 + 4a \sin^2 \left(\frac{\beta h}{2} \right)},$$

from which it is clear that we have $|z| \leq 1$ for all values of a. Hence, the method is stable for all values of a.

References

[1] Gerald W. Recktenwald, *Finite-Difference Approximations to the Heat Equation*, (2011).

[2] Walter A. Strauss. *Partial Differential Equations: an Introduction*. John Wiley, 2008.

[3] G.D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, 2008.