# 3
# Numerical Solutions of PDEs

*There's no sense in being precise when you don't even know what you're talking about.*- John von Neumann (1903-1957)

Most of the book has dealt with finding exact solutions to some generic problems. However, most problems of interest cannot be solved exactly. The heat, wave, and Laplace equations are linear partial differential equations and can be solved using separation of variables in geometries in which the Laplacian is separable. However, once we introduce nonlinearities, or complicated non-constant coefficients intro the equations, some of these methods do not work. Even when separation of variables or the method of eigenfunction expansions gave us exact results, the computation of the resulting series had to be done on a computer and inevitably one could only use a finite number of terms of the expansion. So, therefore, it is sometimes useful to be able to solve differential equations numerically. In this chapter we will introduce the idea of numerical solutions of partial differential equations. We will introduce finite difference method and the idea of stability. Other common approaches may be added later.

## 3.1 *The Finite Difference Method*

The heat equation can be solved using separation of variables. However, many partial differential equations cannot be solved exactly and one needs to turn to numerical solutions. The heat equation is a simple test case for using numerical methods. Here we will use the simplest method, finite differences.

Let us consider the heat equation in one dimension,

$$u_t = ku_{xx}.$$

Boundary conditions and an initial condition will be applied later. The starting point is figuring out how to approximate the derivatives in this equation.

Recall that the partial derivative, $u_t$, is defined by

$$\frac{\partial u}{\partial t} = \lim_{\Delta t \to \infty} \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}.$$

Therefore, we can use the approximation

$$\frac{\partial u}{\partial t} \approx \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}. \tag{3.1}$$

This is called a forward difference approximation.

In order to find an approximation to the second derivative, $u_{xx}$, we start with the forward difference

$$\frac{\partial u}{\partial x} \approx \frac{u(x+\Delta x, t) - u(x,t)}{\Delta x}.$$

Then,

$$\frac{\partial u_x}{\partial x} \approx \frac{u_x(x+\Delta x, t) - u_x(x,t)}{\Delta x}.$$

We need to approximate the terms in the numerator. It is customary to use a backward difference approximation. This is given by letting $\Delta x \to -\Delta x$ in the forward difference form,

$$\frac{\partial u}{\partial x} \approx \frac{u(x,t) - u(x-\Delta x, t)}{\Delta t}. \tag{3.2}$$

Applying this to $u_x$ evaluated at $x = x$ and $x = x + \Delta x$, we have

$$u_x(x,t) \approx \frac{u(x,t) - u(x-\Delta x, t)}{\Delta x},$$

and

$$u_x(x+\Delta x, t) \approx \frac{u(x+\Delta x, t) - u(x,t)}{\Delta x}.$$

Inserting these expressions into the approximation for $u_{xx}$, we have

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} &= \frac{\partial u_x}{\partial x} \\
&\approx \frac{u_x(x+\Delta x, t) - u_x(x,t)}{\Delta x} \\
&\approx \frac{\frac{u(x+\Delta x,t)-u(x,t)}{\Delta x}}{\Delta x} - \frac{\frac{u(x,t)-u(x-\Delta x,t)}{\Delta x}}{\Delta x} \\
&= \frac{u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t)}{(\Delta x)^2}.
\end{aligned} \tag{3.3}$$

This approximation for $u_{xx}$ is called the central difference approximation of $u_{xx}$.

Combining Equation (3.1) with (3.3) in the heat equation, we have

$$\frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} \approx k \frac{u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t)}{(\Delta x)^2}.$$

Solving for $u(x, t+\Delta t)$, we find

$$u(x, t+\Delta t) \approx u(x,t) + \alpha \left[ u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t) \right], \tag{3.4}$$

where $\alpha = k\frac{\Delta t}{(\Delta x)^2}$.

In this equation we have a way to determine the solution at position $x$ and time $t + \Delta t$ given that we know the solution at three positions, $x$, $x + \Delta x$, and $x + 2\Delta x$ at time $t$.

$$u(x, t + \Delta t) \approx u(x, t) + \alpha \left[u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)\right]. \qquad (3.5)$$

A shorthand notation is usually used to write out finite difference schemes. The domain of the solution is $x \in [a, b]$ and $t \geq 0$. We seek approximate values of $u(x, t)$ at specific positions and times. We first divide the interval $[a, b]$ into $N$ subintervals of width $\Delta x = (b - a)/N$. Then, the endpoints of the subintervals are given by

$$x_i = a + i\Delta x, \quad i = 0, 1, \ldots, N.$$

Similarly, we take time steps of $\Delta t$, at times

$$t_j = j\Delta t, \quad j = 0, 1, 2, \ldots.$$

This gives a grid of points $(x_i, t_j)$ in the domain.

At each grid point in the domain we seek an approximate solution to the heat equation, $u_{i,j} \approx u(x_i, t_j)$. Equation (3.5) becomes

$$u_{i,j+1} \approx u_{i,j} + \alpha \left[u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right]. \qquad (3.6)$$

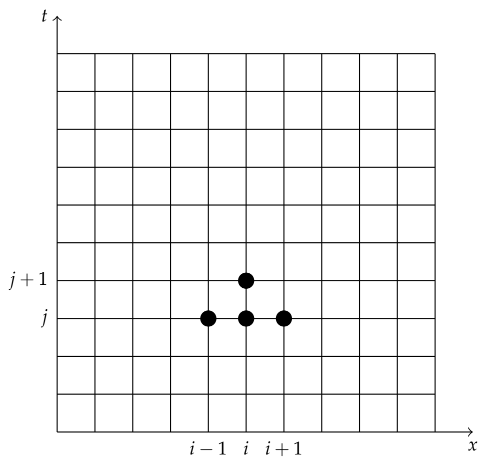

Figure 3.1: This *stencil* indicates the four types of terms in the finite difference scheme in Equation (3.7). The black circles represent the four terms in the equation, $u_{i,j}$ $u_{i-1,j}$ $u_{i+1,j}$ and $u_{i,j+1}$.

Equation (3.7) is the finite difference scheme for solving the heat equation. This equation is represented by the stencil shown in Figure 3.1. The black circles represent the four terms in the equation, $u_{i,j}$ $u_{i-1,j}$ $u_{i+1,j}$ and $u_{i,j+1}$.

Let's assume that the initial condition is given by

$$u(x, 0) = f(x).$$

Then, we have $u_{i,0} = f(x_i)$. Knowing these values, denoted by the open circles in Figure 3.2, we apply the stencil to generate the solution on the $j = 1$ row. This is shown in Figure 3.2.

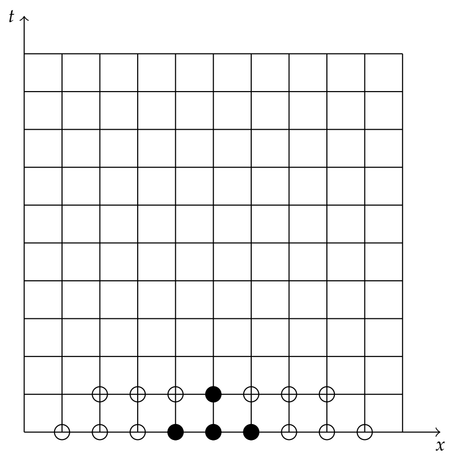

Figure 3.2: Applying the stencil to the row of initial values gives the solution at the next time step.

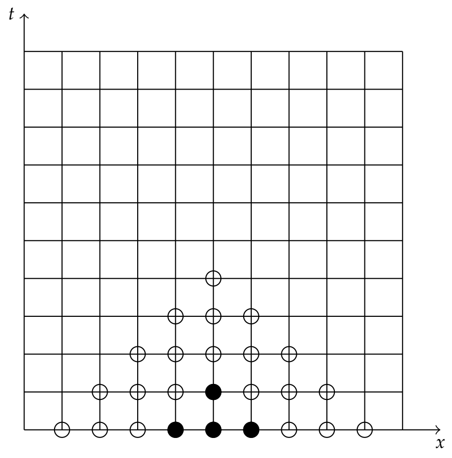

Figure 3.3: Continuation of the process provides solutions at the indicated points.

Further rows are generated by successively applying the stencil on each row, using the known approximations of $u_{i,j}$ at each level. This gives the values of the solution at the open circles shown in Figure 3.3. We notice that the solution can only be obtained at a finite number of points on the grid.

In order to obtain the missing values, we need to impose boundary conditions. For example, if we have Dirichlet conditions at $x = a$,

$$u(a,t) = 0,$$

or $u_{0,j} = 0$ for $j = 0, 1, \ldots,$ then we can fill in some of the missing data points as seen in Figure 3.4.

The process continues until we again go as far as we can. This is shown in Figure 3.5.

We can fill in the rest of the grid using a boundary condition at $x = b$. For Dirichlet conditions at $x = b$,

$$u(b,t) = 0,$$

or $u_{N,j} = 0$ for $j = 0, 1, \ldots,$ then we can fill in the rest of the missing data points as seen in Figure 3.6.

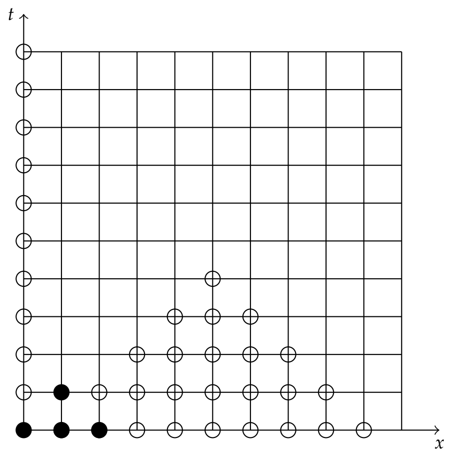

Figure 3.4: Knowing the values of the solution at $x = a$, we can fill in more of the grid.

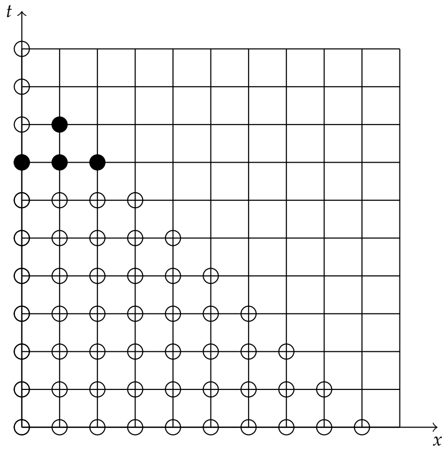

Figure 3.5: Knowing the values of the solution at other times, we continue to fill the grid as far as the stencil can go.

We could also use Neumann conditions. For example, let

$$u_x(a,t) = 0.$$

The approximation to the derivative gives

$$\left.\frac{\partial u}{\partial x}\right|_{x=a} \approx \frac{u(a+\Delta x,t) - u(a,t)}{\Delta x} = 0.$$

Then,

$$u(a+\Delta x,t) - u(a,t),$$

or $u_{0,j} = u_{1,j}$, for $j = 0, 1, \ldots$. Thus, we know the values at the boundary and can generate the solutions at the grid points as before.

We now have to code this using software. We can use MATLAB to do this. An example of the code is given below. In this example we specify the length of the rod, $L = 1$, and the heat constant, $k = 1$. The code is run for $t \in [0, 0.1]$.

The grid is created using $N = 10$ subintervals in space and $M = 50$ time steps. This gives dx = $\Delta x$ and dt = $\Delta t$. Using these values, we find the numerical scheme constant $\alpha = k\Delta t/(\Delta x)^2$.

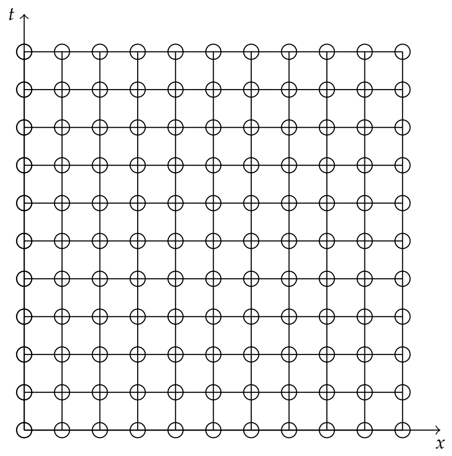

Figure 3.6: Using boundary conditions and the initial condition, the grid can be fill in through any time level.

Nest, we define $x_i = i * dx$, $i = 0, 1, \ldots, N$. However, in MATLAB, we cannot have an index of 0. We need to start with $i = 1$. Thus, $x_i = (i-1) * dx$, $i = 1, 2, \ldots, N+1$.

Next, we establish the initial condition. We take a simple condition of

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

We have enough information to begin the numerical scheme as developed earlier. Namely, we cycle through the time steps using the scheme. There is one loop for each time step. We will generate the new time step from the last time step in the form

$$u_i^{new} = u_i^{old} + \alpha \left[ u_{i+1}^{old} - 2u_i^{old} + u_{i-1}^{old} \right]. \tag{3.7}$$

This is done suing $u0(i) = u_i^{new}$ and $u1(i) = u_i^{old}$.

At the end of each time loop we update the boundary points so that the grid can be filled in as discussed. When done, we can plot the final solution. If we want to show solutions at intermediate steps, we can plot the solution earlier.

```
% Solution of the Heat Equation Using a Forward Difference Scheme

% Initialize Data
%     Length of Rod, Time Interval
%     Number of Points in Space, Number of Time Steps

L=1;
T=0.1;
k=1;
N=10;
M=50;
dx=L/N;
dt=T/M;
alpha=k*dt/dx^2;
```

```
% Position

for i=1:N+1
    x(i)=(i-1)*dx;
end

% Initial Condition

for i=1:N+1
    u0(i)=sin(pi*x(i));
end

% Partial Difference Equation (Numerical Scheme)

for j=1:M
   for i=2:N
      u1(i)=u0(i)+alpha*(u0(i+1)-2*u0(i)+u0(i-1));
   end
   u1(1)=0;
   u1(N+1)=0;
   u0=u1;
end

% Plot solution
plot(x, u1);
```

## 3.2 *Truncation Error*

In the previous section we found a finite difference scheme for numerically solving the one dimensional heat equation. We have from Equations (3.5) and (3.7),

$$\begin{aligned} u(x,t+\Delta t) &\approx u(x,t)+\alpha\left[u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)\right]. && (3.8)\\ u_{i,j+1} &\approx u_{i,j}+\alpha\left[u_{i+1,j}-2u_{i,j}+u_{i-1,j}\right], && (3.9)\end{aligned}$$

where $\alpha = k\Delta t/(\Delta x)^2$. For points $x \in [a,b]$ and $t \geq 0$, we use the scheme to find approximate values of $u(x_i,t_i) = u_{i,j}$ at positions $x_i = a + i\Delta x, \quad i = 0,1,\ldots,N$, and times $t_j = j\Delta t, \quad j = 0,1,2,\ldots$.

In implementing the scheme we have found that there are errors introduced just like when using Euler's Method for ordinary differential equations. These truncations errors can be found by applying Taylor approximations just like we had for ordinary differential equations. In the schemes (3.8) and (3.9), we have not use equality. In order to replace the approximation by an equality, we need t estimate the order of the terms neglected in a Taylor series approximation of the time and space derivatives we have approximated.

We begin with the time derivative approximation. We used the forward difference approximation (3.1),

$$\frac{\partial u}{\partial t} \approx \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}. \tag{3.10}$$

This can be derived from the Tayloer series expansion of $u(x,t+\Delta t)$ about $\Delta t = 0$,

$$u(x,t+\Delta t) = u(x,t) + \frac{\partial u}{\partial t}(x,t)\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(x,t)(\Delta t)^2 + O((\Delta t)^3).$$

Solving for $\frac{\partial u}{\partial t}(x,t)$, we obtain

$$\frac{\partial u}{\partial t}(x,t) = \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} - \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(x,t)\Delta t + O((\Delta t)^2).$$

We see that we have obtained the forward difference approximation (3.1) with the added benefit of knowing something about the error terms introduced in the approximation. Namely, when we approximate $u_t$ with the forward difference approximation (3.1), we are making an error of

$$E(x,t,\Delta t) = -\frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(x,t)\Delta t + O((\Delta t)^2).$$

We have truncated the Taylor series to obtain this approximation and we say that

$$\frac{\partial u}{\partial t} = \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} + O(\Delta t) \tag{3.11}$$

is a first order approximation in $\Delta t$.

In a similar manor, we can obtain the truncation error for the $u_x x$- term. However, instead of starting with the approximation we used in Equation **??**uxx), we will derive a term using the Taylor series expansion of $u(x+\Delta x,t)$ about $\Delta x = 0$. Namely, we begin with the expansion

$$\begin{aligned} u(x+\Delta x,t) &= u(x,t) + u_x(x,t)\Delta x + \frac{1}{2!}u_{xx}(x,t)(\Delta x)^2 + \frac{1}{3!}u_{xxx}(x,t)(\Delta x)^3 \\ &\quad + \frac{1}{4!}u_{xxxx}(x,t)(\Delta x)^4 + \dots . \end{aligned} \tag{3.12}$$

We want to solve this equation for $u_{xx}$. However, there are some obstructions, like needing to know the $u_x$ term. So, we seek a way to eliminate lower order terms. On way is to note that replacing $\Delta x$ by $-\Delta x$ gives

$$\begin{aligned} u(x-\Delta x,t) &= u(x,t) - u_x(x,t)\Delta x + \frac{1}{2!}u_{xx}(x,t)(\Delta x)^2 - \frac{1}{3!}u_{xxx}(x,t)(\Delta x)^3 \\ &\quad + \frac{1}{4!}u_{xxxx}(x,t)(\Delta x)^4 + \dots . \end{aligned} \tag{3.13}$$

Adding these Taylor series, we have

$$\begin{aligned} u(x+\Delta x,t) + u(x+\Delta x,t) &= 2u(x,t) + u_{xx}(x,t)(\Delta x)^2 \\ &\quad + \frac{2}{4!}u_{xxxx}(x,t)(\Delta x)^4 + O((\Delta x)^6). \end{aligned} \tag{3.14}$$

We can now solve for $u_{xx}$ to find

$$\begin{aligned} u_{xx}(x,t) &= \frac{u(x+\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{(\Delta x)^2} \\ &\quad +\frac{2}{4!}u_{xxxx}(x,t)(\Delta x)^2+O((\Delta x)^4). \end{aligned} \tag{3.15}$$

Thus, we have that

$$u_{xx}(x,t)=\frac{u(x+\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{(\Delta x)^2}+O((\Delta x)^2)$$

is the second order in $\Delta x$ approximation of $u_{xx}$.

Combining these results, we find that the heat equation is approximated by

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}=\frac{u(x+\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{(\Delta x)^2}+O\left((\Delta x)^2,\Delta t\right).$$

This has local truncation error that is first order in time and second order in space.

## 3.3 *Stability*

Another consideration for numerical schemes for the heat equation is the stability of the scheme. In implementing the finite difference scheme,

$$u_{m,j+1}=u_{m,j}+\alpha\left[u_{m+1,j}-2u_{m,j}+u_{m-1,j}\right], \tag{3.16}$$

$\alpha = k\Delta t/(\Delta x)^2$, one finds that the solution goes crazy when $\alpha$ is too big. In other words, if you try to push the individual time steps too far into the future, then something goes haywire. We can determine the onset of instability by looking at the solution of this equation for $u_{m,j}$. [Note: We changed index $i$ to $m$ to avoid confusion later in this section.]

The scheme is actually what is called a partial difference equation for $u_{m,j}$. We could write it in terms of difference, such as $u_{m+1,j}-u_{m,j}$ and $u_{m,j+1}-u_{m,j}$. The furthest apart the time steps are are one unit and the spatial points are two units apart. We can see this in the stencils in Figure 3.1. So, this is a second order partial difference equation similar to the idea that the heat equation is a second order partial differential equation. The heat equation can be solved using the method of separation of variables. The difference scheme can also be solved in a similar fashion. We will show how this can lead to product solutions.

We begin by assuming that $u_{mj}=X_mT_j$, a product of functions of the indices $m$ and $j$. Inserting this guess into the finite difference equation, we have

$$\begin{aligned} u_{m,j+1} &= u_{m,j}+\alpha\left[u_{m+1,j}-2u_{m,j}+u_{m-1,j}\right], \\ X_mT_{j+1} &= X_mT_j+\alpha\left[X_{m+1}-2X_m+X_{m-1}\right]T_j, \\ \frac{T_{j+1}}{T_j} &= \frac{\alpha X_{m+1}+(1-2\alpha)X_m+\alpha X_{m-1}}{X_m}. \end{aligned} \tag{3.17}$$

Noting that we have a function of $j$ equal to a function of $m$, then we can set each of these to a constant, $\lambda$. Then, we obtain two ordinary difference equations:

$$\begin{aligned} T_{j+1} &= \lambda T_j, & (3.18)\\ \alpha X_{m+1} + (1-2\alpha)X_m + \alpha X_{m-1} &= \lambda X_m. & (3.19) \end{aligned}$$

The first equation is a simple first order difference equation and can be solved by iteration:

$$\begin{aligned} T_{j+1} &= \lambda T_j,\\ &= \lambda(\lambda T_{j-1}) = \lambda^2 T_{j-1},\\ &= \lambda^3 T_{j-2},\\ &= \lambda^{j+1} T_0, & (3.20) \end{aligned}$$

The second difference equation can be solved by making a guess in the same spirit as solving a second order constant coefficient differential equation.Namely, let $X_m = \xi^m$ for some number $\xi$. This gives

$$\begin{aligned} \alpha X_{m+1} + (1-2\alpha)X_m + \alpha X_{m-1} &= \lambda X_m,\\ \xi^{m-1}\left[\alpha\xi^2 + (1-2\alpha)\xi + \alpha\right] &= \lambda \xi^m\\ \alpha\xi^2 + (1-2\alpha-\lambda)\xi + \alpha &= 0. & (3.21) \end{aligned}$$

This is an equation foe $\xi$ in terms of $\alpha$ and $\lambda$. Due to the boundary conditions, we expect to have oscillatory solutions. So, we can guess that $\xi = |\xi|e^{i\theta}$, where $i$ here is the imaginary unit. We assume that $|\xi| = 1$, and thus $\xi = e^{i\theta}$ and $X_m = \xi^m = e^{im\theta}$. Since $x_m = m\Delta x$, we have $X_m = e^{ix_m\theta/\Delta x}$. We define $\beta = theta/\Delta$, to give $X_m = e^{i\beta x_m}$ and $\xi = e^{i\beta\Delta x}$.

Inserting this value for $\xi$ into the quadratic equation for $\xi$, we have

$$\begin{aligned} 0 &= \alpha\xi^2 + (1-2\alpha-\lambda)\xi + \alpha\\ &= \alpha e^{2i\beta\Delta x} + (1-2\alpha-\lambda)e^{i\beta\Delta x} + \alpha\\ &= e^{i\beta\Delta x}\left[\alpha(e^{i\beta\Delta x} + e^{-i\beta\Delta x}) + (1-2\alpha-\lambda)\right]\\ &= e^{i\beta\Delta x}\left[2\alpha\cos(\beta\Delta x) + (1-2\alpha-\lambda)\right]\\ \lambda &= 2\alpha\cos(\beta\Delta x) + 1 - 2\alpha. & (3.22) \end{aligned}$$

So, we have found that

$$u_{mj} = X_m T_j = \lambda^m(a\cos\alpha x_m + b\sin\alpha x_m), \quad a^2 + b^2 = h_0^2,$$

and

$$\lambda = 2\alpha\cos(\beta\Delta x) + 1 - 2\alpha.$$

For the solution to remain bounded, or stable, we need $|\lambda| \leq 1$.

Therefore, we have the inequality

$$-1 \leq 2\alpha\cos(\beta\Delta x) + 1 - 2\alpha \leq 1.$$

Since $\cos(\beta \Delta x) \leq 1$, the upper bound is obviously satisfied. Since $-1 \leq \cos(\beta \Delta x)$, the lower bound is satisfied for $-1 \leq -2s + 1 - 2s$, or $s \leq \frac{1}{2}$. Therefore, the stability criterion is satisfied when

$$\alpha = k \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}.$$