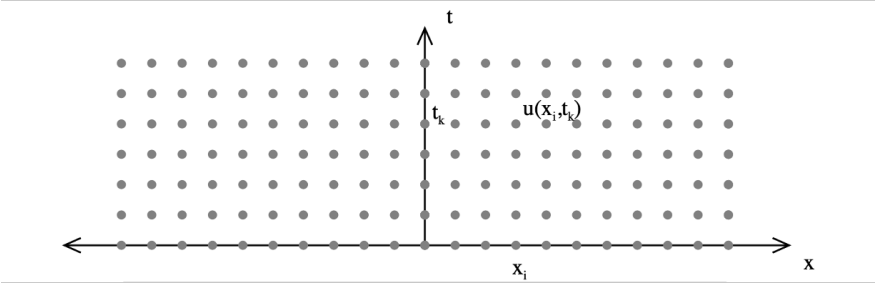
# 1 Diffusion Equations

## 1.1 Review

We discussed the numerical solutions of the diffusion equation in previous lectures. The diffusion equation in one spatial dimension is given by

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \tag{1}$$

If only the initial condition is given $u(x, 0) = \eta(x)$ (and it implies $-\infty < x < \infty$), the question is called an initial value problem (Cauchy problem). The diffusion equation describes the diffusion of a quantity $u$ in space and time.




- The discreization over space and time is illustrated above. The grid points are denoted by $(x_j, t_n)$ and use approximation $U_j^n \approx u(x_j, t_n)$ where $j$ is the spatial index and $n$ is the time index.

- We used the finite difference method to discretize the time derivative and spatial derivative.

- The MOL method is introduced to discuss the numerical solution of the diffusion equation, in which we discretize the spatial derivative using the finite difference and treat the time derivative as as system of ordinary differential equation (ODE).

- The FTCS, BTCS, and Crank-Nicholson methods are used to solve the diffusion equation.

- We discussed the initial and boundary value problems, and how to enforce the boundary conditions in the numerical methods.

- We discussed the convergence and stability of numerical methods for PDEs. We also discussed the error analysis of numerical methods.

- von Neumann stability analysis is used to analyze the stability of numerical methods. And we use the amplication factor $G(\xi)$ and eigengridfunction $W_j$ to analyze the stability of the numerical method.

  We note the von Neumann stability analysis is based on the Fourier transform of the numerical method.

## 1.2 Multi-dimension Diffusion Equation

In two spatial dimensions, the diffusion equation is given by

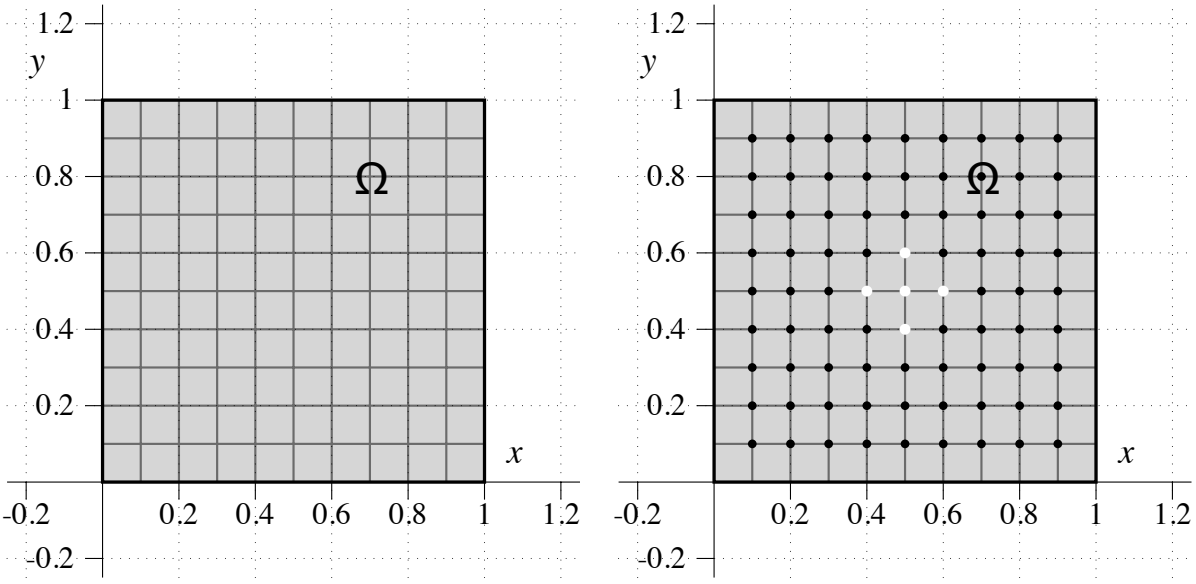
$$\frac{\partial u}{\partial t} - \kappa \nabla^2 u \equiv \frac{\partial u}{\partial t} - \kappa(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = f(x, y, t) \tag{2}$$

where $f(x, y, t)$ is an known function, $\nabla^2$ is the Laplacian operator, and $u(x, y, t)$ is the unknown function. Here in 2D, the initial condition is given by $u(x, y, 0) = \eta(x, y)$, and the boundary condition is given by $u(x, y, t) = \eta(x, y, t)$.

In 2D, we need to discretize the equation in two dimensions, and enforce the boundary conditions in two dimensions. The spatial step $\triangle x$ and $\triangle y$ are the grid spacings in the $x$ and $y$ directions, respectively. And the most easy setup is to assume they are equal, i.e., $\triangle x = \triangle y = h$. Then $(x_i, y_j)$ is the grid point in the $i$-th row and $j$-th column, and the grid point is given by $(x_i, y_j) = (x_0 + i\triangle x, y_0 + j\triangle y)$, where $i$ and $j$ are the grid indices, and the point $(x_0, y_0)$ is the origin of the grid. Now the approximation of $u(x, y, t)$ at the grid point $(x_i, y_j)$ and time level $t_n$ is denoted by $U_{ij}^n \approx u(x_i, y_j, t_n)$.

Here the Laplacian operator $\nabla^2$ is defined as $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$, and we can discretize it using the standard 5-point stencil to get $\nabla_h^2$, applied to $U_{ij}^n$ as

$$\begin{aligned}\nabla_h^2 U_{ij}^n =& \frac{U_{i+1,j}^n - 2U_{ij}^n + U_{i-1,j}^n}{h^2} + \frac{U_{i,j+1}^n - 2U_{ij}^n + U_{i,j-1}^n}{h^2} \\ =& \frac{U_{i+1,j}^n + U_{i-1,j}^n + U_{i,j+1}^n + U_{i,j-1}^n - 4U_{ij}^n}{h^2}\end{aligned} \tag{3}$$




If $\Delta x = \Delta y = h = 1/(m+1)$, we have $m^2$ grid points, we can estimate the error $\nabla_h^2 u$ at the grid point $(x_i, y_j)$ using Taylor expansion:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2}\Big|_{(x,y)=(x_i,y_j)} =$$

$$\frac{-4u_{i,j} + u_{i-1,j} + u_{i+1,j} + u_{i,j+1} + u_{i,j-1}}{h^2} - \tau_{i,j},$$

where $\tau_{i,j} = \mathcal{O}(h^2)$.

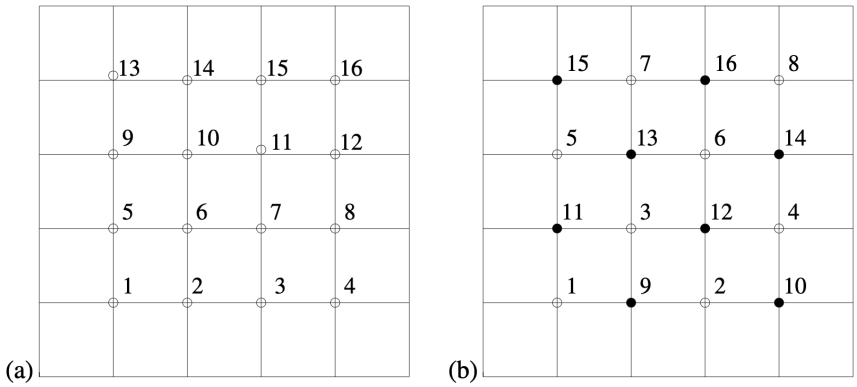
If we discretize the Laplacian operator using the 5-point stencil above, and trapezoidal rule for the time derivative, we can get the following discretization in Crank-Nicholson method ($k = \Delta t$ is the time step size):

$$\frac{U_{ij}^{n+1} - U_{ij}^{n}}{k} - \frac{1}{2}\kappa\left(\nabla_h^2 U_{ij}^{n+1} + \nabla_h^2 U_{ij}^{n}\right) = f_{ij}^{n+1/2} \tag{4}$$

where $f_{ij}^{n+1/2} = f(x_i, y_j, (t_n + t_{n+1})/2)$ is the middle point of the function $f$ at time level $n$ and $n+1$. Rearranging the above equation gives

$$\begin{aligned} U_{ij}^{n+1} =& U_{ij}^{n} + \frac{k}{2}\kappa\left(\nabla_h^2 U_{ij}^{n+1} + \nabla_h^2 U_{ij}^{n}\right) + k f_{ij}^{n+1/2} \Longrightarrow \\ U_{ij}^{n+1} - \frac{k}{2}\kappa\nabla_h^2 U_{ij}^{n+1} =& U_{ij}^{n} + \frac{k}{2}\kappa\nabla_h^2 U_{ij}^{n} + k f_{ij}^{n+1/2} \Longrightarrow \\ (I - \frac{k}{2}\kappa\nabla_h^2) U_{ij}^{n+1} =& (I + \frac{k}{2}\kappa\nabla_h^2) U_{ij}^{n} + k f_{ij}^{n+1/2} \end{aligned} \tag{5}$$

We note the matrix $A = I - \frac{k}{2}\kappa\nabla_h^2$ is a tridiagonal matrix, and the matrix $B = I + \frac{k}{2}\kappa\nabla_h^2$ is also having special structure. The structure of both $A$ and $B$ is determined by how to order the grid points. For example



In general, if we stack variables sequentially: $u_{i,j}$ is located in the $(i + (j - 1) \cdot m)$-th entry of $\vec{U}$

$$\vec{U} = \begin{bmatrix} u^{[1]} \\ u^{[2]} \\ \vdots \\ u^{[m]} \end{bmatrix}, \quad \text{where} \quad u^{[j]} = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{m,j} \end{bmatrix}.$$

The result matrix is

$$A = \begin{bmatrix} T & I_p & & & \\ I_p & T & I_p & & \\ & I_p & T & I_p & \\ & & \ddots & \ddots & \ddots \\ & & & I_p & T \end{bmatrix},$$

where $I_p$ is an $m \times m$ identity matrix multiplied with $k/h^2$, and

$$T = \begin{bmatrix} 1 - \frac{4k\kappa}{2h^2} & \frac{k}{h^2} & & & & \\ \frac{k}{h^2} & 1 - \frac{4k\kappa}{2h^2} & \frac{k}{h^2} & & & \\ & \frac{k}{h^2} & 1 - \frac{4k\kappa}{2h^2} & \frac{k}{h^2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{k}{h^2} & 1 - \frac{4k\kappa}{2h^2} \end{bmatrix}.$$

The matrix $T$ is a tridiagonal matrix, and the matrix $A$ is a block tridiagonal matrix. And we know the eigenvalues of the block tridiagonal matrix $A$ are given by (let $\kappa = 1$)

$$\begin{aligned}\lambda_{p,q} =& 1 + \frac{4k}{2h^2} - \frac{k}{h^2}\cos(\frac{p\pi}{m+1}) + \frac{k}{h^2}\cos(\frac{q\pi}{m+1}) \\ =& 1 - \frac{k}{h^2}[(\cos(\frac{p\pi}{m+1}) - 1) + (\cos(\frac{q\pi}{m+1}) - 1)]\end{aligned}$$

It's now easy to see that solving the 2D diffusion equation using CN method needs to solve a system of linear equations in (5), which has a block tridiagonal coefficient matrix, and the direct methods is not efficient. We usually use the iterative methods to solve the system of linear equations.

### 1.2.1 Locally One-dimensional Method

Instead of solving (5) directly so the $x$ adn $y$ directions are coupled, there is an alternative approach that is to replace this fully coupled single time step with a sequence of steps, each of which is coupled in only one space direction, resulting in a set of tridiagonal systems which can be solved much more easily. One example is the locally one-dimensional (LOD) method ($\kappa = 1,\ f = 0$):

$$\begin{aligned}U_{ij}^{*} =& U_{ij}^{n} + \frac{k}{2}\left(D_x^2 U_{ij}^{n} + D_x^2 U_{ij}^{*}\right) \\ U_{ij}^{n+1} =& U_{ij}^{*} + \frac{k}{2}\left(D_y^2 U_{ij}^{*} + D_y^2 U_{ij}^{n+1}\right)\end{aligned} \tag{6}$$

where $D_x^2$ and $D_y^2$ are the finite difference operators in the $x$ and $y$ directions, respectively. The LOD method is a semi-implicit method, and it is unconditionally stable. We can write it in the matrix form as

$$\begin{aligned}(I - \frac{k}{2}D_x^2)U^* &= (I + \frac{k}{2}D_x^2)U^n \\ (I - \frac{k}{2}D_y^2)U^{n+1} &= (I + \frac{k}{2}D_y^2)U^*\end{aligned} \tag{7}$$

where $U^*$ is the intermediate solution.

In the first step of (6), we solve the tridiagonal system to approximate $u_t = u_{xx}$ along $x$ over the time $k = \triangle t$, and the result is $U^*$. And then in the second step, we solve the tridiagonal system to approximate $u_t = u_{yy}$ along $y$ over the time $k = \triangle t$, and the result is $U^{n+1}$.

Note in the first step when we solve $U^*$, for any fixed value of $j$ we have a tridiagonal system of equations to solve for $U_{ij}(i = 1, 2, \cdots, m)$. The system obtained for each value of $j$ is completely decoupled from the system obtained for other values of $j$. And in the second step when we solve $U^{n+1}$, for any fixed value $i$ we have a tridiagonal system of equations to solve for $U_{ij}(j = 1, 2, \cdots, m)$. So this LOD greatly reduces the computational cost and complexity of solving the diffusion equation in 2D.

**Boundary conditions:** Note in solving the second step of (6), we need to have values of $U_{ij}^*$ at the boundary points, i.e., $U_{i,0}^*$, $U_{i,m+1}^*$ along the bottom,

and $U_{i,0}^{n+1}$, $U_{i,m+1}^{n+1}$ along the top. To get $U_{i,j}^{*}$ with $j = 0$ or $j = m + 1$, we can solve the first step of (6) with $j = 0$ and $j = m + 1$.

To solve the first step of (6), we need to have values of $U_{0,j}^{*}$ along the left boundary, and $U_{m+1,j}^{*}$ along the right boundary. To get $U_{i,j}^{*}$ with $i = 0$ or $i = m + 1$, we can use the second equation in (6) to solve for $U_{i,j}^{*}$ with $i = 0$ or $i = m + 1$. For example, at $i = 0$, the second line of (6) gives a system of algebraic equations for $U_{0,j}^{*}$, $U_{1,j}^{*}$, $U_{2,j}^{*}$, $\cdots$, $U_{m,j}^{*}$, in terms of $U_{0,j}^{n+1}$, $U_{1,j}^{n+1}$, $U_{2,j}^{n+1}$, $\cdots$, $U_{m,j}^{n+1}$, which are available from the boundary conditions (Dirichlet).

### 1.2.2 Alternating Direction Implicit (ADI) Method

The ADI method is another method to solve the diffusion equation in 2D. The feature of the ADI method is that the two steps each involve discretization in only one spatial direction at the advanced $(n+1)$ time level (giving decoupled tridiagonal systems again) but coupled with discretization in the opposite direction at the old time level. The classical method of this form is

$$\begin{aligned} U_{ij}^{*} &= U_{ij}^{n} + \frac{k}{2}\left(D_y^2 U_{ij}^{n} + D_x^2 U_{ij}^{*}\right) \\ U_{ij}^{n+1} &= U_{ij}^{*} + \frac{k}{2}\left(D_x^2 U_{ij}^{*} + D_y^2 U_{ij}^{n+1}\right) \end{aligned} \tag{8}$$

We can see the decoupled tridiagonal systems are obtained in the matrix form as

$$\begin{aligned}(I - \frac{k}{2}D_x^2)U^* =&(I + \frac{k}{2}D_y^2)U^n \\ (I - \frac{k}{2}D_y^2)U^{n+1} =&(I + \frac{k}{2}D_x^2)U^*\end{aligned} \tag{9}$$

Comparing to the LOD method (6), we can see the ADI method involves diffusion in both the $x$ and $y$ directions at the same time. And we can treat $U^*$ as the intermediate solution, and $U^{n+1}$ as the final solution. The ADI method is also unconditionally stable, and it is second order accurate in both space and time.