Finite Difference Method

Rongrong Liu

Math 320 Final Project

1 Introduction

Finite difference method is also called the method of lines. Just as the literal meaning, it divides the two dimensional space into lines. It is defined as a method that reduces the partial differential equations to algebraic equations by discretization and approximation.

Discretization helps divide the domain into grids and identify these finite number of discrete points, each of which corresponds to a linear equation by finite difference approximation of the derivatives. All these linear equations together can be converted to form a matrix. We will gain the final solution by solving the inversion of the matrix.

In this paper, I will first introduce the finite difference method through the derivation of finite difference approximation from Taylor's series, and then apply this method to solve the LaPlace's and Poisson's equations respectively. Finally, I will analyze the error produced.

2 Derivation of Finite Difference Approximation

The derivation of finite difference method comes from Taylor's series:

$U(t, x_0+\Delta x) = U(t, x_0)+\Delta x U_x(t, x_0)+\frac{\Delta x^2}{2!}U_{xx}(t, x_0)+...+\frac{\Delta x^(n-1)}{(n-1)!}U_{n-1}(t, x_0)+ O(\Delta x^n)$

Truncate the equation to the first order gives

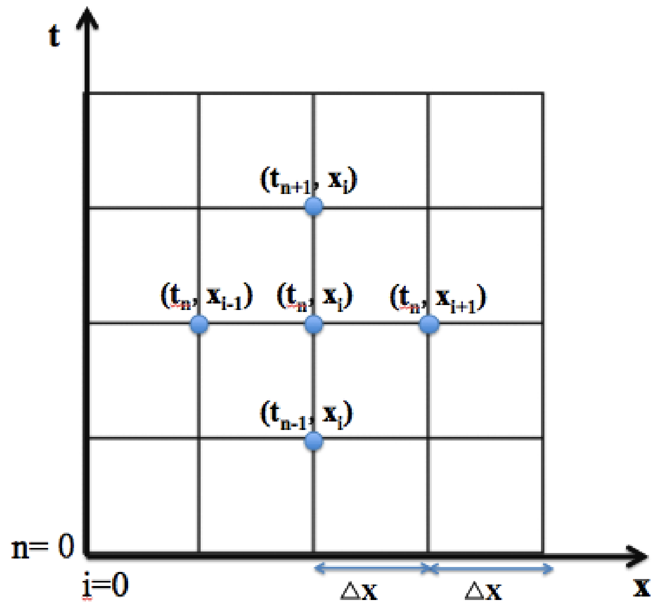
$U(t, x_0 + \Delta x) = U(t, x_0) + \Delta x U_x(t, x_0) + O(\Delta x^2)$

From this, we get the derivative of U holding time to be constant

$U_x(t, x_0) = \frac{U(t,x_0+\Delta x)-U(t,x_0)}{\Delta x} - \frac{O(\Delta x^2)}{\Delta x}$

$U_x(t, x_0) = \frac{U(t,x_0+\Delta x)-U(t,x_0)}{\Delta x} - O(\Delta x)$

This equation holds for all (t,x_0). We assume constant grid spacing Δx in x so that $x_{i+1} = x_i + \Delta x$. Extend this to any grid point (t_n, x_i).



Notation: denote $U_i^n = U(\mathrm{t}_n,\, x_i)$

$U_x(t_n, x_i) = \frac{U(t_n,x_{i+1})-U(t_n,x_i)}{\Delta x} - O(\Delta x)$

If we eliminate the remainder term $O(\Delta x)$, we can get the *forward method* of the finite difference approximation:

$U_x(t_n, x_i) \approx \frac{U_{i+1}^n - U_i^n}{\Delta x}$

Now, replace Δx by$-\Delta x$

$U(t, x_0 - \Delta x) = U(t, x_0) - \Delta x U_x(t, x_0) + O(\Delta x^2)$

$U_x(t, x_0) = \frac{U(t,x_0)-U(t,x_0-\Delta x)}{\Delta x} + \frac{O(\Delta x^2)}{\Delta x}$

$U_x(t, x_0) = \frac{U(t,x_0)-U(t,x_0-\Delta x)}{\Delta x} + O(\Delta x)$

Just as how we derive the forward method, eliminating the remainder term gives the *backward method* of finite difference approximation in the discrete case:

$U_x(t_n, x_i) \approx \frac{U_i^n - U_{i-1}^n}{\Delta x}$

If we add the two equations together and divide it by 2, we get the *center method*:

$U_x(t_n, x_i) \approx \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}$

Use the third order Taylor's Series:

$U(t, x_0 + \Delta x) = U(t, x_0) + \Delta x U_x(t, x_0) + \frac{\Delta x^2}{2!} U_{xx}(t, x_0) + \frac{\Delta x^3}{3!} U_{xxx}(t, x_0) + O(\Delta x^4)$

$U(t, x_0 - \Delta x) = U(t, x_0) - \Delta x U_x(t, x_0) + \frac{\Delta x^2}{2!} U_{xx}(t, x_0) - \frac{\Delta x^3}{3!} U_{xxx}(t, x_0) + O(\Delta x^4)$

$U(t, x_0 + \Delta x) + U(t, x_0 - \Delta x) = 2U(t, x_0) + \Delta x^2 U_{xx}(t, x_0) + O(\Delta x^4)$

Assuming equal space in x and generalize the case:

$U_{i+1}^n + U_{i-1}^n = 2U_i^n + \Delta x^2 U_{xx}(t_n, x_i) + O(\Delta x^4)$

$U_{xx}(t_n, x_i) \approx \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}$

This equation is the key to solve LaPlace's and Poisson's equations, which will be all mentioned in the next two sections.

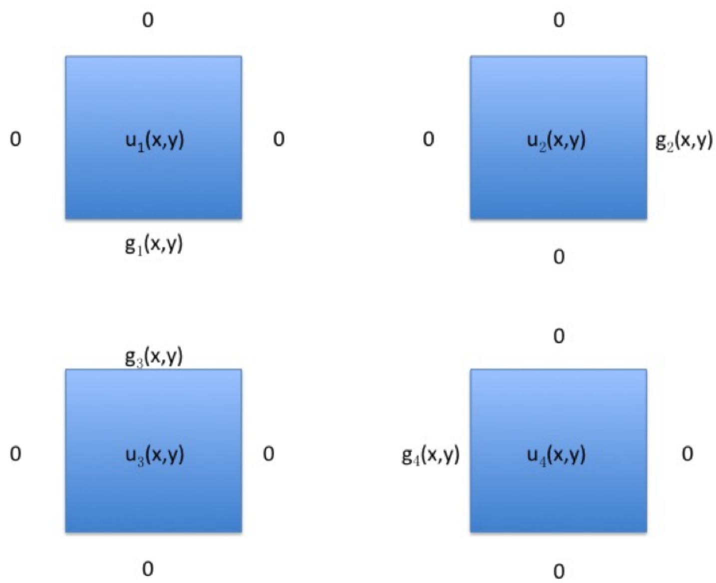
3 2D LaPlace's Equation by FDM

LaPlace's equation: $\Delta U = U_{xx} + U_{yy} = 0$

The LaPlace's equation will be solved by principle of superposition and separation of variables.

First, separate U(x,y) into four U's $U_1(x, y), U_2(x, y), U_3(x, y), U_4(x, y)$. All of the U's are rectangles with functions on three of the sides to be zero and one side to be nonzero. As shown in the graph below, there are four case in total. Then, Adding the four cases together will give the generalized answer of $U(x, y)$.

$U(x, y) = U_1(x, y) + U_2(x, y) + U_3(x, y) + U_4(x, y)$



Finite difference method will be used to solve each U_1, U_2, U_3, U_4.

From the equation derived from Taylor's Series, for any point on the grid (x_i, y_j), we have

$$\frac{U_{i-1,j}-2U_{i,j}+U_{i+1,j}}{\Delta x}+\frac{U_{i,j-1}-2U_{i,j}+U_{i,j+1}}{\Delta y}=0$$

When $\Delta x = \Delta y = h$

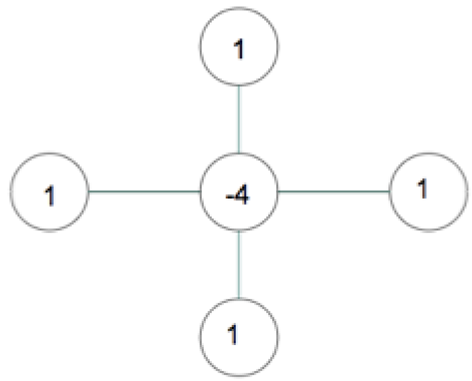
$$\frac{U_{i-1,j}-2U_{i,j}+U_{i-1,j}+U_{i,j-1}-2U_{i,j}+U_{i,j+1}}{h^2}=0$$

Since $h > 0$

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j} = 0$$

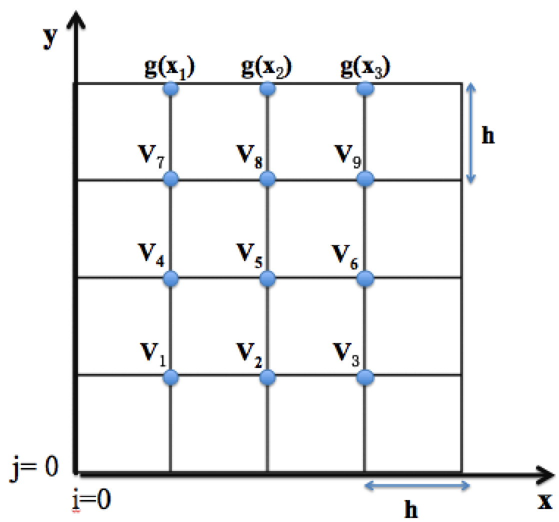
Use $U_3(x, y)$ and discretize $U_3(x, y)$ into a n× n grid. We generate a system of linear equations of the unknowns $U_{i,j}$ at the inner grid. Meanwhile, apply the boundary conditions $U_{0,j} = 0, U_{n,j} = 0, U_{i,0} = 0, U_{i,n} = g(x)$

From the equation above, we can generate a model which set the coefficient (i, j) to be -4, and the coefficients of $(i-1, j), (i+1, j), (i, j-1), (i, j+1)$ to be 1.



Five−point stencil of inner grid points

Take a 4×4 grid as an example to illustrate the adoption of FDM, similar method can be used to solve U_1, U_2, U_4.



In this case, we will have 9 unknowns and 9 linear equations. As shown in the figure, the unknowns are denoted as $v_1, v_2, ..., v_9$. The grid points on on x=0,4 and y=0 all equal to zero, and grid points on y=4, $g(x_1) + g(x_2) + g(x_3) = g(x)$ Therefore

$-4v_1 + v_2 + \ldots + v_4 + \ldots + \ldots + \ldots + \ldots + \ldots = 0$

$v_1 - 4v_2 + v_3 + \ldots + v_5 + \ldots + \ldots + \ldots + \ldots = 0$

$\ldots + v_2 + 4v_3 + \ldots + \ldots + v_6 + \ldots + \ldots + \ldots = 0$

$v_1 + \ldots + \ldots - 4v_4 + v_5 + \ldots + v_7 + \ldots + \ldots = 0$

$\ldots + v_2 + \ldots + v_4 - 4v_5 + v_6 + \ldots + v_8 + \ldots = 0$

$\ldots + \ldots + v_3 + \ldots + v_5 - 4v_6 + \ldots + \ldots + v_9 = 0$

$\ldots + \ldots + \ldots + v_4 + \ldots + \ldots + 4v_7 + v_8 + \ldots = -g(x_1)$

$\ldots + \ldots + \ldots + \ldots + v_5 + \ldots + v_7 - 4v_8 + v_9 = -g(x_2)$

$\ldots + \ldots + \ldots + \ldots + \ldots + v_6 + \ldots + v_8 - 4v_9 = -g(x_3)$

This system of linear equations generate a matrix A, which flattens the matrix V formed by the unknowns of the inner grid points.

In this case, A=

$$\left[\begin{array}{ccc|ccc|ccc}
-4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4
\end{array}\right]$$

$$V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix}$$

d is the matrix formed by the right side of the equation, which is initial boundary condition.

$$d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \text{-g(x}_1\text{)} \\ \text{-g(x}_2\text{)} \\ \text{-g(x}_3\text{)} \end{bmatrix}$$

Multiplication of matrix A and V satisfies: A V= d

$$\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \text{-g(x}_1\text{)} \\ \text{-g(x}_2\text{)} \\ \text{-g(x}_3\text{)} \end{bmatrix}$$

In order to generalize A, we find the block structure of A. First, denote

$$D = \begin{bmatrix} -4 & 1 & \dots & \dots & 0 \\ 1 & -4 & 1 & \dots & 0 \\ 0 & 1 & -4 & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & 1 & -4 \end{bmatrix}$$

D is a $(n-1) \times (n-1)$ matrix.

$$I = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

I is an identity matrix of dimension $(n-1) \times (n-1)$.

Then A will become A= $\begin{bmatrix} D & I & \dots & \dots & 0 \\ I & D & I & \dots & 0 \\ 0 & I & D & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & I \\ 0 & 0 & \dots & I & D \end{bmatrix}$ A$\in R^{(n-1)^2 \times (n-1)^2}$.

Finally, we can solve U by the inverse matrix of A: $U = A^{-1}d$

This approach to get U (actually U_3 in this example) will also be applied to U_1, U_2, U_4. The final result of the LaPlace's equation is the sum of these U's:

$U(x,y) = U_1(x,y) + U_2(x,y) + U_3(x,y) + U_4(x,y)$

4 2D Poisson's Equation With Homogeneous Boundary Conditions by FDM

Poisson's equation:

$\Delta U = U_{xx} + U_{yy} = -f(x,y)$ in domain $\Omega = (0,1) \times (0,1)$

$U = 0$ on Γ=$\partial\Omega$

To solve the Poisson's equation, we use a similar method of solving the LaPlace's equation.

Just as the section above,from the equation derived from Taylor's Series, for any point on the grid (x_i, y_j), we have

$\frac{U_{i-1,j}-2U_{i,j}+U_{i+1,j}}{\Delta x}+\frac{U_{i,j-1}-2U_{i,j}+U_{i,j+1}}{\Delta y}$=0

When $\Delta x = \Delta y = h$

$U_{i-1,j}$+$U_{i+1,j}$+$U_{i,j-1}$+$U_{i,j+1}$-4$U_{i,j}$= h^2f$_{i,j}$

This should also satisfy the boundary conditions:

$U_{0,j} = 0, U_{n,j} = 0, U_{i,0} = 0, U_{i,n} = 0$

Using the five-point model shown in the last section, we are able to set up a system of linear equations:

$-4U_{1,1} + U_{2,1} + U_{1,2} = -h^2 f_{1,1}$

$U_{1,1} - 4U_{2,1} + U_{3,1} + U_{1,2} = -h^2 f_{2,1}$

$\vdots$

$U_{n-1,1} - 4U_{n,1} + U_{n,2} = -h^2 f_{n,1}$

$-\ 4U_{1,2} + U_{2,2} + U_{1,3} + U_{1,1} = -h^2 f_{1,2}$

$U_{1,2} - 4U_{2,2} + U_{3,2} + U_{2,3} + U_{2,1} = -h^2 f_{2,2}$

$U_{n-1,2} - 4U_{n,2} + U_{n,3} + U_{n,1} = -h^2 f_{n,2}$

$\vdots$

From these equations, we generate a coefficient matrix A that, which is the same A in the last section:

$$A = \begin{bmatrix} D & I & \dots & \dots & 0 \\ I & D & I & \dots & 0 \\ 0 & I & D & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & I \\ 0 & 0 & \dots & I & D \end{bmatrix}$$

where I is a $(n-1)\times(n-1)$ identity matrix and D= $\begin{bmatrix} -4 & 1 & \dots & \dots & 0 \\ 1 & -4 & 1 & \dots & 0 \\ 0 & 1 & -4 & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & 1 & -4 \end{bmatrix}$

Then we would like to find the unknown U by $U = A^{-1}d$, where

U=$[U_{1,1}, U_{2,1}, ..., U_{n,1}|U_{1,2}, U_{2,2}, ..., U_{n,2}|......|U_{1,n}, U_{2,n}, ..., U_{n,n}]^T$

d=$-h^2[f_{1,1}, f_{2,1}, ..., f_{n,1}|f_{1,2}, f_{2,2}, ..., f_{n,2}|......|f_{1,n}, f_{2,n}, ..., f_{n,n}]^T$

5 2D Poisson's Equation With Inhomogeneous Boundary Conditions by FDM

What if the U≠0 at the boundary? That is to say

$\Delta U = U_{xx} + U_{yy} = -f(x, y)$ in domain $\Omega = (0, 1) \times (0, 1)$

$U(x, y) = g(x, y)$ on Γ=∂Ω

Then our approach is to add an equation at every boundary grid points which satisfies the boundary conditions.

That is to say, at each (x_0, y_j), (x_i, y_0),(x_i, y_n),(x_n, y_j) we have some more equations:

$U_{0,j} = g_{0,j}$

U$_{i,0} = g_{i,0}$

U$_{n,j} = g_{n,j}$

U$_{i,n} = g_{i,n}$

Therefore, instead of $(n-1) \times (n-1)$ unknowns(i.e matrix),we have $(n+1) \times (n+1)$(i.e matrix) now. This approach does not change structure of the matrix, but add rows with a one at the diagonal. The dimension of A and U changes from $(n-1) \times (n-1)$ to $(n+1) \times (n+1)$.

6 Truncation Error

Truncation error is the error produced when we eliminate the $O(\Delta x)^n$ term as we do the finite difference approximation. Analyze the accuracy of first order derivatives.

(I). Forward difference approximation:

$U_x(t, x_0) = \frac{U(t,x+\Delta x_0)-U(t,x_0)}{\Delta x} - O(\Delta x)$

Truncation error $= O(\Delta x) = \frac{O(\Delta x)^2}{\Delta x} = -\frac{\Delta x}{2}U_{xx}(t, x_0) - \frac{\Delta x^2}{6}U_{xxx}(t, x_0) + ...$

(II). Backward difference approximation:

$U_x(t, x_0) = \frac{U(t,x_0)-U(t,x_0-\Delta x)}{\Delta x} + O(\Delta x)$

Truncation error $= O(\Delta x) = \frac{O(\Delta x^2)}{-\Delta x} = \frac{\Delta x}{2}U_{xx}(t, x_0) - \frac{\Delta x^2}{6}U_{xxx}(t, x_0) + ...$

(III). Central difference approximation:

$U_x(t, x_0) = \frac{U(t, x_0+\Delta x) - U(t, x_0-\Delta x)}{\Delta x} + O(\Delta x^2)$

Truncation error $= O(\Delta x^2) = -\frac{\Delta x^2}{6} U_{xxx}(t, x_0) + ...$

(IV). Second Order Derivatives of central difference approximation

$U_{i+1}^n + U_{i-1}^n = 2U_i^n + \Delta x^2 U_{xx}(t_n, x_i) + O(\Delta x^4)$

$U_{xx}(t_n, x_i) \approx \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}$

Truncation error $= \frac{O(\Delta x^4)}{\Delta x^2} = \frac{\Delta x^2}{12} U_{xxxx}(t_n, x_i) + ...$

7 Conclusion

As illustrated in this paper, finite difference approximation plays the most essential role to reduce partial differential equations to ordinary differential equations. Finite difference method combines the geometric mathematical thinking - by overlaying domain with grids and each grid point represents an unknown in the linear equation system, and algebraic mathematical think - by converting the linear equation system into matrices. This paper only talks about the second order approximation, but the accuracy of finite difference method will enhance as we use higher order approximation.

Finite difference method is a practical computational method in a lot of fields such as electrical engineering because the Poisson's equation is a good model for the electrostatic system behavior. In addition, it is also widely used to solve heat equation $U_t(x, t) = kU_{xx}(x, t)$ and wave equation $U_{tt}(x, t) = c^2 U_{xx}(x, t)$ with certain boundary conditions; hence, it is also a tool for research on heat transfer and wave motion.