Neumann Boundary conditions on 2D grid with non-uniform mesh space for elliptic PDE

Nasser M. Abbasi

Spring 2010 Compiled on January 29, 2024 at 2:42am

Contents

1 Introduction

This is a derivation of the 2D Laplacian finite difference approximation on 2D grid with Neumann boundary conditions for solving the elliptic PDE. The PDE is

$$
\begin{aligned}
-\nabla^2 u &= f(x,y) \\
-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) &= f(x,y)
\end{aligned}
$$

The derivation is given for the general case of non-uniform grid (meaning h_x is not equal to h_y). At the end it is simplified to uniform grid by setting $h_x = h_y$.

Notation used is based on the following grid where it is assumed that the grid size is $n \times n$

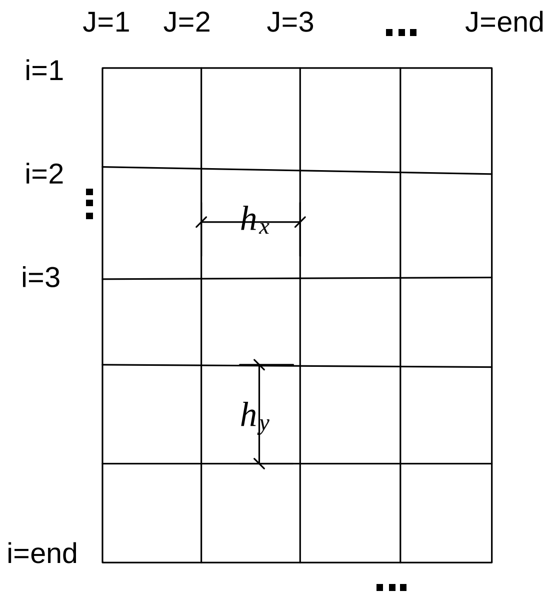

Figure 1: grid notation

Starting with centered difference scheme

$$-\left(\frac{U_{i,j-1}-2U_{i,j}+U_{i,j+1}}{h_x^2}+\frac{U_{i-1,j}-2U_{i,j}+U_{i+1,j}}{h_y^2}\right)=f_{i,j}$$

And solving for $U_{i,j}$ gives

$$\begin{aligned}2\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)U_{ij}+\frac{-U_{i,j-1}-U_{i,j+1}}{h_x^2}+\frac{-U_{i-1,j}-U_{i+1,j}}{h_y^2}&=f_{i,j}\\ 2\left(\frac{h_x^2+h_y^2}{h_x^2h_y^2}\right)U_{ij}+\frac{-h_y^2(U_{i,j-1}+U_{i,j+1})-h_x^2(U_{i-1,j}+U_{i+1,j})}{h_x^2h_y^2}&=f_{i,j}\\ \frac{2\left(h_x^2+h_y^2\right)}{h_x^2h_y^2}U_{ij}-\frac{h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})}{h_x^2h_y^2}&=f_{i,j}\end{aligned}$$

Hence

$$U_{i,j}=\frac{1}{2\left(h_x^2+h_y^2\right)}\left[h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})\right]+\frac{h_x^2h_y^2}{2\left(h_x^2+h_y^2\right)}f_{i,j}\qquad(1A)$$

Therefore, for node i,j the equation $-\nabla^2u=f(x,y)$ becomes

$$\begin{aligned}-\left(\frac{h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})-2\left(h_x^2+h_y^2\right)U_{i,j}}{h_x^2h_y^2}\right)&=f_{i,j}\\ -2\left(h_x^2+h_y^2\right)U_{i,j}+h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})&=-h_x^2h_y^2f_{i,j}\end{aligned}$$

The above formula reduces to the standard one when $h_x = h_y$

$$U_{i,j} = \frac{U_{i,j-1} + U_{i,j+1} + U_{i-1,j} + U_{i+1,j} + h^2 f_{i,j}}{4}$$

Therefore, the equation $-\nabla^2 u = f(x,y)$ becomes

$$-(-4U_{i,j} + U_{i,j-1} + U_{i,j+1} + U_{i-1,j} + U_{i+1,j}) = h^2 f_{i,j}$$

1.1 Derivations of formulas for Neumann boundary conditions

Below is the derivation of the discretization for the case when Neumann boundary conditions are used.

1.1.1 Left edge

Given a 2D grid, if there exists a Neumann boundary condition on an edge, for example, on the left edge, then this implies that $\frac{\partial u}{\partial x}$ in the normal direction to the edge is some function of y.

Assuming that $\frac{\partial u}{\partial x} = g(y)$ on the left edge is as shown in the following diagram

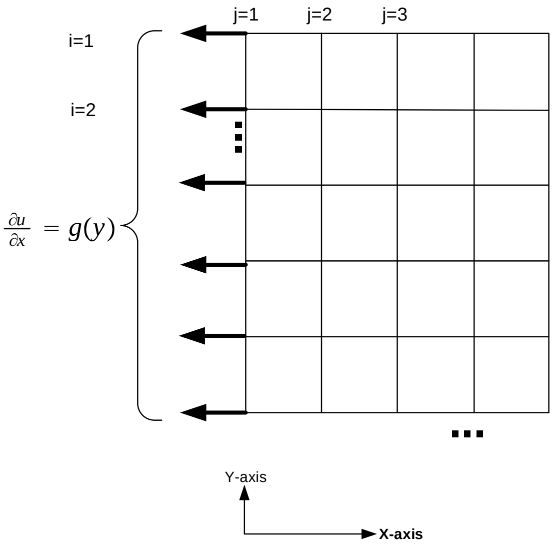

Figure 2: Left edge

The value of the unknown $u(x,y)$ is not given on the left boundary, only its derivative (in the normal direction to the edge, pointing outwards) is given. Therefore it is not possible to use the standard 5-point Laplacian on the first column $j = 1$, since that requires u be known on the left edge.

Therefore, when $j = 1$, then $U_{i,1}$ is not known (if this was a Dirichlet boundary conditions, then $U_{i,1}$ will be known).

Neumann boundary conditions are handled as follows. Taking the case of the left edge, an imaginary boundary located h_x to the left of the actual left edge is added as shown below

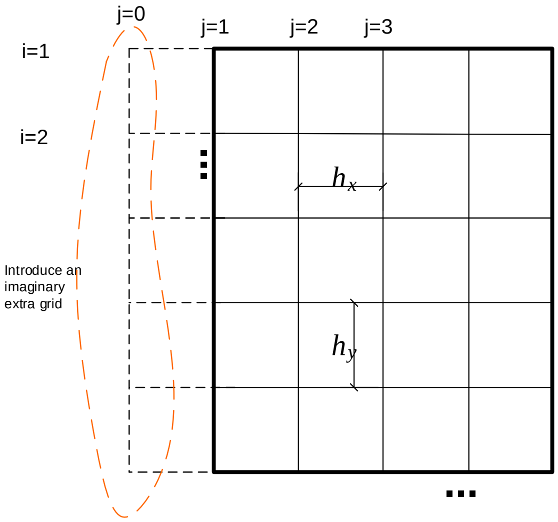

Figure 3: grid

Then $\frac{\partial u}{\partial x}$ is approximated on the line $j = 1$ by using $\frac{\partial u}{\partial x} \approx \frac{U_{i,2} - U_{i,0}}{2h_x}$. Therefore for $j = 1$ the following results

$$\frac{U_{i,2} - U_{i,0}}{2h_x} = g_{i,1} \tag{2}$$

From eq (1A) above, given below again

$$U_{i,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,j-1} + U_{i,j+1}) + h_x^2(U_{i-1,j} + U_{i+1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,j}$$

When $j = 1$ the above becomes

$$U_{i,1} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,0} + U_{i,2}) + h_x^2(U_{i-1,1} + U_{i+1,1})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,j} \tag{3}$$

But using Eq. (2) gives

$$U_{i,0} = U_{i,2} - 2h_x g_{i,1}$$

Substituting the above into Eq. (3) results in

$$U_{i,1} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,2} - 2h_x g_{i,1} + U_{i,2}) + h_x^2(U_{i-1,1} + U_{i+1,1})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,j} \quad (3A)$$

An expression for U on the left edge is obtained. This will be added to the set of the equations to solve for.

Hence, For Neumann boundary conditions, the process starts by looking for the edge which have this condition specified on it, and for each such edge a set of equations as the above are added to the current set of equations for the internal points.

Therefore, for node $(i, 1)$ the equation $-\nabla^2 u = f$ becomes

$$\frac{2\left(h_x^2 + h_y^2\right) U_{i,1} - h_y^2(2U_{i,2} - 2h_x g_{i,1}) - h_x^2(U_{i-1,1} + U_{i+1,1})}{h_x^2 h_y^2} = f_{i,j}$$

$$2\left(h_x^2 + h_y^2\right) U_{i,1} - h_y^2(2U_{i,2} - 2h_x g_{i,1}) - h_x^2(U_{i-1,1} + U_{i+1,1}) = h_x^2 h_y^2 f_{i,j}$$

The edge points which have Neumann boundary conditions are initialized the same way as the internal units before the iterative process begins.

Eq. (3A) above reduces to the standard one when $h_x = h_y$ giving

$$U_{i,1} = \frac{1}{4}\left(2U_{i,2} - 2h_x g_{i,1} + U_{i-1,1} + U_{i+1,1} + h^2 f_{i,j}\right)$$

Therefore, the equation $-\nabla^2 u = f$ for such a node becomes

$$4U_{i,1} - 2U_{i,2} - U_{i-1,1} - U_{i+1,1} = h^2 f_{i,j} - 2h_x g_{i,1}$$

The term $2h_x g_{i,1}$ is moved to the RHS since it is a known quantity and will go into the force vector since only unknowns are kept in the LHS.

1.1.2 Right edge

Now the case for right edge is considered. Assuming that $\frac{\partial u}{\partial x} = q(y)$ is given.

$\frac{\partial u}{\partial x}$ is approximated on the line $j = end$ by using $\frac{\partial u}{\partial x} \approx \frac{U_{i,end-1} - U_{i,end+1}}{2h_x}$, hence for the last column $(j = n)$ this results in

$$\frac{U_{i,n-1} - U_{i,n+1}}{2h_x} = q_{i,n} \quad (1)$$

From Eq. (1A) above, shown below again

$$U_{i,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,j-1} + U_{i,j+1}) + h_x^2(U_{i-1,j} + U_{i+1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,j}$$

For $j = n$ it becomes

$$U_{i,n} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,n-1} + U_{i,n+1}) + h_x^2(U_{i-1,n} + U_{i+1,n})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,n} \tag{2}$$

But from Eq. (1)

$$U_{i,n+1} = U_{i,n-1} - 2h_x q_{i,n}$$

Substituting the above into Eq. (2) results in

$$U_{i,n} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(2U_{i,n-1} - 2h_x q_{i,n}) + h_x^2(U_{i-1,n} + U_{i+1,n})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{i,n} \tag{3}$$

Therefore, for node (i, n) the equation $-\nabla^2 u = f$ becomes

$$\frac{2\left(h_x^2 + h_y^2\right) U_{i,n} - h_y^2(2U_{i,n-1} - 2h_x q_{i,n}) - h_x^2(U_{i-1,n} + U_{i+1,n})}{h_x^2 h_y^2} = f_{i,n}$$

Eq. (3) reduces to the standard one when $h_x = h_y$ giving

$$U_{i,n} = \frac{1}{4}\left(2U_{i,n-1} - 2h_x q_{i,n} + U_{i-1,n} + U_{i+1,n} + h^2 f_{i,n}\right)$$

Therefore, equation $-\nabla^2 u = f$ for such a node becomes

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

1.1.3 Top edge

Assuming $\frac{\partial u}{\partial y} = \beta(x)$ is given for the top edge. $\frac{\partial u}{\partial x}$ is approximated on the line $i = 1$ by using $\frac{\partial u}{\partial y} \approx \frac{U_{2,j} - U_{0,j}}{2h_y}$, resulting in

$$\frac{U_{2,j} - U_{0,j}}{2h_y} = \beta_{1,j} \tag{1}$$

Using Eq. (1A) above, shown below again

$$U_{i,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,j-1} + U_{i,j+1}) + h_x^2(U_{i-1,j} + U_{i+1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{1,j}$$

For the $i = 1$ it becomes

$$U_{1,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{1,j-1} + U_{1,j+1}) + h_x^2(U_{0,j} + U_{2,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{1,j} \tag{2}$$

But from Eq. (1)

$$U_{0,j} = U_{2,j} - 2h_y\ \beta_{1,j}$$

Substituting the above into Eq. (2) results in

$$U_{1,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{1,j-1} + U_{1,j+1}) + h_x^2(2U_{2,j} - 2h_y\ \beta_{1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{1,j} \quad (3)$$

Therefore, for node $(1, j)$ equation $-\nabla^2 u = f$ becomes

$$\frac{2\left(h_x^2 + h_y^2\right) U_{1,j} - h_y^2(U_{1,j-1} + U_{1,j+1}) - h_x^2(2U_{2,j} - 2h_y\ \beta_{1,j})}{h_x^2 h_y^2} = f_{1,j}$$

$$2\left(h_x^2 + h_y^2\right) U_{1,j} - h_y^2(U_{1,j-1} + U_{1,j+1}) - h_x^2(2U_{2,j} - 2h_y\ \beta_{1,j}) = h_x^2 h_y^2 f_{1,j}$$

Eq. (3) reduces to the standard one when $k = 0$ and when $h_x = h_y$ giving

$$U_{1,j} = \frac{U_{1,j-1} + U_{1,j+1} + 2U_{2,j} - 2h\ \beta_{1,j} + h^2 f_{1,j}}{4}$$

And equation $-\nabla^2 u = f$ reduces to

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

1.1.4 Bottom edge

Assuming $\frac{\partial u}{\partial y} = \alpha(x)$ is given on the bottom edge.

$\frac{\partial u}{\partial x}$ is approximated on the line $(i = n)$ by using $\frac{\partial u}{\partial y} \approx \frac{U_{n-1,j} - U_{n+1,j}}{2h_y}$ giving

$$\frac{U_{n-1,j} - U_{n+1,j}}{2h_y} = \alpha_{n,j} \quad (1)$$

From Eq. (1A) above, shown below again

$$U_{i,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{i,j-1} + U_{i,j+1}) + h_x^2(U_{i-1,j} + U_{i+1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{n,j}$$

For $i = n$ it becomes

$$U_{n,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{n,j-1} + U_{n,j+1}) + h_x^2(U_{n-1,j} + U_{n+1,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{n,j} \quad (2)$$

But from Eq. (1)

$$U_{n+1,j} = U_{n-1,j} - 2h_y\ \alpha_{n,j}$$

Substituting the above into Eq. (2) results in

$$U_{n,j} = \frac{1}{2\left(h_x^2 + h_y^2\right)}\left[h_y^2(U_{n,j-1} + U_{n,j+1}) + h_x^2(2U_{n-1,j} - 2h_y\ \alpha_{n,j})\right] + \frac{h_x^2 h_y^2}{2\left(h_x^2 + h_y^2\right)} f_{n,j} \quad (3)$$

Therefore, for node (n, j) equation $-\nabla^2 u = f$ becomes

$$\frac{\left(2\left(h_x^2 + h_y^2\right) - h_x^2 h_y^2 k^2\right) U_{n,j} - h_y^2(U_{n,j-1} + U_{n,j+1}) - h_x^2(2U_{n-1,j} - 2h_y\ \alpha_{n,j})}{h_x^2 h_y^2} = f_{n,j}$$

Eq. (3) reduces to the standard one when $k = 0$ and when $h_x = h_y$ giving

$$U_{n,j} = \frac{U_{n,j-1} + U_{n,j+1} + 2U_{n-1,j} - 2h\ \alpha_{n,j} + h^2 f_{n,j}}{4}$$

Therefore, equation $-\nabla^2 u = f$ reduces to

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

The following diagram shows a typical stencil for the edges under Neumann boundary conditions

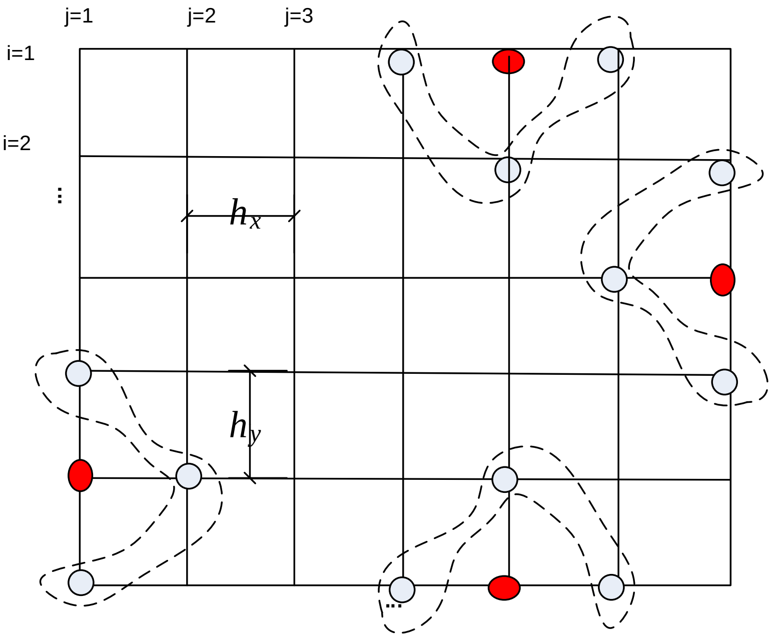

Figure 4: typical stencil for the edges under Neumann boundary conditions

2 Generating the solution equation for Jacobi iteration

From Eq. (1A) it was found that

$$U_{i,j}=\frac{1}{2\left(h_x^2+h_y^2\right)}\left[h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})\right]+\frac{h_x^2h_y^2}{2\left(h_x^2+h_y^2\right)}f_{i,j} \tag{1}$$

$-\nabla^2u=f$ is the following

$$-\left(\frac{h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})-2\left(h_x^2+h_y^2\right)U_{i,j}}{h_x^2h_y^2}\right)=f_{i,j}$$

Hence

$$\begin{aligned}U_{i,j}^{k+1}&=\frac{1}{2\left(h_x^2+h_y^2\right)}\left[h_y^2\left(U_{i,j-1}^k+U_{i,j+1}^k\right)+h_x^2\left(U_{i-1,j}^k+U_{i+1,j}^k\right)\right]+\frac{h_x^2h_y^2}{2\left(h_x^2+h_y^2\right)}f_{i,j}\\&=\frac{h_y^2\left(U_{i,j-1}^k+U_{i,j+1}^k\right)+h_x^2\left(U_{i-1,j}^k+U_{i+1,j}^k\right)+h_x^2h_y^2f_{i,j}}{2\left(h_x^2+h_y^2\right)}\end{aligned}$$

The residue is

$$\begin{aligned}r_{ij}&=f_{ij}+\nabla^2u\\&=f_{ij}+\left(\frac{h_y^2(U_{i,j-1}+U_{i,j+1})+h_x^2(U_{i-1,j}+U_{i+1,j})-2\left(h_x^2+h_y^2\right)U_{i,j}}{h_x^2h_y^2}\right)\end{aligned} \tag{2}$$

Eq. (2) reduces to the following when $h_x=h_y$ Where

$$r_{ij}=f_{ij}-\frac{4U_{ij}+U_{i,j-1}+U_{i,j+1}+U_{i-1,j}+U_{i+1,j}}{h^2}$$

3 Generating the solution equation for Gauss-Seidel

When iteration runs from top to bottom, left to right, the following can be used

$$U_{i,j}^k=\frac{h_y^2\left(U_{i,j-1}^k+U_{i,j+1}^k\right)+h_x^2\left(U_{i-1,j}^k+U_{i+1,j}^k\right)+h_x^2h_y^2f_{i,j}}{2\left(h_x^2+h_y^2\right)}$$

Some books write the above as

$$U_{i,j}^{k+1}=\frac{h_y^2\left(U_{i,j-1}^{k+1}+U_{i,j+1}^k\right)+h_x^2\left(U_{i-1,j}^{k+1}+U_{i+1,j}^k\right)+h_x^2h_y^2f_{i,j}}{2\left(h_x^2+h_y^2\right)}$$

It leads to the same result in software, as long as iterations is run left to right, top to down, using the index notation given here.

Another option is to run the iterations bottom to top, right to left.

As long as one is consistent, same result will be obtained. The idea is to use the most recent updated cell in the matrix all the time.

The residue is

$$\begin{aligned} r_{ij} &= f_{ij} + \nabla^2 u \\ &= f_{ij} + \left(\frac{h_y^2(U_{i,j-1} + U_{i,j+1}) + h_x^2(U_{i-1,j} + U_{i+1,j}) - 2\left(h_x^2 + h_y^2\right) U_{i,j}}{h_x^2 h_y^2} \right) \end{aligned} \tag{2}$$

It is important in the above, to find the residual before updating the cell. In the code, the following should be used

$$\begin{aligned} r_{ij} &= f_{ij} + \left(\frac{h_y^2\left(U_{i,j-1}^k + U_{i,j+1}^k\right) + h_x^2\left(U_{i-1,j}^k + U_{i+1,j}^k\right) - 2\left(h_x^2 + h_y^2\right) U_{i,j}^k}{h_x^2 h_y^2} \right) \\ U_{i,j}^k &= \frac{h_y^2\left(U_{i,j-1}^k + U_{i,j+1}^k\right) + h_x^2\left(U_{i-1,j}^k + U_{i+1,j}^k\right) + h_x^2 h_y^2 f_{i,j}}{2\left(h_x^2 + h_y^2\right)} \end{aligned}$$

4 References

1. Applied Mathematics by David Logan, chapter 8
2. `http://en.wikipedia.org/wiki/Helmholtz/equation`