6 Finite difference methods for two dimensional second order elliptic problems

There are many important applications of elliptic partial differential equations. Below are some examples:

- Laplace equation.

$$u_{xx} + u_{yy} = 0. \tag{6.1}$$

- Poisson equation.

$$u_{xx} + u_{yy} = f. \tag{6.2}$$

  The solution to a Laplace or Poisson equation sometimes is called a potential equation. This is because for a vector field $\mathbf{v}$ that is conservative meaning that $curl(\mathbf{v}) = \mathbf{0}$, its potential function u satisfies $u_{xx} + u_{yy} = \nabla \cdot \mathbf{v}$, where $\nabla = [\,\frac{\partial}{\partial x},\ \frac{\partial}{\partial y}\,]^T$ is the gradient operator.

- Helmholtz equation.

$$u_{xx} + u_{yy} + \lambda u = f. \tag{6.3}$$

  If $\lambda < 0$, it is called generalized Helmholtz equation and it is easy to solve. If $\lambda > 0$ is big, then the problem is hard to solve.

- General self-adjoint elliptic PDEs

$$\nabla \cdot a(x,y)\nabla u(x,y) + \lambda(x,y)u = f(x,y), \tag{6.4}$$

$$\text{or} \quad (au_x)_x + (au_y)_y + \lambda(x,y)u = f(x,y), \tag{6.5}$$

  We should assume that $a(x,y)$ does not change the sign in the solution domain.

- General elliptic PDEs (diffusion and advection equations)

$$a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} + d(x,y)u_x + e(x,y)u_y + g(x,y)u(x,y) = f(x,y)$$

  It can be re-written as

$$\nabla \cdot a(x,y)\nabla u(x,y) + \mathbf{w}(x,y) \cdot \nabla u + c(x,y)u = f(x,y) \tag{6.6}$$

  after some transformation, where $\mathbf{w}(x,y)$ is a vector.

Below are some examples of non-linear elliptic PDEs.

- Diffusion and reaction equation.

$$\nabla \cdot (a(x,y)\nabla u(x,y)) = f(u). \tag{6.7}$$

The $\nabla \cdot a(x,y)\nabla u(x,y)$ is called the diffusion term, the non-linear term $f(u)$ is called the reaction term. If $a(x,y) \equiv 1$, it is also called a non-linear Poisson equation.

- p-Laplacian equation

$$div\left(|Du|^{p-2}Du\right) = 0, \quad p \geq 2. \tag{6.8}$$

- Minimal surface equation

$$div\left(\frac{Du}{\sqrt{1+|Du|^2}}\right) = 0. \tag{6.9}$$

Note that an elliptic PDE, $P(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})u = 0$, can be regarded as the steady state solution of a corresponding parabolic equation $u_t = P(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})u$.

If a linear PDE is defined on a rectangle domain, then we can use a finite difference approximation dimension by dimension both for the equation and the boundary condition. The most difficult part of a finite difference method is how to solve the resulting linear system of equation efficiently.

6.1 Boundary conditions and compatibility condition.

Assume a second order elliptic PDE is defined on a domain Ω. Let $\partial\Omega$ be the boundary of Ω and $\mathbf{n}$ is the unit normal direction (positive according to the right side rule), see Fig. 8.

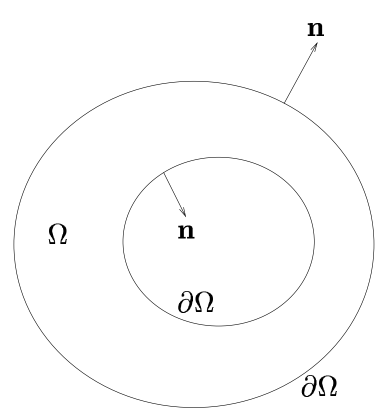

Figure 8: A diagram of a two dimensional domain Ω, its boundary $\partial\Omega$ and its unit normal direction.

Some commonly used boundary conditions are listed below:

- Dirichlet boundary condition: the solution $u(x,y)|_{\partial\Omega} = u_0(x,y)$ is given along the boundary.
- Neumann or flux boundary condition: the normal derivative $\frac{\partial u}{\partial \mathbf{n}} = \nabla u \cdot \mathbf{n} = u_n = u_x n_x + u_y n_y = g(x,y)$ is given along the boundary, where $\mathbf{n} = (n_x, n_y)$ is the unit normal direction.
- Robin or mixed boundary condition: $\left(\alpha(x,y)u(x,y) + \beta(x,y)\frac{\partial u}{\partial \mathbf{n}}\right)\Big|_{\partial\Omega} = \gamma(x,y)$ is given along the boundary $\partial\Omega$.
- For some domains, for example, we can define a periodic boundary condition. For example, if $\Omega = [a,\, b] \times [c,\, d]$, the periodic boundary condition in the x- direction is $u(a,y) = u(b,y)$, the periodic boundary condition in the y- direction is $u(x,c) = u(x,d)$.

We can have different boundary condition on different parts of the boundary. For example, for a channel flow in a domain $[a,b] \times [c,d]$, where $b - a >> d - c$, it is reasonable to assume that the flux boundary condition at $x = a$ and $x = b$, but non-slip $\mathbf{u} = 0$ along the boundaries $y = c$ and $y = d$.

For a Poisson equation with pure Neumann boundary condition, the solution does not exist unless the *compatibility* condition is satisfied. If u is a solution to the following problem

$$\Delta u = f(x,y), \quad (x,y) \in \Omega, \quad \frac{\partial u}{\partial \mathbf{n}}|_{\partial\Omega} = g(x,y).$$

Integrate and use the Green's theorem, we can get

$$\begin{aligned} \iint_\Omega \Delta u dx dy &= \iint_\Omega f(x,y) dx dy \\ \oint_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} ds &= \iint_\Omega f(x,y) dx dy \\ \oint_{\partial\Omega} g(x,y) ds &= \iint_\Omega f(x,y) dx dy. \end{aligned}$$

If the compatibility condition is satisfied and $\partial\Omega$ is smooth, then the solution do exist but it is not unique. If $u(x,y)$ is a solution, then $u(x,y) + C$ is also a solution with arbitrary constant C. In this case, we can specify the solution at a particular point, for example, $u(x_0, y_0) = 0$ to make the solution well defined.

6.2 The central finite difference method with five point stencil for Poisson equation.

Consider the Poisson equation

$$u_{xx} + u_{yy} = f(x,y), \quad (x,y) \in \Omega = (a,b) \times (c,d), \tag{6.10}$$

$$u(x,y)|_{\partial\Omega} = u_0(x,y), \qquad \text{Dirichlet BC.} \tag{6.11}$$

If $f \in L^2(\Omega)$, then the solution exists and it is unique. Analytic solution is rarely available. Now we discuss how to use the finite difference equation to solve the Poisson equation.

- Step 1: Generate a grid. A uniform Cartesian grid can be used:

$$x_i = a + ih_x, \quad i = 0, 1, 2, \cdots m, \quad h_x = \frac{b-a}{m}, \tag{6.12}$$

$$y_j = c + jh_y, \quad j = 0, 1, 2, \cdots n, \quad h_y = \frac{d-c}{n}. \tag{6.13}$$

  We want to find an approximate solution U_{ij} to the exact solution at all the grid points (x_i, y_j) where $u(x_i, y_j)$ is unknown. So there are $(m-1)(n-1)$ unknown for Dirichlet boundary condition.

- Step 2: Substitute the partial derivatives with a finite difference formula in terms of the function values at grid points to get.

$$\frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{(h_x)^2} + \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{(h_y)^2}$$
$$= f_{ij} + T_{ij}, \quad i = 1, \cdots m-1, \quad j = 1, \cdots n-1,$$

  where $f_{ij} = f(x_i, y_j)$. The local truncation error satisfies

$$T_{ij} \sim \frac{(h_x)^2}{12}\frac{\partial^4 u}{\partial x^4} + \frac{(h_y)^2}{12}\frac{\partial^4 u}{\partial y^4}. \tag{6.14}$$

  Define

$$h = \max\{\, h_x, \; h_y \,\} \tag{6.15}$$

  The finite difference discretization is consistent if

$$\lim_{h \to 0} \|\mathbf{T}\| = 0. \tag{6.16}$$

  Therefore the discretization is consistent and second order accurate.

  If we remove the error term in the equation above, and replace the exact solution $u(x_i, y_j)$ with the approximate solution U_{ij} which is the solution of the linear system of equations

$$\frac{U_{i-1,j} + U_{i+1,j}}{(h_x)^2} + \frac{U_{i,j-1} + U_{i,j+1}}{(h_y)^2} - \left(\frac{2}{(h_x)^2} + \frac{2}{(h_y)^2} \right) U_{ij} = f_{ij} \tag{6.17}$$

  The finite difference scheme at a grid point (x_i, y_j) involves five grid points, east, north, west, south, and the center. The center is called the master grid point.

- Solve the linear system of equations to get an approximate solution at grid points (how?).

- Error analysis, implementation, visualization etc.

6.3 Matrix-vector form of the finite difference equations.

Generally, if one wants to use a direct method such as Gaussian elimination method or sparse matrix techniques, then one needs to find out the matrix structure. If one use an iterative method, such as Jacobi, Gauss Seidel, SOR(ω) methods, then it may be not necessarily to have the matrix and vector form.

In the matrix vector form $A\mathbf{U} = \mathbf{F}$, the unknown is a one dimensional array. For the two dimensional Poisson equations, the unknowns U_{ij} are a two dimensional array. Therefore we need to order it to get a one dimensional array. We also need to order the finite difference equations. It is common practice that we use the same ordering for the equations and for the unknowns.

There are two commonly used ordering. One is called the *natural ordering* that fits sequential computers. The other one is called the *red and black ordering* that fits parallel computers.

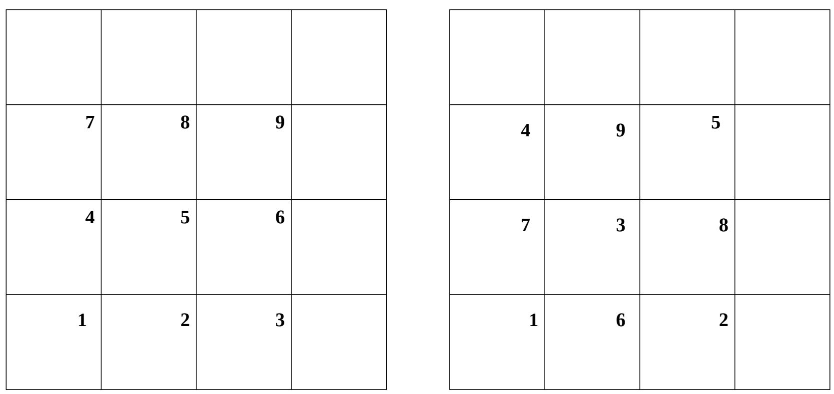

Figure 9: The natural ordering (left) and the red-black ordering (right).

The natural row ordering.

In the natural row ordering, we order the unknowns/equations row-wise, therefore the k-th equation corresponding to (i, j) with the following relation

$$k = i + (m-1)(j-1), \quad i = 1, 2, \cdots, m-1, \quad j = 1, 2, \cdots, n-1. \tag{6.18}$$

We use the following example to verify the matrix-vector form of the finite difference equations.

Assume that $h_x = h_y = h$, $m = n = 4$, so we will have nine equations and nine unknowns. The coefficient matrix is 9 by 9! To write down the matrix-vector form, we use a one-dimensional array $\mathbf{x}$ to express the unknown U_{ij}.

$$\begin{aligned} &x_1 = U_{11}, \quad x_2 = U_{21}, \quad x_3 = U_{31}, \quad x_4 = U_{12}, \quad x_5 = U_{22}, \\ &x_6 = U_{32}, \quad x_7 = U_{13}, \quad x_8 = U_{23}, \quad x_9 = U_{33}. \end{aligned} \tag{6.19}$$

If we order the equations the same way as we order the unknowns, then the nine equations from the standard central finite difference scheme using the five point stencil are

$$\begin{aligned}
\frac{1}{h^2}\left(-4x_1 + x_2 + x_4\right) &= f_{11} - \frac{u_{01} + u_{10}}{h^2}, \\
\frac{1}{h^2}\left(x_1 - 4x_2 + x_3 + x_5\right) &= f_{21} - \frac{u_{20}}{h^2} \\
\frac{1}{h^2}\left(x_2 - 4x_3 + x_6\right) &= f_{31} - \frac{u_{30} + u_{41}}{h^2} \\
\frac{1}{h^2}\left(x_1 - 4x_4 + x_5 + x_7\right) &= f_{12} - \frac{u_{02}}{h^2} \\
\frac{1}{h^2}\left(x_2 + x_4 - 4x_5 + x_6 + x_8\right) &= f_{22} \\
\frac{1}{h^2}\left(x_3 + x_5 - 4x_6 + x_9\right) &= f_{32} - \frac{u_{42}}{h^2} \\
\frac{1}{h^2}\left(x_4 - 4x_7 + x_8\right) &= f_{13} - \frac{u_{03} + u_{14}}{h^2} \\
\frac{1}{h^2}\left(x_5 + x_7 - 4x_8 + x_9\right) &= f_{23} - \frac{u_{24}}{h^2} \\
\frac{1}{h^2}\left(x_6 + x_8 - 4x_9\right) &= f_{33} - \frac{u_{34} + u_{43}}{h^2}.
\end{aligned}$$

Now we can write down the coefficient matrix easily. It is *block tridiagonal* and has the following form:

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & 0 \\ I & B & I \\ 0 & I & B \end{bmatrix} \tag{6.20}$$

where I is a 3×3 identity matrix:

$$B = \begin{bmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{bmatrix}$$

For a general n by n grid, we will have

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & & I & B \end{bmatrix}, \qquad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -4 \end{bmatrix}.$$

Note that $-A$ is a symmetric positive definite matrix and it is weakly diagonally dominant. Therefore A is non-singular and there is a unique solution.

The matrix-vector form is useful to understand the structure of the linear system of equations, and it may be necessary if a direct method (such as Gaussian elimination) or sparse matrix techniques are used for solving the system. However, it is more convenient sometimes to use the two parameters system (i, j), especially if an iterative method is used to solve the system. It is more intuitive and useful to visualize the data using two index system.

The eigenvalues and eigenvectors of A can be indexed by two parameters p and k corresponding to wave numbers in the x and y directions. The (p, k)-th eigenvector $u^{p,k}$ has n^2 elements for a n by n matrix of the form above:

$$u_{ij}^{p,k} = \sin(p\pi ih)\ \sin(k\pi jh), \quad i, j = 1, 2, \cdots n \tag{6.21}$$

for $p, k = 1, 2, \cdots n$. The corresponding eigenvalues are

$$\lambda^{p,k} = \frac{2}{h^2}\Big(\cos(p\pi h) - 1) + \cos(k\pi h) - 1)\Big). \tag{6.22}$$

The least dominant eigenvalue (the smallest in the magnitude) is

$$\lambda^{1,1} = -2\pi + O(h^2). \tag{6.23}$$

The dominant eigenvalue (the largest in the magnitude) is

$$\lambda^{n/2,n/2} \sim -\frac{4}{h^2}. \tag{6.24}$$

Therefore we have the following estimates:

$$\begin{aligned} \|A\|_2 \sim \max|\lambda^{p,k}| = \frac{4}{h^2}, \qquad \|A^{-1}\|_2 = \frac{1}{\min|\lambda^{p,k}|} \sim \frac{1}{2\pi}, \\ cond_2(A) = \|A\|_2\|A^{-1}\|_2 \sim \frac{2}{\pi h^2} = O(n^2). \end{aligned} \tag{6.25}$$

Since the condition number is considered to be large, we should use double precision to reduce the effect of round off errors.

7 Maximum Principle and Error Analysis

Given an elliptic operator

$$L = a\frac{\partial^2}{\partial x^2} + 2b\frac{\partial^2}{\partial x\partial y} + c\frac{\partial^2}{\partial y^2}, \quad b^2 - ac < 0, \quad \text{for} \quad (x,y)\in\Omega,\ a>0,\ c>0.$$

The maximum principle is the following theorem.

Theorem 7.1 *If $u(x,y)\in C^3(\Omega)$ satisfies $Lu(x,y)\geq 0$ is a bounded domain Ω, then $u(x,u)$ has its maximum on the boundary.*

Proof: If the theorem is not true, then there is an interior point $(x_0,y_0)\in\Omega$ such that $u(x_0,y_0)\geq u(x,u)$ for all $(x,y)\in\Omega$. Therefore, form the necessary condition of the local extrema, we have

$$\frac{\partial u}{\partial x}(x_0,y_0)=0, \quad \frac{\partial u}{\partial y}(x_0,y_0)=0.$$

Since (x_0,y_0) is not on the boundary and $u(x,y)$ is continuous, there is a neighborhood of (x_0,y_0) within the domain Ω that we can use the Taylor expansion:

$$u(x_0+\Delta x, y_0+\Delta y) = u(x_0,y_0) + \frac{1}{2}\left((\Delta x)^2 u_{xx}^0 + 2\Delta x\Delta y u_{xy}^0 + (\Delta y)^2 u_{yy}^0\right) + O((\Delta x)^3, (\Delta y)^3)$$

where the superscript of 0 indicating that the functions are evaluated at (x_0,y_0), for example, $u_{xx}^0 = \frac{\partial^2 u}{\partial x^2}(x_0,y_0)$. Since we have $u(x_0+\Delta x, y_0+\Delta y)\leq u(x_0,y_0)$ for all Δx and Δy that are small enough. We have

$$\frac{1}{2}\left((\Delta x)^2 u_{xx}^0 + 2\Delta x\Delta y u_{xy}^0 + (\Delta y)^2 u_{yy}^0\right)\leq 0. \tag{7.1}$$

On the other hand, from the given condition we know that

$$Lu^0 = a^0 u_{xx}^0 + 2b^0 u_{xy}^0 + c^0 u_{yy}^0 \geq 0. \tag{7.2}$$

where $a^0 = a(x_0,y_0)$ and so forth. In order to match the Taylor expansion to get an contradiction, we re-write the inequality above as

$$\left(\sqrt{\frac{a^0}{M}}\right)^2 u_{xx}^0 + 2\sqrt{\frac{a^0}{M}}\frac{b^0}{\sqrt{a^0 M}}u_{xy}^0 + \left(\frac{b^0}{\sqrt{a^0M}}\right)^2 u_{yy}^0 + \frac{u_{yy}^0}{M}\left(c^0 - \frac{(b^0)^2}{a_0}\right)\geq 0, \tag{7.3}$$

where $M>0$ is a constant. The role of M is to make some choices of Δx and Δy that are small enough.

Next we take

$$\Delta x = \sqrt{\frac{a^0}{M}}, \qquad \Delta y = \frac{b^0}{\sqrt{a^0M}}.$$

From (7.1), we know that

$$\frac{a^0}{M}u_{xx}^0 + \frac{2b^0}{M}u_{xy}^0 + \frac{b^0}{a^0 M}u_{yy}^0 \le 0. \tag{7.4}$$

Now we take

$$\Delta x = 0, \qquad \Delta y = \sqrt{\left(c^0 - \frac{(b^0)^2}{a^0}\right)/M}.$$

From (7.1) again, we know that

$$(\Delta y)^2 u_{yy}^0 = \frac{1}{M}\left(c^0 - \frac{(b^0)^2}{a^0}\right) u_{yy}^0 \le 0. \tag{7.5}$$

From (7.4) and (7.11), we know that the left hand side of (7.3) should not be positive which contradicts the the condition

$$Lu^0 = a^0 u_{xx}^0 + 2b^0 u_{xy}^0 + c^0 u_{yy}^0 \ge 0.$$

This completes the proof. □

Similarly, if $Lu \le 0$, then u has its minimum value on the boundary of Ω.

For general elliptic equations with no cross derivative term u_{xy}, the maximum principle is the following. Let

$$Lu = au_{xx} + 2bu_{xy} + cu_{yy} + d_1 u_x + d_2 u_y + eu = 0, \quad (x,y) \in \Omega,$$
$$b^2 - ac < 0, \quad a > 0,\ c > 0, \quad e \le 0,$$

where Ω is a bounded domain. Then $u(x,y)$ can not have a positive local maximum or a negative local minimum in the interior of Ω. This can be easily proved using Theorem 7.1.

7.1 Discrete maximum principle.

Theorem 7.2 *Given a grid function U_{ij}, $i = 0, 1, \cdots, m$, $j = 0, 1, 2, \cdots, n$. If the discrete Laplacian operator (using the central five-point stencil) satisfies*

$$\begin{aligned} &\Delta_h U_{ij} = \frac{U_{i-1j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} \ge 0, \\ &i = 1, 2, \cdots m-1, \qquad j = 1, 2, \cdots n-1. \end{aligned} \tag{7.6}$$

Then U_{ij} attains its maximum on the boundary. If $\Delta_h U_{ij} \le 0$, then U_{ij} attains its minimum on the boundary.

Proof: Assume the theorem is not true, then U_{ij} has its maximum at an interior grid point (i_0, j_0). We have $U_{i_0,j_0} \ge U_{i,j}$ for all i and j's. Therefore we have

$$U_{i_0,j_0} \ge \frac{1}{4}\left(U_{i_0-1j_0} + U_{i_0+1,j_0} + U_{i_0,j_0-1} + U_{i_0,j_0+1}\right).$$

On the other hand, from the condition $\Delta_h U_{ij} \geq 0$, we know that

$$U_{i_0,j_0} \leq \frac{1}{4}\left(U_{i_0-1j_0} + U_{i_0+1,j_0} + U_{i_0,j_0-1} + U_{i_0,j_0+1}\right).$$

This contradicts the inequality above except that all U_{ij} at the four neighbors of (i_0, j_0) have the same value as $U(i_0, j_0)$. So U_{i_0-1,j_0} is also a maximum. If we apply the same arguments several times, then we will reach the boundary and know that U_{0,j_0} is also a maximum. From the proof we can see that, if U_{ij} has its maximum in interior, then U_{ij} is a constant.

If $\Delta_h U_{ij} \leq 0$, then we can consider $-U_{ij}$ which gives us the conclusion. □

7.2 Error estimates of the FD method for Poisson equations.

Lemma 7.1 *Let U_{ij} be a grid function that satisfies*

$$\Delta_h U_{ij} = \frac{U_{i-1j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} = f_{ij}, \quad i,j = 0, 1, \cdots, n \tag{7.7}$$

with homogeneous boundary condition. Then

$$\|\mathbf{U}\|_\infty = \max_{0\leq i,j\leq n} |U_{ij}| \leq \frac{1}{8} \max_{1\leq i,j\leq n} |\Delta_h U_{ij}| = \frac{1}{8} \max_{0\leq i,j\leq n} |f_{ij}|. \tag{7.8}$$

Proof: Define a grid function

$$w_{ij} = \frac{1}{4}\left((x_i - \frac{1}{2})^2 + (y_j - \frac{1}{2})^2\right) \tag{7.9}$$

where

$$x_i = -1 + ih, \quad y_j = -1 + jh, \quad i,j = 0, 1, \cdots n, \ h = \frac{2}{n},$$

corresponding to the continuous function $w(x) = \frac{1}{4}\left((x - \frac{1}{2})^2 + (y - \frac{1}{2})^2\right)$. We have

$$\Delta_h w_{ij} = \left. (w_{xx} + w_{yy})\right|_{(x_i,y_j)} + \frac{h^2}{12}\left(\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4}\right)_{(x_i^*,y_j^*)} = 1, \tag{7.10}$$

where (x_i^*, y_j^*) is some point near (x_i, y_j). Therefore we have

$$\begin{aligned} \Delta_h(U_{ij} - \|f\|_\infty w_{ij}) &= \Delta_h U_{ij} - \|f\|_\infty \leq 0, \\ \Delta_h(U_{ij} + \|f\|_\infty w_{ij}) &= \Delta_h U_{ij} + \|f\|_\infty \geq 0. \end{aligned} \tag{7.11}$$

From the discrete maximum principle, we know that $U_{ij} + \|f\|_\infty w_{ij}$ has its maximum on the boundary while $U_{ij} - \|f\|_\infty w_{ij}$ has its minimum on the boundary. That is

$$-\|f\|_\infty \|w_{ij}\|_{\partial\Omega} \leq U_{ij} - \|f\|_\infty w_{ij} \leq U_{ij},$$

$$\text{and} \qquad U_{ij} \leq U_{ij} + \|f\|_\infty w_{ij} \leq \|f\|_\infty \|w_{ij}\|_{\partial\Omega},$$

since U_{ij} is zero on the boundary and $\|f\|_\infty w_{ij} \geq 0$. It is easy to check that

$$\|w_{ij}\|_{\partial\Omega} = \frac{1}{8},$$

therefore we get

$$-\frac{1}{8}\|f\|_\infty \leq U_{ij} \leq \frac{1}{8}\|f\|_\infty. \tag{7.12}$$

That completes the proof of the lemma. □

Theorem 7.3 *Let U_{ij} be the solution of the finite difference equations using the standard central fine point stencil for Poisson equation with a Dirichlet boundary condition. Then the global error is second order accurate and satisfies:*

$$\|\mathbf{E}\|_\infty = \|\mathbf{U} - \mathbf{u}\|_\infty = \max_{ij} |U_{ij} - u(x_i, y_j)| \leq \frac{h^2}{96}\left(\max|u_{xxxx}| + \max|u_{yyyy}|\right). \tag{7.13}$$

Proof: We know that

$$\Delta_h U_{ij} = f_{ij} + T_{ij}, \qquad \Delta_h E_{ij} = T_{ij},$$

where T_{ij} is the local truncation error at (x_i, y_j) and satisfies

$$|T_{ij}| \leq \frac{h^2}{12}\left(\max|u_{xxxx}| + \max|u_{yyyy}|\right).$$

Therefore from Lemma **??**, we have

$$\|\mathbf{E}\|_\infty \leq \frac{1}{8}\|\mathbf{T}\|_\infty \leq \frac{h^2}{96}\left(\max|u_{xxxx}| + \max|u_{yyyy}|\right).$$

7.3 Finite difference discretization for general second order elliptic PDE using dimension by dimension.

If the domain of the interest is a rectangle $[a, b] \times [c, d]$ and there is no cross derivative term u_{xy} in the equation, then we can discretize the PDE dimension by dimension. We verify this for the example below. Consider

$$\nabla \cdot (p(x, y)\nabla u) - q(x, y)u = f(x, y), \quad \text{or} \quad (pu_x)_x + (pu_y)_y - qu = f.$$

with Dirichlet boundary condition at $x = b$, $y = c$, and $y = d$, but a Neumann boundary condition $u_x = g(y)$ along $x = a$.

If we use a uniform Cartesian grid

$$x_i = a + ih_x, \quad i = 0, 1, \cdots m, \quad h_x = \frac{b-a}{m}$$
$$y_j = c + jh_y, \quad j = 0, 1, \cdots n, \quad h_y = \frac{d-c}{n}$$

If we discretize the PDE dimension by dimension, the finite difference equation then is

$$\begin{aligned} &\frac{p_{i+\frac{1}{2},j}U_{i+1,j} - (p_{i+\frac{1}{2},j} + p_{i-\frac{1}{2},j})U_{ij} + p_{i-\frac{1}{2},j}U_{i-1,j}}{(h_x)^2} \\ &\quad + \frac{p_{i,j+\frac{1}{2}}U_{i,j+1} - (p_{i,j+\frac{1}{2}} + p_{i,j-\frac{1}{2}})U_{ij} + p_{i,j-\frac{1}{2}}U_{i,j-1}}{(h_y)^2} - q_{ij}U_{ij} = f_{ij} \end{aligned} \tag{7.14}$$

for $i = 1, 2, \cdots, m-1$ and $j = 1, 2, \cdots, n-1$, where $p_{i\pm\frac{1}{2},j} = p(x_i \pm h_x/2, y_j)$ and so forth.

For the indices $i = 0, j = 1, 2, \cdots, n-1$, we use the ghost point method to deal with the Neumann boundary condition. Using the central finite difference scheme for the flux boundary condition

$$\frac{U_{1,j} - U_{-1,j}}{2h_x} = g(y_j), \quad \text{or} \quad U_{-1,j} = U_{1,j} - 2h_x\, g(y_j), \quad j = 1, 2, \cdots, n-1.$$

Plugging this into the finite difference equation at $(0, j)$, we get

$$\begin{aligned} &\frac{(p_{-\frac{1}{2},j} + p_{\frac{1}{2},j})U_{1,j} - (p_{\frac{1}{2},j} + p_{-\frac{1}{2},j})U_{0j}}{(h_x)^2} \\ &\quad + \frac{p_{0,j+\frac{1}{2}}U_{0,j+1} - (p_{0,j+\frac{1}{2}} + p_{0,j-\frac{1}{2}})U_{0j} + p_{0,j-\frac{1}{2}}U_{0,j-1}}{(h_y)^2} \\ &\quad - q_{0j}U_{0j} = f_{0j} + \frac{2\,p_{-\frac{1}{2},j}\, g(y_j)}{h_x}. \end{aligned} \tag{7.15}$$

For general second order elliptic PDE with no cross derivative term u_{xy}:

$$\nabla \cdot (p(x,y)\nabla u) + \mathbf{w} \cdot \nabla u - q(x,y)u = f(x,y),$$

we can still use the central finite difference scheme of $|w| << 1/h$, otherwise, we should use the upwinding scheme to deal with the advection term.

Finite difference formula for cross derivative u_{xy}.

If there is a cross derivative term u_{xy}, we can not simply use the method of dimension by dimension. However, it is easy to derive a centered finite difference scheme for u_{xy}:

$$\begin{aligned} u_{xy} &\sim \delta_x\delta_y u = \delta_x \frac{u(x, y+h_y) - u(x, y-h_y)}{2h_y} \\ &= \frac{u(x-h_x, y-h_y) + u(x+h_x, y+h_y) - u(x+h_x, y-h_y) - u(x-h_x, y+h_y)}{4h_xh_y}. \end{aligned} \tag{7.16}$$

Using Taylor expansion at (x, y), we can show the finite difference formula is consistent and it it second order accurate. The finite difference formula for the PDE that involves a cross derivative term has nine grid points involved.

The resulting linear system of equations for PDEs with a cross derivative term is more difficult to solve because it is not symmetric, and there is no diagonally dominance any more, and there is no upwinding scheme to deal with the cross derivative term.

7.4 Solving the resulting linear system of equations.

The linear system of equations resulted from a finite finite difference discretization is very large for two dimensional problems. For example, using an $n \times n$ grid for an elliptic PDE, the linear system has about n^2 equations. The coefficient matrix has the size of $n^2 \times n^2$. When we take $n = 100$, a modest number, the size of the matrix would be $10^4 \times 10^4$ which can not be stored in most of computers if double precision is used. However, the matrix is very sparse, for a self-adjoint elliptic PDE, the non-zero entries are about $5n^2$. Therefore usually iterative methods or sparse matrix techniques are used. Typically, for an elliptic PDE defined on a rectangle domain (or a disk), several methods listed below are used frequently

- Fast Poisson solvers such as the FFT, cyclic reduction, see []. Usually the implementation is not so easy. It is recommended to use software packages, for example, the Fishpack in Fortran which is free on the Netlib.
- Multigrid solver, for example, MGD9V using a structured grid, for example, a nine-point stencil which includes the five-point stencil as a special case; AMG (algebraic multi-grid solver) if the coefficient Matrix is an M-matrix.
- Sparse matrix techniques. It was quite popular before the multigrid method was developed.
- Simple iterative method such as Jacobi, Gauss-Seidel, SOR(ω). They are easy to implement and fewer restrictions on the coefficient matrix. But they are generally slow in convergence.
- Other iterative method such as CG or PCG (conjugate gradient method with preconditioning), GMRES (generalized minimized residual), and BICG methods for non-symmetric system of equations.

One of the most important advantage of an iterative method is that it only needs the matrix-vector multiplication. So zero entries play no role. In the implementation, we use the component form instead of manipulating the matrix and vector. In other words, the equations are used directly.

7.5 Basic stationary iterative methods.

Give a linear system of equation $A\mathbf{x} = b$, $det(A) \neq 0$. Assume that we can write A as $A = M - N$ where M is an invertible matrix. Then we have $(M - N)\mathbf{x} = b$ or $M\mathbf{x} = N\mathbf{x} + b$ or $\mathbf{x} = M^{-1}N\mathbf{x} + M^{-1}b$. We can form an iteration starting from an initial guess $\mathbf{x}^0$

$$\mathbf{x}^{k+1} = M^{-1}N\mathbf{x}^k + M^{-1}b, \quad k = 1, 2, \cdots \tag{7.17}$$

Such iteration may converge or diverge depending the spectral radius of $\rho(M^{-1}) = \max|\lambda_i(M^{-1})|$. Since $T = M^{-1}N$ is a constant matrix, such a method is called a stationary iterative method.

7.5.1 The Jacobi iterative method– Solve for the diagonals.

If we solve for x_1 in the first equation, and x_2 in the second equation, we can get

$$\begin{aligned}
x_1 &= \frac{1}{a_{11}}\Big(b_1 - a_{12}x_2 - a_{13}x_3 \cdots - a_{1n}x_n \Big) \\
x_2 &= \frac{1}{a_{22}}\Big(b_2 - a_{21}x_1 - a_{23}x_3 \cdots - a_{2n}x_n \Big) \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \\
x_i &= \frac{1}{a_{ii}}\Big(b_i - a_{i1}x_1 - a_{i2}x_2 \cdots - a_{in}x_n \Big) \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \\
x_n &= \frac{1}{a_{nn}}\Big(b_n - a_{i1}x_1 - a_{n2}x_2 \cdots - a_{n,n-1}x_{n-1} \Big)
\end{aligned}$$

Given an initial guess $\mathbf{x}^0$, the Jacobi iterative method is

$$\begin{aligned}
x_1^{k+1} &= \frac{1}{a_{11}}\Big(b_1 - a_{12}x_2^k - a_{13}x_3^k \cdots - a_{1n}x_n^k \Big) \\
x_2^{k+1} &= \frac{1}{a_{22}}\Big(b_2 - a_{21}x_1^k - a_{23}x_3^k \cdots - a_{2n}x_n^k \Big) \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \\
x_i^{k+1} &= \frac{1}{a_{ii}}\Big(b_i - a_{i1}x_1^k - a_{i2}x_2^k \cdots - a_{in}x_n^k \Big) \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \\
x_n^{k+1} &= \frac{1}{a_{nn}}\Big(b_n - a_{i1}x_1^k - a_{n2}x_2^k \cdots - a_{n,n-1}x_{n-1}^k \Big).
\end{aligned}$$

In a compact form, it can be written as

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^k \right), \quad i = 1, 2, \cdots, n, \tag{7.18}$$

which provides the basis for easy programming. For the finite difference equation

$$\frac{U_{i+1} - 2U_i + U_{i+1}}{h^2} = f_i$$

with a Dirichlet boundary condition, we have

$$\begin{aligned}
U_1^{k+1} &= \frac{ua + U_2^k}{2} - \frac{h^2 f_1}{2} \\
U_i^{k+1} &= \frac{U_{i-1}^k + U_{i+1}^k}{2} - \frac{h^2 f_i}{2}, \quad i = 2, 3, \cdots n-1 \\
U_{n-1}^{k+1} &= \frac{U_{n-2}^k + ub}{2} - \frac{h^2 f_{n-1}}{2}.
\end{aligned}$$

For a two dimensional Poisson equation, it is

$$U_{ij}^{k+1} = \frac{U_{i-1,j}^k + U_{i+1,j}^k + U_{i,j-1}^k + U_{i,j+1}^k}{4} - \frac{h^2 f_{ij}}{4}, \quad i,j = 1,2,\cdots,n-1.$$

7.5.2 The Gauss-Seidel iterative method– Solve for the diagonals and use the most updated information.

In the Jacobi iterative method, we update all the components of $\mathbf{x}^{k+1}$ based on $\mathbf{x}^k$. In the Gauss-Seidel iterative method, we use the most updated information as follows:

$$\begin{aligned}
x_1^{k+1} &= \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^k - a_{13}x_3^k \cdots - a_{1n}x_n^k\right) \\
x_2^{k+1} &= \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{k+1} - a_{23}x_3^k \cdots - a_{2n}x_n^k\right) \\
\vdots \quad &\quad \vdots \quad \vdots \quad \vdots \\
x_i^{k+1} &= \frac{1}{a_{ii}}\left(b_i - a_{i1}x_1^{k+1} - a_{i2}x_2^{k+1} \cdots - a_{i,i-1}x_{i-1}^{k+1} - a_{i,i+1}x_{i+1}^k - \cdots - a_{in}x_n^k\right) \\
\vdots \quad &\quad \vdots \quad \vdots \quad \vdots \\
x_n^{k+1} &= \frac{1}{a_{nn}}\left(b_n - a_{i1}x_1^{k+1} - a_{n2}x_2^{k+1} \cdots - a_{n,n-1}x_{n-1}^{k+1}\right)
\end{aligned}$$

or in compact form

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k\right), \quad i = 1,2,\cdots,n, \tag{7.19}$$

Below is a pseudo-code of the Gauss-Seidel iterative method for solving the Poisson equation:

```
Give u0(i,j) and a tolerance tol

err = 1000; k = 0;  u = u0;
while err > tol
  for i=1,n
     for j=1,n
        u(i,j) = ( (u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)) -h^2*f(i,j) )/4
     end
  end
  err = max(max(abs(u-u0)));
  u0 = u;  k = k + 1;    % Next iteration if err > tol
end
```

7.5.3 The successive over relaxation (SOR(ω)) iterative method, an extrapolation technique.

Let $\mathbf{x}_{GS}^{k+1}$ be the new update of the Gauss-Seidel method from $\mathbf{x}^k$. The new update using SOR(ω) is the linear combination of $\mathbf{x}^k$ and $\mathbf{x}_{GS}^{k+1}$:

$$\mathbf{x}^{k+1} = (1-\omega)\mathbf{x}^k + \omega \mathbf{x}_{GS}^{k+1}. \tag{7.20}$$

If the parameter $\omega < 1$, the iteration above is called an interpolation, if $\omega > 1$, it is called an extrapolation, or over relaxation. For elliptic problems, usually we should choose $1 \le \omega < 2$. if $\omega = 1$, we are back to the Gauss-Seidel method.

In the component form, the SOR(ω) method can be written as

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k \right), \quad i = 1, 2, \cdots, n. \tag{7.21}$$

We just need to change one line in the pseudo-code of the Gauss-Seidel method to get the SOR(ω) method:

```
u(i,j) = (1-omega)*u0(i,j) + omega*( u(i-1,j) + u(i+1,j)
           + u(i,j-1) + u(i,j+1) -h^2*f(i,j) )/4
```

The convergence of the SOR(ω) method depends on the choice of ω. For the linear system of equations obtained from the standard five-point stencil applied to a Poisson equation with $h = h_x = h_y = 1/n$, it has been shown that the optimal ω is

$$\omega_{opt} = \frac{2}{1+\sin(\pi/n)} \sim \frac{2}{1+\pi/n}. \tag{7.22}$$

Note that the optimal ω approaches number two as n approaches infinity. For general elliptic PDEs, we do not know the optimal ω but we can use the optimal ω for the Poisson equation as a trial value. It is also recommended to use larger ω than smaller ones. If ω is too large that the iterative method diverges, we would find it quickly because the solution will blow-up. Also the optimal ω is independent of the right hand side.

7.5.4 Convergence of the stationary iterative methods.

For a stationary iterative method, the discussion of the convergence of the method is based on the following theorem (sufficient and necessary condition).

Theorem 7.4 *Given a stationary iteration*

$$\mathbf{x}^{k+1} = T\mathbf{x}^k + c, \tag{7.23}$$

where T is a constant matrix, and c is a constant vector. The vector sequence $\{\mathbf{x}^k\}$ converges for arbitrary $\mathbf{x}^0$ if and only if $\rho(T) < 1$, where $\rho(T)$ is the spectral radius of T defined as

$$\rho(T) = \max |\lambda_i(T)|, \tag{7.24}$$

that is, the largest magnitude of all the eigenvalues of T.

Another commonly used sufficient condition to check the convergence of a stationary iterative method is the following theorem.

Theorem 7.5 *If there is a matrix norm $\|\cdot\|$ such that $\|T\| < 1$, then the stationary iterative method converges for arbitrary initial guess $\mathbf{x}^0$.*

We often check $\|T\|_p$ norm for $p = 1, 2, \infty$.

Now let us discuss the convergence of Jacobi, Gauss-Seidel, and SOR(ω) methods.

Given a linear system $A\mathbf{x} = b$, Let D be the diagonal matrix formed from the diagonal elements of A, $-L$ be the lower triangular part of A, and $-U$ be the upper triangular part of A. The iteration matrices for the three basic iteration methods are

- Jacobi method: $T = D^{-1}(L+U)$, $c = D^{-1}b$.
- Gauss-Seidel method: $T = (D-L)^{-1}U$, $c = (D-L)^{-1}b$.
- SOR(ω) method: $T = (I-\omega L)^{-1}\left((1-\omega)I + \omega U\right)$, $c = \omega(I-\omega L)^{-1}D^{-1}b$.

Theorem 7.6 *If A is strictly diagonally dominant, that is*

$$|a_{ii}| > \sum_{j=1, j\neq n}^{n} |a_{ij}|, \tag{7.25}$$

Then both Jacobi and Gauss-Seidel iterative methods converge. The conclusion is also true when (1): A is weakly diagonally dominant

$$|a_{ii}| \geq \sum_{j=1, j\neq n}^{n} |a_{ij}|; \tag{7.26}$$

(2): the inequality holds for at least one row; (3) A is irreducible.

From the theorem above, we can conclude that both Jacobi and Gauss-Seidel iterative methods converge when they are applied to solve the linear system of equations obtained from the standard central finite difference method for Poisson equations.

7.6 Nine-point discrete Laplacian for Poisson equations.

The purpose of the nine-point discrete Laplacian is to get a compact 4-th order accurate scheme, $\|\mathbf{u}-\mathbf{U}\| \le Ch^4$, for a Poisson equation. One of advantages of high order method is that we can use fewer grid points to get the same order accuracy compared with lower order method. We will have smaller system of equations. One of the disadvantages is that we will have a denser system of equations.

While other methods may be used, we use a symbolic derivation from the second order central scheme for u_{xx}. Recall that

$$\begin{aligned} \delta_{xx}^2 u &= \frac{\partial^2 u}{\partial x^2} + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} + O(h^4) \\ &= \left(1 + \frac{h^2}{12}\frac{\partial^2}{\partial x^2}\right)\frac{\partial^2}{\partial x^2} u + O(h^4). \end{aligned} \tag{7.27}$$

Plugging the operator relation

$$\frac{\partial^2}{\partial x^2} = \delta_{xx}^2 + O(h^2)$$

into equation (7.27), we get

$$\begin{aligned} \delta_{xx}^2 u &= \left(1 + \frac{h^2}{12}(\delta_{xx}^2 + O(h^2))\right)\frac{\partial^2}{\partial x^2} u + O(h^4) \\ &= \left(1 + \frac{h^2}{12}\delta_{xx}^2\right)\frac{\partial^2}{\partial x^2} u + O(h^4). \end{aligned}$$

We solver $\frac{\partial^2}{\partial x^2}$ from the equation above to get

$$\frac{\partial^2}{\partial x^2} = \left(1 + \frac{h^2}{12}\delta_{xx}^2\right)^{-1}\delta_{xx}^2 u + \left(1 + \frac{h^2}{12}\delta_{xx}^2\right)^{-1} O(h^4).$$

Note that

$$\left(1 + \frac{h^2}{12}\delta_{xx}^2\right)^{-1} = 1 - \frac{h^2}{12}\delta_{xx}^2 + O(h^4)$$

if h is sufficient small. We get the symbolic relation

$$\begin{aligned} \frac{\partial^2}{\partial x^2} &= \left(1 + \frac{h^2}{12}\delta_{xx}^2\right)^{-1}\delta_{xx}^2 u + O(h^4), \quad \text{or} \\ \frac{\partial^2}{\partial x^2} &= \left(1 - \frac{h^2}{12}\delta_{xx}^2\right)\delta_{xx}^2 u + O(h^4). \end{aligned}$$

With a Cartesian grid, we can approximate the Poisson equation $\Delta u = f$ to get

$$\left(1 + \frac{h_x^2}{12}\delta_{xx}^2\right)^{-1}\delta_{xx}^2 u + \left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right)^{-1}\delta_{yy}^2 u = f(x,y) + O(h^4)$$

where $h = \max(h_x, h_y)$. We multiply

$$\left(1 + \frac{h_x^2}{12}\delta_{xx}^2\right)\left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right)$$

to the expression above and use the fact that

$$\left(1 + \frac{(\Delta x)^2}{12}\delta_{xx}^2\right)\left(1 + \frac{(\Delta y)^2}{12}\delta_{yy}^2\right) = \left(1 + \frac{(\Delta y)^2}{12}\delta_{yy}^2\right)\left(1 + \frac{(\Delta x)^2}{12}\delta_{xx}^2\right).$$

That is, the are commutative, we can get:

$$\left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right)\delta_{xx}^2 u + \left(1 + \frac{h_x^2}{12}\delta_{xx}^2\right)\delta_{yy}^2 u = \left(1 + \frac{h_x^2}{12}\delta_{xx}^2\right)\left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right) f(x,y) + O(h^4)$$
$$= \left(1 + \frac{h_x^2}{12}\delta_{xx}^2 + \frac{h_y^2}{12}\delta_{yy}^2\right) f(x,y) + O(h^4)$$

Expand the expression above, we get the nine-point scheme for the Poisson equation, for example

$$\left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right)\delta_{xx}^2 U_{ij} = \left(1 + \frac{h_y^2}{12}\delta_{yy}^2\right)\frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{(h_x)^2}$$
$$= \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{(h_x)^2} + \frac{1}{12(h_y)^2}\Big(U_{i-1,j-1}, -2U_{i-1,j} + U_{i-1,j+1}$$
$$- 2U_{i,j-1} + 4U_{ij} - 2U_{i,j+1} + U_{i+1,j-1} - 2U_{i+1,j} + U_{i+1,j+1}\Big)$$

For the special case when $h_x = h_y = h$, the finite difference coefficients and the linear combination of f are expressed in Fig. 10.

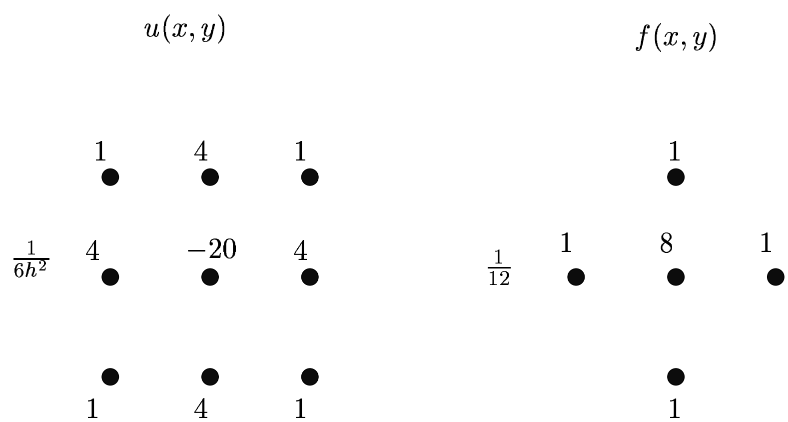

Figure 10: The coefficients of the finite difference scheme using the nine-point stencil.

Possible advantages and disadvantages of nine-point finite difference methods for Poisson equations include.

- It is fourth order accurate and it is still compact. The coefficient matrix is still block tri-diagonal.
- Less grid orientation effects compared with the standard five point finite difference scheme.

Note that if we apply

$$\frac{\partial^2}{\partial x^2} = \left(1 - \frac{h^2}{12}\delta_{xx}^2\right)\delta_{xx}^2 u + O(h^4).$$

to the Poisson equation, we will get another nine-point finite difference scheme which is not compact and stronger grid orientation effects.

7.7 A finite difference method for Poisson equation using polar coordinates.

If the domain of the interest is a circle, or an annulus, or a fan etc., see Fig. 11. It is much easier to use polar coordinates:

$$x = r\cos\theta, \qquad y = r\sin\theta. \tag{7.28}$$

Under the polar coordinates, the Poisson equation is

$$\begin{aligned} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} &= f(r,\theta) \\ \text{or} \quad \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} &= f(r,\theta). \end{aligned}$$

If the origin is not in the domain, $R_1 \le r \le R_2$, $\theta_l \le \theta \le \theta_r$, we can use a uniform grid in polar coordinates to discretize the Poisson equation:

$$\begin{aligned} r_i &= R_1 + i\Delta r, \quad i = 0,1,\cdots,m, \quad \Delta r = \frac{R_2 - R_1}{m} \\ \theta_j &= \theta_l + j\Delta\theta, \quad j = 0,1,\cdots,N, \quad \Delta\theta = \frac{\theta_r - \theta_l}{N}. \end{aligned}$$

The discretized equation is (using the conservative form)

$$\begin{aligned} &\frac{1}{r_i}\frac{r_{i-\frac{1}{2}}U_{i-1,j} - (r_{i-\frac{1}{2}} + r_{i+\frac{1}{2}})U_{ij} + r_{i+\frac{1}{2}}U_{i+1,j}}{(\Delta r)^2} \\ &\quad + \frac{1}{r_i^2}\frac{U_{i,j-1} - 2U_{ij} + U_{i,j+1}}{(\Delta\theta)^2} = f(r_i,\theta_j). \end{aligned} \tag{7.29}$$

where U_{ij} is an approximation to the solution $u(r_i, \theta_j)$.

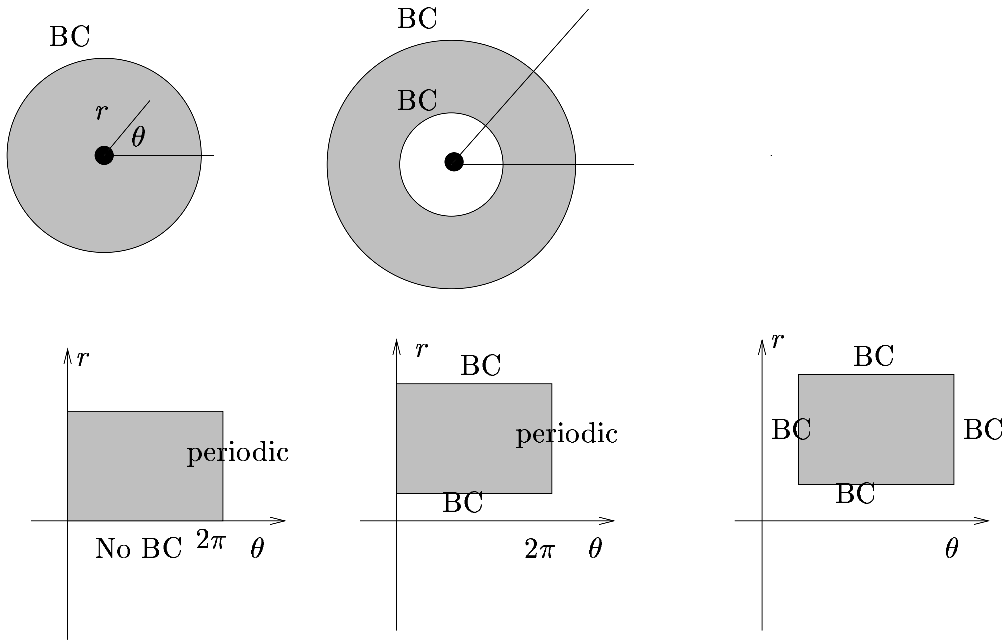

Figure 11: A diagram of domains and boundary conditions that may be better solved in polar coordinates.

7.8 Treating the boundary conditions.

If the origin is within the domain, then $0 \le \theta \le 2\pi$, we should use the *periodic boundary condition* in θ direction, that is $u(r,\theta) = u(r,\theta+2\pi)$. In the r direction, $R_1 = 0$ needs special attention. There are different methods in the literature in dealing with the pole singularity. Some methods will lead to an un-desirable structure of the coefficient matrix of the finite difference equation. One clever approach discussed here is to use the staggered grid:

$$r_i = (i - \frac{1}{2})\Delta r, \quad \Delta r = \frac{R_2}{m - \frac{1}{2}}, \quad i = 1, 2, \cdots, m. \tag{7.30}$$

Notice that $r_1 = \Delta r/2$ and $r_m = R_2$. We can use the conservative form of the discretization at $i = 2, \cdots, m-1$, except for $i = 1$. At $i = 1$, we use the non-conservative form to take care of the pole singularity:

$$\frac{U_{0j} - 2U_{1j} + U_{2j}}{(\Delta r)^2} + \frac{1}{r_1}\frac{U_{2j} - U_{0j}}{2\Delta r} + \frac{1}{r_1^2}\frac{U_{1,j-1} - 2U_{1j} + U_{1,j+1}}{(\Delta\theta)^2} = f(r_1, \theta_j).$$

Note that $r_0 = -\Delta r/2$ and $r_1 = \Delta r/2$. We can see that the coefficient of U_{0j}, which is the approximation at the ghost point r_0, is zero! The above finite difference equation is simplified to

$$\frac{-2U_{1j}+U_{2j}}{(\Delta r)^2}+\frac{1}{r_1}\frac{U_{2j}}{2\Delta r}+\frac{1}{r_1^2}\frac{U_{1,j-1}-2U_{1j}+U_{1,j+1}}{(\Delta \theta)^2}=f(r_1,\theta_j).$$

We still have a diagonally dominant system of linear equations.

7.9 Use FFT to solve Poisson equation in polar coordinates.

Since the solution $u(r,\theta)$ is periodic in θ is the origin is an interior point, we can approximate it by the truncated Fourier series as

$$u(r,\theta)=\sum_{n=-N/2}^{N/2-1} u_n(r)e^{in\theta} \tag{7.31}$$

where $i=\sqrt{-1}$ and $u_n(r)$ is the complex Fourier coefficient given by

$$u_n(r)=\frac{1}{N}\sum_{k=0}^{N-1} u(r,\theta)e^{-ink\theta}. \tag{7.32}$$

Plugging (7.31) in to the Poisson equation, we get

$$\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial u_n}{\partial r}\right)-\frac{n^2}{r^2}u_n \quad = \quad f_n(r), \quad n=-N/2,\cdots,N/2-1, \tag{7.33}$$

where $f_n(r)$ is the n-th coefficient of the Fourier series of $f(r,\theta)$ defined in (7.32). For each n, we can discretize in the r direction using the staggered grid to get a tridiagonal system of equations which can be solved easily.

Assuming a Dirichlet boundary condition $u(r_{max},\theta)=u^{BC}(\theta)$ at $r=r_{max}$, we can use the Fourier transform

$$u_n^{BC}(r_{max})=\frac{1}{N}\sum_{k=0}^{N-1} u^{BC}(\theta)e^{-ink\theta} \tag{7.34}$$

to find $u_n^{BC}(r_{max})$ which is the boundary condition for the ordinary differential equation.

Once we have the Fourier coefficient u_n, we can use the inverse Fourier transform (7.31) to get an approximate solution to the Poisson equation.