澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

不惑新航
揚帆追夢
1981
2021
SET SAIL ANEW ON
THE RUBY JUBILEE

Finite Difference Methods for 2D Elliptic PDEs

MATH 3014
Monday & Thursday 14:30-15:45
Instructor: **Dr. Luo Li**
https://www.fst.um.edu.mo/personal/liluo/math3014/

Department of Mathematics
Faculty of Science and Technology

Examples of Linear and Nonlinear Equations of Elliptic PDEs

- Laplace equations in 2D

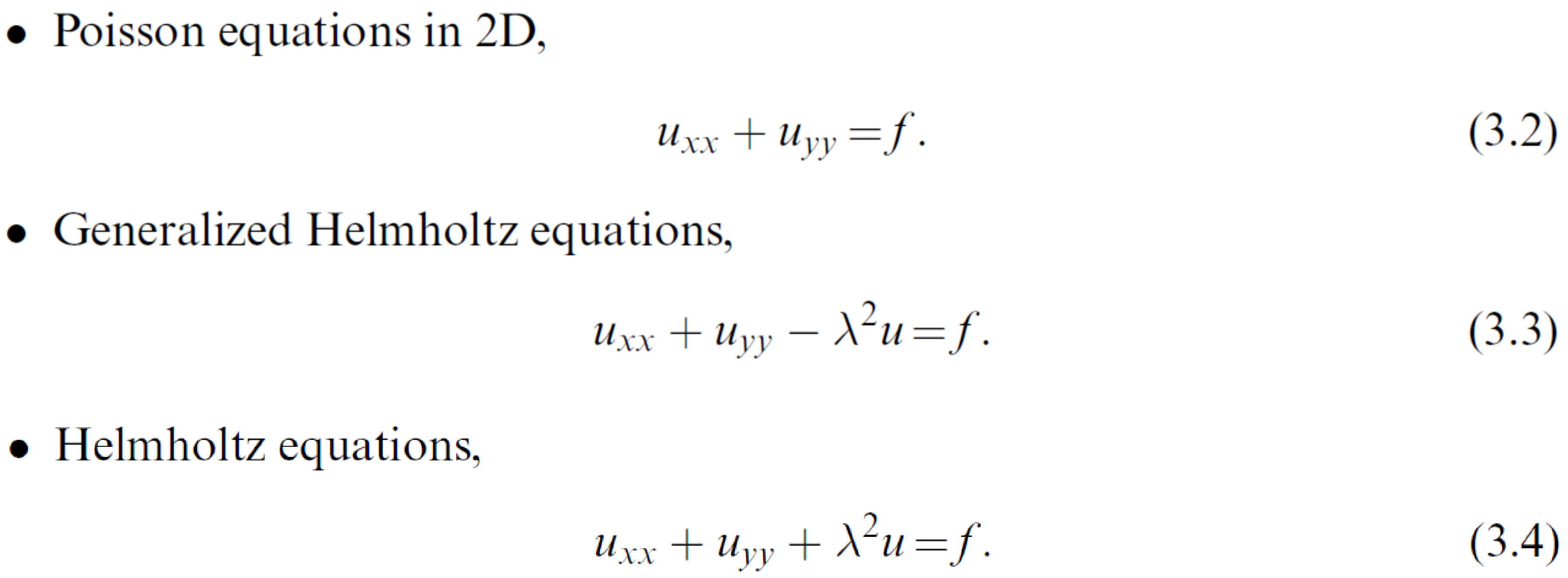
$$\Delta u = \nabla^2 u = \nabla \cdot \nabla u = u_{xx} + u_{yy} = 0\,. \qquad (3.1)$$

In 2D, the gradient operator is $\nabla = [\,\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\,]^T$

The divergence of the vector v is $\nabla \cdot \mathbf{v} = div(\mathbf{v}) = \frac{\partial v1}{\partial x} + \frac{\partial v2}{\partial y}$

(3.1) means that the conservative vector field $\mathbf{v} = \nabla u$ is also divergence free, i.e., $div(\mathbf{v}) = \nabla \cdot \mathbf{v} = 0$

The solution u is sometimes called a potential function.



- Poisson equations in 2D,

$$u_{xx} + u_{yy} = f. \tag{3.2}$$

- Generalized Helmholtz equations,

$$u_{xx} + u_{yy} - \lambda^2 u = f. \tag{3.3}$$

- Helmholtz equations,

$$u_{xx} + u_{yy} + \lambda^2 u = f. \tag{3.4}$$

- Many incompressible flow solvers are based on solving one or several Poisson or Helmholtz equations.
- The Helmholtz equation arises in scattering problems.
- The problem is hard to solve numerically if λ is large.

The incompressible Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u}$$

$$\nabla\cdot\mathbf{u} = 0$$

Chorin's projection method for solving the above equation:

$$(1) \qquad \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n\cdot\nabla)\mathbf{u}^n + \nu\nabla^2\mathbf{u}^n$$

$$(2) \qquad \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p^{n+1}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} \xrightarrow{\nabla\cdot\mathbf{u}^{n+1} = 0} \nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla\cdot\mathbf{u}^*$$

- General self-adjoint elliptic PDEs,

$$\nabla \cdot (a(x,y)\nabla u(x,y)) - q(x,y)u = f(x,y) \tag{3.5}$$

$$\text{or} \quad (au_x)_x + (au_y)_y - q(x,y)u = f(x,y)\,. \tag{3.6}$$

$a(x,y) \geq a_0 > 0$, where a_0 is a constant, and $q(x,y) \geq 0$.

- General elliptic PDEs (diffusion and advection equations),

$$\begin{aligned} &a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} \\ &\qquad + d(x,y)u_x + e(x,y)u_y + g(x,y)u(x,y) = f(x,y), \quad (x,y) \in \Omega, \end{aligned}$$

if $b^2 - ac < 0$ for all $(x,y) \in \Omega$.

3.1 Boundary and Compatibility Conditions

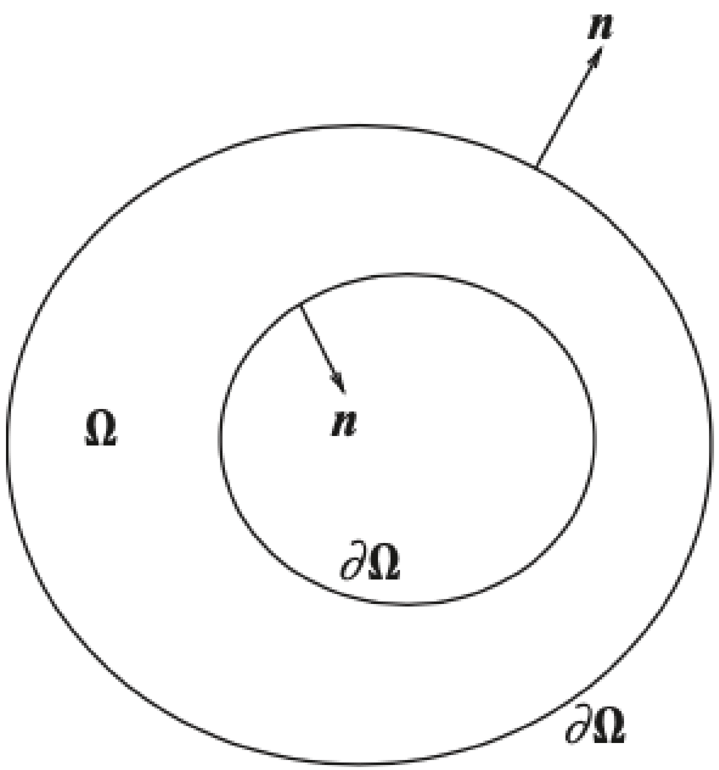


Figure 3.1. A diagram of a 2D domain Ω, its boundary $\partial\Omega$, and its unit normal direction.

- Dirichlet boundary condition: the solution is known on the boundary,

$$u(x,y)|_{\partial\Omega} = u_0(x,y)\,.$$

- Neumann or flux boundary condition: the normal derivative is given along the boundary,

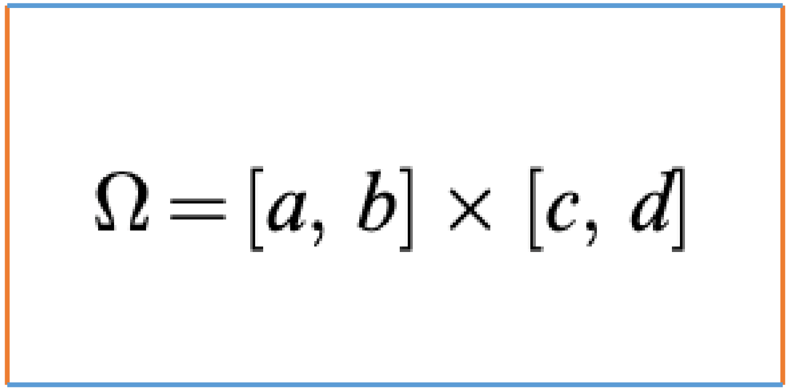
$$\frac{\partial u}{\partial n} \equiv \mathbf{n}\cdot\nabla u = u_n = u_x n_x + u_y n_y = g(x,y)\,,$$

where $\mathbf{n} = (n_x, n_y)$ $(n_x^2 + n_y^2 = 1)$ is the unit normal direction.

- In some cases, a boundary condition is periodic

$$u(a,y) = u(b,y)$$

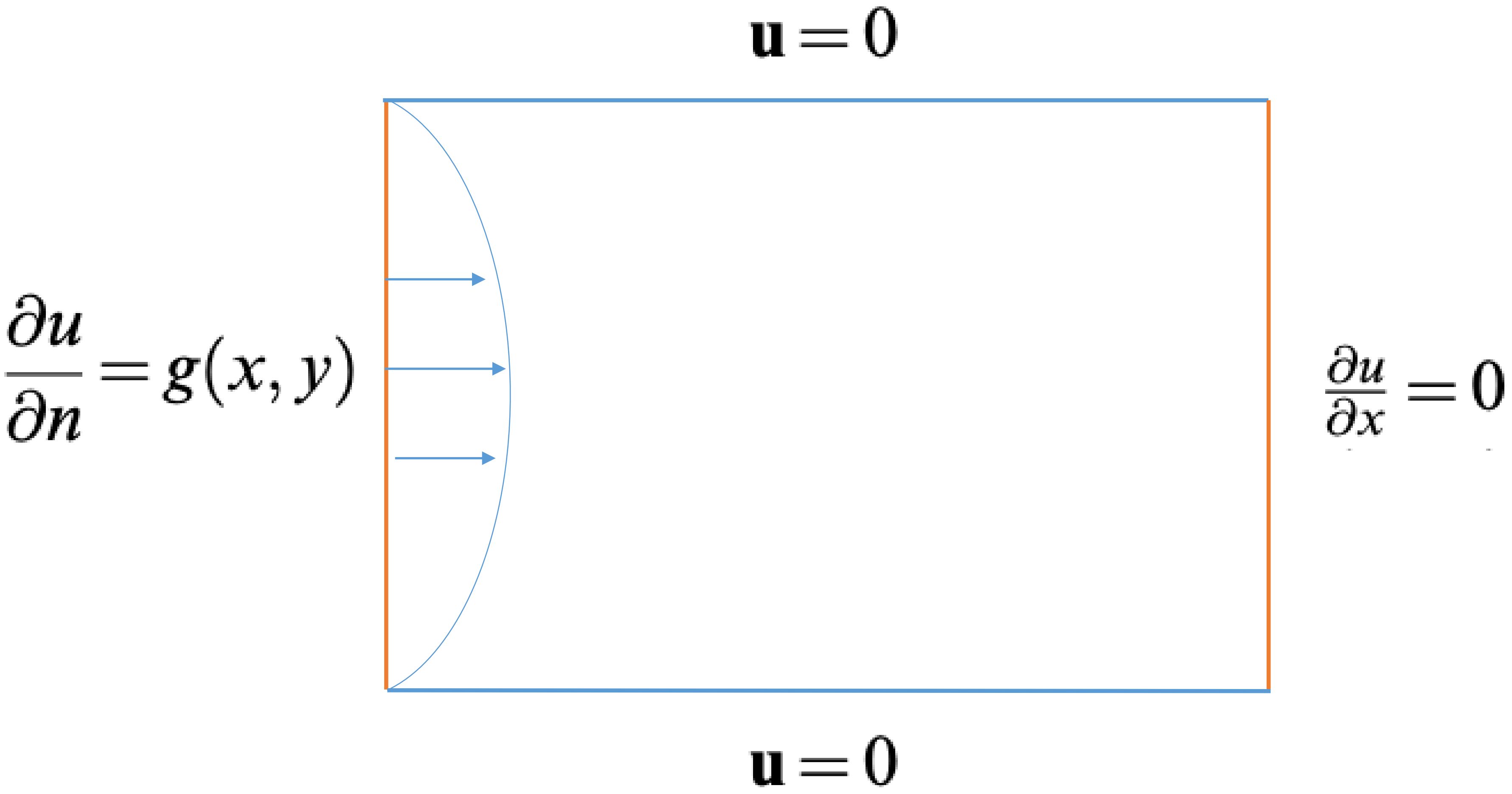
$$u(x,c) = u(x,d)$$



$$\Omega = [a,\ b] \times [c,\ d]$$

An example:

different boundary conditions on different parts of the boundary
(A typical case in fluid dynamics: flow passing through a tube)



$$\mathbf{u} = 0$$

$$\frac{\partial u}{\partial n} = g(x, y)$$

$$\frac{\partial u}{\partial x} = 0$$

$$\mathbf{u} = 0$$

Compatibility condition for a Poisson equation with a purely Neumann boundary condition

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

On integrating over the domain Ω

$$\iint_\Omega \Delta u dxdy = \iint_\Omega f(x,y)dxdy\,,$$

and applying the Green's theorem gives

$$\iint_\Omega \Delta u dxdy = \oint_{\partial\Omega} \frac{\partial u}{\partial n} ds\,,$$

so we have the compatibility condition

$$\iint_\Omega \Delta u dxdy = \oint_{\partial\Omega} g\, ds = \iint_\Omega f(x,y)dxdy \qquad (3.11)$$

3.2 The Central Finite Difference Method for Poisson Equations

$$u_{xx} + u_{yy} = f(x, y), \quad (x, y) \in \Omega = (a, b) \times (c, d), \tag{3.12}$$

$$u(x, y)|_{\partial\Omega} = u_0(x, y)\,. \tag{3.13}$$

- Step 1: Generate a grid. For example, a uniform Cartesian grid can be generated with two given parameters m and n:

$$x_i = a + ih_x, \quad i = 0, 1, 2, \ldots, m, \quad h_x = \frac{b - a}{m}, \tag{3.14}$$

$$y_j = c + jh_y, \quad j = 0, 1, 2, \ldots, n, \quad h_y = \frac{d - c}{n}. \tag{3.15}$$

In seeking an approximate solution U_{ij} at the grid points (x_i, y_j) where $u(x, y)$ is unknown, there are $(m - 1)(n - 1)$ unknowns.

- Step 2: Approximate the partial derivatives at grid points with finite difference formulas involving the function values at nearby grid points.

$$\frac{u(x_{i-1},y_j)-2u(x_i,y_j)+u(x_{i+1},y_j)}{(h_x)^2}+\frac{u(x_i,y_{j-1})-2u(x_i,y_j)+u(x_i,y_{j+1})}{(h_y)^2}$$

$$=f_{ij}+T_{ij},\quad i=1,\ldots,m-1,\quad j=1,\ldots,n-1, \tag{3.16}$$

The local truncation error

$$h=\max\{h_x,\,h_y\}\,.$$

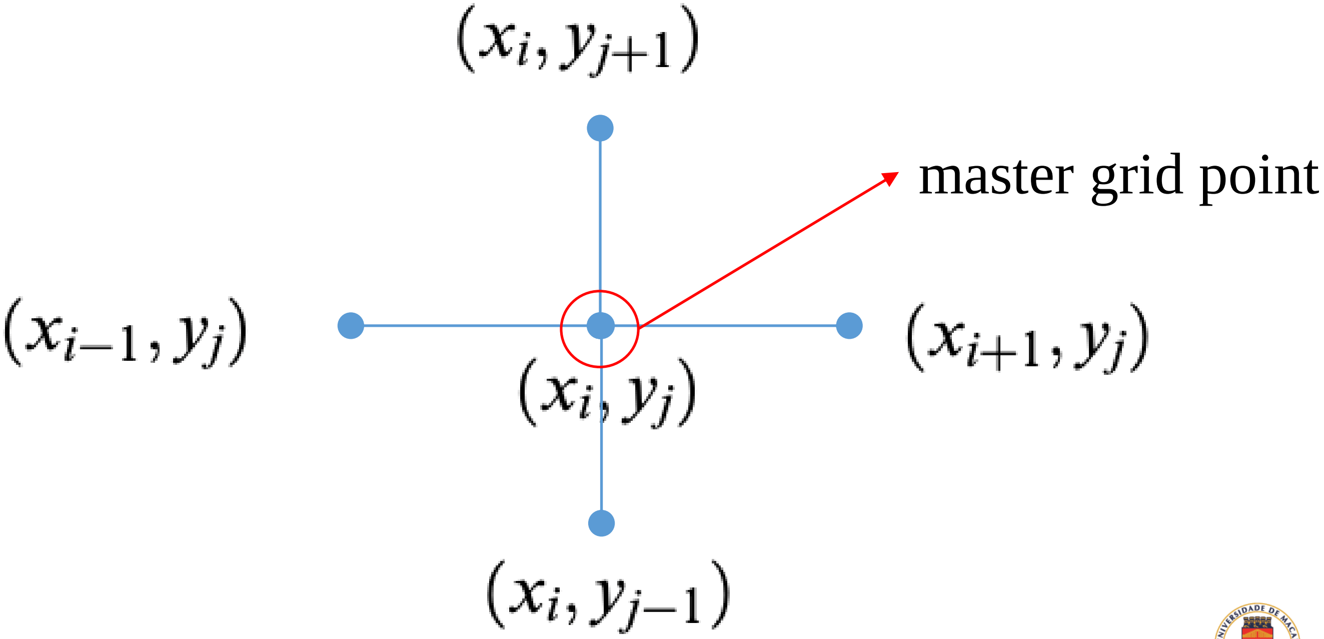
$$T_{ij}\sim\frac{(h_x)^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i,y_j)+\frac{(h_y)^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i,y_j)+O(\boxed{h^4}), \tag{3.17}$$

Recall that $T(x)=\dfrac{u(x-h)-2u(x)+u(x+h)}{h^2}-u''(x)=\dfrac{h^2}{12}u^{(4)}(x)+\cdots=O(h^2)$

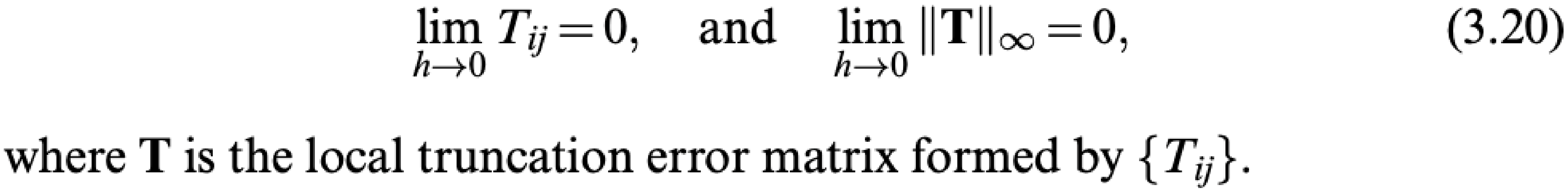
Three-point central finite difference formula

$$\frac{U_{i-1,j}+U_{i+1,j}}{(h_x)^2}+\frac{U_{i,j-1}+U_{i,j+1}}{(h_y)^2}-\left(\frac{2}{(h_x)^2}+\frac{2}{(h_y)^2}\right)U_{ij}=f_{ij}, \tag{3.19}$$

$$i=1,2,\ldots,m-1, \quad j=1,2,\ldots,n-1\,.$$



The finite difference discretization is second-order accurate and consistent



$$\lim_{h\to 0} T_{ij} = 0, \quad \text{and} \quad \lim_{h\to 0} \|\mathbf{T}\|_\infty = 0, \tag{3.20}$$

where $\mathbf{T}$ is the local truncation error matrix formed by $\{T_{ij}\}$.

- Step 3: Solve the linear system of algebraic equations (3.19), to get the approximate values for the solution at all of the grid points.

- Step 4: Error analysis, implementation, visualization, etc.

3.2.1 The Matrix–vector Form of the FD Equations

$$AU = \mathbf{F}$$

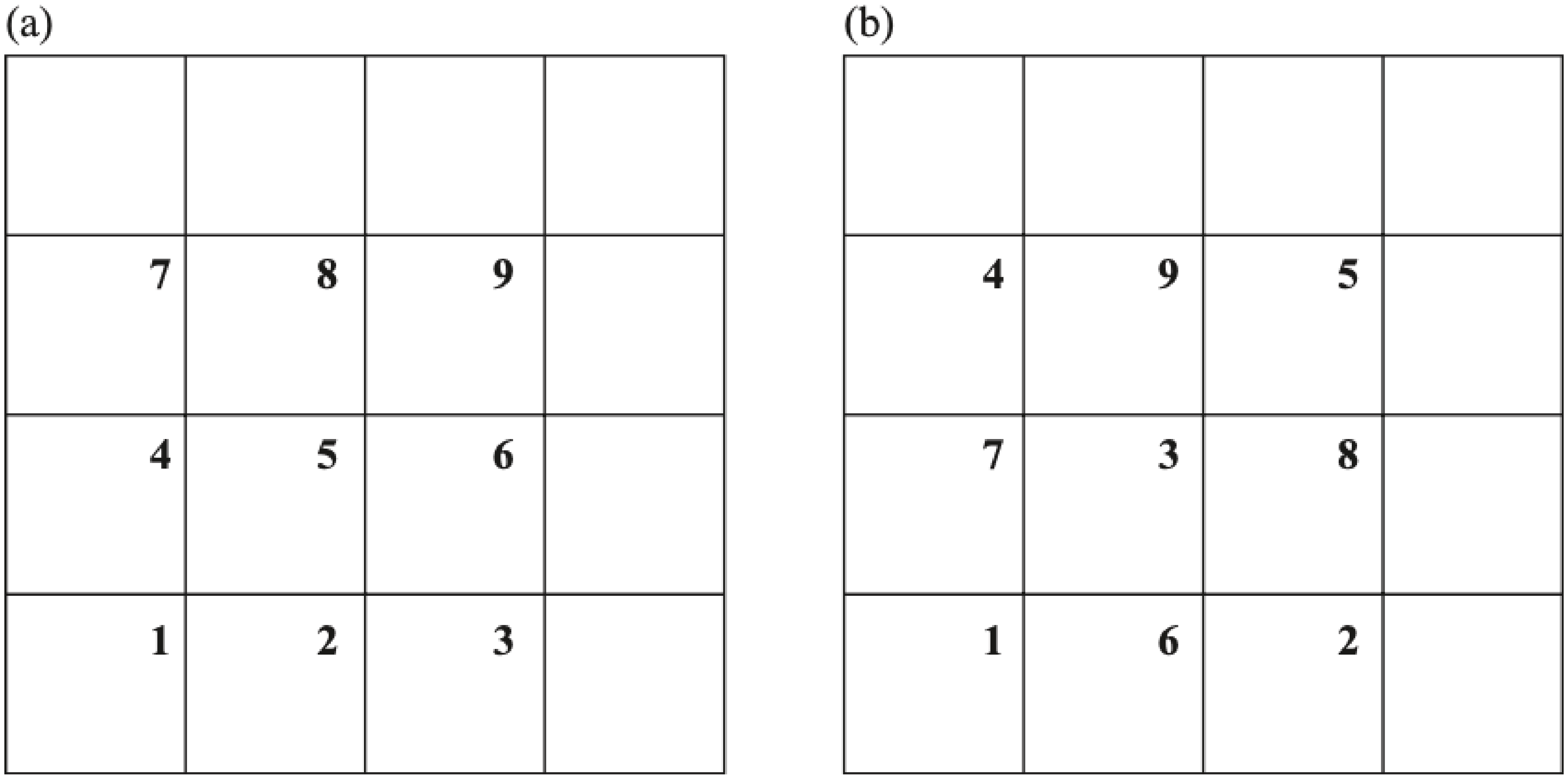
unknowns {Uij} are a 2D array

U_{13}	U_{23}	U_{33}	
U_{12}	U_{22}	U_{32}	
U_{11}	U_{21}	U_{31}	

$$\xrightarrow{\text{ordering}}$$

1D array

$$\mathbf{U} = \begin{bmatrix} U_{11} \\ U_{21} \\ U_{31} \\ \vdots \\ U_{23} \\ U_{33} \end{bmatrix}$$



(a)

7	8	9	
4	5	6	
1	2	3	

(b)

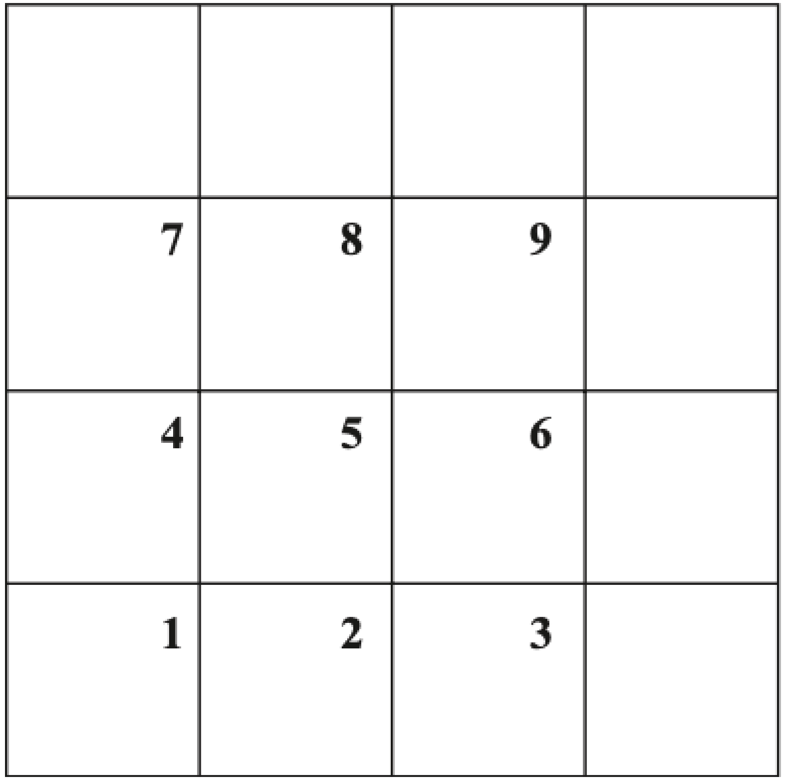
4	9	5	
7	3	8	
1	6	2	

Figure 3.2. (a) The natural ordering and (b) the red–black ordering.

3.2.1.1 The Natural Row Ordering

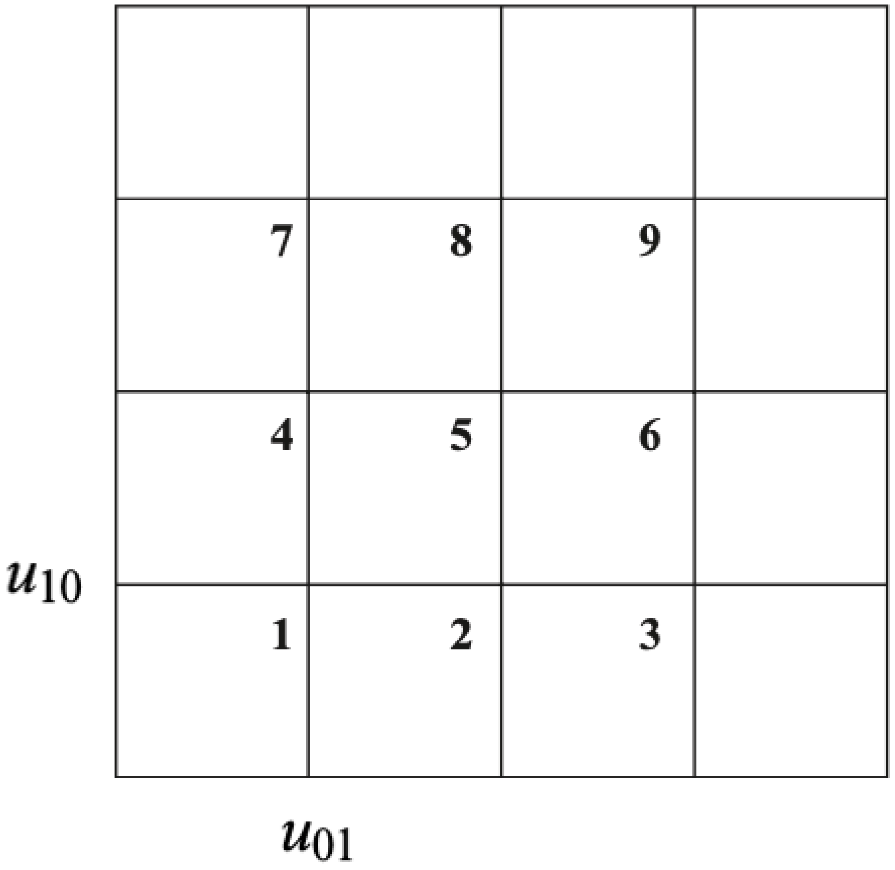
The k-th finite difference equation corresponding to (i, j)

$$k = i + (m-1)(j-1), \quad i = 1, 2, \ldots, m-1, \quad j = 1, 2, \ldots, n-1 \quad (3.21)$$



$$h_x = h_y = h$$

$$\mathbf{U} = \begin{bmatrix} U_{11} \\ U_{21} \\ U_{31} \\ \vdots \\ U_{23} \\ U_{33} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_8 \\ x_9 \end{bmatrix}$$

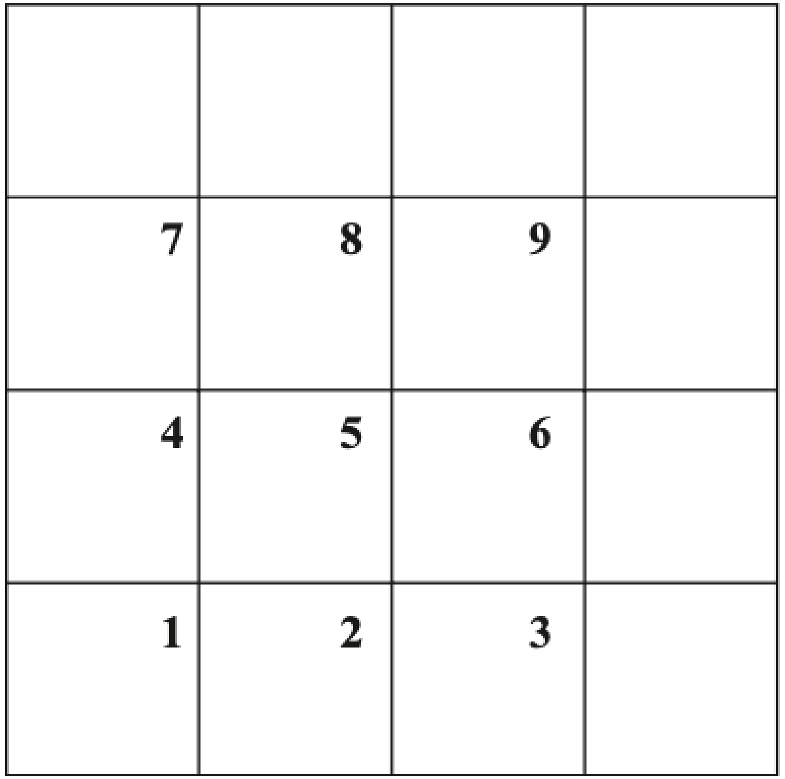


$$Eqn.1: \quad \frac{1}{h^2}\left(-4x_1 + x_2 + x_4\right) = f_{11} - \frac{u_{01} + u_{10}}{h^2}$$

$$Eqn.2: \quad \frac{1}{h^2}\left(x_1 - 4x_2 + x_3 + x_5\right) = f_{21} - \frac{u_{20}}{h^2}$$

$$Eqn.3: \quad \frac{1}{h^2}\left(x_2 - 4x_3 + x_6\right) = f_{31} - \frac{u_{30} + u_{41}}{h^2}$$

$$Eqn.4: \quad \frac{1}{h^2}\left(x_1 - 4x_4 + x_5 + x_7\right) = f_{12} - \frac{u_{02}}{h^2}$$



$$Eqn.5: \quad \frac{1}{h^2}(x_2 + x_4 - 4x_5 + x_6 + x_8) = f_{22}$$

$$Eqn.6: \quad \frac{1}{h^2}(x_3 + x_5 - 4x_6 + x_9) = f_{32} - \frac{u_{42}}{h^2}$$

$$Eqn.7: \quad \frac{1}{h^2}(x_4 - 4x_7 + x_8) = f_{13} - \frac{u_{03} + u_{14}}{h^2}$$

$$Eqn.8: \quad \frac{1}{h^2}(x_5 + x_7 - 4x_8 + x_9) = f_{23} - \frac{u_{24}}{h^2}$$

$$Eqn.9: \quad \frac{1}{h^2}(x_6 + x_8 - 4x_9) = f_{33} - \frac{u_{34} + u_{43}}{h^2}.$$

The corresponding coefficient matrix is *block tridiagonal*,

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & 0 \\ I & B & I \\ 0 & I & B \end{bmatrix}, \tag{3.23}$$

where I is the 3×3 identity matrix and

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

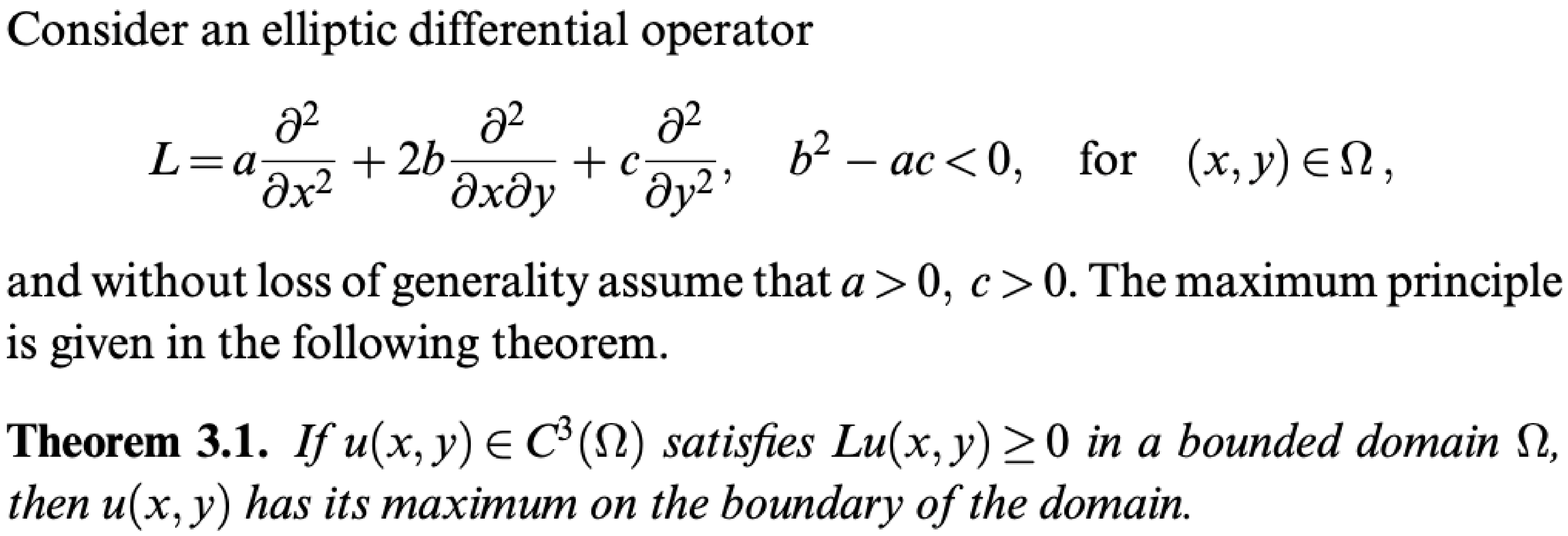
In general, for an $n+1$ by $n+1$ grid we obtain

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & \end{bmatrix}_{n^2\times n^2}, \quad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & \end{bmatrix}_{n\times n}$$

- $-A$ is symmetric positive definite ⇒ A is nonsingular/invertible ⇒The solution of $A\mathbf{U} = \mathbf{F}$ is unique
- $-A$ is weakly diagonally dominant ⇒ $A\mathbf{U} = \mathbf{F}$ can be solved by iterative methods efficiently

i.e., Jacobi, Gauss–Seidel, or SOR(ω), …

3.3 The Maximum Principle and Error Analysis

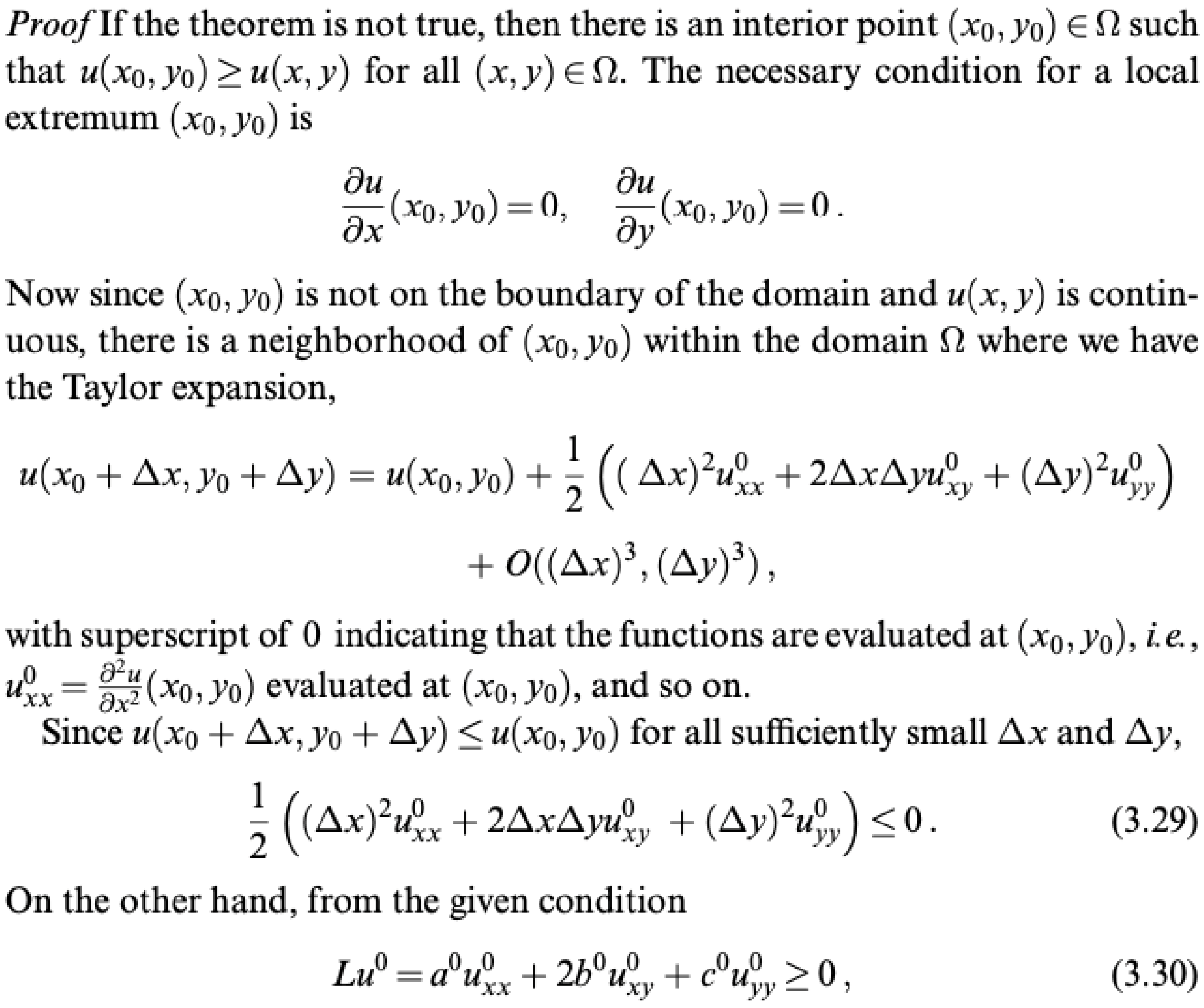


Consider an elliptic differential operator

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

and without loss of generality assume that $a > 0,\ c > 0$. The maximum principle is given in the following theorem.

Theorem 3.1. *If $u(x,y) \in C^3(\Omega)$ satisfies $Lu(x,y) \geq 0$ in a bounded domain Ω, then $u(x,y)$ has its maximum on the boundary of the domain.*



Proof If the theorem is not true, then there is an interior point $(x_0, y_0) \in \Omega$ such that $u(x_0, y_0) \geq u(x, y)$ for all $(x, y) \in \Omega$. The necessary condition for a local extremum (x_0, y_0) is

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

Now since (x_0, y_0) is not on the boundary of the domain and $u(x, y)$ is continuous, there is a neighborhood of (x_0, y_0) within the domain Ω where we have the Taylor expansion,

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

with superscript of 0 indicating that the functions are evaluated at (x_0, y_0), *i.e.*, $u_{xx}^0 = \frac{\partial^2 u}{\partial x^2}(x_0, y_0)$ evaluated at (x_0, y_0), and so on.

Since $u(x_0 + \Delta x, y_0 + \Delta y) \leq u(x_0, y_0)$ for all sufficiently small Δx and Δy,

$$\frac{1}{2}\left((\Delta x)^2 u_{xx}^0 + 2\Delta x \Delta y u_{xy}^0 + (\Delta y)^2 u_{yy}^0 \right) \leq 0\,. \tag{3.29}$$

On the other hand, from the given condition

$$Lu^0 = a^0 u_{xx}^0 + 2b^0 u_{xy}^0 + c^0 u_{yy}^0 \geq 0\,, \tag{3.30}$$

… (Find the contradiction between (3.29) and (3.30))

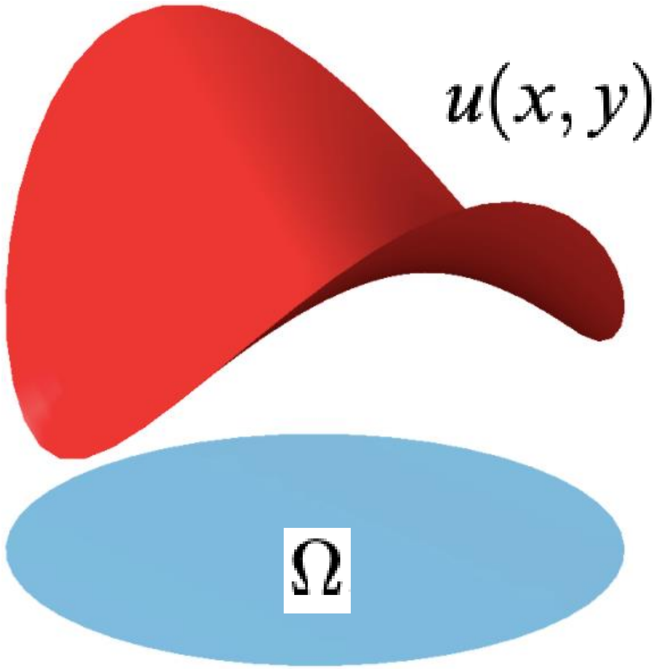
Full Proof on P56-57 of the Textbook:
Zhilin Li et al., Numerical Solution of Differential Equations -- Introduction to Finite Difference and Finite Element Methods.

On the other hand, if $Lu \leq 0$ then the minimum value of u is on the boundary of Ω. For general elliptic equations the maximum principle is as follows. Let

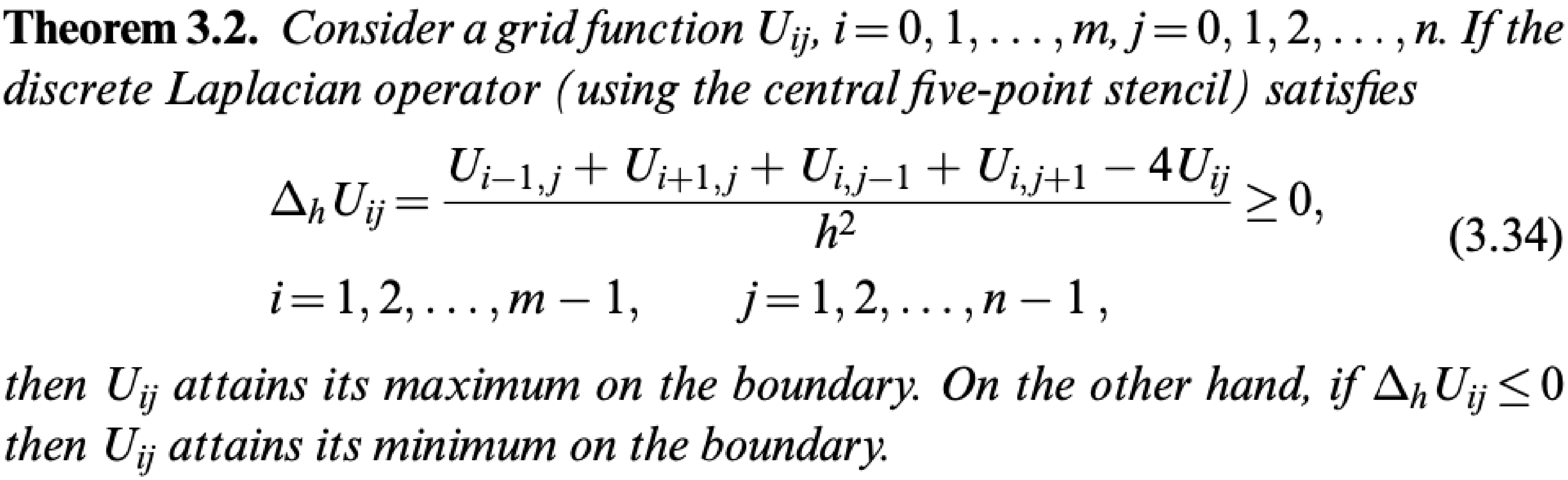
$$Lu = au_{xx} + 2bu_{xy} + cu_{yy} + d_1u_x + d_2u_y + eu = 0, \quad (x,y) \in \Omega,$$

$$b^2 - ac < 0, \quad a > 0,\ c > 0, \quad e \leq 0,$$

where Ω is a bounded domain. Then from Theorem 3.1, $u(x,y)$ cannot have a positive local maximum or a negative local minimum in the interior of Ω.



3.3.1 The Discrete Maximum Principle



Theorem 3.2. *Consider a grid function* U_{ij}, $i = 0, 1, \ldots, m$, $j = 0, 1, 2, \ldots, n$. *If the discrete Laplacian operator (using the central five-point stencil) satisfies*

$$\Delta_h U_{ij} = \frac{U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} \geq 0, \qquad (3.34)$$

$$i = 1, 2, \ldots, m-1, \qquad j = 1, 2, \ldots, n-1\,,$$

then U_{ij} *attains its maximum on the boundary. On the other hand, if* $\Delta_h U_{ij} \leq 0$ *then* U_{ij} *attains its minimum on the boundary.*

Compared to Theorem 3.1

$$L = a\frac{\partial^2}{\partial x^2} + 2b\frac{\partial^2}{\partial x \partial y} + c\frac{\partial^2}{\partial y^2} \geq 0$$

$$a = c = 1, b = 0$$

Proof Assume that the theorem is not true, so U_{ij} has its maximum at an interior grid point (i_0, j_0). Then $U_{i_0,j_0} \geq U_{i,j}$ for all i and j, and therefore

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

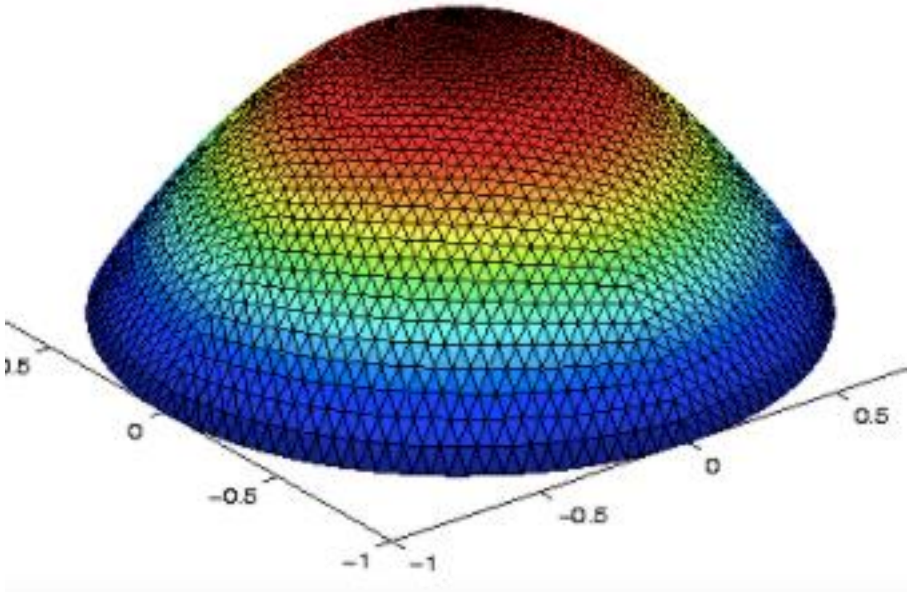
On the other hand, from the condition $\Delta_h U_{ij} \geq 0$

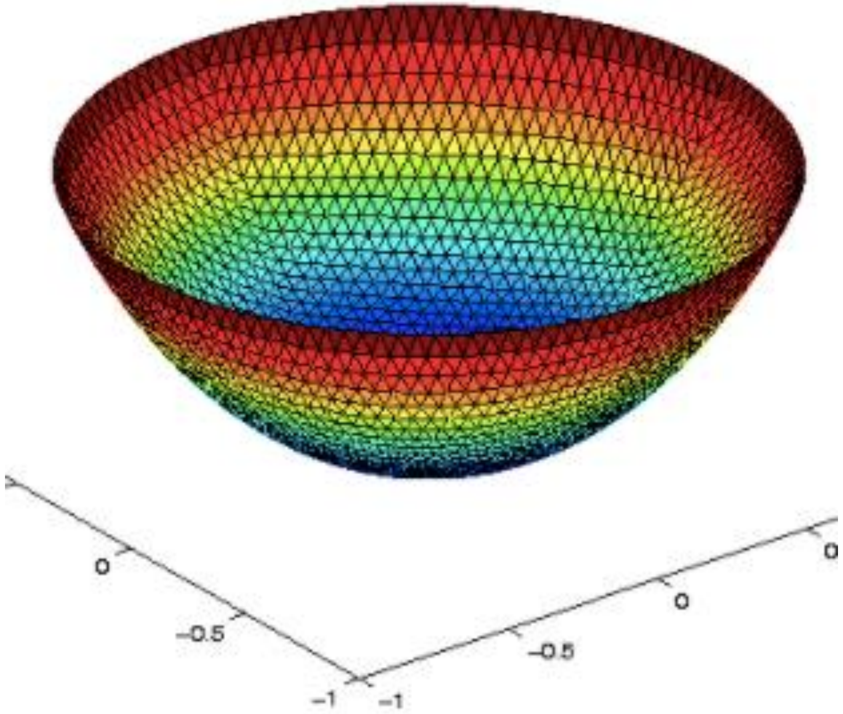
$$U_{i_0,j_0} \leq \frac{1}{4}\left(U_{i_0-1,j_0} + U_{i_0+1,j_0} + U_{i_0,j_0-1} + U_{i_0,j_0+1}\right),$$

contradiction

Unless? $U_{i_0-1,j_0} = U_{i_0+1,j_0} = U_{i_0,j_0-1} = U_{i_0,j_0+1} = U_{i_0,j_0}$

If U looks like this, what's the sign of $\Delta_h U_{ij}$

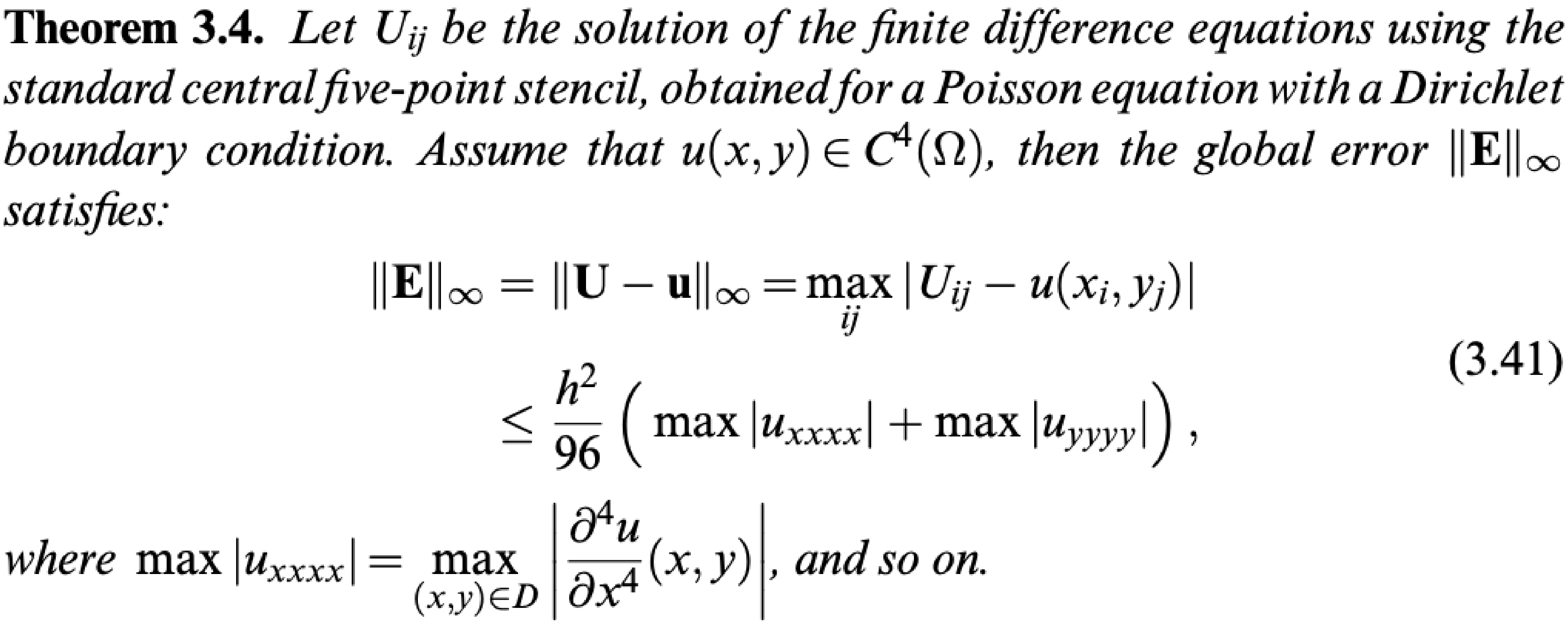




$\Delta_h U_{ij} \leq 0$

$\Delta_h U_{ij} \geq 0$

3.3.2 Error Estimates of the Finite Difference Method for Poisson Equations



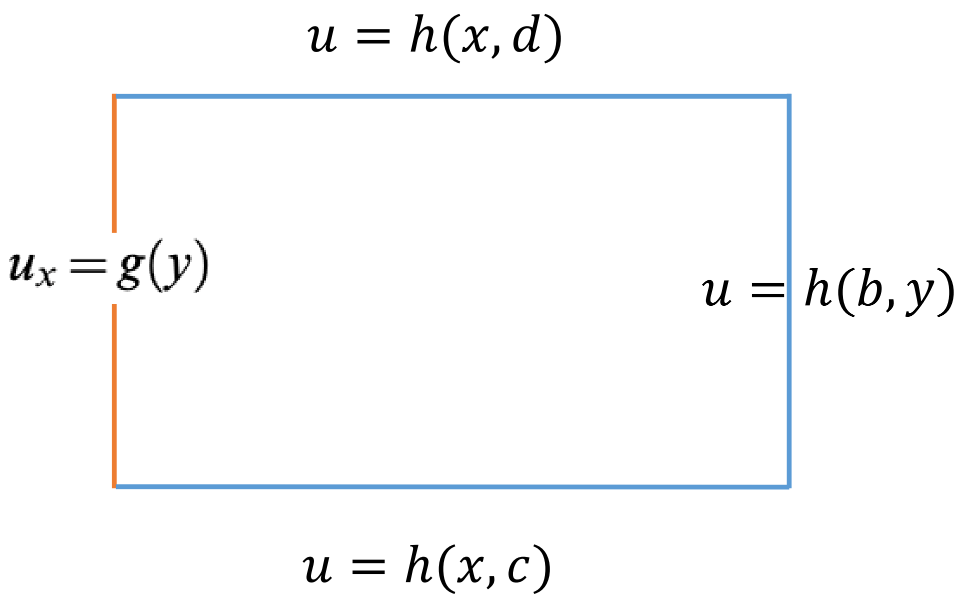
Theorem 3.4. *Let U_{ij} be the solution of the finite difference equations using the standard central five-point stencil, obtained for a Poisson equation with a Dirichlet boundary condition. Assume that $u(x,y) \in C^4(\Omega)$, then the global error $\|\mathbf{E}\|_\infty$ satisfies:*

$$\begin{aligned} \|\mathbf{E}\|_\infty = \|\mathbf{U}-\mathbf{u}\|_\infty &= \max_{ij} |U_{ij} - u(x_i, y_j)| \\ &\le \frac{h^2}{96}\left(\max|u_{xxxx}| + \max|u_{yyyy}| \right), \end{aligned} \tag{3.41}$$

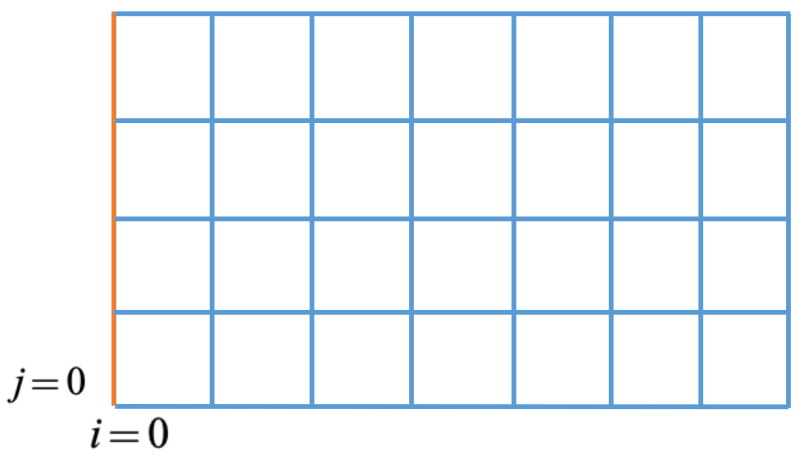
where $\max|u_{xxxx}| = \max\limits_{(x,y)\in D} \left| \frac{\partial^4 u}{\partial x^4}(x,y) \right|$, *and so on.*

3.4 Finite Difference Methods for General Second-order Elliptic PDEs

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

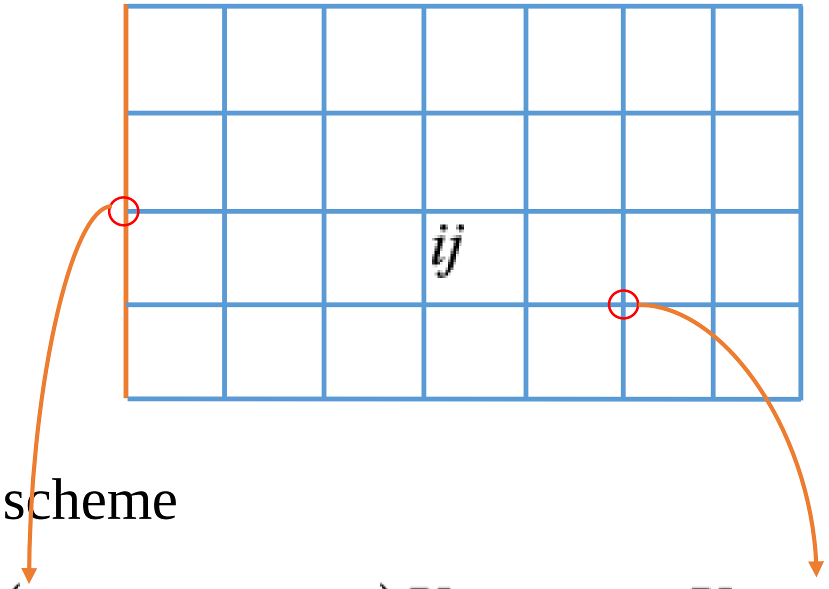


A uniform Cartesian grid



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

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



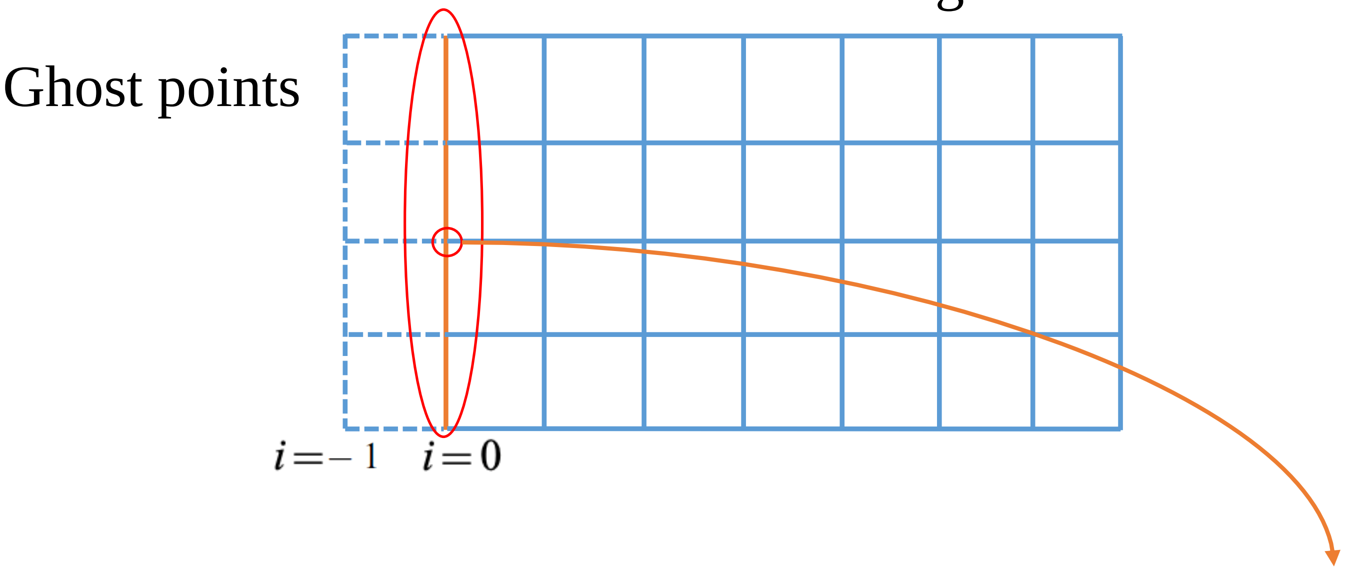
The finite difference scheme

$$\frac{p_{i+\frac{1}{2},j}U_{i+1,j}-(p_{i+\frac{1}{2},j}+p_{i-\frac{1}{2},j})U_{ij}+p_{i-\frac{1}{2},j}U_{i-1,j}}{(h_x)^2}$$

$$+\frac{p_{i,j+\frac{1}{2}}U_{i,j+1}-(p_{i,j+\frac{1}{2}}+p_{i,j-\frac{1}{2}})U_{ij}+p_{i,j-\frac{1}{2}}U_{i,j-1}}{(h_y)^2}-q_{ij}U_{ij}=f_{ij}\quad(3.42)$$

where $p_{i\pm\frac{1}{2},j}=p(x_i\pm h_x/2,y_j)$

A uniform Cartesian grid



Central finite difference
scheme for the flux boundary condition:

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

At $(0,j)$

$$\frac{(p_{-\frac{1}{2},j}+p_{\frac{1}{2},j})U_{1,j}-(p_{\frac{1}{2},j}+p_{-\frac{1}{2},j})U_{0j}}{(h_x)^2}$$

$$+\frac{p_{0,j+\frac{1}{2}}U_{0,j+1}-(p_{0,j+\frac{1}{2}}+p_{0,j-\frac{1}{2}})U_{0j}+p_{0,j-\frac{1}{2}}U_{0,j-1}}{(h_y)^2}$$

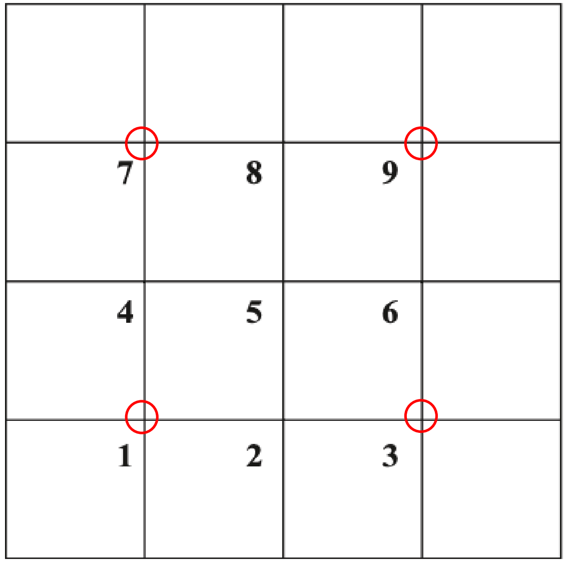
$$-q_{0j}U_{0j}=f_{0j}+\frac{2p_{-\frac{1}{2},j}\,g(y_j)}{h_x}. \qquad (3.43)$$

3.4.1 A Finite Difference Formula for Approximating the Mixed Derivative u_{xy}

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{i,j} = \frac{\left(\frac{\partial u}{\partial y}\right)_{i+1,j} - \left(\frac{\partial u}{\partial y}\right)_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x)^2$$

$$\left(\frac{\partial u}{\partial y}\right)_{i+1,j} = \frac{u_{i+1,j+1} - u_{i+1,j-1}}{2\Delta y} + \mathcal{O}(\Delta y)^2$$

$$\left(\frac{\partial u}{\partial y}\right)_{i-1,j} = \frac{u_{i-1,j+1} - u_{i-1,j-1}}{2\Delta y} + \mathcal{O}(\Delta y)^2$$



$\Rightarrow$ Centered finite difference scheme:

$$u_{xy}(x_i, y_j) \approx \frac{u(x_{i-1}, y_{j-1}) + u(x_{i+1}, y_{j+1}) - u(x_{i+1}, y_{j-1}) - u(x_{i-1}, y_{j+1})}{4h_x h_y}. \tag{3.44}$$

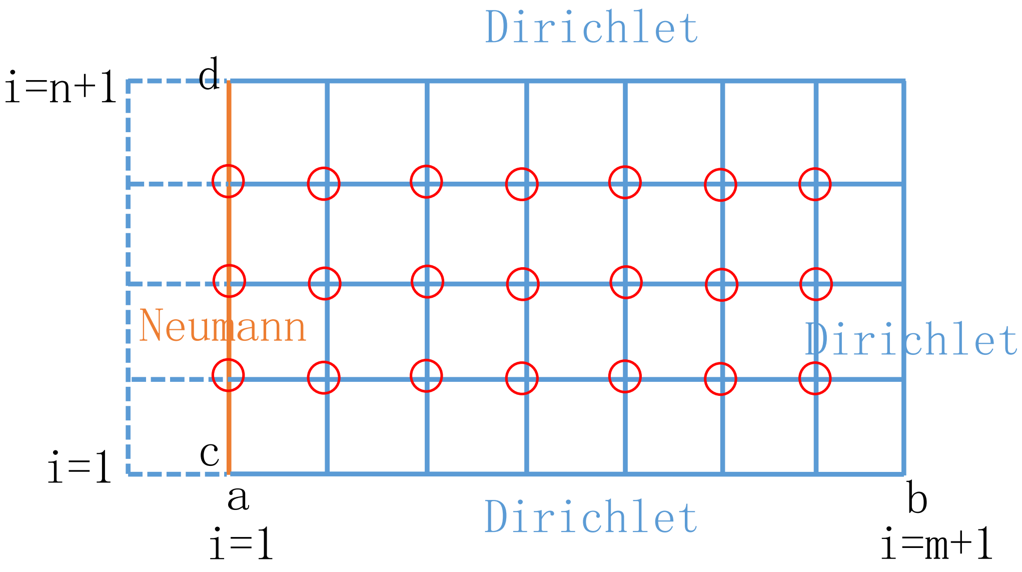
- The discretization is second-order accurate
- The stencil involves nine grid points
- The linear system is no longer diagonally dominant thus is difficult to solve

3.8.1 A Matlab Code for Poisson Equations using A\F

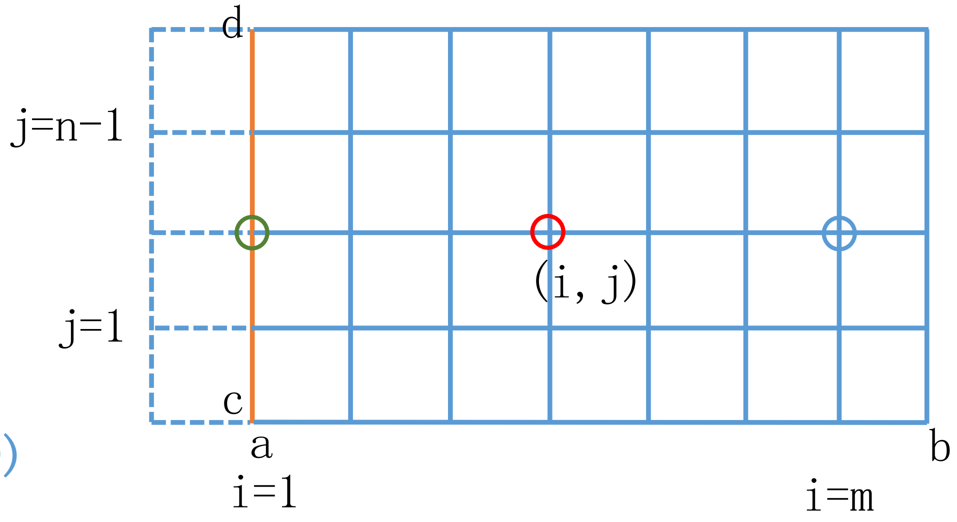
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;  close all
a = 1; b=2; c = -1; d=1;
m=32; n=64;

hx = (b-a)/m; hx1 = hx*hx; x=zeros(m+1,1);
for i=1:m+1,
  x(i) = a + (i-1)*hx;
end
hy = (d-c)/n; hy1 = hy*hy; y=zeros(n+1,1);
for i=1:n+1,
  y(i) = c + (i-1)*hy;
end

M = (n-1)*m; A = sparse(M,M); bf = zeros(M,1);
```



```
for j = 1:n-1,
  for i=1:m,
    k = i + (j-1)*m;
    bf(k) = f(x(i),y(j+1));
    A(k,k) = -2/hx1 -2/hy1;          → (3.19)
    if i == 1
        A(k,k+1) = 2/hx1;
        bf(k) = bf(k) + 2*ux(y(j+1))/hx;          → (3.43)
    else
       if i==m
         A(k,k-1) = 1/hx1;
         bf(k) = bf(k) - ue(x(i+1),y(j+1))/hx1;          → (3.19)
    else
             A(k,k-1) = 1/hx1; A(k,k+1) = 1/hx1;          → (3.19)
    end
  end
```



```
%-- y direction --------------

    if j == 1
        A(k,k+m) = 1/hy1;
        bf(k) = bf(k) - ue(x(i),c)/hy1;            → (3. 19)
    else
       if j==n-1
         A(k,k-m) = 1/hy1;
         bf(k) = bf(k) - ue(x(i),d)/hy1;           → (3. 19)
       else
          A(k,k-m) = 1/hy1; A(k,k+m) = 1/hy1;
       end
     end
  end
end

  U = A \bf;
```

$$\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & & I & B \end{bmatrix} \qquad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -4 \end{bmatrix}_{m \times m}$$

```
%--- Transform back to (i,j) form to plot the solution ---

j = 1;
for k=1:M
  i = k - (j-1)*m ;
  u(i,j) = U(k);
  u2(i,j) = ue(x(i),y(j+1));
  j = fix(k/m) + 1;
end

% Analyze and Visualize the result.

e = max( max( abs(u-u2)))          % The maximum error
x1=x(1:m); y1=y(2:n);

mesh(y1,x1,u); title('The solution plot'); xlabel('y');
ylabel('x'); figure(2); mesh(y1,x1,u-u2); title('The error plot');
xlabel('y'); ylabel('x');
```

y(1) is on the bottom boundary, which is not included here.

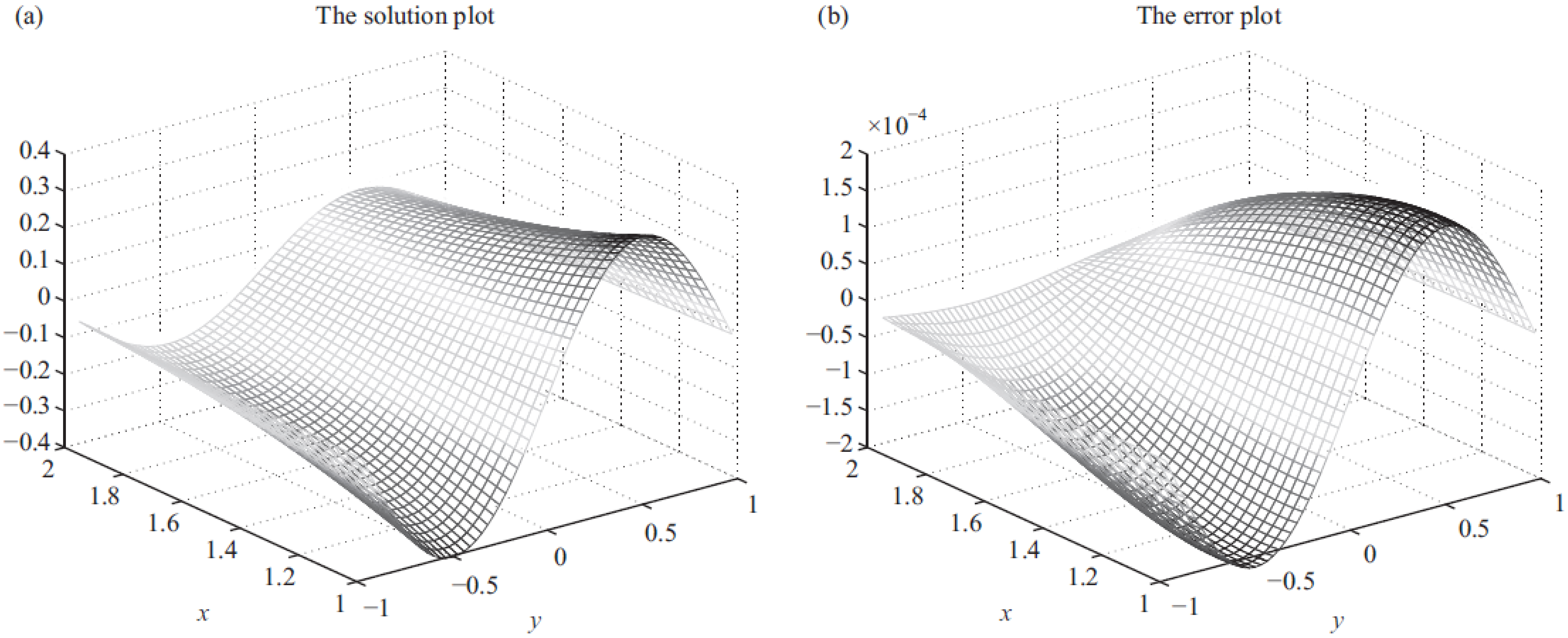


Figure 3.5. (a) The mesh plot of the computed finite difference solution $[1, 2] \times [-1, 1]$ and (b) the error plot. Note that we can see the errors are zeros for Dirichlet boundary conditions, and the errors are not zero for Neumann boundary condition at $x = 1$.

3.5 Solving the Resulting Linear System of Algebraic Equations

$$A\mathbf{U} = \mathbf{F}$$

In general, for an $n+1$ by $n+1$ grid we obtain

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & \end{bmatrix}_{n^2\times n^2}, \quad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & \end{bmatrix}_{n\times n}$$

- For $n = 100$, the $O(10^4 \times 10^4)$ matrix cannot be stored in most modern computers if the desirable double precision is used.
- A is sparse since the nonzero entries are about $O(5n^2)$.

Advantages of iterative methods

- Zero entries play no role in the matrix-vector multiplications
- For some methods, there is no need to manipulate the matrix and vector forms
- Usually less operations than direct methods (LU factorization, Gauss elimination)

$$A\mathbf{x} = b$$

where A is nonsingular ($det(A) \neq 0$), if $A = M - N$ can be written as where M is an invertible matrix, then we have

$$(M - N)\mathbf{x} = b \quad \text{or} \quad \mathbf{x} = M^{-1}N\mathbf{x} + M^{-1}b$$

We may iterate starting from an initial guess $\mathbf{x}^0$,

$$\mathbf{x}^{k+1} = M^{-1}N\mathbf{x}^k + M^{-1}b, \quad k = 0, 1, 2, \dots, \qquad (3.45)$$

the iteration converges or diverges depending on the spectral radius of

$$\rho(M^{-1}N) = \max|\lambda_i(M^{-1}N)|.$$

3.5.1 The Jacobi Iterative Method

The idea of the Jacobi iteration is to solve for the variables on the diagonals and then form the iteration.

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

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

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

It can be written compactly as

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^k\right), \quad i = 1, 2, \ldots, n\,, \qquad (3.46)
$$

For 1D Poisson equation,

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

with Dirichlet boundary conditions $U_0 = ua$ and $U_n = ub$, we have

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

and for a 2D Poisson equation,

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

$$i, j = 1, 2, \ldots, n-1 \text{ assuming } m = n.$$

3.5.2 The Gauss–Seidel Iterative Method

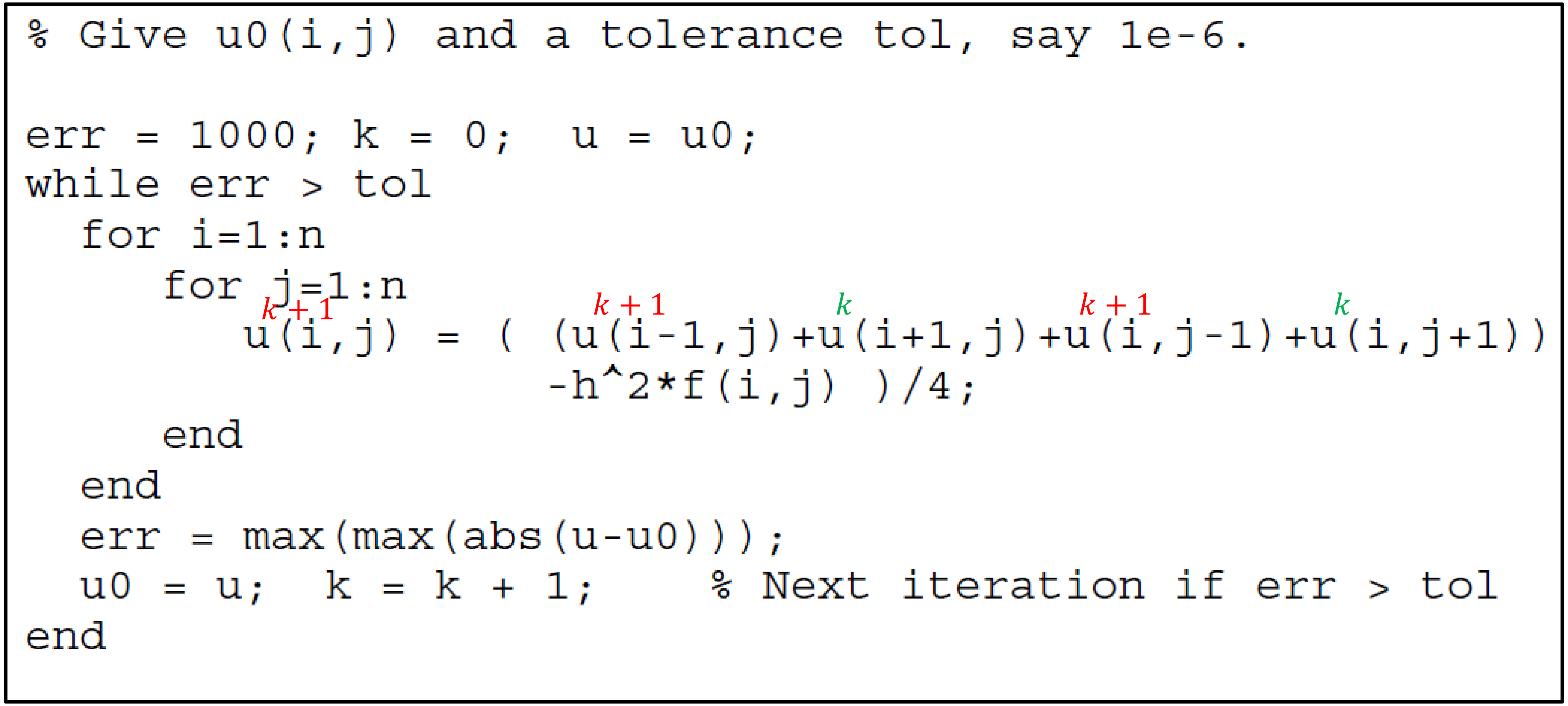
In the Gauss–Seidel iterative method the most updated information is used as follows:

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

or in a compact form

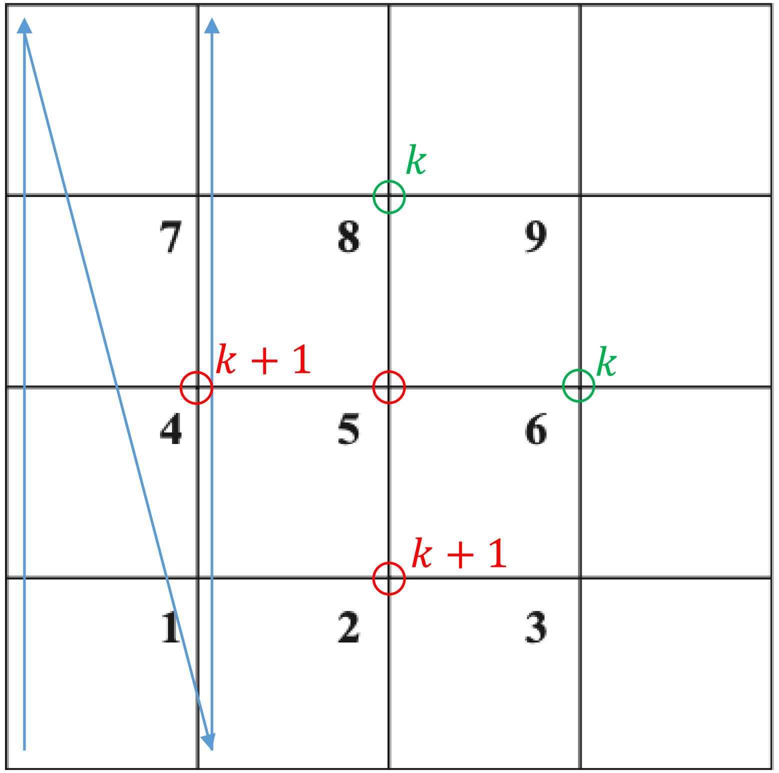
$$
x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k\right), \qquad i = 1, 2, \ldots, n. \qquad (3.47)
$$

A pseudo-code



```
% Give u0(i,j) and a tolerance tol, say 1e-6.

err = 1000; k = 0;  u = u0;
while err > tol
  for i=1:n
     for j=1:n
        $u^{k+1}$(i,j) = ( ($u^{k+1}$(i-1,j)+$u^{k}$(i+1,j)+$u^{k+1}$(i,j-1)+$u^{k}$(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
```


k
7
8
9
k + 1
k
4
5
6
k + 1
1
2
3

3.5.3 The Successive Overrelaxation Method SOR(ω)

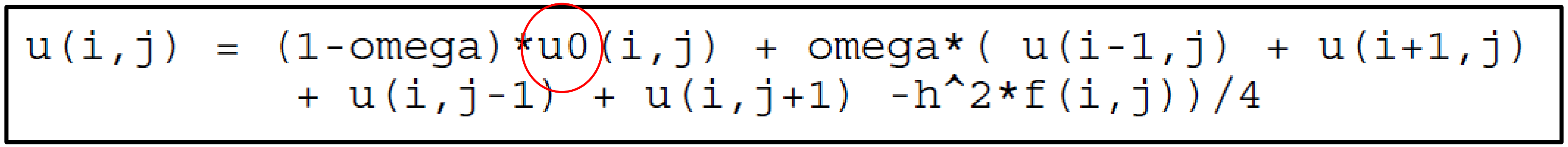
The idea of the successive overrelaxation (SOR(ω)) iteration is based on an extrapolation technique.

$$\mathbf{x}^{k+1} = (1-\omega)\mathbf{x}^{k} + \omega\mathbf{x}_{GS}^{k+1}\,, \qquad (3.48)$$

In component form:

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^{k}\right), \qquad (3.49)$$

A pseudo-code:



```
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
```

u0 is from the solution of last solution, u is the current solution at k+1

The convergence of the SOR(ω) method depends on the choice of ω.

$$\begin{cases} 0 < \omega < 1 & \text{: Interpolation} \\ \omega > 1 & \text{: Extrapolation or over relaxation} \\ \omega = 1 & \text{: the Gauss–Seidel method} \end{cases}$$

For elliptic problems, we usually choose $1 \leq \omega < 2$

For five-point stencil applied to a Poisson equation with $h = h_x = h_y = 1/n$,

$$\omega_{opt} = \frac{2}{1 + \sin(\pi/n)} \sim \frac{2}{1 + \pi/n}, \qquad (3.50)$$

The optimal ω is unknown for general elliptic PDEs, we can use the optimal ω for the Poisson equation as a trial value.

3.5.4 Convergence of Stationary Iterative Methods

Theorem 3.5. *Given a stationary iteration*

$$\mathbf{x}^{k+1} = T\mathbf{x}^k + c, \tag{3.51}$$

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

$$\rho(T) = \max |\lambda_i(T)|, \tag{3.52}$$

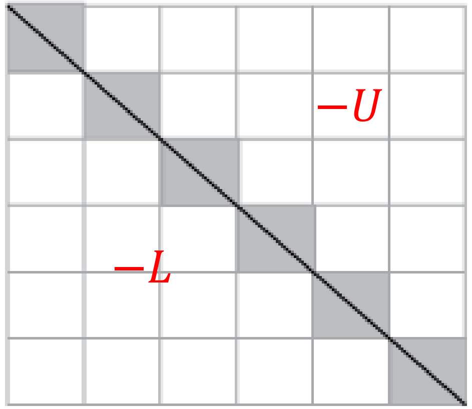
i.e., the largest magnitude of all the eigenvalues of T.

Theorem 3.6. *If there is a matrix norm $\|\cdot\|$ such that $\|T\| < 1$, then the stationary iterative method converges for* $\boxed{\textit{arbitrary initial guess } \mathbf{x}^0}$.

We often check whether $\|T\|_p < 1$ for $p = 1, 2, \infty$, and if there is just one norm such that $\|T\| < 1$, then the iterative method is convergent. However, if $\|T\| \geq 1$ there is no conclusion about the convergence.

Convergence of the Jacobi, Gauss–seidel, and SOR(ω) Methods

$$A = D - L - U$$



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

Theorem 3.7. *If A is strictly row diagonally dominant, i.e.,*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|\,, \tag{3.53}$$

then both the Jacobi and Gauss–Seidel iterative methods converge. The conclusion is also true when (1): A is weakly row diagonally dominant

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|; \tag{3.54}$$

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

For an elliptic PDE defined on a rectangle domain or a disk

- Simple iterative methods such as Jacobi, Gauss–Seidel, SOR(ω)
- Fast Poisson solvers such as the fast Fourier transform (FFT) or cyclic reduction
- Multigrid solvers, either geometric multigrid or algebraic multigrid
- Gradient descent method
- Krylov subspace methods such as the conjugate gradient (CG) or preconditioned conjugate gradient (PCG), generalized minimized residual (GMRES), biconjugate gradient (BICG) method for nonsymmetric system of equations.

Gradient descent method

Solving a linear system

$$Ax^* = b$$

→

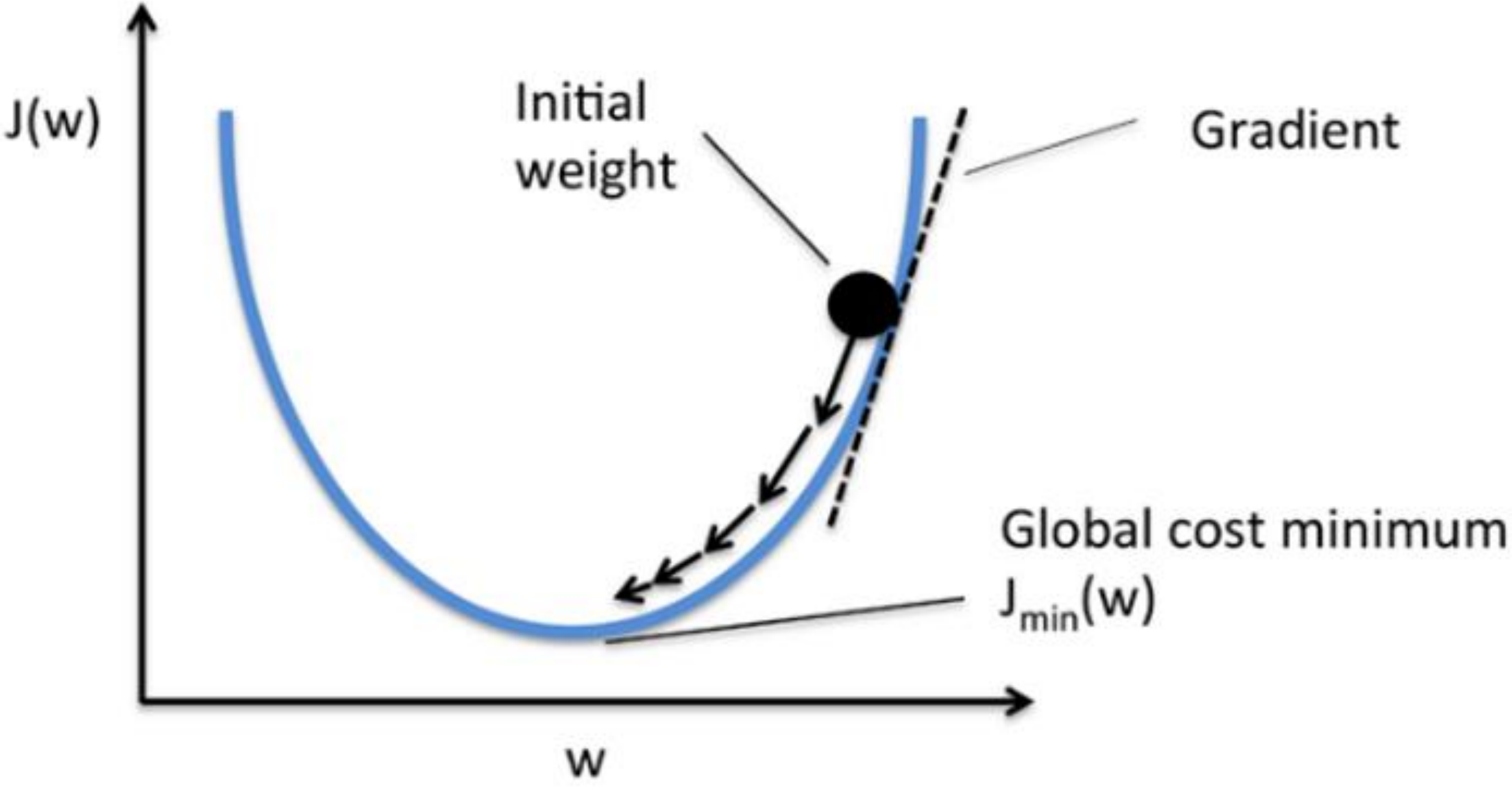
Finding minimum of a function

$$f(x) = \frac{1}{2} x^T A x - x^T b \, .$$

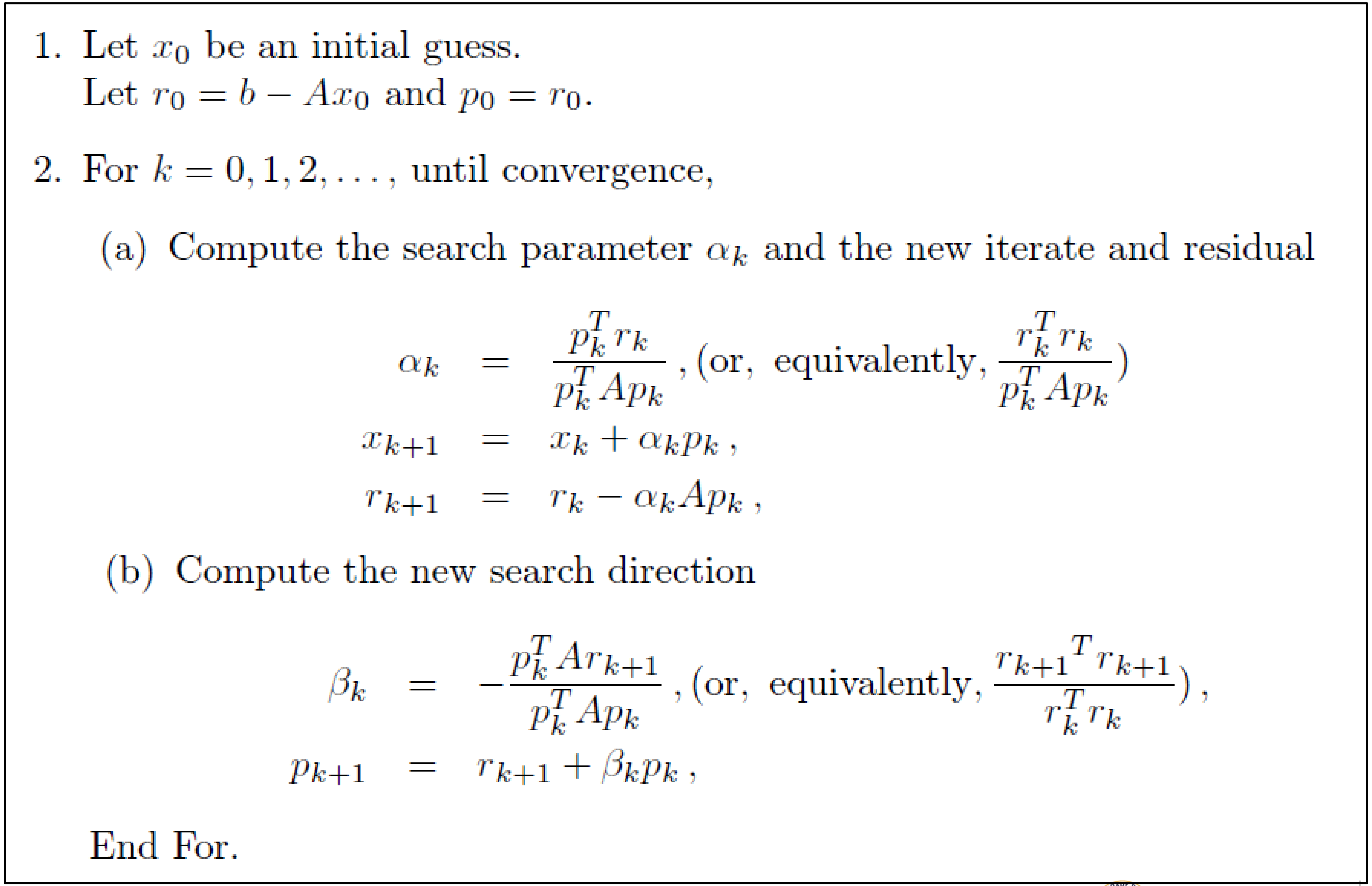
$$\min_x f(x)$$

```
1. Pick $x_0$ .
2. For $k = 0, 1, \ldots,$
    (a) Evaluate $p_k = -\nabla f(x_k) = r_k$.
    (b) Let $x_{k+1} = x_k + \alpha_k p_k$, where $\alpha_k$ is the minimizer of $\min_\alpha f(x_k + \alpha p_k)$.
   End For.
```

Gradient descent method



The Conjugate Gradient Algorithm



1. Let x_0 be an initial guess.
   Let $r_0 = b - Ax_0$ and $p_0 = r_0$.

2. For $k = 0, 1, 2, \ldots,$ until convergence,

   (a) Compute the search parameter α_k and the new iterate and residual

   $$\begin{aligned} \alpha_k &= \frac{p_k^T r_k}{p_k^T A p_k}\,, \text{(or, equivalently, } \frac{r_k^T r_k}{p_k^T A p_k}) \\ x_{k+1} &= x_k + \alpha_k p_k\,, \\ r_{k+1} &= r_k - \alpha_k A p_k\,, \end{aligned}$$

   (b) Compute the new search direction

   $$\begin{aligned} \beta_k &= -\frac{p_k^T A r_{k+1}}{p_k^T A p_k}\,, \text{(or, equivalently, } \frac{{r_{k+1}}^T r_{k+1}}{r_k^T r_k})\,, \\ p_{k+1} &= r_{k+1} + \beta_k p_k\,, \end{aligned}$$

   End For.

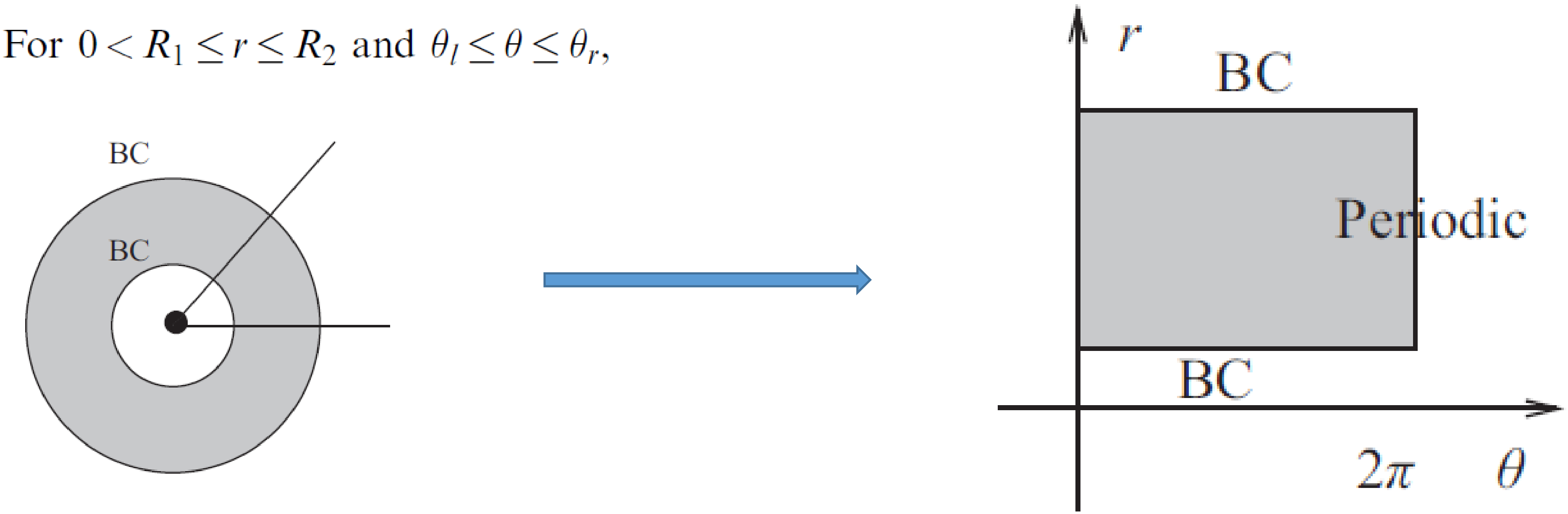
Reference (On UMMoodle):

Dianne P. O'Leary, Notes on Some Methods for Solving Linear Systems.

3.7 A Finite Difference Method for Poisson Equations in Polar Coordinates

$$u_{xx}+u_{yy}=f \xrightarrow{x=r\cos\theta,\, y=r\sin\theta} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}=f(r,\theta)$$

For $0<R_1\le r\le R_2$ and $\theta_l\le\theta\le\theta_r$,



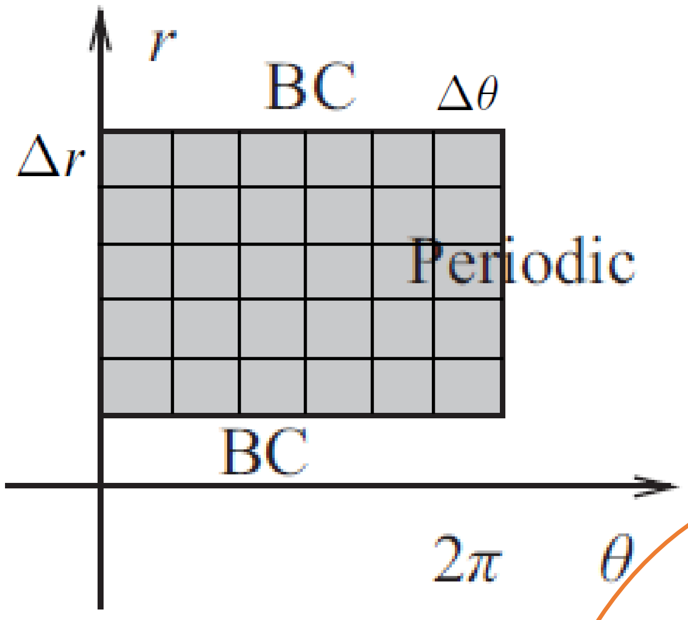
Poisson Equations in Polar Coordinates

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}=f(r,\theta)$$

Using a uniform grid:

$$r_i=R_1+i\Delta r,\quad i=0,1,\ldots,m,\quad \Delta r=\frac{R_2-R_1}{m},$$

$$\theta_j=\theta_l+j\Delta\theta,\quad j=0,1,\ldots,n,\quad \Delta\theta=\frac{\theta_r-\theta_l}{n},$$



The central finite difference scheme:

$$(p(x)u'(x))'$$

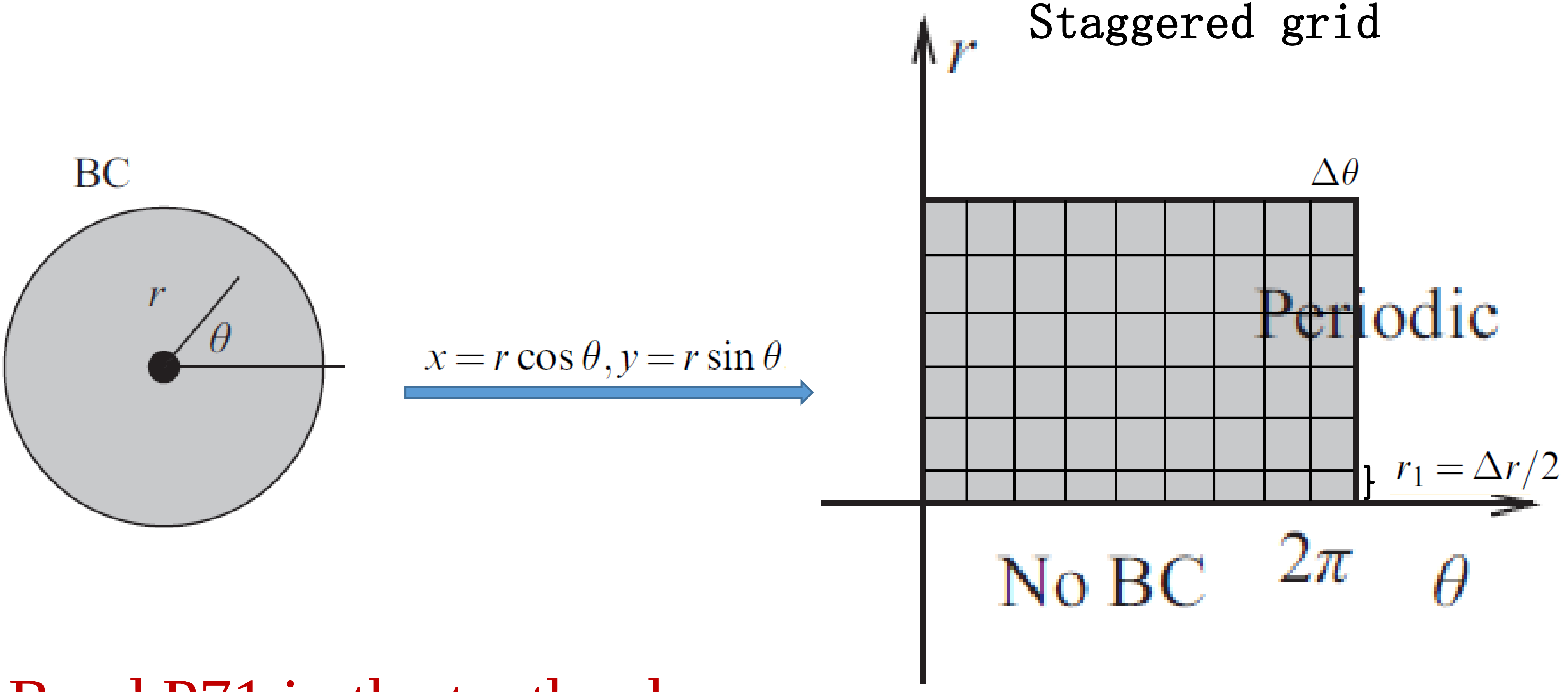
$$\downarrow$$

$$\frac{p_{i+\frac{1}{2}}\frac{u(x_{i+1})-u(x_i)}{h}-p_{i-\frac{1}{2}}\frac{u(x_i)-u(x_{i-1})}{h}}{h}$$

$$\frac{1}{r_i}\frac{r_{i-\frac{1}{2}}U_{i-1,j}-(r_{i-\frac{1}{2}}+r_{i+\frac{1}{2}})U_{ij}+r_{i+\frac{1}{2}}U_{i+1,j}}{(\Delta r)^2}$$

$$+\frac{1}{r_i^2}\frac{U_{i,j-1}-2U_{ij}+U_{i,j+1}}{(\Delta\theta)^2}=f(r_i,\theta_j)\,, \qquad (3.58)$$

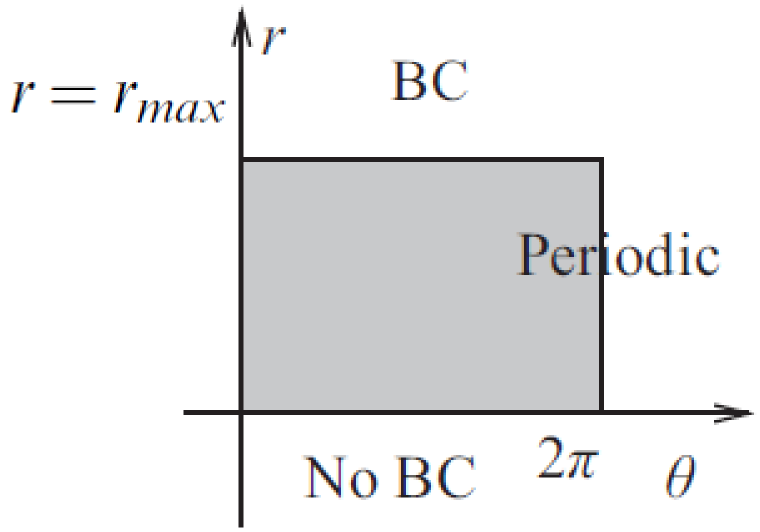
3.7.1 Treating the Polar Singularity



Read P71 in the textbook

3.7.2 Using the FFT to Solve Poisson Equations in Polar Coordinates

PDE
$$\begin{cases} \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u}{\partial r}\right)+\dfrac{1}{r^2}\dfrac{\partial^2 u}{\partial \theta^2}=f(r,\theta) \\ u(r_{max},\theta)=u^{BC}(\theta) \text{ at } r=r_{max} \end{cases}$$



1. Approximate u by the truncated Fourier series

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

2. Substitute into the Poisson equation

4. Substitute back to the Fourier series

ODE
$$\begin{cases} \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(\dfrac{1}{r}\dfrac{\partial u_n}{\partial r}\right)-\dfrac{n^2}{r^2}u_n=f_n(r), \quad n=-N/2,\ldots,N/2-1, \\ u_n^{BC}(r_{max})=\dfrac{1}{N}\displaystyle\sum_{k=0}^{N-1}u^{BC}(\theta)e^{-ink\theta} \quad \text{at } r=r_{max}, \end{cases}$$

$\Rightarrow u_n$

3.Solve the ODE system