# Finite Difference Methods (FDMs) 1

# $1^{st}$-order Approximation

Recall Taylor series expansion:

$$u(x+\Delta x) = u(x) + \frac{\partial u}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial^2 u}{\partial x^2}(\Delta x)^2 + \dots$$

Forward difference:

$$\frac{\partial u}{\partial x} = \frac{u(x+\Delta x) - u(x)}{\Delta x} + \mathcal{O}(\Delta x)$$

Backward difference:

$$\frac{\partial u}{\partial x} = \frac{u(x) - u(x-\Delta x)}{\Delta x} + \mathcal{O}(\Delta x)$$

Central difference:

$$\frac{\partial u}{\partial x} = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

# $2^{nd}$-order Approximation

Forward difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+2\Delta x) - 2u(x+\Delta x) + u(x)}{(\Delta x)^2} + \mathcal{O}(\Delta x)$$

Backward difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x) - 2u(x-\Delta x) + u(x-2\Delta x)}{(\Delta x)^2} + \mathcal{O}(\Delta x)$$

Central difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} + \mathcal{O}(\Delta x^2)$$

# 2D Elliptic PDEs

The general elliptic problem that is faced in 2D is to solve

$$-\nabla^2 u + a u_x + b u_y + c u = f \quad \text{in } D \qquad \textbf{(14.1)}$$

where

$-\nabla^2 u \equiv \mathrm{div}(\mathrm{grad}\, u) \equiv u_{xx} + u_{yy}$ is the Laplacian

$u_x$ is gradient of $u$ in $x$ direction

$u_y$ is gradient of $u$ in $y$ direction

Equation (14.1) is to be solved on some bounded domain $D$ in 2-dimensional Euclidean space $\mathbb{R}^2$ with boundary $\Gamma$ that has conditions

$$\frac{\partial u}{\partial n} + \alpha u = g(x, y) \quad \text{on } \Gamma \qquad \textbf{(14.2)}$$

# 2D Poisson Equation (Dirichlet Problem)

The 2D Poisson equation is given by

$$-\nabla^2 u = f \tag{14.3}$$

with boundary conditions

$$u(x,y)|_\Gamma = g(x,y) \tag{14.4}$$

There is no initial condition, because the equation does not depend on time, hence it becomes a boundary value problem.

Suppose that the domain is $D = (a,b) \times (c,d)$ and equation (14.3) is to be solved in $D$ subject to Dirichlet boundary conditions. The domain is covered by a square grid of size

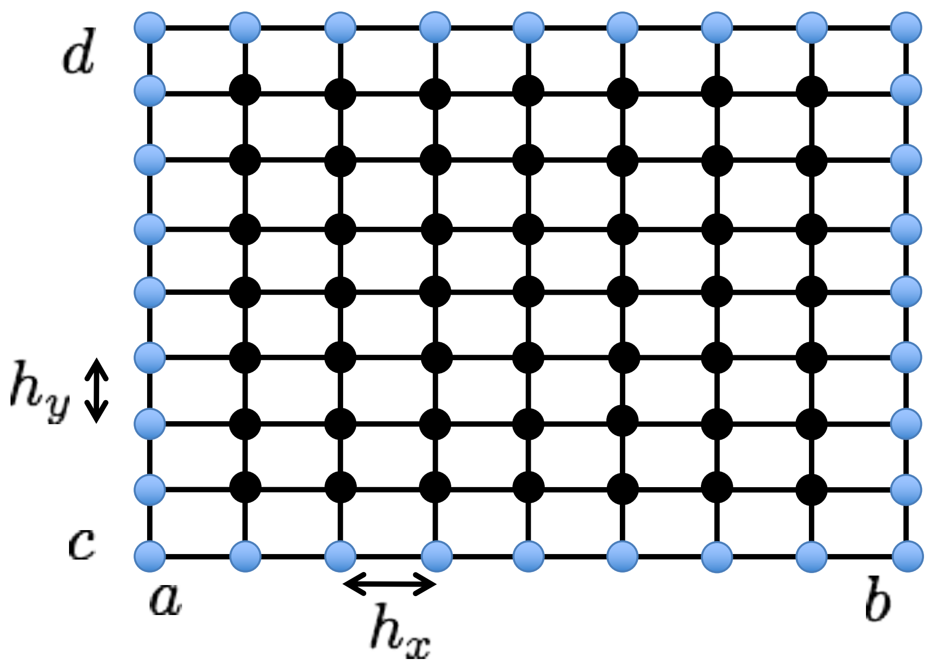
$$h_x \times h_y.$$

# 2D Poisson Equation (Dirichlet Problem)

**Step 1**: Generate a grid. For example, a uniform Cartesian grid can be generated as

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

$$y_k = c + kh_y\,, \quad k = 0, 1, 2, \ldots, N\,, \quad h_y = \frac{d-c}{N}$$



# 2D Poisson Equation (Dirichlet Problem)

Generating a grid in MATLAB:

```
% Define domain
a = 0; b = 1;
c = 0; d = 1;

% Define grid sizes
M = 5; % number of points
N = 5; % number of points
hx = (b-a)/(M-1); % length of sub-intervals in x-axis
hy = (d-c)/(N-1); % length of sub-intervals in y-axis

% Generate 2D arrays of grids
[X,Y] = meshgrid(a:hx:b,c:hy:d);
```

# 2D Poisson Equation (Dirichlet Problem)

We denote by $U$ a grid function whose value $U_{i,k}$ at a typical point $(x_i, y_k)$ in domain $D$ is intended to approximate the exact solution $u(x_i, y_k) \equiv u_{i,k}$ at that point.

The solution at the boundary nodes (blue dots) is known from the boundary conditions (BCs) and the solution at the internal grid points (black dots) are to be approximated.

Step 2: At a typical internal grid point $(x_i, y_k)$ we approximate the partial derivatives of $u$ by second order central difference,, which is second order accurate since the remainder term is $\mathcal{O}(h^2)$.

$$u_{xx} \approx \frac{u(x_{i-1}, y_k) - 2u(x_i, y_k) + u(x_{i+1}, y_k)}{(h_x)^2}$$

# 2D Poisson Equation (Dirichlet Problem)

$$u_{yy} \approx \frac{u(x_i, y_{k-1}) - 2u(x_i, y_k) + u(x_i, y_{k+1})}{(h_y)^2}$$

Then the Poisson equation is approximated as

$$-\left(\frac{u(x_{i-1}, y_k) - 2u(x_i, y_k) + u(x_{i+1}, y_k)}{(h_x)^2} + \frac{u(x_i, y_{k-1}) - 2u(x_i, y_k) + u(x_i, y_{k+1})}{(h_y)^2}\right) = f_{i,k} + T_{i,k}\,, \qquad \textbf{(14.5)}$$

$$i = 1, 2, \ldots, M-1, \qquad k = 1, 2, \ldots, N-1$$

where $f_{i,k} = f(x_i, y_k)$.

# 2D Poisson Equation (Dirichlet Problem)

The local truncation error satisfies

$$T_{i,k} \sim \frac{(h_x)^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i, y_k) + \frac{(h_y)^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, y_k) + \mathcal{O}(h^4)$$

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

The finite difference discretization is consistent if

$$\lim_{h \to 0} \|T\| = 0$$

We ignore the error term and replace the exact solution values $u(x_i, y_k)$ at the grid points with the approximate solution values $U_{i,k}$, that is

# 2D Poisson Equation (Dirichlet Problem)

$$-\left(\frac{U_{i-1,k}+U_{i+1,k}}{(h_x)^2}+\frac{U_{i,k-1}+U_{i,k+1}}{(h_y)^2}\right) + \left(\frac{2}{(h_x)^2}+\frac{2}{(h_y)^2}\right)U_{i,k}=f_{i,k} \qquad \textbf{(14.6)}$$

$$i=1,2,\ldots,M-1, \qquad k=1,2,\ldots,N-1$$

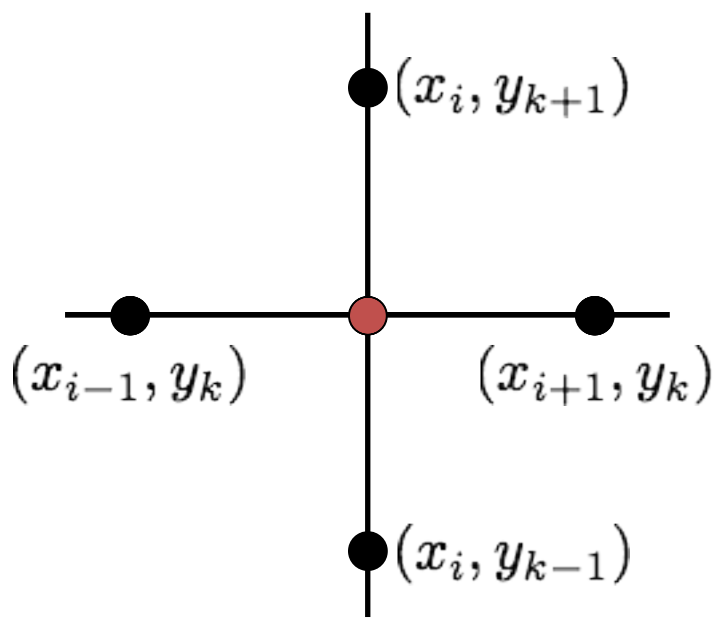
The finite difference equation at the grid point $(x_i, y_k)$ involves five grid points in a five-point stencil: $(x_{i-1}, y_k)$, $(x_{i+1}, y_k)$, $(x_i, y_{k-1})$, $(x_i, y_{k+1})$, and $(x_i, y_k)$. The center $(x_i, y_k)$ is called the master grid point, where the finite difference equation is used to approximate the PDE.

# 2D Poisson Equation (Dirichlet Problem)

Suppose that $h_x = h_y = h$, then equation (14.6) can be rearranged into

$$4U_{i,k} - U_{i-1,k} - U_{i,k-1} - U_{i+1,k} - U_{i,k+1} = h^2 f_{i,k} \tag{14.7}$$

The difference replacement (14.7) is depicted as in the five-point stencil:



# 2D Poisson Equation (Dirichlet Problem)

**Step 3**: Solve the linear system of algebraic equations (14.7) to get the approximate values for the solution at all grid points.

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

The linear system of equations (14.7) will transform into a matrix-vector form:

$$A\mathbf{U} = \mathbf{F} \qquad \textbf{(14.8)}$$

where, from 2D Poisson equations, the unknowns $U_{i,k}$ are a 2D array which we will order into a 1D array.

# 2D Poisson Equation (Dirichlet Problem)

While it is conventional to represent systems of linear algebraic equations in matrix-vector form, it is not always necessary to do so. The matrix-vector format is useful for explanatory purposes and (usually) essential if a direct linear equation solver is to be used, such as Gaussian elimination or LU factorization.

For particularly large systems, iterative solution methods are more efficient and these are usually designed so as not to require the construction of a coefficient matrix but work directly with approximation (14.7).

Since $h_x = h_y = h$ and suppose that $M = N = 5$. Hence, there are 9 equations and 9 unknowns. The first decision to be taken is how to organize the unknowns $U_{i,k}$ into a vector.

# 2D Poisson Equation (Dirichlet Problem)

The equations are ordered in the same way as the unknowns so that each row of the matrix of coefficients representing the left of (14.7) will contain at most 5 non-zero entries with the coefficient 4 appearing on the diagonal.

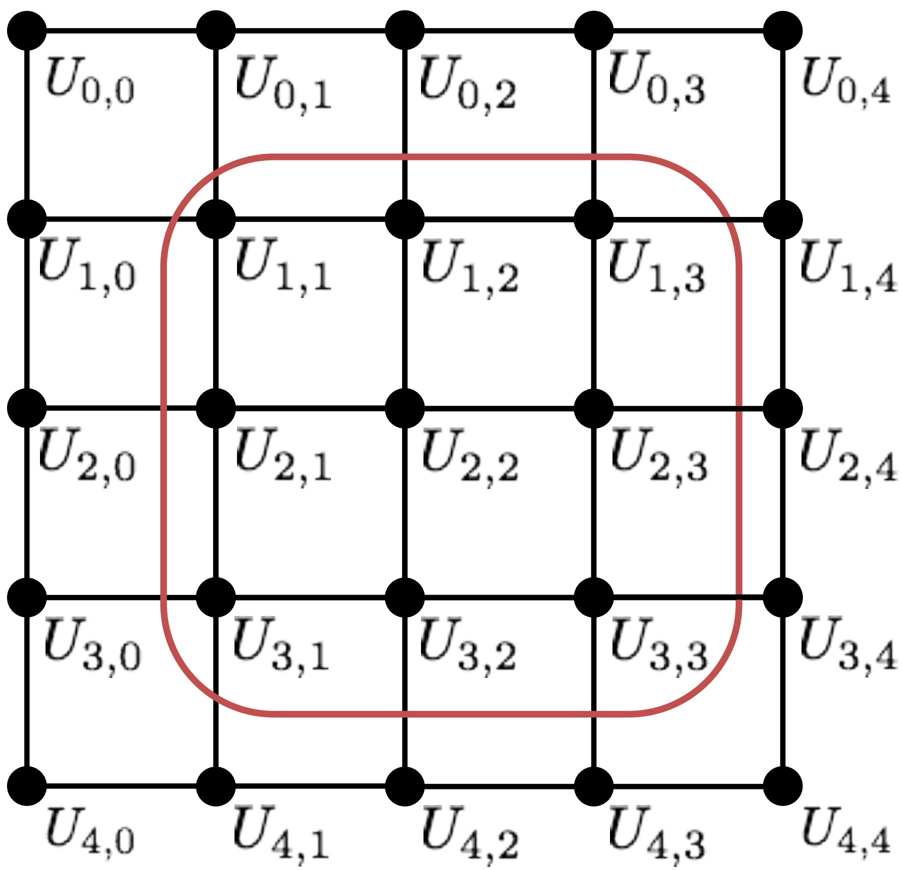
When (14.7) is applied at points adjacent to the boundary, one or more of the neighboring values will be known from the BCs and the corresponding term will be moved to the right side of the equations. For example, when $i = k = 1$:

$$4U_{1,1} - U_{1,2} - U_{2,1} = h^2 f_{1,1} + U_{1,0} + U_{0,1}$$

The values of $U_{1,0}$ and $U_{0,1}$ are known from the BCs, hence they are on the right side of the equations. Then the first row of the matrix will contain only three non-zero entries.

# 2D Poisson Equation (Dirichlet Problem)

Inner grid points are to be approximated




$$\mathbf{U}_1 = \begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{3,1} \end{bmatrix} \quad \mathbf{U}_2 = \begin{bmatrix} U_{1,2} \\ U_{2,2} \\ U_{3,2} \end{bmatrix}$$

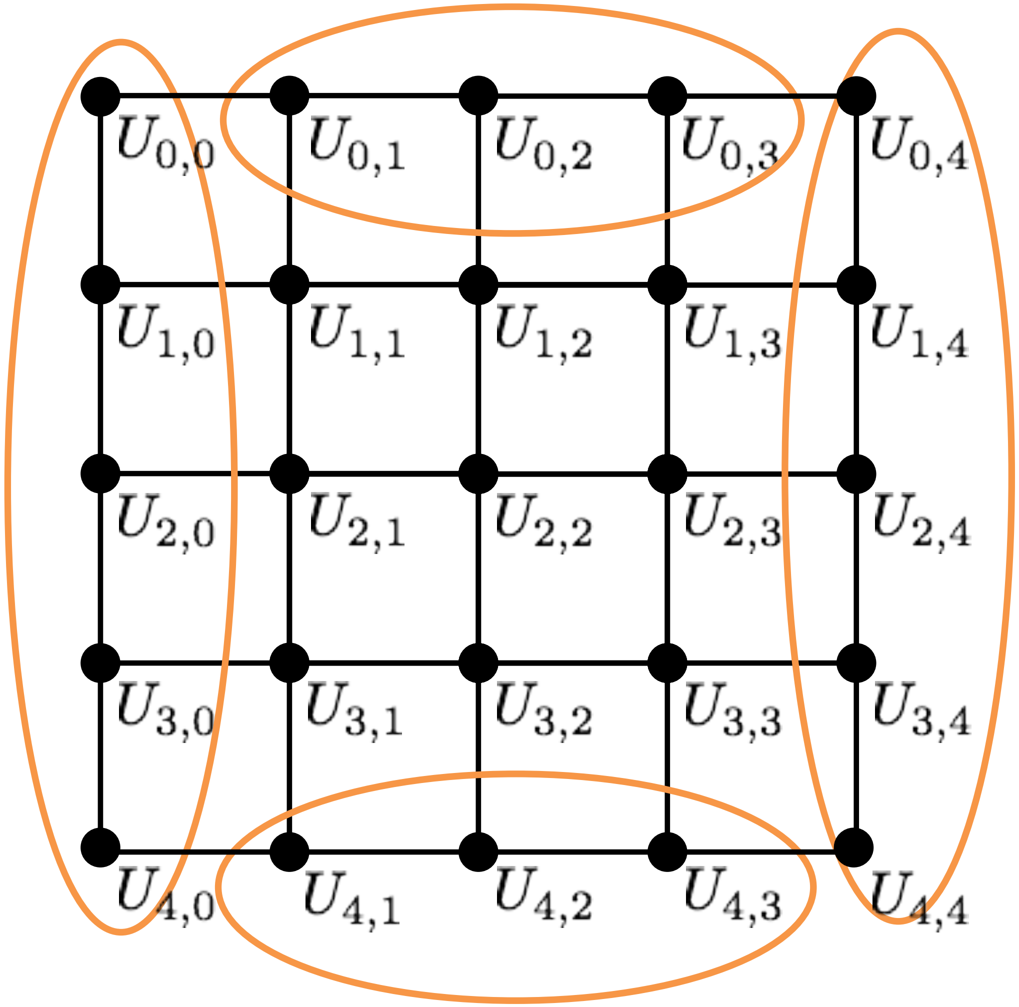
$$\mathbf{U}_3 = \begin{bmatrix} U_{1,3} \\ U_{2,3} \\ U_{3,3} \end{bmatrix}$$

Arranging into a vector:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \end{bmatrix}$$

# 2D Poisson Equation (Dirichlet Problem)

Outer grid points are known from the boundary conditions.



# 2D Poisson Equation (Dirichlet Problem)

Working on the first column of the inner grid points gives us

$$i=1,k=1:\ 4U_{1,1}-U_{1,2}-U_{2,1}=h^2 f_{1,1}+U_{1,0}+U_{0,1}$$

$$i=2,k=1:\ 4U_{2,1}-U_{1,1}-U_{3,1}-U_{2,2}=h^2 f_{2,1}+U_{2,0}$$

$$i=3,k=1:\ 4U_{3,1}-U_{2,1}-U_{3,2}=h^2 f_{3,1}+U_{3,0}+U_{4,1}$$

Arranging in matrix-vector form yields

$$\begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{3,1} \end{bmatrix}+\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} U_{1,2} \\ U_{2,2} \\ U_{3,2} \end{bmatrix}=h^2\begin{bmatrix} f_{1,1} \\ f_{2,1} \\ f_{3,1} \end{bmatrix}+\begin{bmatrix} U_{1,0}+U_{0,1} \\ U_{2,0} \\ U_{3,0}+U_{4,1} \end{bmatrix}$$

# 2D Poisson Equation (Dirichlet Problem)

The second column of the inner grid points gives

$$i = 1, k = 2: \; 4U_{1,2} - U_{2,2} - U_{1,1} - U_{1,3} = h^2 f_{1,2} + U_{0,2}$$

$$i = 2, k = 2: \; 4U_{2,2} - U_{1,2} - U_{3,2} - U_{2,1} - U_{2,3} = h^2 f_{2,2}$$

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

Arranging in matrix-vector form yields:

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{3,1} \end{bmatrix} + \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{1,2} \\ U_{2,2} \\ U_{3,2} \end{bmatrix}$$

$$+ \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} U_{1,3} \\ U_{2,3} \\ U_{3,3} \end{bmatrix} = h^2 \begin{bmatrix} f_{1,2} \\ f_{2,2} \\ f_{3,2} \end{bmatrix} + \begin{bmatrix} U_{0,2} \\ 0 \\ U_{4,2} \end{bmatrix}$$

# 2D Poisson Equation (Dirichlet Problem)

The third column gives

$$i = 1, k = 3 \quad 4U_{1,3} - U_{2,3} - U_{1,2} = h^2 f_{1,3} + U_{0,3} + U_{1,4}$$

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

$$i = 3, k = 3 : 4U_{3,3} - U_{2,3} - U_{3,2} = h^2 f_{3,3} + U_{4,3} + U_{3,4}$$

Arranging in matrix-vector form yields

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} U_{1,2} \\ U_{2,2} \\ U_{3,2} \end{bmatrix} + \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{1,3} \\ U_{2,3} \\ U_{3,3} \end{bmatrix} = h^2 \begin{bmatrix} f_{1,3} \\ f_{2,3} \\ f_{3,3} \end{bmatrix} + \begin{bmatrix} U_{0,3} + U_{1,4} \\ U_{2,4} \\ U_{4,3} + U_{3,4} \end{bmatrix}$$

# 2D Poisson Equation (Dirichlet Problem)

Top BC

Bottom BC

Left BC

$$
\begin{bmatrix}
\begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & \\
\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \\
 & \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}
\end{bmatrix}
\begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{3,1} \\ U_{1,2} \\ U_{2,2} \\ U_{3,2} \\ U_{1,3} \\ U_{2,3} \\ U_{3,3} \end{bmatrix}
= h^2 \begin{bmatrix} f_{1,1} \\ f_{2,1} \\ f_{3,1} \\ f_{1,2} \\ f_{2,2} \\ f_{3,2} \\ f_{1,3} \\ f_{2,3} \\ f_{3,3} \end{bmatrix}
+ \begin{bmatrix} U_{1,0} + U_{0,1} \\ U_{2,0} \\ U_{3,0} + U_{4,1} \\ U_{0,2} \\ 0 \\ U_{4,2} \\ U_{1,4} + U_{0,3} \\ U_{2,4} \\ U_{3,4} + U_{4,3} \end{bmatrix}
$$

Right BC

# 2D Poisson Equation (Dirichlet Problem)

For general case, we focus on the equations on the *i*-th column of the grid. Since the unknowns in this column are linked only to unknowns on the two neighboring columns, these can be expressed as

$$-\mathbf{U}_{i-1} + B\mathbf{U}_i - \mathbf{U}_{i+1} = h^2\mathbf{f}_i + \mathbf{g}_i$$

where $B$ is the $(M-2)\times(M-2)$ tridiagonal matrix,

$$B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & & \\ & & & -1 & 4 \end{bmatrix}_{(M-2)\times(M-2)}$$

# 2D Poisson Equation (Dirichlet Problem)

Methods to generate tridiagonal matrix in MATLAB

```
% (1) Use the "for" loop
D = zeros(M,M);
D(1,1) = beta;
D(1,2) = -alpha;
for i=2:M-1
    D(i,i) = beta;
    D(i,i-1) = -alpha;
    D(i,i+1) = -alpha;
end
D(M,M-1) = -alpha;
D(M,M) = beta;

% (2) Use the built-in function "diag"
r2 = beta*ones(M,1);
r = -alpha*ones(M-1,1);
A = diag(r2,0) + diag(r(1:M-1),1) + diag(r(1:M-1),-1);
% or
s2 = beta*ones(M,1);
s = -alpha*ones(M-1,1);
B = diag(s2,0) + diag(s,1) + diag(s,-1);
```

# 2D Poisson Equation (Dirichlet Problem)

The vector

$$\mathbf{g}_i = \begin{bmatrix} g_{0,i} \\ 0 \\ \vdots \\ 0 \\ g_{M,i} \end{bmatrix}$$

arises from the top and bottom boundaries. Also, when $i = 1$ or $M - 1$, boundary conditions from the vertical edges are applied, so that

$$\mathbf{U}_0 \equiv \mathbf{g}_0 = \begin{bmatrix} g_{0,1} \\ g_{0,2} \\ \vdots \\ g_{0,M-1} \end{bmatrix} \qquad \mathbf{U}_M \equiv \mathbf{g}_M = \begin{bmatrix} g_{M,1} \\ g_{M,2} \\ \vdots \\ g_{M,M-1} \end{bmatrix}$$

# 2D Poisson Equation (Dirichlet Problem)

The difference equation can now be expressed as a system of the form

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

where $A$ is a $(M-2)^2 \times (M-2)^2$ matrix and the unknowns and the right hand side vector $\mathbf{U}, \mathbf{F} \in \mathbb{R}^{(M-2)^2}$.

$A$ has the tridiagonal matrix structure:

$$A = \frac{1}{h^2}\begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & & \\ & & & -I & B \end{bmatrix}_{(M-2)^2\times(M-2)^2}$$

# 2D Poisson Equation (Dirichlet Problem)

where $I$ is the $(M-2) \times (M-2)$ identity matrix, and the right hand side vector $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 + (\mathbf{g}_0 + \mathbf{g}_1)/h^2 \\ \mathbf{f}_2 + \mathbf{g}_2/h^2 \\ \vdots \\ \mathbf{f}_{M-2} + \mathbf{g}_{M-2}/h^2 \\ \mathbf{f}_{M-1} + (\mathbf{g}_{M-1} + \mathbf{g}_M)/h^2 \end{bmatrix}_{((M-2)^2,1)}$$

**It is essential to store matrices *A*, *B*, and *I* as "sparse" matrices**, only the non-zero entries are stored.

# 2D Poisson Equation (Dirichlet Problem)

Generating matrices $B$ and $A$ in MATLAB

```
% Build matrix B
r2 = 2*ones(M,1);
r = -ones(M-1,1);
B = diag(r2,0) + diag(r,1) + diag(r,-1);

% Sparse matrix B
B = sparse(B);

% Build sparse identity matrix
I = speye(M-1);

% Build tridiagonal block matrix A
A = kron(B,I) + kron(I,B);
```

# Example

Solve the BVP defined by:

$$-\nabla^2 u = 20\cos(3\pi x)\sin(2\pi y)$$

on a unit square with boundary conditions:

$$u(0, y) = y^2$$

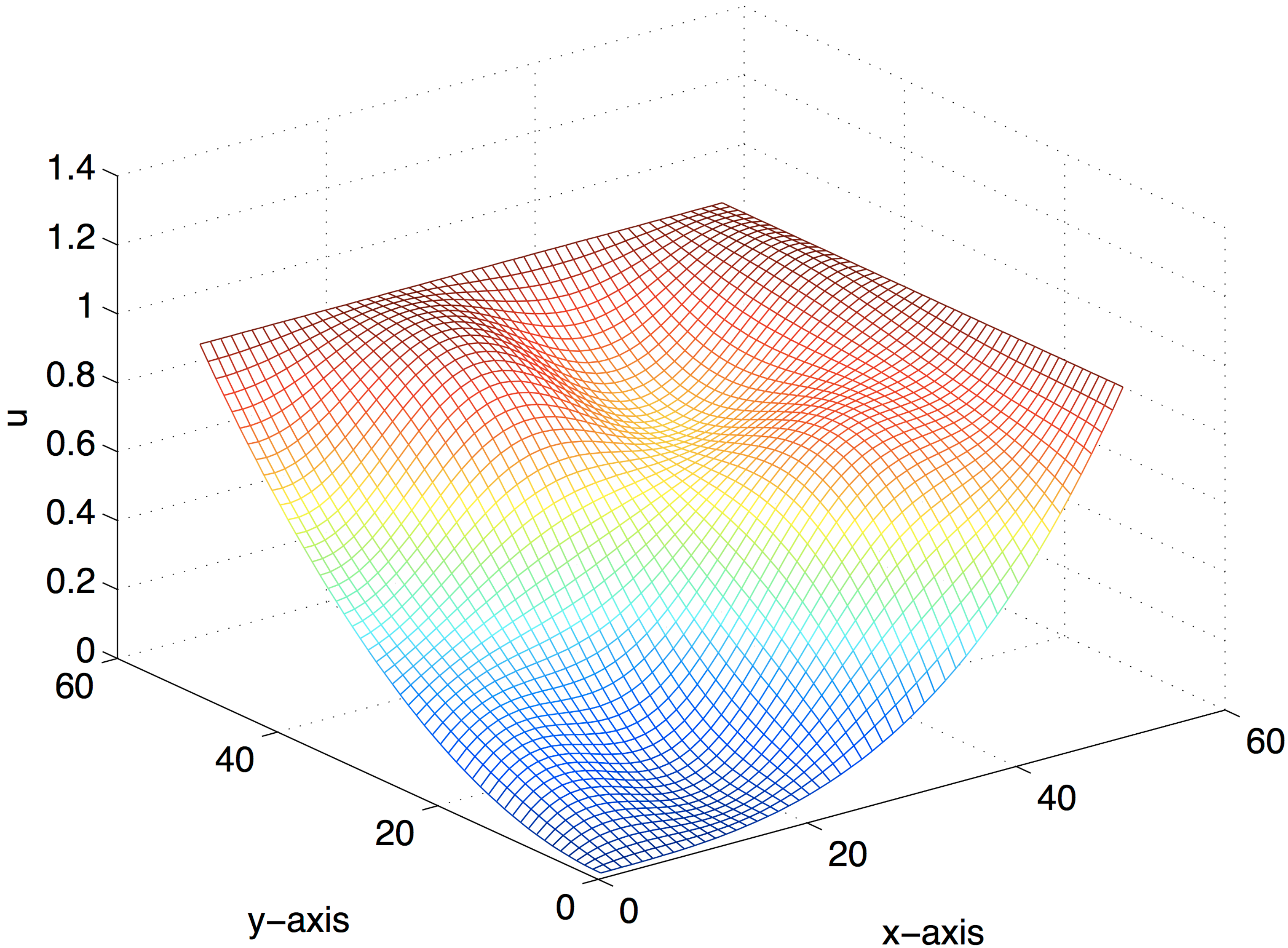
$$u(1, y) = 1$$

$$u(x, 0) = x^3$$

$$u(x, 1) = 1$$

using second order approximation (central finite difference).

# Example



# 2D Poisson Equation

The right hand side function in MATLAB

```
function f = poisson_rhs(X,Y)

f = 20*cos(3*pi*X).*sin(2*pi*Y);
```

Dirichlet BCs in MATLAB

```
function G = poisson_bcs(X,Y,M)

G(:,1) = Y(:,1).^2;      % left
G(:,M+1) = ones(M+1,1); % right
G(1,:)   = X(1,:).^3;   % bottom
G(M+1,:) = ones(1,M+1); % top
```

# Neumann Problem

Poisson equation (14.3) is to be solved on the square domain $D = (0, a) \times (0, b)$ subject to Neumann boundary condition

$$\partial_n u = g(x, y), \quad \text{for } (x, y) \in \Gamma$$

where $\partial_n u$ denotes differentiation in the direction of the outward normal to $\Gamma$. The normal is not well defined at corners of the domain and $g(x, y)$ need not be continuous there.

To generate a finite difference approximation of this problem we use the same grid as before and Poisson equation (14.3) is approximated at internal grid points by the five-point stencil.

# Neumann Problem

At vertical boundaries, where $\partial_n = \pm\partial_x$, subtracting the Taylor expansions

$$u(x+h,y) = u + hu_x + \frac{1}{2}h^2 u_{xx} + \frac{1}{6}h^3 u_{xxx} + \mathcal{O}(h^4)$$

$$u(x-h,y) = u - hu_x + \frac{1}{2}h^2 u_{xx} - \frac{1}{6}h^3 u_{xxx} + \mathcal{O}(h^4)$$

gives us

$$u(x+h,y) - u(x-h,y) = 2hu_x + \mathcal{O}(h^2)$$

Rearrange to get

$$u_x(x,y) = \frac{1}{2h}(u(x+h,y) - u(x-h,y)) + \mathcal{O}(h^2) \quad \textbf{(14.9)}$$

# Neumann Problem

Along the top boundary where $\partial_n = -\partial_x$ (note the negative sign appropriate for the *outward* normal on this boundary), when this is applied at the grid point $(0, y_k)$ and we neglect the remainder term, we obtain

$$\frac{1}{2h}(U_{0-1,k} - U_{0+1,k}) = g_{0,k} \quad k = 0, 1, 2, \ldots, N$$

or

$$\frac{1}{2h}(U_{-1,k} - U_{1,k}) = g_{0,k} \qquad \textbf{(14.10)}$$

This approximation involves the value of *U* at the ‘fictitious’ grid point $(-1, k)$ which lies outside the domain *D*.

We write the approximation (14.7) at the boundary points

$$4U_{0,k} - U_{-1,k} - U_{1,k} - U_{0,k+1} - U_{0,k-1} = h^2 f_{0,k} \quad \textbf{(14.11)}$$

# Neumann Problem

We may eliminate $U_{-1,k}$ between (14.10) and (14.11) to obtain a difference formula with a four point stencil by rearranging the formula in (14.10) into

$$U_{-1,k} = U_{1,k} + 2hg_{0,k} \qquad \textbf{(14.12)}$$

and substitute this into (14.11) to get

$$4U_{0,k} - (U_{1,k} + 2hg_{0,k}) - U_{1,k} - U_{0,k+1} - U_{0,k-1} = h^2 f_{0,k}$$

$$4U_{0,k} - 2U_{1,k} - 2hg_{0,k} - U_{0,k+1} - U_{0,k-1} = h^2 f_{0,k}$$

or

$$2U_{0,k} - U_{1,k} - \frac{1}{2}U_{0,k+1} - \frac{1}{2}U_{0,k-1} = \frac{1}{2}h^2 f_{0,k} + hg_{0,k} \qquad \textbf{(14.13)}$$

for $k = 1, 2, \ldots, N-1$

# Neumann Problem

Along the bottom boundary or at $i = M$ where now the *outward* normal is positive or $\partial_n = \partial_x$, we obtain

$$\frac{1}{2h}(U_{M+1,k} - U_{M-1,k}) = g_{M,k} \qquad \textbf{(14.14)}$$

where we have substitution for the fictitious grid point $(M+1, k)$ that lies outside the domain *D*:

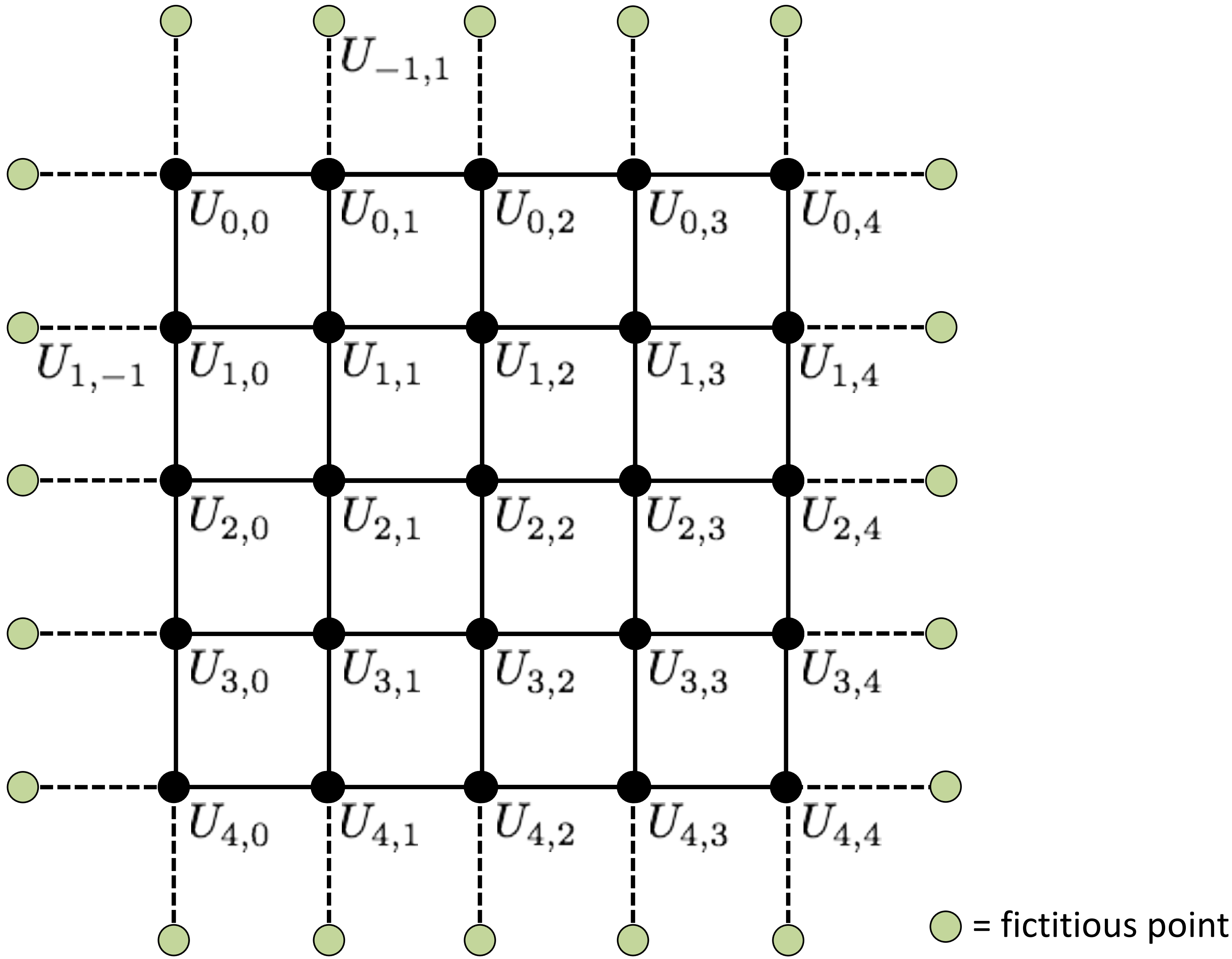
$$U_{M+1,k} = 2hg_{M,k} + U_{M-1,k} \qquad \textbf{(14.15)}$$

Similar to the top boundary, the approximation (14.7) at the bottom boundary becomes

$$2U_{M,k} - U_{M-1,k} - \frac{1}{2}U_{M,k+1} - \frac{1}{2}U_{M,k-1} = \frac{1}{2}h^2 f_{M,k} + hg_{M,k} \qquad \textbf{(14.16)}$$

for $k = 1, 2, \ldots, N-1$

# Neumann Problem



# Neumann Problem

Equation (14.13) also holds with $k = N$ at the corner $(0, b)$ where we also have $\partial_n u = \partial_y u = g$ (note the positive sign) which is approximated by

$$\frac{1}{2h}(U_{0,N+1} - U_{0,N-1}) = g_{0,N} \qquad \textbf{(14.17)}$$

Combining (14.17) with (14.13) at $k = N$ we find

$$2U_{0,N} - U_{1,N} - U_{0,N-1} = \frac{1}{2}h^2 f_{0,N} + 2hg_{0,N}$$

Hence, for the corner point $(0, N)$ on the top right:

$$U_{0,N} - \frac{1}{2}U_{1,N} - \frac{1}{2}U_{0,N-1} = \frac{1}{4}h^2 f_{0,N} + hg_{0,N} \qquad \textbf{(14.18)}$$

# Neumann Problem

For the point along $y = 0$ where the outward normal becomes $\partial_n = -\partial_y$, the approximation is represented by

$$\frac{1}{2h}(U_{i,0-1} - U_{i,0+1}) = g_{i,0}$$

to get the fictitious point

$$U_{i,-1} = 2hg_{i,0} + U_{i,1} \qquad \textbf{(14.19)}$$

Hence at the top left point $(0, 0)$ on the left boundary, substituting (14.12) and (14.19) into (14.7) we get

$$U_{0,0} - \frac{1}{2}U_{0,1} - \frac{1}{2}U_{1,0} = \frac{1}{4}h^2 f_{0,0} + hg_{0,0} \qquad \textbf{(14.20)}$$

# Neumann Problem

In summary, for Neumann BC we need approximation for for the corner points, which, from

$$4U_{i,k} - U_{i+1,k} - U_{i-1,k} - U_{i,k+1} - U_{i,k-1} = f_{i,k}$$

the approximation at point

$(0,0):$ $$4U_{0,0} - U_{1,0} - U_{-1,0} - U_{0,1} - U_{0,-1} = f_{0,0}$$

$(0,N):$

$$4U_{0,N} - U_{1,N} - U_{-1,N} - U_{0,N+1} - U_{0,N-1} = f_{0,N}$$

$(M,0):$

$$4U_{M,0} - U_{M+1,0} - U_{M-1,0} - U_{M,1} - U_{M,-1} = f_{M,0}$$

$(M,N):$

$$4U_{M,N} - U_{M+1,N} - U_{M-1,N} - U_{M,N+1} - U_{M,N-1} = f_{M,N}$$

# Neumann Problem

The points outside the boundary on the bottom

$U_{-1,0}$ and $U_{-1,N}$ are approximated using (14.12).

The points outside the boundary on the top

$U_{M+1,0}$ and $U_{M+1,N}$ are approximated using (14.15).

The points outside the boundary on the left

$U_{0,-1}$ and $U_{M,-1}$ are approximated using (14.19).

The points outside the boundary on the right

$U_{0,N+1}$ and $U_{M,N+1}$ are approximated using (14.17).

# 2D Poisson Equation (Neumann Problem)

As before, the 1st column from the left grid points gives us:

$$i = 0, k = 0 \ : 4U_{0,0} - U_{-1,0} - U_{0,-1} - U_{1,0} - U_{0,1} = h^2 f_{0,0}$$

$$4U_{0,0} - 2U_{1,0} - 2U_{0,1} = h^2 f_{0,0} + 4hg_{0,0}$$

$$i = 1, k = 0 \ : 4U_{1,0} - U_{0,0} - U_{1,-1} - U_{2,0} - U_{1,1} = h^2 f_{1,0}$$

$$4U_{1,0} - U_{0,0} - 2U_{1,1} - U_{2,0} = h^2 f_{1,0} + 2hg_{1,0}$$

$$i = 2, k = 0 \ : 4U_{2,0} - U_{1,0} - 2U_{2,1} - U_{3,0} = h^2 f_{2,0} + 2hg_{2,0}$$

$$i = 3, k = 0 \ : 4U_{3,0} - U_{2,0} - 2U_{3,1} - U_{4,0} = h^2 f_{3,0} + 2hg_{3,0}$$

$$i = 4, k = 0 \ : 4U_{4,0} - 2U_{3,0} - 2U_{4,1} = h^2 f_{4,0} + 4hg_{4,0}$$

# 2D Poisson Equation (Neumann Problem)

Arranging in matrix-vector form yields

$$\begin{bmatrix} 4 & -2 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -2 & 4 \end{bmatrix} \begin{bmatrix} U_{0,0} \\ U_{1,0} \\ U_{2,0} \\ U_{3,0} \\ U_{4,0} \end{bmatrix} + \begin{bmatrix} -2 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} U_{0,1} \\ U_{1,1} \\ U_{2,1} \\ U_{3,1} \\ U_{4,1} \end{bmatrix} =$$

$$h^2 \begin{bmatrix} f_{0,0} \\ f_{1,0} \\ f_{2,0} \\ f_{3,0} \\ f_{4,0} \end{bmatrix} + \begin{bmatrix} 4hg_{0,0} \\ 2hg_{1,0} \\ 2hg_{2,0} \\ 2hg_{3,0} \\ 4hg_{4,0} \end{bmatrix}$$

# 2D Poisson Equation (Neumann Problem)

The 2nd column of the left grid points gives us

$$i = 0, k = 1 \; : 4U_{0,1} - U_{-1,1} - U_{0,0} - U_{1,1} - U_{0,2} = h^2 f_{0,1}$$

$$4U_{0,1} - U_{0,0} - 2U_{1,1} - U_{0,2} = h^2 f_{0,1} + 2hg_{0,1}$$

$$i = 1, k = 1 \; : 4U_{1,1} - U_{0,1} - U_{1,0} - U_{2,1} - U_{1,2} = h^2 f_{1,1}$$

$$i = 2, k = 1 \; : 4U_{2,1} - U_{1,1} - U_{2,0} - U_{3,1} - U_{2,2} = h^2 f_{2,1}$$

$$i = 3, k = 1 \; : 4U_{3,1} - U_{2,1} - U_{3,0} - U_{4,1} - U_{3,2} = h^2 f_{3,1}$$

$$i = 4, k = 1 \; : 4U_{4,1} - U_{3,1} - U_{4,0} - U_{5,1} - U_{4,2} = h^2 f_{4,1}$$

$$4U_{4,1} - 2U_{3,1} - U_{4,0} - U_{4,2} = h^2 f_{4,1} + 2hg_{4,1}$$

# 2D Poisson Equation (Neumann Problem)

Arranging in the matrix-vector form:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} U_{0,0} \\ U_{1,0} \\ U_{2,0} \\ U_{3,0} \\ U_{4,0} \end{bmatrix} + \begin{bmatrix} 4 & -2 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -2 & 4 \end{bmatrix} \begin{bmatrix} U_{0,1} \\ U_{1,1} \\ U_{2,1} \\ U_{3,1} \\ U_{4,1} \end{bmatrix} +$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} U_{0,2} \\ U_{1,2} \\ U_{2,2} \\ U_{3,2} \\ U_{4,2} \end{bmatrix} = h^2 \begin{bmatrix} f_{0,1} \\ f_{1,1} \\ f_{2,1} \\ f_{3,1} \\ f_{4,1} \end{bmatrix} + \begin{bmatrix} 2hg_{0,1} \\ 0 \\ 0 \\ 0 \\ 2hg_{4,1} \end{bmatrix}$$

# Neumann Problem

Proceeding in this manner for each of the boundary segments and each corner, we arrive at $M^2$ linear equations for the values of U in the domain and on boundaries. We then should assemble the equations into a matrix form

$$A\mathbf{U} = h^2\mathbf{F} + h\mathbf{G} \qquad \textbf{(14.16)}$$

where we use the same column-wise ordering as before.

$$A = \begin{bmatrix} B & -2I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -2I & B \end{bmatrix}_{M^2 \times M^2}$$

# Neumann Problem

where

$$B = \begin{bmatrix} 4 & -2 & & & & \\ -1 & 4 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & -1 & 4 & -1 \\ & & & & -2 & 4 \end{bmatrix}_{M \times M}$$

# Example

Solve the BVP defined by:

$$-\nabla^2 u = 20\cos(3\pi x)\sin(2\pi y)$$

on a unit square with zero-flux on all boundaries:

$$u_x(0,y) = 0$$

$$u_x(1,y) = 0$$

$$u_y(x,0) = 0$$

$$u_y(x,1) = 0$$

using second order approximation (central finite difference).

# Example