SF2520 — Applied numerical methods

Lecture 7

Finite differences for elliptic equations
Error analysis
Two-dimensional case

Olof Runborg
Numerical analysis
Department of Mathematics, KTH

2023-09-25

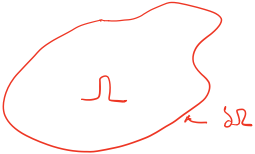
Today's lecture

- Summary of last lecture
- Finite difference methods for elliptic problems
  - Error analysis in 1D
  - Method in two dimensions

Summary of last lecture

Elliptic equations

- Model elliptic equation is the Poisson equation,



$$
\begin{aligned}
-\Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \\
u(\boldsymbol{x}) &= g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial\Omega,
\end{aligned}
$$

  for some given functions f, g and domain $\Omega \subset \mathbb{R}^d$.

- In one dimension, two-point boundary value problem,

$$
-\frac{d}{dx}\left(\kappa(x)\frac{du}{dx}\right) + p(x)\frac{du}{dx} + q(x)u = f(x), \qquad x \in (a,b),
$$

$$
u(a) = \alpha, \qquad u(b) = \beta.
$$

- Other boundary conditions

$$
u_x(a) = \alpha \ \ \text{(Neumann)}, \qquad u_x(a) = \alpha_0 u(a) + \alpha_1 \ \ \text{(Robin)},
$$

$$
u(a) = u(b), \qquad u_x(a) = u_x(b) \ \ \text{(periodic)}.
$$

- Can also have a fully nonlinear two-point boundary value problem

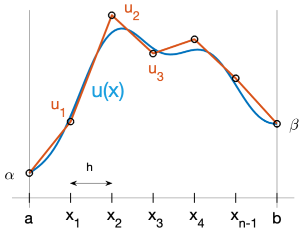
$$
-u_{xx} = F\left(x, u, u_x\right), \qquad u(a) = \alpha, \quad u(b) = \beta.
$$

Numerical methods in 1D – finite difference method

Finite difference method for:

$$-\frac{d}{dx}\left(\kappa(x)\frac{du}{dx}\right) + p(x)\frac{du}{dx} + q(x)u = f(x),$$

when $a < x < b$, and boundary conditions $u(a) = \alpha$ and $u(b) = \beta$.



1. Discretize: $x_j = a + jh$ and $u_j \approx u(x_j)$, etc.
2. Approximate derivatives with (second order) differences

$$-\frac{\kappa_{j+\frac{1}{2}}u(x_{j+1}) - (\kappa_{j+\frac{1}{2}} + \kappa_{j-\frac{1}{2}})u(x_j) + \kappa_{j-\frac{1}{2}}u(x_{j-1})}{h^2} + p_j\frac{u(x_{j+1}) - u(x_{j-1})}{2h} + q_j u(x_j) = f_j + O(h^2).$$

3. Define the approximation

$$\underbrace{\left(-\frac{\kappa_{j-\frac{1}{2}}}{h^2} - \frac{p_j}{2h}\right)}_{a_j} u_{j-1} + \underbrace{\left(\frac{\kappa_{j-\frac{1}{2}} + \kappa_{j+\frac{1}{2}}}{h^2} + q_j\right)}_{b_j} u_j + \underbrace{\left(-\frac{\kappa_{j+\frac{1}{2}}}{h^2} + \frac{p_j}{2h}\right)}_{c_j} u_{j+1} = f_j.$$

4. Apply boundary conditions: $b_1 u_1 + c_1 u_2 = f_1 - a_1\alpha$ for $u(a) = \alpha$, etc.
5. Formulate as matrix equation: $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$

Numerical methods – finite difference method

Finite difference method in 1D leads to linear system of equations $A\boldsymbol{u} = \boldsymbol{f}$:

$$\begin{pmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{n-2} & b_{n-2} & c_{n-2} & \\ & & & a_{n-1} & b_{n-1} & \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 - a_1\alpha \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} - c_{n-1}\beta \end{pmatrix},$$

where

$$a_j = -\frac{\kappa_{j-\frac{1}{2}}}{h^2} - \frac{p_j}{2h}, \qquad b_j = \frac{\kappa_{j-\frac{1}{2}} + \kappa_{j+\frac{1}{2}}}{h^2} + q_j, \qquad c_j = -\frac{\kappa_{j+\frac{1}{2}}}{h^2} + \frac{p_j}{2h}.$$

- Robin conditions at $x = a$

$$\frac{du(a)}{dx} = \alpha_0 u(a) + \alpha_1, \qquad u(b) = \beta.$$

  done by either ghost point method + central approximation of du/dx, or by skewed approximation.
- Only changes first row of matrix and first element of right hand side. (One more unknown is added in ghost point method.)

Error analysis

- Consider the model problem

$$-u_{xx} = f(x), \qquad u(a) = \alpha, \quad u(b) = \beta.$$

- This leads to the discretization

$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = f_j,$$

- Introduce some vector notation,

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{pmatrix} \approx \boldsymbol{u}_{\text{ex}} = \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{n-1}) \end{pmatrix}.$$

- Discretization leads to the matrix form $A\boldsymbol{u} = \boldsymbol{f}$, where

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \quad \boldsymbol{f} = \begin{pmatrix} f_1 + \frac{\alpha}{h^2} \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} + \frac{\beta}{h^2} \end{pmatrix}, \qquad h = \frac{b-a}{n}.$$

Error analysis, local truncation error

- Want to study the global error $e_j = u(x_j) - u_j$ and use "RMS" norm to measure it

$$||\boldsymbol{e}||_{\mathrm{rms}} := \sqrt{\frac{1}{n-1}\sum_{j=1}^{n-1} e_j^2} = \frac{1}{\sqrt{n-1}}||\boldsymbol{e}||_2, \qquad \boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{n-1} \end{pmatrix} = \boldsymbol{u}_{\mathrm{ex}} - \boldsymbol{u}.$$

- Define local truncation error ℓ_j as the mismatch of exact solution entered in scheme,

$$-\frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} = f(x_j) + \ell_j \quad \Rightarrow \quad A\boldsymbol{u}_{\mathrm{ex}} = \boldsymbol{f} + \boldsymbol{\ell}, \qquad \boldsymbol{\ell} = \begin{pmatrix} \ell_1 \\ \ell_2 \\ \vdots \\ \ell_{n-1} \end{pmatrix}.$$

- Then ℓ_j is simply the error in the difference approximation of u_{xx}, since $-u_{xx} = f$. If u is smooth enough, $\exists M$ independent of h such that

$$|\ell_j| \leq Mh^2 \quad \text{and} \quad ||\boldsymbol{\ell}||_{\mathrm{rms}} = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n-1} |\ell_j|^2} \leq M\sqrt{\frac{1}{n-1}\sum_{j=1}^{n-1} h^4} = Mh^2.$$

Error analysis, local to global error

- We can relate the local and global errors,

$$\boldsymbol{A}\boldsymbol{e} = A(\boldsymbol{u}_{\mathrm{ex}} - \boldsymbol{u}) = \boldsymbol{\ell} \quad \Rightarrow \quad \boldsymbol{e} = A^{-1}\boldsymbol{\ell},$$

- Gives us the estimate

$$||\boldsymbol{e}||_{\mathrm{rms}} = ||A^{-1}\boldsymbol{\ell}||_{\mathrm{rms}} = \frac{||A^{-1}\boldsymbol{\ell}||_2}{\sqrt{n-1}} \leq ||A^{-1}||_2||\boldsymbol{\ell}||_{\mathrm{rms}} \leq M||A^{-1}||_2 h^2.$$

- We thus need to determine how $||A^{-1}||_2$ depends on h.

(Note: $A \sim h^{-2}$ and $A \in \mathbb{R}^{(n-1)\times(n-1)}$ which depends on h since $h = (b-a)/n$.)

- A is a symmetrix matrix. Therefore, if λ_k are its eigenvalues,

$$||A||_2 = \max_{1\leq k\leq n-1} |\lambda_k|, \qquad ||A^{-1}||_2 = \frac{1}{\min_{1\leq k\leq n-1} |\lambda_k|}.$$

- Eigenvalues can be computed explicitly in this case. (See Edsberg A.2.)

$$\lambda_k = \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2n}\right).$$

Error analysis, local to global error, cont.

- From derivations (see notes): There is a constant $C = 4/(b-a)^2$ independent of h such that

$$\min_{1\leq k\leq n-1} |\lambda_k| = \frac{4}{h^2}\sin^2\left(\frac{\pi}{2n}\right) \geq C$$

and

$$||A^{-1}||_2 = \frac{1}{\min_{1\leq k\leq n-1} |\lambda_k|} \leq 1/C.$$

so that

$$||\boldsymbol{e}||_{\text{rms}} \leq M||A^{-1}||_2 h^2 \leq \frac{M}{C}h^2 =: \bar{C}h^2,$$

where $\bar{C}$ is independent of h.

Error analysis, remarks

- Second order accuracy.
- Condition number of A important for iterative methods and stability. It is given as

$$\kappa(A) := ||A||_2 \cdot ||A^{-1}||_2 = \frac{\max_{1 \le k \le n-1} |\lambda_k|}{\min_{1 \le k \le n-1} |\lambda_k|} = \frac{\frac{4}{h^2}\sin^2\left(\frac{(n-1)\pi}{2n}\right)}{\frac{4}{h^2}\sin^2\left(\frac{\pi}{2n}\right)} \sim \frac{\frac{4}{h^2}}{C}$$

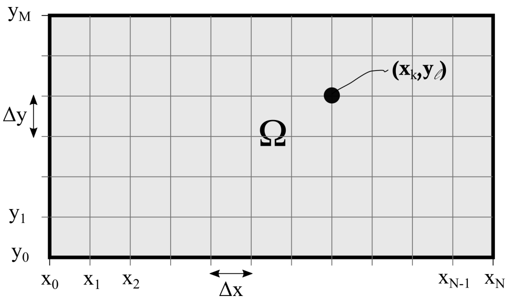
- Hence, $\kappa(A) = O(h^{-2})$. Fine discretizations lead to ill-conditioned systems.
- Derivations above for model problem $-u_{xx} = f$, but result generalizes to most other elliptic problems.
  (See notes on error analysis of FD methods for BVP on homepage.)

Numerical methods in 2D – finite difference method

Finite difference method for the Poisson equation

$$-\Delta u = f, \quad \text{in } \Omega,$$
$$u = g, \quad \text{on } \partial\Omega.$$

when Ω is the rectangle $[0, L_x] \times [0, L_y]$.



Follow the same steps as in 1D.

1. Discretize

   Introduce the Cartesian grid where gridlines $\parallel$ axes,

   $$x_k = k\Delta x, \qquad y_\ell = \ell \Delta y, \qquad \Delta x = \frac{L_x}{N} \qquad \Delta y = \frac{L_y}{M},$$

   and the approximations

   $$u_{k,\ell} \approx u(x_k, y_\ell).$$

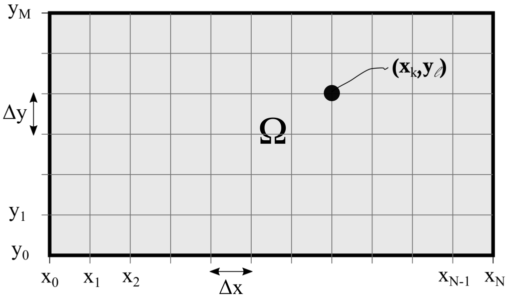
   - We assume that $\Delta x = \Delta y$ henceforth. (Need ML_x/L_y integer.)
   - In total $(N+1)(M+1)$ points/unknowns.

Numerical methods in 2D – finite difference method

Finite difference method for the Poisson equation

$$-\Delta u = f, \quad \text{in } \Omega,$$
$$u = g, \quad \text{on } \partial\Omega.$$

when $\Omega = [0, L_x] \times [0, L_y]$.



2. Approximate $\Delta = \partial_{xx} + \partial_{yy}$ with differences

For every inner point, $1 \le k \le N-1$ and $1 \le \ell \le M-1$,

$$u_{xx}(x_k, y_\ell) = \frac{u(x_k + \Delta x, y_\ell) - 2u(x_k, y_\ell) + u(x_k - \Delta x, y_\ell)}{\Delta x^2} + O(\Delta x^2),$$
$$u_{yy}(x_k, y_\ell) = \frac{u(x_k, y_\ell + \Delta y) - 2u(x_k, y_\ell) + u(x_k, y_\ell - \Delta y)}{\Delta y^2} + O(\Delta y^2).$$

This gives (upon entering it into the equation) and using $\Delta x = \Delta y$,

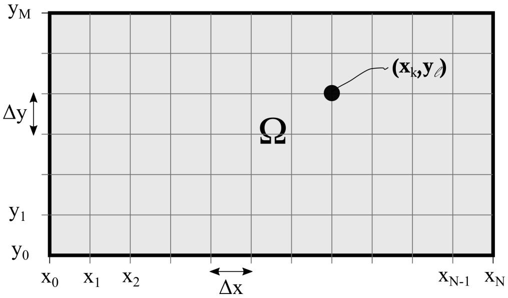
$$-\frac{u(x_{k+1}, y_\ell) + u(x_{k-1}, y_\ell) + u(x_k, y_{\ell+1}) + u(x_k, y_{\ell-1}) - 4u(x_k, y_\ell)}{\Delta x^2} = f(x_k, y_\ell) + O(\Delta x^2).$$

Numerical methods in 2D – finite difference method

Finite difference method for the Poisson equation

$$\begin{aligned} -\Delta u &= f, \quad \text{in } \Omega, \\ u &= g, \quad \text{on } \partial\Omega. \end{aligned}$$

when $\Omega = [0, L_x] \times [0, L_y]$.

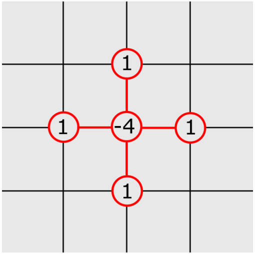


3. Define the approximation

   Neglecting $O(\Delta x^2)$ and replacing $u(x_k, y_\ell)$ by $u_{k,\ell}$,

$$-\frac{u_{k+1,\ell} + u_{k-1,\ell} + u_{k,\ell+1} + u_{k,\ell-1} - 4u_{k,\ell}}{\Delta x^2} = f(x_k, y_\ell),$$

   for inner points, $1 \leq k \leq N-1$ and $1 \leq \ell \leq M-1$.



- This is the "five-point formula" for approximating the Laplace operator Δ.
- Called a *computational stencil*.

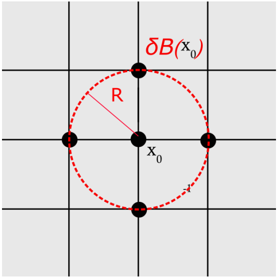
Remark

- The solution u is called harmonic if $f = 0$,

$$-\Delta u = 0, \quad \text{in } \Omega,$$

- For harmonic functions the following mean value theorem holds,

$$\frac{1}{2\pi R}\int_{\partial B(\boldsymbol{x}_0)} u(x)dx = u(\boldsymbol{x}_0).$$



- In the discrete case, the five-point formula gives

$$-\frac{u_{k+1,\ell} + u_{k-1,\ell} + u_{k,\ell+1} + u_{k,\ell-1} - 4u_{k,\ell}}{\Delta x^2} = 0.$$

- This implies that $u_{k,\ell}$ is the mean value of its neighbours,

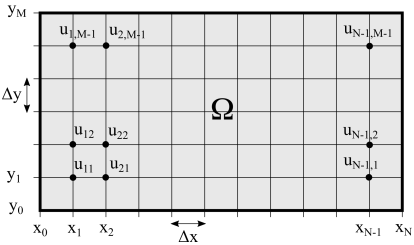
$$u_{k,\ell} = \frac{1}{4}\left(u_{k+1,\ell} + u_{k-1,\ell} + u_{k,\ell+1} + u_{k,\ell-1}\right).$$

Numerical methods in 2D – finite difference method

FD for Poisson equation

$$-\Delta u = f, \quad \text{in } \Omega,$$
$$u = g, \quad \text{on } \partial\Omega.$$

We have $(N+1)(M+1)-4$ unknowns (no corner points!) but only $(N-1)(M-1)$ equations.



4. Apply boundary conditions ($u_{k,\ell} = g(x_k, y_\ell)$ for $(x_k, y_\ell) \in \partial\Omega$)

$$\begin{aligned}
u_{0,\ell} &= g(x_0, y_\ell) = g(0, y_\ell), && (M-1 \text{ eqs.})\\
u_{N,\ell} &= g(x_N, y_\ell) = g(L_x, y_\ell), && (M-1 \text{ eqs.})\\
u_{k,0} &= g(x_k, y_0) = g(x_k, 0), && (N-1 \text{ eqs.})\\
u_{k,M} &= g(x_k, y_M) = g(x_k, L_y), && (N-1 \text{ eqs.})
\end{aligned}$$

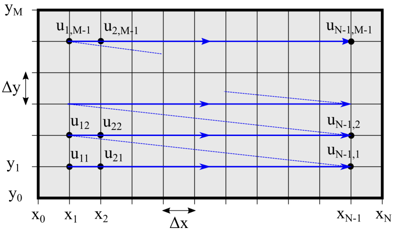
Gives $2(N-1)+2(M-1) = (N+1)(M+1)-4-(N-1)(M-1)$ additional eqs.

- Leads to modified equations for points next to the boundary, e.g. $k=1$,

$$-\frac{u_{2,\ell}+u_{1,\ell+1}+u_{1,\ell-1}-4u_{1,\ell}}{\Delta x^2} = f(x_1, y_\ell) + \frac{g(x_0, y_\ell)}{\Delta x^2}, \qquad 2 \le \ell \le M-2.$$

- Neumann conditions by ghost points or skewed stencils (Edsberg 7.3).

Numerical methods in 2D – finite difference method



5. Formulate as matrix equation

- More tricky than in 1D. Must first select ordering of the unknowns in the vector.
- $\boldsymbol{u}$ contains only inner points.
- Same ordering of $\boldsymbol{f}$ ($f_{k,\ell} = f(x_k, y_\ell)$, no BC yet)

$$\boldsymbol{u} = \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{N-1,1} \\ \hline u_{12} \\ u_{22} \\ \vdots \\ u_{N-1,2} \\ \hline \\ \vdots \\ \\ \hline u_{1,M-1} \\ u_{2,M-1} \\ \vdots \\ u_{N-1,M-1} \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} f_{11} \\ f_{21} \\ \vdots \\ f_{N-1,1} \\ \hline f_{12} \\ f_{22} \\ \vdots \\ f_{N-1,2} \\ \hline \\ \vdots \\ \\ \hline f_{1,M-1} \\ f_{2,M-1} \\ \vdots \\ f_{N-1,M-1} \end{pmatrix} \in \mathbb{R}^{(N-1)(M-1)}$$

Numerical methods in 2D – finite difference method

Matrix form $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$ when $g \equiv 0$ and $M = N = 5$,

$$
\frac{1}{\Delta x^2}
\left(\begin{array}{cccc|cccc|cccc|cccc}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
& -1 & 4 & -1 & & & -1 & & & & & & & & & \\
& & -1 & 4 & & & & -1 & & & & & & & & \\
\hline
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
& -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
& & -1 & & & -1 & 4 & -1 & & & -1 & & & & & \\
& & & -1 & & & -1 & 4 & & & & -1 & & & & \\
\hline
& & & & -1 & & & & 4 & -1 & & & -1 & & & \\
& & & & & -1 & & & -1 & 4 & -1 & & & -1 & & \\
& & & & & & -1 & & & -1 & 4 & -1 & & & -1 & \\
& & & & & & & -1 & & & -1 & 4 & & & & -1 \\
\hline
& & & & & & & & -1 & & & & 4 & -1 & & \\
& & & & & & & & & -1 & & & -1 & 4 & -1 & \\
& & & & & & & & & & -1 & & & -1 & 4 & -1 \\
& & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right)
\left(\begin{array}{c}
u_{11} \\ u_{21} \\ u_{31} \\ u_{41} \\ \hline u_{12} \\ u_{22} \\ u_{32} \\ u_{42} \\ \hline u_{13} \\ u_{23} \\ u_{33} \\ u_{43} \\ \hline u_{14} \\ u_{24} \\ u_{34} \\ u_{44}
\end{array}\right)
=
\left(\begin{array}{c}
f_{11} \\ f_{21} \\ f_{31} \\ f_{41} \\ \hline f_{12} \\ f_{22} \\ f_{32} \\ f_{42} \\ \hline f_{13} \\ f_{23} \\ f_{33} \\ f_{43} \\ \hline f_{14} \\ f_{24} \\ f_{34} \\ f_{44}
\end{array}\right)
$$

$$
\frac{-u_{k+1,\ell} - u_{k-1,\ell} - u_{k,\ell+1} - u_{k,\ell-1} + 4u_{k,\ell}}{\Delta x^2} = f_{k,\ell}.
$$

Matrix form $A\boldsymbol{u} = \boldsymbol{f}$ when $g \equiv 0$ and general M, N.

$$
\frac{1}{\Delta x^2}
\left(\begin{array}{cccc|cccc|cccc|cccc}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
 & \ddots & \ddots & \ddots & & & \ddots & & & & & & & & & \\
 & & -1 & 4 & & & & -1 & & & & & & & & \\
\hline
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
 & -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
 & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & & & & & \\
 & & & -1 & & & -1 & 4 & & & & -1 & & & & \\
\hline
 & & & & & & & & & & & & & & & \\
 & & & & & \ddots & & & & \ddots & & & & \ddots & & \\
 & & & & & & & & & & & & & & & \\
\hline
 & & & & & & & & -1 & & & & 4 & -1 & & \\
 & & & & & & & & & -1 & & & -1 & 4 & -1 & \\
 & & & & & & & & & & \ddots & & & \ddots & \ddots & \ddots \\
 & & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right)
\left(\begin{array}{c}
u_{11} \\ u_{21} \\ \vdots \\ u_{N-1,1} \\
\hline
u_{12} \\ u_{22} \\ \vdots \\ u_{N-1,2} \\
\hline
\vdots \\ \vdots \\
\hline
u_{1,M-1} \\ u_{1,M-1} \\ \vdots \\ u_{N-1,M-1}
\end{array}\right)
=
\left(\begin{array}{c}
f_{11} \\ f_{21} \\ \vdots \\ f_{N-1,1} \\
\hline
f_{12} \\ f_{22} \\ \vdots \\ f_{N-1,2} \\
\hline
\vdots \\ \vdots \\
\hline
f_{1,M-1} \\ f_{2,M-1} \\ \vdots \\ f_{N-1,M-1}
\end{array}\right).
$$

A is block tridiagonal with $(M-1) \times (M-1)$ blocks of size $(N-1) \times (N-1)$.

Numerical methods in 2D – finite difference method

For general Dirichlet boundary conditions where $g \neq 0$, the right hand side would be modified at the points next to the boundary as,

$$
\boldsymbol{f} = \begin{pmatrix} f_{11} \\ f_{21} \\ \vdots \\ f_{N-1,1} \\ \hline f_{12} \\ f_{22} \\ \vdots \\ f_{N-1,2} \\ \hline \vdots \\ \vdots \\ \hline f_{1,M-1} \\ f_{2,M-1} \\ \vdots \\ f_{N-1,M-1} \end{pmatrix} + \frac{1}{\Delta x^2} \underbrace{\begin{pmatrix} g(x_1,0) \\ g(x_2,0) \\ \vdots \\ g(x_{N-1},0) \\ \hline 0 \\ 0 \\ \vdots \\ 0 \\ \hline \vdots \\ \vdots \\ \hline g(x_1,L_y) \\ g(x_2,L_y) \\ \vdots \\ g(x_{N-1},L_y) \end{pmatrix}}_{\text{BC at } y=0 \text{ and } y=L_y} + \frac{1}{\Delta x^2} \underbrace{\begin{pmatrix} g(0,y_1) \\ 0 \\ \vdots \\ 0 \\ g(L_x,y_1) \\ \hline g(0,y_2) \\ 0 \\ \vdots \\ 0 \\ g(L_x,y_2) \\ \hline \vdots \\ \vdots \\ \hline g(0,y_{M-1}) \\ 0 \\ \vdots \\ 0 \\ g(L_x,y_{M-1}) \end{pmatrix}}_{\text{BC at } x=0 \text{ and } x=L_x} .
$$

Numerical methods in 2D – finite difference method

- Matrix form is $A\boldsymbol{u} = \boldsymbol{f}$ with

$$A = \frac{1}{\Delta x^2}\begin{pmatrix} T & -I & & \\ -I & T & -I & \\ & \ddots & \ddots & \ddots \\ & & -I & T \end{pmatrix}, \qquad T = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)},$$

  and I is the $(M-1)\times(M-1)$ identity matrix.
- A can alternatively be written

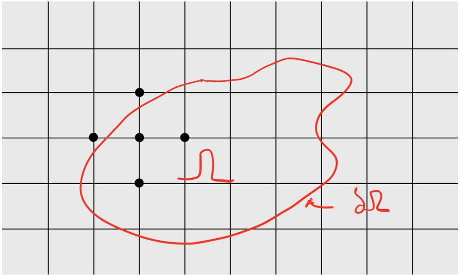
$$A = \frac{1}{\Delta x^2}\mathrm{tridiag}(-I, T, -I), \qquad T = \mathrm{tridiag}(-1, 4, 1),$$

  i.e. block tridiagonal with $(M-1)\times(M-1)$ tridiagonal blocks of size $(N-1)\times(N-1)$.
- A is sparse with bandwidth $N-1$.
- Direct solvers for an $n\times n$ matrix with bandwidth p cost $O(np^2)$.
- Here, if $N = M$, we have $n \sim N^2$ and $p \sim N$, giving cost $O(N^4)$.
  (More about this later in the numerical linear algebra part of course.)

Numerical methods in 2D – finite difference method

Remarks:

- Method is second order accurate.
  (Proved in the same way as in 1D.)
- In 3D a 7-point stencil is used. A is again block tridiagonal, but with bandwidth $\sim N^2$ (instead of $\sim N$). Computational cost to solve $A\boldsymbol{u} = \boldsymbol{f}$ is $O(N^7)$.
- When domain is not a rectangle, finite differences are more difficult to use since BC are hard to impose. Adaptivity even harder.



Numerical methods in 2D – finite difference method

Matlab

- `reshape(u,M,N)` — converts a vector/matrix to an $M \times N$ matrix.
- `kron(C,D)` — returns the Kronecker product of C and D

$$C \otimes D =: \begin{pmatrix} c_{11}D & c_{12}D & \cdots & c_{1,n}D \\ c_{21}D & c_{22}D & \cdots & c_{2,n}D \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1}D & c_{n,2}D & \cdots & c_{n,n}D \end{pmatrix}, \qquad C = \{c_{k,\ell}\} \in \mathbb{R}^{n\times n}.$$

- Note: A can be written in concise form using the Kronecker product (see notes):

$$A = I_{M-1} \otimes S_N + S_M \otimes I_{N-1},$$

where I_n is the $n \times n$ identity matrix and S_n is the 1D discretization,

$$S_n = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}, \qquad \Delta x = \frac{L_x}{n}.$$