SF2520 — Applied numerical methods

Lecture 8

Elliptic equations

Collocation
Finite elements
COMSOL Multiphysics

Olof Runborg
Numerical analysis
Department of Mathematics, KTH

2023-09-28

Today's lecture

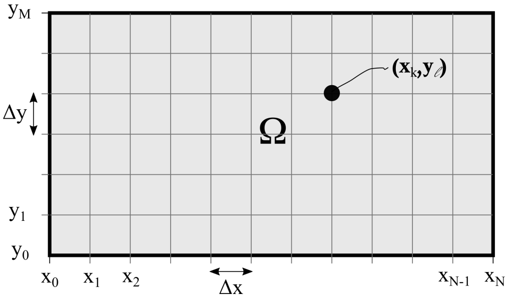
- Recap of previous lecture
- Ansatz methods for elliptic equations
    - Collocation method
    - Finite element method, 1D
    - Finite element method, 2D
- Introduction to COMSOL Multiphysics

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 $[0, L_x] \times [0, L_y]$.



1. Discretize (define M, N, Δx, x_k, y_ℓ and $u_{k,\ell} \approx u(x_k, y_\ell)$)
2. Approximate $\Delta = \partial_{xx} + \partial_{yy}$ with differences

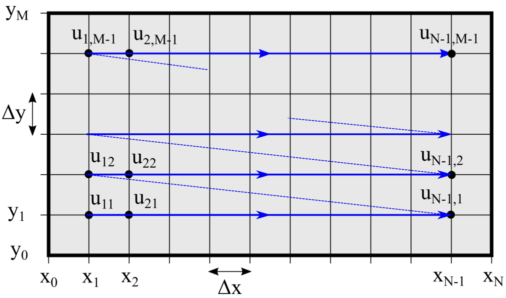
$$-\frac{u(x_{k+1}, y_\ell) + u(x_{k-1}, y_\ell) + u(x_k, y_{\ell+1}) + u(x_k, y_{\ell-1}) - 4u(x_k, y_\ell)}{\Delta x^2} = f(x_k, y_\ell) + O(\Delta x^2).$$

3. Define the approximation ("five-point formula")

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

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

Numerical methods in 2D – finite difference method



5. Formulate as matrix equation
   - Want form: $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$
   - Select ordering of the unknowns in the vector.
   - $\boldsymbol{u}$ contains only inner points.
   - Same ordering of $\boldsymbol{f}$.
   - Both vectors in $\mathbb{R}^{(N-1)(M-1)}$.

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

Numerical methods in 2D – finite difference method

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

- A is block tridiagonal with $(M-1)\times(M-1)$ blocks of size $(N-1)\times(N-1)$.
- Second order accuracy (both in 1D and 2D).

Ansatz methods for elliptic equations

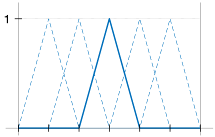
Consider 1D and the two-point boundary value problem

$$-\frac{d^2u}{dx^2} = f(x), \qquad u(a) = u(b) = 0.$$

- Introduce a set of basis functions $\{\phi_k\}$ with $k = 1, \dots, n$, and approximate solution as linear combination of those

$$u(x) \approx \sum_{k=1}^{n} u_k \phi_k(x) =: u_h(x),$$

  where $\{u_k\}$ are to be determined.
- This is called an ansatz of the solution. In ansatz methods one looks for $\{u_k\}$ so that $u - u_h \to 0$ as $n \to \infty$.
- Different types of basis functions can be used:
  - polynomials
  - $\sin(kx)$, $\cos(kx)$ (used in "spectral methods")
  - "hat functions"
  - etc.



- We require that $\phi_k(a) = \phi_k(b) = 0$ for all k.

Ansatz methods for elliptic equations

Collocation methods

$$-u''(x) = f(x), \qquad u(a) = u(b) = 0.$$

- Require PDE be satisfied exactly in n points $\{x_\ell\}_{\ell=1}^n$

$$-u_h''(x_\ell) = -\sum_{k=1}^{n} u_k \phi_k''(x_\ell) = f(x_\ell), \qquad \ell = 1, \ldots, n.$$

- Leads to a linear system of equations $A\boldsymbol{u} = \boldsymbol{f}$, where

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}, \quad A = -\begin{pmatrix} \phi_1''(x_1) & \phi_2''(x_1) & \cdots & \phi_n''(x_1) \\ \phi_1''(x_2) & \phi_2''(x_2) & \cdots & \phi_n''(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1''(x_n) & \phi_2''(x_n) & \cdots & \phi_n''(x_n) \end{pmatrix} \quad \boldsymbol{f} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) \end{pmatrix}.$$

- If ϕ_k are compactly supported (local) then A is sparse but in general not symmetric.
- Need smooth basis functions. Accuracy depends on choice of points and ϕ_j.

Ansatz methods for elliptic equations

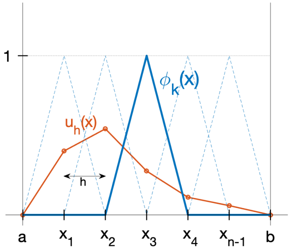
Galerkin methods

$$-u''(x) = f(x), \qquad u(a) = u(b) = 0.$$

- Require that the residual
$$r_h(x) := -u_h''(x) - f(x),$$
satisfies
$$\int_a^b r_h(x)\phi_k(x)dx = 0, \qquad k = 1, \ldots, n.$$
- Can be expressed as: *"residual is orthogonal (in L^2) to all basis functions ϕ_k,"* symbolically
$$r_h \perp \text{span}\{\phi_k\}.$$
- Compare the finite-dimensional case, if $A \in \mathbb{R}^{n\times n}$,
$$\boldsymbol{Ax} = \boldsymbol{b} \quad \Leftrightarrow \quad \boldsymbol{e}_j^T(A\boldsymbol{x} - \boldsymbol{b}) = 0, \qquad j = 1, \ldots, n.$$

Finite element method (FEM)

- = Galerkin method with piecewise polynomial basis functions (e.g. hat functions).
- Discretize the interval as for finite differences, $x_k = a + kh$, $h = (b - a)/n$.
- Basis functions ϕ_k = piecewise linear hat functions, $k = 1, \ldots, n - 1$,



$$\phi_k(x_j) = \begin{cases} 0, & j \neq k, \\ 1, & j = k, \end{cases}$$

$$u_h(x) = \sum_{k=1}^{n-1} u_k \phi_k(x).$$

- Approximation u_h is also a piecewise linear function, since it is a linear combination of ϕ_k.
- $u_h(a) = u_h(b) = 0$ so BC automatically satisified.

Finite element method 1D

FEM approximation

$$u_h(x) := \sum_{k=1}^{n-1} u_k \phi_k(x), \qquad r_h(x) = -u_h''(x) - f(x), \qquad \int_a^b r_h(x)\phi_\ell(x)dx = 0, \ \ \forall \ell.$$

We have for $\ell = 1, \ldots, n-1$,

$$\begin{aligned}
0 &= \int_a^b r_h(x)\phi_\ell(x)dx = -\int_a^b u_h''(x)\phi_\ell(x)dx - \int_a^b f(x)\phi_\ell(x)dx \\
&= \{\text{partial integration}\} = \int_a^b u_h'(x)\phi_\ell'(x)dx - \int_a^b f(x)\phi_\ell(x)dx \\
&= \sum_{k=1}^{n-1} u_k \underbrace{\int_a^b \phi_k'(x)\phi_\ell'(x)dx}_{=a_{\ell,k}} - \underbrace{\int_a^b f(x)\phi_\ell(x)dx}_{=f_\ell} = \sum_{k=1}^{n-1} a_{\ell,k} u_k - f_\ell.
\end{aligned}$$

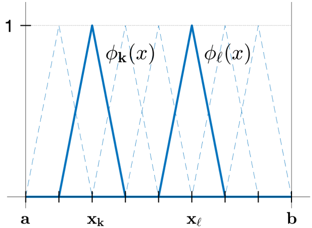
This is a linear system of equations $A\boldsymbol{u} = \boldsymbol{f}$

(Where $A = \{a_{\ell,k}\}$, $\boldsymbol{u} = \{u_k\}$ and $\boldsymbol{f} = \{f_\ell\}$.)

Note: $u_h''(x)$ is not a well-defined function when ϕ_k are piecewise linear. (It is just a distribution.) A more rigorous derivation of FEM starts from the the weak form of the Poisson equation, where only $u_h'(x)$ needs to be well-defined. More on this later!

Finite element method 1D

A closer look at "stiffness matrix" A



Recall

$$a_{\ell,k} = \int_a^b \phi_k'(x)\phi_\ell'(x)dx,$$

$$\phi_k'(x) = \begin{cases} \frac{1}{h}, & x_{k-1} < x < x_k, \\ -\frac{1}{h}, & x_k < x < x_{k+1}, \\ 0, & \text{otherwise.} \end{cases}$$

- $a_{\ell,k} = a_{k,\ell}$ so A is symmetric.
- $a_{\ell,k} = 0$ if $|k-\ell| \geq 2$ so A is tridiagonal (sparse).
- For $\ell = k$,

$$a_{\ell,\ell} = \int_{x_k-h}^{x_\ell+h} \phi_\ell'(x)^2 dx = \int_{x_\ell-h}^{x_\ell+h} \frac{1}{h^2} dx = \frac{2}{h}.$$

- For $k = \ell + 1$,

$$a_{\ell,\ell+1} = \int_{x_\ell}^{x_\ell+h} \phi_\ell'(x)\phi_{\ell+1}'(x)dx = \int_{x_\ell}^{x_\ell+h} \frac{-1}{h} \cdot \frac{1}{h} dx = \frac{-1}{h}.$$

- For $k = \ell - 1$ we get the same by symmetry: $a_{\ell,\ell-1} = -1/h$.

Finite element method 1D

A closer look at A and $\boldsymbol{f}$

We get

$$A = \frac{1}{h}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

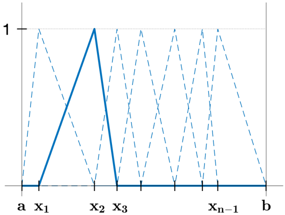
- Same as for finite differences! (Except multiplied by h.)
- Moreover,

$$f_\ell = \int_a^b f(x)\phi_\ell(x)dx = \int_{x_\ell-h}^{x_\ell+h} f(x)\phi_\ell(x)dx = hf(x_\ell) + O(h^3).$$

  Also (almost) same as for finite differences! (Except multiplied by h.)
- Matrix form $A\boldsymbol{u} = \boldsymbol{f}$ almost same as finite differences in 1D.

Remarks

- Second order approximation
$$|u - u_h| = O(h^2),$$
where h = max distance between points.
- Possible to use other $\{\phi_k\}$ to get higher order method. E.g. if ϕ_k = piecewise polynomial of degree p we get order $p + 1$.
- Easy to do adaptivity. Points do not need to be uniformly distributed. This just changes $\{\phi_k\}$.



- Stability and error estimates can be derived for very general cases. FEM solution is "optimal": best piecewise linear approximation measured in energy norm. Good theoretical foundations.
- Works for many, many other PDEs and boundary conditions.

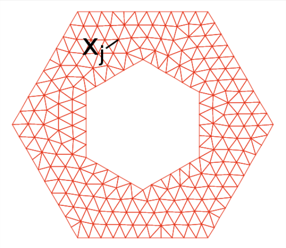
Finite element method 2D

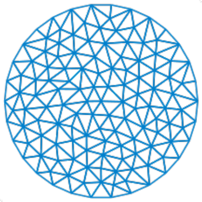
Consider the Poisson equation in 2D

$$
\begin{aligned}
-\Delta u(x) &= f(x), \qquad x \in \Omega, \\
u(x) &= 0, \qquad x \in \partial\Omega.
\end{aligned}
$$

where $\Omega \subset \mathbb{R}^2$ is a bounded open set.

- Discretize Ω. Introduce a *triangulation* with the nodes $x_j \in \mathbb{R}^2$,





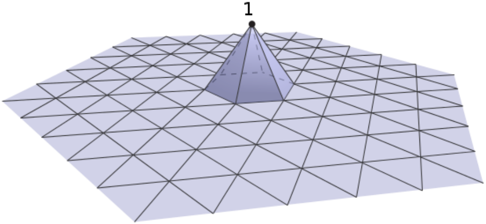
= "computational grid", "mesh"

Finite element method 2D

- Introduce basis functions $\{\phi_k\}_{k=1}^n$ and make the ansatz

$$u(x) \approx u_h(x) := \sum_{k=1}^{n} u_k \phi_k(x),$$

where u_k are coefficients to be determined.
- Choose in particular $\phi_k(x) =$ "pyramid" functions



$$\phi_k(x) = \begin{cases} 1, & x = x_k, \\ 0, & x = x_j, \\ & j \neq k \\ \text{linear}, & \text{otherwise}. \end{cases}$$

- Use ordering such that $x_1, \ldots, x_n$ are the inner nodes. (Many possibilities.)

Finite element method 2D

$$u(x) \approx u_h(x) := \sum_{k=1}^{n} u_k \phi_k(x)$$

- Since $\phi_k(x) =$ "pyramid" functions $\Rightarrow$
  - $u_h(x)$ is piecewise linear
  - Boundary condition $u_h = 0$ is automatically satisfied if Ω is a polygon.
- Let $r_h(x)$ denote the residual

$$r_h(x) = -\Delta u_h(x) - f(x).$$

Find u_k such that $r_h(x)$ *orthogonal* to all basis functions $\phi_\ell(x)$ in the L_2-innerproduct,

$$\int_\Omega r_h(x)\phi_\ell(x)dx = 0, \qquad \ell = 1, \ldots, n.$$

= "Galerkin-approximation".

Note: $\Delta u_h(x)$ is not a well-defined function when ϕ_j are piecewise linear. (It is just a distribution.) A more rigorous derivation of FEM starts from the the weak form of the Poisson equation, where only ∇u_h needs to be well-defined. More on this later!

Finite element method 2D

We get for $\ell = 1, \ldots, n$,

$$
\begin{aligned}
0 &= \int_\Omega (-\Delta u_h(x) - f(x))\phi_\ell(x)dx \\
&= \{\text{partial integration}\} = \int_\Omega \nabla u_h(x) \cdot \nabla \phi_\ell(x)dx - \int_\Omega f(x)\phi_\ell(x)dx \\
&= \sum_{k=1}^{n} u_k \underbrace{\int_\Omega \nabla \phi_k(x) \cdot \nabla \phi_\ell(x)dx}_{=a_{\ell,k}} - \underbrace{\int_\Omega f(x)\phi_\ell(x)dx}_{=f_\ell}.
\end{aligned}
$$

This is a linear system of equations

$$
\sum_{k=1}^{n} a_{\ell,k} u_k = f_\ell, \qquad \ell = 1, \ldots, n, \qquad A\boldsymbol{u} = \mathbf{f}.
$$

where

$$
A = \{a_{\ell,k}\}, \quad \boldsymbol{u} = \{u_k\}, \quad \mathbf{f} = \{f_\ell\}.
$$

Remarks

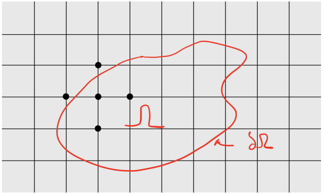
- Second order accuracy when ϕ_k pyramid functions. Error $\sim h^2$ where $h =$ size of largest triangle.
- If ϕ_k = piecewise quadratic, cubic, etc. functions, higher order accuracy is obtained.
- System matrix A is sparse and symmetric as before since

$$a_{\ell,k} = \int_\Omega \nabla\phi_k(x) \cdot \nabla\phi_\ell(x)dx,$$

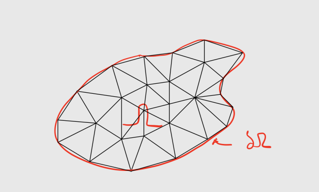
  is zero if x_k not a neighbor of x_ℓ. May be banded if a good ordering of the nodes is used.

Remarks (cont.)

- Triangulations can easier handle complicated geometries and adaptivity, compared to Cartesian grids used in finite difference methods. More triangles can be added locally where the solutions changes rapidly.



vs.



- Building A can be done in a fairly systematic way by considering one triangle at a time and integrating the three pairs $\nabla\phi_k \cdot \nabla\phi_\ell$ which are non-zero there ("assembly" process, see Edsberg 7.4.3). Easier than the finite difference case.
- Good stability properties and theoretical foundation. Error estimates for very general situations available.

COMSOL Multiphysics

COMSOL uses the finite element method to solve PDEs.

Getting started:

1. Install on your own computer or use in computer labs at KTH. OBS! Need mail exchange with IT department at KTH to get license. (During business hours!). Don't do this at the last minute!
2. In the lab rooms you can e.g. start COMSOL (version 6.0) from the terminal:

   ```
   prompt> comsol
   ```
3. Go through tutorial! (See homepage.)

COMSOL Multiphysics

Example: Want to solve e.g. the 2D PDE

$$\begin{aligned} -\Delta u(x,y) &= 10e^{10y}, \qquad (x,y) \in \Omega, \\ u(x,y) &= \sin(10x), \qquad (x,y) \in \partial\Omega. \end{aligned}$$

where Ω is an ellipse.

Basic steps:

1. Choose PDE type and dimension in ModelWizard. You can use the "Classical PDE" interface or the general form

$$eu_{tt} + du_t - \nabla \cdot (c\nabla u + \alpha u - \gamma) + au + \beta \cdot \nabla u = f,$$

   where you specify the coefficients $d, c, f, \ldots$.
2. Build the geometry.
3. Specify PDE-coefficients.
4. Specify boundary conditions.
5. Genererate a computational mesh (a triangulation).
6. Solve the problem.
7. Plot the result (post-processing).