Scientific Computing I

Module 8: Discretisation of PDEs

Michael Bader

Lehrstuhl Informatik V

Winter 2008/2009

Part I: Finite Differences

Part II: Finite Element Methods

The Model Problem

- 2D Poisson Equation on unit square:

$$\frac{\partial^2}{\partial x^2}u(x,y) + \frac{\partial^2}{\partial y^2}u(x,y) = f(x,y) \qquad \text{in } \Omega = (0,1)^2$$

- Dirichlet boundary conditions:

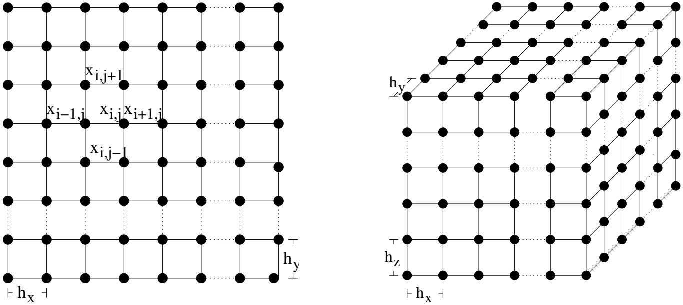
$$u(x,y) = g(x,y) \qquad \text{on } \partial\Omega$$

Part I

Finite Differences

Grid Generation

- generate a grid on the given domain



- Compute values of unknown function u at each grid point:

$$u_{ij} \approx u(x_{ij}) \qquad u_{ijk} \approx u(x_{ijk})$$

Finite Difference Discretisation

- Replace derivatives (at each grid point) by difference quotients:

$$\frac{\partial^2 u}{\partial x^2}(x_{i,j}) \approx \frac{u(x_{i+1,j}) - 2u(x_{i,j}) + u(x_{i-1,j})}{h_x^2}$$

$$\frac{\partial^2 u}{\partial y^2}(x_{i,j}) \approx \frac{u(x_{i,j+1}) - 2u(x_{i,j}) + u(x_{i,j-1})}{h_y^2}$$

- leads to linear system of equations ($h := h_x = h_y$):

$$\begin{aligned} \frac{1}{h^2}\big(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} & \\ + u_{i,j-1} + u_{i-1,j}\big) &= f(x_{i,j}) \quad x_{i,j} \in (0,1)^2 \\ u(x_{i,j}) &= g(x_{i,j}) \quad x_{i,j} \in \partial\Omega \end{aligned}$$

System of Linear Equations

- objective: write linear system in matrix-vector-form:

$$A_h x_h = f_h$$

- x_h contains the unknowns u_{ij}
  $\Rightarrow$ requires *sequentialisation* of the unknowns
- for example, with simple row-wise numbering of the grid points:
  $x_h = (u_{1,1}, \ldots, u_{1,n}, u_{2,1}, \ldots, u_{2,n}, \ldots, u_{n,1}, \ldots, u_{n,n})$

System of Linear Equations (2)

- A_h is a sparse matrix (five-diagonal)
- A_h is a block-tridiagonal matrix:

$$A_h = \frac{1}{h^2} \begin{pmatrix} B_h & I & & \\ I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & I & B_h \end{pmatrix}$$

- $B_h = \text{tridiag}(1, -4, 1)$
- I the identity matrix
- boundary values to right-hand side

Stencil Notation

- illustrate matrix structure as a *discretisation stencil*
- represents one line of the matrix
- matrix elements placed according to their corresponding geometrical position
- stencils for the Poisson equation (h^2 factors ignored):

$$1D: \quad \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \qquad 2D: \quad \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}$$

Accuracy

- for 5-point Poisson stencil;
  order of accuracy: $\|u_h - u\| = \mathcal{O}(h^2) = \mathcal{O}(N^{-2})$
- *curse of dimension:*
  for that, we need $\mathcal{O}(N^d)$ points

Possibilities of an improvement:

- use higher-order discretisation
  - via higher order terms of Taylor series
  - leads to larger stencils
    (involving more neighbouring grid points)
- use locally refined (*adaptive*) grids

Part II

Finite Element Methods

Finite Elements – Main Ingredients

1. compute a *function* as numerical solution; search in a function space W_h:

$$u_h = \sum_j u_j \varphi_j(x), \qquad \mathrm{span}\{\varphi_1, \ldots, \varphi_J\} = W_h$$

2. solve *weak form* of PDE to reduce regularity properties

$$u'' = f \quad \longrightarrow \quad \int v'u' \,\mathrm{d}x = \int vf \,\mathrm{d}x$$

3. choose basis functions with *local support*, e.g.:

$$\varphi_j(x_i) = \delta_{ij}$$

Choose Test and Ansatz Space

- search for solution functions u_h of the form

$$u_h = \sum_j u_j \varphi_j(x)$$

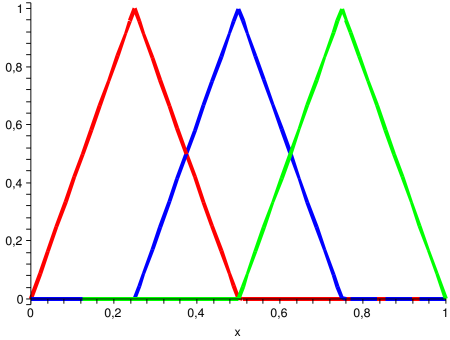
- the basis (ansatz) functions $\varphi_j(x)$ build a vector space (or function space) W_h

$$\text{span}\{\varphi_1, \ldots, \varphi_J\} = W_h$$

- the “best” solution u_h in this function space is wanted

Example: Nodal Basis

$$\varphi_i(x) := \begin{cases} \frac{1}{h}(x - x_{i-1}) & x_{i-1} < x < x_i \\ \frac{1}{h}(x_{i+1} - x) & x_i < x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$



Weak Forms and Weak Solutions

- consider a PDE $Lu = f$ (e.g. $Lu = \Delta u$)
- transformation to the *weak form*:

$$\langle Lu, v \rangle = \int vLu \, \mathrm{dx} = \int vf \, \mathrm{dx} = \langle f, v \rangle \quad \forall v \in V$$

  V a certain class of functions
- "real solution" u also solves the weak form
- $\langle Lu, v \rangle$ a *bilinear form*; often written as:

$$a(u, v) = \langle f, v \rangle \quad \forall v \in V$$

Weak Form of the Poisson Equation

- Poisson equation with Dirichlet conditions:

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

- weak form:

$$-\int_\Omega v \Delta u \, \mathrm{d}\Omega = \int_\Omega v f \, \mathrm{d}\Omega \quad \forall v$$

- apply Green's formula:

$$-\int_\Omega v \Delta u \, \mathrm{d}\Omega = \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega - \int_{\partial\Omega} v \cdot \nabla u \, ds$$

- choose functions v such that $v = 0$ on $\partial\Omega$:

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega = \int_\Omega v f \, \mathrm{d}\Omega \quad \forall v$$

Weak Form of the Poisson Equation (2)

- Poisson equation with Dirichlet conditions:

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

- transformed into weak form:

$$\int_\Omega \nabla v \cdot \nabla u \,\mathrm{d}\Omega = \int_\Omega v f \,\mathrm{d}\Omega \quad \forall v$$

- weaker requirements for a solution u:
  twice differentiabale $\rightarrow$ *first derivative integrable*
- remember use of nodal basis: availability of first vs. second derivative!

Choose Test and Ansatz Space

- search for solutions u_h in a function space W_h:

$$u_h = \sum_j u_j \varphi_j(x)$$

where $\text{span}\{\varphi_j\} = W_h$ ("ansatz space")

- insert into weak solution

$$\int vL\left(\sum_j u_j \varphi_j(x)\right) \mathrm{dx} = \int vf \,\mathrm{dx} \quad \forall v \in V$$

Choose Test and Ansatz Space (2)

- choose a basis $\{\psi_i\}$ of the *test* space V
- then: if all basis functions ψ_i satisfy

$$\int \psi_i L\left(\sum_j u_j \varphi_j(x)\right) \mathrm{dx} = \int \psi_i f \, \mathrm{dx} \quad \forall \psi_i$$

  then all $v \in V$ satisfy the equation
- leads to system of equations for unknowns u_j (one equation per test basis function ψ_i)
- V is often chosen to be identical to W_h (Ritz-Galerkin method)

Discretisation – Finite Elements

- L linear $\Rightarrow$ system of linear equations

$$\sum_j \Big(\underbrace{\int \psi_i L \varphi_j(x)\,\mathrm{d}x}_{=:A_{ij}} \Big) u_j = \int \psi_i f \,\mathrm{d}x \quad \forall \psi_i$$

- aim: make matrix A *sparse* $\rightarrow$ most $A_{ij} = 0$
- approach: local basis functions on a discretisation grid
- ψ_j, φ_j zero everywhere except in grid cells adjacent to grid point x_j
- $A_{ij} = 0$, if ψ_i and φ_j don't overlap

Example Problem: Poisson 1D

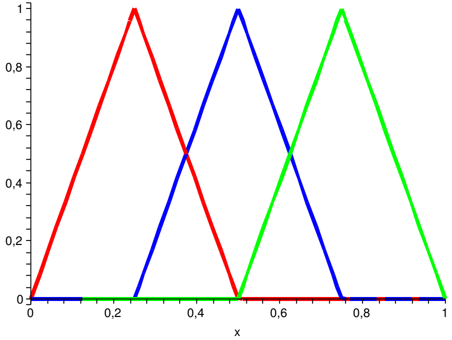
- in 1D: $u''(x) = f(x)$ on $\Omega = (0,1)$,
  hom. Dirichlet boundary cond.: $u(0) = u(1) = 0$
- weak form:

$$\int_0^1 v'(x) \cdot u'(x)\,\mathrm{dx} = \int_0^1 v(x) f(x)\,\mathrm{dx} \quad \forall v$$

- compuational grid:
  $x_i = ih$, (for $i = 1, \ldots, n-1$); mesh size $h = 1/n$
- $V = W$: piecewise linear functions
  (on intervals $[x_i, x_{i+1}]$)

Nodal Basis

$$\varphi_i(x) := \begin{cases} \frac{1}{h}(x - x_{i-1}) & x_{i-1} < x < x_i \\ \frac{1}{h}(x_{i+1} - x) & x_i < x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$



Nodal Basis – System of Equations

- stiffness matrix:

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

- right hand sides (assume $f(x) = \alpha \in \mathbb{R}$):

$$\int_0^1 \varphi_i(x) f(x)\,\mathrm{dx} = \int_0^1 \varphi_i(x)\alpha\,\mathrm{dx} = \alpha h$$

- system of equations very similar to finite differences

Element Stiffness Matrices

- domain Ω splitted into finite elements $\Omega^{(k)}$:

$$\Omega = \Omega^{(1)} \cup \Omega^{(2)} \cup \cdots \cup \Omega^{(n)}$$

- observation: basis functions are defined element-wise
- use: $\int_a^b f(x)\mathrm{d}x = \int_a^c f(x)\mathrm{d}x + \int_c^b f(x)\mathrm{d}x$
- element-wise evaluation of the integrals:

$$\int_\Omega \nabla v \cdot \nabla u \,\mathrm{d}x = \sum_k \int_{\Omega^{(k)}} \nabla v \cdot \nabla u \,\mathrm{d}x$$

$$\int_\Omega v f \,\mathrm{d}x = \sum_i \int_{\Omega^{(i)}} v f \,\mathrm{d}x$$

Element Stiffness Matrices (2)

- leads to local stiffness matrices for each element:

$$\underbrace{\int_{\Omega^{(k)}} \nabla \phi_i \cdot \nabla \phi_j \, dx}_{=:A_{ij}^{(k)}}$$

- and respective element systems:

$$A^{(k)} x = b^{(k)}$$

- accumulate to obtain global system:

$$\underbrace{\sum_k A^{(k)}}_{=:A} x = \sum_k b^{(k)}$$

Element Stiffness Matrices (3)

Some comments on notation:

- assume: 1D problem, n elements (i.e. intervals)
- in each element only two basis functions are non-zero!
- hence, almost all $A_{ij}^{(k)}$ are zero:

$$A_{ij}^{(k)} = \int_{\Omega^{(k)}} \nabla\phi_i \cdot \nabla\phi_j \,\mathrm{d}x$$

- only 2×2 elements of $A^{(k)}$ are non-zero
- therefore convention to omit zero columns/rows
  $\Rightarrow$ leaves only unknowns that are in $\Omega^{(k)}$

Example: 1D Poisson

- $\Omega = [0,1]$ splitted into $\Omega^{(k)} = [x_{k-1}, x_k]$
- nodal basis; leads to element stiffness matrix:

$$A^{(k)} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

- consider only two elements:

$$A^{(1)} + A^{(2)} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

- in stencil notation:

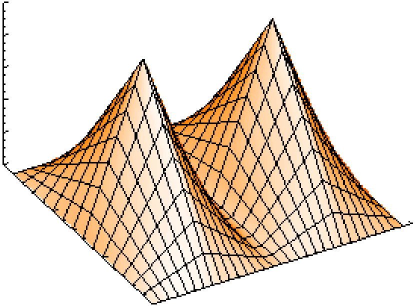
$$[-1 \quad 1^*] + [1^* \quad -1] \rightarrow [-1 \quad 2 \quad -1]$$

Example: 2D Poisson

- $-\Delta u = f$ on domain $\Omega = [0,1]^2$
- splitted into $\Omega^{(i,j)} = [x_{i-1}, x_i] \times [x_{j-1}, x_j]$
- bilinear basis functions

$$\varphi_{ij}(x,y) = \varphi_i(x)\varphi_j(y)$$

- "pagoda" functions



Example: 2D Poisson (2)

- leads to element stiffness matrix:

$$A^{(k)} = \begin{pmatrix} 2 & -\frac{1}{2} & -\frac{1}{2} & -1 \\ -\frac{1}{2} & 2 & -1 & -\frac{1}{2} \\ -\frac{1}{2} & -1 & 2 & -\frac{1}{2} \\ -1 & -\frac{1}{2} & -\frac{1}{2} & 2 \end{pmatrix}$$

- accumulation leads to 9-point stencil

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Typical workflow

1. choose elements:
   - quadratic or cubic cells
   - triangles (structured, unstructured)
   - tetrahedra, etc.
2. set up basis functions for each element $\Omega^{(k)}$; for example, at all nodes $x_i \in \Omega^{(k)}$

$$\begin{aligned} \varphi_i(x_i) &= 1 \\ \varphi_i(x_j) &= 0 \quad \text{for all } j \neq i \end{aligned}$$

3. for element stiffness matrix, compute all

$$A_{ij}^{(k)} = \int_{\Omega^{(k)}} \varphi_i L \varphi_j \, \mathrm{d}\Omega$$

4. accumulate global stiffness matrix

Time-Dependent Problems

Example: 1D Heat Equation

- $u_t = u_{xx} + f$ on domain $\Omega = [0,1]$ for $t \in [0, t_{\text{end}}]$
- spatial discretisation: weak form

$$\begin{aligned} \int v u_t \,\mathrm{dx} &= \int v u_{xx} \,\mathrm{dx} + \int v f \,\mathrm{dx} \\ \frac{\partial}{\partial t}\left(\int v u \,\mathrm{dx} \right) &= \int v u_{xx} \,\mathrm{dx} + \int v f \,\mathrm{dx} \end{aligned}$$

- spatial discretisation – finite elements:

$$\frac{\partial}{\partial t}(M_h u_h) = A_h u_h + f_h$$

M_h: mass matrix, A_h: stiffness matrix,
$u_h = u_h(t)$

Time-Dependent Problems (2)

Solve a system of ordinary differential equations:

- after spatial discretisation (M_h constant):

$$\frac{\partial}{\partial t} M_h(u_h) = A_h u_h + f_h$$

- u_h a vector of time-dependent functions:

$$u_h = (u_1(t), \ldots, u_i(t), \ldots, u_n(t))^T$$

- usually: approximate M_h by a simpler matrix (diagonal matrix, e.g.) $\rightarrow$ "mass lumping"

A Road to Theory

- weak formulation is equivalent to variational approach:
  solution u minimises an energy functional
- *best approximation* property:

$$\left\| u - u_h^{FE} \right\|_a \leq \inf_{u_h \in V} \| u - u_h \|_a$$

  in terms of the norm induced by the bilinear form a (energy norm)
- thus: error bounded by interpolation error (in energy norm)