Numerical Computation 2

Andrew Hazel

Introduction

- Want to solve partial differential equations that describe physical systems.
- Numerical (approximate) methods are often the only option for sufficiently complex problems.
- Method should be robust, accurate and, ideally, easy to implement.
- We shall describe methods for the solution of Navier–Stokes equations, equations of solid mechanics and fluid-structure interaction problems.

Initial value problems: "simulation"

- Many physical problems can be expressed as first-order evolution equations

$$\frac{\partial \mathbf{u}(t)}{\partial t} = \mathcal{F}(\mathbf{u}(t), t).$$

- If we know the initial state $\mathbf{u}(0)$, then computing evolution of the system is straightforward.
- Discretise time into uniform intervals of length Δt
- Let $\mathbf{u}^m = \mathbf{u}(m\Delta t)$ and then use an explicit discretisation of the time derivative and iterate *e.g.*

$$\mathbf{u}^{m+1} = \mathbf{u}^m + \Delta t + \mathcal{F}(\mathbf{u}^m, m\Delta t).$$

- Euler's method, Runge-Kutta schemes, etc.

Boundary value problems

- We shall fix ideas by a very simple example

$$\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = f(x), \quad u(0) = u(1) = 0, \tag{1}$$

where $x \in [0, 1]$ and $f(x)$ is a forcing function.

- Exact solutions:

$$\begin{array}{ll} f(x) = x, & u(x) = (x^3 - x)/6 \\ f(x) = \sin(2\pi x), & u(x) = -\frac{\sin(2\pi x)}{4\pi^2} \end{array}$$

- Equation (1) is a two-point boundary value problem.

Finite difference methods

- **Idea:** Divide interval $x \in [0, 1]$ into fixed number of points and evaluate (1) at each point.
- Divide domain into N intervals of width Δx and let $x_i = i\Delta x$ and $u_i = u(i\Delta x)$, $i \in 0, \ldots N$
- Thus, at the n-th point we have

$$\left[\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right]_n = f(x_n), \tag{2}$$

- How do we approximate $\left[\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right]_n$ using only u_i and x_i?

Finite difference methods

- **Idea:** Fit polynomial through points and differentiate it twice.
- For non-trivial second derivative, need quadratic, so fit polynomial through u_{n-1}, u_n and u_{n+1} and then

$$\left[\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right]_n \approx \frac{u_{n-1} - 2u_n + u_{n+1}}{(\Delta x)^2}.$$

- Local error in approximation is $O((\Delta x)^2)$ (Taylor series)
- Equation (2) becomes

$$\left[\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right]_n \approx \frac{u_{n-1} - 2u_n + u_{n+1}}{(\Delta x)^2} = f(x_n), \qquad (3)$$

- We have a discrete problem for $N+1$ unknowns u_i.

Finite difference methods: Boundary conditions

- Equation (3) can only be applied at $N-1$ interior points.
- Use the two boundary conditions

$$u(0) = u_0 = 0, \quad \text{and} \quad u(1) = u_N = 0.$$

- Now have $N+1$ equations for $N+1$ unknowns

$$\frac{1}{(\Delta x)^2}\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} 0 \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \\ 0 \end{pmatrix}.$$

Finite difference methods: Boundary conditions

- Equation (3) can only be applied at $N-1$ interior points.
- Use the two boundary conditions

$$u(0) = u_0 = 0, \quad \text{and} \quad u(1) = u_N = 0.$$

- Now have $N+1$ equations for $N+1$ unknowns

$$\begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = (\Delta x)^2 \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \end{pmatrix},$$

- Use row operations to eliminate u_0 and u_N
- A pure linear algebra problem!

Octave/MATLAB implementation

- $f(x) = x$

```
%Number of intervals
N=100;
%Distance between points
dx = 1/N;
%Set the x coordinates of each interior points as a column vector
x = linspace(dx,1-dx,N-1)';
%Set up column vector of ones
d = ones(N-1,1);
%Set up sparse matrix with the Poisson tridiagonal structure
A = spdiags([d -2*d d],[-1 0 1],N-1,N-1);
%Now solve the linear system
u = A\(x*dx*dx);
%Plot the solution
plot(x,u);
%Compute the 2-norm of the error
e = (u - (x.*x.*x - x)/6);
norm(e,2)
```

- Norm is below machine precision ... why?

Finite difference methods: Summary

- Conceptually simple.
- Increasing accuracy straightforward(ish).
- Implementation of more general boundary conditions not easy
- Extensions to higher dimensions and complex domains a challenge.

Finite volume methods

- **Idea:** Discretise conservation (integral) form of equation.
- Divide domain into N control volumes of length Δx.
- Integrate (1) over each control volume

$$\int_{x_n}^{x_{n+1}} \frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\,\mathrm{d}x = \left[\frac{\mathrm{d}u}{\mathrm{d}x}\right]_{x_n}^{x_{n+1}} = \int_{x_n}^{x_{n+1}} f(x)\,\mathrm{d}x.$$

$$\Rightarrow \left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x_{n+1}} - \left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x_n} = \int_{x_n}^{x_{n+1}} f(x)\mathrm{d}x. \qquad (4)$$

- How do we approximate the derivative of the unknown at the boundaries of each volume?
- How do we integrate the forcing function?

Finite volume methods

- **Idea:** Assume unknown varies linearly
- Let $u_{n+1/2}$ be the value halfway between u_n and u_{n+1}

$$\left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x_n} = \frac{u_{n+1/2} - u_{n-1/2}}{\Delta x}.$$

- Equation (4) becomes

$$\frac{u_{n+3/2} - 2u_{n+1/2} + u_{n-1/2}}{\Delta x} = \int_{x_n}^{x_{n+1}} f(x)\mathrm{d}x.$$

- **Idea:** Integral of forcing is average value, $\bar{f}$, multiplied by Δx

$$\frac{u_{n+3/2} - 2u_{n+1/2} + u_{n-1/2}}{(\Delta x)^2} = \bar{f}. \tag{5}$$

- Note similarity with finite difference scheme.

Finite volume methods: boundary conditions

- **Idea:** Value of unknowns at edge of domains are found by linear interpolation

$$u_0 = \frac{u_{1/2} + u_{-1/2}}{2}.$$

- $u_{-1/2}$ is an artificial or "ghost" point.

$$\frac{1}{(\Delta x)^2}\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} u_{-1/2} \\ u_{1/2} \\ u_{3/2} \\ \vdots \\ u_{N-3/2} \\ u_{N-1/2} \\ u_{N+1/2} \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{f}_{1/2} \\ \bar{f}_{3/2} \\ \vdots \\ \bar{f}_{N-3/2} \\ \bar{f}_{N-1/2} \\ 0 \end{pmatrix},$$

Finite volume methods: boundary conditions

- **Idea:** Value of unknowns at edge of domains are found by linear interpolation

$$u_0 = \frac{u_{1/2} + u_{-1/2}}{2}.$$

- $u_{-1/2}$ is an artificial or "ghost" point.

$$\begin{pmatrix} -3 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -3 \end{pmatrix} \begin{pmatrix} u_{1/2} \\ u_{3/2} \\ \vdots \\ u_{N-3/2} \\ u_{N-1/2} \end{pmatrix} = (\Delta x)^2 \begin{pmatrix} \bar{f}_{1/2} \\ \bar{f}_{3/2} \\ \vdots \\ \bar{f}_{N-3/2} \\ \bar{f}_{N-1/2} \end{pmatrix}.$$

- Use row operations again

Finite volume methods: Summary

- Need an integral form of the equation.
- Easier handling of flux boundary conditions.
- Increasing accuracy not so straightforward.
- Implementation of more general boundary conditions not easy.
- Extensions to complex domains a challenge.

Finite element method

- **Idea:** Start from weak (rather than strong) form of equation.
- Strong form of problem in residual form

$$\mathcal{R}(x, u(x)) \equiv \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} - f(x) = 0, \quad u(0) = u(1) = 0,$$

- Weak form is a weighted residual

$$\int_0^1 \mathcal{R}(x, u(x))\, \phi_i(x) \mathrm{d}x = 0, \quad u(0) = u(1) = 0,$$

where $\phi_i(x)$ are any test functions for which the integral exists.

Finite element method

- **Idea:** Start from weak (rather than strong) form of equation.
- Strong form of problem in residual form

$$\mathcal{R}(x, u(x)) \equiv \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} - f(x) = 0, \quad u(0) = u(1) = 0,$$

- Weak form is a weighted residual

$$\int_0^1 \mathcal{R}(x, u(x))\, \phi_i(x) \mathrm{d}x = 0, \quad u(0) = u(1) = 0,$$

  where $\phi_i(x)$ are any test functions for which the integral exists.
- Choosing $\phi_i(x) = \delta(x - x_i)$, $i \in \{1, N-1\}$, gives finite difference approximation.

Finite element method

- **Idea:** Start from weak (rather than strong) form of equation.
- Strong form of problem in residual form

$$\mathcal{R}(x, u(x)) \equiv \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} - f(x) = 0, \quad u(0) = u(1) = 0,$$

- Weak form is a weighted residual

$$\int_0^1 \mathcal{R}(x, u(x))\, \phi_i(x) \mathrm{d}x = 0, \quad u(0) = u(1) = 0,$$

  where $\phi_i(x)$ are any test functions for which the integral exists.
- Choosing $\phi_i(x) = \delta(x - x_i)$, $i \in \{1, N-1\}$, gives finite difference approximation.
- Choosing

$$\phi_i(x) = \begin{cases} 1 & x_i < x < x_{i+1}, \\ 0 & \text{otherwise}, \end{cases}$$

  leads to the finite volume approximation.

Finite element method

- **Idea:** Reduce differentiability requirements (integrate by parts)

$$\left[\frac{\mathrm{d}u}{\mathrm{d}x}\phi_i(x)\right]_0^1 - \int_0^1 \frac{\mathrm{d}u}{\mathrm{d}x}\frac{\mathrm{d}\phi_i}{\mathrm{d}x}\,\mathrm{d}x - \int_0^1 f(x)\phi_i(x)\,\mathrm{d}x = 0.$$

- If test functions vanish on the domain boundaries $\phi_i(0) = \phi_i(1) = 0$ then

$$\int_0^1 \frac{\mathrm{d}u}{\mathrm{d}x}\frac{\mathrm{d}\phi_i}{\mathrm{d}x}\,\mathrm{d}x + \int_0^1 f(x)\phi_i(x)\,\mathrm{d}x = 0.$$

- How do we approximate u?
- How do we choose ϕ_i?
- How do we integrate?

Finite element method

- **Idea:** Expand u in a (truncated) set of basis functions.

$$u(x) = \sum_{j=0}^{N} u_j \psi_j(x).$$

- **Idea:** Choose the test functions to be equal to the basis functions $\phi_i(x) = \psi_i(x)$ (Galerkin method).

$$\int_0^1 \sum_{j=0}^{N} u_j \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \, \mathrm{d}x + \int_0^1 f(x)\psi_i(x) \, \mathrm{d}x = 0.$$

- We now have $N + 1$ equations for $N + 1$ unknowns.

Finite element method

- **Idea:** Expand u in a (truncated) set of basis functions.

$$u(x) = \sum_{j=0}^{N} u_j \psi_j(x).$$

- **Idea:** Choose the test functions to be equal to the basis functions $\phi_i(x) = \psi_i(x)$ (Galerkin method).

$$\int_0^1 \sum_{j=0}^{N} u_j \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \, \mathrm{d}x + \int_0^1 f(x)\psi_i(x) \, \mathrm{d}x = 0.$$

- We now have $N+1$ equations for $N+1$ unknowns.
- If the derivatives of the basis functions are orthogonal with respect to the inner product defined by the integral then we will have an explicit formula for the unknown coefficients $\{u_j\}$.
- Could pick eigenfunctions of the Laplace operator that vanish at the domain boundaries (spectral method).

Finite element method

- **Idea:** Choose low dimensional polynomials of finite support as a basis.
- For $N+1$ nodes at locations x_j, $j \in \{0, N\}$ choose basis functions so that $\psi_i(x_j) = \delta_{ij}$.
- If the functions are piecewise linear then

$$\psi_i(x) = \begin{cases} 0 & x < x_{i-1} \\ \frac{x - x_{i-1}}{x_i - x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & x_i \leq x \leq x_{i+1} \\ 0 & x > x_{i+1} \end{cases}$$

Finite element method

- Hence the governing equation becomes

$$\sum_{j=i-1}^{i+1} u_j \int_{x_{i-1}}^{x_{i+1}} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \, \mathrm{d}x + \int_{x_{i-1}}^{x_{i+1}} f(x) \psi_i(x) \, \mathrm{d}x = 0.$$

- Assume uniformly spaced nodes such that $x_{i+1} - x_i = \Delta x$, then

$$\frac{\mathrm{d}\psi_i(x)}{\mathrm{d}x} = \begin{cases} 0 & x < x_{i-1} \\ 1/\Delta x & x_{i-1} \leq x \leq x_i \\ -1/\Delta x & x_i \leq x \leq x_{i+1} \\ 0 & x > x_{i+1} \end{cases}$$

Finite element method

- **Idea:** Break up the integral into sum of integrals over elements
- Define the element e_i to be the domain $x_{i-1} \leq x \leq x_i$.

$$\sum_{j=i-1}^{i+1} u_j \int_{x_{i-1}}^{x_i} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \,\mathrm{d}x + \sum_{j=i-1}^{i+1} u_j \int_{x_i}^{x_{i+1}} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \,\mathrm{d}x$$

$$+ \int_{x_{i-1}}^{x_i} f(x)\psi_i(x) \,\mathrm{d}x + \int_{x_i}^{x_{i+1}} f(x)\psi_i(x) \,\mathrm{d}x = 0.$$

Finite element method

- **Idea:** Break up the integral into sum of integrals over elements
- Define the element e_i to be the domain $x_{i-1} \le x \le x_i$.

$$\sum_{j=i-1}^{i+1} u_j \int_{x_{i-1}}^{x_i} \frac{\mathrm{d}\psi_j}{\mathrm{d}x}\frac{\mathrm{d}\psi_i}{\mathrm{d}x}\,\mathrm{d}x + \sum_{j=i-1}^{i+1} u_j \int_{x_i}^{x_{i+1}} \frac{\mathrm{d}\psi_j}{\mathrm{d}x}\frac{\mathrm{d}\psi_i}{\mathrm{d}x}\,\mathrm{d}x$$

$$+ \int_{x_{i-1}}^{x_i} f(x)\psi_i(x)\,\mathrm{d}x + \int_{x_i}^{x_{i+1}} f(x)\psi_i(x)\,\mathrm{d}x = 0.$$

- Now

$$\int_{x_{i-1}}^{x_i} \frac{\mathrm{d}\psi_j}{\mathrm{d}x}\frac{\mathrm{d}\psi_i}{\mathrm{d}x}\,\mathrm{d}x = \begin{cases} -1/\Delta x, & j = i-1, \\ 1/\Delta x, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Finite element method

- **Idea:** Break up the integral into sum of integrals over elements
- Define the element e_i to be the domain $x_{i-1} \leq x \leq x_i$.

$$\sum_{j=i-1}^{i+1} u_j \int_{x_{i-1}}^{x_i} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \,\mathrm{d}x + \sum_{j=i-1}^{i+1} u_j \int_{x_i}^{x_{i+1}} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \,\mathrm{d}x$$

$$+ \int_{x_{i-1}}^{x_i} f(x)\psi_i(x)\,\mathrm{d}x + \int_{x_i}^{x_{i+1}} f(x)\psi_i(x)\,\mathrm{d}x = 0.$$

- and

$$\int_{x_i}^{x_{i+1}} \frac{\mathrm{d}\psi_j}{\mathrm{d}x} \frac{\mathrm{d}\psi_i}{\mathrm{d}x} \,\mathrm{d}x = \begin{cases} 1/\Delta x, & j = i, \\ -1/\Delta x, & j = i+1, \\ 0, & \text{otherwise.} \end{cases}$$

Finite element method

- Hence the equation corresponding to the i-th test function is

$$u_{i-1} - 2u_i + u_{i+1} = \int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1})\,\mathrm{d}x + \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x)\,\mathrm{d}x.$$

- If the forcing is linear $f(x) = x$, we recover the finite difference method

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2} = x_i.$$

- Had to be much more specific about choices of basis and test functions.
- Easier to see how to generalise the method.

Finite element method: Summary

- Harder to understand ... more steps required.
- Weighted residuals can encompass other methods.
- Weak form broadens the class of allowed solutions.
- Easier to generalise to non-uniform spacing.
- Easier to generalise to higher-order approximations.