Object-oriented scientific computing

Pras Pathmanathan

Summer 2012

Nonlinear Problems

Nonlinear problems

Consider a nonlinear elliptic problem, such as

$$\nabla \cdot (D(u)\boldsymbol{\nabla} u) + f = 0$$

with boundary conditions

$$\begin{aligned} u &= 0 && \text{on } \Gamma_1 \\ D(u)\boldsymbol{\nabla} u \cdot \mathbf{n} &= g && \text{on } \Gamma_2 \end{aligned}$$

Computing the weak form as before, we obtain: find $u \in \mathcal{V}_0$ satisfying

$$\int_\Omega (D(u)\boldsymbol{\nabla} u) \cdot \boldsymbol{\nabla} v \, \mathrm{d}V - \int_\Omega fv \, \mathrm{d}V - \int_{\Gamma_2} gv \, \mathrm{d}S = 0 \qquad \forall v \in \mathcal{V}_0$$

Write this as: find $u \in \mathcal{V}_0$ satisfying

$$\mathcal{F}(u, v) = 0 \qquad \forall v \in \mathcal{V}_0$$

Nonlinear problems

Consider a nonlinear elliptic problem, such as

$$\nabla \cdot (D(u)\boldsymbol{\nabla} u) + f = 0$$

with boundary conditions

$$\begin{aligned} u &= 0 && \text{on } \Gamma_1 \\ D(u)\boldsymbol{\nabla} u \cdot \mathbf{n} &= g && \text{on } \Gamma_2 \end{aligned}$$

Computing the weak form as before, we obtain: find $u \in \mathcal{V}_0$ satisfying

$$\int_\Omega (D(u)\boldsymbol{\nabla} u) \cdot \boldsymbol{\nabla} v \,\mathrm{d}V - \int_\Omega fv \,\mathrm{d}V - \int_{\Gamma_2} gv \,\mathrm{d}S = 0 \qquad \forall v \in \mathcal{V}_0$$

Write this as: find $u \in \mathcal{V}_0$ satisfying

$$\mathcal{F}(u, v) = 0 \qquad \forall v \in \mathcal{V}_0$$

Nonlinear problems

Consider a nonlinear elliptic problem, such as

$$\nabla \cdot (D(u)\boldsymbol{\nabla} u) + f = 0$$

with boundary conditions

$$\begin{aligned} u &= 0 && \text{on } \Gamma_1 \\ D(u)\boldsymbol{\nabla} u \cdot \mathbf{n} &= g && \text{on } \Gamma_2 \end{aligned}$$

Computing the weak form as before, we obtain: find $u \in \mathcal{V}_0$ satisfying

$$\int_\Omega (D(u)\boldsymbol{\nabla} u) \cdot \boldsymbol{\nabla} v \, \mathrm{d}V - \int_\Omega fv \, \mathrm{d}V - \int_{\Gamma_2} gv \, \mathrm{d}S = 0 \qquad \forall v \in \mathcal{V}_0$$

Write this as: find $u \in \mathcal{V}_0$ satisfying

$$\mathcal{F}(u, v) = 0 \qquad \forall v \in \mathcal{V}_0$$

Nonlinear problems

The finite element problem is obtained as before: find $u_h \in \mathcal{V}_0^h$ satisfying

$$\mathcal{F}(u_h, v) = 0 \qquad \forall v \in \mathcal{V}_0^h$$

i.e. find coefficients $U_1, \ldots, U_N$ of $u_h = \sum U_i \phi_i$ such that

$$\mathcal{F}(u_h, \phi_i) = 0 \qquad \text{for } i = 1, \ldots, N$$

This is a general N-dimensional nonlinear system.

An iterative approach is required to solve nonlinear systems. Let u_h^k (equivalently, $\mathbf{U^k} = [U_1^k, \ldots, U_N^k]$) be the current guess. Then the vector $\mathbf{F}^k$ defined by

$$F_i^k = \mathcal{F}(u_h^k, \phi_i)$$

is known as the k-th **residual vector**. We require a guess satisfying

$$\|\mathbf{F}^k\| < \mathrm{TOL}$$

Nonlinear problems

The finite element problem is obtained as before: find $u_h \in \mathcal{V}_0^h$ satisfying

$$\mathcal{F}(u_h, v) = 0 \qquad \forall v \in \mathcal{V}_0^h$$

i.e. find coefficients $U_1, \ldots, U_N$ of $u_h = \sum U_i \phi_i$ such that

$$\mathcal{F}(u_h, \phi_i) = 0 \qquad \text{for } i = 1, \ldots, N$$

This is a general N-dimensional nonlinear system.

An iterative approach is required to solve nonlinear systems. Let u_h^k (equivalently, $\mathbf{U^k} = [U_1^k, \ldots, U_N^k]$) be the current guess. Then the vector $\mathbf{F}^k$ defined by

$$F_i^k = \mathcal{F}(u_h^k, \phi_i)$$

is known as the k-th **residual vector**. We require a guess satisfying

$$\|\mathbf{F}^k\| < \mathrm{TOL}$$

Nonlinear problems

The finite element problem is obtained as before: find $u_h \in \mathcal{V}_0^h$ satisfying

$$\mathcal{F}(u_h, v) = 0 \qquad \forall v \in \mathcal{V}_0^h$$

i.e. find coefficients $U_1, \ldots, U_N$ of $u_h = \sum U_i \phi_i$ such that

$$\mathcal{F}(u_h, \phi_i) = 0 \qquad \text{for } i = 1, \ldots, N$$

This is a general N-dimensional nonlinear system.

An iterative approach is required to solve nonlinear systems. Let u_h^k (equivalently, $\mathbf{U^k} = [U_1^k, \ldots, U_N^k]$) be the current guess. Then the vector $\mathbf{F}^k$ defined by

$$F_i^k = \mathcal{F}(u_h^k, \phi_i)$$

is known as the k-th **residual vector**. We require a guess satisfying

$$\|\mathbf{F}^k\| < \mathrm{TOL}$$

Newton's method

Suppose we want to solve the nonlinear set of N equations

$$\mathbf{F}(\mathbf{U}) = 0$$

Given an initial guess $\mathbf{U}^0$, Newton's method is: let $\mathbf{U}^{k+1} = \mathbf{U}^k + \delta\mathbf{U}^{k+1}$, where $\delta\mathbf{U}^{k+1}$ satisfies the linear system

$$J(\mathbf{U}^k)\,\delta\mathbf{U}^{k+1} = -\mathbf{F}(\mathbf{U}^k)$$

where $J_{ij} = \frac{\partial F_i}{\partial U_j}$.

Newton's method provides quadratic convergence when the current guess is 'close enough' to the true solution. To avoid initial divergence however, it may be necessary to use **damping**

$$\mathbf{U}^{k+1} = \mathbf{U}^k + s^k\delta\mathbf{U}^{k+1}$$

for some s^k generally smaller than 1. (There are various ways to go about choosing s^k, the simplest is to pick one from a small list of possibilities which leads to the smallest $\|\mathbf{F}\|$).

Newton's method

Suppose we want to solve the nonlinear set of N equations

$$\mathbf{F}(\mathbf{U}) = 0$$

Given an initial guess $\mathbf{U}^0$, Newton's method is: let $\mathbf{U}^{k+1} = \mathbf{U}^k + \delta\mathbf{U}^{k+1}$, where $\delta\mathbf{U}^{k+1}$ satisfies the linear system

$$J(\mathbf{U}^k)\,\delta\mathbf{U}^{k+1} = -\mathbf{F}(\mathbf{U}^k)$$

where $J_{ij} = \frac{\partial F_i}{\partial U_j}$.

Newton's method provides quadratic convergence when the current guess is 'close enough' to the true solution. To avoid initial divergence however, it may be necessary to use **damping**

$$\mathbf{U}^{k+1} = \mathbf{U}^k + s^k\delta\mathbf{U}^{k+1}$$

for some s^k generally smaller than 1. (There are various ways to go about choosing s^k, the simplest is to pick one from a small list of possibilities which leads to the smallest $\|\mathbf{F}\|$).

Alternative nonlinear solvers

- There are other methods for solving nonlinear systems, for example solve $x = f(x)$ using **fixed point iterations**: $x^{n+1} = f(x^n)$.
- For $\mathbf{F}(\mathbf{U}) = 0$, this is

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{F}(\mathbf{U}^n)$$

- Very loosely speaking, methods which use the Jacobian will be more effective.
- If used, the Jacobian can be either provided analytically (if so, has to be calculated on paper on paper and coded up); or estimated numerically (slow).

Alternative nonlinear solvers

- There are other methods for solving nonlinear systems, for example solve $x = f(x)$ using **fixed point iterations**: $x^{n+1} = f(x^n)$.
- For $\mathbf{F}(\mathbf{U}) = 0$, this is

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{F}(\mathbf{U}^n)$$

- Very loosely speaking, methods which use the Jacobian will be more effective.
- If used, the Jacobian can be either provided analytically (if so, has to be calculated on paper on paper and coded up); or estimated numerically (slow).

Alternative nonlinear solvers

- There are other methods for solving nonlinear systems, for example solve $x = f(x)$ using **fixed point iterations**: $x^{n+1} = f(x^n)$.
- For $\mathbf{F}(\mathbf{U}) = 0$, this is

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{F}(\mathbf{U}^n)$$

- Very loosely speaking, methods which use the Jacobian will be more effective.
- If used, the Jacobian can be either provided analytically (if so, has to be calculated on paper on paper and coded up); or estimated numerically (slow).

Simplifications with time-dependent problems

Nonlinear problems that are time-dependent can sometimes be dealt with straightforwardly. Consider, for example, a nonlinear diffusion equation:

$$u_t = \nabla \cdot (D(u)\boldsymbol{\nabla} u)$$

Consider some time-discretisations. The following is fully-explicit so will suffer from stability problems:

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\nabla u^n)$$

However, this next discretisation is *semi-implicit* and no longer nonlinear (linear in u^{n+1})

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\nabla u^{n+1})$$

Simplifications with time-dependent problems

Nonlinear problems that are time-dependent can sometimes be dealt with straightforwardly. Consider, for example, a nonlinear diffusion equation:

$$u_t = \nabla \cdot (D(u)\boldsymbol{\nabla} u)$$

Consider some time-discretisations. The following is fully-explicit so will suffer from stability problems:

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\boldsymbol{\nabla} u^n)$$

However, this next discretisation is *semi-implicit* and no longer nonlinear (linear in u^{n+1})

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\nabla u^{n+1})$$

Simplifications with time-dependent problems

Nonlinear problems that are time-dependent can sometimes be dealt with straightforwardly. Consider, for example, a nonlinear diffusion equation:

$$u_t = \nabla \cdot (D(u)\boldsymbol{\nabla} u)$$

Consider some time-discretisations. The following is fully-explicit so will suffer from stability problems:

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\boldsymbol{\nabla} u^n)$$

However, this next discretisation is *semi-implicit* and no longer nonlinear (linear in u^{n+1})

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot (D(u^n))\boldsymbol{\nabla} u^{n+1})$$

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix)
  - Solve $J\Delta\mathbf{U} = -\mathbf{F}^k$
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\Delta\mathbf{U}$, choosing s appropriately
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). Now increment k

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\Delta\mathbf{U} = -\mathbf{F}^k$
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\Delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). Now increment k.

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\,\delta\mathbf{U} = -\mathbf{F}^k$.
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\,\delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). Increment k.

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\,\delta\mathbf{U} = -\mathbf{F}^k$.
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\,\delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). [Now increment k]

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\,\delta\mathbf{U} = -\mathbf{F}^k$.
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\,\delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). [Now increment k]

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\,\delta\mathbf{U} = -\mathbf{F}^k$.
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\,\delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). [Now increment k]

Solving nonlinear problems with finite elements and Newton's method

Firstly, decide whether the Jacobian (if used) is to be computed numerically or analytically. If the latter, both the residual and Jacobian need to be 'assembled' in a finite element manner.

- Choose a (good!) initial guess $\mathbf{U}^0$
- Compute the initial residual $\mathbf{F}^0 = \mathbf{F}(\mathbf{U}^0)$ (loop over elements, compute elemental contribution, add to full vector).
- While $\|\mathbf{F}^k\| > \mathrm{TOL}$
  - Compute $J(\mathbf{U}^k)$ (loop over elements, compute elemental contribution, add to full matrix).
  - Solve $J\,\delta\mathbf{U} = -\mathbf{F}^k$.
  - Set $\mathbf{U}^{k+1} = \mathbf{U}^k + s\,\delta\mathbf{U}$, choosing s appropriately.
  - Compute $\mathbf{F}^{k+1}$ (loop over elements, compute elemental contribution, add to full vector). [Now increment k]

Summary

Summary

ODEs

- Various solvers possible, which differ in their *order of convergence* and *stability* properties
- Explicit solvers are fast, implicit solvers are stable.
- We haven't discussed adaptive solvers, which tend to be the best type

PDEs

- Finite differences is easy to implement and appropriate for simple equations & regular geometries & no Neumman boundary conditions
- Finite elements is better for more complicated problems
- For finite elements: set up weak problem (on paper), then: choose mesh/basis functions, assemble linear system, solve linear system

Summary

ODEs

- Various solvers possible, which differ in their *order of convergence* and *stability* properties
- Explicit solvers are fast, implicit solvers are stable.
- We haven't discussed adaptive solvers, which tend to be the best type

PDEs

- Finite differences is easy to implement and appropriate for simple equations & regular geometries & no Neumman boundary conditions
- Finite elements is better for more complicated problems
- For finite elements: set up weak problem (on paper), then: choose mesh/basis functions, assemble linear system, solve linear system

Summary

Nonlinear problems

- Best method is usually Newton's method
- Really need to analytically calculate the Jacobian though, which can be painful
- Other options: numerical Jacobian, fixed point, semi-implicit discretisation for time-dependent problems

Linear systems

- Direct solvers / Gaussian elimination is appropriate for smaller problems
- Iterative solvers are used for larger problems
- Preconditioning the system is important for iterative methods

Summary

Nonlinear problems

- Best method is usually Newton's method
- Really need to analytically calculate the Jacobian though, which can be painful
- Other options: numerical Jacobian, fixed point, semi-implicit discretisation for time-dependent problems

Linear systems

- Direct solvers / Gaussian elimination is appropriate for smaller problems
- Iterative solvers are used for larger problems
- Preconditioning the system is important for iterative methods

Summary

Computing integrals

- Numerical quadrature rules make this easy
- Transform integral to reference triangle or unit square - evaluate function at quad points and sum
- Just need to look up quadrature points and weights

An overview of alternative methods for solving PDEs (not lectured)

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \, \mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \, \mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \, \mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \, \mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \, \mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \, \mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \, \mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \, \mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \, \mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \, \mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \, \mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \, \mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \, \mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \, \mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \, \mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \, \mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \, \mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \, \mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \, \mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \, \mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \,\mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \,\mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \,\mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \,\mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \,\mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Finite volume methods

- Very commonly used for hyperbolic PDEs (for which the FE method tends to have trouble) and in computational fluid dynamics
- As with FE, FV is based on *integral formulations*.
- The domain is broken down into *control volumes* (similar to 'elements').
- One unknown computed per element (i.e. no need for 'node')—this can be considered to be the average value of u in the control volume.

Consider the advection equation $u_t + \nabla \cdot \mathbf{f}(u) = 0$. Integrate over a control volume Ω_i of volume V_i:

$$\int_{\Omega_i} u_t \,\mathrm{d}V = \int_{\Omega_i} -\nabla \cdot \mathbf{f}(u) \,\mathrm{d}V = -\int_{\partial\Omega_i} \mathbf{f}(u) \cdot \mathbf{n} \,\mathrm{d}S$$

Using an explicit time-discretisation, and $\int_{\Omega_i} U^n \,\mathrm{d}V \approx V_i U_i^n$, we obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{V_i} \int_{\partial\Omega_i} \mathbf{f}(u^n) \cdot \mathbf{n} \,\mathrm{d}S$$

See eg `http://www.comp.leeds.ac.uk/meh/Talks/FVTutorial.pdf` for more details

Methods of weighted residuals

In FE we used an *integral formulation of the PDE*, eg: *find* $u \in \mathcal{V}$ *such that:*

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, \mathrm{d}V = \int_\Omega f v \, \mathrm{d}V + \int_{\Gamma_2} g v \, \mathrm{d}S \quad \forall v \in \mathcal{V}$$

Write this as: *find* $u \in \mathcal{V}$ *such that:* $a(u, v) = l(v) \quad \forall v \in \mathcal{V}$

To *discretise* the integral equation, we replace $\mathcal{V}$ by *finite-dimensional subspaces* (of dimension N): *find* $u^{approx} \in \mathcal{W}_1$ *such that:*

$$a(u^{approx}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

Choosing bases:

$$\begin{aligned} \mathcal{W}_1 &= \mathrm{span}\{\phi_1, \ldots, \phi_N\} \\ \mathcal{W}_2 &= \mathrm{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

(ie $u^{approx} = \sum \alpha_i \phi_i$), we can obtain N equations for N unknowns.

Different methods are based on different choices of $\mathcal{W}_1$ and $\mathcal{W}_2$.

Methods of weighted residuals

In FE we used an *integral formulation of the PDE*, eg: *find* $u \in \mathcal{V}$ *such that:*

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, \mathrm{d}V = \int_\Omega f v \, \mathrm{d}V + \int_{\Gamma_2} g v \, \mathrm{d}S \quad \forall v \in \mathcal{V}$$

Write this as: *find* $u \in \mathcal{V}$ *such that:* $a(u, v) = l(v) \quad \forall v \in \mathcal{V}$

To *discretise* the integral equation, we replace $\mathcal{V}$ by *finite-dimensional subspaces* (of dimension N): *find* $u^{approx} \in \mathcal{W}_1$ *such that:*

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

Choosing bases:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\} \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

(ie $u^{\text{approx}} = \sum \alpha_i \phi_i$), we can obtain N equations for N unknowns.

Different methods are based on different choices of $\mathcal{W}_1$ and $\mathcal{W}_2$.

Methods of weighted residuals

In FE we used an *integral formulation of the PDE*, eg: *find* $u \in \mathcal{V}$ *such that:*

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, \mathrm{d}V = \int_\Omega f v \, \mathrm{d}V + \int_{\Gamma_2} g v \, \mathrm{d}S \quad \forall v \in \mathcal{V}$$

Write this as: *find* $u \in \mathcal{V}$ *such that:* $a(u, v) = l(v) \quad \forall v \in \mathcal{V}$

To *discretise* the integral equation, we replace $\mathcal{V}$ by *finite-dimensional subspaces* (of dimension N): *find* $u^{approx} \in \mathcal{W}_1$ *such that:*

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

Choosing bases:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\} \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

(ie $u^{\text{approx}} = \sum \alpha_i \phi_i$), we can obtain N equations for N unknowns.

Different methods are based on different choices of $\mathcal{W}_1$ and $\mathcal{W}_2$.

Methods of weighted residuals

In FE we used an *integral formulation of the PDE*, eg: *find* $u \in \mathcal{V}$ *such that:*

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, \mathrm{d}V = \int_\Omega f v \, \mathrm{d}V + \int_{\Gamma_2} g v \, \mathrm{d}S \quad \forall v \in \mathcal{V}$$

Write this as: *find* $u \in \mathcal{V}$ *such that:* $a(u, v) = l(v) \quad \forall v \in \mathcal{V}$

To *discretise* the integral equation, we replace $\mathcal{V}$ by *finite-dimensional subspaces* (of dimension N): *find* $u^{approx} \in \mathcal{W}_1$ *such that:*

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

Choosing bases:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\} \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

(ie $u^{\text{approx}} = \sum \alpha_i \phi_i$), we can obtain N equations for N unknowns.

Different methods are based on different choices of $\mathcal{W}_1$ and $\mathcal{W}_2$.

Methods of weighted residuals

Find $u^{\text{approx}} \in \mathcal{W}_1$ such that:

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

with:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\}. \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

- **Galerkin methods**: use $\phi = \chi$, i.e. $\mathcal{W}_1 = \mathcal{W}_2$
- **Collocation methods**: use δ-functions for χ's (i.e. replace integrals with point evaluations (at N collocation points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$)
- **(Continuous) Galerkin FEM**: use $\mathcal{W}_1 = \mathcal{W}_2$ and take the ϕ_i to be continuous and **piecewise polynomial** on element
  - As we know in practice we just consider 1 canonical element and define the basis functions on this (the *shape functions*)
  - Elements could be tetrahedral/hexahedral, shape functions could be linear, quadratic, cubic Hermite and more...
- **Discontinuous Galerkin FEM**: ϕ_i piecewise polynomial but no longer continuous across elements
- **Spectral methods**: ϕ_k globally continuous and infinitely differentiable (for example, $\phi_k(x) = exp(ikx)$)

Methods of weighted residuals

Find $u^{\text{approx}} \in \mathcal{W}_1$ such that:

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

with:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\}. \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

- **Galerkin methods**: use $\phi = \chi$, i.e. $\mathcal{W}_1 = \mathcal{W}_2$
- **Collocation methods**: use δ-functions for χ's (i.e. replace integrals with point evaluations (at N collocation points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$)
- **(Continuous) Galerkin FEM**: use $\mathcal{W}_1 = \mathcal{W}_2$ and take the ϕ_i to be continuous and **piecewise polynomial** on element
  - As we know in practice we just consider 1 canonical element and define the basis functions on this (the *shape functions*)
  - Elements could be tetrahedral/hexahedral, shape functions could be linear, quadratic, cubic Hermite and more...
- **Discontinuous Galerkin FEM**: ϕ_i piecewise polynomial but no longer continuous across elements
- **Spectral methods**: ϕ_k globally continuous and infinitely differentiable (for example, $\phi_k(x) = exp(ikx)$)

Methods of weighted residuals

Find $u^{\text{approx}} \in \mathcal{W}_1$ such that:

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

with:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\}. \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

- **Galerkin methods**: use $\phi = \chi$, i.e. $\mathcal{W}_1 = \mathcal{W}_2$
- **Collocation methods**: use δ-functions for χ's (i.e. replace integrals with point evaluations (at N collocation points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$)
- **(Continuous) Galerkin FEM**: use $\mathcal{W}_1 = \mathcal{W}_2$ and take the ϕ_i to be continuous and **piecewise polynomial** on element
  - As we know in practice we just consider 1 canonical element and define the basis functions on this (the *shape functions*)
  - Elements could be tetrahedral/hexahedral, shape functions could be linear, quadratic, cubic Hermite and more...
- **Discontinuous Galerkin FEM**: ϕ_i piecewise polynomial but no longer continuous across elements
- **Spectral methods**: ϕ_k globally continuous and infinitely differentiable (for example, $\phi_k(x) = exp(ikx)$)

Methods of weighted residuals

Find $u^{\text{approx}} \in \mathcal{W}_1$ such that:

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

with:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\}. \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

- **Galerkin methods**: use $\phi = \chi$, i.e. $\mathcal{W}_1 = \mathcal{W}_2$
- **Collocation methods**: use δ-functions for χ's (i.e. replace integrals with point evaluations (at N collocation points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$)
- **(Continuous) Galerkin FEM**: use $\mathcal{W}_1 = \mathcal{W}_2$ and take the ϕ_i to be continuous and **piecewise polynomial** on element
  - As we know in practice we just consider 1 canonical element and define the basis functions on this (the *shape functions*)
  - Elements could be tetrahedral/hexahedral, shape functions could be linear, quadratic, cubic Hermite and more...
- **Discontinuous Galerkin FEM**: ϕ_i piecewise polynomial but no longer continuous across elements
- **Spectral methods**: ϕ_k globally continuous and infinitely differentiable (for example, $\phi_k(x) = exp(ikx)$)

Methods of weighted residuals

Find $u^{\text{approx}} \in \mathcal{W}_1$ such that:

$$a(u^{\text{approx}}, v) = l(v) \quad \forall v \in \mathcal{W}_2$$

with:

$$\begin{aligned} \mathcal{W}_1 &= \text{span}\{\phi_1, \ldots, \phi_N\}. \\ \mathcal{W}_2 &= \text{span}\{\chi_1, \ldots, \chi_N\} \end{aligned}$$

- **Galerkin methods**: use $\phi = \chi$, i.e. $\mathcal{W}_1 = \mathcal{W}_2$
- **Collocation methods**: use δ-functions for χ's (i.e. replace integrals with point evaluations (at N collocation points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$)
- **(Continuous) Galerkin FEM**: use $\mathcal{W}_1 = \mathcal{W}_2$ and take the ϕ_i to be continuous and **piecewise polynomial** on element
  - As we know in practice we just consider 1 canonical element and define the basis functions on this (the *shape functions*)
  - Elements could be tetrahedral/hexahedral, shape functions could be linear, quadratic, cubic Hermite and more...
- **Discontinuous Galerkin FEM**: ϕ_i piecewise polynomial but no longer continuous across elements
- **Spectral methods**: ϕ_k globally continuous and infinitely differentiable (for example, $\phi_k(x) = exp(ikx)$)

Spectral methods

- There are both spectral-collocation methods (work with the strong form) or spectral-Galerkin methods (work with the weak form)

- Various choices of basis functions ($\mathcal{W}_1$) are possible, for example
- For problems with **periodic boundary conditions**, use $\phi_k(x) = exp(ikx)$
  - i.e. $u^{approx} = \sum \alpha_k \phi_k$ approximates u with a cut-off Fourier series
- For problems with **non-periodic boundary conditions**: use a set of 'orthogonal polynomials' for ϕ_k, such a Legendre or Chebychev polynomials

- For problems with *smooth data* (initial condition, boundary conditions, forces etc are smooth functions), spectral methods give exceptional rates of convergence.
- For more info, see e.g. `http://www.lorene.obspm.fr/palma.pdf`

Spectral methods

- There are both spectral-collocation methods (work with the strong form) or spectral-Galerkin methods (work with the weak form)

- Various choices of basis functions ($\mathcal{W}_1$) are possible, for example
- For problems with **periodic boundary conditions**, use $\phi_k(x) = exp(ikx)$
  - i.e. $u^{approx} = \sum \alpha_k \phi_k$ approximates u with a cut-off Fourier series
- For problems with **non-periodic boundary conditions**: use a set of 'orthogonal polynomials' for ϕ_k, such a Legendre or Chebychev polynomials

- For problems with *smooth data* (initial condition, boundary conditions, forces etc are smooth functions), spectral methods give exceptional rates of convergence.
- For more info, see e.g. `http://www.lorene.obspm.fr/palma.pdf`

Spectral methods

- There are both spectral-collocation methods (work with the strong form) or spectral-Galerkin methods (work with the weak form)

- Various choices of basis functions ($\mathcal{W}_1$) are possible, for example
- For problems with **periodic boundary conditions**, use $\phi_k(x) = exp(ikx)$
  - i.e. $u^{\text{approx}} = \sum \alpha_k \phi_k$ approximates u with a cut-off Fourier series
- For problems with **non-periodic boundary conditions**: use a set of 'orthogonal polynomials' for ϕ_k, such a Legendre or Chebychev polynomials

- For problems with *smooth data* (initial condition, boundary conditions, forces etc are smooth functions), spectral methods give exceptional rates of convergence.
- For more info, see e.g. `http://www.lorene.obspm.fr/palma.pdf`