Outline

1. Organization
2. Quadrilateral/Quadratic Basis Functions & Elements
3. Time integration
Outline

1. Organization
2. Quadrilateral/Quadratic Basis Functions & Elements
3. Time integration
Two-day Schedule

Part 1: Day 3: Wednesday

Chapter 8

- **Quadratic** basis functions in 1D & on triangles
- **Quadrilaterals** & reference coordinates
- **Curved** quadratic triangle
- Example: Stokes flow
- Exercises in the afternoon.

From:

J. van Kan, A. Segal, & F. Vermolen 2008
Numerical methods in Scientific Computing
VSSD
Two-day Schedule

Part 1: Day 3: Wednesday
Chapter 8

- Quadratic basis functions in 1D & on triangles
- Quadrilaterals & reference coordinates
- Curved quadratic triangle
- Example: Stokes flow
- Exercises in the afternoon.

From:

J. van Kan, A. Segal, & F. Vermolen 2008
Numerical methods in Scientific Computing
VSSD
Two-day Schedule

Part 2: Day 4: Thursday
Chapters 8, 10: 8.7, 10.1-10.5

- Overview 4th-order problems, Hermitian interpolation
- Ch. 10: Time integration heat equation: inequality for error
- Spatial discretization & consistency
- Time discretization & stability
- Worked example shallow water system: mesh of triangles/quadrilaterals
- Exercises in afternoon.

From:
J. van Kan, A.Segal, & F. Vermolen 2008
Numerical methods in Scientific Computing
VSSD
Two-day Schedule

Part 2: Day 4: Thursday
Chapters 8, 10: 8.7, 10.1-10.5

- Overview 4th-order problems, Hermitian interpolation
- **Ch. 10: Time integration** heat equation: inequality for error
- Spatial discretization & consistency
- Time discretization & stability
- **Worked example shallow water system**: mesh of triangles/quadrilaterals
- Exercises in afternoon.

From:

**J. van Kan, A. Segal, & F. Vermolen** 2008
**Numerical methods in Scientific Computing**
**VSSD**
Organizational details

Examination

- Theoretical exercises: three sets. Fred Vermolen.
- Numerical exercise for students from Twente and, optionally, others.
- Oral exam over the material for students from Twente.


Consultation: Onno Bokhove
Examination

- Theoretical exercises: three sets. Fred Vermolen.
- Numerical exercise for students from Twente and, optionally, others.
- Oral exam over the material for students from Twente.


Consultation: Onno Bokhove
Examination

- Theoretical exercises: three sets. Fred Vermolen.
- Numerical exercise for students from Twente and, optionally, others.
- Oral exam over the material for students from Twente.

**Web pages:** Fred Vermolen and Onno Bokhove: pdf-files, exercises, et cetera.

**Consultation:** Onno Bokhove
Outline

1. Organization

2. Quadrilateral/Quadratic Basis Functions & Elements

3. Time integration
Quadratic triangles

1D Lagrange interpolation; or, alternative geometric derivation:

- **1D reference element** $\zeta \in [-1, 1]$: 3 nodes, left, right, midpoint
- Each basis function alternately unity at -1, 0, 1, resp., and zero elsewhere
- **Basis functions:** $\zeta(\zeta - 1)/2, (1 - \zeta^2), \zeta(1 + \zeta)/2$ or $\lambda_i(2\lambda_i - 1)$ for $i = 1, 2$ and $4\lambda_1\lambda_2$
- **Disadvantage:** basis functions don’t sum up to unity & some are negative and positive
- **Solution:** Bernstein polynomials: $(1 + \zeta)^2/4, (1 - \zeta^2)/2, (1 - \zeta)^2/4$
Lagrange polynomials

Bernstein polynomials
Quadratic triangles

2D construction; or, alternative geometric derivation:

- **2D reference element** \( \zeta \in [0, 1], \eta \in [0, 1 - \zeta] \): 6 nodes, on vertices and mid points edges
- Each basis function unity at 6 nodes \((0, 0), (1, 0), (0, 1), (0.5, 0), (0.5, 0.5), (0, 0.5)\) and zero elsewhere
- **Basis functions**: \( \psi_i = \lambda_i (2\lambda_i - 1), \psi_{ij} = 4\lambda_i \lambda_j \) for \( i \neq j = 1, 2, 3 \)
- **Disadvantage**: basis functions don’t sum up to unity & some are negative as well as postive
- Evaluation integrals: Newton-Cotes, or Gaussian quadratures (online: . . . )
- **Solution**: Bernstein polynomials: . . . ?
Quadrilateral elements

2D construction:
- **Square reference element**, e.g., \((\zeta, \eta) \in [-1, 1]^2\);
  alternative \((\zeta, \eta) \in [0, 1]^2\)
- Continuous, ’linear’ functions on reference element:
  \[ \phi_i = \frac{(1 \pm \zeta)(1 \pm \eta)}{4}; \quad i = 1, 2, 3, 4 \]
- Includes polynomials \(1, \zeta, \eta\) and \(\zeta\eta\)
- Transform back to \((x, y)\)-space:
  \[ (x, y) = \sum_i (x_i, y_i) \phi_i(\zeta, \eta) \]
  for \(i = 1, 2, 3, 4\)
- **Isoparametric transformations**: straight sides mappings, nodes one-to-one
- Invertible if quadrilateral convex, better if angles < 135°
Quadratic Basis Functions & Elements

Lagrange polynomials on triangle
Quadrilateral elements

2D construction:

- Hence, $\phi_i(x, y)$ implicitly defined
- Transform integrals from $(x, y)$–space to reference space $(\zeta, \eta)$–space.
- Evaluate integrals in references coordinates: transformations, . . .
- Use Newton-Cotes, or Gaussian quadrature rules ( . . . online).
Curved quadratic triangles

- Simply use 6 quadratic basis functions in mapping:

\[(x, y) = \sum_{i=1}^{6} (x_i, y_i) \phi_i\]  \hspace{1cm} (1)

with \(\phi_1 = \phi_1, \phi_2 = \phi_2, \phi_3 = \phi_3, \phi_4 = \phi_{12}, \phi_5 = \phi_{23}, \phi_6 = \phi_{13}\)

and corresponding vertices and mid point nodes \((x_i, y_i)\)

- Angles triangle less than 135° (check Jacobian of mapping)

- Use quadrature for integrals on reference element: Newton-Cotes or Gaussian.
Example: Stokes flow

- Write problem in vector form
- Use Gauss’ rule and apply boundary conditions
- Determine conditions on test functions logically
- . . .
Fourth order problems and Hermitian interpolation

- Continuity requirements on derivatives as well: $C_0$ & $C_1$ (why?) or expand system by using auxiliary variables thus reducing the highest order derivatives
- $C_0$ & $C_1$: one exercise on Hermitian interpolation

$$w_h = \sum_{j=1}^{2} w_j \psi_{j0}(x) + \sum_{j=1}^{2} \frac{d w_j}{d x} \psi_{j1}(x)$$

(2)

with as unknown the values $w_j$ at the nodes and their derivatives $\left(\frac{d w}{d x}\right)_j$

- Requirements: $\psi_{j0}(x_i) = \delta_{ij}$, $\left(\frac{d \psi_{j0}}{d x}\right)(x_i) = 0$,
  $\psi_{j1}(x_i) = 0$, $\left(\frac{d \psi_{j0}}{d x}\right)(x_i) = \delta_{ij}$
Fourth order problems and Hermitian interpolation

- **Results** derived via calculation on reference element:
  \[
  \psi_{10} = \lambda_1^2(2\lambda_2 + 1), \quad \psi_{20} = \lambda_2^2(2\lambda_1 + 1), \\
  \psi_{11} = 2\lambda_1^2\lambda_2 h_K/2, \quad \psi_{21} = -2\lambda_1 \lambda_2^2 h_k/2 \text{ in element } K \text{ of size } h_K
  \]

- Details of use Hermitian interpolation given in reader of Jaap van der Vegt & Onno Bokhove (online)

- Graphs:
Outline

1. Organization
2. Quadrilateral/Quadratic Basis Functions & Elements
3. Time integration
Time integration

Ch. 10.1-10.5

- Ch. 10: Heat equation: inequality for error
- Spatial discretization & consistency
- Time discretization & stability
- Worked example shallow water system: mesh of triangles/quadrilaterals.
Inequality on Equilibration

- Consider IVP on $\Omega \in \mathbb{R}^2$ for $u = u(x, y, t)$:
  \[
  \partial_t u = \nabla^2 u + f(x, y, t)
  \]  
  (3)

- with initial condition $u_0 = u(x, y, 0)$ and boundary conditions
  \[
  u(x, y, t) = g_1(x, y, t) \quad \text{on } \Gamma_1
  \]  
  \[
  \partial_n u = g_2(x, y, t) \quad \text{on } \Gamma_2
  \]  
  \[
  (\sigma u) + \partial_n u = g_3(x, y, t) \quad \text{on } \Gamma_3,
  \]  
  (4) (5) (6)

- and steady state or equilibrium solution $u_E(x, y)$;
- then . . .
Inequality on Equilibration

then

$$R(t) < R(t_0)e^{-\gamma(t-t_0)} \quad \text{for } t > t_0, \gamma > 0$$  \hspace{1cm} (7)

with deviation from equilibrium $$R(t) = \int_\Omega (u - u_E)^2 d\Omega$$

The FEM discretization of (3) gives system of ODE’s

$$M\frac{du}{dt} = Su + f$$  \hspace{1cm} (8)

with vector of unknowns $$u$$, projected forcing vector $$f$$, mass matrix $$M$$, and Laplace matrix $$S$$.

There is a discrete version of the inequality on equilibration —exercise 10.3.3.
\( \theta \)-method for time discretization of ODE’s of FEM discretisation reads:

\[
(M - \theta \Delta t S)u^{n+1} = (M + (1 - \theta)\Delta t S)u^n + (1 - \theta)f^n + \theta f^{n+1} \tag{9}
\]

- Explicit forward Euler: \( \theta = 0 \)
- Implicit backward Euler: \( \theta = 1 \)
- Crank-Nicholson: \( \theta = 1/2 \)

- \( M \) diagonal when Newton-Cotes integration used. Why?
  Exercise 10.4.1.
Time integration

- 'Error equation' for error $\epsilon = y - u$ (exact $y$ reads $d\epsilon/dt = A\epsilon$) as follows from system $M \frac{du}{dt} = Su + f$  \(10\)

- System absolutely stable only if real part of eigenvalues of $A = M^{-1}S$ negative: $Re(\lambda_k) < 0$

- Time discretization defines amplification matrix in:

  $\epsilon^{n+1} = G(\Delta tA)\epsilon^n$ \(11\)

- After we diagonalize $A$ introducing eigenvalue $\lambda$, scalar case becomes relevant

  $\epsilon^{n+1} = C(\Delta t\lambda)\epsilon^n$ from $\frac{d\epsilon}{dt} = \lambda\epsilon$ \(12\)

- Numerical stability achieved when eigenvalue $\mu_k$ of $G(\Delta tA)$ satisfies:

  $|\mu_k| = |C(\Delta t\lambda_k)| < 1$ \(13\)
Time integration

Gershgorin’s circle theorem

- Eigenvalues $\lambda$ of system also follow from generalized eigenvalue problem

$$Sx = \lambda Mx \quad (14)$$

- For diagonal $M$, Gershgorin theorem is:

$$|\lambda| < \sup_k \frac{1}{|m_{kk}|} \sum_{i=1}^{N} |s_{ki}| \quad (15)$$

- In 1D FEM with lumping $m_{ii} = (h_{i-1} + h_i)/2$, this gives

$$|\lambda_{max}| = \sup_i \frac{2}{h_{i-1} + h_i} \left( \frac{2}{h_{i-1}} + \frac{2}{h_i} \right) < \sup_i \frac{4}{h_{i-1} h_i} \quad (16)$$
Time integration

Stability

In FEM with lumping $m_{ii} = (h_{i-1} + h_i)/2$, Gershgorin’s theorem gives

$$|\lambda_{max}| < \sup_i \frac{4}{h_{i-1}h_i}$$

(17)

Hence, the stability criterion becomes:

$$\Delta t < \frac{c}{\lambda_{max}} = \frac{1}{2} \inf_i (h_{i-1}h_i)$$

(18)

with $c = 2$ for forward Euler.

Stability of forward Euler follows from Von Neumann analysis on $\epsilon^{n+1} = (1 + \Delta t \lambda)\epsilon^n$

Von Neumann stability analysis: substitute $\epsilon = q$ for complex $q$, and require amplification $|q| < 1$; recall that $\lambda$ is negative . . . .
**Worked example: potential flow linear shallow water equations**

Linear shallow water equations:

\[
\begin{align*}
\partial_t \eta + \nabla \cdot (H(x, y) \nabla \phi) &= 0 \\
\partial_t \phi + g \eta &= 0,
\end{align*}
\]

where:

- \( H = H(x, y) \) is given rest depth
- \( \eta = \eta(x, y, t) \) deviate of free surface (the wave)
- \( \phi = \phi(x, y, t) \) the velocity potential
- \( \nabla = (\partial_x, \partial_y) \) is the horizontal gradient
- \( g = 9.8 \text{m/s}^2 \) is constant of acceleration of gravity.
Variational principle/Ritz method:

\[ 0 = \delta \int_0^T \int_0^{L_x} \int_0^{L_y} \phi \partial_t \eta - \frac{1}{2} H |\nabla \phi|^2 - \frac{1}{2} g \eta^2 \, dx \, dy \, dt \] (21)

Initial and boundary conditions:

- At \( t = 0 \): \( \eta(x, 0) = \eta_0(x), \phi(x, 0) = \phi_0(x) \); also \( \delta \eta|_0^T = 0 \)
- Periodic boundaries or solid walls with Neumann condition such that \( \nabla \phi \cdot \hat{n} = 0 \) with normal \( \hat{n} \) to the wall.
**Worked example: goal**

- **Goal:** Find a FEM discretization (weak formulation or via variational principle/Ritz’ method) in 2D on quadrilaterals and triangles, using Crank-Nicolson time discretization, or symplectic Euler forward scheme.

- **Symplectic Euler forward:** first update $\phi$ and then $\eta$ while using the new update of $\phi^{n+1}$. A 2nd-order version is Stormer-Verlet where $\phi$ is updated $\Delta t/2$, then $\eta$ for $\Delta t$, and $\phi$ from intermediate steps $\phi^{n+1/2}$ and $\eta^{n+1}$ (google online).
Worked example: plan for discretization

Steps involved:

1. Find weak formulation using global test functions and integration, or substitute expansion into variational principle/Ritz method
2. Define global matrices and derive ordinary differential equations for unknowns $\eta_j$ and $\phi_j$ on nodes: ...
3. Define (topologically regular) mesh: lots of indexing
4. Define mapping from global to local node numbers:
5. Define elemental matrices: ...
6. Assemble global matrices from local elemental matrices; pseudo code: ...
7. Define time discretization: ...
8. Solve matrix system to find unknowns next time step: ...
9. Plot/export output when required
10. Finish program at $t = T_{end}$