Mimetic Methods for Diffusion Problems

DANSIS seminar: High order discretisation 21/3/2018

Kennet Olesen
PhD
Aarhus University 2013-2016
A diffusion problem is typically constructed out of 2 parts

- An equilibrium or balance part
  - E.g. mass balance, energy balance etc.

\[ \nabla \cdot \mathbf{q} = f \]

Structure of a diffusion problem

\[ Q_1,+ - Q_1,- + Q_2,+ - Q_2,- + Q_3,+ - Q_3,- = F \]

This is what we want to mimic in our discretization!
Special interpolation polynomials: Edge polynomials

In [1] interpolation polynomials are derived, which are based on integrated line values

$$
\beta^h(\xi) = \sum_{i=1}^{N} B_i e_i(\xi) \quad B_i = \int_{\xi_{i-1}}^{\xi_i} \beta(\xi) d\xi
$$

$$
e_i(\xi) = -\sum_{j=0}^{i-1} \frac{dh_j(\xi)}{d\xi} \quad \int_{\xi_{k-1}}^{\xi_k} e_i(\xi) = \begin{cases} 1 & \text{for } i = k \\ 0 & \text{for } i \neq k \end{cases}
$$

$h$: Lagrange polynomials

Special interpolation polynomials: Edge polynomials

\[ \alpha^h(\xi) = \sum_{i=0}^{N} \alpha_i h_i(\xi) \]

\[ \beta^h(\xi) = \sum_{i=1}^{N} B_i e_i(\xi) \]

\[ B_i = \int_{\xi_{i-1}}^{\xi_i} \beta(\xi) d\xi \]
Interpolation in 3D -> tensor products

\[
\Psi_{ijk}(\xi_1, \xi_2, \xi_3) = d_i(\xi_1) d_j(\xi_2) d_k(\xi_3)
\]

\[
\Psi_{ijk}(\xi_1, \xi_2, \xi_3) = e_i(\xi_1) e_j(\xi_2) e_k(\xi_3)
\]
Differentiation

• Say we have the approximation:

\[ \alpha^h(\xi) = \sum_{i=0}^{N} \alpha_i h_i(\xi) \]

• We choose to approximate the derivative by:

\[
\frac{d\alpha^h(\xi)}{d\xi} = \sum_{i=1}^{N} \left( \int_{i-i}^{i} \frac{d\alpha^h(\xi)}{d\xi} \, d\xi \right) e_i(\xi)
\]

• This reduces to (first fundamental theorem of calculus):

\[
\frac{d\alpha^h(\xi)}{d\xi} = \sum_{i=1}^{N} (\alpha_i - \alpha_{i-1}) \, e_i(\xi)
\]
Divergence

\[ \nabla \cdot q^h = \frac{\partial q_1^h(\xi_1, \xi_2, \xi_3)}{\partial \xi_1} + \frac{\partial q_2^h(\xi_1, \xi_2, \xi_3)}{\partial \xi_2} + \frac{\partial q_3^h(\xi_1, \xi_2, \xi_3)}{\partial \xi_3} \]

\[ q_1^h(\xi_1, \xi_2, \xi_3) = \sum_{i=0}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} Q_{i,j,k}^1 h_i(\xi_1) e_j(\xi_2) e_k(\xi_3) \]

\[ q_2^h(\xi_1, \xi_2, \xi_3) = \sum_{i=1}^{N} \sum_{j=0}^{N} \sum_{k=1}^{N} Q_{i,j,k}^2 e_i(\xi_1) h_j(\xi_2) e_k(\xi_3) \]

\[ q_3^h(\xi_1, \xi_2, \xi_3) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=0}^{N} Q_{i,j,k}^3 e_i(\xi_1) e_j(\xi_2) h_k(\xi_3) \]

\[ \nabla \cdot q^h = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} (Q_{i,j,k}^1 - Q_{i-1,j,k}^1 + Q_{i,j,k}^2 - Q_{i,j-1,k}^2 + Q_{i,j,k}^3 - Q_{i,j,k-1}^3) e_i(\xi_1) e_j(\xi_2) e_k(\xi_3) \]
Divergence – Incidence matrix

\[ E_{(3,2)} = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \]
The discrete balance equation

\[ \nabla \cdot q^h = f^h \]

This is written by:

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \left( Q_{i,j,k}^1 - Q_{i-1,j,k}^1 + Q_{i,j,k}^2 - Q_{i,j-1,k}^2 + Q_{i,j,k}^3 - Q_{i,j,k-1}^3 \right) e_i(\xi_1) e_j(\xi_2) e_k(\xi_3) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} F_{i,j,k} e_i(\xi_1) e_j(\xi_2) e_k(\xi_3)
\]

They share common interpolation polynomials:

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \left( Q_{i,j,k}^1 - Q_{i-1,j,k}^1 + Q_{i,j,k}^2 - Q_{i,j-1,k}^2 + Q_{i,j,k}^3 - Q_{i,j,k-1}^3 - F_{i,j,k} \right) e_i(\xi_1) e_j(\xi_2) e_k(\xi_3) = 0
\]

This implies:

\[
Q_{i,j,k}^1 - Q_{i-1,j,k}^1 + Q_{i,j,k}^2 - Q_{i,j-1,k}^2 + Q_{i,j,k}^3 - Q_{i,j,k-1}^3 - F_{i,j,k} = 0
\]

Exact balance equation for our mesh
Putting together the system of equations

- Problem: More unknowns than equations

- I fix this when I discretize the constitutive equation
  \[ q^h = k \nabla \alpha^h \]

- Weigh with \( q^h \) and integrate over domain
  \[
  \frac{1}{k} (q^h, q^h)_\Omega = (q^h, \nabla \alpha^h)_\Omega = -(\nabla \cdot q^h, \alpha^h)_\Omega + (q^h, \alpha^h)_{\partial \Omega} \tag{Integration by parts}
  \]

- Choose same amount of discrete points for \( \alpha^h \) as number of elements

- Square system of equations
Putting together the system of equations

- Balance equation + constitutive equation → Square system of equations

\[
\begin{bmatrix}
0 & E_{(3,2)} \\
(E_{(3,2)}V^h)^T & \frac{1}{k} H^h
\end{bmatrix}
\begin{bmatrix}
\Delta_{\alpha} \\
\Delta_{Q}
\end{bmatrix} = \begin{bmatrix}
\Delta_F \\
B^h \Delta_{\alpha}
\end{bmatrix}
\]

Flux boundary conditions strongly enforced
Potential boundary conditions weakly enforced

- Multiply balance equation with \(V^h\) → Symmetric system of equations

\[
\begin{bmatrix}
0 & V^h E_{(3,2)} \\
(E_{(3,2)}V^h)^T & \frac{1}{k} H^h
\end{bmatrix}
\begin{bmatrix}
\Delta_{\alpha} \\
\Delta_{Q}
\end{bmatrix} = \begin{bmatrix}
V^h \Delta_F \\
B^h \Delta_{\alpha}
\end{bmatrix}
\]
Complex geometry

- Mapping will only affect the matrices with expansion polynomials and not the incidence matrices.

- I.e. mapping will not affect the mimicking property of the balance equation.

\[ E_{(3,2)} \Delta Q = \Delta_F \]
Test case

• Choose potential field
• Calculate $q$ and $f$
• Apply as boundary conditions

• Solve on different meshes
• Refine mesh
  • Mesh size $h_{el}$ ↓
• Evaluate in $100 \times 100$ points in each element
• Plot the maximum value in the entire domain

Error of the Potential ($\alpha = \tilde{\phi}$)

Residual of balance equation ($R = \nabla \cdot q^h - f^h$)
Mimicking vector valued fields

- So far we have considered Poisson’s equation of a scalar field.
- But what about Poisson’s equation of a vector field?
  - For example the equilibrium of forces in Continuum mechanics.

\[ \nabla \cdot \sigma = -f \]
Mimicking vector valued fields

• The procedure is the same just in 3 directions
  \[ D\Delta_T = -\Delta_F \]

• With
  \[ D = \begin{bmatrix} E_{(3,2)} & 0 & 0 \\ 0 & E_{(3,2)} & 0 \\ 0 & 0 & E_{(3,2)} \end{bmatrix} \]

• Consider the structural problem
  • Constitutive equations
    \[ \varepsilon_{ij} = \frac{\partial}{\partial x_i} u_j - \omega_{ij} = C_{ijkl} \sigma_{kl} \]
  • Equilibrium of forces
    \[ \frac{\partial}{\partial x_i} \sigma_{ij} + f_j = 0 \]
  • Symmetry of the stress tensor
    \[ \sigma_{ij} - \sigma_{ji} = 0 \]
Test case

\[ \sigma_{21} = \sigma_{21}^{ex} \]
\[ \sigma_{22} = \sigma_{22}^{ex} \]

\[ \sigma_{11} = \sigma_{11}^{ex} \]
\[ \sigma_{12} = \sigma_{12}^{ex} \]

\[ \sigma_{12} \cdot n_i = 0 \]

\[ u_1 = 0 \]
\[ \sigma_{12} = 0 \]

\[ u_2 = 0 \]
\[ \sigma_{21} = 0 \]

\[ r = 0.5 \]
Test case

\[ \begin{align*}
\sigma_{22} &= \sigma_{12} = 0 \\
\sigma_{11} &= 1 \\
\sigma_{ij} &= 0 \\
v_1 = v_2 &= 0 \quad \uparrow 0.1
\end{align*} \]

Mimetic Methods for Diffusion Problems
Thank you for your time!