Integral Equations and Fast Algorithms
Lecture 20: High-order numerics, IE discretizations

CS 598AK · October 31, 2013

Outline

Intro to Numerics

IE discretization: Nyström

Generalized Vandermonde matrices

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = ?$$

Generalized Vandermonde matrices

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} \text{MODAL COEFFS} \end{pmatrix} = \begin{pmatrix} \text{NODAL COEFFS} \end{pmatrix}$$

Instantly generalizes to multiple dimensions

Generalized Vandermonde matrices

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} \text{MODAL COEFFS} \end{pmatrix} = \begin{pmatrix} \text{NODAL COEFFS} \end{pmatrix}$$

Instantly generalizes to multiple dimensions

Invertible?

Generalized Vandermonde matrices

$$
\begin{pmatrix}
\varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\
\varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\
\vdots & \vdots & \ddots & \vdots \\
\varphi_0(x_n) & \varphi_1(\ldots & &
\end{pmatrix}
\begin{pmatrix} \text{DAL COEFFS} \end{pmatrix}
=
\begin{pmatrix} \text{DAL COEFFS} \end{pmatrix}
$$

Instantly generalizes to multiple dimensions

Invertible?

Conditioning?

Interpolation node placement demo

Vandermonde conditioning demo

2D basis demo

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation *(Hint: solve using V^T)*

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation *(Hint: solve using V^T)*
- Differentiation

Common operations

(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation *(Hint: solve using V^T)*
- Differentiation
- Indefinite Integration

Common operations

(Generalized) Vandermonde matrices simplify common operations:

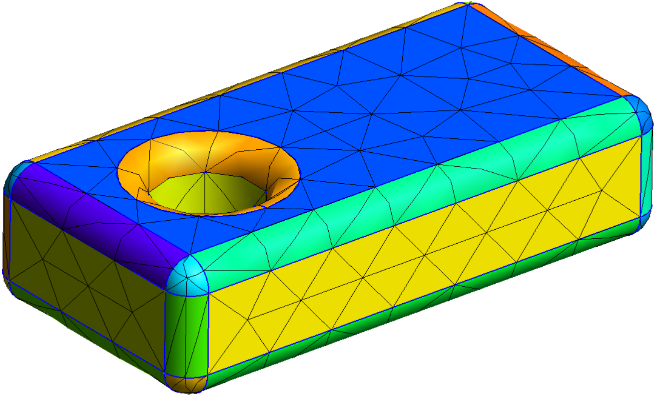
- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation *(Hint: solve using V^T)*
- Differentiation
- Indefinite Integration
- Inner product

Common operations

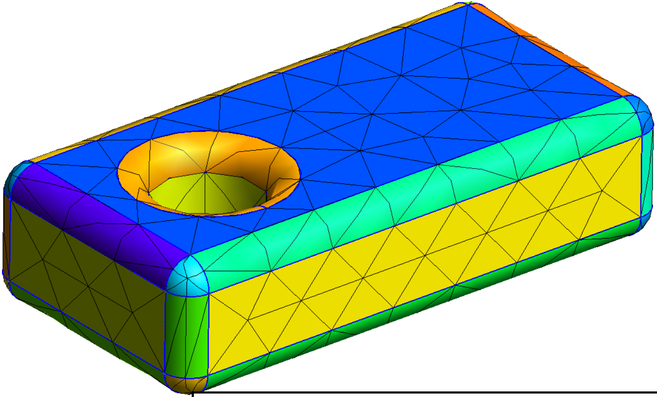
(Generalized) Vandermonde matrices simplify common operations:

- Modal $\leftrightarrow$ Nodal ("Global interpolation")
  - Filtering
  - Up-/Oversampling
- Point interpolation *(Hint: solve using V^T)*
- Differentiation
- Indefinite Integration
- Inner product
- Definite integration

Unstructured mesh

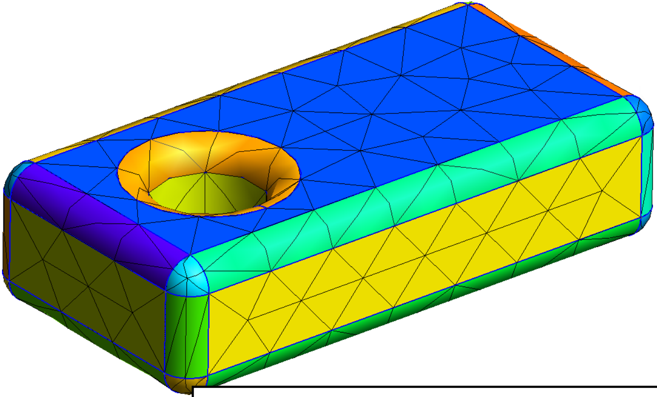


Unstructured mesh



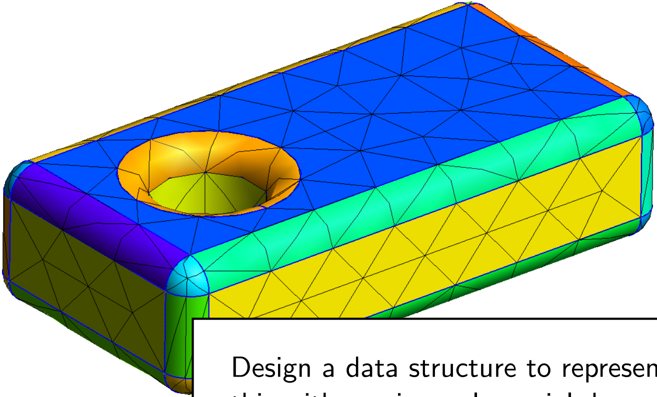
Design a data structure to represent this

Unstructured mesh



Design a data structure to represent this with varying polynomial degrees

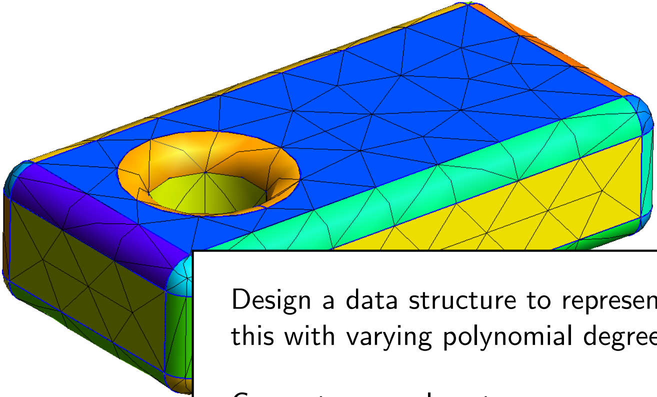
Unstructured mesh



Design a data structure to represent this with varying polynomial degrees

Compute normal vectors

Unstructured mesh

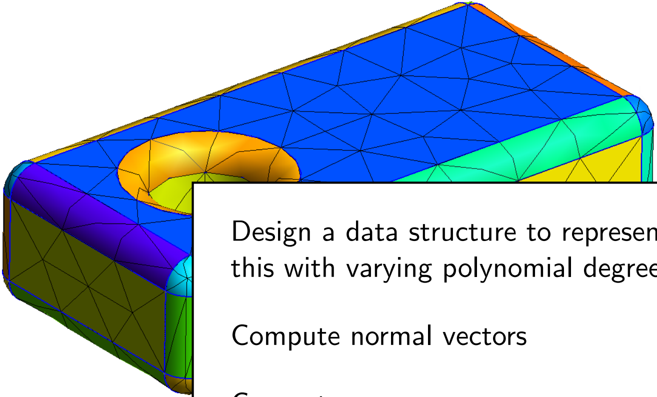


Design a data structure to represent this with varying polynomial degrees

Compute normal vectors

Compute area

Unstructured mesh



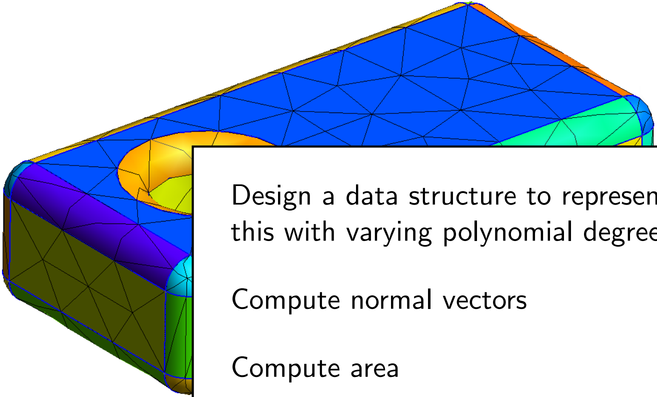
Design a data structure to represent this with varying polynomial degrees

Compute normal vectors

Compute area

Compute integral of a function

Unstructured mesh



Design a data structure to represent this with varying polynomial degrees

Compute normal vectors

Compute area

Compute integral of a function
How is the function represented?

Outline

Intro to Numerics

IE discretization: Nyström

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integral by quadrature:
  $\int_\Gamma f(y)dy \to \sum_{k=1}^n \omega_k f(y_k)$
- Evaluate quadrature'd IE at quadrature nodes, solve

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integral by quadrature:
  $\int_\Gamma f(y)dy \to \sum_{k=1}^n \omega_k f(y_k)$
- Evaluate quadrature'd IE at quadrature nodes, solve

Projection

- Consider residual: $R := \varphi - A\varphi - f$
- Pick projection P_n onto finite-dimensional subspace
  $P_n\varphi := \sum_{k=1}^n \langle \varphi, v_k\rangle w_k \quad \to \quad$ DOFs $\langle \varphi, v_k\rangle$
- Solve $P_n R = 0$

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integral by quadrature:
  $\int_\Gamma f(y)dy \to \sum_{k=1}^{n} \omega_k f(y_k)$
- Evaluate quadrature'd IE at quadrature nodes, solve

Projection

- Consider residual: $R := \varphi - A\varphi - f$
- Pick projection P
  $P_n\varphi := \sum_{k=1}^{n}\langle\varphi,$
- Solve $P_n R = 0$

Important in projection methods:
*sub*space (e.g. of $C(\Gamma)$)

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integral by quadrature: $\int_\Gamma f(y)dy \to \sum_{k=1}^n \omega_k f(y_k)$
- Evaluate quadrature'd IE at quadrature nodes, solve

Projection

- Consider residual:
- Pick projection P
  $P_n\varphi := \sum_{k=1}^n \langle\varphi,$
- Solve $P_n R = 0$

Important in projection methods: *sub*space (e.g. of $C(\Gamma)$)

Possible bases for projection?

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integral by quadrature:
  $\int_\Gamma f(y)dy \to \sum_{k=}^{n}$
- Evaluate quadratu

Projection

- Consider residual:
- Pick projection P
  $P_n\varphi := \sum_{k=1}^{n}\langle\varphi,$
- Solve $P_n R = 0$

Important in projection methods: *sub*space (e.g. of $C(\Gamma)$)

Possible bases for projection?

- Galerkin: $v_k = w_k$
- Collocation:
  $v_k = \delta(x_k)$, $w_k(x_j) = \delta_{jk}$
- Petrov-Galerkin: $v_k \neq w_k$

IE discretization: Overview

$$\varphi(x) - \int_\Gamma K(x,y)\varphi(y)dy = f(y)$$

Nyström

- Approximate integ
  $\int_\Gamma f(y)dy \to \sum_{k=}^{n}$
- Evaluate quadratu

Projection

- Consider residual:
- Pick projection P
  $P_n\varphi := \sum_{k=1}^{n}\langle\varphi,$
- Solve $P_n R = 0$

Important in projection methods:
*sub*space (e.g. of $C(\Gamma)$)

Possible bases for projection?

- Galerkin: $v_k = w_k$
- Collocation:
  $v_k = \delta(x_k)$, $w_k(x_j) = \delta_{jk}$
- Petrov-Galerkin: $v_k \neq w_k$

Equivalent to projection:
Test IE with test functions

IE discretization: Overview

$$\varphi(x) - \int K(x,y)\varphi(y)dy = f(x)$$

Nyström

- Approximate integ…
  $\int_\Gamma f(y)dy \to \sum_{k=…}^{n}$ …
- Evaluate quadratu…

Projection

- Consider residual: …
- Pick projection P…
  $P_n\varphi := \sum_{k=1}^{n}\langle\varphi,$ …
- Solve $P_n R = 0$

Important in projection methods:
*sub*space (e.g. of $C(\Gamma)$)

Possible bases for projection?

- Galerkin: $v_k = w_k$
- Collocation:
  $v_k = \delta(x_k)$, $w_k(x_j) = \delta_{jk}$
- Petrov-Galerkin: $v_k \neq w_k$

Equivalent to projection:
Test IE with test functions

Collocation and Nyström: the same?

Questions?

?