Numerical Methods for PDEs

Integral Equation Methods, Lecture 2
Numerical Quadrature

Notes by Suvranu De and J. White

April 28, 2003
Outline

Easy technique for computing integrals
  Piecewise constant approach

Gaussian Quadrature
  Convergence properties
  Essential role of orthogonal polynomials
  Multidimensional Integrals

Techniques for singular kernels
  Adaptation and variable transformation
  Singular quadrature.
3D Laplace’s Equation

Basis Function Approach

Centroid Collocation

Put collocation points at panel centroids

\[ \Psi(x_{c,i}) = \sum_{j=1}^{n} \alpha_j \int_{\text{panel } j} \frac{1}{A_{i,j} \| x_{c,i} - x' \|} dS' \]

Collocation point \( x_{c,i} \)

\[
\begin{bmatrix}
A_{i,1} & \cdots & A_{i,n} \\
\vdots & \ddots & \vdots \\
A_{n,1} & \cdots & A_{n,n}
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\vdots \\
\alpha_n
\end{bmatrix}
= 
\begin{bmatrix}
\Psi(x_{c_1}) \\
\vdots \\
\Psi(x_{c_n})
\end{bmatrix}
\]
3D Laplace’s Equation

Basis Function Approach

Calculating Matrix Elements

\[ A_{i,j} = \int_{\text{panel } j} \frac{1}{\left\| x_{c_i} - x' \right\|} dS' \]

One point quadrature approximation

\[ A_{i,j} \approx \frac{\text{Panel Area}}{\left\| x_{c_i} - x_{\text{centroid}_j} \right\|} \]

Four point quadrature approximation

\[ A_{i,j} \approx \sum_{j=1}^{4} \frac{0.25 \cdot \text{Area}}{\left\| x_{c_i} - x_{\text{point}_j} \right\|} \]
\[ \Psi(x) = \int_0^1 g(x, x') \sigma(x') \, dS' \quad x \in [0, 1] \]

**Centroid collocated piecewise constant scheme**

\[ \Psi(x_{c_i}) = \sum_{j=1}^{n} \sigma_j \int_{x_{j-1}}^{x_j} g(x_{c_i}, x') \, dS' \]

*to be evaluated*
Normalized 1D Problem

\[ \int_{0}^{1} f(x) \, dx \approx f \left( \frac{1}{2} \right) \]

Area under the curve is approximated by a rectangle.
\[ \int_0^1 f(x) \, dx \approx \frac{1}{2} f \left( \frac{1}{4} \right) + \frac{1}{2} f \left( \frac{3}{4} \right) \]

Area under the curve is approximated by two rectangles.
Normalized 1D Problem

Simple Quadrature Scheme

General n-Point Formula

\[ \int_0^1 f(x) \, dx \approx \sum_{i=1}^n \frac{1}{n} f \left( \frac{i - \frac{1}{2}}{n} \right) \]

Key questions about the method:
How fast do the errors decay with \( n \)?
Are there better methods?
Normalized 1D Problem

Simple Quadrature Scheme

Numerical Example

\[ \int_0^1 \sin(x) \, dx \approx \sum_{i=1}^{n} \frac{1}{n} \sin \left( \frac{i - \frac{1}{2}}{n} \right) \]

![Graph showing error vs. n]
Normalized 1D Problem

General Quadrature Scheme

General 1D Form

\[ \int_0^1 f(x) \, dx \simeq \sum_{i=1}^{n} w_i \cdot f(x_i) \]

Free to pick the **evaluation points**.
Free to pick the **weights** for each point.

An n-point formula has 2n degrees of freedom!
Result should be exact if $f(x)$ is a polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_l x^l = p_l(x)$$

Select $x_i$’s and $w_i$’s such that

$$\int_0^1 p_l(x) \, dx = \sum_{i=1}^n w_i p_l(x_i)$$

for ANY polynomial upto (and including) $l^{th}$ order

With $2n$ degrees of freedom, $l = 2n - 1$
Consider the Taylor series for $f(x)$

$$f(x) = f(0) + \frac{\partial f(0)}{\partial x}x + \cdots + \frac{1}{l!} \frac{\partial^l f(0)}{\partial x^l} x^l + R_{l+1}$$

$R_{l+1}$ is the remainder

$$R_{l+1} = \frac{1}{(l + 1)!} \frac{\partial^{l+1} f(\tilde{x})}{\partial x^{l+1}} x^{l+1}$$

where $\tilde{x} \in [0, x]$
Using the Taylor series results and the exactness criteria

\[
\int_0^1 f(x)dx - \sum_{i=1}^n w_if(x_i) = \frac{1}{(l+1)!} \int_0^1 \frac{\partial^{l+1} f(\tilde{x}(x))}{\partial x^{l+1}} x^{l+1} dx
\]

Assuming derivatives of \( f(x) \) are bounded on \([0, 1]\)

\[
\left| \int_0^1 f(x)dx - \sum_{i=1}^n w_if(x_i) \right| \leq \frac{K}{(l+1)!}
\]

Convergence is very fast!!
Exactness condition requires

\[ \int_0^1 p_l(x) \, dx = \int_0^1 (a_0 + a_1 x + a_2 x^2 + \cdots + a_l x^l) \, dx = \sum_{i=1}^n w_i p_l(x_i) \]

for any set of \( l + 1 \) coefficients \( a_0, a_1, \ldots, a_l \)

Equivalently

\[ \int_0^1 a_0 \, dx + \int_0^1 a_1 x \, dx + \int_0^1 a_2 x^2 \, dx + \cdots + \int_0^1 a_l x^l \, dx = \sum_{i=1}^n w_i p_l(x_i) \]
Exactness condition will be satisfied if and only if

\[ \int_0^1 dx = \sum_{i=1}^{n} w_i \cdot 1 \]

\[ \int_0^1 x \, dx = \sum_{i=1}^{n} w_i \cdot x_i \]

\[ \int_0^1 x^l \, dx = \sum_{i=1}^{n} w_i \cdot x_i^l \]
Reorganizing exactness equations

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_n \\
\vdots & \vdots & \ddots & \vdots \\
x_1^l & x_2^l & \cdots & x_n^l
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
- \begin{bmatrix}
1 \\
\vdots \\
1 \\
\int_0^1 x^l \, dx
\end{bmatrix} = 0
\]

Nonlinear, since \(x_i\)'s and \(w_i\)'s are unknowns
Normalized 1D Problem

General Quadrature Scheme

Computing the Points and Weights

Could use **Newton’s Method**

\[ F(y) = 0 \Rightarrow J_F(y^k)(y^{k+1} - y^k) = -F(y^k) \]

The nonlinear function for Newton is then

\[
F(w, x) = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_n \\
\vdots & \vdots & \ddots & \vdots \\
x_1' & x_2' & \cdots & x_n'
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
- \begin{bmatrix}
1 \\
\vdots \\
\vdots \\
1 \\
\int_0^1 x^l dx
\end{bmatrix} = 0
\]
Normalized 1D Problem

General Quadrature Scheme

Computing the Points and Weights

Newton Method Jacobian reveals problem

$$J_F \left( \begin{pmatrix} w \\ x \end{pmatrix} \right) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ x_1 & x_2 & \cdots & x_n & w_1 & w_2 & \cdots & w_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^l & x_2^l & \cdots & x_n^l & lw_1x_1^{l-1} & \cdots & \cdots & lw_1x_n^{l-1} \end{bmatrix}$$

Columns become linearly dependent for high order
Exactness criteria will be satisfied if and only if

\[
\begin{align*}
\int_{0}^{1} c_0(x) \, dx &= \sum_{i=1}^{n} w_i c_0(x_i) \\
\int_{0}^{1} c_1(x) \, dx &= \sum_{i=1}^{n} w_i c_1(x_i) \\
\vdots & \quad \vdots \\
\int_{0}^{1} c_i(x) \, dx &= \sum_{i=1}^{n} w_i c_i(x_i)
\end{align*}
\]

BUT

Each \( c_i \) polynomial must contain an \( x^i \) term and be linearly independent.
For the normalized integral, two polynomials are said to be **orthogonal** if

$$\int_0^1 c_i(x)c_j(x)dx = 0 \quad \text{for } j \neq i$$

The above integral is often referred to as an inner product and ascribed the notation

$$(c_i, c_j) = \int_0^1 c_i(x)c_j(x)dx$$

The connection between polynomial inner products and vector inner products can be seen by sampling.
Consider rewriting the exactness criteria

\[
\int_{0}^{1} c_0(x) \, dx = \sum_{i=1}^{n} w_i c_0(x_i) \\
\int_{0}^{1} c_{n-1}(x) \, dx = \sum_{i=1}^{n} w_i c_{n-1}(x_i)
\]

Low order terms

\[
\int_{0}^{1} c_n(x) \, dx = \sum_{i=1}^{n} w_i c_n(x_i) \\
\int_{0}^{1} c_{2n-1}(x) \, dx = \sum_{i=1}^{n} w_i c_{2n-1}(x_i)
\]

High Order Terms

Recall that \( l(\# \text{ polys}) = 2n - 1(\# \text{ of coefficients}) \)
Can write the higher order terms differently
\[ \int_0^1 c_n(x) dx = \sum_{i=1}^{n} w_i c_n(x_i) \Rightarrow \int_0^1 c_n(x) c_0(x) dx = \sum_{i=1}^{n} w_i c_n(x_i) c_0(x_i) \]
\[ \vdots \]
\[ \int_0^1 c_{2n-1}(x) dx = \sum_{i=1}^{n} w_i c_{2n-1}(x_i) \Rightarrow \int_0^1 c_n(x) c_{n-1}(x) dx = \sum_{i=1}^{n} w_i c_n(x_i) c_{n-1}(x_i) \]

The products \( c_n(x) c_j(x) \) are linearly independent!
Normalized 1D Problem

General Quadrature Scheme

Using Orthogonality and Roots

Use orthogonal polynomials

\[ \int_{0}^{1} c_n(x) c_0(x) \, dx = \sum_{i=1}^{n} w_i c_n(x_i) c_0(x_i) \]

\[ \int_{0}^{1} c_n(x) c_{n-1}(x) \, dx = \sum_{i=1}^{n} w_i c_n(x_i) c_{n-1}(x_i) \]

Pick the \( x_i \)'s to be \( n \) roots of \( c_n(x) \)

The higher order constraints are exactly satisfied!
An abbreviated exactness equation

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
c_0(x_1) & \cdots & \cdots & c_0(x_n) \\
\vdots & \vdots & \ddots & \vdots \\
c_{n-1}(x_1) & \cdots & \cdots & c_{n-1}(x_n)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
\vdots \\
\vdots \\
\int_0^1 c_{n-1}(x) \, dx
\end{bmatrix}
\]

Now linear, as \( x_i \)'s are known!
Rows are sampled orthogonal polynomials!
1. Construct \( n + 1 \) orthogonal polynomials
\[
\int_0^1 c_i(x)c_j(x)dx = 0 \quad \text{for } j \neq i
\]

2. Compute \( n \) roots, \( x_i, i = 1, \ldots, n \) of the \( n^{th} \) order orthogonal polynomial such that \( c_n(x_i) = 0 \)

3. Solve a linear system for the weights \( w_i \)

4. Approximate the integral as a sum
\[
\int_0^1 f(x)dx = \sum_{i=1}^n w_i f(x_i)
\]
Normalized 1D Problem

\[ \int_0^1 f(x) \, dx \simeq \sum_{i=1}^{n} w_i f(x_i) \]

Key properties of the method

- An \( n \)-point Gauss quadrature rule is exact for polynomials of order \( 2n - 1 \)
- Error is proportional to \( \left( \frac{1}{2n} \right)^{2n} \)
Normalized 1D Problem

Simple Quadrature Scheme

General n-Point Formula

\[ \int_0^1 f(x) \, dx \approx \sum_{i=1}^{n} \frac{1}{n} f \left( \frac{i - \frac{1}{2}}{n} \right) \]

Key property of the method

- Error is proportional to \( \frac{1}{n^2} \)
Normalized 1D Problem

Comparing Simple Quad and Gauss Quad

![Graph comparing Simple Quad and Gauss-Quad methods with error and number of points on logarithmic scales. The graph shows a superior performance of Gauss-Quad in terms of error for a given number of points.]

\[ \int_{-1}^{1} \cos(2\pi x) \, dx \]
Normalized 1D Problem

Comparing Simple Quad and Gauss Quad

Evaluation Point Placement

Simple-Quadrature Points

-1  0  1

Gauss-Quadrature Points

-1  0  1

Notice the clustering at interval ends

SMA-HPC ©2002 MIT  Numerical Quadrature  27
The Singular Kernel Problem

3D Laplace Example

Calculating the "Self-Term"

\[ A_{i,i} = \int_{\text{panel } i} \frac{1}{\|x_{c_i} - x'\|} \, dS' \]

One point quadrature approximation

\[ A_{i,i} \approx \frac{\text{Panel Area}}{\|x_{c_i} - x_{c_i}\|} \]

\[ A_{i,i} = \int_{\text{panel } i} \frac{1}{\|x_{c_i} - x'\|} \, dS' \text{ is an integrable singularity} \]
The Singular Kernel Problem

Symmetrized 1D Example

\[ \int_{-1}^{1} \frac{1}{\sqrt{|x|}} \, dx \approx \sum_{i=1}^{n} w_i \frac{1}{\sqrt{|x_i|}} \]

Quad Point
Note no \( x_i = 0 \)
The Singular Kernel Problem

Symmetrized 1D Example

Integrable and Nonintegrable Singularities

\[ f(x) = \frac{1}{\sqrt{|x|}} \]

\[ f(x) = \frac{1}{|x|} \]

Area → finite

Area → \infty
The Singular Kernel Problem

Comparing Quadrature Schemes

Approximately Integrating \( \int_{-1}^{1} \frac{1}{\sqrt{|x|}} \, dx \)

Error

Gauss-Quad

Large errors even with many points!

Simple Quad

Number of Points

2 4 6 8 10 12 14 16
Subdivide the integration interval
\[ \int_{-1}^{1} \frac{1}{\sqrt{|x|}} \, dx = \int_{-1}^{-0.1} \frac{1}{\sqrt{|x|}} \, dx + \int_{-0.1}^{0} \frac{1}{\sqrt{|x|}} \, dx + \int_{0}^{0.1} \frac{1}{\sqrt{|x|}} \, dx + \int_{0.1}^{1} \frac{1}{\sqrt{|x|}} \, dx \]

Use Gauss quadrature in each subinterval
Polynomials fit subintervals better
Expensive if many subintervals used.
The Singular Kernel Problem

Improved Techniques

Change of Variables - for Simple Cases

Change variables to eliminate singularity

\[ y^2 = x \]

\[ \Rightarrow 2y \, dy = dx \]

\[ \int_{-1}^{1} \frac{1}{\sqrt{|x|}} \, dx = 2 \int_{0}^{1} \frac{1}{\sqrt{|y^2|}} 2y \, dy = 2 \int_{0}^{1} 2 \, dy \]

Apply Gauss quadrature on desingularized integrand
Integrand has known singularity \( s(x) \)

\[
\int_{-1}^{1} f(x)s(x)\,dx \quad \text{where } f(x) \text{ is smooth}
\]

Develop a quadrature formula exact for

\[
\int_{-1}^{1} p_l(x)s(x)\,dx \quad \text{where } p_l(x) \text{ is polynomial of order } l
\]

Calculate weights like Gauss quadrature
The Singular Kernel Problem

Improved Techniques

Singular Quadrature Weights

\[
\begin{bmatrix}
    1 & 1 & \cdots & 1 \\
    c_0(x_1) & \cdots & \cdots & c_0(x_n) \\
    \vdots & \ddots & \ddots & \vdots \\
    c_{n-1}(x_1) & \cdots & \cdots & c_{n-1}(x_n)
\end{bmatrix}
\begin{bmatrix}
    w_1 \\
    w_2 \\
    \vdots \\
    w_n
\end{bmatrix}
= 
\begin{bmatrix}
    \int_1^s s(x) \, dx \\
    \vdots \\
    \int_{-1}^{c_{n-1}(x)} s(x) \, dx
\end{bmatrix}
\]

Need (analytic) formulas for integral of \( c(x) s(x) \)
Summary

Easy technique for computing integrals
  Piecewise constant approach

Gaussian quadrature
  Faster convergence
  Essential role of orthogonal polynomials

Techniques for singular kernels
  Adaptation and Variable Transformation
  Singular quadrature

What about multiple dimensions?