Numerical Methods for PDEs

Integral Equation Methods, Lecture 2
Numerical Quadrature

Notes by Suvranu De and J. White

April 28, 2003

1 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.

2 3D Laplace's Equation

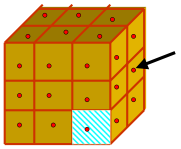
2.1 Basis Function Approach

2.1.1 Centroid Collocation

Slide 1

Put collocation points at panel centroids

$$\Psi(x_{c_i}) = \sum_{j=1}^{n} \alpha_j \underbrace{\int_{panel\ j} \frac{1}{\left\| x_{c_i} - x' \right\|} dS'}_{A_{i,j}}$$



$$\begin{bmatrix} A_{1,1} & \cdots & \cdots & A_{1,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{n,1} & \cdots & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \Psi(x_{c_1}) \\ \vdots \\ \vdots \\ \Psi(x_{c_n}) \end{bmatrix}$$

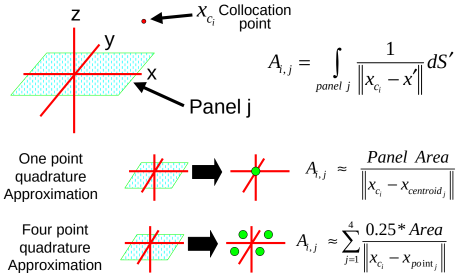
Note 1

In the last lecture we were introduced to integral equations and several different techniques for discretizing them were described. It was pointed out that one of the most popular means of obtaining a discrete set of equations is to use a piecewise constant centroid collocation scheme. We consider a simple problem of solving Laplace's equation in 3D. The potential, Ψ is prescribed on the surface of the cube and we need to compute the charge distribution σ. In order to do that we break the surface of the cube up into n panels and assume a constant charge distribution on each panel. Mathematically, this corresponds to assuming piecewise constant basis functions, each basis function, φ_i being compactly supported on the i^{th} panel. The resulting semi-discrete equation is a function of the spatial variable x. In order to obtain a discrete set of equations, we assume that this semi-discrete equation is satisfied exactly at the centroids of the panels. This gives rise to the set of n equations corresponding to the n panels. Mathematically, this process of collocation corresponds to setting the residual orthogonal to a set of delta functions located at the panel centroids.

This process leaves us with a set of n linear algebraic equations to solve. The quantities of interest being the collocation point weights α_j's. In today's lecture we will concentrate on obtaining the entries of the n-by-n matrix shown in the slide above. In the centroid collocation technique, the matrix entries involve integrals of the Green's function over the panels. Physically, the term A_{ij} of this matrix corresponds to the potential at the centroid of the i^{th} panel due to unit charge density distribution on the j^{th} panel.

2.1.2 Calculating Matrix Elements

Slide 2



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

One point quadrature Approximation

$$A_{i,j} \approx \frac{Panel\ Area}{\left\| x_{c_i} - x_{centroid_j} \right\|}$$

Four point quadrature Approximation

$$A_{i,j} \approx \sum_{j=1}^{4} \frac{0.25 * Area}{\left\| x_{c_i} - x_{point_j} \right\|}$$

Note 2

One very simple way of computing the integral in A_{ij}, for a panel j which is far removed from panel i, is to simply replace the integral by the integrand evaluated at the centroid of the panel j. Of course, this is too simplistic for panels which interact "strongly" with the panel i. For these panels which are closer by, we may use a four-point integration scheme. In our minds we split the panel up into four equal subpanels and write the integral over the entire panel j as the sum of four integrals on these four subpanels. Then use the same trick as before, replacing each of these four integrals by the product of the integrand, evaluated at the centroid of each of these subpanels and the area of the subpanel. From intution we believe that this scheme is going to give us a more accurate answer. The question, however, is whether this is the best way to go or are there better techniques? Can we do the same kind of integration on panel i? To be able to answer these questions we will take a brief look at how numerical integration is performed, a field of study known as "quadrature".

3 Normalized 1D Problem

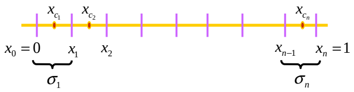
3.1 Basis Function Approach

3.1.1 Collocation Discretization of 1D Equation

SLIDE 3

$$\Psi(x) = \int_0^1 g(x, x')\sigma(x')dS' \qquad x \in [0, 1]$$

Centroid collocated piecewise constant scheme



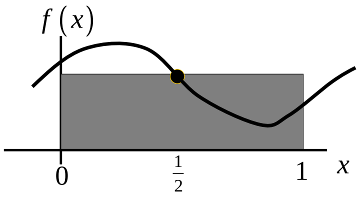
$$\Psi(x_{c_i}) = \textstyle\sum_{j=1}^{n} \sigma_j \underbrace{\int_{x_{j-1}}^{x_j} g(x_{c_i}, x')dS'}_{\textit{to be evaluated}}$$

Note 3

Lets take a simple example in 1D. The domain is the segment $[0, 1]$ of the real line. We want to solve the integral equation shown at the top of the slide on this domain. The Green's function is denoted by $g(x, x')$. In a centroid collocation scheme, we divide the domain into n segments $[x_j, x_{j+1}] \quad j = 0, \ldots, n$, with $x_0 = 0$ and x_n=1. The charge density σ is assumed to be piecewise constant on each of these intervals. The potential, Ψ is then evaluated at the centroids x_{c_i}. This results in n equations in n variables, the collocation weights $\alpha_i, i = 1, \ldots, n$ which can be written in matrix form. Our task is to first evaluate the entries of the matrix and subsequently solve the set of equations. Note that the ij^{th} entry of the matrix is an integral of the Green's function, evaluated at the collocation point x_{c_i}, over the interval $[x_j, x_{j+1}]$, which is the interval over which the basis function φ_j is nonzero (recall that we have chosen a piecewise constant approximation). If, however, we decided to choose a different set of basis functions, this integral would be nonzero only on the support of the basis functions (or, to be precise, on the intersection of the support of the basis function and $[0, 1]$).

3.2 Simple Quadrature Scheme

$$\int_0^1 f(x)dx \simeq f\left(\frac{1}{2}\right)$$



Area under the curve is approximated by a rectangle

Note 4

For the time being lets concentrate on the topic of developing a good numerical technique for evaluating the integral of a function$f(x)$ on the domain $[0,1]$. We assume that the integrand is a "smooth" function, though we will examine this assumption later. First we are going to develop a naive approach for obtaining a good approximation of the integral of this function on this interval, which we call the "simple quadrature scheme".

The simplest thing we can do is to replace the integral with the the product of the integrand, evaluated at a point inside the interval, and the length of the interval, which in this case is unity. If we choose the point of evaluation as the centroid of the interval, i.e. $x = 0.5$, we call the scheme "midpoint quadrature". A midpoint quadrature scheme replaces the area under the curve $f(x)$ by a rectangle whose height is the function $f(x)$ evaluated at $x = 0.5$. The scheme is exact when $f(x)$ is a constant. However, what is less obvious is that the scheme is exact when $f(x)$ is a linear function of x as well. The most obvious way of seeing this is by realizing that when $f(x)$ is a straight line, the area under it is a trapezoid. This trapezoid has exactly the same area as the rectangle which this scheme uses to approximate the integral (can you see why?).

Lets try to derive this in a slightly different way. Instead of the interval being $[0,1]$ we consider an interval $[0,h], h > 0$ to be a bit more general. We may exapnd $f(x)$ about the centroid of this interval, $\bar{x} = \frac{h}{2}$

$$f(x) = f(\bar{x}) + \Delta(x)\frac{df(\bar{x})}{dx} + \frac{\Delta(x)^2}{2!}\frac{d^2f(\xi)}{dx^2} \quad for\ some\ \xi \in [0,h]$$

where $\Delta(x) = x - \bar{x}$. The last term in the expansion is the Taylor series remainder. Lets integrate this exapansion over the interval $[0,h]$

$$\int_0^h f(x)dx = hf(\bar{x}) + \frac{h^3}{24}\frac{d^2f(\xi)}{dx^2}$$

Hence the error in the midpoint quadrature approximation is

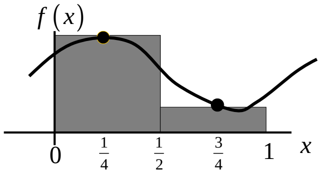
$$E = \int_0^h f(x)dx - hf(\bar{x}) = \frac{h^3}{24}\frac{d^2f(\xi)}{dx^2}$$

A straight line or a constant would generate a zero as the second derivative and therefore the above expression tells us that the error is identically zero for both these cases.
Now, another important piece of information that we obtain from this little exercise is that the error scales as the cube of the domain length. This tells us that for small enough intervals our simple rule is OK but for larger domain lengths we have to seek better methods.

3.2.1 Improving the Accuracy

SLIDE 5

$$\int_0^1 f(x)dx \simeq \frac{1}{2}f\left(\frac{1}{4}\right) + \frac{1}{2}f\left(\frac{3}{4}\right)$$



Area under the curve is approximated by two rectangles

Note 5

One way of improving may be to divide the interval $[0, 1]$ into subintervals $[0, 0.5]$ and $[0.5, 1]$ and write the integral

$$\int_0^1 f(x)dx = \int_0^{0.5} f(x)dx + \int_{0.5}^1 f(x)dx$$

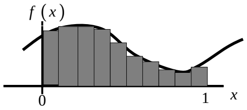
and apply a midpoint rule to the integral on each subinterval. We obtain a scheme shown in the slide. The factor $\frac{1}{2}$ appearing in front of $f(\frac{1}{2})$ and $f(\frac{3}{4})$ are just the domain lengths.

▷ ***Exercise 1*** Can you come up with an expression for the error in this case? How much does the accuracy improve? ■

3.2.2 General n-Point Formula

SLIDE 6

$$\int_0^1 f(x)dx \simeq \sum_{i=1}^{n} \frac{1}{n} f\left(\frac{i-\frac{1}{2}}{n}\right)$$



Key questions about the method:
How fast do the errors decay with n?
Are there better methods?

Note 5

As you can see dividing the interval into two reduces the error and there is no reason to stop at just two subintervals when we can have n subintevals and repeat our midpoint quadrature rule on each subinterval. We obtain the scheme

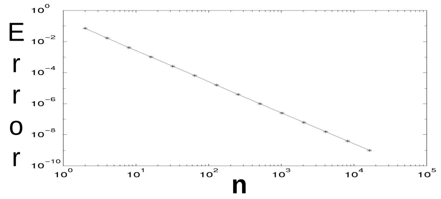
$$\int_0^1 f(x)dx = \sum_{i=1}^{n} \underbrace{\frac{1}{n}}_{\text{subinterval length}} f(x_{c_i})$$

where the centroid of the i^{th} subinterval is $x_{c_i} = \frac{1}{2}(\frac{i-1}{n} + \frac{i}{n}) = \frac{i-\frac{1}{2}}{n}$. There is no doubt that we gain, but the key question is by how much? How does this gain scale with the number of subintervals used? And finally, are there clever ways of obtaining better accuracy with less effort?

3.2.3 Numerical Example

SLIDE 7

$$\int_0^1 sin(x)dx \simeq \sum_{i=1}^{n} \frac{1}{n} sin\left(\frac{i-\frac{1}{2}}{n}\right)$$



Note 6

Lets look at an example of integrating $f(x) = sin(x)$ on our domain. We obtain progressively better answers to the integral by increasing the number of subintervals n. The error in evaluating the integral is plotted as a function of the number of subintervals (n). The error appears to be going down as $\mathcal{O}\left(\frac{1}{n^2}\right)$. Lets see why.

From what we have just seen, the error inside the i^{th} subinterval (of length '$h = \frac{1}{n}$') is $\frac{h^3}{24}\frac{d^2 f(\xi_i)}{dx^2}$ for some $\xi_i \in [\frac{i-1}{n}, \frac{i}{n}]$. Hence, for the entire interval $[0, 1]$ we can sum these errors and obtain the error, E_n for an approximation using

'n' subintervals as

$$E_n = \frac{nh^3}{24} \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} \frac{d^2 f(\xi_i)}{dx^2} \right)}_{call\ this\ `M'}$$

It is easy to see that if $f(x)$ is a continuous function, 'M' (being the mean) must be bounded by the maximum and the minumum of $f(x)$ on the interval $[0, 1]$ and hence, there must exist some $\xi \in [0, 1]$ such that $M = d^2 f(\xi)/dx^2$. Hence we obtain the estimate

$$E_n = \frac{nh^3}{24} \frac{d^2 f(\xi)}{dx^2} = \frac{1}{24n^2} \frac{d^2 f(\xi)}{dx^2}$$

since $h = 1/n$. This error estimate tells us that the scheme is again exact for constants and linear functions on the domain (no higher order polynomials!) and, for a smooth function, the error decays **algebraically.**

3.3 General Quadrature Scheme

3.3.1 General 1D Form

SLIDE 8

$$\int_0^1 f(x)dx \simeq \sum_{i=1}^{n} \underbrace{w_i}_{weight} \underbrace{f(x_i)}_{Evaluation\ Point}$$

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

An n-point formula has 2n degrees of freedom!

Note 7

After all the hard work we did dividing the domian into subintervals, we realize that we cannot even integrate a parabola exactly on the domain. There must be something that we can do to improve this scheme. We go back and look at the general form of the quadrature approximation scheme. All we are doing is approximating an integral by a weighted sum of function evaluations as shown in this slide. So far we have been choosing these weights as the subinterval lengths. We have also been choosing all the evaluation points. The weights are just some normalizing factors which ensure that the approximation is exact if $f(x) = 1$ and the equality of areas of trapezoids and rectangles that we discussed gives us the extra polynomial accuracy of being able to obtain the area under a straing line exactly. So, what would happen if we were to choose both the integration points and the weights intelligently? For an n-point formula we have

'n' weights and 'n' evaluation points to choose. That gives us '$2n$' degrees of freedom. Hence we must be able to integrate a polynomial of degree at most '$(2n-1)$' . This simple idea gives rise to the so called "Gauss quadrature" scheme.

3.3.2 Point-Weight Selection Criteria

SLIDE 9

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, $l = 2n - 1$

Note 8

Let $p_l(x)$ denote a polynomial of degree l in the variable x ($a_l \neq 0$). We want to select the weights and integration points such that the formula

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

is exact for all polynomials of degree upto (and including) l. Obviously, with $2n$ degrees of freedom, the best we can do is $l = 2n - 1$.

3.3.3 Why the Exactness Criteria?

SLIDE 10

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 $\tilde{x} \in [0, x]$

Note 9

Of all functions, why are we interested in integrating polynomials? The reason comes from the structure of Taylor's series expansion. The Taylor expansion of a function in a local neighborhood of a point (here this point is chosen as 0 without loss of generality) is nothing but a power series expansion! The higher the order of polynomials that our scheme can integrate the higher the order of the remainder term in the expansion. The integral of the remainer over the domain is precisely the error in numerical integration.

3.3.4 Estimating the Error

SLIDE 11

Using the Taylor series results and the exactness criteria
$\int_0^1 f(x)dx - \sum_{i=1}^n w_i f(x_i) = \underbrace{\frac{1}{(l+1)!}\int_0^1 \frac{\partial^{l+1} f(\tilde{x}(x))}{\partial x^{l+1}} x^{l+1} dx}_{Remainder}$ Assuming derivatives of $f(x)$ are bounded on $[0,1]$

$$\left| \int_0^1 f(x)dx - \sum_{i=1}^n w_i f(x_i) \right| \le \frac{K}{(l+1)!}$$

Convergence is **very** fast!!

Note 10

Assume that our scheme is exact upto a polynomial order 'l'. That means we can integrate the first $(l+1)$ terms in this Taylor series expansion exactly. The error in numerical integration

$$E = \int_0^1 f(x)dx - \sum_{i=1}^n w_i f(x_i) = \frac{1}{(l+1)!}\int_0^1 \frac{\partial^{l+1} f(\tilde{x}(x))}{\partial x^{l+1}} x^{l+1} dx.$$

3.3.5 Meeting the Exactness Criteria

SLIDE 12

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

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

Note 11

This slide needs little clarification. Our exactness criterion is

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

which is the same as

$$a_0\int_0^1 dx+a_1\int_0^1 xdx+\cdots+a_l\int_0^1 x^l dx = a_0\sum_{i=1}^n w_i+a_1\sum_{i=1}^n w_i x_i+\cdots+a_l\sum_{i=1}^n w_i {x_i}^l$$

For this to be an identity for the $(l+1)$ arbitrary coefficients a_i, we must have the $(l+1)$ conditions

$$\sum_{i=1}^n w_i {x_i}^j = \int_0^1 x^j dx \qquad for\ j = 0, 1, \dots, l$$

3.3.6 Meeting the Exactness Criteria

Slide 13

Exactness condition will be satisfied if and only if

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

$$\int_0^1 xdx = \sum_{i=1}^n w_i \cdot x_i$$

$$\vdots$$

$$\int_0^1 x^l dx = \sum_{i=1}^n w_i \cdot {x_i}^l$$

3.3.7 Meeting the Exactness Criteria

Slide 14

Reorganizing exactness equations

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

Nonlinear, since x_i's and w_i's are unknowns

Note 12

Now what is a practical way of computing the evaluation points and weights? We can write the exactness criteria into a matrix from. The system of equations is not easy to solve since x_i's and w_i's are unknowns.

3.3.8 Computing the Points and Weights

SLIDE 15

Could use **Newton's Method**

$$F(y) = 0 \Rightarrow J_F\left(y^k\right)\left(y^{k+1} - y^k\right) = -F\left(y^k\right)$$

The nonlinear function for Newton is then

$$F\begin{pmatrix} w \\ x \end{pmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^l & x_2^l & \cdots & x_n^l \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} - \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \int_0^1 x^l dx \end{bmatrix} = 0$$

Note 13

Newton's method is an iterative technique for finding a value y such that $F(y) = 0$. The method is based on linearizing the problem about a guess at y, and then updating the value of y by solving the linearized problem. In particular, the iterate y^{k+1} is determined from y^k by solving the linear system of equations

$$F(y^k) + J_F(y^k)\left(y^{k+1} - y^k\right) = 0$$

where $J_F(y^k)$ is the Jacobian (multidimensional derivative) of the nonlinear function $F(y)$. The iteration is continued until the updated y is sufficiently close to the exact solution, a criterion that can be difficult to verify. Newton's method does not always converge, a phenomenon that is more likely when $J_F(y)$ is nearly singular. For more about Newton's method, see the 6.336/16.920/2.096 course notes.

3.3.9 Computing the Points and Weights

SLIDE 16

Newton Method Jacobian reveals problem

$$J_F\begin{pmatrix} w \\ x \end{pmatrix} = \overset{\xleftrightarrow{\hspace{8em} 2n \hspace{8em}}}{\left[\begin{array}{cc} \underbrace{\begin{matrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^l & x_2^l & \cdots & x_n^l \end{matrix}}_{\substack{\text{Columns become} \\ \text{linearly dependent for} \\ \text{high order}}} & \begin{matrix} 0 & 0 & \cdots & 0 \\ w_1 & w_2 & \cdots & w_n \\ \vdots & \vdots & \ddots & \vdots \\ lw_1x_1^{l-1} & \cdots & \cdots & lw_1x_n^{l-1} \end{matrix} \end{array}\right]}$$

Note 14

Looking at the Jacobian of the problem, we realize that the first 'n' columns become increasingly linearly dependent for large l. This is bound to happen since we are looking into the space $span\{1, x, \ldots, x^l\}$. This basis always becomes ill conditioned with increasing l. The solution is to obtain a polynomial basis that is "normalized" in some sense so that it is properly conditioned.

3.3.10 Use Different Polynomials

SLIDE 17

Exactness criteria will be satisfied if and only if

$$\left.\begin{aligned}\int_0^1 c_0(x)\,dx &= \sum_{i=1}^{n} w_i c_0(x_i)\\ \int_0^1 c_1(x)\,dx &= \sum_{i=1}^{n} w_i c_1(x_i)\\ &\vdots\\ \int_0^1 c_l(x)\,dx &= \sum_{i=1}^{n} w_i c_l(x_i)\end{aligned}\right\}$$

BUT

Each c_i polynomial must

Contain an x^i term

Be linearly independent

Note 15

The only difference from the previous set of criteria is that these polynomials have better properties than the ones we chose before.

3.3.11 Orthogonal Polynomials

SLIDE 18

For the normalized integral, two poynomials are said to be **orthogonal** if

$$\int_0^1 c_i(x)c_j(x)dx = 0 \qquad for\ j \neq i$$

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.

3.3.12 Exploiting the Different Polynomials

SLIDE 19

Consider rewriting the exactness criteria

$$\underbrace{\begin{aligned}\int_0^1 c_0(x)\,dx &= \sum_{i=1}^{n} w_i c_0(x_i)\\ &\vdots\\ \int_0^1 c_{n-1}(x)\,dx &= \sum_{i=1}^{n} w_i c_{n-1}(x_i)\end{aligned}}_{\text{Low order terms}} \qquad \underbrace{\begin{aligned}\int_0^1 c_n(x)\,dx &= \sum_{i=1}^{n} w_i c_n(x_i)\\ &\vdots\\ \int_0^1 c_{2n-1}(x)\,dx &= \sum_{i=1}^{n} w_i c_{2n-1}(x_i)\end{aligned}}_{\text{High Order Terms}}$$

Recall that $l(\#\ polys) = 2n - 1(\#\ of\ coefficients)$

Note 16

We just call the first $(n-1)$ conditions as the "lower order terms" and the last n conditions as the "higher order terms".

3.3.13 Exploiting the Different Polynomials

SLIDE 20

Can write the higher order terms differently
$\int_0^1 c_n(x)dx = \sum_{i=1}^n w_i c_n(x_i) \Rightarrow \int_0^1 c_n(x)c_0(x)dx = \sum_{i=1}^n w_i c_n(x_i)c_0(x_i)$

$$\vdots$$

$$\vdots$$

$$\int_0^1 c_{2n-1}(x)dx = \sum_{i=1}^n w_i c_{2n-1}(x_i) \Rightarrow \int_0^1 c_n(x)c_{n-1}(x)dx = \sum_{i=1}^n w_i c_n(x_i)c_{n-1}(x_i)$$

The products $c_n(x)c_j(x)$ are linearly independent!

Note 17

In this slide we express the "higer order terms" as conditions involving "lower order terms".

3.3.14 Using Orthogonality and Roots

SLIDE 21

$$\cancelto{0}{\int_0^1 c_n(x)c_0(x)\,dx} = \cancelto{0}{\sum_{i=1}^{n} w_i c_n(x_i)c_0(x_i)}$$

$$\vdots$$

$$\cancelto{0}{\int_0^1 c_n(x)c_{n-1}(x)\,dx} = \cancelto{0}{\sum_{i=1}^{n} w_i c_n(x_i)c_{n-1}(x_i)}$$

Use orthogonal polynomials
Pick the x_i's to be n roots of $c_n(x)$
The higher order constraints are exactly satisfied!

Note 18

This elegant step relies on polynomial orthogonality. If we choose the polynomial $c_n(x)$ such that it is orthogonal to all polynomials of inferior degree (i.e. $c_0(x), c_1(x), \ldots, c_{n-1}(x)$) and the x_i's are roots of this polynomial, then the higher order n conditions are automatically satisfied. Note that for this derivation we used polynomials which are orthogonal on the interval $[0, 1]$. Such polynomials are shifted and scaled versions of the classical Legendre polynomials, which are orthogonal on the interval $[-1, 1]$.

3.3.15 Satisfying the Lower Order Constraints

SLIDE 22

An abbreviated exactness equation

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

Now **linear**, x_i's are known
Rows are sampled orthogonal polynomials.

Note 19

By using the roots of $c_n(x)$ for the x_i's, the higher order constraints are automatically satisfied. Since the x_i's are now known, only the weights are still unknown. The lower n constraints can be used to determine the weights, generating a linear system.

3.4 Gaussian Quadrature Summary

3.4.1 Algorithm Steps

SLIDE 23

1. Construct $n + 1$ orthogonal polynomials

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

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

Note 20

This slide summarizes the technique of finding weights and integration points for Gauss quadrature.

3.4.2 Accuracy Result

Slide 24

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

Key properties of the method

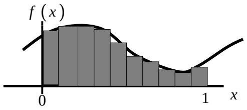
- An n-point Gauss quadrature rule is **exact** for polynomials of order $2n-1$
- Error is proportional to $\left(\frac{1}{2n}\right)^{2n}$

3.5 Simple Quadrature Scheme

3.5.1 General n-Point Formula

Slide 25

$$\int_0^1 f(x)dx \simeq \sum_{i=1}^{n} \frac{1}{n} f\left(\frac{i-\frac{1}{2}}{n}\right)$$



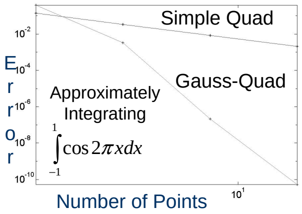
Key property of the method

- Error is proportional to $\frac{1}{n^2}$

▷ ***Exercise 2*** Do you see that the simple quadrature scheme is a special case of Gauss quadrature? ■

3.6 Comparing Simple Quad and Gauss Quad

Slide 26



Note 21

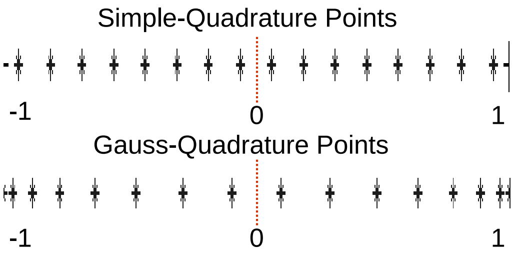
Notice that for a smooth function $f(x) = cos(2\pi x)$, which is infinitely differentiable, Gauss quadrature far outperforms the simple quadrature scheme

---

3.7 Comparing Simple Quad and Gauss Quad

3.7.1 Evaluation Point Placement

SLIDE 27



Notice the clustering at interval ends

Note 22

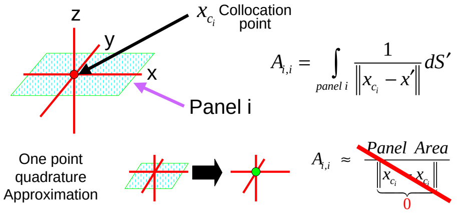
In the Gauss quadrature scheme the evaluation points are roots of Legendre polynomials which are clustered at the ends of the interval.

4 The Singular Kernel Problem

4.1 3D Laplace Example

4.1.1 Calculating the "Self-Term"

SLIDE 28


z
y
x
x_{c_i} Collocation point
Panel i
One point quadrature Approximation

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

$$A_{i,i} \approx \cancel{\frac{Panel\ Area}{\underbrace{\left\| x_{c_i} - x_{c_i} \right\|}_{0}}}$$

$$\boxed{A_{i,i} = \int_{panel\ i} \frac{1}{\left\| x_{c_i} - x' \right\|} dS' \text{ is an integrable singularity}}$$

Note 23

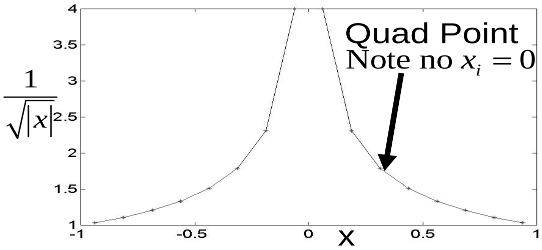
Now lets go back to our problem of solving Laplace's equation on a 3D domain using boundary integral representation. We realize that we now have some sophisticated tools to handle integrals of functions that are smooth. But what about the integral on the panel where the centroid x_{c_i} is located? The Green's function blows up at the centroid. However, the function is integrable because the integrand blows up at a rate that is slower than the rate at which the surface measure goes to zero in the vicinity of the singularity. So we know that the integral exists and is finite, but is Gauss quadrature capable of performing well in the presence of this singularity?

4.2 Symmetrized 1D Example

4.2.1 Example

SLIDE 29

$$\int_{-1}^{1} \frac{1}{\sqrt{|x|}} dx \simeq \sum_{i=1}^{n} w_i \frac{1}{\sqrt{|x_i|}}$$

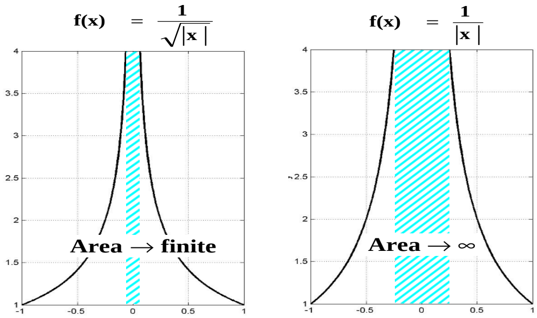


Note 24

In 1D we look at a function $f(x) = \frac{1}{\sqrt{|x|}}$ which is integrable on $[-1, 1]$ but has a singularity at $x = 0$.

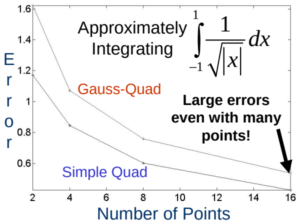
4.2.2 Integrable and Nonintegrable Singularities

SLIDE 30



4.3 Comparing Quadrature Schemes

SLIDE 31



Note 25

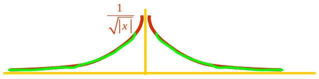
We observe that Gauss quadrature is not very good at integrating this function. The convergence is rather poor. As a matter of fact, it is more inaccurate than the simple quadrature scheme. In the next few slides we present several techniques of handling integrals with singularities (which are integrable, of course)

- Subinterval (adaptive) quadrature
- Change of variables of integration
- Singular (Gaussian) quadrature

4.4 Improved Techniques

4.4.1 Subinterval (Adaptive) Quadrature

SLIDE 32



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.

4.4.2 Change of Variables - for Simple Cases

SLIDE 33

Change variables to eliminate singularity

$$y^2 = x$$

$$\Rightarrow 2ydy = dx$$

$\int_{-1}^{1}\frac{1}{\sqrt{|x|}}dx = 2\int_{0}^{1}\frac{1}{\sqrt{|y^2|}}2ydy = 2\int_{0}^{1}2dy$

Apply Gauss quadrature on desingularized intergrand

4.4.3 Singular Quadrature - Complicated Cases

SLIDE 34

Integrand has known singularity $s(x)$
$\int_{-1}^{1} f(x)s(x)dx$ where $f(x)$ is smooth

Develop a quadrature formula exact for

$\int_{-1}^{1} p_l(x)s(x)dx$ where $p_l(x)$ is polynomial of order l

Calculate weights like Gauss quadrature

Note 26

It is possbile to generate Gaussian quadrature schemes of the form

$$\int_{-1}^{1} s(x)f(x)dx = \sum_{i=1}^{n} w_i f(x_i)$$

for functions which have a known singularity $s(x) > 0$. The quadrature formula needs to be exact when $f(x)$ is a polynomial of order at most l. Not surprisingly, it turns out that the integration points are the n roots of a polynomial $c_n(x)$ of degree $n = (l+1)/2$ which is orhogonal to all polynomials of inferior degree with respect to the weight s(x), i.e.

$$\int_{-1}^{1} s(x)c_n(x)c_j(x) = 0 \qquad for\ j = 0, 1, \ldots, (n-1).$$

An example is the singular integral

$$I = \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}$$

Here, $s(x) = 1/\sqrt{1-x^2}$ and the corresponding orthogonal polynomials turn out to be the Chebyshev polynomials. The integration points are given in closed form by

$$x_i = cos\left(\pi \frac{2i-1}{2n}\right)$$

and the corresponding weights are $w_i = \pi/n$.

4.4.4 Singular Quadrature Weights

SLIDE 35

$$\begin{bmatrix} c_0(x_1) & \cdots & c_0(x_n) \\ \vdots & \vdots & \vdots \\ c_{n-1}(x_1) & \cdots & c_{n-1}(x_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \int_{-1}^{1} c_0(x)s(x)dx \\ \vdots \\ \int_{-1}^{1} c_{n-1}(x)s(x)dx \end{bmatrix}$$

Need (analytic) formulas for integrals of $c(x)s(x)$

Note 27

The lower order constraints can be used to compute the integration weights.

5 Summary

SLIDE 36

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?