Method of Undetermined Coefficients

Quadrature Rules

Integration

Scientific Computing: An Introductory Survey
Chapter 8 – Numerical Integration and Differentiation

Prof. Michael T. Heath
Department of Computer Science
University of Illinois at Urbana-Champaign

Copyright © 2002. Reproduction permitted for noncommercial, educational use only.

Integration

- For \( f : \mathbb{R} \to \mathbb{R} \), definite integral over interval \([a, b]\)
  \[
  I(f) = \int_a^b f(x) \, dx
  \]
  is defined by limit of Riemann sums
  \[
  R_n = \sum_{i=1}^n (x_{i+1} - x_i) f(\xi_i)
  \]
  - Riemann integral exists provided integrand \( f \) is bounded and continuous almost everywhere
  - Absolute condition number of integration with respect to perturbations in integrand is \( b - a \)
  - Integration is inherently well-conditioned because of its smoothing effect

Quadrature Rules

- An \( n \)-point quadrature rule has form
  \[
  Q_n(f) = \sum_{i=1}^n w_i f(x_i)
  \]
  - Points \( x_i \) are called nodes or abscissas
  - Multipliers \( w_i \) are called weights
  - Quadrature rule is
    - open if \( a < x_1 \) and \( x_n < b \)
    - closed if \( a = x_1 \) and \( x_n = b \)

Method of Undetermined Coefficients

- Alternative derivation of quadrature rule uses method of undetermined coefficients
- To derive \( n \)-point rule on interval \([a, b]\), take nodes \( x_1, \ldots, x_n \) as given and consider weights \( w_1, \ldots, w_n \) as coefficients to be determined
- Force quadrature rule to integrate first \( n \) polynomial basis functions exactly, and by linearity, it will then integrate any polynomial of degree \( n - 1 \) exactly
- Thus we obtain system of moment equations that determines weights for quadrature rule

Outline

- Numerical Integration
- Numerical Differentiation
- Richardson Extrapolation

Quadrature Rules, continued

- Quadrature rules are based on polynomial interpolation
  - Integrand function \( f \) is sampled at finite set of points
  - Polynomial interpolating those points is determined
  - Integral of interpolant is taken as estimate for integral of original function
  - In practice, interpolating polynomial is not determined explicitly but used to determine weights corresponding to nodes
  - If Lagrange is interpolation used, then weights are given by
    \[
    w_i = \int_a^b \ell_i(x), \quad i = 1, \ldots, n
    \]

Example: Undetermined Coefficients

- Derive 3-point rule
  \[
  Q_3(f) = w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3)
  \]
  on interval \([a, b]\) using monomial basis
  - Take \( x_1 = a, x_2 = (a + b)/2, \) and \( x_3 = b \) as nodes
  - First three monomials are \( 1, x, \) and \( x^2 \)
  - Resulting system of moment equations is
    \[
    \begin{align*}
    w_1 \cdot 1 + w_2 \cdot 1 + w_3 \cdot 1 &= \int_a^b 1 \, dx = x_n = b - a \\
    w_1 \cdot a + w_2 \cdot (a + b)/2 + w_3 \cdot b &= \int_a^b x \, dx = (x^2/2)|_a^b = (b^3 - a^3)/2 \\
    w_1 \cdot a^2 + w_2 \cdot ((a + b)/2)^2 + w_3 \cdot b^2 &= \int_a^b x^2 \, dx = (x^3/3)|_a^b = (b^3 - a^3)/3
    \end{align*}
    \]
Example: Newton-Cotes Quadrature

In matrix form, linear system is
\[
\begin{bmatrix}
1 & 1 & 1 \\
(a + b)/2 & b & (a + b)/2 \\
5 & (a + b)/3 & b^3
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
w_3
\end{bmatrix}
= \begin{bmatrix}
b - a \\
(b^2 - a^2)/2 \\
(b^3 - a^3)/3
\end{bmatrix}
\]

Solving system by Gaussian elimination, we obtain weights
\[
w_1 = \frac{b - a}{6}, \quad w_2 = \frac{2(b - a)}{3}, \quad w_3 = \frac{b - a}{6}
\]
which is known as Simpson's rule

Accuracy of Quadrature Rules

- Quadrature rule is of degree \( d \) if it is exact for every polynomial of degree \( d \), but not exact for some polynomial of degree \( d + 1 \)
- By construction, \( n \)-point interpolatory quadrature rule is of degree at least \( n - 1 \)
- Rough error bound
\[
|I(f) - Q_n(f)| \leq \frac{1}{2} b^{n+1} \|f^{(n)}\|_{\infty}
\]
where \( h = \max(x_{i+1} - x_i : i = 1, \ldots, n - 1) \), shows that \( Q_n(f) \to I(f) \) as \( n \to \infty \), provided \( f^{(n)} \) remains well behaved
- Higher accuracy can be obtained by increasing \( n \) or by decreasing \( h \)

Stability of Quadrature Rules

- Absolute condition number of quadrature rule is sum of magnitudes of weights,
\[
\sum_{i=1}^{n} |w_i|
\]
- If weights are all nonnegative, then absolute condition number of quadrature rule is \( b - a \), same as that of underlying integral, so rule is stable
- If any weights are negative, then absolute condition number can be much larger, and rule can be unstable

Example: Newton-Cotes Quadrature

Approximate integral \( I(f) = \int_a^b f(x) \, dx \approx 0.746824 \),
\[
M(f) = (1/0) \exp(-1/4) \approx 0.778801
\]
\[
T(f) = (1/2) \exp(0) + \exp(-1) \approx 0.683940
\]
\[
S(f) = (1/6) \exp(0) + 4 \exp(-1/4) + \exp(-1) \approx 0.747180
\]

Method of Undetermined Coefficients

- More generally, for any \( n \) and choice of nodes \( x_1, \ldots, x_n \),

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_n \\
x_1^2 & x_2^2 & \cdots & x_n^2
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
w_3
\end{bmatrix}
= \begin{bmatrix}
b - a \\
(b^2 - a^2)/2 \\
(b^3 - a^3)/3
\end{bmatrix}
\]
determines weights \( w_1, \ldots, w_n \)

Progressive Quadrature Rules

- Sequence of quadrature rules is progressive if nodes of \( Q_{n_k} \) are subset of nodes of \( Q_{n_{k+1}} \) for \( n_{k+1} > n_k \)
- For progressive rules, function evaluations used in one rule can be reused in another, reducing overall cost
- To attain higher accuracy, we can increase number of points \( n \) or subdivide interval into smaller subintervals
- In either case, efficiency is enhanced if successive rules are progressive so that fewer new evaluations of integrand are required

Newton-Cotes Quadrature

Newton-Cotes quadrature rules use equally spaced nodes in interval \([a, b] \)
- Midpoint rule
\[
M(f) = (b - a) f \left( \frac{a + b}{2} \right)
\]
- Trapezoid rule
\[
T(f) = \frac{b - a}{2} \left( f(a) + f(b) \right)
\]
- Simpson's rule
\[
S(f) = \frac{b - a}{6} \left( f(a) + 4 f \left( \frac{a + b}{2} \right) + f(b) \right)
\]

Error Estimation

- Expanding integrand \( f \) in Taylor series about midpoint \( m = (a + b)/2 \) of interval \([a, b] \),
\[
\begin{align*}
f(x) &= f(m) + f'(m)(x - m) + \frac{f''(m)}{2!}(x - m)^2 \\
&\quad + \frac{f'''(m)}{3!}(x - m)^3 + \frac{f^{(4)}(m)}{4!}(x - m)^4 + \ldots
\end{align*}
\]
- Integrating from \( a \) to \( b \), odd-order terms drop out, yielding
\[
I(f) = f(m)(b - a) + \frac{f''(m)}{24}(b - a)^3 + \frac{f^{(4)}(m)}{1920}(b - a)^5 + \ldots
\]
where \( E(f) \) and \( F(f) \) represent first two terms in error expansion for midpoint rule
Error Estimation, continued

- If we substitute \( x = a \) and \( x = b \) into Taylor series, add two series together, observe once again that odd-order terms drop out, solve for \( f(n) \), and substitute into midpoint rule, we obtain
  \[
  I(f) = T(f) - 2E(f) - 4F(f) - \ldots
  \]
- Thus, provided length of interval is sufficiently small and \( f^{(2k)} \) is well behaved, midpoint rule is about twice as accurate as trapezoid rule
- Halving length of interval decreases error in either rule by factor of about \( 1/8 \)

Example: Error Estimation

- We illustrate error estimation by computing approximate value for integral \( \int_a^b x^2 \, dx = 1/3 \)
  \[
  M(f) = (1 - 0)(1/2)^2 = 1/4
  \]
  \[
  T(f) = 1 - 0 (0^2 + 1^2) = 1/2
  \]
  \[
  E(f) \approx (T(f) - M(f))/3 = (1/4)/3 = 1/12
  \]
- Error in \( M(f) \) is about \( 1/12 \), error in \( T(f) \) is about \(-1/6 \)
- Also,
  \[
  S(f) = (2/3)M(f) + (1/3)T(f) = (2/3)(1/4) + (1/3)(1/2) = 1/3
  \]
  which is exact for this integral, as expected

Accuracy of Newton-Cotes Quadrature

- In general, odd-order Newton-Cotes rule gains extra degree beyond that of polynomial interpolant on which it is based
- \( n \)-point Newton-Cotes rule is of degree \( n - 1 \) if \( n \) is even, but of degree \( n \) if \( n \) is odd
- This phenomenon is due to cancellation of positive and negative errors

Drawbacks of Newton-Cotes Rules

- Newton-Cotes quadrature rules are simple and often effective, but they have drawbacks
- Using large number of equally spaced nodes may incur erratic behavior associated with high-degree polynomial interpolation (e.g., weights may be negative)
- Indeed, every \( n \)-point Newton-Cotes rule with \( n \geq 11 \) has at least one negative weight, and \( \sum_{j=1}^{n} |w_j| \rightarrow \infty \) as \( n \rightarrow \infty \), so Newton-Cotes rules become arbitrarily ill-conditioned
- Newton-Cotes rules are not of highest degree possible for number of nodes used

Gaussian Quadrature

- Gaussian quadrature rules are based on polynomial interpolation, but nodes as well as weights are chosen to maximize degree of resulting rule
- With \( 2n \) parameters, we can attain degree of \( 2n - 1 \)
- Gaussian quadrature rules can be derived by method of undetermined coefficients, but resulting system of moment equations that determines nodes and weights is nonlinear
- Also, nodes are usually irrational, even if endpoints of interval are rational
- Although inconvenient for hand computation, nodes and weights are tabulated in advance and stored in subroutine for use on computer
Example: Gaussian Quadrature Rule

- Derive two-point Gaussian rule on $[-1, 1]$,
  \[ G_2(f) = w_1 f(x_1) + w_2 f(x_2) \]
  where nodes $x_i$ and weights $w_i$ are chosen to maximize degree of resulting rule
- We use method of undetermined coefficients, but now nodes as well as weights are unknown parameters to be determined
- Four parameters are to be determined, so we expect to be able to integrate cubic polynomials exactly, since cubics depend on four parameters

Change of Interval, continued

- One solution of this system of four nonlinear equations in four unknowns is given by
  \[ x_1 = -1/\sqrt{3}, \quad x_2 = 1/\sqrt{3}, \quad w_1 = 1, \quad w_2 = 1 \]
- Another solution reverses signs of $x_1$ and $x_2$
- Resulting two-point Gaussian rule has form
  \[ G_2(f) = f(-1/\sqrt{3}) + f(1/\sqrt{3}) \]
  and by construction it has degree three
- In general, for each $n$ there is unique $n$-point Gaussian rule, and it is of degree $2n - 1$
- Gaussian quadrature rules can also be derived using orthogonal polynomials

Change of Interval, continued

- Many transformations are possible, but simple linear transformation
  \[ t = \frac{(b - a)x + a\beta - bx}{\beta - a} \]
  has advantage of preserving degree of quadrature rule

Example, continued

- Requiring rule to integrate first four monomials exactly gives moment equations
  \[
  \begin{align*}
  w_1 + w_2 &= \int_{-1}^{1} 1 \, dx = x^0_{-1,1} = 2 \\
  w_1 x_1 + w_2 x_2 &= \int_{-1}^{1} x \, dx = (x^2/2)_{-1,1} = 0 \\
  w_1 x_1^2 + w_2 x_2^2 &= \int_{-1}^{1} x^2 \, dx = (x^3/3)_{-1,1} = 2/3 \\
  w_1 x_1^3 + w_2 x_2^3 &= \int_{-1}^{1} x^3 \, dx = (x^4/4)_{-1,1} = 0
  \end{align*}
  \]
Colorado State University — Michael T. Heath
Scientific Computing 31 / 61

Gaussian Quadrature

- Gaussian quadrature rules have maximal degree and optimal accuracy for number of nodes used
- Weights are always positive and approximate integral always converges to exact integral as $n \to \infty$
- Unfortunately, Gaussian rules of different orders have no nodes in common (except possibly midpoint), so Gaussian rules are not progressive
- Thus, estimating error using Gaussian rules of different order requires evaluating integrand function at full set of nodes of both rules

Progressive Gaussian Quadrature

- Avoiding this additional work is motivation for Kronrod quadrature rules
- Such rules come in pairs, $n$-point Gaussian rule $G_n$, and $(2n + 1)$-point Kronrod rule $K_{2n+1}$, whose nodes are optimally chosen subject to constraint that all nodes of $G_n$ are reused in $K_{2n+1}$
- $(2n + 1)$-point Kronrod rule is of degree $3n + 1$, whereas true $(2n + 1)$-point Gaussian rule would be of degree $4n + 1$
- In using Gauss-Kronrod pair, value of $K_{2n+1}$ is taken as approximation to integral, and error estimate is given by
  \[ (200|G_n - K_{2n+1}|)^{1/5} \]
Composite Quadrature

- Alternative to using more nodes and higher degree rule is to subdivide original interval into subintervals, then apply simple quadrature rule in each subinterval.
- Summing partial results then yields approximation to overall integral.
- This approach is equivalent to using piecewise interpolation to derive composite quadrature rule.
- Composite rule is always stable if underlying simple rule is stable.
- Approximate integral converges to exact integral as number of subintervals goes to infinity provided underlying simple rule has degree at least zero.

Examples: Composite Quadrature

- Composite quadrature offers simple means of estimating error by using two different levels of subdivision, which is easily made progressive.
- For example, halving interval length reduces error in midpoint or trapezoid rule by factor of about 1/8.
- Halving width of each subinterval means twice as many subintervals are required, so overall reduction in error is by factor of about 1/4.
- If $h$ denotes subinterval length, then dominant term in error of composite midpoint or trapezoid rules is $O(h^2)$.
- Dominant term in error of composite Simpson’s rule is $O(h^4)$, so halving subinterval length reduces error by factor of about 1/16.

Adaptive Quadrature, continued

- Adaptive quadrature tends to be effective in practice, but it can be fooled: both approximate integral and error estimate can be completely wrong.
- Integrand function is sampled at only finite number of points, so significant features of integrand may be missed.
- For example, interval of integration may be very wide but “interesting” behavior of integrand may be confined to narrow range.
- Sampling by automatic routine may miss interesting part of integrand behavior, and resulting value for integral may be completely wrong.

Integrating Tabular Data

- If integrand is defined only by table of its values at discrete points, then reasonable approach is to integrate piecewise interpolant.
- For example, integrating piecewise linear interpolant to tabular data gives composite trapezoid rule.
- Excellent method for integrating tabular data is to use Hermite cubic or cubic spline interpolation.
- In effect, overall integral is computed by integrating each of cubic pieces that make up interpolant.
- This facility is provided by many spline interpolation packages.
Improper Integrals

To compute integral over infinite or semi-infinite interval, several approaches are possible

- Replace infinite limits of integration by carefully chosen finite values
- Transform variable of integration so that new interval is finite, but care must be taken not to introduce singularities
- Use quadrature rule designed for infinite interval

Double Integrals

Approaches for evaluating double integrals include

- Use automatic one-dimensional quadrature routine for each dimension, one for outer integral and another for inner integral
- Use product quadrature rule resulting from applying one-dimensional rule to successive dimensions
- Use non-product quadrature rule for regions such as triangles

Multiple Integrals

- To evaluate multiple integrals in higher dimensions, only generally viable approach is Monte Carlo method
- Function is sampled at \( n \) points distributed randomly in domain of integration, and mean of function values is multiplied by area (or volume, etc.) of domain to obtain estimate for integral
- Error in estimate goes to zero as \( 1/\sqrt{n} \), so to gain one additional decimal digit of accuracy requires increasing \( n \) by factor of 100
- For this reason, Monte Carlo calculations of integrals often require millions of evaluations of integrand

Integral Equations

- Typical integral equation has form
  \[
  \int_a^b K(s, t)u(t)\, dt = f(s)
  \]
  where kernel \( K \) and right-hand side \( f \) are known functions, and unknown function \( u \) is to be determined
- Solve numerically by discretizing variables and replacing integral by quadrature rule
  \[
  \sum_{j=1}^{n} w_j K(s_j, t_i)u(t_i) = f(s_i), \quad i = 1, \ldots, n
  \]
  This system of linear algebraic equations \( Ax = y \), where \( a_{ij} = w_j K(s_i, t_j) \), \( y_i = f(s_i) \), and \( x_j = u(t_j) \), is solved for \( x \) to obtain discrete sample of approximate values of \( u \)

Numerical Differentiation

- Differentiation is inherently sensitive, as small perturbations in data can cause large changes in result
- Differentiation is inverse of integration, which is inherently stable because of its smoothing effect
- For example, two functions shown below have very similar definite integrals but very different derivatives

- Richardson Extrapolation
- Finite Difference Approximations
- Automatic Differentiation

Numerical Integration

- Constrained optimization
- Regularization
- Truncated SVD
Finite Difference Approximations

- Given smooth function \( f : \mathbb{R} \rightarrow \mathbb{R} \), we wish to approximate its first and second derivatives at point \( x \).
- Consider Taylor series expansions:
  \[
  f(x + h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \cdots
  \]
  \[
  f(x - h) = f(x) - f'(x)h + \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 + \cdots
  \]
- Solving for \( f'(x) \) in first series, obtain forward difference approximation:
  \[
  f'(x) = \frac{f(x + h) - f(x) - f''(x)h}{h} + \cdots \approx \frac{f(x + h) - f(x)}{h} - f''(x)h^2 + \cdots
  \]
  which is first-order accurate since dominant term in remainder of series is \( O(h) \).

Richardson Extrapolation

- In many problems, such as numerical integration or differentiation, approximate value for some quantity is computed based on some step size.
- Ideally, we would like to obtain limiting value as step size approaches zero, but we cannot take step size arbitrarily small because of excessive cost or rounding error.
- Based on values for nonzero step sizes, however, we may be able to estimate value for step size of zero.
- One way to do this is called Richardson extrapolation.

- Suppose we have computed \( F \) for two step sizes, say \( h \) and \( h/q \) for some positive integer \( q \).
- Then we have:
  \[
  F(h) = a_0 + a_1h^p + O(h^r)
  \]
  \[
  F(h/q) = a_0 + a_1(h/q)^p + O(h^r) = a_0 + a_1q^{-r}h^p + O(h^r)
  \]
  This system of two linear equations in two unknowns \( a_0 \) and \( a_1 \) is easily solved to obtain:
  \[
  a_0 = F(h) - F(h/q)q^p \frac{1}{q^p - 1} + O(h^r)
  \]
  Accuracy of improved value, \( a_0 \), is \( O(h^r) \).

Automatic Differentiation

- Computer program expressing function is composed of basic arithmetic operations and elementary functions, each of whose derivatives is easily computed.
- Derivatives can be propagated through program by repeated use of chain rule, computing derivative of function step by step along with function itself.
- Result is true derivative of original function, subject only to rounding error but suffering no discretization error.
- Software packages are available implementing this automatic differentiation (AD) approach.

Richardson Extrapolation, continued

- Let \( F(h) \) denote value obtained with step size \( h \).
- If we compute value of \( F \) for some nonzero step sizes, and if we know theoretical behavior of \( F(h) \) as \( h \rightarrow 0 \), then we can extrapolate from known values to obtain approximate value for \( F(0) \).
- Suppose that:
  \[
  F(h) = a_0 + a_1h^p + O(h^r)
  \]
  as \( h \rightarrow 0 \) for some \( p \) and \( r \), with \( r > p \).
- Assume we know values of \( p \) and \( r \), but not \( a_0 \) or \( a_1 \) (indeed, \( F(0) = a_0 \) is what we seek).
Example: Richardson Extrapolation

- Use Richardson extrapolation to improve accuracy of finite difference approximation to derivative of function $\sin(x)$ at $x = 1$.
- Using first-order accurate forward difference approximation, we have
  
  $$F(h) = a_0 + a_1 h + O(h^2)$$

  so $p = 1$ and $r = 2$ in this instance.
- Using step sizes of $h = 0.5$ and $h/2 = 0.25$ (i.e., $q = 2$), we obtain
  
  $$F(h) = \frac{\sin(1.5) - \sin(1)}{0.5} = 0.312048$$
  $$F(h/2) = \frac{\sin(1.25) - \sin(1)}{0.25} = 0.430055$$

  Michael T. Heath  Scientific Computing 07 / 61

---

Example: Romberg Integration

- As another example, evaluate
  
  $$\int_{0}^{\pi/2} \sin(x) \, dx$$

- Using composite trapezoid rule, we have
  
  $$F(h) = a_0 + a_1 h^2 + O(h^4)$$

  so $p = 2$ and $r = 4$ in this instance.
- With $h = \pi/2$, $F(h) = F(\pi/2) = 0.785398$
- With $q = 2$, $F(h/2) = F(\pi/4) = 0.948059$

  Michael T. Heath  Scientific Computing 08 / 61

---

Romberg Integration

- Continued Richardson extrapolations using composite trapezoid rule with successively halved step sizes is called Romberg integration.
- It is capable of producing very high accuracy (up to limit imposed by arithmetic precision) for very smooth integrands.
- It is often implemented in automatic (though nonadaptive) fashion, with extrapolations continuing until change in successive values falls below specified error tolerance.

  Michael T. Heath  Scientific Computing 09 / 61