Numerical Integration

Sanzheng Qiao

Department of Computing and Software
McMaster University

February, 2014

Outline

1. Introduction
2. Rectangle Rule
3. Trapezoid Rule
4. Error Estimates
5. Simpson’s Rule
6. Richardson’s Extrapolation
7. Adaptive Quadrature
8. 2D Quadrature
9. Software Packages

Introduction

Another (better) term: quadrature

Problem: Given a function $f(x)$ on $x \in [a, b]$, calculate

$$I(f) = \int_a^b f(x)dx.$$

Introduction (cont.)

Partition

$$a = x_1 < x_2 < \cdots < x_{n+1} = b,$$

and denote $h_i = x_{i+1} - x_i$. Then

$$I(f) = \sum_{i=1}^{n} I_i \qquad I_i = \int_{x_i}^{x_{i+1}} f(x)dx$$

Quadrature rule: Approximation of I_i

Composite quadrature rule: Approximation of $I(f)$ as the sum of I_i

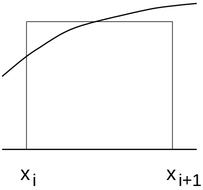
Rectangle rule

We use piecewise constant (degree zero polynomial) to approximate $f(x)$ on each $[x_i, x_{i+1}]$. The function is evaluated at the midpoint

$$y_i = \frac{x_i + x_{i+1}}{2}, \qquad i = 1, ..., n,$$

then the rectangle (quadrature) rule is:

$$I_i \approx h_i f(y_i)$$



Rectangle rule (cont.)

The composite rectangle rule is:

$$R(f) = \sum_{i=1}^{n} h_i f(y_i)$$

A weighted sum of function values.

Often the major computation is the evaluation of the function. Thus the complexity is measured by the number of function evaluations.

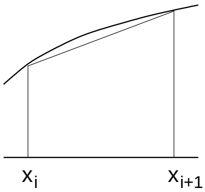
In the rectangle rule

Function evaluations: n. Evaluated at midpoints y_i.

Trapezoid rule

We use piecewise linear interpolation (degree one polynomial) to approximate $f(x)$ on each $[x_i, x_{i+1}]$. The function is evaluated at the endpoints:

$$I_i \approx h_i \frac{f(x_i) + f(x_{i+1})}{2}$$



Trapezoid rule (cont.)

Composite trapezoid rule:

$$\begin{aligned} T(f) &= \sum_{i=1}^{n} h_i \frac{f(x_i) + f(x_{i+1})}{2} \\ &= \frac{h_1}{2} f(x_1) + \frac{h_1 + h_2}{2} f(x_2) + \dots \\ &\quad + \frac{h_{n-1} + h_n}{2} f(x_n) + \frac{h_{n+1}}{2} f(x_{n+1}) \end{aligned}$$

A weighted sum of function values.

Function evaluations: $n+1$. Evaluated at endpoints x_i

Error in rectangle rule

Taylor expansion $f(x)$ about the midpoint $y_i = (x_i + x_{i+1})/2$:

$$f(x) = f(y_i) + \sum_{p=1}^{\infty} \frac{(x - y_i)^p}{p!} f^{(p)}(y_i).$$

Integrate the both sides and note that

$$\int_{x_i}^{x_{i+1}} (x - y_i)^p dx = \begin{cases} \frac{h_i^{p+1}}{(p+1)2^p} & \text{even p} \\ 0 & \text{odd p} \end{cases}$$

Error in rectangle rule (cont.)

Then

$$\begin{aligned} I_i &= \int_{x_i}^{x_{i+1}} f(x)dx \\ &= h_i f(y_i) + \frac{1}{24} h_i^3 f''(y_i) + \frac{1}{1920} h_i^5 f^{iv}(y_i) + \cdots \end{aligned}$$

When h_i is small, the error

$$\begin{aligned} & I(f) - R(f) \\ \approx \quad & \frac{1}{24} \sum_{i=1}^{n} h_i^3 f''(y_i) + \frac{1}{1920} \sum_{i=1}^{n} h_i^5 f^{iv}(y_i) \end{aligned}$$

For equal spacing, $h_i = h$, we have the error in the composite rectangle rule:

$$I(f) - R(f) \approx \frac{h^3}{24} \sum_{i=1}^{n} f''(y_i) + \frac{h^5}{1920} \sum_{i=1}^{n} f^{iv}(y_i)$$

Error in trapezoid rule

In order to make the error in the trapezoid rule comparable with that in the rectangle rule, we expand $f(x)$ at the midpoint y_i. Substituting $x = x_i$ and $x = x_{i+1}$ in the Taylor expansion, we have

$$\begin{aligned} f(x_i) &= f(y_i) + \sum_{p=1}^{\infty} (-1)^p \frac{h_i^p}{2^p p!} f^{(p)}(y_i) \\ f(x_{i+1}) &= f(y_i) + \sum_{p=1}^{\infty} \frac{h_i^p}{2^p p!} f^{(p)}(y_i) \end{aligned}$$

Thus

$$\frac{f(x_i) + f(x_{i+1})}{2} = f(y_i) + \frac{1}{8} h_i^2 f''(y_i) + \frac{1}{384} h_i^4 f^{iv}(y_i) + \cdots$$

Error in trapezoid rule (cont.)

Recall that in the case of rectangle rule, we had

$$\begin{aligned} I_i &= \int_{x_i}^{x_{i+1}} f(x)dx \\ &= h_i f(y_i) + \frac{1}{24} h_i^3 f''(y_i) + \frac{1}{1920} h_i^5 f^{iv}(y_i) + \cdots \end{aligned}$$

Combining the above two equations, we have

$$\begin{aligned} I_i &= \int_{x_i}^{x_{i+1}} f(x)dx \\ &= h_i \frac{f(x_i) + f(x_{i+1})}{2} - \frac{1}{12} h_i^3 f''(y_i) - \frac{1}{480} h_i^5 f^{iv}(y_i) + \cdots \end{aligned}$$

Then the error in the composite trapezoid rule is

$$I(f) - T(f) \approx -\frac{1}{12} \sum_{i=1}^{n} h_i^3 f''(y_i) - \frac{1}{480} \sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

Remarks

Compare

$$I(f) - R(f) \approx \frac{h^3}{24} \sum_{i=1}^{n} f''(y_i) + \frac{h^5}{1920} \sum_{i=1}^{n} f^{iv}(y_i)$$

and

$$I(f) - T(f) \approx -\frac{1}{12} \sum_{i=1}^{n} h_i^3 f''(y_i) - \frac{1}{480} \sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

Usually rectangle rule (degree zero approximation) is more accurate than trapezoid rule (degree one approximation). Surprised?

Remarks

Observe

$$I(f) - R(f) \approx \frac{h^3}{24} \sum_{i=1}^{n} f''(y_i) + \frac{h^5}{1920} \sum_{i=1}^{n} f^{iv}(y_i)$$

and

$$I(f) - T(f) \approx -\frac{1}{12} \sum_{i=1}^{n} h_i^3 f''(y_i) - \frac{1}{480} \sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

Using $I(f) - R(f) \approx \frac{1}{3}(T(f) - R(f))$, we can estimate the error in $R(f)$ using $T(f)$ and $R(f)$. Similarly, $I(f) - T(f) \approx \frac{2}{3}(R(f) - T(f))$ can be used to estimate the error in $T(f)$. (But they are approximations, it is possible that $R(f) - T(f) = 0$ whereas $I(f) - R(f) \neq 0$).

Remarks

Observe

$$I(f) - R(f) \approx \frac{h^3}{24} \sum_{i=1}^{n} f''(y_i) + \frac{h^5}{1920} \sum_{i=1}^{n} f^{iv}(y_i)$$

When each h_i is cut in half, $I(f) - R_{\frac{1}{2}}(f) \approx \frac{1}{4}(I(f) - R(f))$. (Why?) Similarly for the trapezoid rule. Doubling the number of panels in either the rectangle rule or the trapezoid rule, it can be expected to roughly quadruple the accuracy.

This can be used to estimate the error as well as improving the accuracy. (How?)

Example

Compute

$$\int_0^{\frac{\pi}{2}} \sin(x)dx$$

using the trapezoid rule.

m	QCTrap(sin, 0.000, 1.571, m)	error
3	0.9480594489685199	5.2e-2
5	0.9871158009727753	1.3e-2
9	0.9967851718861696	3.2e-3
17	0.9991966804850722	8.0e-4

where m is the number of points, that is, $m - 1$ is the number of intervals.

Simpson's rule

Recall the rectangle rule

$$R(f) = I(f) - \frac{1}{24}\sum_{i=1}^{n} h_i^3 f''(y_i) - \frac{1}{1920}\sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

and the trapezoid rule

$$T(f) = I(f) + \frac{1}{12}\sum_{i=1}^{n} h_i^3 f''(y_i) + \frac{1}{480}\sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

Combining the above two equations (canceling the $O(h_i^3)$ term), we get a more accurate method (Simpson's rule):

$$\begin{aligned} S(f) &= \frac{2}{3}R(f) + \frac{1}{3}T(f) \\ &= I(f) + \frac{1}{2880}\sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots \end{aligned}$$

Simpson's rule (cont.)

Simpson's rule:

$$I_i = \frac{2}{3} h_i f(\frac{x_i + x_{i+1}}{2}) + \frac{1}{3} h_i \frac{f(x_i) + f(x_{i+1})}{2}$$

Composite Simpson's rule:

$$S(f) = \sum_{i=1}^{n} \frac{1}{6} h_i \left[f(x_i) + 4f(\frac{x_i + x_{i+1}}{2}) + f(x_{i+1}) \right]$$

Function evaluations: $2n + 1$

Error

$$I(f) - S(f) = -\frac{1}{2880} \sum_{i=1}^{n} h_i^5 f^{iv}(y_i) + \cdots$$

Remarks

- Simpson's rule can also be derived by using piecewise quadratic (degree two) approximation.
- Actually, Simpson's rule is exact for cubic function (one extra order of accuracy), since the error term involves the fourth derivatives.
- Doubling the number of panels in Simpson's rule can be expected to reduce the error by roughly the factor of $1/16$.

A general technique: Richardson's extrapolation

Idea: Combining two approximations (e.g., $R(f)$ and $T(f)$) which have similar error terms to achieve a more accurate approximation (e.g., $S(f)$).

Example. Combining $S(f)$ and $S_{\frac{1}{2}}(f)$ to obtain an approximation which has error of order h_i^7. This gives the Romberg quadrature.

Question

What are the weights?

Answer: $\frac{16}{15}S_{\frac{1}{2}}(f) - \frac{1}{15}S(f)$

What is adaptive quadrature?

Given a predetermined tolerance ϵ, the algorithm automatically determines the panel sizes so that the computed approximation Q satisfies

$$\left| Q - \int_a^b f(x)dx \right| < \epsilon$$

Software interface: quad(fname, a, b, tol)

Why adaptive?

The algorithm uses large panel sizes for smooth parts and small panel sizes for the parts where the function changes rapidly. Thus the prescribed accuracy is attained at as small a cost in computing time. (Measured by the number of function evaluations.)

Basic idea

Compute two approximations (Simpson's rule):
one-panel formula

$$P_i = \frac{h_i}{6}\left[f(x_i) + 4f(x_i + \frac{h_i}{2}) + f(x_i + h_i)\right]$$

two-panel formula

$$Q_i = \frac{h_i}{12}\Big[\; f(x_i) + 4f(x_i + \frac{h_i}{4}) + 2f(x_i + \frac{h_i}{2}) + 4f(x_i + \frac{3h_i}{4}) + f(x_i + h_i)\Big]$$

Basic idea (cont.)

Note

- From P_i to Q_i, we need only two function evaluations $f(x_i + \frac{h_i}{4})$ and $f(x_i + \frac{3h_i}{4})$
- Q_i is the sum of two P's from two subintervals of length $h_i/2$

Compare P_i and Q_i to obtain an estimate of their accuracy.

$$I_i - P_i = c\, h_i^5 f^{iv}(x_i + \frac{h_i}{2}) + \cdots$$

$$I_i - Q_i = c \left(\frac{h_i}{2}\right)^5 \left[f^{iv}(x_i + \frac{h_i}{4}) + f^{iv}(x_i + \frac{3h_i}{4}) \right] + \cdots$$

Error estimation

Using the approximation

$$f^{iv}(x_i + \frac{h_i}{4}) + f^{iv}(x_i + \frac{3h_i}{4}) \approx 2f^{iv}(x_i + \frac{h_i}{2}),$$

we have

$$I_i - Q_i \approx 2c \left(\frac{h_i}{2}\right)^5 f^{iv}(x_i + \frac{h_i}{2}) + \cdots$$

Thus we have a relation between the errors in Q_i and P_i:

$$I_i - Q_i \approx \frac{1}{2^4}(I_i - P_i) + \cdots$$

Reformulate the above

$$I_i - Q_i \approx \frac{1}{2^4 - 1}(Q_i - P_i) + \cdots$$

Now the accuracy of Q_i is expressed in terms of $Q_i - P_i$

Scheme

Bisect each subinterval until

$$\frac{1}{2^4-1}|Q_i - P_i| \le \frac{h_i}{b-a}\epsilon$$

Then

$$\begin{aligned}
\left|\int_a^b f(x)dx - \sum_{i=1}^{n} Q_i\right| &\le \frac{1}{2^4-1}\sum_{i=1}^{n}|Q_i - P_i| \\
&\le \frac{\epsilon}{b-a}\sum_{i=1}^{n} h_i = \epsilon
\end{aligned}$$

function [I, err] = AdaptQuad(fname,a,b,tol,maxLev)

```
if maxLev==0
   too many levels of recursion, quit;
compute one-panel quadrature R1;
compute two-panel quadrature R2;
use R1 and R2 to estimate error in R2;
if the estimated error < tol
   return R2 and estimated error;
else
   [I1, err1] =
     AdaptQuad(fname,a,mid,tol/2,maxLev-1);
   [I2, err2] =
     AdaptQuad(fname,mid,b,tol/2,maxLev-1);
   I = I1 + I2;
   err = err1 + err2;
```

Example

Compute

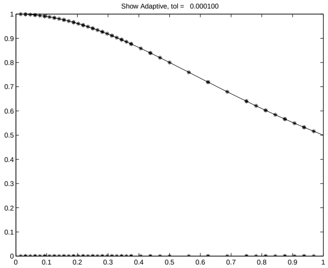
$$\frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx$$

using the adaptive rectangle rule.

`AdaptQRec('datan',0,1,0.0001,10)`: 0.785396

estimated error: 7.28×10^{-5}

actual error: 2.23×10^{-6}



2D quadrature

Consider a 2D integral

$$I = \int_a^b \int_c^d f(x, y) dy dx$$

and let

$$g(x) = \int_c^d f(x, y) dy.$$

Applying the composite trapezoid rule to

$$\int_a^b g(x) dx$$

we get the numerical integration

$$\sum_{i=1}^{m-1} \frac{g(x_i) + g(x_{i+1})}{2} h_x.$$

2D quadrature (cont.)

Written in vector form:

$$h_x w_x^{\mathrm{T}} \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_m) \end{bmatrix}$$

where $w_x^{\mathrm{T}} = [1/2, 1, ..., 1, 1/2]$.

2D quadrature (cont.)

Again, applying the composite trapezoid rule to each $g(x_i)$, we get

$$g(x_i) = \int_c^d f(x_i, y)dy \approx \sum_{j=1}^{n-1} \frac{f(x_i, y_j) + f(x_i, y_{j+1})}{2} h_y.$$

In vector form

$$g(x_i) \approx h_y[f(x_i, y_1), ..., f(x_i, y_n)]w_y$$

where $w_y = [1/2, 1, ..., 1, 1/2]^{\mathrm{T}}$.

2D quadrature (cont.)

Finally, we have the numerical integration, in matrix-vector form:

$$Q = h_x h_y w_x^{\mathrm{T}} F w_y$$

where

$$F = \begin{bmatrix} f(x_1, y_1) & \cdots & f(x_1, y_n) \\ \vdots & \ldots & \vdots \\ f(x_m, y_1) & \cdots & f(x_m, y_n) \end{bmatrix}.$$

Example

$$\int_{-2}^{2}\int_{-1}^{1} e^{-(x^2+2y^2)/4}\,dydx$$

$m = n$

Subintervals	Integral	Relative Time
2	4.39508052	1.00
4	4.93166539	0.97
8	5.06690648	1.00
16	5.10073164	1.62
32	5.10918854	1.75
64	5.11130280	4.13
128	5.11183136	13.21
256	5.11196350	44.88

Software packages

IMSL	qdag, qdags, twodq, qand
MATLAB	quad, quad1, dblquad
NAG	d01ajf, d01daf, d01fcf
Octave	quad, quadl, trapz

Summary

- Composite quadrature rules: Rectangle rule, trapezoid rule, Simpson's rule
- Richardson's extrapolation technique: Combining two quadrature rules with similar error terms to achieve a more accurate quadrature rule by canceling the leading error term; Combining one-panel and two-panel results to estimate errors
- Adaptive quadrature: By using error estimates, determine the panel sizes so that the computed approximation satisfies a predetermined tolerance
- 2D quadrature: Formulation of the problem

References

[1] George E. Forsyth and Michael A. Malcolm and Cleve B. Moler. Computer Methods for Mathematical Computations. Prentice-Hall, Inc., 1977.
Ch 5.