6 Numerical Integration

Compute $\int_a^b f(x)\,dx$, where $f(x)$ is a given function

6.1 Introduction

Numerical integration, also known as *quadrature*, is intrinsically a much more accurate procedure than numerical differentiation. Quadrature approximates the definite integral

$$\int_a^b f(x)\,dx$$

by the sum

$$I = \sum_{i=0}^{n} A_i f(x_i)$$

where the *nodal abscissas* x_i and *weights* A_i depend on the particular rule used for the quadrature. All rules of quadrature are derived from polynomial interpolation of the integrand. Therefore, they work best if $f(x)$ can be approximated by a polynomial.

Methods of numerical integration can be divided into two groups: Newton–Cotes formulas and Gaussian quadrature. Newton–Cotes formulas are characterized by equally spaced abscissas and include well-known methods such as the trapezoidal rule and Simpson's rule. They are most useful if $f(x)$ has already been computed at equal intervals or can be computed at low cost. Because Newton–Cotes formulas are based on local interpolation, they require only a piecewise fit to a polynomial.

In Gaussian quadrature, the locations of the abscissas are chosen to yield the best possible accuracy. Because Gaussian quadrature requires fewer evaluations of the integrand for a given level of precision, it is popular in cases where $f(x)$ is expensive to evaluate. Another advantage of Gaussian quadrature is ability to handle integrable singularities, enabling us to evaluate expressions such as

$$\int_0^1 \frac{g(x)}{\sqrt{1-x^2}}dx$$

provided that $g(x)$ is a well-behaved function.

6.2 Newton–Cotes Formulas

Consider the definite integral

$$\int_a^b f(x)\,dx \tag{6.1}$$

We divide the range of integration (a, b) into n equal intervals of length $h = (b-a)/n$, as shown in Fig. 6.1, and denote the abscissas of the resulting nodes by $x_0, x_1, \ldots, x_n$. Next, we approximate $f(x)$ by a polynomial of degree n that intersects all the nodes. Lagrange's form of this polynomial, Eq. (3.1a), is

$$P_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x)$$

where $\ell_i(x)$ are the cardinal functions defined in Eq. (3.1b). Therefore, an approximation to the integral in Eq. (6.1) is

$$I = \int_a^b P_n(x)dx = \sum_{i=0}^{n}\left[f(x_i)\int_a^b \ell_i(x)dx\right] = \sum_{i=0}^{n} A_i f(x_i) \tag{6.2a}$$

where

$$A_i = \int_a^b \ell_i(x)dx, \quad i = 0, 1, \ldots, n \tag{6.2b}$$

Equations (6.2) are the *Newton–Cotes formulas*. Classical examples of these formulas are the *trapezoidal rule* ($n = 1$), *Simpson's rule* ($n = 2$), and *3/8 Simpson's rule* ($n = 3$). The most important of these is the trapezoidal rule. It can be combined with Richardson extrapolation into an efficient algorithm known as *Romberg integration*, which makes the other classical rules somewhat redundant.

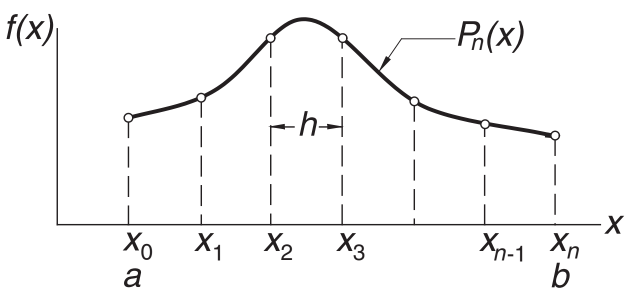


Figure 6.1. Polynomial approximation of $f(x)$.

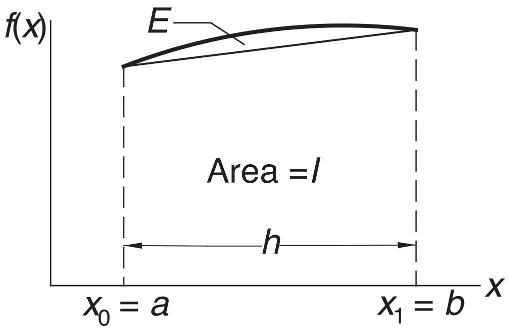


Figure 6.2. Trapezoidal rule.

Trapezoidal Rule

If $n = 1$ (one panel), as illustrated in Fig 6.2, we have $\ell_0 = (x - x_1)/(x_0 - x_1) = -(x - b)/h$. Therefore,

$$A_0 = \frac{1}{h}\int_a^b (x - b)\,dx = \frac{1}{2h}(b-a)^2 = \frac{h}{2}$$

Also, $\ell_1 = (x - x_0)/(x_1 - x_0) = (x - a)/h$, so that

$$A_1 = \frac{1}{h}\int_a^b (x - a)\,dx = \frac{1}{2h}(b-a)^2 = \frac{h}{2}$$

Substitution in Eq. (6.2a) yields

$$I = \left[f(a) + f(b)\right]\frac{h}{2} \tag{6.3}$$

which is known as the *trapezoidal rule*. It represents the area of the trapezoid in Fig. 6.2.

The error in the trapezoidal rule

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

is the area of the region between $f(x)$ and the straight-line interpolant, as indicated in Figure 6.2. It can be obtained by integrating the interpolation error in Eq. (3.3):

$$\begin{aligned} E &= \frac{1}{2!}\int_a^b (x - x_0)(x - x_1)f''(\xi)dx = \frac{1}{2}f''(\xi)\int_a^b (x-a)(x-b)dx \\ &= -\frac{1}{12}(b-a)^3 f''(\xi) = -\frac{h^3}{12}f''(\xi) \qquad (6.4) \end{aligned}$$

Composite Trapezoidal Rule

In practice the trapezoidal rule is applied in a piecewise fashion. Figure 6.3 shows the region (a, b) divided into n panels, each of width h. The function $f(x)$ to be integrated is approximated by a straight line in each panel. From the trapezoidal rule we obtain

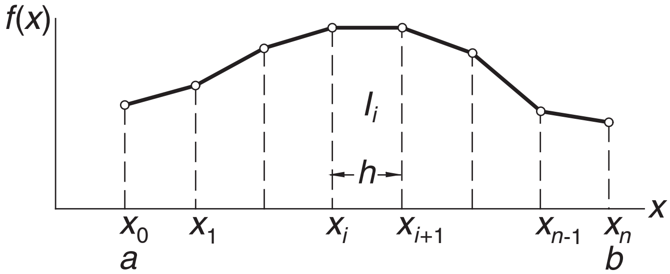


Figure 6.3. Composite trapezoidal rule.

for the approximate area of a typical (ith) panel

$$I_i = \left[f(x_i) + f(x_{i+1})\right]\frac{h}{2}$$

Hence, the total area, representing $\int_a^b f(x)\,dx$, is

$$I = \sum_{i=0}^{n-1} I_i = \left[f(x_0) + 2f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-1}) + f(x_n)\right]\frac{h}{2} \tag{6.5}$$

which is the *composite trapezoidal rule*.

The truncation error in the area of a panel is – see Eq. (6.4)

$$E_i = -\frac{h^3}{12}f''(\xi_i)$$

where ξ_i lies in (x_i, x_{i+1}). Hence, the truncation error in Eq. (6.5) is

$$E = \sum_{i=0}^{n-1} E_i = -\frac{h^3}{12}\sum_{i=0}^{n-1} f''(\xi_i) \tag{a}$$

But

$$\sum_{i=0}^{n-1} f''(\xi_i) = n\bar{f}''$$

where $\bar{f}''$ is the arithmetic mean of the second derivatives. If $f''(x)$ is continuous, there must be a point ξ in (a, b) at which $f''(\xi) = \bar{f}''$, enabling us to write

$$\sum_{i=0}^{n-1} f''(\xi_i) = nf''(\xi) = \frac{b-a}{h}f''(\xi)$$

Therefore, Eq. (a) becomes

$$E = -\frac{(b-a)h^2}{12}f''(\xi) \tag{6.6}$$

It would be incorrect to conclude from Eq. (6.6) that $E = ch^2$ (c being a constant), because $f''(\xi)$ is not entirely independent of h. A deeper analysis of the error[1] shows

[1] The analysis requires familiarity with the *Euler-Maclaurin summation formula*, which is covered in advanced texts.

that if $f(x)$ and its derivatives are finite in (a, b), then

$$E = c_1 h^2 + c_2 h^4 + c_3 h^6 + \dots \tag{6.7}$$

Recursive Trapezoidal Rule

Let I_k be the integral evaluated with the composite trapezoidal rule using 2^{k-1} panels. Note that if k is increased by 1, the number of panels is doubled. Using the notation

$$H = b - a$$

Equation (6.5) yields the following results for $k = 1$, 2, and 3.

$k = 1$ (1 panel):

$$I_1 = \left[f(a) + f(b)\right]\frac{H}{2} \tag{6.8}$$

$k = 2$ (2 panels):

$$I_2 = \left[f(a) + 2f\left(a + \frac{H}{2}\right) + f(b)\right]\frac{H}{4} = \frac{1}{2}I_1 + f\left(a + \frac{H}{2}\right)\frac{H}{2}$$

$k = 3$ (4 panels):

$$\begin{aligned} I_3 &= \left[f(a) + 2f\left(a + \frac{H}{4}\right) + 2f\left(a + \frac{H}{2}\right) + 2f\left(a + \frac{3H}{4}\right) + f(b)\right]\frac{H}{8} \\ &= \frac{1}{2}I_2 + \left[f\left(a + \frac{H}{4}\right) + f\left(a + \frac{3H}{4}\right)\right]\frac{H}{4} \end{aligned}$$

We can now see that for arbitrary $k > 1$ we have

$$I_k = \frac{1}{2}I_{k-1} + \frac{H}{2^{k-1}}\sum_{i=1}^{2^{k-2}} f\left[a + \frac{(2i-1)H}{2^{k-1}}\right], \quad k = 2, 3, \dots \tag{6.9a}$$

which is the *recursive trapezoidal rule*. Observe that the summation contains only the new nodes that were created when the number of panels was doubled. Therefore, the computation of the sequence $I_1, I_2, I_3, \dots, I_k$ from Eqs. (6.8) and (6.9) involves the same amount of algebra as the calculation of I_k directly from Eq. (6.5). The advantage of using the recursive trapezoidal rule is that it allows us to monitor convergence and terminate the process when the difference between I_{k-1} and I_k becomes sufficiently small. A form of Eq. (6.9a) that is easier to remember is

$$I(h) = \frac{1}{2}I(2h) + h\sum f(x_{\text{new}}) \tag{6.9b}$$

where $h = H/n$ is the width of each panel.

■ `trapezoid`

The function `trapezoid` computes I_k (`Inew`), given I_{k-1} (`Iold`) using Eqs. (6.8) and (6.9). We can compute $\int_a^b f(x)\,dx$ by calling `trapezoid` with $k = 1, 2, \dots$ until the desired precision is attained.

```
## module trapezoid
''' Inew = trapezoid(f,a,b,Iold,k).
    Recursive trapezoidal rule:
    Iold = Integral of f(x) from x = a to b computed by
    trapezoidal rule with 2^(k-1) panels.
    Inew = Same integral computed with 2^k panels.
'''
def trapezoid(f,a,b,Iold,k):
    if k == 1:Inew = (f(a) + f(b))*(b - a)/2.0
    else:
        n = 2**(k -2 )      # Number of new points
        h = (b - a)/n       # Spacing of new points
        x = a + h/2.0
        sum = 0.0
        for i in range(n):
            sum = sum + f(x)
            x = x + h
        Inew = (Iold + h*sum)/2.0
    return Inew
```

Simpson's Rules

Simpson's 1/3 rule can be obtained from the Newton–Cotes formulas with $n = 2$, that is, by passing a parabolic interpolant through three adjacent nodes, as shown in Fig.6.4. The area under the parabola, which represents an approximation of $\int_a^b f(x)\,dx$, is (see derivation in Example 6.1)

$$I = \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]\frac{h}{3} \tag{a}$$

To obtain the *composite Simpson's 1/3 rule*, the integration range (a, b) is divided into n panels (n even) of width $h = (b - a)/n$ each, as indicated in Fig. 6.5. Applying

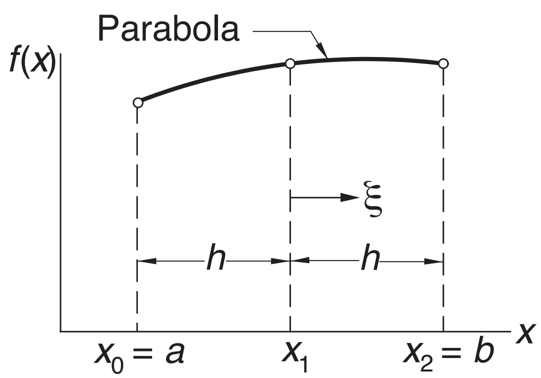


Figure 6.4. Simpson's 1/3 rule.

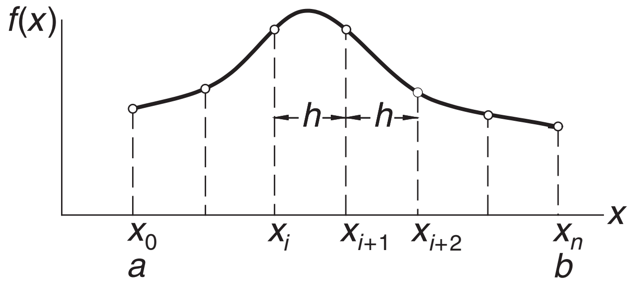


Figure 6.5. Composite Simpson's 1/3 rule.

Eq. (a) to two adjacent panels, we have

$$\int_{x_i}^{x_{i+2}} f(x)\,dx \approx \left[f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\right]\frac{h}{3} \tag{b}$$

Substituting Eq. (b) into

$$\int_a^b f(x)dx = \int_{x_0}^{x_m} f(x)\,dx = \sum_{i=0,2,\ldots}^{n}\left[\int_{x_i}^{x_{i+2}} f(x)dx\right]$$

yields

$$\int_a^b f(x)\,dx \approx I = [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots \tag{6.10}$$

$$\cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]\frac{h}{3}$$

The composite Simpson's 1/3 rule in Eq. (6.10) is perhaps the best-known method of numerical integration. Its reputation is somewhat undeserved, since the trapezoidal rule is more robust, and Romberg integration is more efficient.

The error in the composite Simpson's rule is

$$E = \frac{(b-a)h^4}{180} f^{(4)}(\xi) \tag{6.11}$$

from which we conclude that Eq. (6.10) is exact if $f(x)$ is a polynomial of degree 3 or less.

Simpson's 1/3 rule requires the number of panels n to be even. If this condition is not satisfied, we can integrate over the first (or last) three panels with *Simpson's 3/8 rule*:

$$I = \left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]\frac{3h}{8} \tag{6.12}$$

and use Simpson's 1/3 rule for the remaining panels. The error in Eq. (6.12) is of the same order as in Eq. (6.10).

EXAMPLE 6.1

Derive Simpson's 1/3 rule from the Newton–Cotes formulas.

Solution Referring to Figure 6.4, Simpson's 1/3 rule uses three nodes located at $x_0 = a$, $x_1 = \left(a+b\right)/2$, and $x_2 = b$. The spacing of the nodes is $h = (b-a)/2$. The cardinal functions of Lagrange's three-point interpolation are – see Section 3.2

$$\ell_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \qquad \ell_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$

$$\ell_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

The integration of these functions is easier if we introduce the variable ξ with origin at x_1. Then the coordinates of the nodes are $\xi_0 = -h$, $\xi_1 = 0$, $\xi_2 = h$, and Eq. (6.2b) becomes $A_i = \int_a^b \ell_i(x) = \int_{-h}^{h} \ell_i(\xi)d\xi$. Therefore,

$$A_0 = \int_{-h}^{h} \frac{(\xi-0)(\xi-h)}{(-h)(-2h)} d\xi = \frac{1}{2h^2}\int_{-h}^{h}(\xi^2 - h\xi)d\xi = \frac{h}{3}$$

$$A_1 = \int_{-h}^{h} \frac{(\xi+h)(\xi-h)}{(h)(-h)} d\xi = -\frac{1}{h^2}\int_{-h}^{h}(\xi^2 - h^2)d\xi = \frac{4h}{3}$$

$$A_2 = \int_{-h}^{h} \frac{(\xi+h)(\xi-0)}{(2h)(h)} d\xi = \frac{1}{2h^2}\int_{-h}^{h}(\xi^2 + h\xi)d\xi = \frac{h}{3}$$

Equation (6.2a) then yields

$$I = \sum_{i=0}^{2} A_i f(x_i) = \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]\frac{h}{3}$$

which is Simpson's 1/3 rule.

EXAMPLE 6.2

Evaluate the bounds on $\int_0^\pi \sin(x)\,dx$ with the composite trapezoidal rule using (1) eight panels and (2) 16 panels.

Solution of Part (1) With eight panels there are nine nodes spaced at $h = \pi/8$. The abscissas of the nodes are $x_i = i\pi/8$, $i = 0, 1, \ldots, 8$. From Eq. (6.5) we get

$$I = \left[\sin 0 + 2\sum_{i=1}^{7} \sin\frac{i\pi}{8} + \sin\pi\right]\frac{\pi}{16} = 1.97423$$

The error is given by Eq. (6.6):

$$E = -\frac{(b-a)h^2}{12}f''(\xi) = -\frac{(\pi-0)(\pi/8)^2}{12}(-\sin\xi) = \frac{\pi^3}{768}\sin\xi$$

where $0 < \xi < \pi$. Because we do not know the value of ξ, we cannot evaluate E, but we can determine its bounds:

$$E_{\min} = \frac{\pi^3}{768}\sin(0) = 0 \qquad E_{\max} = \frac{\pi^3}{768}\sin\frac{\pi}{2} = 0.040\,37$$

Therefore, $I + E_{\min} < \int_0^{\pi} \sin(x)\,dx < I + E_{\max}$, or

$$1.974\,23 < \int_0^{\pi} \sin(x)\,dx < 2.014\,60$$

The exact integral is, of course, 2.

Solution of Part (2) The new nodes created by doubling of the panels are located at the midpoints of the old panels. Their abscissas are

$$x_j = \pi/16 + j\pi/8 = (1 + 2j)\pi/16, \quad j = 0, 1, \ldots, 7$$

Using the recursive trapezoidal rule in Eq. (6.9b), we get

$$I = \frac{1.974\,23}{2} + \frac{\pi}{16}\sum_{j=0}^{7}\sin\frac{(1+2j)\pi}{16} = 1.993\,58$$

and the bounds on the error become (note that E is quartered when h is halved) $E_{\min} = 0$, $E_{\max} = 0.040\,37/4 = 0.010\,09$. Hence,

$$1.993\,58 < \int_0^{\pi} \sin(x)\,dx < 2.003\,67$$

EXAMPLE 6.3

Estimate $\int_0^{2.5} f(x)\,dx$ from the data

x	0	0.5	1.0	1.5	2.0	2.5
$f(x)$	1.5000	2.0000	2.0000	1.6364	1.2500	0.9565

Solution We use Simpson's rules because they are more accurate than the trapezoidal rule. Because the number of panels is odd, we compute the integral over the first three panels by Simpson's 3/8 rule, and use the 1/3 rule for the last two panels:

$$\begin{aligned} I &= \left[f(0) + 3f(0.5) + 3f(1.0) + f(1.5)\right]\frac{3(0.5)}{8} \\ &\quad + \left[f(1.5) + 4f(2.0) + f(2.5)\right]\frac{0.5}{3} \\ &= 2.8381 + 1.2655 = 4.1036 \end{aligned}$$

EXAMPLE 6.4

Use the recursive trapezoidal rule to evaluate $\int_0^{\pi} \sqrt{x}\cos x\,dx$ to six decimal places. How many panels are needed to achieve this result?

Solution The program listed here utilizes the function `trapezoid`.

```
#!/usr/bin/python
## example6_4
from math import sqrt,cos,pi
from trapezoid import *
```

```
def f(x): return sqrt(x)*cos(x)

Iold = 0.0
for k in range(1,21):
    Inew = trapezoid(f,0.0,pi,Iold,k)
    if (k > 1) and (abs(Inew - Iold)) < 1.0e-6: break
    Iold = Inew
print ''Integral ='',Inew
print ''nPanels ='',2**(k-1)
raw_input(''\nPress return to exit''))
```

The output from the program is:

```
Integral = -0.894831664853
nPanels = 32768
```

Hence, $\int_0^\pi \sqrt{x}\cos x\,dx = -0.894\,832$, requiring 32,768 panels. The slow convergence is the result of all the derivatives of $f(x)$ being singular at $x = 0$. Consequently, the error does not behave as shown in Eq. (6.7): $E = c_1 h^2 + c_2 h^4 + \cdots$, but is unpredictable. Difficulties of this nature can often be remedied by a change in variable. In this case, we introduce $t = \sqrt{x}$ so that $dt = dx/(2\sqrt{x}) = dx/(2t)$, or $dx = 2t\,dt$. Thus,

$$\int_0^\pi \sqrt{x}\cos x\,dx = \int_0^{\sqrt{\pi}} 2t^2 \cos t^2\,dt$$

Evaluation of the integral on the right-hand side was completed with 4096 panels.

6.3 Romberg Integration

Romberg integration combines the trapezoidal rule with Richardson extrapolation (see Section 5.3). Let us first introduce the notation

$$R_{i,1} = I_i$$

where, as before, I_i represents the approximate value of $\int_a^b f(x)dx$ computed by the recursive trapezoidal rule using 2^{i-1} panels. Recall that the error in this approximation is $E = c_1 h^2 + c_2 h^4 + \ldots$, where

$$h = \frac{b-a}{2^{i-1}}$$

is the width of a panel.

Romberg integration starts with the computation of $R_{1,1} = I_1$ (one panel) and $R_{2,1} = I_2$ (two panels) from the trapezoidal rule. The leading error term $c_1 h^2$ is then eliminated by Richardson extrapolation. Using $p = 2$ (the exponent in the leading error term) in Eq. (5.9) and denoting the result by $R_{2,2}$, we obtain

$$R_{2,2} = \frac{2^2 R_{2,1} - R_{1,1}}{2^{2-1}} = \frac{4}{3} R_{2,1} - \frac{1}{3} R_{1,1} \tag{a}$$

It is convenient to store the results in an array of the form

$$\begin{bmatrix} R_{1,1} & \\ R_{2,1} & R_{2,2} \end{bmatrix}$$

The next step is to calculate $R_{3,1} = I_3$ (four panels) and repeat the Richardson extrapolation with $R_{2,1}$ and $R_{3,1}$, storing the result as $R_{3,2}$:

$$R_{3,2} = \frac{4}{3} R_{3,1} - \frac{1}{3} R_{2,1} \tag{b}$$

The elements of array **R** calculated so far are

$$\begin{bmatrix} R_{1,1} & \\ R_{2,1} & R_{2,2} \\ R_{3,1} & R_{3,2} \end{bmatrix}$$

Both elements of the second column have an error of the form $c_2 h^4$, which can also be eliminated with Richardson extrapolation. Using $p = 4$ in Eq. (5.9), we get

$$R_{3,3} = \frac{2^4 R_{3,2} - R_{2,2}}{2^{4-1}} = \frac{16}{15} R_{3,2} - \frac{1}{15} R_{2,2} \tag{c}$$

This result has an error of $\mathcal{O}(h^6)$. The array has now expanded to

$$\begin{bmatrix} R_{1,1} & & \\ R_{2,1} & R_{2,2} & \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix}$$

After another round of calculations we get

$$\begin{bmatrix} R_{1,1} & & & \\ R_{2,1} & R_{2,2} & & \\ R_{3,1} & R_{3.2} & R_{3,3} & \\ R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} \end{bmatrix}$$

where the error in $R_{4,4}$ is $\mathcal{O}(h^8)$. Note that the most accurate estimate of the integral is always the last diagonal term of the array. This process is continued until the difference between two successive diagonal terms becomes sufficiently small. The general extrapolation formula used in this scheme is

$$R_{i,j} = \frac{4^{j-1} R_{i,j-1} - R_{i-1,j-1}}{4^{j-1} - 1}, \quad i > 1, \quad j = 2, 3, \ldots, i \tag{6.13a}$$

A pictorial representation of Eq. (6.13a) is

$$\begin{array}{ccccc} \boxed{R_{i-1,j-1}} & & & & \\ & \searrow & & & \\ & & \alpha & & \\ & & & \searrow & \\ \boxed{R_{i,j-1}} & \rightarrow & \beta & \rightarrow & \boxed{R_{i,j}} \end{array} \tag{6.13b}$$

where the multipliers α and β depend on j in the following manner:

j	2	3	4	5	6
α	$-1/3$	$-1/15$	$-1/63$	$-1/255$	$-1/1023$
β	$4/3$	$16/15$	$64/63$	$256/255$	$1024/1023$

(6.13c)

The triangular array is convenient for hand computations, but computer implementation of the Romberg algorithm can be carried out within a one-dimensional array $\mathbf{R}'$. After the first extrapolation – see Eq. (a) – $R_{1,1}$ is never used again, so it can be replaced with $R_{2,2}$. As a result, we have the array

$$\begin{bmatrix} R'_1 = R_{2,2} \\ R'_2 = R_{2,1} \end{bmatrix}$$

In the second extrapolation round, defined by Eqs. (b) and (c), $R_{3,2}$ overwrites $R_{2,1}$ and $R_{3,3}$ replaces $R_{2,2}$, so the array contains

$$\begin{bmatrix} R'_1 = R_{3,3} \\ R'_2 = R_{3,2} \\ R'_3 = R_{3,1} \end{bmatrix}$$

and so on. In this manner, R'_1 always contains the best current result. The extrapolation formula for the kth round is

$$R'_j = \frac{4^{k-j} R'_{j+1} - R'_j}{4^{k-j} - 1}, \quad j = k-1, k-2, \ldots, 1 \tag{6.14}$$

■ romberg

The algorithm for Romberg integration is implemented in the function `romberg`. It returns the integral and the number of panels used. Richardson's extrapolation is carried out by the subfunction `richardson`.

```
## module romberg
''' I,nPanels = romberg(f,a,b,tol=1.0e-6).
    Romberg intergration of f(x) from x = a to b.
    Returns the integral and the number of panels used.
'''
from numpy import zeros
from trapezoid import *
```

```
def romberg(f,a,b,tol=1.0e-6):

    def richardson(r,k):
        for j in range(k-1,0,-1):
            const = 4.0**(k-j)
            r[j] = (const*r[j+1] - r[j])/(const - 1.0)
        return r

    r = zeros(21)
    r[1] = trapezoid(f,a,b,0.0,1)
    r_old = r[1]
    for k in range(2,21):
        r[k] = trapezoid(f,a,b,r[k-1],k)
        r = richardson(r,k)
        if abs(r[1]-r_old) < tol*max(abs(r[1]),1.0):
            return r[1],2**(k-1)
        r_old = r[1]
    print "Romberg quadrature did not converge"
```

EXAMPLE 6.5

Show that $R_{k,2}$ in Romberg integration is identical to the composite Simpson's 1/3 rule in Eq. (6.10) with 2^{k-1} panels.

Solution Recall that in Romberg integration $R_{k,1} = I_k$ denoted the approximate integral obtained by the composite trapezoidal rule with $n = 2^{k-1}$ panels. Denoting the abscissas of the nodes by $x_0, x_1, \ldots, x_n$, we have from the composite trapezoidal rule in Eq. (6.5)

$$R_{k,1} = I_k = \left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \frac{h}{2}$$

When we halve the number of panels (panel width $2h$), only the even-numbered abscissas enter the composite trapezoidal rule, yielding

$$R_{k-1,1} = I_{k-1} = \left[f(x_0) + 2\sum_{i=2,4,\ldots}^{n-2} f(x_i) + f(x_n) \right] h$$

Applying Richardson extrapolation yields

$$\begin{aligned} R_{k,2} &= \frac{4}{3} R_{k,1} - \frac{1}{3} R_{k-1,1} \\ &= \left[\frac{1}{3} f(x_0) + \frac{4}{3}\sum_{i=1,3,\ldots}^{n-1} f(x_i) + \frac{2}{3}\sum_{i=2,4,\ldots}^{n-2} f(x_i) + \frac{1}{3} f(x_n) \right] h \end{aligned}$$

which agrees with Eq. (6.10).

EXAMPLE 6.6

Use Romberg integration to evaluate $\int_0^{\pi} f(x)\,dx$, where $f(x) = \sin x$. Work with four decimal places.

Solution From the recursive trapezoidal rule in Eq. (6.9b) we get

$$
\begin{aligned}
R_{1,1} &= I(\pi) = \frac{\pi}{2}\left[f(0) + f(\pi)\right] = 0 \\
R_{2,1} &= I(\pi/2) = \frac{1}{2}I(\pi) + \frac{\pi}{2}f(\pi/2) = 1.5708 \\
R_{3,1} &= I(\pi/4) = \frac{1}{2}I(\pi/2) + \frac{\pi}{4}\left[f(\pi/4) + f(3\pi/4)\right] = 1.8961 \\
R_{4,1} &= I(\pi/8) = \frac{1}{2}I(\pi/4) + \frac{\pi}{8}\left[f(\pi/8) + f(3\pi/8) + f(5\pi/8) + f(7\pi/8)\right] \\
&= 1.9742
\end{aligned}
$$

Using the extrapolation formulas in Eqs. (6.13), we can now construct the following table:

$$
\begin{bmatrix}
R_{1,1} & & & \\
R_{2,1} & R_{2,2} & & \\
R_{3,1} & R_{3.2} & R_{3,3} & \\
R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4}
\end{bmatrix}
=
\begin{bmatrix}
0 & & & \\
1.5708 & 2.0944 & & \\
1.8961 & 2.0046 & 1.9986 & \\
1.9742 & 2.0003 & 2.0000 & 2.0000
\end{bmatrix}
$$

It appears that the procedure has converged. Therefore, $\int_0^{\pi} \sin x\,dx = R_{4,4} = 2.0000$, which is, of course, the correct result.

EXAMPLE 6.7

Use Romberg integration to evaluate $\int_0^{\sqrt{\pi}} 2x^2 \cos x^2\,dx$ and compare the results with Example 6.4.

Solution

```
#!/usr/bin/python
## example6_7
from math import cos,sqrt,pi
from romberg import *

def f(x): return 2.0*(x**2)*cos(x**2)

I,n = romberg(f,trapezoid,0,sqrt(pi))
print ''Integral ='',I
print ''nPanels ='',n
raw_input(''\nPress return to exit'')
```

The results of running the program are:

```
Integral = -0.894831469504
nPanels = 64
```

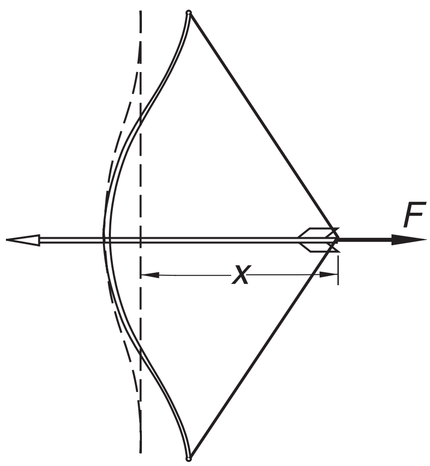
It is clear that Romberg integration is considerably more efficient than the trapezoidal rule – it required 64 panels as compared to 4096 panels for the trapezoidal rule in Example 6.4.

PROBLEM SET 6.1

1. Use the recursive trapezoidal rule to evaluate $\int_0^{\pi/4} \ln(1 + \tan x)dx$. Explain the results.
2. The table shows the power P supplied to the driving wheels of a car as a function of the speed v. If the mass of the car is $m = 2000$ kg, determine the time Δt it takes for the car to accelerate from 1 m/s to 6 m/s. Use the trapezoidal rule for integration. *Hint*: $\Delta t = m \int_{1s}^{6s} (v/P)\, dv$, which can be derived from Newton's law $F = m(dv/dt)$ and the definition of power $P = Fv$.

v (m/s)	0	1.0	1.8	2.4	3.5	4.4	5.1	6.0
P (kW)	0	4.7	12.2	19.0	31.8	40.1	43.8	43.2

3. Evaluate $\int_{-1}^{1} \cos(2\cos^{-1} x)dx$ with Simpson's 1/3 rule using 2, 4, and 6 panels. Explain the results.
4. Determine $\int_1^{\infty} (1 + x^4)^{-1} dx$ with the trapezoidal rule using five panels and compare the result with the "exact" integral 0.243 75. *Hint*: use the transformation $x^3 = 1/t$.



5. The following table gives the pull F of the bow as a function of the draw x. If the bow is drawn 0.5 m, determine the speed of the 0.075-kg arrow when it leaves the bow. *Hint*: the kinetic energy of the arrow equals the work done in drawing the bow; that is, $mv^2/2 = \int_0^{0.5\text{m}} F\, dx$.

x (m)	0.00	0.05	0.10	0.15	0.20	0.25
F (N)	0	37	71	104	134	161
x (m)	0.30	0.35	0.40	0.45	0.50	
F (N)	185	207	225	239	250	

6. Evaluate $\int_0^2 \left(x^5 + 3x^3 - 2\right) dx$ by Romberg integration.
7. Estimate $\int_0^\pi f(x)\, dx$ as accurately as possible, where $f(x)$ is defined by the data

x	0	$\pi/4$	$\pi/2$	$3\pi/4$	π
$f(x)$	1.0000	0.3431	0.2500	0.3431	1.0000

8. Evaluate

$$\int_0^1 \frac{\sin x}{\sqrt{x}}\, dx$$

with Romberg integration. *Hint*: use transformation of variables to eliminate the singularity at $x = 0$.
9. Newton–Cotes formulas for evaluating $\int_a^b f(x)\, dx$ were based on polynomial approximations of $f(x)$. Show that if $y = f(x)$ is approximated by a natural cubic spline with evenly spaced knots at $x_0, x_1, \ldots, x_n$, the quadrature formula becomes

$$I = \frac{h}{2}\left(y_0 + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n\right) - \frac{h^3}{24}\left(k_0 + 2k_1 + k_2 + \cdots + 2k_{n-1} + k_n\right)$$

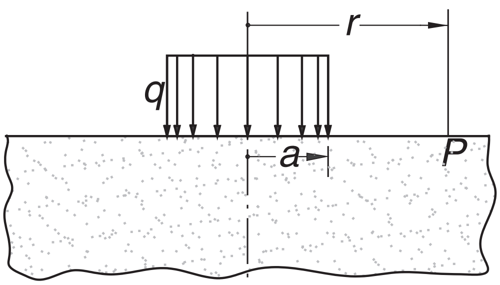
where h is the distance between the knots and $k_i = y_i''$. Note that the first part is the composite trapezoidal rule; the second part may be viewed as a "correction" for curvature.
10. ■ Evaluate

$$\int_0^{\pi/4} \frac{dx}{\sqrt{\sin x}}$$

with Romberg integration. *Hint*: use the transformation $\sin x = t^2$.
11. ■ The period of a simple pendulum of length L is $\tau = 4\sqrt{L/g}\,h(\theta_0)$, where g is the gravitational acceleration, θ_0 represents the angular amplitude, and

$$h(\theta_0) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - \sin^2(\theta_0/2)\sin^2\theta}}$$

Compute $h(15^\circ)$, $h(30^\circ)$, and $h(45^\circ)$ and compare these values with $h(0) = \pi/2$ (the approximation used for small amplitudes).
12. ■

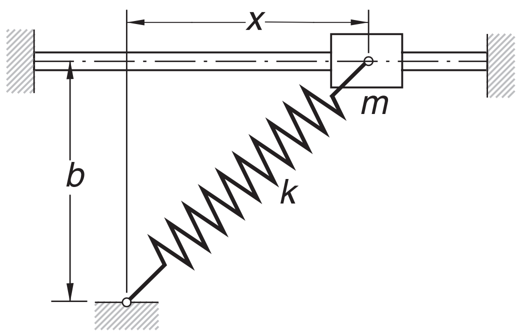


The figure shows an elastic half-space that carries uniform loading of intensity q over a circular area of radius a. The vertical displacement of the surface at point P can be shown to be

$$w(r) = w_0 \int_0^{\pi/2} \frac{\cos^2\theta}{\sqrt{(r/a)^2 - \sin^2\theta}} d\theta, \qquad r \geq a$$

where w_0 is the displacement at $r = a$. Use numerical integration to determine w/w_0 at $r = 2a$.

13. ■



The mass m is attached to a spring of free length b and stiffness k. The coefficient of friction between the mass and the horizontal rod is μ. The acceleration of the mass can be shown to be (you may wish to prove this) $\ddot{x} = -f(x)$, where

$$f(x) = \mu g + \frac{k}{m}(\mu b + x)\left(1 - \frac{b}{\sqrt{b^2 + x^2}}\right)$$

If the mass is released from rest at $x = b$, its speed at $x = 0$ is given by

$$v_0 = \sqrt{2\int_0^b f(x)dx}$$

Compute v_0 by numerical integration using the data $m = 0.8$ kg, $b = 0.4$ m, $\mu = 0.3$, $k = 80$ N/m, and $g = 9.81$ m/s^2.

14. ■ Debye's formula for the heat capacity C_V of a solid is $C_V = 9Nkg(u)$, where

$$g(u) = u^3 \int_0^{1/u} \frac{x^4 e^x}{(e^x - 1)^2} dx$$

The terms in this equation are

$N =$ number of particles in the solid

$k =$ Boltzmann constant

$u = T/\Theta_D$

$T =$ absolute temperature

$\Theta_D =$ Debye temperature

Compute $g(u)$ from $u = 0$ to 1.0 in intervals of 0.05 and plot the results.

15. ■ A power spike in an electric circuit results in the current

$$i(t) = i_0 e^{-t/t_0} \sin(2t/t_0)$$

across a resistor. The energy E dissipated by the resistor is

$$E = \int_0^\infty R\left[i(t)\right]^2 dt$$

Find E using the data $i_0 = 100\,\text{A}$, $R = 0.5\,\Omega$, and $t_0 = 0.01$ s.

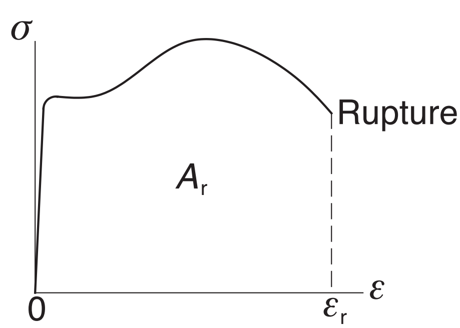
16. ■ An alternating electric current is described by

$$i(t) = i_0\left(\sin\frac{\pi t}{t_0} - \beta\sin\frac{2\pi t}{t_0}\right)$$

where $i_0 = 1$ A, $t_0 = 0.05$ s, and $\beta = 0.2$. Compute the root-mean-square current, defined as

$$i_{\text{rms}} = \sqrt{\frac{1}{t_0}\int_0^{t_0} i^2(t)\,dt}$$

17. ■ (a) Derive the composite trapezoidal rule for unevenly spaced data. (b) Consider the stress–strain diagram obtained from a uniaxial tension test.



The area under the diagram is

$$A_r = \int_{\varepsilon=0}^{\varepsilon_r} \sigma\,d\varepsilon$$

where ε_r is the strain at rupture. This area represents the work that must be performed on a unit volume of the test specimen in order to cause rupture; it is called the *modulus of toughness*. Use the result of Part (a) to estimate the modulus of toughness for nickel steel from the following test data:

σ (MPa)	ε
586	0.001
662	0.025
765	0.045
841	0.068
814	0.089
122	0.122
150	0.150

Note that the spacing of data is uneven.

6.4 Gaussian Integration

Gaussian Integration Formulas

We found that the Newton–Cotes formulas for approximating $\int_a^b f(x)dx$ work best if $f(x)$ is a smooth function, such as a polynomial. This is also true for Gaussian quadrature. However, Gaussian formulas are also good at estimating integrals of the form

$$\int_a^b w(x)f(x)dx \tag{6.15}$$

where $w(x)$, called the *weighting function*, can contain singularities, as long as they are integrable. An example of such an integral is $\int_0^1 (1+x^2)\ln x\,dx$. Sometimes infinite limits, as in $\int_0^\infty e^{-x}\sin x\,dx$, can also be accommodated.

Gaussian integration formulas have the same form as the Newton–Cotes rules,

$$I = \sum_{i=0}^{n} A_i f(x_i) \tag{6.16}$$

where, as before, I represents the approximation to the integral in Eq. (6.15). The difference lies in the way that the weights A_i and nodal abscissas x_i are determined. In Newton–Cotes integration the nodes were evenly spaced in (a, b), that is, their locations were predetermined. In Gaussian quadrature the nodes and weights are chosen so that Eq. (6.16) yields the exact integral if $f(x)$ is a polynomial of degree $2n+1$ or less, that is,

$$\int_a^b w(x)P_m(x)dx = \sum_{i=0}^{n} A_i P_m(x_i), \quad m \le 2n+1 \tag{6.17}$$

One way of determining the weights and abscissas is to substitute $P_0(x) = 1$, $P_1(x) = x, \ldots, P_{2n+1}(x) = x^{2n+1}$ in Eq. (6.17) and solve the resulting $2n+2$ equations

$$\int_a^b w(x)x^j dx = \sum_{i=0}^{n} A_i x_i^j, \quad j = 0, 1, \ldots, 2n+1$$

for the unknowns A_i and x_i.

As an illustration, let $w(x) = e^{-x}$, $a = 0$, $b = \infty$, and $n = 1$. The four equations determining x_0, x_1, A_0, and A_1 are

$$\int_0^{\infty} e^{-x} dx = A_0 + A_1$$

$$\int_0^{1} e^{-x} x\, dx = A_0 x_0 + A_1 x_1$$

$$\int_0^{1} e^{-x} x^2 dx = A_0 x_0^2 + A_1 x_1^2$$

$$\int_0^{1} e^{-x} x^3 dx = A_0 x_0^3 + A_1 x_1^3$$

After evaluating the integrals, we get

$$A_0 + A_1 = 1$$

$$A_0 x_0 + A_1 x_1 = 1$$

$$A_0 x_0^2 + A_1 x_1^2 = 2$$

$$A_0 x_0^3 + A_1 x_1^3 = 6$$

The solution is

$$x_0 = 2 - \sqrt{2} \qquad A_0 = \frac{\sqrt{2}+1}{2\sqrt{2}}$$

$$x_1 = 2 + \sqrt{2} \qquad A_1 = \frac{\sqrt{2}-1}{2\sqrt{2}}$$

so that the integration formula becomes

$$\int_0^{\infty} e^{-x} f(x) dx \approx \frac{1}{2\sqrt{2}} \left[(\sqrt{2}+1)\, f\left(2-\sqrt{2}\right) + (\sqrt{2}-1)\, f\left(2+\sqrt{2}\right) \right]$$

Because of the nonlinearity of the equations, this approach will not work well for large n. Practical methods of finding x_i and A_i require some knowledge of orthogonal polynomials and their relationship to Gaussian quadrature. There are, however, several "classical" Gaussian integration formulas for which the abscissas and weights have been computed with great precision and tabulated. These formulas can be used without knowing the theory behind them, because all one needs for Gaussian integration are the values of x_i and A_i. If you do not intend to venture outside the classical formulas, you can skip the next two topics of this article.

*Orthogonal Polynomials

Orthogonal polynomials are employed in many areas of mathematics and numerical analysis. They have been studied thoroughly and many of their properties are known. What follows is a very small compendium of a large topic.

The polynomials $\varphi_n(x)$, $n = 0, 1, 2, \ldots$ (n is the degree of the polynomial) are said to form an *orthogonal set* in the interval (a, b) with respect to the *weighting*

Name	Symbol	a	b	$w(x)$	$\int_a^b w(x)\left[\varphi_n(x)\right]^2 dx$
Legendre	$p_n(x)$	-1	1	1	$2/(2n+1)$
Chebyshev	$T_n(x)$	-1	1	$(1-x^2)^{-1/2}$	$\pi/2 \quad (n>0)$
Laguerre	$L_n(x)$	0	∞	e^{-x}	1
Hermite	$H_n(x)$	$-\infty$	∞	e^{-x^2}	$\sqrt{\pi}2^n n!$

Table 6.1

function $w(x)$ if

$$\int_a^b w(x)\varphi_m(x)\varphi_n(x)dx = 0, \quad m \neq n \tag{6.18}$$

The set is determined, except for a constant factor, by the choice of the weighting function and the limits of integration. That is, each set of orthogonal polynomials is associated with certain $w(x)$, a, and b. The constant factor is specified by standardization. Some of the classical orthogonal polynomials, named after well-known mathematicians, are listed in Table 6.1. The last column in the table shows the standardization used.

Orthogonal polynomials obey recurrence relations of the form

$$a_n\varphi_{n+1}(x) = (b_n + c_n x)\varphi_n(x) - d_n\varphi_{n-1}(x) \tag{6.19}$$

If the first two polynomials of the set are known, the other members of the set can be computed from Eq. (6.19). The coefficients in the recurrence formula, together with $\varphi_0(x)$ and $\varphi_1(x)$, are given in Table 6.2.

The classical orthogonal polynomials are also obtainable from the formulas

$$\begin{aligned}
p_n(x) &= \frac{(-1)^n}{2^n n!}\frac{d^n}{dx^n}\left[\left(1-x^2\right)^n\right] \\
T_n(x) &= \cos(n\cos^{-1}x), \quad n>0 \\
L_n(x) &= \frac{e^x}{n!}\frac{d^n}{dx^n}\left(x^n e^{-x}\right) \\
H_n(x) &= (-1)^n e^{x^2}\frac{d^n}{dx^n}(e^{-x^2})
\end{aligned} \tag{6.20}$$

Name	$\varphi_0(x)$	$\varphi_1(x)$	a_n	b_n	c_n	d_n
Legendre	1	x	$n+1$	0	$2n+1$	n
Chebyshev	1	x	1	0	2	1
Laguerre	1	$1-x$	$n+1$	$2n+1$	-1	n
Hermite	1	$2x$	1	0	2	2

Table 6.2

and their derivatives can be calculated from

$$
\begin{aligned}
(1-x^2)\,p_n'(x) &= n\left[-xp_n(x) + p_{n-1}(x)\right] \\
(1-x^2)\,T_n'(x) &= n\left[-x\,T_n(x) + n\,T_{n-1}(x)\right] \\
x\,L_n'(x) &= n\left[L_n(x) - L_{n-1}(x)\right] \\
H_n'(x) &= 2n\,H_{n-1}(x)
\end{aligned}
\tag{6.21}
$$

Other properties of orthogonal polynomials that have relevance to Gaussian integration are:

- $\varphi_n(x)$ has n real, distinct zeroes in the interval (a, b).
- The zeroes of $\varphi_n(x)$ lie between the zeroes of $\varphi_{n+1}(x)$.
- Any polynomial $P_n(x)$ of degree n can be expressed in the form

$$P_n(x) = \sum_{i=0}^{n} c_i \varphi_i(x) \tag{6.22}$$

- It follows from Eq. (6.22) and the orthogonality property in Eq. (6.18) that

$$\int_a^b w(x)\,P_n(x)\varphi_{n+m}(x)\,dx = 0, \quad m \geq 0 \tag{6.23}$$

*Determination of Nodal Abscissas and Weights

Theorem The nodal abscissas $x_0, x_1, \ldots, x_n$ are the zeroes of the polynomial $\varphi_{n+1}(x)$ that belongs to the orthogonal set defined in Eq. (6.18).

Proof We start the proof by letting $f(x) = P_{2n+1}(x)$ be a polynomial of degree $2n+1$. Because the Gaussian integration with $n+1$ nodes is exact for this polynomial, we have

$$\int_a^b w(x)\,P_{2n+1}(x)\,dx = \sum_{i=0}^{n} A_i\,P_{2n+1}(x_i) \tag{a}$$

A polynomial of degree $2n+1$ can always be written in the form

$$P_{2n+1}(x) = Q_n(x) + R_n(x)\varphi_{n+1}(x) \tag{b}$$

where $Q_n(x)$, $R_n(x)$, and $\varphi_{n+1}(x)$ are polynomials of the degree indicated by the subscripts.[2] Therefore,

$$\int_a^b w(x)\,P_{2n+1}(x)\,dx = \int_a^b w(x)\,Q_n(x)\,dx + \int_a^b w(x)\,R_n(x)\varphi_{n+1}(x)\,dx$$

But according to Eq. (6.23) the second integral on the right-hand side vanishes, so that

$$\int_a^b w(x)\,P_{2n+1}(x)\,dx = \int_a^b w(x)\,Q_n(x)\,dx \tag{c}$$

[2] It can be shown that $Q_n(x)$ and $R_n(x)$ are unique for a given $P_{2n+1}(x)$ and $\varphi_{n+1}(x)$.

Because a polynomial of degree n is uniquely defined by $n + 1$ points, it is always possible to find A_i such that

$$\int_a^b w(x)\, Q_n(x) dx = \sum_{i=0}^{n} A_i\, Q_n(x_i) \tag{d}$$

In order to arrive at Eq. (a), we must choose for the nodal abscissas x_i the roots of $\varphi_{n+1}(x) = 0$. According to Eq. (b), we then have

$$P_{2n+1}(x_i) = Q_n(x_i), \quad i = 0, 1, \ldots, n \tag{e}$$

which, together with Eqs. (c) and (d), leads to

$$\int_a^b w(x)\, P_{2n+1}(x) dx = \int_a^b w(x)\, Q_n(x) dx = \sum_{i=0}^{n} A_i\, P_{2n+1}(x_i)$$

This completes the proof.

Theorem

$$A_i = \int_a^b w(x)\ell_i(x) dx, \quad i = 0, 1, \ldots, n \tag{6.24}$$

where $\ell_i(x)$ are the Lagrange's cardinal functions spanning the nodes at $x_0, x_1, \ldots x_n$. These functions were defined in Eq. (4.2).

Proof Applying Lagrange's formula, Eq. (4.1), to $Q_n(x)$ yields

$$Q_n(x) = \sum_{i=0}^{n} Q_n(x_i)\ell_i(x)$$

which, upon substitution in Eq. (d), gives us

$$\sum_{i=0}^{n} \left[Q_n(x_i) \int_a^b w(x)\ell_i(x) dx \right] = \sum_{i=0}^{n} A_i\, Q_n(x_i)$$

or

$$\sum_{i=0}^{n} Q_n(x_i) \left[A_i - \int_a^b w(x)\ell_i(x) dx \right] = 0$$

This equation can be satisfied for arbitrary $Q(x)$ of degree n only if

$$A_i - \int_a^b w(x)\ell_i(x) dx = 0, \quad i = 0, 1, \ldots, n$$

which is equivalent to Eq. (6.24).

It is not difficult to compute the zeroes x_i, $i = 0, 1, \ldots, n$ of a polynomial $\varphi_{n+1}(x)$ belonging to an orthogonal set by one of the methods discussed in Chapter 4. Once the zeroes are known, the weights A_i, $i = 0, 1, \ldots, n$ could be found from Eq. (6.24).

However, the following formulas (given without proof) are easier to compute:

$$\begin{aligned} &\text{Gauss–Legendre} \quad A_i = \frac{2}{(1-x_i^2)\left[p'_{n+1}(x_i)\right]^2} \\ &\text{Gauss–Laguerre} \quad A_i = \frac{1}{x_i\left[L'_{n+1}(x_i)\right]^2} \\ &\text{Gauss–Hermite} \quad A_i = \frac{2^{n+2}\,(n+1)!\sqrt{\pi}}{\left[H'_{n+1}(x_i)\right]^2} \end{aligned} \tag{6.25}$$

Abscissas and Weights for Classical Gaussian Quadratures

Here we list some classical Gaussian integration formulas. The tables of nodal abscissas and weights, covering $n = 1$ to 5, have been rounded off to six decimal places. These tables should be adequate for hand computation, but in programming you may need more precision or a larger number of nodes. In that case you should consult other references[3] or use a subroutine to compute the abscissas and weights within the integration program.[4]

The truncation error in Gaussian quadrature

$$E = \int_a^b w(x)f(x)dx - \sum_{i=0}^{n} A_i f(x_i)$$

has the form $E = K(n)f^{(2n+2)}(c)$, where $a < c < b$ (the value of c is unknown; only its bounds are given). The expression for $K(n)$ depends on the particular quadrature being used. If the derivatives of $f(x)$ can be evaluated, the error formulas are useful in estimating the error bounds.

Gauss–Legendre Quadrature

$$\int_{-1}^{1} f(\xi)d\xi \approx \sum_{i=0}^{n} A_i f(\xi_i) \tag{6.26}$$

This is the most-often-used Gaussian integration formula (see Table 6.3). The nodes are arranged symmetrically about $\xi = 0$, and the weights associated with a symmetric pair of nodes are equal. For example, for $n = 1$ we have $\xi_0 = -\xi_1$ and $A_0 = A_1$. The truncation error in Eq. (6.26) is

$$E = \frac{2^{2n+3}\left[(n+1)!\right]^4}{(2n+3)\left[(2n+2)!\right]^3} f^{(2n+2)}(c), \quad -1 < c < 1 \tag{6.27}$$

To apply Gauss–Legendre quadrature to the integral $\int_a^b f(x)dx$, we must first map the integration range (a, b) into the "standard" range $(-1, 1)$. We can accomplish this

[3] M. Abramowitz, and I. A. Stegun, *Handbook of Mathematical Functions* (Dover Publications, 1965). A. H. Stroud and D. Secrest, *Gaussian Quadrature Formulas* (Prentice-Hall, 1966).

[4] Several such subroutines are listed in W. H. Press et al, *Numerical Recipes in Fortran 90* (Cambridge University Press, 1996).

$\pm\xi_i$	A_i	$\pm\xi_i$	A_i
$n=1$		$n=4$	
0.577 350	1.000 000	0.000 000	0.568 889
$n=2$		0.538 469	0.478 629
0.000 000	0.888 889	0.906 180	0.236 927
0.774 597	0.555 556	$n=5$	
$n=3$		0.238 619	0.467 914
0.339 981	0.652 145	0.661 209	0.360 762
0.861 136	0.347 855	0.932 470	0.171 324

Table 6.3

by the transformation

$$x = \frac{b+a}{2} + \frac{b-a}{2}\xi \tag{6.28}$$

Now $dx = d\xi(b-a)/2$, and the quadrature becomes

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\sum_{i=1}^{n} A_i f(x_i) \tag{6.29}$$

where the abscissas x_i must be computed from Eq. (6.28). The truncation error here is

$$E = \frac{(b-a)^{2n+3}\left[(n+1)!\right]^4}{(2n+3)\left[(2n+2)!\right]^3} f^{(2n+2)}(c), \quad a < c < b \tag{6.30}$$

Gauss–Chebyshev Quadrature

$$\int_{-1}^{1} \left(1-x^2\right)^{-1/2} f(x)dx \approx \frac{\pi}{n+1}\sum_{i=0}^{n} f(x_i) \tag{6.31}$$

Note that all the weights are equal: $A_i = \pi/(n+1)$. The abscissas of the nodes, which are symmetric about $x = 0$, are given by

$$x_i = \cos\frac{(2i+1)\pi}{2n+2} \tag{6.32}$$

The truncation error is

$$E = \frac{2\pi}{2^{2n+2}(2n+2)!} f^{(2n+2)}(c), \quad -1 < c < 1 \tag{6.33}$$

Gauss–Laguerre Quadrature

$$\int_0^{\infty} e^{-x} f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i) \tag{6.34}$$

where the weights and the abscissas are given in Table 6.4.

x_i	A_i	x_i	A_i
$n=1$		$n=4$	
0.585 786	0.853 554	0.263 560	0.521 756
3.414 214	0.146 447	1.413 403	0.398 667
$n=2$		3.596 426	(−1)0.759 424
0.415 775	0.711 093	7.085 810	(−2)0.361 175
2.294 280	0.278 517	12.640 801	(−4)0.233 670
6.289 945	(−1)0.103 892	$n=5$	
$n=3$		0.222 847	0.458 964
0.322 548	0.603 154	1.188 932	0.417 000
1.745 761	0.357 418	2.992 736	0.113 373
4.536 620	(−1)0.388 791	5.775 144	(−1)0.103 992
9.395 071	(−3)0.539 295	9.837 467	(−3)0.261 017
		15.982 874	(−6)0.898 548

Table 6.4 Multiply numbers by 10^k, where k is given in parentheses

$$E = \frac{\left[(n+1)!\right]^2}{(2n+2)!} f^{(2n+2)}(c), \quad 0 < c < \infty \tag{6.35}$$

Gauss–Hermite Quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)\,dx \approx \sum_{i=0}^{n} A_i f(x_i) \tag{6.36}$$

The nodes are placed symmetrically abut $x = 0$ as indicated in Table 6.5.

$$E = \frac{\sqrt{\pi}(n+1)!}{2^2(2n+2)!} f^{(2n+2)}(c), \quad 0 < c < \infty \tag{6.37}$$

$\pm x_i$	A_i	$\pm x_i$	A_i
$n=1$		$n=4$	
0.707 107	0.886 227	0.000 000	0.945 308
$n=2$		0.958 572	0.393 619
0.000 000	1.181 636	2.020 183	(−1) 0.199 532
1.224745	0.295 409	$n=5$	
$n=3$		0.436 077	0.724 629
0.524 648	0.804 914	1.335 849	0.157 067
1.650 680	(−1)0.813 128	2.350 605	(−2)0.453 001

Table 6.5 Multiply numbers by 10^k, where k is given in parentheses

Gauss Quadrature with Logarithmic Singularity

$$\int_0^1 f(x)\ln(x)dx \approx -\sum_{i=0}^{n} A_i f(x_i) \tag{6.38}$$

The weights and the abscissas are given in Table 6.6.

$$E = \frac{k(n)}{(2n+1)!} f^{(2n+1)}(c), \quad 0 < c < 1 \tag{6.39}$$

where $k(1) = 0.00\,285$, $k(2) = 0.000\,17$, $k(3) = 0.000\,01$.

x_i	A_i	x_i	A_i
$n = 1$		$n = 4$	
0.112 009	0.718 539	(−1)0.291 345	0.297 893
0.602 277	0.281 461	0.173 977	0.349 776
$n = 2$		0.411 703	0.234 488
(−1)0.638 907	0.513 405	0.677314	(−1)0.989 305
0.368 997	0.391 980	0.894 771	(−1)0.189 116
0.766 880	(−1)0.946 154	$n = 5$	
$n = 3$		(−1)0.216 344	0.238 764
(−1)0.414 485	0.383 464	0.129 583	0.308 287
0.245 275	0.386 875	0.314 020	0.245 317
0.556 165	0.190 435	0.538 657	0.142 009
0.848 982	(−1)0.392 255	0.756 916	(−1)0.554 546
		0.922 669	(−1)0.101 690

Table 6.6 Multiply numbers by 10^k, where k is given in parentheses

■ gaussNodes

The function `gaussNodes` listed here[5] computes the nodal abscissas x_i and the corresponding weights A_i used in Gauss–Legendre quadrature over the "standard" interval $(-1, 1)$. It can be shown that the approximate values of the abscissas are

$$x_i = \cos\frac{\pi(i + 0.75)}{m + 0.5}$$

where $m = n + 1$ is the number of nodes, also called the *integration order*. Using these approximations as the starting values, the nodal abscissas are computed by finding the nonnegative zeroes of the Legendre polynomial $p_m(x)$ with Newton's method (the negative zeroes are obtained from symmetry). Note that `gaussNodes` calls the subfunction `legendre`, which returns $p_m(t)$ and its derivative as the tuple `(p,dp)`.

[5] This function is an adaptation of a routine in W. H. Press et al., *Numerical Recipes in Fortran 90* (Cambridge University Press, 1996).

```
## module gaussNodes
''' x,A = gaussNodes(m,tol=10e-9)
    Returns nodal abscissas {x} and weights {A} of
    Gauss--Legendre m-point quadrature.
'''
from math import cos,pi
from numpy import zeros

def gaussNodes(m,tol=10e-9):

    def legendre(t,m):
        p0 = 1.0; p1 = t
        for k in range(1,m):
            p = ((2.0*k + 1.0)*t*p1 - k*p0)/(1.0 + k )
            p0 = p1; p1 = p
        dp = m*(p0 - t*p1)/(1.0 - t**2)
        return p,dp

    A = zeros(m)
    x = zeros(m)
    nRoots = (m + 1)/2          # Number of non-neg. roots
    for i in range(nRoots):
        t = cos(pi*(i + 0.75)/(m + 0.5))  # Approx. root
        for j in range(30):
            p,dp = legendre(t,m)          # Newton-Raphson
            dt = -p/dp; t = t + dt        # method
            if abs(dt) < tol:
                x[i] = t; x[m-i-1] = -t
                A[i] = 2.0/(1.0 - t**2)/(dp**2) # Eq.(6.25)
                A[m-i-1] = A[i]
                break
    return x,A
```

■ gaussQuad

The function `gaussQuad` utilizes `gaussNodes` to evaluate $\int_a^b f(x)\,dx$ with Gauss–Legendre quadrature using m nodes. The function routine for $f(x)$ must be supplied by the user.

```
## module gaussQuad
''' I = gaussQuad(f,a,b,m).
    Computes the integral of f(x) from x = a to b
    with Gauss--Legendre quadrature using m nodes.
'''
from gaussNodes import *
```

```
def gaussQuad(f,a,b,m):
    c1 = (b + a)/2.0
    c2 = (b - a)/2.0
    x,A = gaussNodes(m)
    sum = 0.0
    for i in range(len(x)):
        sum = sum + A[i]*f(c1 + c2*x[i])
    return c2*sum
```

EXAMPLE 6.8

Evaluate $\int_{-1}^{1}(1-x^2)^{3/2}dx$ as accurately as possible with Gaussian integration.

Solution As the integrand is smooth and free of singularities, we could use Gauss–Legendre quadrature. However, the exact integral can be obtained with the Gauss–Chebyshev formula. We write

$$\int_{-1}^{1}\left(1-x^2\right)^{3/2}dx = \int_{-1}^{1}\frac{\left(1-x^2\right)^2}{\sqrt{1-x^2}}dx$$

The numerator $f(x) = (1-x^2)^2$ is a polynomial of degree 4, so that Gauss–Chebyshev quadrature is exact with three nodes.

The abscissas of the nodes are obtained from Eq. (6.32). Substituting $n = 2$, we get

$$x_i = \cos\frac{(2i+1)\pi}{6}, \quad i = 0, 1, 2$$

Therefore,

$$\begin{aligned} x_0 &= \cos\frac{\pi}{6} = \frac{\sqrt{3}}{2} \\ x_1 &= \cos\frac{\pi}{2} = 0 \\ x_2 &= \cos\frac{5\pi}{6} = \frac{\sqrt{3}}{2} \end{aligned}$$

and Eq. (6.31) yields

$$\begin{aligned} \int_{-1}^{1}\left(1-x^2\right)^{3/2}dx &\approx \frac{\pi}{3}\sum_{i=0}^{2}\left(1-x_i^2\right)^2 \\ &= \frac{\pi}{3}\left[\left(1-\frac{3}{4}\right)^2 + (1-0)^2 + \left(1-\frac{3}{4}\right)^2\right] = \frac{3\pi}{8} \end{aligned}$$

EXAMPLE 6.9

Use Gaussian integration to evaluate $\int_{0}^{0.5}\cos\pi x\ln x\,dx$.

Solution We split the integral into two parts:

$$\int_{0}^{0.5}\cos\pi x\ln x\,dx = \int_{0}^{1}\cos\pi x\ln x\,dx - \int_{0.5}^{1}\cos\pi x\ln x\,dx$$

The first integral on the right-hand side, which contains a logarithmic singularity at $x = 0$, can be computed with the special Gaussian quadrature in Eq. (6.38). Choosing $n = 3$, we have

$$\int_0^1 \cos \pi x \ln x \, dx \approx -\sum_{i=0}^{3} A_i \cos \pi x_i$$

The sum is evaluated in the following table:

x_i	$\cos \pi x_i$	A_i	$A_i \cos \pi x_i$
0.041 448	0.991 534	0.383 464	0.380 218
0.245 275	0.717 525	0.386 875	0.277 592
0.556 165	−0.175 533	0.190 435	−0.033 428
0.848 982	−0.889 550	0.039 225	−0.034 892
			$\Sigma = 0.589\,490$

Thus,

$$\int_0^1 \cos \pi x \ln x \, dx \approx -0.589\,490$$

The second integral is free of singularities, so it can be evaluated with Gauss–Legendre quadrature. Choosing $n = 3$, we have

$$\int_{0.5}^1 \cos \pi x \ln x \, dx \approx 0.25 \sum_{i=0}^{3} A_i \cos \pi x_i \ln x_i$$

where the nodal abscissas are – see Eq. (6.28)

$$x_i = \frac{1 + 0.5}{2} + \frac{1 - 0.5}{2}\xi_i = 0.75 + 0.25\xi_i$$

Looking up ξ_i and A_i in Table 6.3 leads to the following computations:

ξ_i	x_i	$\cos \pi x_i \ln x_i$	A_i	$A_i \cos \pi x_i \ln x_i$
−0.861 136	0.534 716	0.068 141	0.347 855	0.023 703
−0.339 981	0.665 005	0.202 133	0.652 145	0.131 820
0.339 981	0.834 995	0.156 638	0.652 145	0.102 151
0.861 136	0.965 284	0.035 123	0.347 855	0.012 218
				$\Sigma = 0.269\,892$

from which

$$\int_{0.5}^1 \cos \pi x \ln x \, dx \approx 0.25(0.269\,892) = 0.067\,473$$

Therefore,

$$\int_0^1 \cos \pi x \ln x \, dx \approx -0.\,589\,490 - 0.067\,473 = -0.\,656\,96\,3$$

which is correct to six decimal places.

EXAMPLE 6.10

Evaluate as accurately as possible

$$F = \int_0^\infty \frac{x+3}{\sqrt{x}} e^{-x} dx$$

Solution In its present form, the integral is not suited to any of the Gaussian quadratures listed in this section. But using the transformation

$$x = t^2 \qquad dx = 2t\, dt$$

the integral becomes

$$F = 2\int_0^\infty (t^2+3)e^{-t^2} dt = \int_{-\infty}^\infty (t^2+3)e^{-t^2} dt$$

which can be evaluated exactly with the Gauss–Hermite formula using only two nodes ($n = 1$). Thus,

$$\begin{aligned} F &= A_0(t_0^2+3) + A_1(t_1^2+3) \\ &= 0.886\,227\left[(0.707\,107)^2+3\right] + 0.886\,227\left[(-0.707\,107)^2+3\right] \\ &= 6.203\,59 \end{aligned}$$

EXAMPLE 6.11

Determine how many nodes are required to evaluate

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

with Gauss–Legendre quadrature to six decimal places. The exact integral, rounded to six places, is 1.418 15.

Solution The integrand is a smooth function, hence it is suited for Gauss–Legendre integration. There is an indeterminacy at $x = 0$, but this does not bother the quadrature because the integrand is never evaluated at that point. We used the following program that computes the quadrature with 2, 3, . . . nodes until the desired accuracy is reached:

```
## example 6_11
from math import pi,sin
from gaussQuad import *

def f(x): return (sin(x)/x)**2

a = 0.0; b = pi;
Iexact = 1.41815
for m in range(2,12):
    I = gaussQuad(f,a,b,m)
    if abs(I - Iexact) < 0.00001:
```

```
        print ''Number of nodes ='',m
        print ''Integral ='', gaussQuad(f,a,b,m)
        break
raw_input(''\nPress return to exit'')
```

The program output is

```
Number of nodes = 5
Integral = 1.41815026778
```

EXAMPLE 6.12

Evaluate numerically $\int_{1.5}^{3} f(x)\,dx$, where $f(x)$ is represented by the unevenly spaced data

x	1.2	1.7	2.0	2.4	2.9	3.3
$f(x)$	−0.362 36	0.128 84	0.416 15	0.737 39	0.970 96	0.987 48

Knowing that the data points lie on the curve $f(x) = -\cos x$, evaluate the accuracy of the solution.

Solution We approximate $f(x)$ by the polynomial $P_5(x)$ that intersects all the data points, and then evaluate $\int_{1.5}^{3} f(x)dx \approx \int_{1.5}^{3} P_5(x)dx$ with the Gauss–Legendre formula. Because the polynomial is of degree 5, only three nodes ($n = 2$) are required in the quadrature.

From Eq. (6.28) and Table 6.6, we obtain for the abscissas of the nodes

$$x_0 = \frac{3+1.5}{2} + \frac{3-1.5}{2}(-0.774597) = 1.6691$$

$$x_1 = \frac{3+1.5}{2} = 2.25$$

$$x_2 = \frac{3+1.5}{2} + \frac{3-1.5}{2}(0.774597) = 2.8309$$

We now compute the values of the interpolant $P_5(x)$ at the nodes. This can be done using the modules `newtonPoly` or `neville` listed in Section 3.2. The results are

$$P_5(x_0) = 0.098\,08 \qquad P_5(x_1) = 0.628\,16 \qquad P_5(x_2) = 0.952\,16$$

From Gauss–Legendre quadrature

$$I = \int_{1.5}^{3} P_5(x)dx = \frac{3-1.5}{2}\sum_{i=0}^{2} A_i P_5(x_i)$$

we get

$$I = 0.75\,[0.555\,556(0.098\,08) + 0.888\,889(0.628\,16) + 0.555\,556(0.952\,16)]$$

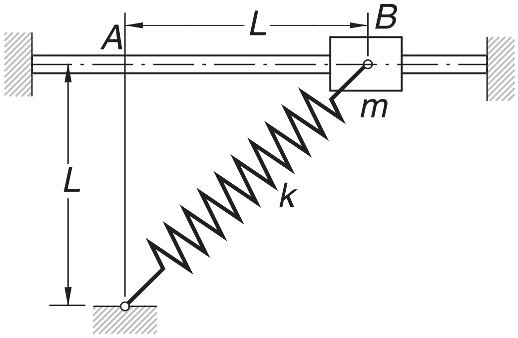
$$= 0.856\,37$$

Comparison with $-\int_{1.5}^{3} \cos x\,dx = 0.856\,38$ shows that the discrepancy is within the roundoff error.

PROBLEM SET 6.2

1. Evaluate
$$\int_1^{\pi} \frac{\ln(x)}{x^2 - 2x + 2} dx$$
with Gauss–Legendre quadrature. Use (a) two nodes, and (b) four nodes.
2. Use Gauss–Laguerre quadrature to evaluate $\int_0^{\infty} (1 - x^2)^3 e^{-x}\, dx$.
3. Use Gauss–Chebyshev quadrature with six nodes to evaluate
$$\int_0^{\pi/2} \frac{dx}{\sqrt{\sin x}}$$
Compare the result with the "exact" value 2.62206. *Hint*: substitute $\sin x = t^2$.
4. The integral $\int_0^{\pi} \sin x\, dx$ is evaluated with Gauss–Legendre quadrature using four nodes. What are the bounds on the truncation error resulting from the quadrature?
5. How many nodes are required in Gauss–Laguerre quadrature to evaluate $\int_0^{\infty} e^{-x} \sin x\, dx$ to six decimal places?
6. Evaluate as accurately as possible
$$\int_0^1 \frac{2x + 1}{\sqrt{x(1 - x)}} dx$$
Hint: substitute $x = (1 + t)/2$.
7. Compute $\int_0^{\pi} \sin x \ln x\, dx$ to four decimal places.
8. Calculate the bounds on the truncation error if $\int_0^{\pi} x \sin x\, dx$ is evaluated with Gauss–Legendre quadrature using three nodes. What is the actual error?
9. Evaluate $\int_0^2 (\sinh x/x)\, dx$ to four decimal places.
10. ■ Evaluate the integral
$$\int_0^{\infty} \frac{x\, dx}{e^x + 1}$$
by Gauss–Legendre quadrature to six decimal places. *Hint*: substitute $e^x = \ln(1/t)$.
11. ■ The equation of an ellipse is $x^2/a^2 + y^2/b^2 = 1$. Write a program that computes the length
$$S = 2 \int_{-a}^{a} \sqrt{1 + (dy/dx)^2}\, dx$$
of the circumference to five decimal places for a given a and b. Test the program with $a = 2$ and $b = 1$.
12. ■ The error function, which is of importance in statistics, is defined as
$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$
Write a program that uses Gauss–Legendre quadrature to evaluate erf(x) for a given x to six decimal places. Note that erf(x) = 1.000 000 (correct to six decimal places) when $x > 5$. Test the program by verifying that erf(1.0) = 0.842 701.

13. ■

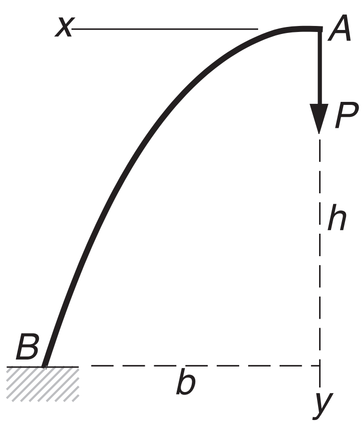


The sliding weight of mass m is attached to a spring of stiffness k that has an undeformed length L. When the mass is released from rest at B, the time it takes to reach A can be shown to be $t = C\sqrt{m/k}$, where

$$C = \int_0^1 \left[\left(\sqrt{2}-1\right)^2 - \left(\sqrt{1+z^2}-1\right)^2\right]^{-1/2} dz$$

Compute C to six decimal places. *Hint*: the integrand has singularity at $z = 1$ that behaves as $(1 - z^2)^{-1/2}$.

14. ■



A uniform beam forms the semiparabolic cantilever arch AB. The vertical displacement of A due to the force P can be shown to be

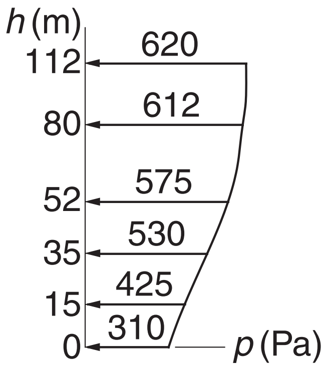
$$\delta_A = \frac{Pb^3}{EI} C\left(\frac{h}{b}\right)$$

where EI is the bending rigidity of the beam and

$$C\left(\frac{h}{b}\right) = \int_0^1 z^2 \sqrt{1 + \left(\frac{2h}{b}z\right)^2}\, dz$$

Write a program that computes $C(h/b)$ for any given value of h/b to four decimal places. Use the program to compute $C(0.5)$, $C(1.0)$, and $C(2.0)$.

15. ■ There is no elegant way to compute $I = \int_0^{\pi/2} \ln(\sin x)\, dx$. A "brute force" method that works is to split the integral into several parts: from $x = 0$ to 0.01, from 0.01 to 0.2, and from $x = 0.02$ to $\pi/2$. In the first part, we can use the approximation $\sin x \approx x$, which allows us to obtain the integral analytically. The other two parts can be evaluated with Gauss–Legendre quadrature. Use this method to evaluate I to six decimal places.
16. ■



The pressure of wind was measured at various heights on a vertical wall, as shown on the diagram. Find the height of the pressure center, which is defined as

$$\bar{h} = \frac{\int_0^{112\text{ m}} h\, p(h)\, dh}{\int_0^{112\text{ m}} p(h)\, dh}$$

Hint: fit a cubic polynomial to the data and then apply Gauss–Legendre quadrature.
17. ■ Write a function that computes $\int_{x_1}^{x_n} y(x)\, dx$ from a given set of data points of the form

x_1	x_2	x_3	$\cdots$	x_n
y_1	y_2	y_3	$\cdots$	y_n

The function must work for unevenly spaced x-values. Test the function with the data given in Prob. 17, Problem Set 6.1. *Hint*: fit a cubic spline to the data points and apply Gauss–Legendre quadrature to each segment of the spline.

*6.5 Multiple Integrals

Multiple integrals, such as the area integral $\int\int_A f(x, y)\, dx\, dy$, can also be evaluated by quadrature. The computations are straightforward if the region of integration has a simple geometric shape, such as a triangle or a quadrilateral. Because of complications in specifying the limits of integration on x and y, quadrature is not a practical means of evaluating integrals over irregular regions. However, an irregular region A can always be approximated as an assembly triangular or quadrilateral subregions

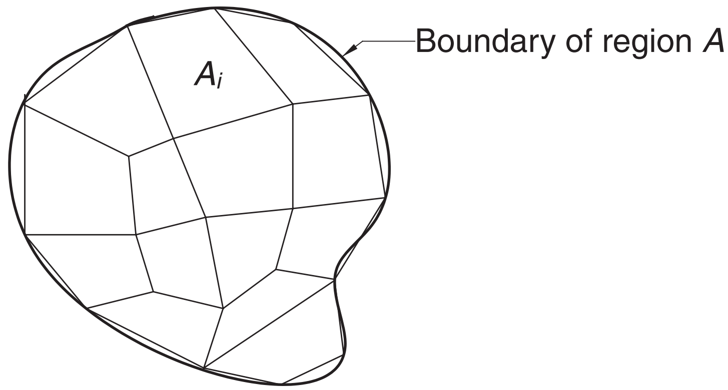


Figure 6.6. Finite element model of an irregular region.

$A_1, A_2, \ldots$, called *finite elements*, as illustrated in Fig. 6.6. The integral over A can then be evaluated by summing the integrals over the finite elements:

$$\int\int_A f(x, y)\,dx\,dy \approx \sum_i \int\int_{A_i} f(x, y)\,dx\,dy$$

Volume integrals can be computed in a similar manner, using tetrahedra or rectangular prisms for the finite elements.

Gauss–Legendre Quadrature over a Quadrilateral Element

Consider the double integral

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi, \eta)\,d\eta\,d\xi$$

over the rectangular element shown in Fig. 6.7(a). Evaluating each integral in turn by Gauss–Legendre quadrature using $n + 1$ integration points in each coordinate

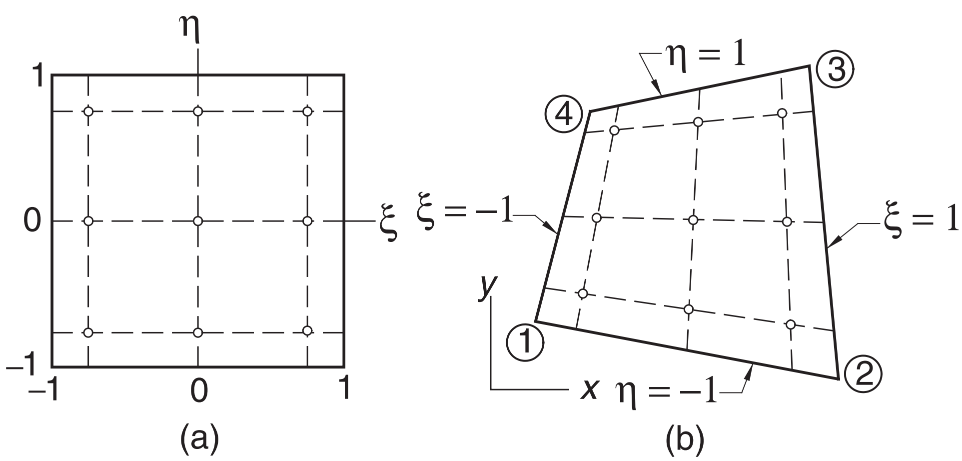


Figure 6.7. Mapping a quadrilateral into the standard rectangle.

direction, we obtain

$$I = \int_{-1}^{1} \sum_{i=0}^{n} A_i f(\xi_i, \eta)\, d\eta = \sum_{j=0}^{n} A_j \left[\sum_{i=0}^{n} A_i f(\xi_i, \eta_i) \right]$$

or

$$I = \sum_{i=0}^{n} \sum_{j=0}^{n} A_i A_j f(\xi_i, \eta_j) \tag{6.40}$$

As noted previously, the number of integration points in each coordinate direction, $m = n + 1$, is called the *integration order*. Figure 6.7(a) shows the locations of the integration points used in third-order integration ($m = 3$). Because the integration limits were the "standard" limits $(-1, 1)$ of Gauss–Legendre quadrature, the weights and the coordinates of the integration points are as listed in Table 6.3.

In order to apply quadrature to the quadrilateral element in Fig. 6.7(b), we must first map the quadrilateral into the "standard" rectangle in Fig. 6.7(a). By mapping we mean a coordinate transformation $x = x(\xi, \eta)$, $y = y(\xi, \eta)$ that results in one-to-one correspondence between points in the quadrilateral and in the rectangle. The transformation that does the job is

$$x(\xi, \eta) = \sum_{k=1}^{4} N_k(\xi, \eta) x_k \qquad y(\xi, \eta) = \sum_{k=1}^{4} N_k(\xi, \eta) y_k \tag{6.41}$$

where (x_k, y_k) are the coordinates of corner k of the quadrilateral and

$$\begin{aligned} N_1(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 - \eta) \\ N_2(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ N_3(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 + \eta) \\ N_4(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned} \tag{6.42}$$

The functions $N_k(\xi, \eta)$, known as the *shape functions*, are bilinear (linear in each coordinate). Consequently, straight lines remain straight upon mapping. In particular, note that the sides of the quadrilateral are mapped into the lines $\xi = \pm 1$ and $\eta = \pm 1$.

Because mapping distorts areas, an infinitesimal area element $dA = dx\, dy$ of the quadrilateral is not equal to its counterpart $dA' = d\xi\, d\eta$ of the rectangle. It can be shown that the relationship between the areas is

$$dx\, dy = |J(\xi, \eta)|\; d\xi\, d\eta \tag{6.43}$$

where

$$J(\xi, \eta) = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \tag{6.44a}$$

is the known as the *Jacobian matrix* of the mapping. Substituting from Eqs. (6.41) and (6.42) and differentiating, the components of the Jacobian matrix are

$$\begin{aligned} J_{11} &= \frac{1}{4}\left[-(1-\eta)x_1 + (1-\eta)x_2 + (1+\eta)x_3 - (1-\eta)x_4\right] \\ J_{12} &= \frac{1}{4}\left[-(1-\eta)y_1 + (1-\eta)y_2 + (1+\eta)y_3 - (1-\eta)y_4\right] \\ J_{21} &= \frac{1}{4}\left[-(1-\xi)x_1 - (1+\xi)x_2 + (1+\xi)x_3 + (1-\xi)x_4\right] \\ J_{22} &= \frac{1}{4}\left[-(1-\xi)y_1 - (1+\xi)y_2 + (1+\xi)y_3 + (1-\xi)y_4\right] \end{aligned} \tag{6.44b}$$

We can now write

$$\int\int_A f(x, y)\,dx\,dy = \int_{-1}^{1}\int_{-1}^{1} f\left[x(\xi,\eta), y(\xi,\eta)\right]\,|J(\xi,\eta)|\,d\xi\,d\eta \tag{6.45}$$

Because the right-hand-side integral is taken over the "standard" rectangle, it can be evaluated using Eq. (6.40). Replacing $f(\xi, \eta)$ in Eq. (6.40) by the integrand in Eq. (6.45), we get the following formula for Gauss–Legendre quadrature over a quadrilateral region:

$$I = \sum_{i=0}^{n}\sum_{j=0}^{n} A_i A_j\, f\left[x(\xi_i, \eta_j), y(\xi_i, \eta_j)\right]\,\left|J(\xi_i, \eta_j)\right| \tag{6.46}$$

The ξ and η coordinates of the integration points and the weights can again be obtained from Table 6.3.

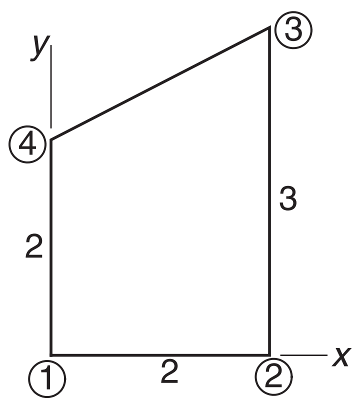
■ gaussQuad2

The function `gaussQuad2` in this module computes $\int\int_A f(x, y)\,dx\,dy$ over a quadrilateral element with Gauss–Legendre quadrature of integration order m. The quadrilateral is defined by the arrays **x** and **y**, which contain the coordinates of the four corners ordered in a *counterclockwise direction* around the element. The determinant of the Jacobian matrix is obtained by calling the function `jac`; mapping is performed by `map`. The weights and the values of ξ and η at the integration points are computed by `gaussNodes` listed in the previous section (note that ξ and η appear as s and t in the listing).

```
from gaussNodes import *
from numpy import zeros,dot

def gaussQuad2(f,x,y,m):

    def jac(x,y,s,t):
        J = zeros((2,2))
        J[0,0] = -(1.0 - t)*x[0] + (1.0 - t)*x[1]  \
                + (1.0 + t)*x[2] - (1.0 + t)*x[3]
```

```
    J[0,1] = -(1.0 - t)*y[0] + (1.0 - t)*y[1]  \
            + (1.0 + t)*y[2] - (1.0 + t)*y[3]
    J[1,0] = -(1.0 - s)*x[0] - (1.0 + s)*x[1]  \
            + (1.0 + s)*x[2] + (1.0 - s)*x[3]
    J[1,1] = -(1.0 - s)*y[0] - (1.0 + s)*y[1]  \
            + (1.0 + s)*y[2] + (1.0 - s)*y[3]
    return (J[0,0]*J[1,1] - J[0,1]*J[1,0])/16.0

def map(x,y,s,t):
    N = zeros(4)
    N[0] = (1.0 - s)*(1.0 - t)/4.0
    N[1] = (1.0 + s)*(1.0 - t)/4.0
    N[2] = (1.0 + s)*(1.0 + t)/4.0
    N[3] = (1.0 - s)*(1.0 + t)/4.0
    xCoord = dot(N,x)
    yCoord = dot(N,y)
    return xCoord,yCoord

s,A = gaussNodes(m)
sum = 0.0
for i in range(m):
    for j in range(m):
        xCoord,yCoord = map(x,y,s[i],s[j])
        sum = sum + A[i]*A[j]*jac(x,y,s[i],s[j])  \
                   *f(xCoord,yCoord)
return sum
```

EXAMPLE 6.13



Evaluate the integral

$$I = \int\int_A (x^2 + y)\, dx\, dy$$

analytically by first transforming it from the quadrilateral region A shown to the "standard" rectangle.

Solution The corner coordinates of the quadrilateral are

$$\mathbf{x}^T = \begin{bmatrix} 0 & 2 & 2 & 0 \end{bmatrix} \qquad \mathbf{y}^T = \begin{bmatrix} 0 & 0 & 3 & 2 \end{bmatrix}$$

The mapping is

$$\begin{aligned}
x(\xi, \eta) &= \sum_{k=1}^{4} N_k(\xi, \eta) x_k \\
&= 0 + \frac{(1+\xi)(1-\eta)}{4}(2) + \frac{(1+\xi)(1+\eta)}{4}(2) + 0 \\
&= 1 + \xi \\
y(\xi, \eta) &= \sum_{k=1}^{4} N_k(\xi, \eta) y_k \\
&= 0 + 0 + \frac{(1+\xi)(1+\eta)}{4}(3) + \frac{(1-\xi)(1+\eta)}{4}(2) \\
&= \frac{(5+\xi)(1+\eta)}{4}
\end{aligned}$$

which yields for the Jacobian matrix

$$J(\xi, \eta) = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{1+\eta}{4} \\ 0 & \dfrac{5+\xi}{4} \end{bmatrix}$$

Thus, the area scale factor is

$$|J(\xi, \eta)| = \frac{5+\xi}{4}$$

Now we can map the integral from the quadrilateral to the standard rectangle. Referring to Eq. (6.45), we obtain

$$\begin{aligned}
I &= \int_{-1}^{1}\int_{-1}^{1} \left[\left(\frac{1+\xi}{2}\right)^2 + \frac{(5+\xi)(1+\eta)}{4} \right] \frac{5+\xi}{4}\, d\xi\, d\eta \\
&= \int_{-1}^{1}\int_{-1}^{1} \left(\frac{15}{8} + \frac{21}{16}\xi + \frac{1}{2}\xi^2 + \frac{1}{16}\xi^3 + \frac{25}{16}\eta + \frac{5}{8}\xi\eta + \frac{1}{16}\xi^2\eta \right) d\xi\, d\eta
\end{aligned}$$

If we note that only even powers of ξ and η contribute to the integral, the integral simplifies to

$$I = \int_{-1}^{1}\int_{-1}^{1} \left(\frac{15}{8} + \frac{1}{2}\xi^2 \right) d\xi\, d\eta = \frac{49}{6}$$

EXAMPLE 6.14

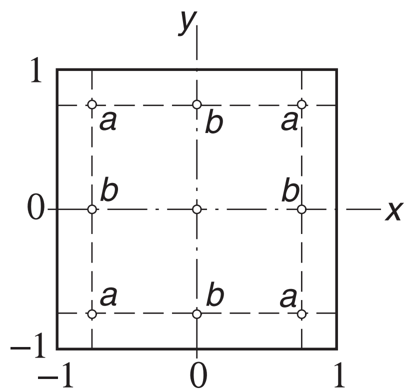
Evaluate the integral

$$\int_{-1}^{1}\int_{-1}^{1}\cos\frac{\pi x}{2}\cos\frac{\pi y}{2}\,dx\,dy$$

by Gauss–Legendre quadrature of order 3.

Solution From the quadrature formula in Eq. (6.40), we have

$$I=\sum_{i=0}^{2}\sum_{j=0}^{2}A_iA_j\cos\frac{\pi x_i}{2}\cos\frac{\pi y_j}{2}$$



The integration points are shown in the figure; their coordinates, and the corresponding weights are listed in Table 6.3. Note that the integrand, the integration points, and the weights are all symmetric about the coordinate axes. It follows that the points labeled a contribute equal amounts to I; the same is true for the points labeled b. Therefore,

$$\begin{aligned}I &= 4(0.555\,556)^2\cos^2\frac{\pi(0.774\,597)}{2}\\ &\quad +4(0.555\,556)(0.888\,889)\cos\frac{\pi(0.774\,597)}{2}\cos\frac{\pi(0)}{2}\\ &\quad +(0.888\,889)^2\cos^2\frac{\pi(0)}{2}\\ &= 1.623\,391\end{aligned}$$

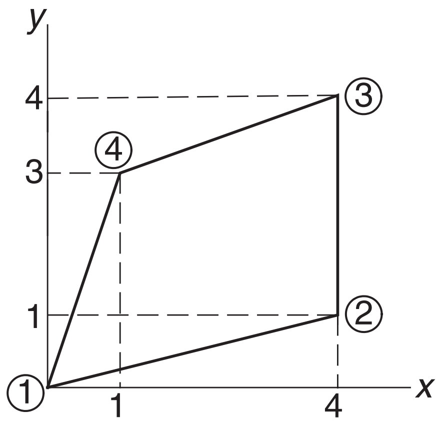
The exact value of the integral is $16/\pi^2 \approx 1.621\,139$.

EXAMPLE 6.15

Utilize `gaussQuad2` to evaluate $I=\int\int_A f(x,y)\,dx\,dy$ over the quadrilateral shown, where

$$f(x,y)=(x-2)^2(y-2)^2$$

Use enough integration points for an "exact" answer.



Solution The required integration order is determined by the integrand in Eq. (6.45):

$$I = \int_{-1}^{1}\int_{-1}^{1} f\left[x(\xi,\eta), y(\xi,\eta)\right] \left|J(\xi,\eta)\right| d\xi\, d\eta \tag{a}$$

We note that $|J(\xi,\eta)|$, defined in Eqs. (6.44), is biquadratic. Because the specified $f(x, y)$ is also biquadratic, the integrand in Eq. (a) is a polynomial of degree 4 in both ξ and η. Thus, third-order integration is sufficient for an "exact" result.

```
#!/usr/bin/python
## example 6_15
from gaussQuad2 import *
from numpy import array

def f(x,y): return ((x - 2.0)**2)*((y - 2.0)**2)

x = array([0.0, 4.0, 4.0, 1.0])
y = array([0.0, 1.0, 4.0, 3.0])
m = eval(raw_input(''Integration order ==> ''))
print ''Integral ='', gaussQuad2(gaussNodes,f,x,y,m)
raw_input(''\nPress return to exit'')
```

Running the preceding program produced the following result:

```
Integration order ==> 3
Integral = 11.3777777778
```

Quadrature over a Triangular Element

A triangle may be viewed as a degenerate quadrilateral with two of its corners occupying the same location, as illustrated in Fig. 6.8. Therefore, the integration formulas over a quadrilateral region can also be used for a triangular element. However, it is

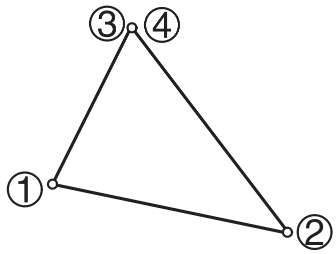


Figure 6.8. Degenerate quadrilateral.

computationally advantageous to use integration formulas specially developed for triangles, which we present without derivation.[6]

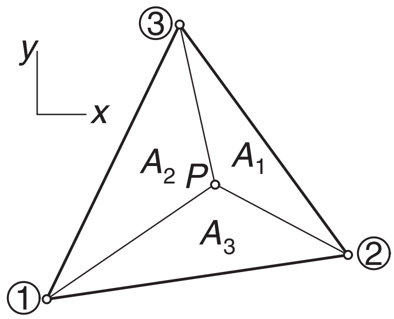


Figure 6.9. Triangular element.

Consider the triangular element in Fig. 6.9. Drawing straight lines from the point P in the triangle to each of the corners, we divide the triangle into three parts with areas A_1, A_2, and A_3. The *area coordinates* of P are defined as

$$\alpha_i = \frac{A_i}{A}, \quad i = 1, 2, 3 \tag{6.47}$$

where A is the area of the element. Because $A_1 + A_2 + A_3 = A$, the area coordinates are related by

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \tag{6.48}$$

Note that α_i ranges from 0 (when P lies on the side opposite to corner i) to 1 (when P is at corner i).

A convenient formula for computing A from the corner coordinates (x_i, y_i) is

$$A = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \tag{6.49}$$

The area coordinates are mapped into the Cartesian coordinates by

$$x(\alpha_1, \alpha_2, \alpha_3) = \sum_{i=1}^{3} \alpha_i x_i \qquad y(\alpha_1, \alpha_2, \alpha_3) = \sum_{i=1}^{3} \alpha_i y_i \tag{6.50}$$

[6] The triangle formulas are extensively used in finite method analysis. See, for example, O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Vol. 1, 4th ed. (McGraw-Hill, 1989).

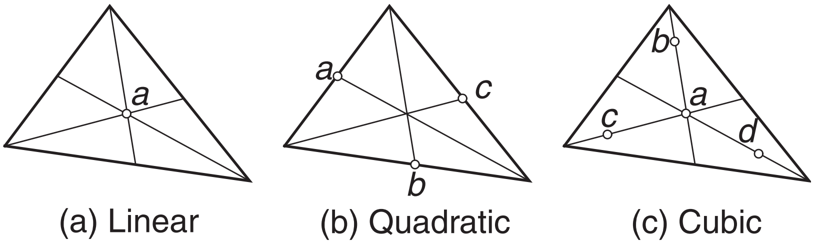


Figure 6.10. Integration points of triangular elements.

The integration formula over the element is

$$\int\int_A f\left[x(\boldsymbol{\alpha}),\, y(\boldsymbol{\alpha})\right]\, dA = A \sum_k W_k f\left[x(\boldsymbol{\alpha}_k),\, y(\boldsymbol{\alpha}_k)\right] \tag{6.51}$$

where $\boldsymbol{\alpha}_k$ represents the area coordinates of the integration point k, and W_k are the weights. The locations of the integration points are shown in Figure 6.10, and the corresponding values of $\boldsymbol{\alpha}_k$ and W_k are listed in Table 6.7. The quadrature in Eq. (6.51) is exact if $f(x, y)$ is a polynomial of the degree indicated.

Degree of $f(x, y)$	Point	$\boldsymbol{\alpha}_k$	W_k
(a) Linear	a	1/3, 1/3, 1/3	1
(b) Quadratic	a	1/2, 0 , 1/2	1/3
	b	1/2, 1/2, 0	1/3
	c	0, 1/2 , 1/2	1/3
(c) Cubic	a	1/3, 1/3, 1/3	−27/48
	b	1/5, 1/5, 3/5	25/48
	c	3/5. 1/5 , 1/5	25/48
	d	1/5, 3/5 , 1/5	25/48

Table 6.7

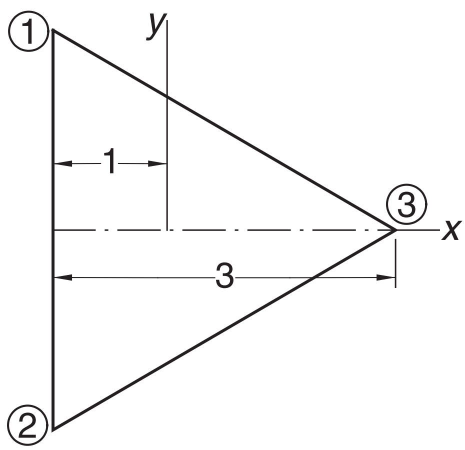
■ `triangleQuad`

The function `triangleQuad` computes $\int\int_A f(x, y)\, dx\, dy$ over a triangular region using the cubic formula – case (c) in Fig. 6.10. The triangle is defined by its corner coordinate arrays `xc` and `yc`, where the coordinates are listed in a *counterclockwise* order around the triangle.

```
## module triangleQuad
''' integral = triangleQuad(f,xc,yc).
    Integration of f(x,y) over a triangle using
    the cubic formula.
    {xc},{yc} are the corner coordinates of the triangle.
'''
from numpy import array,dot
```

```
def triangleQuad(f,xc,yc):
    alpha = array([[1.0/3, 1.0/3.0, 1.0/3.0],  \
                   [0.2, 0.2, 0.6],            \
                   [0.6, 0.2, 0.2],            \
                   [0.2, 0.6, 0.2]])
    W = array([-27.0/48.0 ,25.0/48.0, 25.0/48.0, 25.0/48.0])
    x = dot(alpha,xc)
    y = dot(alpha,yc)
    A = (xc[1]*yc[2] - xc[2]*yc[1]      \
       - xc[0]*yc[2] + xc[2]*yc[0]      \
       + xc[0]*yc[1] - xc[1]*yc[0])/2.0
    sum = 0.0
    for i in range(4):
        sum = sum + W[i] * f(x[i],y[i])
    return A*sum
```

EXAMPLE 6.16



Evaluate $I = \int\int_A f(x, y)\, dx\, dy$ over the equilateral triangle shown, where[7]

$$f(x, y) = \frac{1}{2}(x^2 + y^2) - \frac{1}{6}(x^3 - 3xy^2) - \frac{2}{3}$$

Use the quadrature formulas for (1) a quadrilateral and (2) a triangle.

Solution of Part (1) Let the triangle be formed by collapsing corners 3 and 4 of a quadrilateral. The corner coordinates of this quadrilateral are $\mathbf{x} = [-1,\ -1,\ 2,\ 2]^T$ and $\mathbf{y} = \left[\sqrt{3},\ -\sqrt{3},\ 0,\ 0\right]^T$. To determine the minimum required integration order for an exact result, we must examine $f\,[x(\xi, \eta), y(\xi, \eta)]\ |J(\xi, \eta)|$, the integrand in Eqs.

[7] This function is identical to the Prandtl stress function for torsion of a bar with the cross section shown; the integral is related to the torsional stiffness of the bar. See, for example, S. P. Timoshenko and J. N. Goodier, *Theory of Elasticity*, 3rd ed. (McGraw-Hill, 1970).

(6.44). Because $|J(\xi, \eta)|$ is biquadratic and $f(x, y)$ is cubic in x, the integrand is a polynomial of degree 5 in x. Therefore, third-order integration will suffice. The program used for the computations is similar to the one in Example 6.15:

```
#!/usr/bin/python
## example6_16a
from gaussQuad2 import *
from numpy import array
from math import sqrt

def f(x,y):
    return (x**2 + y**2)/2.0           \
           - (x**3 - 3.0*x*y**2)/6.0  \
           - 2.0/3.0

x = array([-1.0,-1.0,2.0,2.0])
y = array([sqrt(3.0),-sqrt(3.0),0.0,0.0])
m = eval(raw_input(''Integration order ==> ''))
print ''Integral ='', gaussQuad2(gaussNodes,f,x,y,m)
raw_input(''\nPress return to exit'')
```

Here is the output:

```
Integration order ==> 3
Integral = -1.55884572681
```

Solution of Part (2) The following program utilizes `triangleQuad`:

```
#!/usr/bin/python
# example6_16b
from numpy import array
from math import sqrt
from triangleQuad import *

def f(x,y):
    return (x**2 + y**2)/2.0           \
           - (x**3 - 3.0*x*y**2)/6.0  \
           - 2.0/3.0

xCorner = array([-1.0, -1.0, 2.0])
yCorner = array([sqrt(3.0), -sqrt(3.0), 0.0])
print ''Integral ='',triangleQuad(f,xCorner,yCorner)
raw_input(''Press return to  exit'')
```

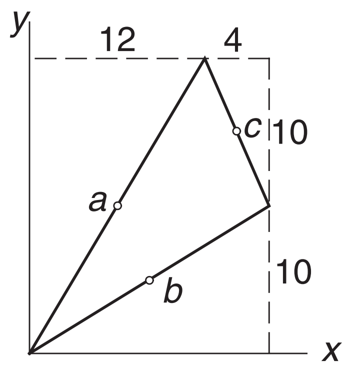
Because the integrand is a cubic, this quadrature is also exact, the result being

```
Integral = -1.55884572681
```

Note that only four function evaluations were required when using the triangle formulas. In contrast, the function had to be evaluated at nine points in part (1).

EXAMPLE 6.17

The corner coordinates of a triangle are $(0, 0)$, $(16, 10)$, and $(12, 20)$. Compute $\int\int_A \left(x^2 - y^2\right)\, dx\, dy$ over this triangle.



Solution Because $f(x, y)$ is quadratic, quadrature over the three integration points shown in Fig. 6.10(b) will be sufficient for an "exact" result. The integration points lie in the middle of each side; their coordinates are $(6, 10)$, $(8, 5)$, and $(14, 15)$. The area of the triangle is obtained from Eq. (6.49):

$$A = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ 0 & 16 & 12 \\ 0 & 10 & 20 \end{vmatrix} = 100$$

From Eq. (6.51) we get

$$\begin{aligned} I &= A\sum_{k=a}^{c} W_k f(x_k, y_k) \\ &= 100\left[\frac{1}{3}f(6, 10) + \frac{1}{3}f(8, 5) + \frac{1}{3}f(14, 15)\right] \\ &= \frac{100}{3}\left[(6^2 - 10^2) + (8^2 - 5^2) + (14^2 - 15^2)\right] = 1800 \end{aligned}$$

PROBLEM SET 6.3

1. Use Gauss–Legendre quadrature to compute

$$\int_{-1}^{1}\int_{-1}^{1}(1 - x^2)(1 - y^2)\, dx\, dy$$

2. Evaluate the following integral with Gauss–Legendre quadrature:

$$\int_{y=0}^{2}\int_{x=0}^{3} x^2 y^2\, dx\, dy$$

3. Compute the approximate value of

$$\int_{-1}^{1}\int_{-1}^{1} e^{-(x^2+y^2)}\,dx\,dy$$

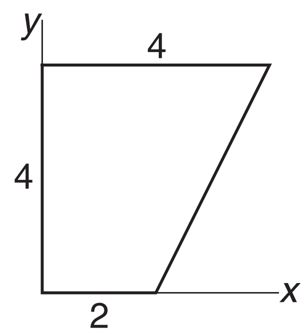
with Gauss–Legendre quadrature. Use integration order (a) 2 and (b) 3. (The "exact" value of the integral is 2.230 985.)

4. Use third-order Gauss–Legendre quadrature to obtain an approximate value of

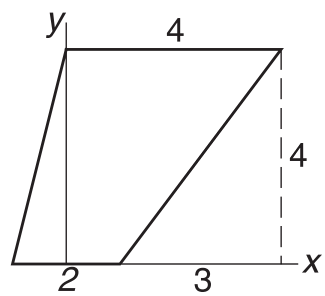
$$\int_{-1}^{1}\int_{-1}^{1} \cos\frac{\pi(x-y)}{2}\,dx\,dy$$

(The "exact" value of the integral is 1.621 139.)

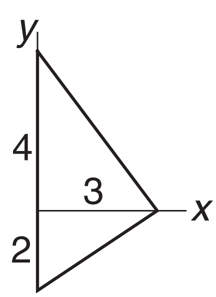
5. Map the integral $\int\int_A xy\,dx\,dy$ from the quadrilateral region shown to the "standard" rectangle, and then evaluate it analytically.



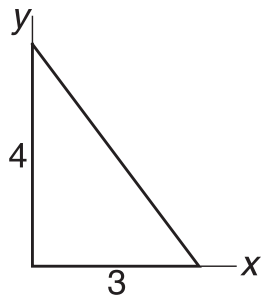
6. Compute $\int\int_A x\,dx\,dy$ over the quadrilateral region shown by first mapping it into the "standard" rectangle and then integrating analytically.



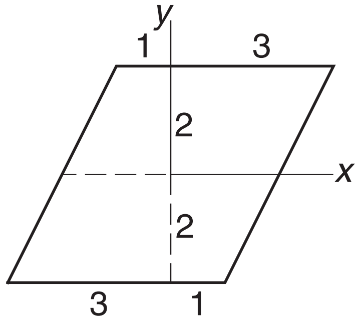
7. Use quadrature to compute $\int\int_A x^2\,dx\,dy$ over the triangle shown.



8. Evaluate $\int\int_A x^3\,dx\,dy$ over the triangle shown in Prob. 7.
9. Use quadrature to evaluate $\int\int_A (3-x)y\,dx\,dy$ over the region shown. Treat the region as (a) a triangular element and (b) a degenerate quadrilateral.

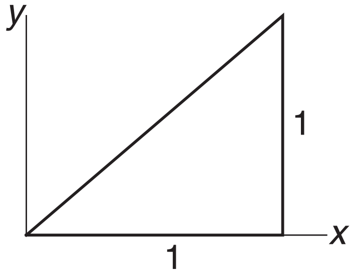


10. Evaluate $\int\int_A x^2 y\,dx\,dy$ over the triangle shown in Prob. 9.
11. ■



Evaluate $\int\int_A xy(2-x^2)(2-xy)\,dx\,dy$ over the region shown.

12. ■ Compute $\int\int_A xy\exp(-x^2)\,dx\,dy$ over the region shown in Prob. 11 to four decimal places.
13. ■

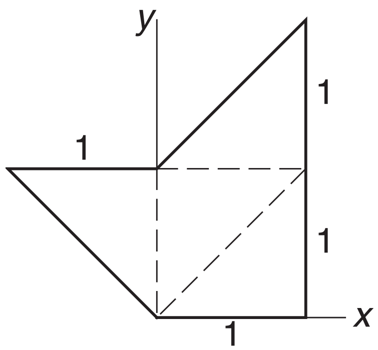


Evaluate $\int\int_A (1-x)(y-x)y\,dx\,dy$ over the triangle shown.

14. ■ Estimate $\int\int_A \sin\pi x\,dx\,dy$ over the region shown in Prob. 13. Use the cubic integration formula for a triangle. (The exact integral is $1/\pi$.)

15. ■ Compute $\int\int_A \sin \pi x \sin \pi (y - x)\, dx\, dy$ to six decimal places, where A is the triangular region shown in Prob. 13. Consider the triangle as a degenerate quadrilateral.

16. ■



Write a program to evaluate $\int\int_A f(x, y)\, dx\, dy$ over an irregular region that has been divided into several triangular elements. Use the program to compute $\int\int_A xy(y - x)\, dx\, dy$ over the region shown.