1. Adaptive integration

1.1. Adaptive Simpson’s rule. Adaptive Simpson’s method, also called adaptive Simpson’s rule, is a method of numerical integration proposed by G.F. Kuncir in 1962. Adaptive Simpson’s method uses an estimate of the error we get from calculating a definite integral using Simpson’s rule. If the error exceeds a user-specified tolerance, the algorithm calls for subdividing the interval of integration in two and applying adaptive Simpson’s method to each subinterval in a recursive manner. The technique is usually much more efficient than composite Simpson’s rule since it uses fewer function evaluations in places where the function is well-approximated by a cubic function. A criterion for determining when to stop subdividing an interval, suggested by J.N. Lyness, is

\[
\frac{1}{15} |S(a, c) + S(c, b) - S(a, b)| < \epsilon,
\]

where \([a, b]\) is an interval with midpoint \(c\), \(S(a, b)\), \(S(a, c)\), and \(S(c, b)\) are the estimates given by Simpson’s rule on the corresponding intervals and \(\epsilon\) is the desired tolerance for the interval. Simpson’s rule is an interpolatory quadrature rule which is exact when the integrand is a polynomial of degree three or lower. Using Richardson extrapolation, the more accurate Simpson estimate for five function values is combined with the less accurate estimate for three function values by applying the correction (see Fig. 1). The thus obtained estimate is exact for polynomials of degree five or less. Now we will work out the details to obtain the criterion (1) and to justify the statement about exactness for the polynomials of degree \(\leq 5\). Indeed, since the Simpson rule is obtained from the trapezoidal rule by one step of Richardson extrapolation, it has error expansion of the form

\[
I(f) := \int_a^b f(x)dx = S(a, b) + \alpha_4 h^4 + \alpha_6 h^6 + \ldots,
\]

where \(h = b - a\), \(S(a, b) = \frac{1}{6} h[f(a) + 4f(c) + f(b)]\), and the coefficients \(\alpha_n\) are of the form

\[
\alpha_n = \gamma_n (f^{(n)}(a) - f^{(n)}(b)).
\]

Making the step twice as small we get

\[
I(f) := \int_a^b f(x)dx = S(a, c) + S(c, b) + \alpha_4 \frac{h^4}{16} + \alpha_6 \frac{h^6}{64} + \ldots.
\]
Multiplying Eq. (3) by 16 and subtracting Eq. (2) from it we obtain
\[ 15I(f) = 15[S(a, c) + S(c, b)] + [S(a, c) + S(c, b) - S(a, b)] + \hat{\alpha}_6 h^6 + \ldots. \]

Therefore
\[ (4) \quad I(f) = S(a, c) + S(c, b) + \frac{1}{15}[S(a, c) + S(c, b) - S(a, b)] + \beta_6 h^6 + \ldots. \]

Comparing Eqs. (3) and (4) we see that the error in Eq. (3) is approximately given by
\[ \frac{1}{15}[S(a, c) + S(c, b) - S(a, b)]. \]
Therefore, Eq. (1) is justified. Furthermore, the coefficient \( \beta_n \) is of the form
\[ \beta_n = \delta_n(f^{(n)}(a) - f^{(n)}(b)). \]

Hence, the five point integration rule given by
\[ (5) \quad Q(f) = S(a, c) + S(c, b) + \frac{1}{15}[S(a, c) + S(c, b) - S(a, b)] \]

is exact on all polynomials of degree \( \leq 5 \).

The program AdaptiveSimpson.c implements adaptive Simpson’s method. This is a recursive algorithm refining intervals whenever the error tolerance is not satisfied. To avoid infinite recursion, the maximal recursion depth is introduced. Integration nodes of \( f(x) = \sin(1 - 30x^2) \) over the interval \([0, 1]\) with the tolerance \( \epsilon = 1.0e - 4 \) is shown in Fig. 2. Totally there are 109 nodes. They are shown both on the graph and on the \( x \)-axis.

**Figure 2.** Integration nodes of \( f(x) = \sin(1 - 30x^2) \) over the interval \([0, 1]\) with the tolerance \( \epsilon = 1.0e - 4 \) used by adaptive Simpson’s rule. Totally there are 109 nodes. They are shown both on the graph and on the \( x \)-axis.
1.2. Clenshaw-Curtis quadrature. This quadrature was preceded by the closely related Fejer quadrature. The advantage of the Clenshaw-Curtis quadrature over the Gaussian quadrature is that it can be made adaptive easily, the nodes are known analytically and can be computed using the fast Fourier transform. Suppose we need to compute the integral

\[ I(f) := \int_{-1}^{1} f(x)dx. \]

We approximate \( I(f) \) as

\[ \int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} \sum_{k=0}^{n} n c_k T_k(x)dx, \]

where \( n \) next to the sum means that the first and the last terms are divided by two. The polynomial

\[ f_n(x) = \sum_{k=0}^{n} n c_k T_k(x) \]

is the Chebyshev interpolation of the second kind, where there collocation points are the extrema of \( T_n(x) \). In this case,

\[ c_k = \frac{2}{n} \sum_{j=0}^{n} n f(x_j)T_k(x_j), \quad x_j = \cos \left( \frac{j\pi}{n} \right), \quad j = 0, 1, \ldots, n. \]

Therefore, the coefficients \( c_k \) are computed using the composite trapezoidal rule. Hence,

\[ \int_{-1}^{1} f(x)dx \approx \sum_{k=0}^{n} n c_k \int_{-1}^{1} T_k(x)dx = \sum_{k=0}^{n} w_k f(x_k) \]

where

\[ x_k = \cos \left( \frac{k\pi}{n} \right), \quad k = 0, 1, \ldots, n, \]

\[ w_k = \frac{g_k}{n} \left( 1 - \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{b_j}{4j^2 - 1} \cos \left( \frac{jk\pi}{n} \right) \right). \]

The values of constants \( g_k \) and \( b_j \) are: \( g_0 = 1, n g_k = 2, 1 \leq k \leq n, b_{n/2} = 1, b_j = 2 \) otherwise. The last values of \( g \) and \( b \) depends on whether \( n \) is odd or even. (Check this!)

References