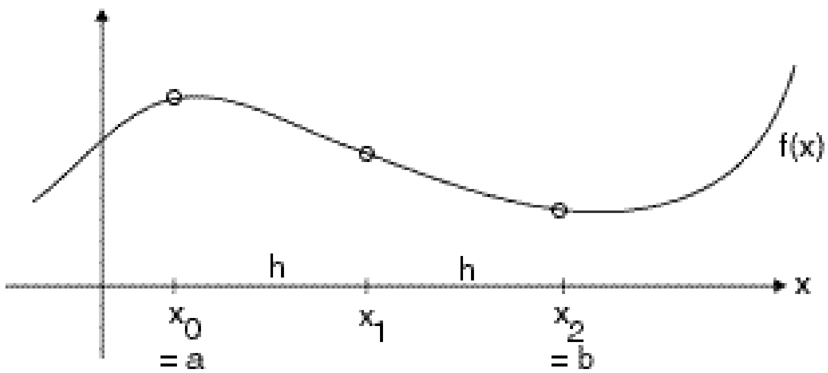
COMPUTER SCIENCE 349A

Handout Number 38

ADAPTIVE QUADRATURE
Section 22.3 of the 6th ed. and the 7th ed.

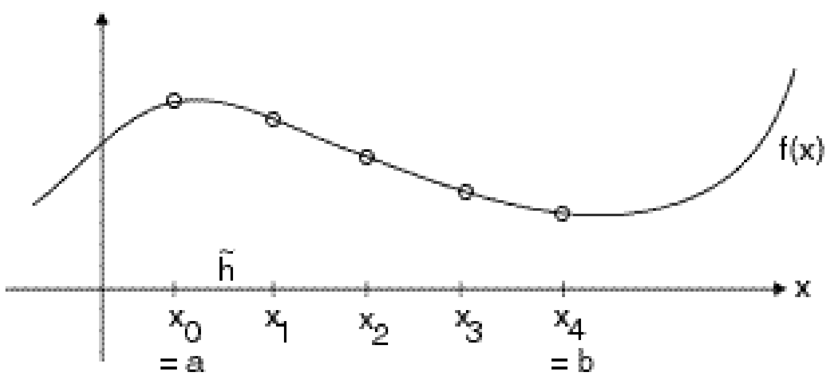
Let S_1 denote the (non-composite) Simpson's rule approximation to $\int_a^b f(x)dx$.



Then

(1) $$\int_a^b f(x)dx = S_1 - \frac{h^5}{90} f^{(4)}(\mu), \quad \text{where } h = \frac{b-a}{2} \text{ and } a < \mu < b.$$

Let S_2 denote the composite Simpson's rule approximation using two applications of Simpson's rule.



Then

(2) $$\int_a^b f(x)dx = S_2 - \frac{b-a}{180}\tilde{h}^4 f^{(4)}(\tilde{\mu}), \quad \text{where } \tilde{h} = \frac{b-a}{4} \text{ and } a < \tilde{\mu} < b$$
$$= S_2 - \frac{h^5}{(16)(90)} f^{(4)}(\tilde{\mu}), \quad \text{since } \tilde{h} = \frac{h}{2} \text{ and } b-a = 2h.$$

<u>Question</u>: How can the values of S_1 and S_2 be used to estimate the accuracy of S_2? To do this, we need to use S_1 and S_2 to <u>estimate the error term</u> for S_2, which (from (2)) is

(3) $$-\frac{h^5}{(16)(90)} f^{(4)}(\tilde{\mu}) \; .$$

In order to do this, we assume that

(4) $$f^{(4)}(\mu) = f^{(4)}(\tilde{\mu}) \, .$$

(Note that if *h* is very small, this is likely a good assumption, and will lead to an accurate estimate of the truncation error of S_2.) Using (4) in (2), from (1) and (2) we obtain

$$S_1 - \frac{h^5}{90} f^{(4)}(\mu) = S_2 - \frac{h^5}{(16)(90)} f^{(4)}(\mu) \, ,$$

from which it follows that

$$-\frac{h^5}{90} f^{(4)}(\mu) = \frac{16}{15}(S_2 - S_1) \, ,$$

and thus from (3), an estimate of the truncation error of S_2 is

$$-\frac{h^5}{(16)(90)} f^{(4)}(\tilde{\mu}) \approx \frac{1}{15}(S_2 - S_1) \, .$$

Consequently, (2) becomes

(5) $$\int_a^b f(x)dx = S_2 - \frac{h^5}{(16)(90)} f^{(4)}(\tilde{\mu}) \approx \underbrace{S_2}_{\substack{\text{approximation} \\ \text{to the integral}}} + \underbrace{\frac{1}{15}(S_2 - S_1)}_{\substack{\text{estimate of the} \\ \text{truncation error}}} \; ,$$

from which it follows that

(6) $$\left| \int_a^b f(x)dx - S_2 \right| \approx \frac{1}{15} \left| S_2 - S_1 \right| \; .$$

How to use this error estimate: Suppose you want to estimate $\int_a^b f(x)dx$ to within an accuracy of ε. If you compute S_1 and S_2 and it is the case that $|S_2 - S_1| < 15\varepsilon$, then from (6) we have

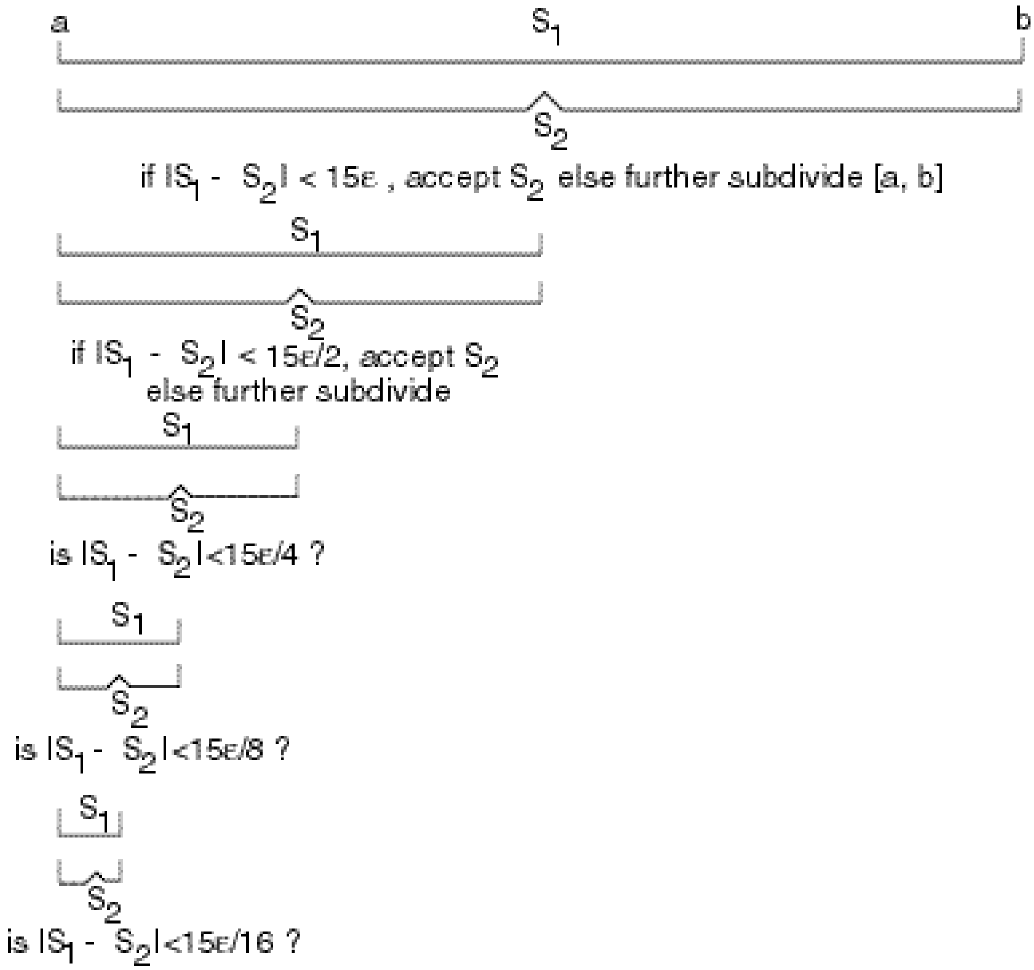
$$\left| \int_a^b f(x)dx - S_2 \right| < \varepsilon \tag{7}$$

and S_2 is the desired approximation. On the other hand, if $|S_2 - S_1|$ is not $< 15\varepsilon$, then you must select a smaller value for h and repeat the procedure.

Note 1. The rightmost part of (5) is the result of applying Richardson's Extrapolation to S_1 and S_2, but it is just S_2 and not that extrapolated value that is used here to approximate the integral.

Note 2. A recursive algorithm for the above procedure is given on page 640 of the 6th ed. or page 642 of the 7th ed.

THE ADAPTIVE QUADRATURE ALGORITHM for approximating $\int_a^b f(x)dx$ to an accuracy of ε :



The algorithm proceeds as follows. The interval $[a,b]$ is repeatedly halved, and on each successive subinterval the two Simpson's rule approximations S_1 and S_2 are computed. This continues, working with the leftmost subinterval of $[a,b]$, until values S_1 and S_2 are obtained that are sufficiently close together that S_2 can be accepted as the approximation of the integral on some subinterval of $[a,b]$.

For example, in the above diagram, if the last two values of S_1 and S_2 satisfy $|S_1 - S_2| < 15\varepsilon/16$, then this value of S_2 is accepted as the approximation to $\int_a^c f(x)dx$, where $c = a + \dfrac{b-a}{16}$, and this approximation has an error of at most $\varepsilon/16$. In order to find approximations to the integral on the rest of $[a,b]$, the algorithm returns to the last (that is, the leftmost) subinterval on which no approximation to the integral has yet been determined, and continues computing values of S_1 and S_2, and further subdividing this subinterval if necessary.

This continues until eventually the entire subinterval $[a,b]$ is subdivided into many subintervals, possibly of many different lengths, and a value S_2 has been accepted as the approximation to the integral on each little subinterval. The approximation to $\int_a^b f(x)dx$ is the sum of all of these values S_2 (and the sum of the errors of all of these approximations will be $\le \varepsilon$, which is the desired accuracy). The length of each subinterval for which a value S_2 is accepted depends on the shape of the graph of $f(x)$. In those parts of $[a,b]$ where $f(x)$ is oscillatory, a very small value of *h* (that is, a very small subinterval) will be required in order that the computed approximation S_2 is sufficiently accurate.

The MATLAB function *quad* can be used to approximate

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

using a recursive adaptive quadrature algorithm based on Simpson's rule. The function $f(x)$ can be coded as a MATLAB function M-file, for example,

```
function y = f(x)
y = ......  ;
```

and <u>must</u> accept a <u>vector</u> x as an argument (recall that that means you must use ./ .* .^ and so on instead of just / * ^). Execution of

```
[Q, fnc] = quad (@f, a, b, tol )
```

results in the computation of an approximation to the above integral using *tol* as the criterion for testing absolute error. The value of the computed approximation is returned as *Q*, and the value of *fnc* is equal to the number of function evaluations of f(x) used to compute this approximation.

Other adaptive quadrature algorithms are available in MATLAB; see *help quad*.

Example.

Use the MATLAB function *quad* to approximate

$$\int_{0.1}^{2} \sin(1/x)dx$$

using tol = 10^{-5} and determine the number of function evaluations required. Use *format long* .

****

```
function y=f(x)
y=sin(1./x);

format long

[Q,fnc]=quad('f',0.1,2,1e-5)

Q =

   1.145574937627094


fnc =

    49
```

****

Note that if this same problem is solved using the composite Trapezoidal rule, it requires over 4000 function evaluations.