which closely approximates the actual error

$$\left| \int_0^{\pi/2} \sin x \, dx - 1.000134585 \right| = 0.000134585,$$

even though $D_x^4 \sin x = \sin x$ varies significantly in the interval $(0, \pi/2)$. ■

When the approximations in (4.38) differ by more than 15ε, we can apply the Simpson's rule technique individually to the subintervals $[a, (a+b)/2]$ and $[(a+b)/2, b]$. Then we use the error estimation procedure to determine if the approximation to the integral on each subinterval is within a tolerance of $\varepsilon/2$. If so, we sum the approximations to produce an approximation to $\int_a^b f(x)\,dx$ within the tolerance ε.

If the approximation on one of the subintervals fails to be within the tolerance $\varepsilon/2$, then that subinterval is itself subdivided, and the procedure is reapplied to the two subintervals to determine if the approximation on each subinterval is accurate to within $\varepsilon/4$. This halving procedure is continued until each portion is within the required tolerance.

Problems can be constructed for which this tolerance will never be met, but the technique is usually successful, because each subdivision typically increases the accuracy of the approximation by a factor of 16 while requiring an increased accuracy factor of only 2.

Algorithm 4.3 details this Adaptive quadrature procedure for Simpson's rule, although some technical difficulties arise that require the implementation to differ slightly from the preceding discussion. For example, in Step 1 the tolerance has been set at 10ε rather than the 15ε figure in Inequality (4.38). This bound is chosen conservatively to compensate for error in the assumption $f^{(4)}(\xi) \approx f^{(4)}(\tilde{\xi})$. In problems where $f^{(4)}$ is known to be widely varying, this bound should be decreased even further.

It is a good idea to include a margin of safety when it is impossible to verify accuracy assumptions.

The procedure listed in the algorithm first approximates the integral on the leftmost subinterval in a subdivision. This requires the efficient storing and recalling of previously computed functional evaluations for the nodes in the right half subintervals. Steps 3, 4, and 5 contain a stacking procedure with an indicator to keep track of the data that will be required for calculating the approximation on the subinterval immediately adjacent and to the right of the subinterval on which the approximation is being generated. The method is easier to implement using a recursive programming language.

ALGORITHM
4.3

Adaptive Quadrature

To approximate the integral $I = \int_a^b f(x)\,dx$ to within a given tolerance:

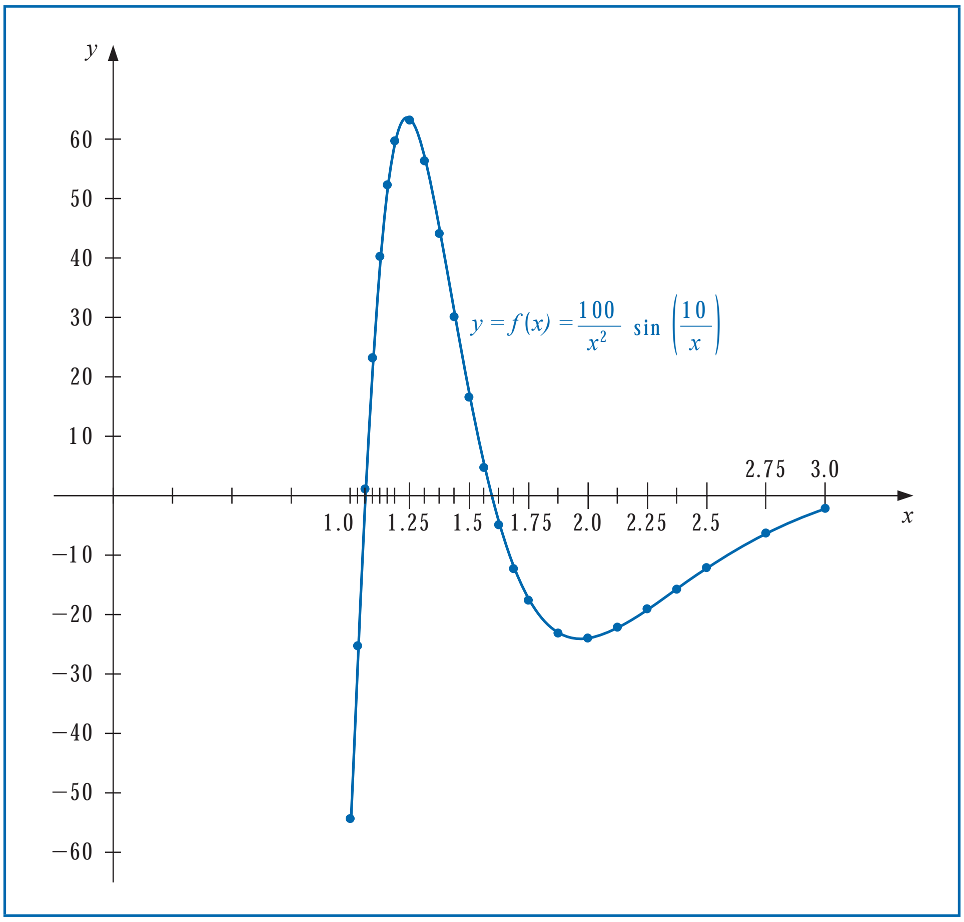
```
INPUT   endpoints $a, b$; tolerance $TOL$; limit $N$ to number of levels.

OUTPUT   approximation $APP$ or message that $N$ is exceeded.

Step 1  Set $APP = 0$;
            $i = 1$;
            $TOL_i = 10\,TOL$;
            $a_i = a$;
            $h_i = (b-a)/2$;
            $FA_i = f(a)$;
            $FC_i = f(a + h_i)$;
            $FB_i = f(b)$;
            $S_i = h_i(FA_i + 4FC_i + FB_i)/3$;   (Approximation from Simpson's
                                                    method for entire interval.)
            $L_i = 1$.
```

```
Step 2  While $i > 0$ do Steps 3–5.
    Step 3  Set $FD = f(a_i + h_i/2)$;
                $FE = f(a_i + 3h_i/2)$;
                $S1 = h_i(FA_i + 4FD + FC_i)/6$;   (Approximations from Simpson's
                                                    method for halves of subintervals.)
                $S2 = h_i(FC_i + 4FE + FB_i)/6$;
                $v_1 = a_i$;   (Save data at this level.)
                $v_2 = FA_i$;
                $v_3 = FC_i$;
                $v_4 = FB_i$;
                $v_5 = h_i$;
                $v_6 = TOL_i$;
                $v_7 = S_i$;
                $v_8 = L_i$.
    Step 4  Set $i = i - 1$.   (Delete the level.)
    Step 5  If $|S1 + S2 - v_7| < v_6$
              then set $APP = APP + (S1 + S2)$
              else
                  if ($v_8 \geq N$)
                    then
                        OUTPUT ('LEVEL EXCEEDED');   (Procedure fails.)
                        STOP.
                    else   (Add one level.)
                        set $i = i + 1$;   (Data for right half subinterval.)
                          $a_i = v_1 + v_5$;
                          $FA_i = v_3$;
                          $FC_i = FE$;
                          $FB_i = v_4$;
                          $h_i = v_5/2$;
                          $TOL_i = v_6/2$;
                          $S_i = S2$;
                          $L_i = v_8 + 1$;
                        set $i = i + 1$;   (Data for left half subinterval.)
                          $a_i = v_1$;
                          $FA_i = v_2$;
                          $FC_i = FD$;
                          $FB_i = v_3$;
                          $h_i = h_{i-1}$;
                          $TOL_i = TOL_{i-1}$;
                          $S_i = S1$;
                          $L_i = L_{i-1}$.
Step 6  OUTPUT ($APP$);   ($APP$ approximates $I$ to within $TOL$.)
        STOP.
```

■

Illustration The graph of the function $f(x) = (100/x^2)\sin(10/x)$ for x in $[1, 3]$ is shown in Figure 4.14. Using the Adaptive Quadrature Algorithm 4.3 with tolerance 10^{-4} to approximate $\int_1^3 f(x)\,dx$ produces -1.426014, a result that is accurate to within 1.1×10^{-5}. The approximation required that Simpson's rule with $n = 4$ be performed on the 23 subintervals whose endpoints are shown on the horizontal axis in Figure 4.14. The total number of functional evaluations required for this approximation is 93.

Figure 4.14



The largest value of h for which the standard Composite Simpson's rule gives 10^{-4} accuracy is $h = 1/88$. This application requires 177 function evaluations, nearly twice as many as Adaptive quadrature. □

Adaptive quadrature can be performed with the *Quadrature* command in the *Numerical-Analysis* subpackage of Maple's *Student* package. In this situation the option $adaptive = true$ is used. For example, to produce the values in the Illustration we first load the package and define the function and interval with

$$f := x \rightarrow \frac{100}{x^2} \cdot \sin\left(\frac{10}{x}\right);\ a := 1.0;\ b := 3.0$$

Then give the *NumericalAnalysis* command

$Quadrature(f(x), x = a..b, adaptive = true, method = [simpson, 10^{-4}], output = information)$