```
function s = aquad(f,a,b,tol,ff)
%function s = aquad(f,a,b,tol,ff)
%
%
% Adaptive quadrature with Simpson's rule.
%
%             s  =  int_a^b  f(x) dx  +  error,
%
%   where |error| < tol (hopefully).
%
% f is a string which is the name of the function (m-file) f
% a, b, and tol are as above
% ff is a 3-tuple [f(a);f((a+b)/2);f(b)], and is an optional argument



h = (b-a)/2;


% On first call, function values at endpoints are not known, so we compute them.
% (on subsequent recursions they are passed in by parent)

if nargin < 5,
  ff = feval(f,[a;a+h;b]);
end



%Simpsons rule for interval [a,b]

ps = (h/3)*([1 4 1]*ff);



%Simpson's rule for [a,a+h] and [a+h,b] (requires 2 new fcn evals)

f1 = feval(f,[a+h/2 ; b-h/2]);
ff1 = [ff(1);f1(1);ff(2)];  ff2 = [ff(2);f1(2);ff(3)];
s1 = (h/6)*([1 4 1]*ff1);   s2 = (h/6)*([1 4 1]*ff2);



% If error estimate is small enough: quit,
% otherwise: cut the interval in half and call aquad for each half
% with the error tolerance halved.

if abs(s1+s2-ps) < 8*tol, %analysis says factor s/b 15, we cautiously take 8
  s = s1+s2;
else
  s = aquad(f,a,a+h,tol/2,ff1) + aquad(f,a+h,b,tol/2,ff2);
end
```