function integral = recursive\_adaptive\_simpson(f, a, b, tolerance, level,max\_level)

%Adaptive simpson algorithm for approximating an integral

h = (b- a)/2;

approximation = h\*(f(a) + 4\*f(a+h) + f(b))/3;

%FD = f(a + h/2);

%FE = f(a + 3\*h/2);

S1 = h\*(f(a) + 4\*f(a + h/2) + f(a+h))/6;

S2 = h\*(f(a+h) + 4\*f(a + 3\*h/2) + f(b))/6;

if abs(S1+S2-approximation) < tolerance

integral = S1+S2;

elseif level >= max\_level

disp("procedure failed")

else

m = (a+b)/2;

integral = recursive\_adaptive\_simpson(f, a, m, tolerance, level+1,max\_level) + recursive\_adaptive\_simpson(f, m, b, tolerance, level+1,max\_level);

end

end

used Wikipedia pseudocode for quadrature

A screenshot of a cell phone

Description automatically generated

ans =

5.113839797275945

This was my attempt at the algorithm in the book but there was something wrong:

(DO NOT GRADE THIS ONE)

function integral = adaptive\_simpson(f, a, b, tolerance, max\_level)

%Adaptive simpson algorithm for approximating an integral

v = zeros(8);

approximation = 0;

a\_vec = zeros(max\_level);

h\_vec = zeros(max\_level);

FA\_vec = zeros(max\_level);

FB\_vec = zeros(max\_level);

FC\_vec = zeros(max\_level);

S\_vec = zeros(max\_level);

L\_vec = zeros(max\_level);

TOL\_vec = zeros(max\_level);

i = 1;

TOL\_vec(i) = 10\*tolerance;

a\_vec(i) = a;

h\_vec(i) = (b- a)/2;

FA\_vec(i) = f(a);

FC\_vec(i) = f (a + h\_vec(i));

FB\_vec(i) = f(b);

S\_vec(i) = h\_vec(i)\*(FA\_vec(i) + 4\*FC\_vec(i) + FB\_vec(i))/3; %simspon method approximation for the whole interval

L\_vec(i) = 1;

while i>0

FD = f(a\_vec(i) + h\_vec(i)/2);

FE = f(a\_vec(i) + 3\*h\_vec(i)/2);

%Approximations from Simpson's method for halves of subintervals

S1 = h\_vec(i)\*(FA\_vec(i) + 4\*FD + FC\_vec(i))/6;

S2 = h\_vec(i)\*(FC\_vec(i) + 4\*FE + FB\_vec(i))/6;

v(1) = a\_vec(i);

v(2) = FA\_vec(i);

v(3) = FC\_vec(i);

v(4) = FB\_vec(i);

v(5) = h\_vec(i);

v(6) = TOL\_vec(i);

v(7) = S\_vec(i);

v(8) = L\_vec(i);

i = i -1; %Delete the level

if abs(S1 + S2 - v(7)) < v(6)

approximation = approximation + (S1 + S2);

break;

elseif v(8) >= max\_level

disp("Level Exceeded"); %procedure fails

break

else

i = i+1; %Data for the right half of the subinterval

a\_vec(i) = v(1) + v(5);

FA\_vec(i) = v(3);

FC\_vec(i) = FE;

FB\_vec(i) = v(4);

h\_vec(i) = v(5)/2;

TOL\_vec(i) = v(6)/2;

S\_vec(i) = S2;

L\_vec(i) = v(8) + 1;

i = i + 1; %Data for the left half of the subinterval

a\_vec(i) = v(1);

FA\_vec(i) = v(2);

FC\_vec(i) = FD;

FB\_vec(i) = v(3);

h\_vec(i) = h\_vec(i-1);

TOL\_vec(i) = TOL\_vec(i-1);

S\_vec(i) = S1;

L\_vec(i) = L\_vec(i-1);

end

end

integral = approximation;

end