Jake Traut

4/10/16

CSCI 3656 Problem Set 11

1. Richardson extrapolation from h = 0.1 to get f’(1) for f(x) = e^x + 0.5 and using center differences

True value e = 2.71828183.

Richardson extrapolation formula :

Q = (2^n \* F(h/2) – F(h))/(2^n – 1) where F(h) = (f(x+h) – f(x-h))/2h for x = 1, F(.1) = 2.72281456 and F(.1/2) = 2.71941459

Then Q = (2^2 \* 2.71941459 – 2.72281456)/3 = 2.71828126

Then error = |e – Q |= .00000056

Compared to centered difference formula error = |e – F(.1)| = .00453273

Thus extrapolation produces a much closer estimate.

1. Trapezoid rule
   1. Using h = 0.1, integral approximation = 3.7341
   2. Using h = 0.2, integral approximation = 4.0940

function integral = trapezoid(x,y,x1,x2,h)

%x and y vectors holding corresponding data points

%x1 and x2 lower and upper bounds for integration

%h is step size

n = length(x);

if(n ~= length(y))

integral = ('do not have pairs of x and y values');

return

end

%check for evenly spaced data

for i = 1:(n-2)

space1 = abs(x(i+1) - x(i));

space2 = abs(x(i+2) - x(i+1));

if(space1 ~= space2)

integral = ('data not evenly spaced');

end

end

%check if the given bounds are valid

lowermatch = 0;

uppermatch = 0;

for i = 1:n

if(x1 == x2)

integral = ('not valid limits of integration');

return

end

if(x1 == x(i))

lowermatch = 1;

end

if(x2 == x(i))

uppermatch = 1;

end

end

if(lowermatch == 0 || uppermatch == 0)

integral = ('not valid limits of integration');

return

end

%check for valid step size

multiple = h / space1;

if(ceil(multiple) ~= floor(multiple))

integral = ('h not a valid step size');

return

end

%good to go, compute the integral via trapezoidal method

integral = 0;

multiple = round(multiple);

for i=1:multiple:n

step = i+multiple;

if(x(i) >= x1 && x(i) < x2 && step <= n)

trapezoid = h/2\*(y(step)+y(i));

integral = integral + trapezoid;

end

end

end

1. Simpson’s 1/3 rule
   1. Using h = 0.1, integral approximation = 3.7314
   2. Using h = 0.2, integral approximation = 3.7314

function integral = simpsonsthird(x,y,x1,x2,h)

%x and y vectors holding corresponding data points

%x1 and x2 lower and upper bounds for integration

%h is step size

n = length(x);

if(n ~= length(y))

integral = ('do not have pairs of x and y values');

return

end

%check for evenly spaced data

for i = 1:(n-2)

space1 = abs(x(i+1) - x(i));

space2 = abs(x(i+2) - x(i+1));

if(space1 ~= space2)

integral = ('data not evenly spaced');

end

end

%check if the given bounds are valid

lowermatch = 0;

uppermatch = 0;

for i = 1:n

if(x1 == x2)

integral = ('not valid limits of integration');

return

end

if(x1 == x(i))

lowermatch = 1;

end

if(x2 == x(i))

uppermatch = 1;

end

end

if(lowermatch == 0 || uppermatch == 0)

integral = ('not valid limits of integration');

return

end

%check for valid step size

multiple = h / space1;

if(ceil(multiple) ~= floor(multiple))

integral = ('h not a valid step size');

return

end

%check if have correct number of points to divide to thirds

if(rem(n,3) ~= 0)

integral = ('invalid data for Simpsons 1/3 rule');

return

end

%good to go, compute the integral via Simpson's 1/3 rule

integral = 0;

one = 1;

multiple = round(multiple);

for i=1:n

two = one+multiple;

three = two+multiple;

if(x(i) == x1)

two = i+multiple;

three = two+multiple;

simpsons = (h/3)\*(y(i) + 4\*y(two) + y(three));

integral = integral + simpsons;

one = three;

elseif(three <= n && x(one) > x1 && x(three) <= x2)

simpsons = (h/3)\*(y(one) + 4\*y(two) + y(three));

integral = integral + simpsons;

one = three;

end

end

end

1. Calculus; function f(x) = e^x + 0.5
   1. ∫1.81.0 f(x)dx => (e^x + 0.5x)|1.81.0 => (e^1.8 + 0.5\*1.8) – (e^1.0 + 0.5) = 3.7313656
   2. Method comparisons
      1. Trapezoidal: Using h = 0.1, |3.7341 – 3.7313656| = .0027344

Using h = 0.2, |4.0940 – 3.7313656|= .3626344

* + 1. Simpson’s 1/3: Using h = 0.1 and h = 0.2, |3.7314 – 3.7313656| = .0000344

As one could expect, Simpson’s method should produce better approximations than trapezoidal if the function has curvature (b/c it uses parabolas to fit the curve rather than linear trapezoids to approximate). As for getting an approximation with better precision, you would want to use smaller h values, to consider more of the data into your calculations.