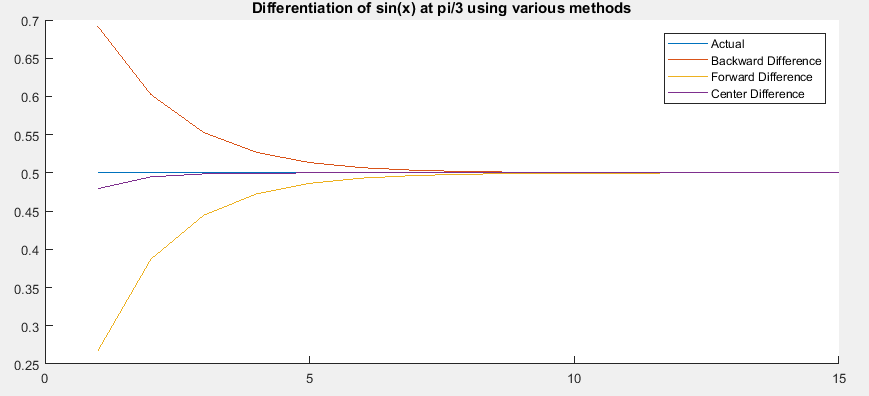
Numerical Computing II

Homework 4

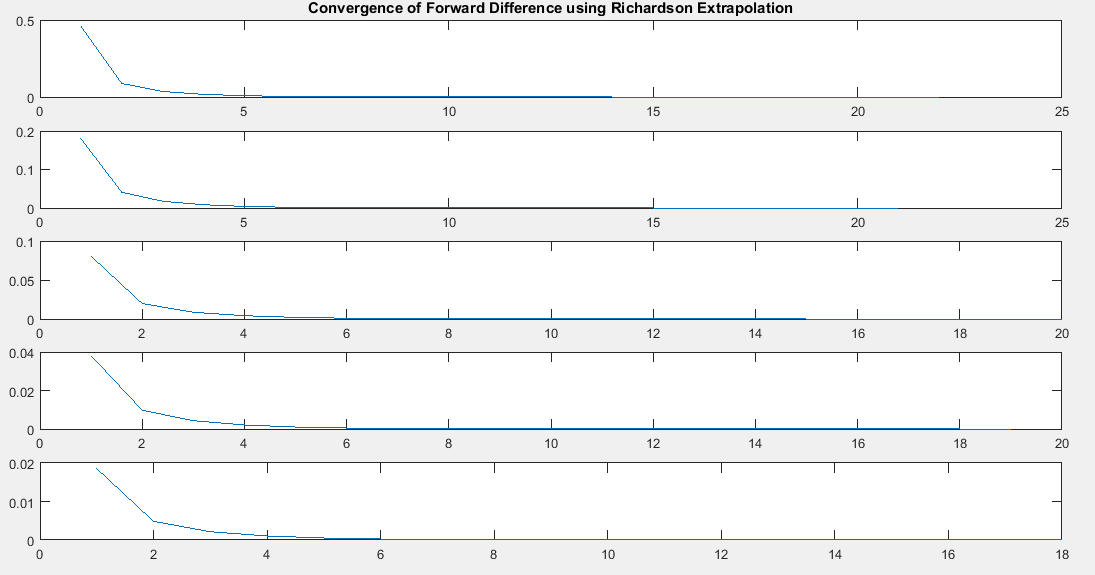
Samuel V

Some Visuals:

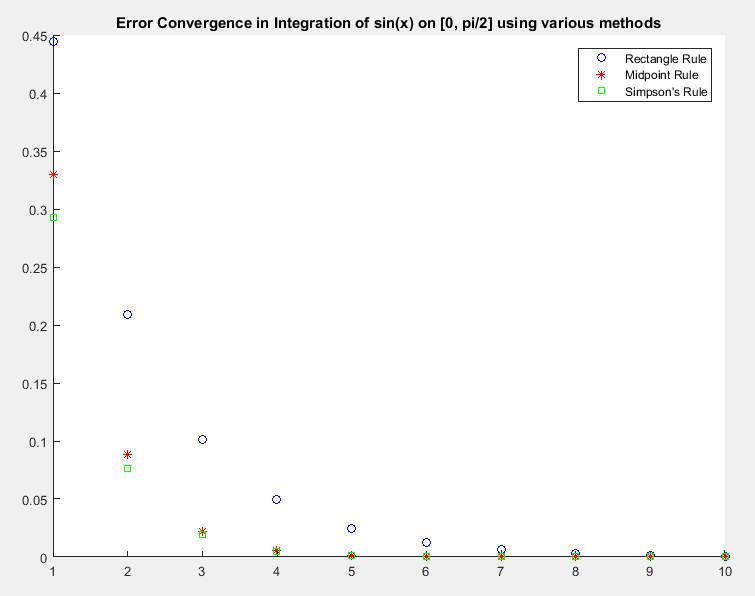
Convergence of derivatives to actual value

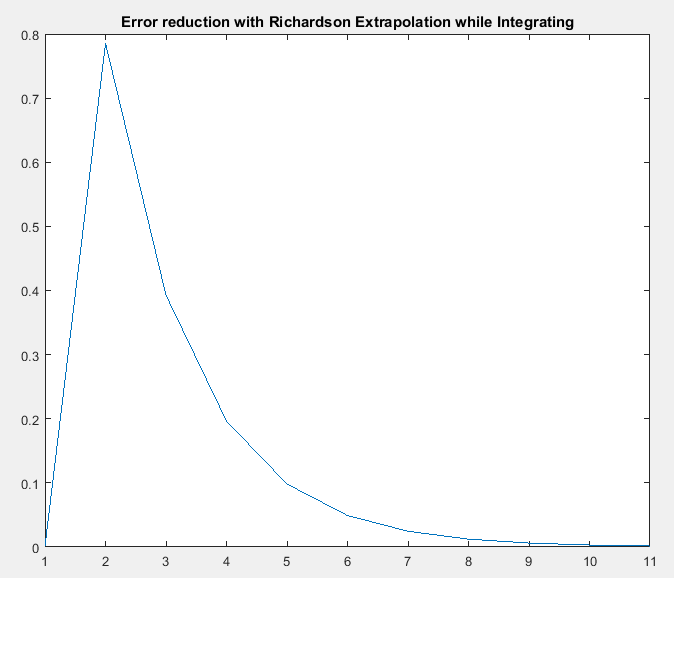


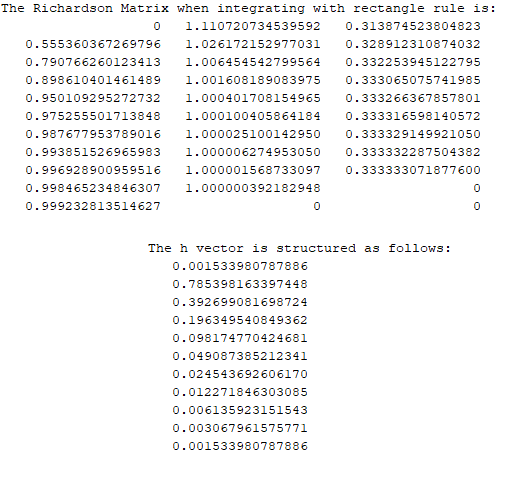
Forward difference error convergence up to n=5.



Error Convergence in basic integration methods







The Code!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This script runs all of the numerical routines! %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

differentiate;

pause;

figure(2)

hold on;

errs = 0;

errMat = 0;

for i = 1:1:15

errs = differentiateRichardson(i);

if i <= 5

subplot(5,1,i)

x = 1:1:length(errs);

plot(x,errs);

if i == 1

title('Convergence of Forward Difference using Richardson Extrapolation');

end

end

end

hold off;

pause;

figure(3)

hold on;

for i = 1:1:10

integrate(i);

end

hold off

legend('Rectangle Rule','Midpoint Rule','Simpson''s Rule');

title('Error Convergence in Integration of sin(x) on [0, pi/2] using various methods');

pause;

integrateRichardson;

pause;

%

% Differentiate sine at pi/3 using various methods.

%

% Get 1-15 to to work with later

domain = 1:1:15;

% Preallocate some vectors to hold derivative values

bd = zeros(15,1);

fd = zeros(15,1);

cd = zeros(15,1);

% Allocate actual derivative value

actual = zeros(15,1);

for i = 1:1:15

actual(i) = 1/2;

end

% Calculate and allocate all derivatives

for i = 1:1:15

bd(i) = ((sin(pi/3)-sin((pi/3) - (2^-i)))/(2^-i));

fd(i) = ((sin((pi/3) + (2^-i)) - sin(pi/3))/(2^-i));

cd(i) = ((sin((pi/3) + (2^-i)) - sin((pi/3) - (2^-i)))/(2\*(2^-i)));

end

% Plot

figure(1)

hold on

plot(domain,actual)

plot(domain,bd)

plot(domain,fd)

plot(domain,cd)

legend('Actual','Backward Difference','Forward Difference', 'Center Difference');

title('Differentiation of sin(x) at pi/3 using various methods');

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Integrates with various quadrature rules using 2^n partitions. %

% sin(x) on [0, pi/2] %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function integrate(n)

% The actual value of the integral is 1.

truevalue = 1;

% Setup the number of partitions.

m = 2^n;

% Init vector to hold partition locations

part = zeros(m,1);

% Fill the partition vector

for i = 1:1:m

part(i) = i\*((pi/2)/m);

end

% Integrate using the rectangle rule.

areaR = 0;

for i = 1:1:m-1

areaR = areaR + ((part(i+1) - part(i))\*sin(part(i)));

end

errorR = 1 - areaR;

% Integrate using the midpoint rule.

areaM = 0;

for i = 1:1:m-1

areaM = areaM + ((part(i+1) - part(i))\*(((1/2)\*sin(part(i))) + ((1/2)\*sin(part(i+1)))));

end

errorM = 1 - areaM;

% Integrate using Simpson's rule.

areaS = 0;

for i = 1:1:m-1

areaS = areaS + ((part(i+1) - part(i))\*(((1/6)\*sin(part(i))) + (((4/6)\*sin((part(i)+part(i+1))/2) + (((1/6)\*sin(part(i+1))))))));

end

errorS = 1 - areaS;

figure(3)

plot(n, errorR,'bo');

plot(n, errorM,'r\*');

plot(n, errorS,'gs');

end

a = 0;

b = pi/2;

x = linspace(0, pi/2);

f = sin(x);

integer = 1;

for n = 0:10

m = 2^n;

h=(b-a)/m;

for i=1:m+1

x(i) = a+(i-1)\*h;

f(i) = sin(x(i));

end

rect(n+1) = sum(f(1:m))\*h;

recterror(n+1) = abs(integer - rect(n+1));

A(n+1) = sum(f(1:m))\*h;

end

A = A';

for i = 1:10

R(i) = (-1)\*(A(i) - 2\*A(i +1));

end

% One less iteration than previous operation

for j = 1:9

M(j) = (R(j) - 2\*R(j + 1))/(1 - (2^2));

end

% takes 10 iters

R(11) = 0;

M(10) = 0;

M(11) = 0;

for n = 1:10

h = [h;(b - a)/(2^n)];

end

R = R';

M = M';

% Construct a beautiful Richardson Matrix

Richardson = [A, R, M];

disp('The Richardson Matrix when integrating with rectangle rule is: ');

disp(Richardson);

disp('The h vector is structured as follows: ');

disp(h);

domain = 1:1:length(h);

figure(4)

plot(domain,h);

title('Error reduction with Richardson Extrapolation while Integrating');