Alan Wang

16.90 Pset #7

Plots and Code

1. 

Script:

%% Alan Wang

%16.90 Pset 7

%Nodal Point Interpolation

%% Setup

n = 8;

xi = 0;

xf = 1;

xj = linspace(xi,xf,n);

u1 = @u1fun;

u2 = @u2fun;

u3 = @u3fun;

%% Part a

points = 49\*n+1;

x = linspace(xi,xf,points);

u1\_values = u1(x);

u2\_values = u2(x);

u3\_values = u3(x);

v1 = nodalApprox(u1,xi,xf,n);

v2 = nodalApprox(u2,xi,xf,n);

v3 = nodalApprox(u3,xi,xf,n);

e1 = abs(u1\_values-v1);

e2 = abs(u2\_values-v2);

e3 = abs(u3\_values-v3);

e1max8 = max(e1);

e2max8 = max(e2);

e3max8 = max(e3);

%plots

figure(1)

plot(x,e1,x,e2,x,e3)

legend('e1','e2','e3')

xlabel('x')

ylabel('e(x)')

title('Pointwise Error between Interpolated Function and Actual Function')

%% Part b

N = [2,4,8,16,32,64];

deltax = zeros(1,length(N));

e1max = zeros(1,length(N));

e2max = zeros(1,length(N));

e3max = zeros(1,length(N));

for i = 1:length(N)

n = N(i);

deltax(i) = 1/(N(i)-1);

points = 49\*n+1;

x = linspace(xi,xf,points);

u1\_values = u1(x);

u2\_values = u2(x);

u3\_values = u3(x);

v1 = nodalApprox(u1,xi,xf,n);

v2 = nodalApprox(u2,xi,xf,n);

v3 = nodalApprox(u3,xi,xf,n);

e1 = abs(u1\_values-v1);

e2 = abs(u2\_values-v2);

e3 = abs(u3\_values-v3);

e1max(i) = max(e1);

e2max(i) = max(e2);

e3max(i) = max(e3);

end

%plots

figure(2)

loglog(deltax,e1max,deltax,e2max,deltax,e3max)

legend('e1','e2','e3')

xlabel('log(deltax)')

ylabel('log(max e(x))')

grid on

title('Maximum Error vs. Deltax')

Functions:

function vx = nodalApprox(fx,xi,xf,n)

%interpolate1D - approximates a function as a linear combination of linear

%nodal basis functions

% fx - function handle for function interpolating

% phi- function handle for basis functions: phifun

% xi - initial x

% xf - final x

% n - number of nodal points

xj = linspace(xi,xf,n);

dx = xj(2)-xj(1);

phi = @phifun;

phiCell = phi(xj,dx);

a = zeros(1,n);

points = 49\*n+1;

vx = zeros(1,points);

for i = 1:n

a(i) = fx(xj(i));

vx = vx + a(i)\*phiCell{i};

end

end

function u1 = u1fun(x)

%u1 piecewise function from 16.90 pset 7

% Detailed explanation goes here

n = length(x);

u1 = zeros(1,n);

for i = 1:n

if x(i)>=0 && x(i)<0.5

u1(i) = 5;

elseif x(i)>0.5 && x(i)<=1

u1(i) = 1;

end

end

end

function u2 = u2fun(x)

%u2 piecewise function from 16.90 pset 7

% Detailed explanation goes here

n = length(x);

u2 = zeros(1,n);

for i = 1:n

if x(i)>=0 && x(i)<0.5

u2(i) = exp(x(i));

elseif x(i)>0.5 && x(i)<=1

u2(i) = exp(2\*(x(i)-.25));

end

end

end

function u3 = u3fun(x)

%u3 piecewise function from 16.90 pset 7

% Detailed explanation goes here

u3 = 5\*(1-x.^3);

end

function phi = phifun(xj,dx)

%basis function

% xj - xj

% dx - change in nodal x

n = length(xj);

phi = cell(1,n);

points = n\*49+1;

x=linspace(0,1,points);

for i = 1:n

phi{i} = zeros(1,length(x));

if i == 1

ind = (x <= xj(i+1));

phi{i} = (xj(i+1)-x)/dx;

phi{i} = phi{i}.\*ind;

elseif i == n

ind = (x >= xj(i-1));

phi{i} = (x-xj(i-1))/dx;

phi{i} = phi{i}.\*ind;

else

for k = 1:length(x)

if x(k)>=xj(i-1) && x(k)<=xj(i)

phi{i}(k) = (x(k) - xj(i-1))/dx;

elseif x(k)>xj(i) && x(k)<=xj(i+1)

phi{i}(k) = (xj(i+1)-x(k))/dx;

else

phi{i}(k) = 0;

end

end

end

end

end