Text, letter

Description automatically generated

Text, letter

Description automatically generated

We have: a = 0, b = 0 and z = 1. We now test for the 6 conditions:

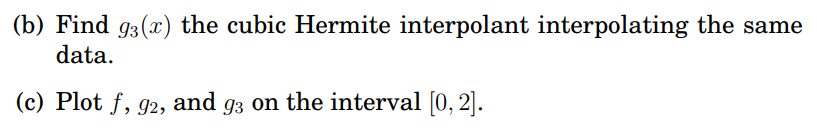


Since 6 conditions are satisfied, g(x): [0, 2] is a quadratic Hermite interpolant with a knot at z = 1 interpolating at 0 and 2

Text

Description automatically generated with medium confidence



  
Diagram

Description automatically generated with medium confidence

Where y1 is f, y2 is g2 and y3 is g3. Graph is made by Desmos. This is a closer look

Chart

Description automatically generated with medium confidence

Text, letter

Description automatically generated

Text, letter

Description automatically generated



Plugging in t = (x – a)/h, we have:



Compute the general form of the coefficients:







In exercise 1, we have a = 0, f(a) = 0, f’(a) = 0, b = 2, f(b) = 16, f’(b) = 32 and h = 2

=> 

=> 

=>

Plugging into p(t), we have:



We have get the same cubic Hermite interpolation like Exercise 1

Text, letter

Description automatically generated



For the first spline, we have a = 0, p1(a) = 0, p1’(a) = 0, z = 1, p1(z) = 0, p’(z) = 0 and h = 1





The final constraint, where z = 1, is given by



=> The corresponding quadratic Hermite interpolant g(x) in Bernstein form is:





=> 

=> as g2(x) the quadratic Hermite interpolation in Exercise 1

Text, letter

Description automatically generated

The Matlab code for quadhermite.m is  
function hermiteQuadCoeff = quadhermite(a, b, z, y1, y2, s1, s2)

c10 = y1;

c22 = y2;

d = 2;

hSpline1 = z - a;

hSpline2 = b - z;

derivativeA = s1;

derivativeB = s2;

c11 = (derivativeA \* hSpline1 / d) + c10;

c21 = c22 - (derivativeB \* hSpline2 / d);

c120 = (hSpline1 \* c21 + hSpline2 \* c11)/(hSpline1 + hSpline2);

hermiteQuadCoeff = [c10, c11, c120, c21, c22];

end

The result of the quadhermite function is:

A picture containing text

Description automatically generated

Text, letter

Description automatically generated

The Matlab code for quadhermite\_test.m is

vars = num2cell([0,2,1,0,16,0,32]);

[a,b,z,y1,y2,s1,s2] = deal(vars{:});

C = quadhermite(a,b,z,y1,y2,s1,s2);

t = 0:1/n:1;

t2 = 1:1/n:2;

B10 = (1-t).^2;

B11 = 2\*t.\*(1-t);

B12 = t.^2;

B20 = (1-t).^2;

B21 = t\*2.\*(1-t);

B22 = t.^2;

x2 = [a:(z-a)/n:z,z:(b-z)/n:b];

g2 = [C(1).\*B10 + C(2).\*B11 + C(3).\*B12, C(3).\*B20 + C(4).\*B21 + C(5).\*B22];

x3 = 2\*t;

B30 = (1-t).^3;

B31 = 3\*t.\*((1-t).^2);

B32 = 3\*(t.^2).\*(1-t);

B33 = t.^3;

g3 = [0.\*B30 + 0.\*B31 - 16/3.\*B32 + 16.\*B33];

f2 = x2.^4;

f3 = x3.^4;

subplot(1,2,1)

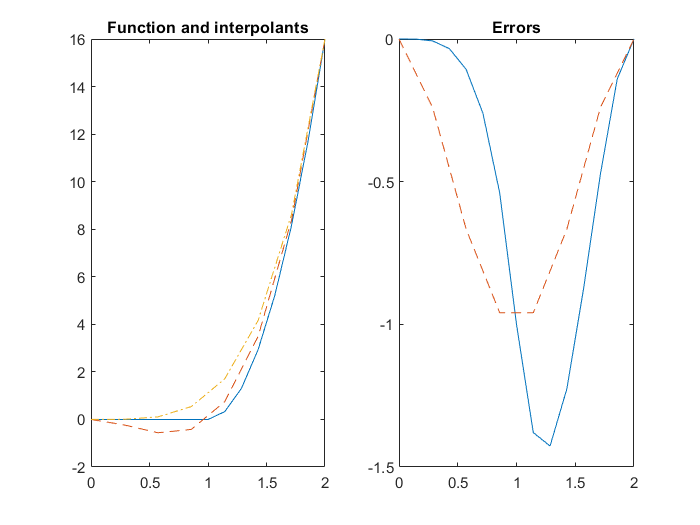
plot(x2,g2,x3,g3,'--',x3,f3,'-.')

title('Function and interpolants')

subplot(1,2,2)

plot(x2,g2-f2,x3,g3-f3,'--')

title('Errors')

The function, interpolants and the errors are in the graph below:  


Text

Description automatically generated

The Matlab code for drawing\_initials.m is

% My initials are N.X.B

% Drawing initial N

x = [1 1.5 2 2.5]; y = [1 3 1 3];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','b','LineWidth',2), plot(x,y,'o'), grid on

% Drawing initial X

x = [3 3.5 4 4.5]; y = [1, 1.25, 2.75, 3];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','g','LineWidth',2), plot(x,y,'o'), grid on

x = [3 3.5 4 4.5]; y = [3, 2.75, 1.25, 1];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','g','LineWidth',2), plot(x,y,'o'), grid on

% Drawing initial B

x = [5 5.5 5]; y = [1, 2, 3];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','r','LineWidth',2), plot(x,y,'o'), grid on

x = [5 6 6.5 5.5]; y = [1, 1, 1.5, 2];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','r','LineWidth',2), plot(x,y,'o'), grid on

x = [5 6 6.5 5.5]; y = [3, 3, 2.5, 2];

n = length(x);

t = 0:1:n-1;

tt = 0:.1:n-1;

xx = spline(t,x,tt); yy = spline(t,y,tt); hold on

plot(xx,yy','r','LineWidth',2), plot(x,y,'o'), grid on

The drawing of my initials N.X.B is given using Spline (I can also use Bezier curves but it requires many control points so I use the default Matlab spline function for simplicity)

