Numerical Analysis project 2 Joseph Pietrluongo

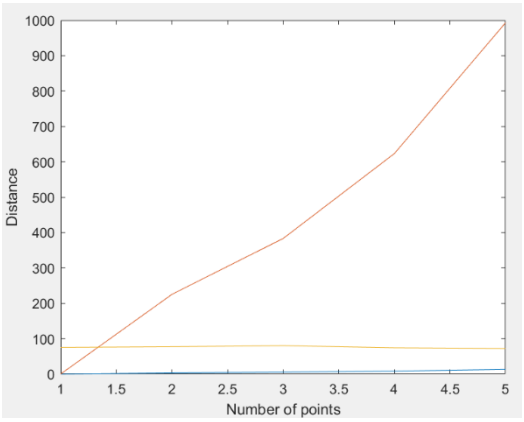
5901749

Input:

T	F(T)	F'(T)
0	0	75
3	225	77
5	383	80
8	623	74
13	993	72

The input provided produced a divided difference table below as the output. The output shown is in descending order in the terminating number for each order, which is the way the MATLAB code produces the output. The function output as the order increases in the difference in speed between intervals, as the order grows larger the change in speed becomes increasingly small.

N^{th} Order	Output
0^{th} order	0.00000
1^{st} order	75.00000
2^{nd} order	0.00000
3^{rd} order	0.22222
4^{th} order	-0.03111
5^{th} order	-0.00644
6^{th} order	0.00226
7^{th} order	-0.00091
8^{th} order	0.00013
9^{th} order	-0.00002



The bright orange line is the distance traveled over time/points

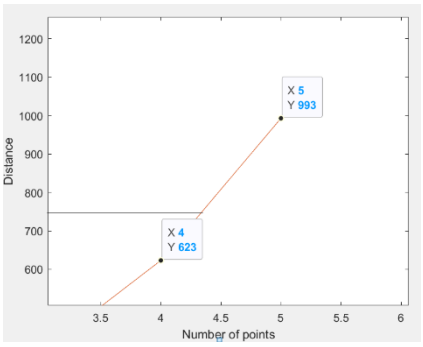
The light orange line is the speed

The blue line is Time.

Analysis:

T = 10	F(10) = 742.50

In the graph we can see that as time increases, the distances increase traveled increases, and at different rates. The x-axis as both the measurement of time and number of inputs, but as the interval of inputs varies it was easier and clearer to have it in measurement of input as all 3 graphs are overlaid. T=10 is not displayed on the graph but would appear in the interval between (4 < x < 5, 742.5). This result would make sense on the graph here:



The point T = 10, can be shown to be approximately where the line touches the graph. As 10 is closer to 8 than it is to 13 it is much closer to the 4^{th} point than the 5^{th}.

```
syms('OK', 'FLAG', 'N', 'I', 'X', 'Q', 'A', 'NAME', 'INP');
 syms('Z', 'K', 'J', 'OUP', 'XX', 'S','x','s','I1');
 TRUE = 1;
 FALSE = 0;
 fprintf(1,'This is Hermite interpolation.\n');
 OK = FALSE;
 while OK == FALSE
 fprintf(1,'Choice of input method:\n');
 fprintf(1,'1. Input entry by entry from keyboard\n');
 fprintf(1,'2. Input data from a text file\n');
 fprintf(1,'3. Generate data using a function F\n');
 fprintf(1,'Choose 1, 2, or 3 please\n');
 FLAG = input(' ');
 if FLAG == 1 | FLAG == 2 | FLAG == 3
 OK = TRUE;
 end
 end
 if FLAG == 1
 OK = FALSE;
 while OK == FALSE
 fprintf(1,'Input the number of data points minus 1\n');
 N = input(' ');
 if N > 0
 OK = TRUE;
 X = zeros(N+1);
 Q = zeros(2*N+2,2*N+2);
 for I = 0:N
 fprintf(1,'Input X(%d), F(X(%d)), and ', I, I);
 fprintf(1,'F''(X(%d)) on separate lines\n ', I);
 X(I+1) = input(' ');
 Q(2*I+1,1) = input(' ');
 Q(2*I+2,2) = input(' ');
 end
 else
 fprintf(1,'Number must be a positive integer\n');
 end
 end
 end
 if FLAG == 2
 fprintf(1,'Has a text file been created with the data in
three columns?\n');
 fprintf(1,'Enter Y or N\n');
 A = input(' ','s');
 if A == 'Y' | A == 'y'
 fprintf(1,'Input the file name in the form - ');
 fprintf(1,'drive:\\name.ext\n');
 fprintf(1,'for example:   A:\\DATA.DTA\n');
 NAME = input(' ','s');
 INP = fopen(NAME,'rt');
 OK = FALSE;
 while OK == FALSE
 fprintf(1,'Input the number of data points minus 1\n');
 N = input(' ');
 if N > 0
 X = zeros(N+1);
 Q = zeros(2*N+2,2*N+2);
 for I = 0:N
 X(I+1) = fscanf(INP, '%f',1);
 Q(2*I+1,1) = fscanf(INP, '%f',1);
 Q(2*I+2,2) = fscanf(INP, '%f',1);
 end
 fclose(INP);
 OK = TRUE;
 else
 fprintf(1,'Number must be a positive integer\n');
 end
 end
 else
 fprintf(1,'Please create the input file in three column ');
 fprintf(1,'form with the X values, F(X), and\n');
 fprintf(1,'derivative values in the corresponding
columns.\n');
 fprintf(1,'The program will end so the input file can ');
 fprintf(1,'be created.\n');
 OK = FALSE;
 end
 end
 if FLAG == 3
 fprintf(1,'Input the function F(x) in terms of x.\n');
 fprintf(1,'For example: sin(x)\n');
 s = input(' ');
 F = inline(s,'x');
 fprintf(1,'Input F''(x) in terms of x.\n');
 s = input(' ');
 FP = inline(s,'x');
 OK = FALSE;
 while OK == FALSE
 fprintf(1,'Input the number of data points minus 1\n');
 N = input(' ');
 if N > 0
 X = zeros(1,N+1);
 Q = zeros(2*N+2,2*N+2);
 for I1 = 0:N
 fprintf(1,'Input X(%d)\n', I1);
 X(I1+1) = input(' ');
 Q(2*I1+1,1) = F(X(I1+1));
 Q(2*I1+2,2) = FP(X(I1+1));
 end
 OK = TRUE;
 else
 fprintf(1,'Number must be a positive integer\n');
 end
 end
 end
 if OK == TRUE

% STEP 1
 Z = zeros(2*N+2);
 for I = 0:N
% STEP 2
 Z(2*I+1) = X(I+1);
 Z(2*I+2) = X(I+1);
 Q(2*I+2,1) = Q(2*I+1,1);
% STEP 3
 if I ~= 0
 Q(2*I+1,2) = (Q(2*I+1,1)-Q(2*I,1))/(Z(2*I+1)-Z(2*I));
 end
 end
% STEP 4
 K = 2*N+1;
 for I = 2:K
 for J = 2:I
 Q(I+1,J+1) = (Q(I+1,J)-Q(I,J))/(Z(I+1)-Z(I-J+1));
 end
 end
% STEP 5
 fprintf(1,'Choice of output method:\n');
 fprintf(1,'1. Output to screen\n');
 fprintf(1,'2. Output to text file\n');
 fprintf(1,'Please enter 1 or 2\n');
 FLAG = input(' ');
 if FLAG == 2
 fprintf(1,'Input the file name in the form -
drive:\\name.ext\n');
 fprintf(1,'for example:   A:\\OUTPUT.DTA\n');
 NAME = input(' ','s');
 OUP = fopen(NAME,'wt');
 else OUP = 1;
 end
 fprintf(OUP, 'HERMITE INTERPOLATING POLYNOMIAL\n\n');
 fprintf(OUP, 'The input data follows:\n');
 fprintf(OUP, '  X, F(X), F''(x)\n');
 for I = 0:N
 fprintf(OUP,'  %12.10e %12.10e
%12.10e\n',X(I+1),Q(2*I+1,1),Q(2*I+2,2));
 end
 fprintf(OUP, '\nThe Coefficients of the Hermite
Interpolation ');
 fprintf(OUP, 'Polynomial\n');
 fprintf(OUP, 'in order of increasing exponent follow:\n\n');
 for I = 0:K
 fprintf(OUP, '  %12.10e\n', Q(I+1,I+1));
 end
 fprintf(1,'Do you wish to evaluate this polynomial?\n');
 fprintf(1,'Enter Y or N\n');
 A = input(' ','s');
 if A == 'Y' | A == 'y'
 fprintf(1,'Enter a point at which to evaluate\n');
 XX = input(' ');
 S = Q(K+1,K+1)*(XX-Z(K));
 for I = 2:K
 J = K-I+1;
 S = (S+Q(J+1,J+1))*(XX-Z(J));
 end
 S = S + Q(1,1);
 fprintf(OUP, 'x-value and interpolated-value\n');
 fprintf(OUP, '  %12.10e %12.10e\n', XX, S);
 end
 if OUP ~= 1
 fclose(OUP);
 fprintf(1,'Output file %s created successfully\n',NAME);
 end
 end
```