HW 6

(MATH/CS 375)

Jacob Hurst

04/10/18

Problem 1:

Code:

function a = interpvan(x, y)

n = length(x);  
V = zeros(n,n);

for i = 1:n  
 V(i,:) = (x(i)\*ones(1,n)).^(0:n-1);  
end

if(isrow(y))  
 y = transpose(y);  
end

a = V\y;  
end

Code:

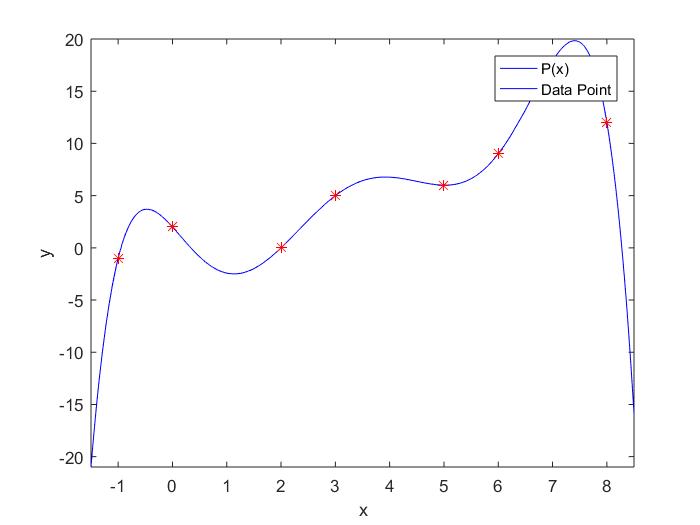
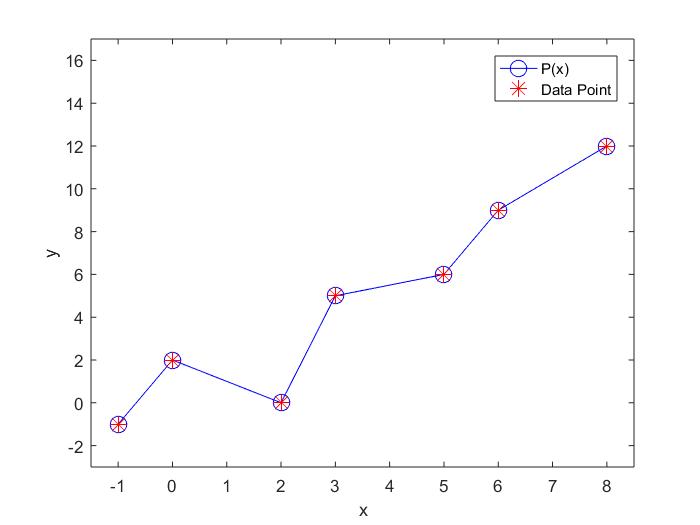
function ynew = vaninterp(a, xnew)

syms x;  
n = length(a);  
ynew = a(n)\*ones(size(xnew));

for i=(n-1):-1:1  
 ynew = (xnew.\*ynew) + a(i);  
end  
end

Problem 1:

1. Also substituted xnew with x to return the function P(x) for second plot, first plot is P(xnew).

Condition number:

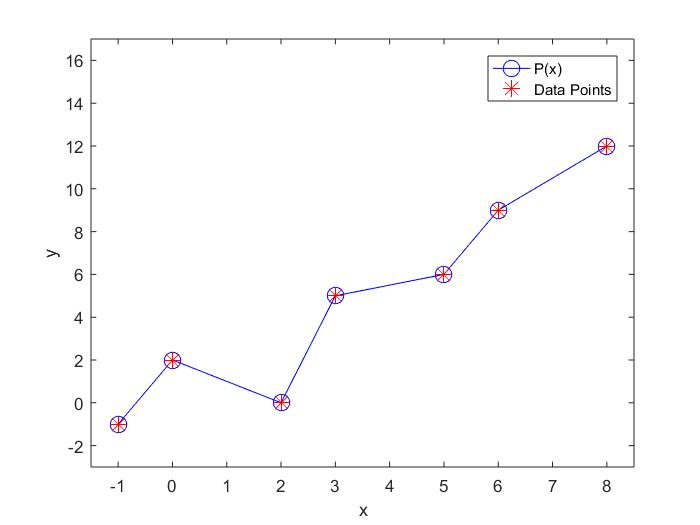
1.3265e+06

Residual:

5.7004e-13

Problem 1:

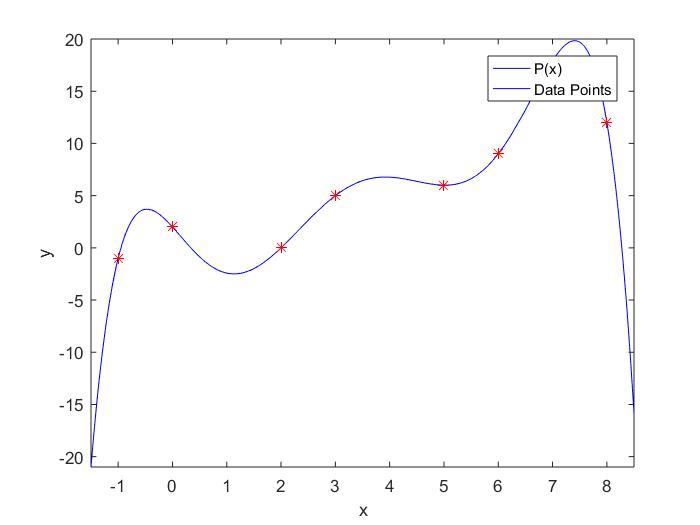
1. Took the same approach in newton’s method of plotting P(xnew) & P(x)

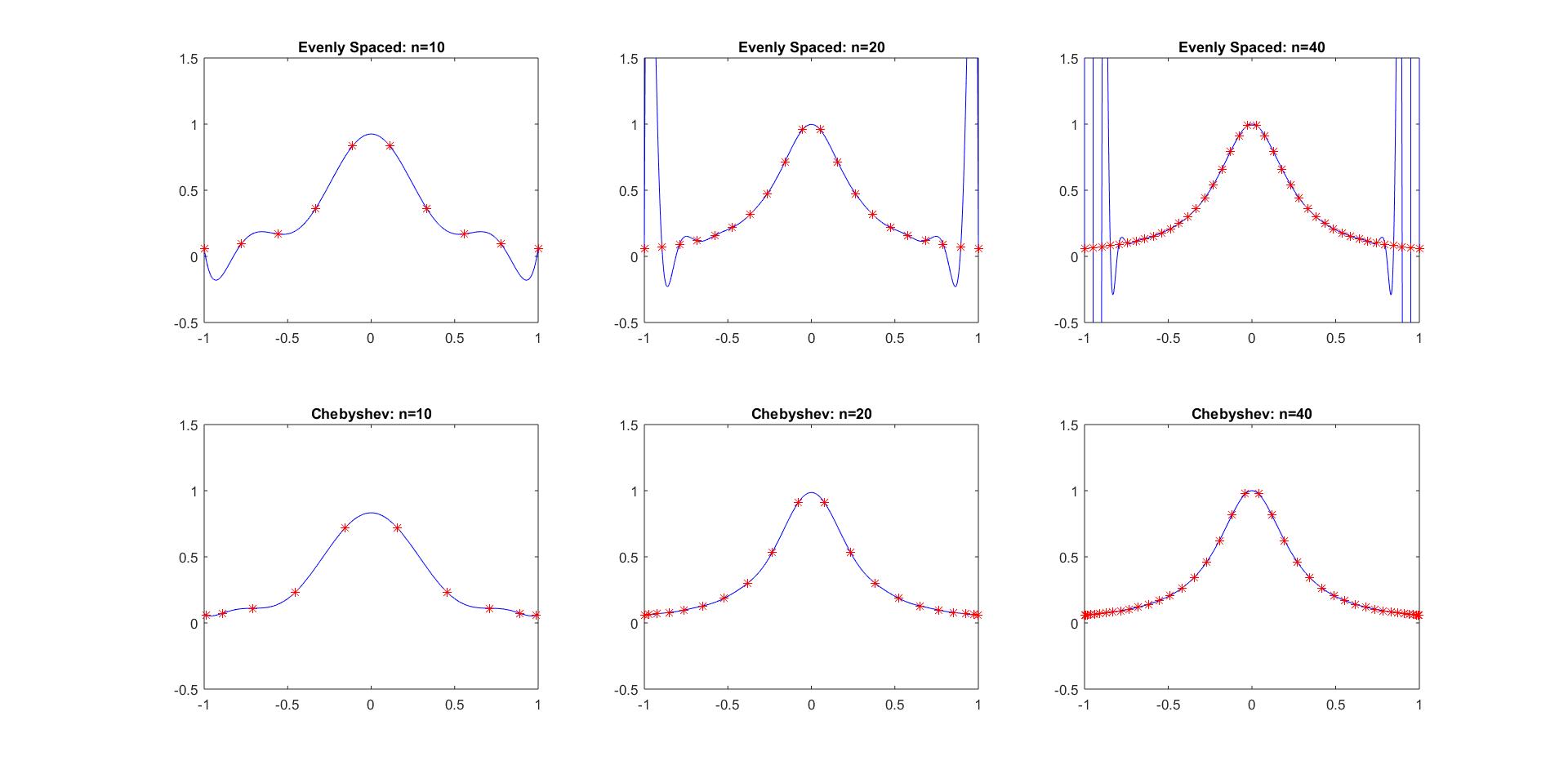


Residual:

6.2585e-14

In both cases, the residual is small, however, the residual of newton’s method is approximately a factor of 10 smaller.



Problem 2:

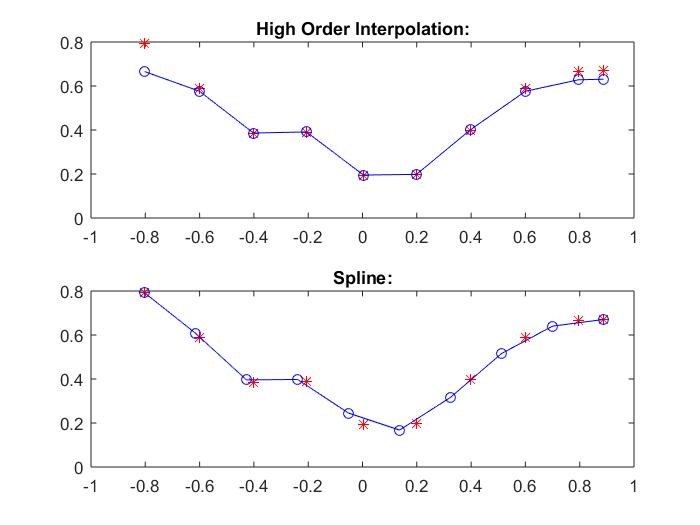
From the resulting plots, we can observe that the functions using evenly spaced points suffered from an increasingly large error around the bounds of the plot as the number of points used increased in size (Runge Phenomenon).

This is in part, due to the order of the polynomial, which is at most, n-1.

The functions using Chebyshev points did not suffer from this phenomenon, in fact, as the number of points used increased in size, the total error decreased.

The Chebyshev points minimized the maximum error.

Problem 3:



From the resulting plots, we can observe a greater error in the plot of the high order polynomial than that of the clamped cubic spline with zero end-slopes.

This error is, again, due to the polynomials degree (at most 9). We can also observe an increase of error around the edges of the high order interpolation plot.

The spline does not suffer from this same error (however, a couple points have slight errors) as it uses piecewise polynomial interpolation.

The total error in the spline is less than that of the high order interpolation.