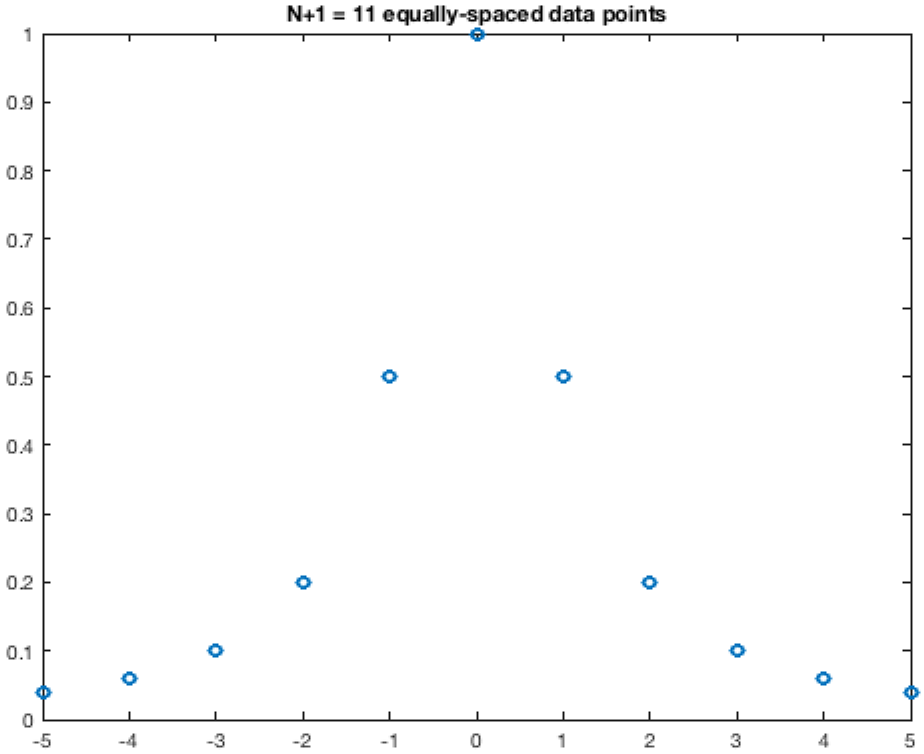
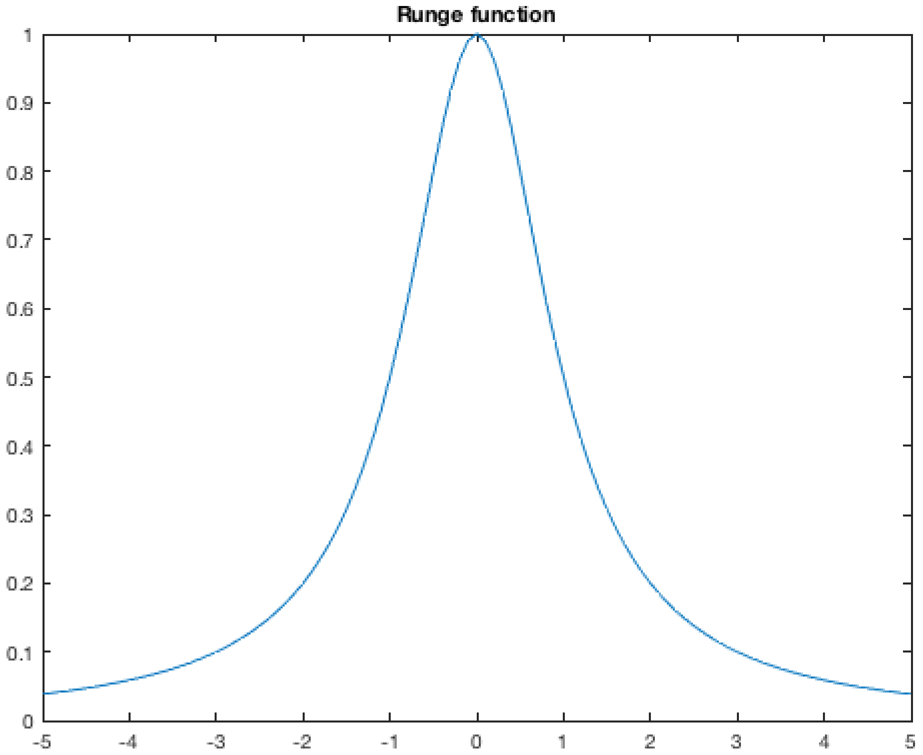
Mazen Alotaibi

Table of Contents

Problem 1

```
N = 10;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
plot(x,y,'o')
title('N+1 = 11 equally-spaced data points')
t = [-5:.1:5];
figure;
plot(t,f(t),'-')
title('Runge function')
```

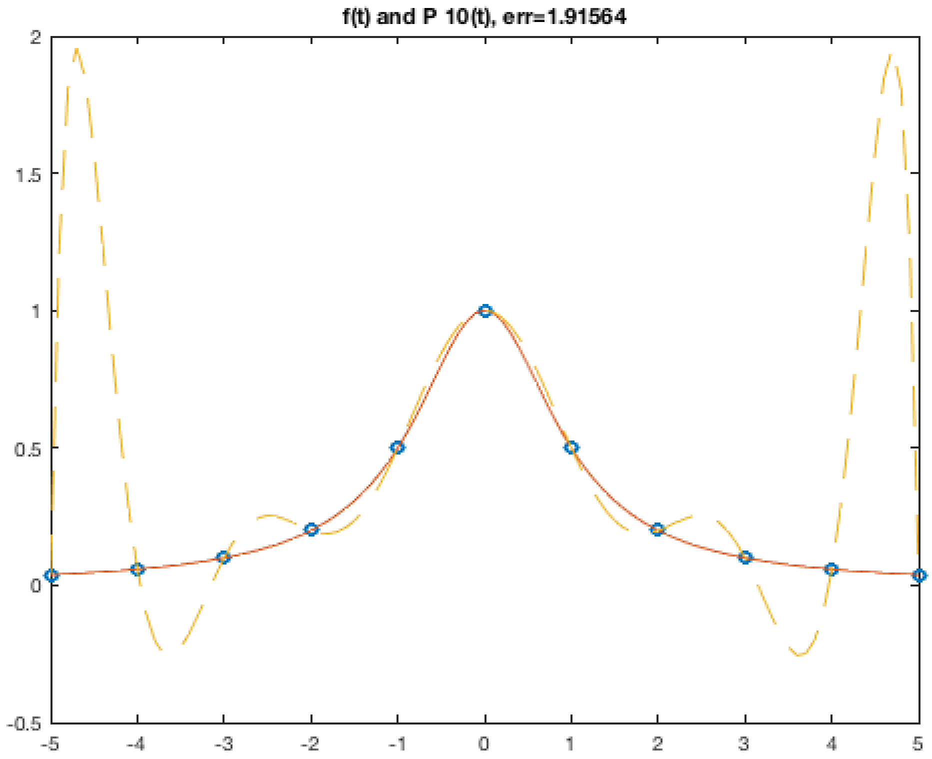

N+1 = 11 equally-spaced data points
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
-4
-3
-2
-1
0
1
2
3
4
5


Runge function
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
-4
-3
-2
-1
0
1
2
3
4
5

Problem 2

```
N = 10;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
t = [-5:.1:5];
PN = polyfit(x,y,N);
v = polyval(PN,t);
err = norm(f(t)-v,inf)
figure;
plot(x,y,'o',t,f(t),'-',t,v,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))
```

```
err =

    1.9156
```

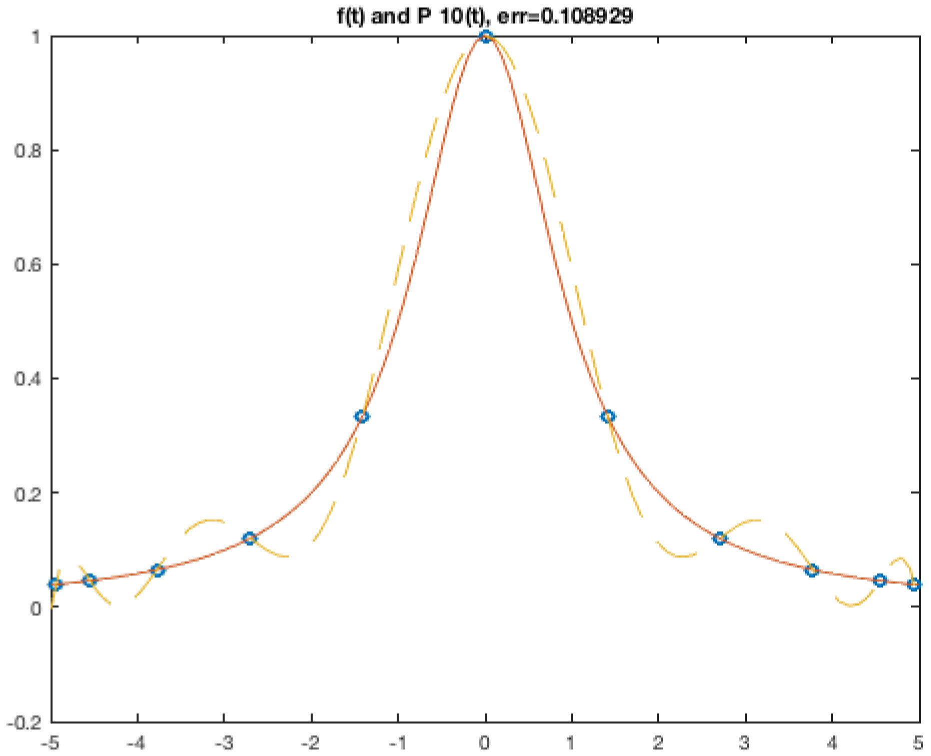


Problem 3

```
N=10;
K = N+1;
a=-5;
```

```
b=5;
xcheb=zeros(1,K);
for i=1:K
    xcheb(i)=(a+b)/2 + (b-a)/2 * cos( (i-.5)*pi/K );
end
ycheb = f(xcheb);
t = [-5:.1:5];
PNcheb = polyfit(xcheb,ycheb,N);
v = polyval(PNcheb,t);
err = norm(f(t)-v,inf)
figure;
plot(xcheb,ycheb,'o',t,f(t),'-',t,v,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))

% When comparing problem 2 and 3, problem 2 has an error equals to
 1.91564
% and problem3 has an error equals to 0.108929, problem 3 has less
 error
% compared to problem 2, which makes more accurate to plot.

% The dots of intersection for problem 2 are far from each other
 compared
% to problem 3's dots of intersection are closer to each other.

% The reason why problem 3 behaved better, it is because
 Interpolation at
% Chebychev nodes approximate better than using standard N-th Degree
% Interpolating Polynomial, as Interpolation at Chebychev nodes's
 results
% a polynomial approximation close to the minimax approximation.
```

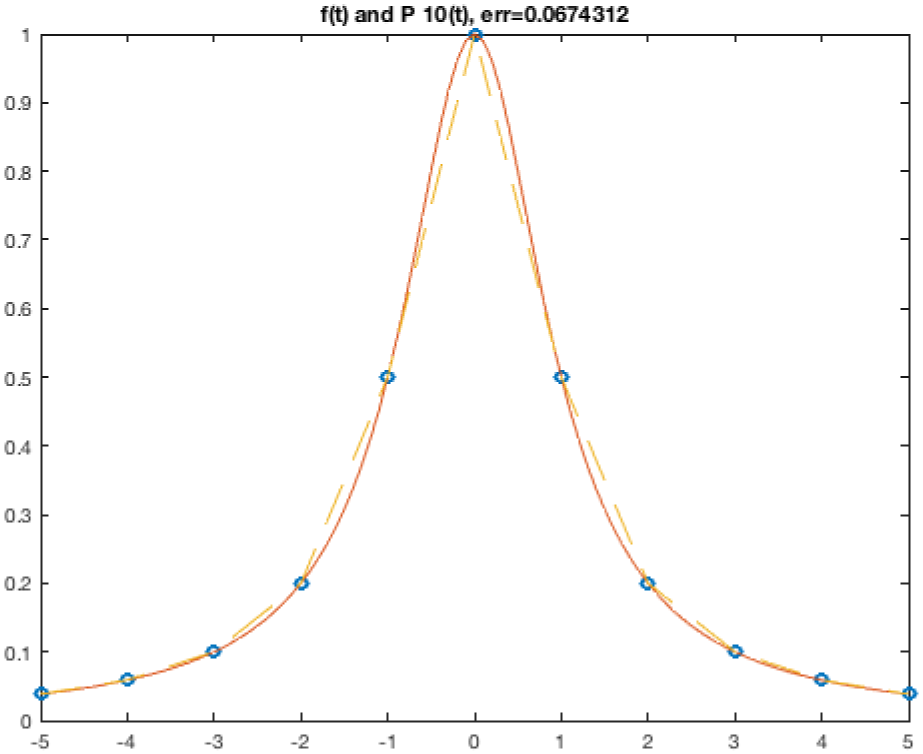
```
err =

    0.1089
```



Problem 4

```
N = 10;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
t = [-5:.1:5];
vlin = interp1(x,y,t,'linear');
err = norm(f(t)-vlin,inf);
figure;
plot(x,y,'o',t,f(t),'-',t,vlin,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))

% When comparing problem 2, 3, and 4, problem2 has an error equals to
% 1.91564, problem3 has an error equals to 0.108929, and problem4 has
 an
% error equals to 0.0674312, problem4 has less error compared to
 problem 2
% and 3.

% The dots of intersection for problem 2 are far from each other
 compared
% to problem 4's dots of intersection are closer to each other, but
 problem
```

```
% 3's dots of intersection and problem 4's dots of intersection are
 similar
% in positioning.
```

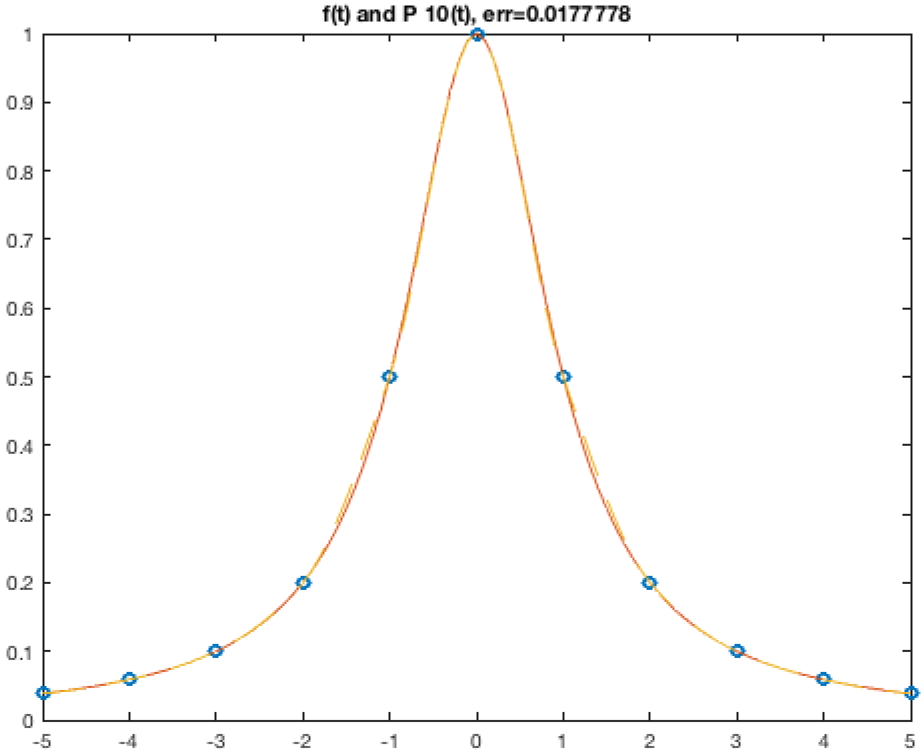


Problem 5

```
N = 10;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
t = [-5:.1:5];
vcub = interp1(x,y,t,'cubic');
err = norm(f(t)-vcub,inf);
figure;
plot(x,y,'o',t,f(t),'-',t,vcub,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))

% When comparing problem 2, 3, 4, and 5, problem2 has an error equals
 to
% 1.91564, problem3 has an error equals to 0.108929, problem4 has an
% error equals to 0.0674312, and problem5 has an error equals to
 0.0177778,
% problem5 has less error compared to problem 2, 3, and 4.

% The dots of intersection for problem 2 are far from each other
 compared
```

```
% to problem 5's dots of intersection are closer to each other, but
 problem
% 5's dots of intersection and problem 3's and 4's dots of
 intersection are
% similar in positioning.
```

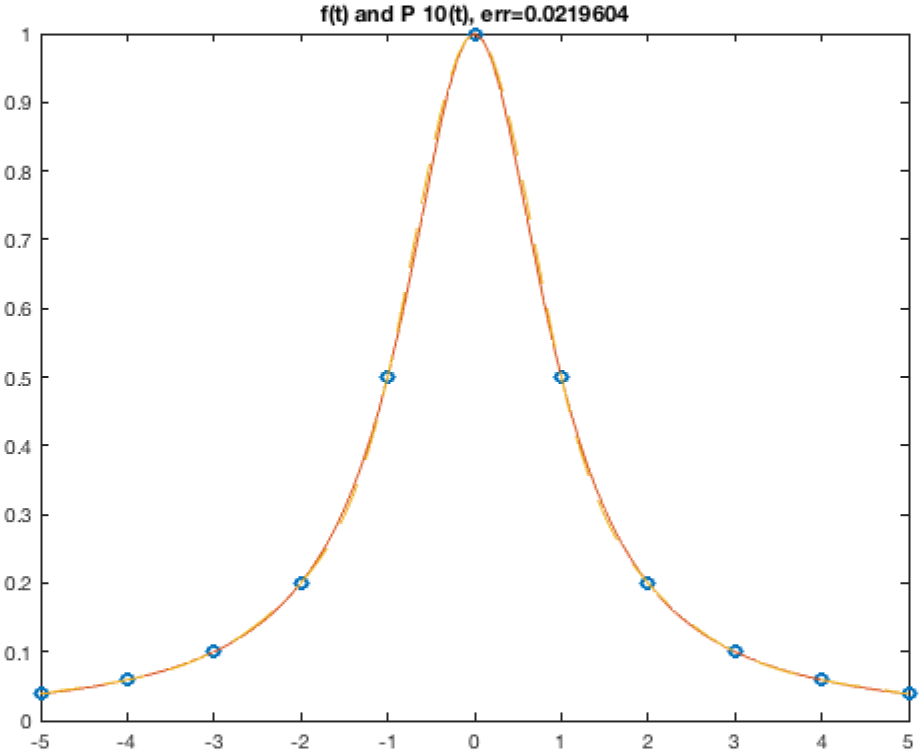
```
Warning: INTERP1(...,'CUBIC') will change in a future release. Use
INTERP1(...,'PCHIP') instead.
```



Problem 6

```
N = 10;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
t = [-5:.1:5];
vspl = interp1(x,y,t,'spline');
err = norm(f(t)-vspl,inf)
figure;
plot(x,y,'o',t,f(t),'-',t,vspl,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))

% When comparing problem 2, 3, 4, 5, and 6, problem2 has an error
 equals to
% 1.91564, problem3 has an error equals to 0.108929, problem4 has an
% error equals to 0.0674312, problem5 has an error equals to
 0.0177778, and
% problem6 has an error equals to 0.0219604, problem 5 has less error
% compared to problem 2, 3, 4, and 6.

% The dots of intersection for problem 2 are far from each other
 compared
% to problem 6's dots of intersection are closer to each other, but
 problem
% 6's dots of intersection and problem 3's, 4's, and 5's dots of
% intersection are similar in positioning.
```

```
err =

    0.0220
```



Problem 7(1)

```
N = 20;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
plot(x,y,'o')
title('N+1 = 11 equally-spaced data points')
```

```
t = [-5:.1:5];
figure;
plot(t,f(t),'-')
title('Runge function')
```

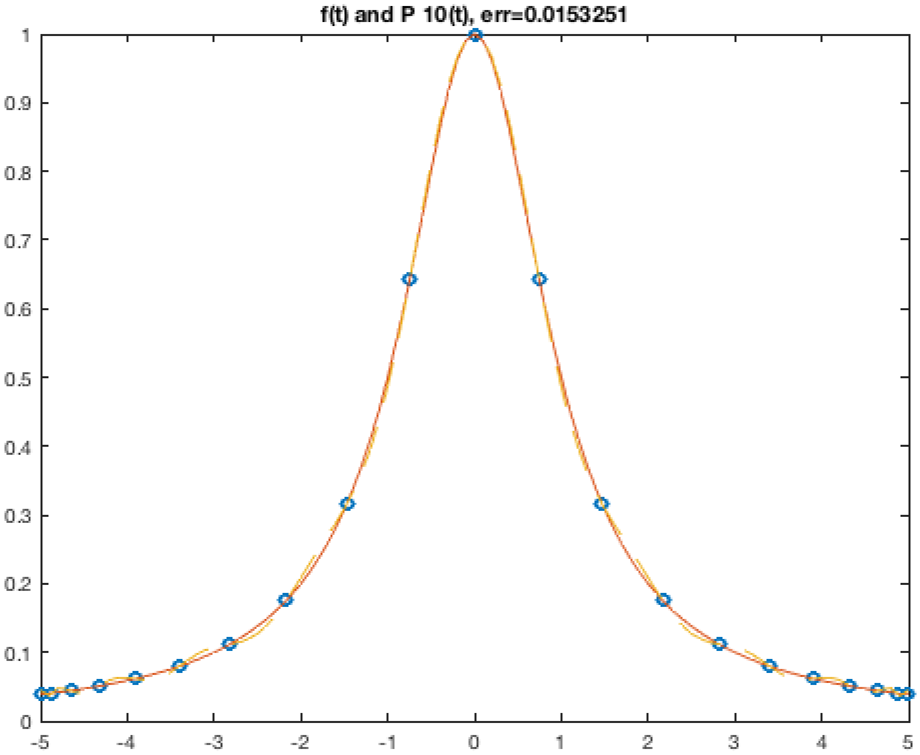
Problem 7(2)

```
N = 20;
x = linspace(-5,5,N+1); %to see values, omit the ;
f = inline('1./(1+x.*x)','x');
y = f(x);
t = [-5:.1:5];
PN = polyfit(x,y,N);
v = polyval(PN,t);
err = norm(f(t)-v,inf)
figure;
plot(x,y,'o',t,f(t),'-',t,v,'--')
title(sprintf('f(t) and P {20}(t), err=%g',err))
```

Problem 7(3)

```
N = 20;
K = N+1;
a=-5;
b=5;
xcheb=zeros(1,K);
for i=1:K
    xcheb(i)=(a+b)/2 + (b-a)/2 * cos( (i-.5)*pi/K );
end
ycheb = f(xcheb);
t = [-5:.1:5];
PNcheb = polyfit(xcheb,ycheb,N);
v = polyval(PNcheb,t);
err = norm(f(t)-v,inf)
figure;
plot(xcheb,ycheb,'o',t,f(t),'-',t,v,'--')
title(sprintf('f(t) and P {10}(t), err=%g',err))

% answer for problem 7:
% It seems that when repeating problem 2 after making N equals to 20
 that
% the error increased dramatically, 58.2781, from problem 2 which the
% error equals to 1.91564, so this caused the intersection dots to be
% apart from each other compared to when repeating problem 3 after
 making N
% equals to 20 that the error decreased dramatically, 0.0153251, from
% problem 3 which the error equals to 0.108929

% To sum up, Interpolation at Chebychev nodes behaves better when the
% number of interserction dots are large compared to N-th Degree
% Interpolating Polynomial that behaves bad when the number of
% intersections dots are large.
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X
 values,
reduce the degree of the polynomial, or try centering and scaling as
 described
in HELP POLYFIT.

err =

    0.0153
```



Published with MATLAB® R2017b