CMPS 251 - Assignment 5

Kinan Dak Al Bab

October 10, 2014

1 Interpolation Error of Polynomial Fit

1.1 Question

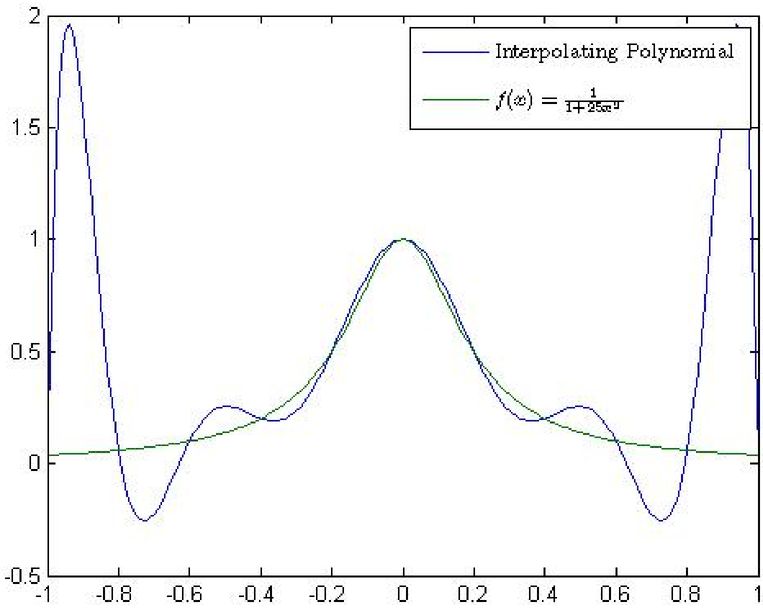
- Using 11 equi-distributed points (10 equal segments) in the interval $[-1, 1]$, find and plot the interpolating polynomial $p(x)$ for the function $f(x) = \frac{1}{1+25x^2}$.
- Observe the large discrepancies between $p(x)$ and the function $f(x)$ that the data came from. (This function is known as the Runge function after Carl Runge who first came up with this example).
- Write down an expression for the error in the interpolating polynomial above? Which part of the expression is responsible for the large errors observed?

1.2 Code

```
frequency = 11; % number of points to sample for interpolation
interval = [-1 1];

distance = interval(2) - interval(1);
step = distance / (frequency - 1); % equal distance taken

X = zeros(frequency, frequency); % Matrix containing 1, x^1, ... x^2, x^3 ... for each point
y = zeros(frequency, 1); % vector containing y0, y1, y2, y3 ...

j = 1;
for i = interval(1):step:interval(2)
    for k = 1:frequency
        X(j, k) = i^(k - 1);
    end

    y(j) = 1 / (1 + 25*i^2);

    j = j + 1;
end

c = X \ y; % solve system of linear equation to interpolate

x = interval(1):0.01:interval(2);

```

```
px = [];
fx = [];
for i = x % plot with a sample taken each 0.01
    xc = zeros(frequency, 1);

    for j = 1:frequency
        xc(j) = i^(j - 1);
    end
        px = [px dot(xc, c)];
    fx = [fx 1 / (1 + 25*i^2)];
end

plot(x, px, x, fx)
l = legend('Interpolating Polynomial', '$f(x) = \frac{1}{1 + ...
    25x^2}$');
set(l, 'Interpreter', 'latex'); % interpret legends as latex
```

1.3 Output



1.4 Comment

The discrepancies have been observed.

1.5 Errors

We know that the error is denoted given this expression

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) : \forall x \ \exists \xi \in [a, b]$$

But in our example the points are equally distanced, let us define h to be this equal distance.

$$\prod_{i=0}^{n}(x-x_i) \leq |\prod_{i=0}^{n}(x-x_i)| \leq \prod_{i=0}^{n}|(x-x_i)| \leq \frac{n!}{4}h^{n+1}$$

Therefore

$$|f(x)-p_n(x)| \leq \frac{f^{(n+1)}(\xi)}{(n+1)!}\frac{n!}{4}h^{n+1}$$

$$|f(x)-p_n(x)| \leq \frac{f^{(n+1)}(\xi)}{4(n+1)}h^{n+1}$$

We are interested in the error upper bound:

$$|f(x)-p_n(x)| \leq \frac{max_{\xi\in[a,b]}(f^{(n+1)}(\xi))}{4(n+1)}h^{n+1}$$

In the expression above we can see that the denominator has no $n!$ term, and the numerator has the term $f^(n+1)$, the dervative usually become bigger in value as we take higher order derivatives, therefore this quantity might produce a very large number causing the large discrepancies.

2 Chebyshev Points

2.1 Question

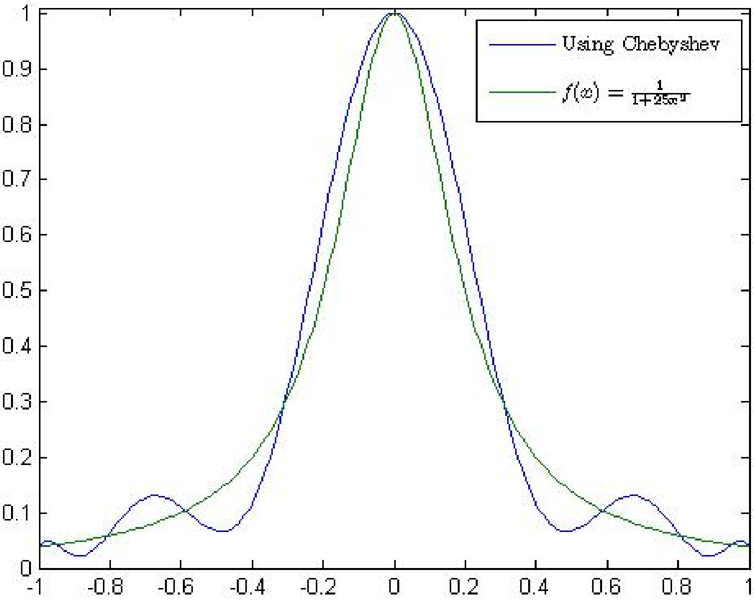
A better distribution of points has a larger density towards the ends of the interval as this exercise shows.

- Perform the preceding experiment using 11 points that are not equally spaced, $xi = cos(\pi : -\frac{\pi}{10} : 0)$. Observe the better quality of the interpolant.
- Why does this set of points produce a better interpolant?

2.2 Code

```matlab
%% The only change is at line 8

frequency = 11; % number of points to sample for interpolation
interval = [-1 1];

X = zeros(frequency, frequency); % Matrix containing 1, x^1, ... x^2, x^3 ... for each point
y = zeros(frequency, 1); % vector containing y0, y1, y2, y3 ...

j = 1;
for i = cos(pi : -pi/10 : 0) % Chebyshev distribution
    for k = 1:frequency
        X(j, k) = i^(k - 1);
    end

    y(j) = 1 / (1 + 25*i^2);

    j = j + 1;
end

c = X \ y; % solve system of linear equation to interpolate

x = interval(1):0.01:interval(2);

px = [];
fx = [];
for i = x % plot with a sample taken each 0.01
    xc = zeros(frequency, 1);

    for j = 1:frequency
        xc(j) = i^(j - 1);
    end

    px = [px dot(xc, c)];
    fx = [fx 1 / (1 + 25*i^2)];
end

plot(x, px, x, fx)
ylim([0 1.01]);
l = legend('Using Chebyshev', '$f(x) = \frac{1}{1 + 25x^2}$');
set(l, 'Interpreter', 'latex'); % interpret legends as latex
```

2.3 Output



2.4 Comment

Using Chebyshev Points we were able to get better approximation of the function (less discrepancies).

Look at the error expression previously stated :

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) : \forall x \; \exists \xi \in [a, b]$$

Since the distance are no longer equal, we can't change the product to get rid of the $n!$ term in the denominator, also, by looking at the actual error for the chebyshev interpolation we can see an exponantial term in the denominator:

$$|f(x) - p_n(x)| \leq \frac{max_{\xi \in [a,b]}(f^{(n+1)}(\xi))}{2^n (n+1)!}$$

The n+1 derivative term is still in the numerator, however the denominator is much much bigger, there for the upper bound of the error is way smaller than in the previous question, which explains the better approximation achieved (less discrepancies).

My intution tells me that the origin of the exponantial in the denominator comes from the way the x's are distributed. They are in a sense picked at an equal distance on the arch of a circle rather than equal distance on the x-axis line. so when we project these points onto the x-axis, we get n points that are more clustered towards both ends, maybe the relative distance between each two points is double (or close to the double) of the distance between the two previous points for the first half, and half that distance for the second half (that's my intution from looking at the points on the arch and their projections on x). hence, this doubling/halving behavoir might be the cause of having 2 raised by a power of n (number of points) in the denominator. (This might be completely wrong)