HW 2

AMATH 585 Tianbo Zhang

1. Muller's method for finding a root of a function $f(x)$ fits a quadratic through three given points, $(x_k, f(x_k))$, $(x_{k-1}, f(x_{k-1}))$, and $(x_{k-2}, f(x_{k-2}))$, and takes the root of this quadratic that is closest to x_k as the next approximation x_{k+1}. Write down a formula for this quadratic. Suppose $f(x) = x^3 - 2$, $x_0 = 0$, $x_1 = 1$, and $x_2 = 2$. Find x_3.

First note quadratic polynomial can be written as:

$g(x) = ax^2 + bx + c$

Then plug in $(x_k, f(x_k)), (x_{k-1}, f(x_{k-1})), (x_{k-2}, f(x_{k-2}))$ we have:

① $a x_k^2 + b x_k + c = f(x_k)$

② $a x_{k-1}^2 + b x_{k-1} + c = f(x_{k-1})$

③ $a x_{k-2}^2 + b x_{k-2} + c = f(x_{k-2})$

$x_k + x_{k+1}$

$(x_k - x_{k-1})(x_k + x_{k-2})$

①-② $\rightarrow$ ④ $(x_k^2 - x_{k-1}^2) a + (x_k - x_{k-1}) b = f(x_k) - f(x_{k-1})$

①-③ $\rightarrow$ ⑤ $(x_k^2 - x_{k-2}^2) a + (x_k - x_{k-2}) b = f(x_k) - f(x_{k-2})$

④ - ⑤ $\cdot \frac{x_k - x_{k-1}}{x_k - x_{k-2}}$: $(x_k - x_{k-1})(x_k + x_{k-1} - x_k - x_{k-2}) a$

$$= f(x_k) - f(x_{k-1}) - \frac{x_k - x_{k-1}}{x_k - x_{k-2}} \left[f(x_k) - f(x_{k-2}) \right]$$

$$\Rightarrow \quad a = \frac{f(x_k) - f(x_{k-1})}{(x_k - x_{k-1})(x_{k-1} - x_{k-2})} - \frac{f(x_k) - f(x_{k-2})}{(x_{k-1} - x_{k-2})(x_k - x_{k-2})}$$

$$\Rightarrow \quad b = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} - (x_k + x_{k-1}) \cdot a$$

$$\Rightarrow \quad c = f(x_k) - a x_k^2 - b x_k$$

For $g(x) = ax^2 + bx + c$

Then we have:

$x_0 = 0 \rightarrow f(x_0) = -2$ $\quad$ $x_1 = 1 \rightarrow f(x_1) = -1$ $\quad$ $x_2 = 2 \rightarrow f(x_2) = 6$

Then plug in $g(x) = ax^2 + bx + c$ we have:

① $c = -2$

② $a + b - 2 = -1 \rightarrow a + b = 1$

③ $4a + 2b - 2 = 6 \rightarrow 4a + 2b = 8 \rightarrow 2a + b = 4$

③-② : $a = 3 \rightarrow$ ② : $b = -2$

Then we have the quadratic to be:

$\boxed{g(x) = 3x^2 - 2x - 2}$

Then solve $f(x)$ for roots we have:

$$x = \frac{2 \pm \sqrt{4 + 24}}{6} = \frac{1 \pm \sqrt{7}}{3}$$

Then note $x = \frac{1 + \sqrt{7}}{3}$ is closer to $x_2 = 2$

Thus we have $x_3 = \frac{1 + \sqrt{7}}{3}$

2. The Chebyshev interpolation points are defined for the interval $[-1,1]$ as $x_j = \cos\left(\frac{j\pi}{n}\right)$, $j = 0, 1, \ldots, n$. Suppose we wish to approximate a function on the interval $[a,b]$. Write down the linear transformation ℓ that maps the interval $[-1,1]$ to $[a,b]$, with $\ell(-1) = a$ and $\ell(1) = b$. What interpolation points should we use on the interval $[a,b]$ to correspond to the Chebyshev points on $[-1,1]$?

First note we have:

$\ell(-1) = a$, $\ell(1) = b$

Then note the general form for linear transformation ℓ is:

$\ell = mx + c$

Then plug in $\ell(-1)$ and $\ell(1)$ we have:

① $-m + c = a$

② $m + c = b$

② - ① : $2m = b - a \rightarrow m = \frac{b-a}{2}$

$\Rightarrow c = \frac{b+a}{2}$

Thus we have:

$$\boxed{\ell(x) = \frac{b-a}{2} x + \frac{b+a}{2}}$$

Then to find the interpolation point we can plug in x_j to $\ell(x)$:

$$\boxed{\frac{b-a}{2} \cos\left(\frac{j\pi}{n}\right) + \frac{b+a}{2} \quad \text{for } j = 0, 1, \ldots, n}$$

3. Coding problem: Consider the Runge function defined on $[-5,5]$:

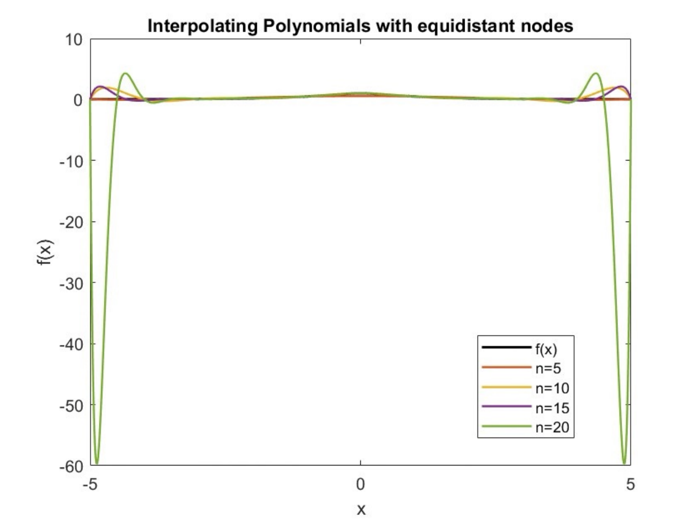
$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5,5].$$

(a) Plot the interpolating polynomials using equidistant nodes $x_i = -5 + \frac{10i}{n}$, $i = 0, \ldots, n$. Try $n = 5,\ 10,\ 15,$ and 20. Plot each of the interpolating polynomials, along with the function f on the same graph.

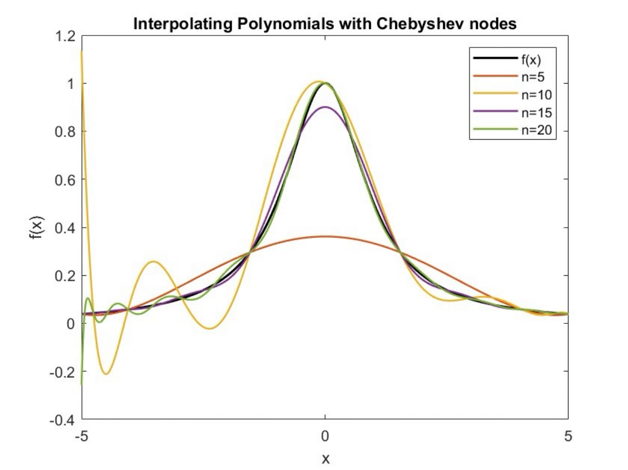
(b) Plot the interpolating polynomials using Chebyshev nodes $x_i = 5\cos\left(\frac{i\pi}{n}\right)$, $i = 0, \ldots, n$. Again try $n = 5,\ 10,\ 15,$ and 20 and plot each of the interpolating polynomials, along with the function f on the same graph. On a separate graph, plot the difference $f(x) - p_{20}(x)$ between f and the degree 20 interpolating polynomial.

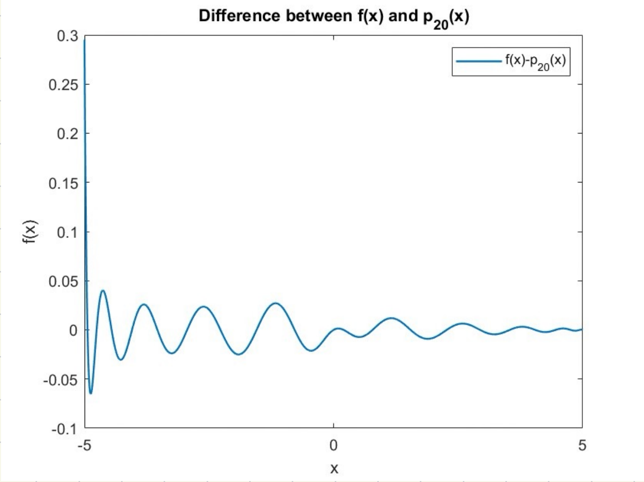
Turn in the three plots produced in (a) and (b); you do not have to turn in your code (although you may if you wish). If you want to use `chebfun` for this work, you can download it from `www.chebfun.org`. Remember, however, that you must define the domain of your function to be $[-5,5]$: `f = chebfun(' 1 / (1 + x^2) ',[-5,5])`. You may also do this problem without chebfun by simply using the Lagrange interpolation formula (see (2.49) and (2.52), pp. 74-75, in the text).

a)



b)





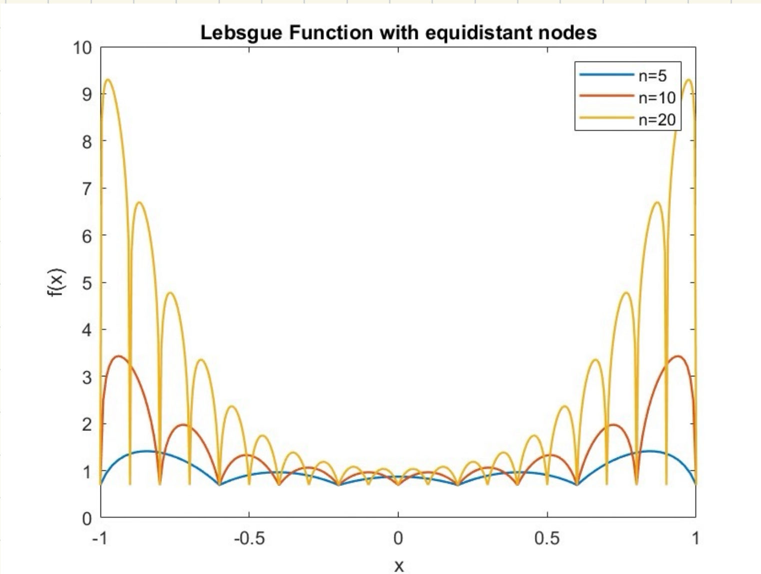
27. Use Matlab to prepare plots of the Lebesgue function for interpolation, $\lambda_n(x)$, $-1 \le x \le 1$, for $n = 5, 10, 20$, with the interpolation nodes x_i being given by

(a) $x_i = -1 + \frac{2i}{n}$, $i = 0, 1, 2, \ldots, n$;

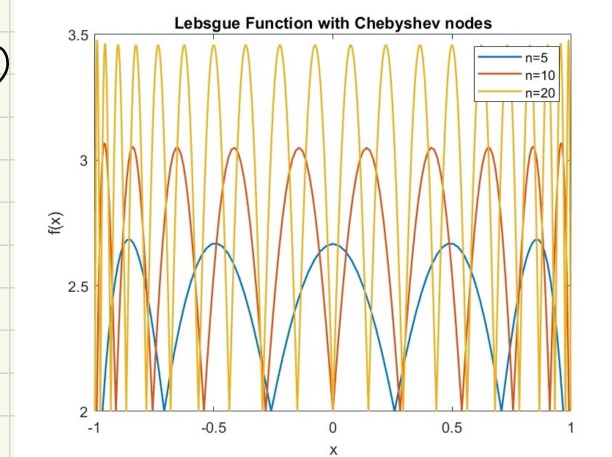
(b) $x_i = \cos \frac{2i+1}{2n+2}\pi$, $i = 0, 1, 2, \ldots, n$.

Compute $\lambda_n(x)$ on a grid obtained by dividing each interval $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$, into 20 equal subintervals. Plot $\log_{10} \lambda_n(x)$ in case (a), and $\lambda_n(x)$ in case (b). Comment on the results.

a)



b)




```
(a)The Lebesgue constant for n = 5 is 4.104960e+00
(b)The Lebesgue constant for n = 5 is 2.683563e+00
(a)The Lebesgue constant for n = 10 is 3.089070e+01
(b)The Lebesgue constant for n = 10 is 3.068587e+00
(a)The Lebesgue constant for n = 20 is 1.098753e+04
(b)The Lebesgue constant for n = 20 is 3.479078e+00
```

I noticed that when using equidistant nodes, the graph of $\log_{10}(\lambda_n(x))$ have peaks towards the end. And as n become bigger the peak also become bigger. Thus I conclude that the Lebsgue function tends to exhibit larger values at the endpoints of the interval.

As for Chebyshev nodes, I noticed that the function is generally smaller and has uniform pattern. And as n becomes bigger, the function clusters more.

38. Show that the power x^n on the interval $-1 \le x \le 1$ can be uniformly approximated by a linear combination of powers $1, x, x^2, \ldots, x^{n-1}$ with error $\le 2^{-(n-1)}$. In this sense, the powers of x become "less and less linearly independent" on [–1,1] with growing exponent n.

First note from lecture we know Chebyshe polynomials:

$T_0(x)=1$, $T_1(x)=x$

$T_{k+1}(x) = 2x T_k(x) - T_{k-1}(x)$

Let $T_{n-1}(x)$ be such polynomial.

Then we have $T_{n-1}(x)$ is a linear combination of $1, x, x^2, \ldots, x^{n-1}$

Then take:

$$\mathring{T}_n(x) = x^n - T_{n-1}(x)$$

Where $\mathring{T}_n(x)$ be the monic Chebyshev polynomial with degree n.

Then note we have:

$\| \mathring{T}_n \|_\infty \le 2^{-(n-1)}$

Thus we have:

$\| x^n - T_{n-1}(x) \|_\infty \le 2^{-(n-1)}$

Which implies $T_{n-1}(x)$ approximate x^n with error $\le 2^{-(n-1)}$ $\square$