Submit on Crowdmark by Tuesday, July 21, 2020, 11pm

Upload one .pdf file with 2 pages: Page 1 is your typed report (your discussions, data and figures on a single page); Page 2 is a listing of your code(s). The assignment is due at 11:00pm. You will receive a Crowdmark link for the upload.

I set the deadline for this assignment to be after our midterm. I recommend, however, that you complete the assignment prior to Midterm 2.

We investigate interpolation of the function

$$f : [-1, 1] \mapsto \mathbb{R}, \qquad f(x) = \frac{1}{1 + 30x^2},$$

very similar to a famous example, the "Runge" function, used to exhibit – yes you guessed it – the "Runge phenomenon".

```
f = @(x) 1./(1 + 30*x.^2 );
```

M1 Polynomial interpolation with equidistant points, $x_k = -1+2k/N$, $k = 0, 1, 2, \ldots N$. Weights for the barycentric formula are $w_k = (-1)^k \binom{N}{k}$.

M2 Polynomial interpolation with Chebyshev points of the second kind, $x_k = \cos(k\pi/N)$, $k = 0, 1, 2, \ldots N$.
Weights for the barycentric formula are $w_k = (-1)^k$, except $w_0 = \frac{1}{2}$, $w_N = (-1)^N \frac{1}{2}$.

M3 Mix and match: Use barycentric formula with equidistant points, but with Chebyshev weights. In this case the interpolating function is no longer a polynomial, but a rational function.

Your coding tasks.

a. You will need to write code to compute the weights for the barycentric formula. Matlab has a built-in command `nchoosek`; You may use that, or compute binomial coefficients recursively, $\binom{N}{0} = 1, \qquad \binom{N}{k+1} = \binom{N}{k}\frac{N-k}{k+1}$.

b. You will have to write code implementing the barycentric formula,

$$p(x) = \frac{\sum_{m=0}^{N} \frac{w_m f_m}{x - x_m}}{\sum_{m=0}^{N} \frac{w_m}{x - x_m}}.$$

This can be done to varying degree of efficiency and flexibility. Remember, that Matlab likes vectors; `for` loops not so much. Ideally, your function can take an array of evaluation points as inputs, and return the values at all of those points simultaneously.
One detail you will have to pay attention to is the case $x - x_k = 0$, i.e., an evaluation point coinciding with an interpolation point. In that case your function should simply assign the given value f_k to $p(x_k)$. If your barycentric code allows vectors as input arguments, then the Matlab function `find` is useful for this task. Otherwise, a simple `if` statement will do.

For your report compare the interpolated values to the exact values of the function f at **321** evenly space points on the interval $[-1,1]$. So finding the maximum error on the interval, translates into finding the largest error at the points `t(j)` below:

```
t = linspace(-1,1,321);   fex = f(t);
```

For all methods plot the maximum errors against N.

- M1
  - p1 Behaviour on the whole interval $[-1,1]$. Interpolate with polynomials of degree $N = 8, 12, 16, 20$; compute the maximum error on the interval [-1,1]. Does the error decrease or increase with N? What is the rate of decay/growth?
  - p2 Behaviour on the smaller interval $[-\frac{1}{2}, \frac{1}{2}]$: Here we use exactly the same computation as in part [p1], i.e., we are still interpolating on the interval $[-1,1]$. We are, however, only looking at the errors on the smaller interval $[-\frac{1}{2}, \frac{1}{2}]$, i.e., only at `t(j)`, with $81 \leq j \leq 241$. Confirm numerically that there is convergence on this interval, by interpolating with some higher-degree polynomials, say $N = 40, 80$.
- M2 Interpolate with polynomials of degree $N = 8, 16, 20, 80, 200, 250$; compute the maximum error on the interval [-1,1]. Does the error decrease or increase with N? What is the observed rate of decay/growth? What is the lowest error achieved? Do you gain accuracy by increasing N from 200 to 250?
- M3 Compute again with $N = 8, 16, 20, 80, 200, 250$; compute the maximum error on the interval [-1,1]. Does the error decrease or increase with N? What is the observed rate of decay/growth. What is the lowest error achieved?

Beyond this laundry list of things to do, I am not giving you any additional instructions as how to present your report, just suggestions. You probably want to consider:

1. Including at least one plot that shows f and some of its interpolating function, but certainly not more than four. One might be fine.
2. What kind of plot should you use to graph the maximum errors vs N: `plot; semilogy;` or `loglog`? Generally, when graphing quantities of very different magnitudes, a logarithmic plot is better; when graphing the function f and its approximations using `plot` might be preferable. Some or perhaps all your error plots could be combined into one plot.
3. The key part of your coding in this assignment is the barycentric formula. Make sure to feature it prominently, and include comments in your code.