MACM 316 - Computing Assignment 5

- Read the *Guidelines for Assignments* first.
- Submit a one-page PDF report to Canvas and upload your Matlab scripts (as m-files). Do not use any other file formats.
- Keep in mind that Canvas discussions are open forums.
- You must acknowledge any collaborations/assistance from colleagues, TAs, instructors etc.

Runge's phenomenon and aliasing

Two major concerns in polynomial interpolation are Runge's phenomenon and aliasing. Runge's phenomenon describes oscillations of the interpolant near the boundaries. Aliasing refers to lower degree polynomials pretending to be higher degree polynomials. In this report you will investigate the effects of both issues.

First, define a set of equispaced points on the interval $[-1, 1]$:

$$x_i = -1 + \frac{2i}{n}, \quad i = 0, ..., n.$$

Next, define the set of Chebyshev points on the same interval:

$$x_i = \cos(i\pi/n), \quad i = 0, ..., n.$$

These two sets will act as your interpolation points. In addition, you'll need a set of points to test your interpolants; this set should also span the interval $[-1, 1]$ but have several more points than your interpolation points.

Download the file `baryinterp.m`, which uses barycentric Lagrange interpolation to construct the Lagrange interpolating polynomial. All inputs are column vectors. Let `x` be the interpolation points, `f` the values of the function to interpolate and `grid` the points to evaluate the interpolant at. The syntax to use `baryinterp()` is then:

```
u = baryinterp(x,f,grid);
```

Measure the error of interpolation for the function

$$f_1(x) = \frac{1}{2+x}, \quad \texttt{f1 = @(x) 1./(2 + x);}$$

for n between 1 and 100. How does Runge's phenomenon affect the accuracy? Which set of interpolation points deals with it better?

Next, measure the error of interpolation for the function

$$f_2(x) = \cos(10\pi x), \quad \texttt{f2 = @(x) cos(10*pi*x);}$$

for n between 1 and 100. How many points do you need to overcome aliasing? Were there any differences between the types of interpolation points?

Tips

- The following lines of code will be useful when defining the interpolation points:

```
x_equi = 0:n; x_equi = -1 + 2*x_equi’/n;
x_cheb = 0:n; x_cheb = cos(pi*x_cheb’/n);
```

- You may use any set of points as the grid on which to test your interpolants. Use the same grid for all tests and make sure the number of points on the grid is at least 10 times as many interpolation points you use. The easiest choice is 1000 equispaced points.

- You are free to use any measure of error. The most common way to measure error is to use the `norm()` function in Matlab:

```
error = norm(f1(grid) - u, p);
```

  Here, the `p` refers to which p–norm the function will use. You can use any whole number (not zero) and you can also use `inf` for the max norm. The most popular choices of `p` are 1, 2 and `inf`.

- If you run into any kind of error message when using the function `baryinterp()` make sure that your inputs are all column vectors. You can do this by checking `size(x)` or by finding their size in the workspace section of the console.