MATH-131 (Numerical Methods for Scientists and Engineers) — Worksheet 7

Semester: Spring 2019, Instructor: Nicholas Knight

Due Mar. 14 at 0859. *Please remember to cite your sources, including collaborators.*

Deliverable: Submit a Live script titled `worksheet7.mlx` via CatCourses (under Assignments). Divide this file into sections, one for each of the following questions, plus an extra (final) section containing all the function definitions. Document each function definition to explain the input and output arguments. Also document key portions of the algorithm to make it clear you understand how your code works.

1. Repeat Worksheet 6 Q1 for the barycentric formula,

$$y = \sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i \Bigg/ \sum_{i=0}^{n} \frac{w_i}{x - x_i},$$

where we have defined the "barycentric weights"

$$w_i = 1 \Bigg/ \prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j), \qquad i = 0, 1, \ldots, n.$$

While mathematically equivalent to the Lagrange interpolation formula, the barycentric formula has the practical advantage that the bulk of the computation — computing the barycentric weights — can be performed "offline" and is efficiently updatable to include more interpolation points.

(a) Implement this as a Matlab function with the signature

```
function y = linterp_bary(X, Y, x)
```

(`X` and `Y` are arrays containing $x_0, \ldots, x_n$ and $y_0, \ldots, y_n$, resp.) You may not use any of Matlab's polynomial routines (like `polyfit` or `interp1`) nor the Symbolic Math Toolbox.

Test your code by

1. picking your favorite polynomial p of degree five,
2. populating `X` with six distinct numbers,
3. setting `Y` to be p evaluated at the corresponding entries of `X`, using `polyval`,
4. picking `x` to be a number besides those in `X`,
5. comparing p(`x`) with `linterp_bary(X, Y, x)`.

For example, implement $x \mapsto x^5 - 1$ by `p = @(x) polyval([1 0 0 0 0 -1], x);`

(b) Extend your code to allow `x` to be an array, so that `y` is an array of the same dimensions, each entry being the Lagrange interpolation formula evaluated at the corresponding entry of `x`. Reusing from Part (a) the polynomial function `p` and the arrays `X` and `Y`, run the following commands

```
x = linspace(min(X)-1, max(X)+1, 100);
plot(x, p(x), 'ko:', x, linterp_bary(X, Y, x), 'rx:');
```

The red $\times$s should coincide with the black $\circ$s.

2. (*Challenge; ungraded*)

   Write a Matlab function that return the coefficients of the Lagrange interpolating polynomial. Your function signature should be

   ```
   function p = linterp_poly(X, Y)
   ```

   Your code cannot not use the Symbolic Math Toolbox. You may find the `poly` and `polyval` routines useful.