Lab Session 4

MA581 : Numerical Computations Lab S. Bora August 24, 2022

Switch to `format long e` for all experiments

Note: To be consistent with MATLAB, consider polynomial interpolation in which indexing of data (nodes and values) starts with 1 rather than 0. Thus we denote the data points as $(x_1, f_1), \ldots, (x_n, f_n)$. The interpolating polynomial $p_n(x)$ will be of degree $n-1$.

1. Write a MATLAB function `v = polyinterp(x, f, u)` that constructs the Lagrange interpolant $p_n(x) = \sum_{j=1}^{n} \ell_j(x) f_j$ of the data $(x_1, f_1), \ldots, (x_n, f_n)$ and evaluates $p_n(x)$ at the points given in the vector u, that is, $v_i = p_n(u_i)$. Store x, f, u and v as column vectors. The vectors u and v are required for plotting $p_n(x)$.

2. Write a MATLAB function `v = barycent(x,f,u)` that generates the Lagrange interpolant $p_n(x)$ in barycentric form

$$p_n(x) = \frac{\sum_{j=1}^{n} \frac{w_j f_j}{x - x_j}}{\sum_{j=1}^{n} \frac{w_j}{x - x_j}}, \quad \text{where} \quad w_j = 1/\prod_{i \neq j}(x_j - x_i),$$

   and computes $v_i = p_n(u_i)$. Note that due to the barycentric form, $p_n(x)$ is not defined at the nodes x_j. Your program must make separate provisions for these exceptional values of u_i so that $p_n(u_i) = f_j$, if $u_i = x_j$ for some i and j.

3. Write a MATLAB function `c = newtdd(x, f)` that generates the vector of coefficients `c` of the Newton interpolating polynomial that passes through $(x_1, f_1), \ldots, (x_n, f_n)$. Your program should form the divided difference table and extract c from it.

4. Write a MATLAB function `y = nest(c,x,u)` that takes the coefficient vector `c` from `c = newtdd(x, f)` and performs nested additions and multiplications (as in `Horner.m`) to evaluate the Newton interpolating polynomial $p_n(x)$ (passing through $(x_1, f_1), \ldots, (x_n, f_n)$) at entries of the vector u such that $v = p_n(u)$.

5. Consider the following population data for the United States:

Year (x)	Population (f)
1900	76, 212, 168
1910	92, 228, 496
1920	106, 021, 537
1930	123, 202, 624
1940	132, 164, 569
1950	151, 325, 798
1960	179, 323, 175
1970	203, 302, 031
1980	226, 542, 199

   There is a unique polynomial of degree eight that interpolates these nine data points, but of course that polynomial can be represented in many different ways. Consider the following possible basis functions $\phi_j(x), j = 1:9$:

(i) $\phi_j(x) := x^{j-1}$

(ii) $\phi_j(x) := ((x - 1935)/35)^{j-1}$

(a) For both sets of basis functions, generate the corresponding Vandermonde matrix A and compute its condition number ($\mathrm{cond}(A) := \|A\| \, \|A^{-1}\|$) using the MATLAB command `cond(A)`. How do the condition numbers compare?

(b) Find the coefficients of the two interpolating polynomials by solving the associated 9×9 system of equations `Aa = f`. (Use `A\f`). The vector `a` contains the coefficients of the interpolating polynomial $p(x)$.

(c) Use appropriate modifications of `Horner.m` to evaluate both the polynomials at 100 equidistant points between 1900 and 1980. Use `linspace` for generating the vector of these points!

(d) Plot both polynomials as well as the original data points on the same graph. To ensure smooth plots, use the data from part (c).

(e) Use `barycent.m` to evaluate the Lagrange interpolating polynomial in barycentric form for the same nine data points at 1000 equidistant points in [1900,1980] and plot the polynomial as well the original data points in the same graph.

(f) Repeat part (e) for the Newton interpolating polynomial on the same data. Use `newtdd.m` and `nest.m` for this.

(g) Use the Vandermonde (from both bases), Lagrange and Newton polynomials to extrapolate the population to 1990. How close are the computed values to the true value of 248,709,873 according to the 1990 census?

6. Use the `linspace` command to generate a column vector z of 1000 equally spaced points in the interval $[0, 2\pi]$. Use `barycent` to evaluate the Lagrange interpolating polynomial of $\sin x$ on the interval $[0, 2\pi]$ with 10 equally spaced nodes at the entries of z. Ensure that the output is a column vector and find the column vector of actual errors at each point. Also generate the column vector of upper bounds on the error at each point. Use `semilogy` to plot the log values of the actual errors and their upper bounds in a single plot vis-a-vis z on the x-axis.

   Repeat the process with 12 equally spaced points.

7. Write a MATLAB function program `x = chebnodes(a,b,n)` to generate n Chebyshev nodes

$$\frac{b+a}{2} + \frac{b-a}{2} \cos\left(\frac{(2j-1)\pi}{2n}\right), \; j = 1, \dots, n$$

   in the interval $[a, b]$.

8. Consider the Runge function $g(x) := 1/(1 + 25x^2)$ for $x \in [-1, 1]$.

(a) For $n = 5, 10, 15, \dots, 45$, choose n equally spaced nodes x in the interval $[-1, 1]$ and find the corresponding interpolating polynomials $p_n(x)$ of g using `barycent` function.

(b) Use `chebnodes.m` to generate $n = 5, 10, 15, \dots, 45$ Chebyshev nodes in $[-1, 1]$ and find the corresponding interpolating polynomials $q_n(x)$ of g using `barycent` function.

(c) Plot $\|g-p_n\|_\infty$ and $\|g-q_n\|_\infty$ vs. n on "`semilogy`" axes for $n=5,10,15,\ldots,45$ on the same plot. You should estimate $\|g-p_n\|_\infty$ (respectively, $\|g-q_n\|_\infty$) by taking of $\texttt{max(abs(g(x)} - \texttt{p}_\texttt{n}\texttt{(x)))}$ (respectively, $\texttt{max(abs(g(x)} - \texttt{q}_\texttt{n}\texttt{(x))))}$ at 1000 equidistant points in $[-1,1]$.

(d) Separately plot (in single figure) $g(y), p_{15}(y)$ and $p_{25}(y)$ where y is a vector of 1000 equally spaced points in $[-1,1]$. Use commands

```
>> axis([-1 1 -0.2 1.2])
>> plot(y,g(y))
>> axis manual
>> hold on
```

before giving the commands to plot $p_{15}(y)$ and $p_{25}(y)$.

(e) Repeat part (d) for $g(y), q_{15}(y)$ and $q_{25}(y)$. The additional commands to set the axis limits using the command `axis` can be skipped in this case.

(f) Choose the vector y as in part (d) and plot the errors $E_n(y) := |g(y) - p_n(y)|$ as functions of y for $n = 15, 25, 35$ in `semilogy` scale on the same plot.

(g) Plot $\hat{E}(y) := |g(y) - q_n(y)|$ for $n = 15, 25, 35$. `semilogy` scale not required for this!

(h) Make a summary of your observations from the plots in (c)-(g). What can you conclude about the interpolation error for the Runge function when using equally spaced nodes and Chebyshev nodes in the interval $[-1,1]$?

Submit a folder containing all figures and a livescript program that contains all comments, answers and codes necessary to produce the required output. Ensure that the answers are correctly numbered. The filename of the livescript program should be Your-rollnumber-MA581-Lab4.mlx and the folder should be named as Your-rollnumber-MA581-Lab4

*** End ***