Homework 5

Due: Wednesday 11:59pm, Nov 27, 2024, on Gradescope

Notes: Please read the syllabus for information on programming expectations, the late homework policy, and the collaboration policy. Please submit both the written and coding portions of your homework on Gradescope. The written portion should be submitted as a PDF (prepared however you like, but LaTeX is excellent), with plots and tables in the appropriate places. Results and plots requested in the problems should be placed in your write-up—you should not simply refer to your code. Please mark the location of each problem in your Gradescope submission—this makes it much easier for us to grade.

Plotting and formatting: Put thought into presenting your data and results in the clearest way possible. Think about what information you want to convey, and how to convey it most effectively. Sometimes, using a table can be useful, but at other times data is best presented using a plot. All tables and plots should be accompanied by some discussion of what we are intended to learn from them.

Choose proper ranges and scales (e.g., semilogx, semilogy, loglog), always label axes, and give meaningful titles. Make sure your axis numbering and labels are large enough to be readable—getting this right will require some exploration in MATLAB, or whatever language you are using. If you do print numbers, in MATLAB for example, use `fprintf` to format the output nicely. Use `format compact` and other format commands to control how MATLAB prints things. When you create figures using MATLAB (or Python or Julia), please try to export them in a vector graphics format (.eps, .pdf, .dxf) rather than raster graphics or bitmaps (.jpg, .png, .gif, .tif). Vector graphics-based plots avoid pixelation and thus look much cleaner.

Code: Try to write clean, concise, and easy-to-read code, with meaningful variable names. Your code must be well-commented. Every line of code should be explained by comments inside the code, unless it is absolutely self-explanatory. Commenting your code is like "showing your work", and grading will take this into account in a similar manner.

You should submit three code files for this assignment: `evalcheb.m`, `chebdiff.m`, and `chebint.m`.

---

1. This problem will explore the convergence of Chebyshev interpolants for a variety of functions. You can use the function in `baryinterp.m` to help, and the script `polyinterp_cheb.m` as a useful example.

   (a) **[5 pts]** Consider the function $f(x) = 1/(1+\alpha^2 x^2)$ for $x \in [-1, 1]$. For $\alpha = 2, 4, 8$, and $n = 25$, plot f, as well as its Chebyshev interpolant of degree $n-1$, on a dense grid, using barycentric interpolation (you can make one plot with all of these curves; make sure to distinguish the curves clearly and use a legend). Then plot the maximum absolute error on $[-1, 1]$ versus n, for increasing values of n and all three values of α (on the same plot) using a `semilogy` plot. Choose appropriate values of n to obtain an error plot demonstrating clear convergence behavior. Do you observe algebraic, exponential, or super-exponential convergence? How does the convergence rate vary with α? Can you explain this behavior qualitatively based on the shape of f for the different values of α?

(b) [**5 pts**] Repeat this, but for $f(x) = |x|$ on $[-1, 1]$ (there is no more α, so you only need to do this for the one function; again make a plot of the function and its interpolant for $n = 25$, as well as an error convergence plot). For the error, use a `loglog` plot, and take $n = 2^j$ for $j = 2, \ldots, 11$. Make sure to use a dense enough grid on which to measure your error! State the rate of algebraic convergence $\mathcal{O}(n^{-k})$ you observe, and indicate it on your plot by plotting the function Cn^{-k} (you can choose C as you like). Why do you think the convergence is so slow?

(c) [**5 pts**] Repeat this, but for the function

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ x^2 & \text{if } x \geq 0 \end{cases}$$

on $[-1, 1]$. For the error, again use a `loglog` plot. This function is continuously differentiable and its second derivative has bounded variation, so we expect $\mathcal{O}(n^{-2})$ convergence of the Chebyshev interpolant. Verify this by including Cn^{-k} as a dashed line on your plot.

(d) [**5 pts**] Repeat this, but for the function $f(x) = \exp(-\alpha^2 x^2)$ on $[-1, 1]$, for $\alpha = 2, 4, 8$. For the error, use a `semilogy` plot. Do you observe algebraic, exponential, or super-exponential convergence? How does the convergence rate vary with α? Can you explain this behavior qualitatively based on the shape of f for the different values of α?

(e) [**5 pts**] Plot the errors for $f(x) = \sin(\alpha x)$ on $[-1, 1]$ for $\alpha = 10, 50, 100$ on a `semilogy` plot for values of n ranging from 5 to 150. Describe the shape of these error curves. Is there a change in behavior near $n \approx \alpha$? Why do you think this is?

2. (a) [**5 pts**] Consider the $n \times n$ Vandermonde matrix $V_{ij} = x_{i-1}^{j-1}$ for $x_0, \ldots, x_{n-1}$ equispaced nodes on $[-1, 1]$. This is the "coefficients to values" matrix for monomial expansions at equispaced nodes. Plot the condition number of this matrix (use MATLAB's `cond` function) for values of n ranging from 2 to 40 (using appropriate axes). Describe the rate of growth of the condition number with n.

(b) [**5 pts**] Consider the analogous $n \times n$ "coefficients to values" matrix for Chebyshev expansions at Chebyshev nodes: $T_{ij} = T_{j-1}(x_{i-1})$ for $x_0, \ldots, x_{n-1}$ the Chebyshev nodes on $[-1, 1]$. Compute the condition number for $n = 10$, $n = 500$, and $n = 1000$. What do you observe?

3. (a) [**10 pts**] Write a function `evalcheb` that takes as input the vector of coefficients of a Chebyshev expansion

$$p(x) = \sum_{j=0}^{n-1} \alpha_j T_j(x),$$

and a point $x \in [-1, 1]$, and evaluates $p(x)$ using the three-term recurrence for Chebyshev polynomials. Submit this function in a file `evalcheb.m`.

(b) **[5 pts]** Consider the function $f(x) = \exp(-x^2)$ on $[-1, 1]$. For increasing values of n, obtain the Chebyshev expansion coefficients $\alpha \in \mathbb{R}^n$ of f by solving the linear system $T\alpha = b$, where $b_i = f(x_{i-1})$, with $x_0, \ldots, x_{n-1}$ the Chebyshev nodes and T the Chebyshev "coefficients to values" matrix defined in Problem 2 (although more efficient methods are possible, you can simply use the MATLAB backslash operator to solve this linear system). Then evaluate the Chebyshev expansion on a dense grid of equispaced nodes using your `evalcheb` function, and plot the maximum absolute error, versus n (using appropriate axes). On the same figure, also plot the error of the Chebyshev interpolant of f evaluated using barycentric interpolation, as in Problem 1. How do the errors compare?

4. This problem will explore Chebyshev spectral differentiation and spectral indefinite integration.

(a) **[10 pts]** Write a function `chebdiff` that takes an positive integer n and returns the $n \times n$ "values to values" Chebyshev spectral differentiation matrix D, which takes the values of a function $f(x)$ at Chebyhev nodes to an approximation of its derivative at those nodes. You can construct this matrix using the simple matrix composition method described in the class notes. Hint: to obtain the matrix AB^{-1} in MATLAB, you can use the "forward slash" command `A/B`. Submit this in a file `chebdiff.m`.

(b) **[5 pts]** Consider the function $f(x) = \exp(-10x^2)$ on $[-1, 1]$, and set $n = 40$. Compute an approximation of $f'(x)$ at Chebyshev nodes using spectral differentiation. Evaluate the resulting Chebyshev interpolant on a dense grid of equispaced nodes (using your preferred method) and plot the result, as well as $f'(x) = -20x \exp(-10x^2)$ itself, on the same plot. In a second panel of the figure, plot the absolute error as a function of x on a log scale (Use MATLAB's `subplot` function to create multiple panels in a figure).

(c) **[10 pts]** Prove the following formula relating Chebyshev polynomials to their derivatives:

$$2T_k(x) = \frac{1}{k+1} T'_{k+1}(x) - \frac{1}{k-1} T'_{k-1}(x),$$

for $k > 1$. Hint: Consider $f_k(\theta) = T_k(\cos(\theta)) = \cos(k\theta)$. First, differentiate and use the sine sum and difference formulas to show that

$$-\frac{1}{\sin(\theta)} \left(\frac{1}{k+1} f'_{k+1}(\theta) - \frac{1}{k-1} f'_{k-1}(\theta) \right) = 2f_k(\theta)$$

Then show that $-\frac{1}{\sin(\theta)} f'_k(\theta) = T'_k(\cos(\theta))$. Use this and the substitution $x = \cos(\theta)$ to obtain the desired result.

(d) **[5 pts]** Using this formula, show that when $k > 1$,

$$\int_{-1}^{x} T_k(t) dt = \frac{1}{2} \left(\frac{1}{k+1} T_{k+1}(x) - \frac{1}{k-1} T_{k-1}(x) \right) + \frac{(-1)^k}{1-k^2}.$$

Also show that

$$\int_{-1}^{x} T_0(t)dt = x + 1 = T_0(x) + T_1(x),$$

and

$$\int_{-1}^{x} T_1(t)dt = x^2/2 - \frac{1}{2} = -\frac{1}{4}T_0(x) + \frac{1}{4}T_2(x).$$

Use these to obtain an expression for the "coefficients to values" Chebyshev spectral indefinite integration matrix, given by

$$S^{cv}_{j+1,k+1} = \int_{-1}^{x_j} T_k(t)\,dt$$

for $j, k = 0, \ldots, n-1$.

(e) **[5 pts]** Describe a method to compute the "values to values" Chebyshev spectral indefinite integration matrix S that takes the values of a function $f(x)$ at Chebyshev nodes to an approximation of $\int_{-1}^{x} f(t)\,dt$ at those nodes.

(f) **[10 pts]** Write a function `chebint` that takes a positive integer n and returns this $n \times n$ matrix, and submit it in a file `chebint.m`.

(g) **[5 pts]** Repeat part (b), but for an approximation of $\int_{-1}^{x} f(t)\,dt$ for $f(x) = \exp(-10x^2)$. Note that

$$\int_{-1}^{x} \exp(-10t^2)dt = \frac{\sqrt{\pi}}{2\sqrt{10}}(\mathrm{erf}(\sqrt{10}x) - \mathrm{erf}(-\sqrt{10})),$$

where $\mathrm{erf}(x)$ is called the error function, and can be evaluated using MATLAB's `erf`.