Department of Mathematics, University of Utah

Analysis of Numerical Methods I
MATH 6610 – Section 001 – Fall 2019
Homework 4
Approximation techniques

Due Thursday, December 5, 2019 by 11:59pm MT

Submission instructions:
Create a private repository on `github.com` named `math6610-homework-4`. Add your LaTeX source files and your Matlab/Python code and push to Github. To submit: grant me (username `akilnarayan`) write access to your repository.
You may grant me write access before you complete the assignment. I will not look at your submission until the due date+time specified above. If you choose this route, I will only grade the assignment associated with the last commit before the due date.
All commits timestamped after the due date+time will be ignored.

P1. This problem concerns univariate polynomial interpolation.

(a) Let $f(x) = x^3 - 1$. Without a computer, compute the degree-3 polynomial that interpolates $f(x)$ at $x = -1, 0, 1, 2$.

(b) Let $g(x) = x^4 - 1$. Without a computer, compute the degree-3 polynomial that interpolates $g(x)$ at $x = -1, 0, 1, 2$.

(c) Let $h(x) = 1/(1 + 5x^2)$. Let $h_N(x)$ denote the degree-$(N-1)$ polynomial interpolant of $h(x)$ at N equispaced points on the interval $[-1, 1]$. Write a program that plots h and the interpolant h_N for $N = 5, 20, 50$.

(d) Write a program that plots the Lebesgue function for equispaced points on this interval for $N = 5, 20, 50$. Use this to explain your findings in the previous part.

(e) Let $j_N(x)$ denote the degree-$(N-1)$ polynomial interpolant of $h(x)$ at N Chebyshev points on $[-1, 1]$. Write a program that plots h and the interpolant h_N for $N = 5, 20, 50$.

(f) Write a program that plots the Lebesgue function for Chebyshev points on this interval for $N = 5, 20, 50$. Use this to explain your findings in the previous part.

P2. Let $w(x)$ be a strictly positive, bounded weight function on an interval I on the real line. (I may be unbounded if w decays at infinity sufficiently quickly.) Given $x_1, \ldots, x_N \in I$, let I_N be the associated degree-$(N-1)$ polynomial interpolation operator, i.e., if f is continuous, then $I_N f$ is degree-$(N-1)$ polynomial that interpolates f at the x_j. Define

$$C_w(I) = \left\{ f : I \to \mathbb{R} \;\middle|\; \|f\|_{w,\infty} < \infty \right\}, \qquad \|f\|_{w,\infty} := \sup_{x \in I} w(x)|f(x)|.$$

Prove the following weighted version of Lebesgue's Lemma,

$$\|f - I_N f\|_{w,\infty} \leq [1 + \Lambda_w] \inf_{p \in P_{N-1}} \|f - p\|_{w,\infty},$$

where P_{N-1} is the space of polynomials of degree at most $N-1$, and

$$\Lambda_w = \sup_{x \in I} w(x) \sum_{j=1}^{N} \frac{|\ell_j(x)|}{w(x_j)},$$

where $\ell_j \in P_{N-1}$ is the cardinal Lagrange interpolant, $\ell_j(x_i) = \delta_{i,j}$.

P3. Define the standard L^2 Sobolev spaces of periodic functions on $[0, 2\pi]$: Given a non-negative integer s,

$$H_p^s([0,2\pi]) = \left\{ f : [0,2\pi] \to \mathbb{C} \;\middle|\; f^{(r)}(0) = f^{(r)}(2\pi) \text{ for } r = 0, \ldots s-1, \text{ and } \|f\|_{H_p^s} < \infty \right\},$$

where $f^{(r)}$ denotes the rth derivative of f (with $f^{(0)} \equiv f$), and

$$\|f\|_{H_p^s}^2 = \sum_{j=0}^{s} \left\| f^{(j)} \right\|_{L^2}^2 = \sum_{j=0}^{s} \int_0^{2\pi} \left| f^{(j)}(x) \right|^2 \mathrm{d}x.$$

Let f_N denote the frequency-$(N-1)$ Fourier Series approximation to f on $[0, 2\pi]$, i.e.,

$$f_N(x) = \sum_{|j|<N} \widehat{f}_j(x) \frac{1}{\sqrt{2\pi}} e^{ijx}.$$

Prove that,

$$\|f - f_N\|_{H_p^j} \leq N^{j-s} \|f\|_{H_p^s}, \qquad 0 \leq j \leq s$$

P4. This problem concerns interpolative quadrature formulas. All these problems should be done *without* a computer.

(a) Compute weights for the closed 4-point Newton-Cotes rule on $[-1, 1]$.

(b) Consider weights w_j and w_j' for a quadrature rule of the form

$$\int_0^1 f(x)\,\mathrm{d}x \approx w_0 f(0) + w_1 f(1) + w_0' f'(0) + w_1' f'(1),$$

where f' is the derivative of f. Compute these weights for a quadrature rule that is exact for all polynomials up to degree 3.

(c) Consider a quadrature rule of the form

$$\int_{-1}^1 f(x)\,\mathrm{d}x \approx \sum_{j=1}^{3} w_j f(x_j),$$

and assume that the x_j are distinct points. Someone claims that this quadrature rule is exact for all polynomials up to degree 3. Is this possible? If so, give conditions on x_j that must be satisfied for this to hold. If it's not possible, prove that it's not possible.

P5. This problem concerns interpolative differentiation formulas. All these problems should be done *without* a computer.

(a) Given $h > 0$, compute weights for the following one-sided differentation formula:

$$f'(x) = w_0 f(x) + w_1 f(x+h) + w_2 f(x+2h) + \mathcal{O}(h^2)$$

(b) Given $h > 0$, compute the weights for the following central differentiation formula:

$$f''(x) = w_{-1} f(x-h) + w_0 f(x) + w_1 f(x+h) + \mathcal{O}(h^2)$$

P6. Recall that the error due to degree-$(N-1)$ polynomial interpolation of a smooth function f on some compact interval $[a,b] \ni x$ is bounded by

$$\|f(x) - I_N f(x)\|_\infty \leq \frac{\|\omega(x)\|_\infty}{N!} \left\| f^{(n+1)} \right\|_\infty, \tag{1}$$

where $\omega(x)$ is the *node polynomial*, defined from the N nodes $\{x_1, \ldots x_N\}$ as

$$\omega(x) = \omega(x; X) := \prod_{j=1}^{N} (x - x_j), \qquad X := \{x_1, \ldots, x_N\}.$$

The only portion of the bound (1) that depends on the nodal choice is the appearance of $\omega(x)$. Thus, one expects that a good interpolation set can be designed if we choose $\{x_1, \ldots x_N\}$ such that

$$\{x_1, \ldots x_N\} = X = \underset{Y \subset [a,b]^N}{\operatorname{argmin}} \|\omega(x; Y)\|_\infty$$

This optimization problem is usually too difficult to solve analytically or computationally (for example, the feasible set is N-dimensional, which complicates matters for large N). A relaxation of this optimization is a *greedy* approach, wherein one iteratively chooses nodes. Since we desire the smallest possible value of $\|\omega\|_\infty$, a greedy scheme would pick a new point where ω is largest:

$$x_{j+1} := \underset{x \in [a,b]}{\operatorname{argmax}} \left|\omega(x; X_j)\right|, \qquad X_{j+1} := X_j \cup \{x_{j+1}\}. \tag{2}$$

This is reasonable since while $\omega(x_{j+1}; X_j)$ may be a large value, $\omega(x_{j+1}; X_{j+1}) = 0$. Let x_1 (and hence X_1) be some given choice of initial node. The sequeunce $\{x_j\}_{j \geq 1}$ produced by (2) is called a *Leja sequence*. Let us relax (2) even further, by iteratively maximizing not over the continuum $[a,b]$, but instead over some given discrete set $Y = \{y_1, \ldots, y_M\} \subset [a,b]$:

$$x_{j+1} := \underset{x \in Y}{\operatorname{argmax}} \left|\omega(x; X_j)\right|, \qquad X_{j+1} := X_j \cup \{x_{j+1}\}. \tag{3}$$

Naturally, this means we terminate the iteration (3) after x_M is chosen. The sequence produced by (3) is a *discrete Leja sequence*. Note that for any $N \leq M$, N-point polynomial interpolation with this sequence uses the first N points $x_1, \ldots, x_N$.

(a) Write a program that computes a 50-point discrete Leja sequence on $[-1, 1]$. You should choose $M \gg 50$, say $M = 1000$, and let Y be equispaced on $[-1, 1]$. Empirically evaluate the distribution of this sequence; how does their distribution compare to say those of equidistant nodes or Chebyshev nodes? How does the N-point interpolation error behave for the function $h(x)$ in problem **P1**? What about the N-point Lebesgue function/constant?

(b) For a given Y, let $V \in \mathbb{R}^{M \times N}$ be a Vandermonde-like matrix with entries,

$$(V)_{j,k} = y_j^{k-1}, \qquad j = 1, \ldots, M, \;\; k = 1, \ldots, N.$$

Consider the row (partial) pivoted LU decomposition of V and a permutation vector p:

$$PV = LU, \qquad p := P \begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ M \end{pmatrix}$$

With the choice $x_1 = y_1$, prove that the first N pivots in p identify the discrete Leja sequence $\{x_j\}_{j=1}^{N}$ in (3). I.e., that $x_j = y_{p_j}$, where p_j are the elements of p.