Spring 2019: Numerical Analysis
Assignment 5 (due May 3, 2019)

Please see the helpful hints at the bottom of the homework. Also, try the Matlab `publish` command to print out your code along with the output. For all Matlab calculations and plots you do, please turn in your code.

1. **[Space of polynomials P_n, 1+2+2pts]** Let P_n be the space of functions defined on $[-1, 1]$ that can be described by polynomials of degree less of equal to n with coefficients in $\mathbb{R}$. P_n is a linear space in the sense of linear algebra, in particular, for $p, q \in P_n$ and $a \in \mathbb{R}$, also $p + q$ and ap are in P_n. Since the monomials $\{1, x, x^2, \ldots, x^n\}$ are a basis for P_n, the dimension of that space is $n + 1$.

   (a) Show that for pairwise distinct points $x_0, x_1, \ldots, x_n \in [-1, 1]$, the Lagrange polynomials $L_k(x)$ are in P_n, and that they are linearly independent, that is, for a linear combination of the zero polynomial with Lagrange polynomials with coefficients α_k, i.e.,

   $$\sum_{k=0}^{n} \alpha_k L_k(x) = 0 \text{ (the zero polynomial)}$$

   necessarily follows that $\alpha_0 = \alpha_1 = \ldots = \alpha_n = 0$. Note that this implies that the $(n + 1)$ Lagrange polynomials also form a basis of P_n.

   (b) Since both the monomials and the Lagrange polynomials are a basis of P_n, each $p \in P_n$ can be written as linear combination of monomials as well as Lagrange polynomials, i.e.,

   $$p(x) = \sum_{k=0}^{n} \alpha_k L_k(x) = \sum_{k=0}^{n} \beta_k x^k, \tag{1}$$

   with appropriate coefficients $\alpha_k, \beta_k \in \mathbb{R}$. As you know from basic matrix theory, there exists a basis transformation matrix that converts the coefficients $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_n)^T$ to the coefficients $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_n)^T$. Show that this basis transformation matrix is given by the so-called Vandermonde matrix $V \in \mathbb{R}^{n+1 \times n+1}$ given by

   $$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} & x_n^n \end{pmatrix},$$

   i.e., the relation between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (1) is given by $\boldsymbol{\alpha} = V\boldsymbol{\beta}$. An easy way to see this is to choose appropriate x in (1).

   (c) Note that since V transforms one basis into another basis, it must be an invertible matrix. Let us compute the condition number of V numerically.[1] Compute the 2-based condition number $\kappa_2(V)$ for $n = 5, 10, 20, 30$ with uniformly spaced nodes $x_i = -1 + (2i)/n$, $i = 0, \ldots, n$. Based on the condition numbers, can this basis transformation be performed accurately?

[1] MATLAB provides the function `vander`, which can be used to assemble V (actually, the transpose of V). Alternatively, one can use a simple loop to construct V.

2. **[Polynomial interpolation and error estimation, 2+2+2+2pts]** Let us interpolate the function $f : [0, 1] \to \mathbb{R}$ defined by $f(x) = \exp(3x)$ using the nodes $x_i = i/2$, $i = 0, 1, 2$ by a quadratic polynomial $p_2 \in \boldsymbol{P}_2$.

   (a) Use the monomial basis $1, x, x^2$ and compute (numerically) the coefficients $c_j \in \mathbb{R}$ such that $p_2(x) = \sum_{j=0}^{2} c_j x^j$. Plot p_2 and f in the same graph.

   (b) Give an alternative form for p_2 using Lagrange interpolation polynomials $L_0(x)$, $L_1(x)$ and $L_2(x)$. Plot the three Lagrange basis polynomials in the same graph.

   (c) Compare the exact interpolation error $E_f(x) := f(x) - p_2(x)$ at $x = 3/4$ with the estimate
   $$|E_f(x)| \leq \frac{M_{n+1}}{(n+1)!} |\pi_{n+1}(x)|,$$
   where $M_{n+1} = \max_{z \in [0,1]} |f^{(n+1)}(z)|$, $f^{(n+1)}$ is the $(n+1)$st derivative of f, and $\pi_{n+1}(x) = (x - x_0)(x - x_1)(x - x_2)$.

   (d) Find a (Hermite) polynomial $p_3 \in \boldsymbol{P}_3$ that interpolates f and f' in x_0, x_1. Give the polynomial p_3 in the Hermite basis, plot f and p_3 in the same graph, and plot the four Hermite basis functions in another graph.

3. **[Polynomial interpolation, 10pts]** Interpolate the function
   $$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0, \end{cases}$$
   on the domain $[-1, 1]$ using Lagrange polynomials with Chebyshev points.

   Describe qualitatively what you see for $n = 2, 4, 8, 16, 32, 64, 128, 256$ interpolation points. Provide a table of the maximum errors
   $$||p_n - f||_\infty = \max_{x \in [-1,1]} |p_n(x) - f(x)|,$$
   and the L_2-errors
   $$||p_n - f||_2 = \sqrt{\int_{-1}^{1} (p_n(x) - f(x))^2 dx}$$
   for each $n = 2, 4, 8, 16, 32, 64, 128, 256$. Do you expect convergence in the maximum norm? How about in the L_2 norm?

4. **[Hermite interpolation, 5pts]** We are given distinct interpolation points x_i, $i = 0, \ldots, n$. Show that the Hermite interpolation polynomials $H_k(x)$ and $K_k(x)$ satisfy the following:
   $$H_k(x_i) = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k, \end{cases} \quad \text{and} \quad H_k'(x_i) = 0, \quad i = 0, \ldots, n.$$

   $$K_k'(x_i) = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k, \end{cases} \quad \text{and} \quad K_k(x_i) = 0, \quad i = 0, \ldots, n.$$

5. **[Newton–Cotes quadrature, 2+4+4pts]** In this question, we want to investigate the surprising result that Newton–Cotes quadrature is exact for polynomials of degree $n+1$, if n is even.

   (a) Why is Newton–Cotes quadrature with n quadrature nodes exact for polynomials of degree n?

   (b) Let w_k be the quadrature weights. Show that $w_k = w_{n-k}$ for $k = 0, \ldots, n$. To help you with this, remember that the quadrature nodes are defined as $x_i = a + i\left(\frac{b-a}{n}\right)$. Also recall the definition of w_k:

   $$w_k = \int_a^b L_k(x)dx.$$

   As a hint, do a change of variables in the integration of the form: $x = x_k - y + x_{n-k}$.

   (c) Using part (b), show that Newton–Cotes quadrature is exact for polynomials of degree $n+1$, when n is even. More specifically, given an arbitrary monic polynomial f of degree $n+1$ (without loss of generality), show that:

   $$\int_a^b f(x)dx = \sum_{k=0}^{n} w_k f(x_k).$$

   As a hint, use the $n+1$ degree polynomial $g(x) = (x - \frac{a+b}{2})^{n+1}$. Also, note that when n is even, the point $\frac{a+b}{2}$ is a quadrature node. Observe that the polynomial $f - g$ is of degree n, implying Newton–Cotes quadrature is exact for this polynomial from part (a).

Note 1: You can approximate the maximum error by evaluating the error $p_n - f$ at a large number of uniformly distributed points, e.g. at $\sim 10n$ points, and determining the difference with maximum absolute value, i.e.

$$||p_n - f||_\infty = \max_{x\in[-1,1]} |p_n(x) - f(x)| \approx \max_{j=0,\ldots,10n} |p_n(\xi_j) - f(\xi_j)|,$$

where $\xi_j = -1 + \frac{2}{10n}j$ for $j = 0 \ldots 10n$.

Note 2: You can approximate the L_2-error by evaluating the definite integral using, e.g., a composite quadrature rule of your choice.

Note 3: You can use the MATLAB function `lagrange_interpolant.m` provided on Piazza to compute the values of the Lagrange interpolants p_n.

Note 4: Recall that the Chebyshev points on the interval $[a, b]$ are

$$x_i = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{i+\frac{1}{2}}{n+1}\right) \text{ for } i = 0, \ldots, n.$$

6. **[Bonus, composite trapezoidal rule, 5pts]** Let $Q(n)$ be the composite trapezoidal rule approximation to $\int_a^b f(x)dx$, with $[a,b]$ divided into n subintervals. Show that

$$\frac{Q(n) - Q(2n)}{Q(2n) - Q(4n)} \to 4 \text{ as } n \to \infty.$$