Mohammad Motamed
MATH 505, Fall 2022 – HW # **2**, Posted: Sep. 27, 2022

Upload your report as a PDF-file on Canvas before Sunday October 23, 11:59pm.

1. Given $n \geq 1$ a set of $m+1$ evaluation points $\{x_i\}_{i=0}^m$, with $m \geq n$, first implement the following functions:

   (a) A function `ortho_coeffs` that computes the coefficients of a set of $n+1$ monic polynomials $\varphi_0, \ldots, \varphi_n$ that are orthogonal w.r.t. the inner product

   $$\langle f, g \rangle_m := \sum_{i=0}^m f(x_i) g(x_i), \qquad \text{with} \ \ m \geq n.$$

   Such monic polynomials satisfy the recursive formula:

   $$\varphi_{-1} = 0, \quad \varphi_0 = 1$$
   $$\varphi_{k+1} = (x - \beta_k)\varphi_k - \gamma_k \varphi_{k-1}, \quad k = 0, 1, \ldots, n-1,$$

   with the coefficients

   $$\beta_k = \frac{\langle x\varphi_k, \varphi_k \rangle_m}{\|\varphi_k\|^2}, \qquad k = 0, 1, \ldots, n,$$

   and

   $$\gamma_k = \begin{cases} 0 & \text{for } k = 0 \\ \dfrac{\|\varphi_k\|^2}{\|\varphi_{k-1}\|^2}, & k = 1, \ldots, n. \end{cases}$$

   Note that the squared norm $\|.\|^2$ in the definition of the coefficients is given by the inner product defined above, that is$\|f\|^2 := \langle f, f \rangle_m$.
   This function should take n and $\{x_i\}_{i=0}^m$ as input and should return the coefficients $\{\beta_k\}_{k=0}^{n-1}$ and $\{\gamma_k\}_{k=0}^{n-1}$.

   (b) A function `evaluate_ortho_bases` that evaluates the $n+1$ monic orthogonal polynomials $\varphi_0, \ldots, \varphi_n$.
   Input: the set of evaluation points $\{x_i\}_{i=0}^m$, and the set of coefficients $\{\beta_k\}_0^{n-1}$ and $\{\gamma_k\}_{k=0}^{n-1}$. (Note that γ_0 is not needed).

   (c) A function `approx_coeffs` that computes the coefficients $\{c_k\}_{k=0}^n$ of the best polynomial approximation $p_n(x) = \sum_{k=0}^n c_k \varphi_k(x)$ of a continuous function f in the norm induced by the provided inner product. Indeed, p_n is the *least squares approximation* of f, that is the best approximation in 2-norm with the particular inner product defined above.
   Input: $\{x_i\}_{i=0}^m$, $\{f(x_i)\}_{i=0}^m$, $\{\beta_k\}_0^{n-1}$ and $\{\gamma_k\}_{k=0}^{n-1}$. Output: $\{c_k\}_{k=0}^n$.

   (d) A function `evaluate_ortho` that evaluates the approximation polynomial $p_n(x)$.
   Input: a set of evaluation points $\{x_i\}_{i=0}^m$, the sets $\{\beta_k\}_0^{n-1}$ and $\{\gamma_k\}_{k=0}^{n-1}$, and the set of approximation coefficients $\{c_k\}_{k=0}^n$.
   In order to improve efficiency you may use *Clenshaw's recursion formula*:

   $$y_{n+2} = y_{n+1} = 0,$$
   $$y_k = (x - \beta_k)\, y_{k+1} - \gamma_{k+1}\, y_{k+2} + c_k, \quad k = n, n-1, \ldots, 1, 0.$$

   Then $p(x) = y_0$. Note that β_n, γ_n and γ_{n+1} are not needed.

Then use these functions, in addition to functions `interpolate` and `evaluate` from Homework 1, and compute the interpolations and the least squares approximation of Problem 2 of Homework 1. Verify that when $m = n$ interpolation and least-square approximation are equivalent and discuss your results and findings.

2. Consider *natural* cubic splines belonging to $\mathcal{S}_3^2(\{x_i\}_{i=0}^m)$ relative to the sets of knots $\{x_i\}_{i=0}^m$ such that

$$a = x_0 < x_1 < x_2 < \cdots < x_{m-1} < x_m = b, \quad m \geq 1.$$

(a) Show that $g(x) = x^2(|x| - 3)$ is a *natural* cubic spline belonging to $\mathcal{S}_3^2(\{x_i\}_{i=0}^4)$ for $\{x_i\}_{i=0}^4 = \{-1,\ -0.7,\ 0,\ 0.4,\ 1\}$.

(b) Denote by $s(f;\cdot) \in \mathcal{S}_3^2(\{x_i\}_{i=0}^m)$ the natural cubic spline approximation to f. Write a function `nat_spline` that computes the natural cubic spline approximation $s(f;\cdot) \in \mathcal{S}_3^2(\{x_i\}_{i=0}^m)$. The function has in input the vectors $\{x_i\}_{i=0}^m$, $\{f(x_i)\}_{i=0}^m$ and a vector of evaluation points $\mathbf{x}$ and returns the natural cubic spline evaluated at $\mathbf{x}$. Compute the spline using the fact that the second derivative of the spline $\sigma(x) = s''(f;x)$ evaluated at the internal points $\{x_i\}_{i=1}^{m-1}$ are given by

$$A\,\boldsymbol{\sigma} = \mathbf{b},$$

where

$$A = \begin{bmatrix} 2(h_1+h_2) & h_2 & & & & \\ h_2 & 2(h_2+h_3) & h_3 & & & \\ & h_3 & 2(h_3+h_4) & h_4 & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{m-2} & 2(h_{m-2}+h_{m-1}) & h_{m-1} \\ & & & h_{m-1} & 2(h_{m-1}+h_m) \end{bmatrix},$$

$$\boldsymbol{\sigma} = (\sigma_1,\ \sigma_2,\ \ldots,\ \sigma_{m-1})^\top,$$

$$\mathbf{b} = (6(df_2 - df_1),\ 6(df_3 - df_2),\ \ldots,\ 6(df_m - df_{m-1}))^\top,$$

with $h_i := x_i - x_{i-1}$ and the $df_i := (f_i - f_{i-1})/h_i$, $i = 1, m$. Once the coefficients $\sigma(x_i)$, $i = 0, 1, \ldots, m$ have been computed, including $\sigma(x_0)$ and $\sigma(x_m)$, one can use a function `eval_spline` that takes as input the vectors $\{x_i\}_{i=0}^m$, $\{f(x_i)\}_{i=0}^m$, $\{\sigma(x_i)\}_{i=0}^m$ and a vector of evaluation points $\mathbf{x}$ and returns the natural cubic spline evaluated at $\mathbf{x}$. The python implementation of `eval_spline` is provided on the course webpage.

(c) Verify the correctness of the spline function, showing that $s(g,\cdot)$, for $g(x) = x^2(|x| - 3)$, using $\{x_i\}_{i=0}^4 = \{-1,\ -0.7,\ 0,\ 0.4,\ 1\}$ is equivalent to $g(x)$, for $x \in [-1,\ 1]$, up to round-off errors.

(d) Use the function `nat_spline` to find the natural cubic spline approximation to $f(x) = \dfrac{1}{1+x^2}$ on $[-5, 5]$, for (i) equally spaced nodes $\{x_i^{\text{eq}}\}_{i=0}^{10}$, and (ii) Chebyshev nodes $\{x_i^{\text{ch}}\}_{i=0}^{10}$. Compare your results with the interpolating polynomial of degree 10 over the two sets of nodes; see Homework 1, Problem 2.

Note: Matlab/Octave `spline` function, implements a not-a-knot twice-differentiable cubic spline, where $s'(x_0) = s'(x_1)$ and $s'(x_{n-1}) = s'(x_n)$. This is not quite a natural spline but it is close to. One can use this function in the case `nat_spline` has not been implemented correcly.