Last updated: Sunday 20th March, 2022 at 19:24.

Written assignment #3

Consider the monomials:

$$m_k(x) = x^k. \tag{1}$$

The monomials $m_0(x), m_1(x), \ldots, m_n(x)$ form a basis for the set of degree n real polynomials:

$$P_n = \left\{ \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_n x^n : (\alpha_0, \ldots, \alpha_n) \in \mathbb{R}^{n+1} \right\}. \tag{2}$$

Define an inner product on the set of square-integrable functions defined on $[-1, 1]$ for a pair of functions $f, g : [-1, 1] \to \mathbb{R}$ by:

$$(f, g) = \int_{-1}^{1} f(x) g(x) dx. \tag{3}$$

This inner product is analogous to the dot product that we're already familiar with for a pair of vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$:

$$\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i. \tag{4}$$

As we saw, we can use Gram-Schmidt orthogonalization to compute the QR decomposition of a matrix. This successively orthogonalizes each column of a matrix with respect to an orthonormal basis that we incrementally form.

Problem 1. Let $n = 5$ and apply the Gram-Schmidt process to $m_0, \ldots, m_5$ to get a new set of orthogonal polynomials $p_0, \ldots, p_5$, and such that:

$$(p_i, p_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Problem 2. Plot $p_0, \ldots, p_5$ on $[-1, 1]$ using matplotlib and insert the plot inline in your submission using the LaTeX `graphicx` package.

Problem 3. As we will see in the coming weeks, orthogonal polynomials are useful for approximating functions. Let $f : [-1, 1] \to \mathbb{R}$ be square-integrable, and let $p_0, \ldots, p_n$ be a set of orthogonal polynomials on $[-1, 1]$. For a function $g : [-1, 1] \to \mathbb{R}$, define $\|g\|$ by:

$$\|g\|^2 = (g, g) = \int_{-1}^{1} g(x)^2 dx. \tag{6}$$

Find the set of coefficients $c_0, \ldots, c_n$ such that:

$$\left\| f - \sum_{i=0}^{n} c_i p_i \right\|^2 \tag{7}$$

is minimized.

Problem 4. Pick three square-integrable functions defined on $[-1, 1]$ which *are not* polynomials. They do not need to be smooth or even continuous. Now, for $n = 5$, compute the coefficients $c_0, \ldots, c_5$ for each function f and form the approximation:

$$\hat{f}_k(x) = c_0 p_0(x) + \cdots + c_k p_k(x), \qquad 0 \leq k \leq 5, \tag{8}$$

and the relative error:

$$e_k = \frac{\left\| \hat{f}_k - f \right\|}{\|f\|}. \tag{9}$$

For each of your three functions, make the following plots:

- The original function f and the approximations $\hat{f}_k$ for each k, plotted on $[-1, 1]$. Put all of these plots on the same graph.
- The pointwise error $\hat{f}_k - f$ for $k = 0, \ldots, 5$ plotted on $[-1, 1]$. Put all of these plots on the same graph.
- The relative error e_k versus k, for $k = 0, \ldots, 5$. Use a `semilogy` plot for this one.

You are free to do this problem by hand (except for the graphing, which must be done using Python), or you can do it using Python. If you use Python, please investigate using `scipy.integrate.quadrature` to compute the integrals—this will make your life easier.