Homework 3

Due: 5:00 PM EDT on Mon Nov 29, 2021

Instructions: The homework consists of written as well as programming exercises. Your homework is your own – while you are free to discuss the problems with other students, each of you must submit your own writeup, code, etc. There is a zero-tolerance policy regarding cheating and submission of identical work.

Submitting your homework consists of two parts:

1. A **single PDF** of your written work as well as *the output of your code* should be uploaded to Gradescope (which can be accessed via Brightspace). When uploading to Gradescope, please make sure to mark which pages contain the solutions to which problems. This saves everyone a lot of time when grading.

2. Any code that you write for the assignment must also be submitted via GitHub Classroom. You will receive an invitation URL for the assignment via email. It is *your responsibility* to setup a Github account and join the GitHub Classroom.

   The code you submit *must be easy* to execute. For example, if it is Matlab code in the file `prob1.m`, the code should run when I execute the command `matlab prob1` from the command line. If the code is in C or Fortran, for example, you must include a script which compiles and executes the code. If you do not submit code for a particular problem, you will not receive credit even if you submitted the output of your code.

---

1. (From Gentle, problem 4.10) Approximate $f(t) = e^t$ on the interval $[-1, 1]$ as

$$f(t) \approx \sum_{k=0}^{5} c_k T_k(t) \tag{0.1}$$

   where the $T_k(t)$ are the Chebyshev polynomials.

   (a) (**4 points**) Write a program to use Algorithm 4.1 (in Gentle) to compute the approximation at a given point t.
   (b) (**2 points**) Graph the function and your approximation.
   (c) (**2 points**) Determine the error at $t = 0$.
   (d) (**2 points**) Determine the integrated squared error.

2. (**10 points**) Consider approximating the function $f(x) = \cos x$ on the interval $[0, \pi]$ in two ways:

   (a) Via polynomial interpolation at n equispaced nodes on $[0, \pi]$, i.e. the nodes

$$x_k = \pi(k-1)/(n-1),$$

   for $k = 1, \ldots, n$.

(b) Via projecting onto the set of polynomials $P_0, P_1, \ldots, P_{n-1}$ which are orthogonal on this interval (these polynomials are merely scaled/translated Legendre polynomials).

Let p_{n-1} denote the degree $n-1$ polynomial that interpolates f at these equispaced nodes x_k, and let $\hat{f}_{n-1}$ denote the approximation to f obtained from projecting onto $P_0, \ldots, P_{n-1}$. For $n = 1, 2, 4, 8, 16$, numerically calculate the L^2 error in these approximations:

$$\begin{aligned} \epsilon_I(n) &= \sqrt{\int_0^\pi |p_{n-1}(x) - f(x)|^2 \, dx}, \\ \epsilon_P(n) &= \sqrt{\int_0^\pi |\hat{f}_{n-1}(x) - f(x)|^2 \, dx}. \end{aligned} \tag{0.2}$$

I recommend using Matlab or Mathematica (and their built-in interpolation and integration routines) in order to compute the interpolants p_{n-1} and the least squares approximations $\hat{f}_{n-1}$, as well as the L^2 errors above. For each n, make a plot showing f, p_{n-1}, and $\hat{f}_{n-1}$, and report your L^2 errors clearly in table format.

3. (**10 points**) As discussed in class, the Bessel function J_0 can be evaluated using the following integral representation:

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \cos\theta) \, d\theta. \tag{0.3}$$

Use the composite trapezoidal rule to evaluate the above integral to an absolute precision of at least 10^{-12} at each of the following values: $x = 1, 2, 4, \ldots, 128, 256$ (be sure to transform the integral, as in class, in order to take advantage of the periodicity of the integrand). For each value of x determine the smallest number of quadrature nodes in the composite trapezoidal rule that can be used to obtain the required accuracy. Print your output in the tabular format show below.

x	$J_0(x)$	Minimum n
1	0.7651976865579666	?
2	?	?
4	?	?
8	?	?
16	?	?
32	?	?
64	?	?
128	?	?
256	?	?

4. For parameters $\alpha, \beta > 0$, the beta distribution has PDF for $x \in (0, 1)$ equal to

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}. \tag{0.4}$$

For the purposes of this problem, assume that α and β are arbitrary positive integers.

(a) (**5 points**) Write a subroutine to compute the CDF of this distribution using adaptive 8th-order Gauss-Legendre quadrature. The CDF should be accurate to at least an absolute precision of $\epsilon = 10^{-8}$. Recall that the CDF is

$$F(x; \alpha, \beta) = \int_0^x f(t; \alpha, \beta)\, dt. \tag{0.5}$$

What are the values of $F(0.25; 3, 3)$ and $F(0.5; 4, 5)$?

(b) (**5 points**) Using the subroutine for the CDF that you wrote in the first part of the problem, write another subroutine that can be used for drawing random samples from this distribution using the inverse CDF method. Draw $N = 10, 100, 1000, 10000, \ldots$ samples with $\alpha = 5$, $\beta = 6$ and show that the sample mean is converging to the expected value $\alpha/(\alpha + \beta)$ at the proper Monte Carlo rate (i.e. $\propto 1/\sqrt{N}$).