Homework 10
APPM 4600 Numerical Analysis, Fall 2025

Due date: Friday, November 14, before midnight, via Gradescope. **Instructor**: Prof. Becker
Revision date: 11/8/2025

Theme: Fourier/trig and integration.

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet is allowed for basic tasks (e.g., looking up definitions on wikipedia) but it is not permissible to search for proofs or to *post* requests for help on forums such as `http://math.stackexchange.com/` or to look at solution manuals. Please write down the names of the students that you worked with. Please also follow our AI policy.

An arbitrary subset of these questions will be graded.

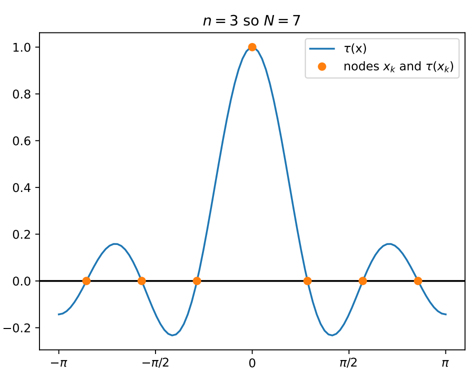
Turn in a PDF (either scanned handwritten work, or typed, or a combination of both) to **Gradescope**, using the link to Gradescope from our Canvas page. Gradescope recommends a few apps for scanning from your phone; see the Gradescope HW submission guide.

We will primarily grade your written work, and computer source code is *not* necessary (and you can use any language you want). You may include it at the end of your homework if you wish (sometimes the graders might look at it, but not always; it will be a bit easier to give partial credit if you include your code). For nicely exporting code to a PDF, see the APPM 4600 HW submission guide FAQ.

Problem 1: Trigonometric polynomials Let $\mathcal{T}_n$ be the set of trigonometric polynomials of degree n, i.e., $\mathcal{T}_n = \operatorname{span}\{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \ldots, \cos(nx), \sin(nx)\}$, which is a $N = 2n+1$ dimensional real vector space[1]. Define

$$\tau(x) = \frac{2}{N}\left(\frac{1}{2} + \cos(x) + \cos(2x) + \ldots + \cos(nx)\right)$$

and clearly $\tau \in \mathcal{T}_n$ (and hence is continuous). Define $x_k = \frac{2\pi}{N}k$ for $k = -n, -n+1, \ldots, n$. A plot of τ is below:



a) Prove that $\tau(x) = \begin{cases} 1 & x = 0 \\ \frac{\sin(\frac{N}{2}x)}{N\sin(\frac{1}{2}x)} & x \neq 0 \end{cases}$. This representation is continuous (as it should be) as you can see via L'Hôpital's rule. *Note: there are several ways to show the equality. If you want a hint for one possible method, hold this paper up to a mirror to read the following:* *Cross-multiply by the denominator, and use the trig identity* $2\sin(\alpha)\cos(\beta) = \sin(\alpha+\beta) + \sin(\alpha-\beta)$.

[1] We have seen another basis for this space, $\{e^{ikx}\}_{k=-n}^{n}$, though this is slightly more awkward in the sense that we allow complex coefficients c_k yet force them to have a conjugate symmetry, $c_{-k} = \overline{c_k}$.

b) Show that $\tau(x_k) = \begin{cases} 1 & k=0 \\ 0 & k \neq 0 \end{cases}$. *This should remind you of Lagrange polynomials. We sometimes call such functions with this property "cardinal functions", and if we take the limit (with appropriate scalings) as $N \to \infty$ then τ becomes $\sin(x)/x$ also known as sinc or "sine cardinal."*

c) Define $\tau_k(x) = \tau(x - x_k)$. Prove that the set $B = \{\tau_k\}_{k=-n}^{n}$ is a basis for $\mathcal{T}_n$.

Problem 2: Trigonometeric interpolation Using the basis $\{\tau_k\}$ defined in the previous problem makes interpolation easy. Let our nodes be the x_k points defined above (note: this is an open formula, so do *not* use `linspace` to create them), and let y_k be the corresponding y-values we wish to interpolate. Then the interpolating trigonometric polynomial is $p_n(x) = \sum_{k=-n}^{n} y_k \tau_k(x)$.

a) Write code to do this. Specifically, compute p_n on $[-\pi, \pi]$ using $N = 2n+1$ nodes for $n = 2, 4, 6, 8, 10$, plotting the interpolant(s) and the original function on the same plot (use 100 or so points in the plot), for the following three functions f (with $y_k = f(x_k)$):

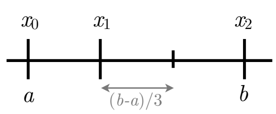
i. $f(x) = e^{\sin(x)}$

ii. $f(x) = \ln\left(2 + \cos(3x)\right)$

iii. $f(x) = \cos^9(x)$

b) For $f(x) = \cos^9(x)$, plot the error for for $n = 2, 4, 6, 8, 10$. Can you give an explanation for the behavior?

Problem 3: Quadrature on non-equispaced nodes Consider (non-composite) quadrature to estimate $\int_a^b f(x)\,dx$ using nodes $x_0 = a$, $x_1 = a + \frac{1}{3}(b-a)$, $x_2 = b$. See the figure below:



Determine a quadrature scheme using these nodes that can integrate quadratic polynomials exactly (i.e., has degree of accuracy 2). *Hint: Interpolate! Without loss of generality, you can write $h = (b-a)/3$ and then let $x_0 = 0$. And you can check your work numerically by seeing if your scheme really does integrate a quadratic exactly.*

Problem 4: Newton-Cotes for large n. For the function $f(x) = \cos(x)$ on the interval $[-.5, 1.5]$, approximate $\int_{-.5}^{1.5} f(x)\,dx$ using (non-composite) Newton-Cotes quadrature of increasing order n, for $n = 2, 3, \ldots, 30$, and record the error (since you can calculate the exact integral by hand). What can you say about how the error changes with n? Does this make sense? Show the error for each n and write a few sentences. *Note: You don't need to derive the Newton-Cotes formulas yourself. You can either look them up in a table or use software. In Python, `scipy.integrate.newton_cotes` will do it. In Matlab, you can find third-party code on the FileExchange.*