Assignment 4

1. **Gram matrix.** Suppose V is a vector space with an inner product $\langle \cdot, \cdot \rangle$ and a basis $v_1, \cdots, v_n$. The *Gram matrix* is the $n \times n$ matrix with entries

$$G_{jk} = \langle v_j, v_k \rangle \ .$$

Show that if $v = x_1 v_1 + \cdots x_n v_n$ and $w = y_1 v_1 + \cdots y_n v_n$,, then

$$\langle x, y \rangle = x^T G y \ .$$

On the right, x and y are column vectors

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad , \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \ .$$

2. **Gram matrix in the monomial basis.** Let $\mathcal{P}_p$ be the vector space of polynomials of degree at most d. This space has a *monomial basis*

$$v_0(x) = 1 \ , \quad v_1(x) = x \ , \quad \cdots \ , v_d(x) = x^d \ .$$

Consider the $L^2[-1,1]$ inner product (you don't have to know what those symbols mean)

$$\langle p, q \rangle = \int_{-1}^{1} p(x) q(x) \, dx \ .$$

   (a) Show that this is an inner product (bilinear, symmetric, positive definite). *Warning.* Don't worry if this seems easy. It is.

   (b) Find the Gram matrix for this inner product in the monomial basis.

3. **Cauchy Schwarz inequality.** For any inner product there is a corresponding *norm* defined by

$$\|x\| = \sqrt{\langle x, x \rangle} \ .$$

For example, in the *Euclidean* inner product has

$$\langle x, y \rangle = x^T y \quad , \quad \|x\| = \sqrt{\sum_{k=1}^{n} x_k^2}$$

The $L^2[-1,1]$ inner product has

$$\|p\| = \left(\int_{-1}^{1} p(x)^2 dx \right)^{\frac{1}{2}} \ .$$

If $p(x) = a_0 + a_1 x + \cdots + a_d x^d$ and the coefficient vector is

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_d \end{pmatrix}$$

then $\|p\| = \sqrt{a^T G a}$. The *Cauchy Schwarz inequality* is that, for any vectors x and y,

$$\langle x, y\rangle^2 \leq \|x\|^2 \|y\|^2 \ .$$

This inequality "makes sense" in that both sides are quadratic in x and y. One of the proofs first uses the minimization problem

$$t_* = \arg \min_t \|x - ty\|^2 \ , \quad M = \min_t \|x - ty\|^2 \ .$$

(a) Find a formula for

$$\frac{d}{dt} \|x - ty\|^2 \ .$$

Hint. If $f(t)$ is any function and $f(t + \Delta t) = f(t) + C\Delta t + O(\Delta t^2)$, then $C = \frac{d}{dt} f(t)$. Apply this to

$$f(t) = \|x - ty\|^2 = \langle x - ty, x - ty\rangle \ .$$

(b) Use the result of part (a) to find first t_* then M.

(c) Use the fact that $M \geq 0$ to prove the Cauchy Schwarz inequality.

- Both ChatGPT and Wikipedia know this proof. Try to get it on your own. It is an important method that is used a lot in linear algebra.
- Differentiating matrix and vector functions, as in part (a), is also important. The idea of part (a) will come up again when we do perturbation theory.
- There is a dot product formula in 2D or 3D: $x \cdot y = |x|\,|y| \cos(\theta)$, where θ is the angle between x and y. With abstract vector spaces and inner products, this would be $\langle x, y\rangle = \|x\|\,\|y\| \cos(\theta)$. In the abstract setting, this is the *definition* of $\cos(\theta)$. The Cauchy Schwarz inequality guarantees that $|\cos(\theta)| \leq 1$, like an ordinary cosine.

4. **Minimum norm solution, orthogonality** . Suppose L is a linear map from vector spaces V to W with kernel $K \subset V$. Suppose we have $x \in V$ with $Lx = y$. We seek $x_* \in V$ with $Lx_* = y$ that minimizes $\|x_*\|$. This is the the *minimum norm solution* to the equation $Lx = y$. Suppose that this norm comes from an inner product: $\|x\|^2 = \langle x, x\rangle$.

(a) Show that $x_* = x + z$ for some $z \in K$ (z in the null space of L).

(b) (*orthogonality principle*) Show that the minimum norm solution satisfies $\langle z, x_* \rangle = 0$ for all $z \in K$. When this happens, we say that x_* is orthogonal to the subspace K (or perpendicular to K). *Hint.* If there is z with $\langle z, x_* \rangle \neq 0$, then there is a t (probably small) so that $\|x_* + tz\| \leq \|x_*\|$. It is easier to work with the squares of these quantities, which are given in terms of the inner product without using square roots.

(c) Show that if $v_1, \cdots, v_m$ is a basis of K, then x_* is orthogonal to K if and only if x_* is orthogonal to each of the basis elements, i.e., $\langle x_*, v_j \rangle = 0$ for each basis element v_j. The number of basis elements, m, is the dimension of K.

5. **Application to polynomials**. Suppose $p \in \mathcal{P}_d$ and the linear map $\mathcal{P}_d \xrightarrow{L} \mathcal{P}_d$ is given by $p(x) \mapsto p'(x)$. Find a basis for the kernel of L. Suppose $q \in \mathcal{P}_{d-1}$ has degree $d-1$. Find p_* with is the minimum norm polynomial (in the $L^@[-1,1])$ norm) that satisfies $p'_* = q$. *Hint.* Use Exercise 4 to show that minimum norm polynomial satisfies $\int_{-1}^{1} p_*(x)dx = 0$. Repeat with $Lp = p''$ and suppose q has degree $d-2$. What is a basis for K in this case?

6. **Adaptive integrator**. Let $I(\Delta x)$ represent one of our integration rules with equal size bins (rectangle, midpoint, Simpson, etc.). Show that if the rule has order of accuracy p, then there is an error estiate formula

$$C\,[\,I(\Delta x) - I(2\Delta x)\,] = I(\Delta x) - I + O(\Delta x^{p+1}) \,. \tag{1}$$

Write a code that takes as input the integration problem (integrand, range of integration, desired accuracy) and seeks a Δx that achieves this accuracy. The code should start with some small number of bins (maybe 10 or 20?) and have a maximum number of bins it will try before reporting failure (maybe a hundred thousand or a million, which could make the code slow in Python). It has a loop using the n and $\frac{1}{2}n$ bin approximation, estimates the error using (1). If the estimated error is bigger than ϵ (the given error target), then n is replaced by $2n$ and the loop is repeated. If n is bigger than the max, the routine reports failure by raising an exception. If it does meet the error target, the final act is to return $I(\Delta x)$ for the finest grid used, with the error estimate added in. The basic $I(\Delta x)$ integration routine should be one of the ones you wrote for Assignment 3. Feel free to modify that to use Simpson's rule, which normally meets the error target with a much smaller n. See if that's true for you.

First apply your code to a problem you know the answer to so you can see whether the approximation it returns meets the error target. Then apply it to the function

$$f(t) = \int_0^1 \cos(tx^2)\,dx \,. \tag{2}$$

Do this for t in the range $[0, 10^4]$ (or whatever upper limit your code can achieve. Do the integral to "plotting accuracy", which means that the error is smaller than the thickness of the line in the plot. Make a plot of $f(t)$ in whatever range you achieve. Note that it would be hard to know what Δx to use in advance, partly because it depends on t. The integral is harder when t is larger. The convenience of an adaptive integrator should be clear.

The integral (2) is an *incomplete Fresnel integral* (pronounced "frenel" because the "s" is silent in this French name). It's "incomplete" because the integral goes to $x = 1$ instead of $x = \infty$. It arises in optics, particularly diffraction.

Feel free to use an AI to help you figure out how to write a `try/except` block in Python. Make sure to test that it works, possibly by giving an impossibly small error target to see what happens. Plotting accuracy is only 1% or maybe .1% error, which is big as this kind of error goes.