Remarks: Problems 1 through 4 are really quick. Problems 5 and 6 involve numerical computations. We are getting to a point in the course where we know enough to solve interesting problems. The result is very useful. Problem 7 has you apply the famous "normal equations" in a setting that is new for you. Problems 8 and 9 show why some norms may be "better" than others.

1. For $z \in \mathbb{C}$ or $\mathbb{C}^n$, let $\bar{z}$ denote the complex conjugate. Show that on $(\mathbb{C}^n, \mathbb{C})$, the definition $< x, y >= x^\top \bar{y}$ satisfies the definition of an inner product used in lecture (which comes from David Luenberger, Optimization by Vector Spaces) while $< x, y >= \bar{x}^\top y$ satisfies Definition 6.2.1, page 185, of Nagy. *You have now learned that in the case of inner product spaces with the field as $\mathbb{C}$, there is no consistency on putting the linearity on the right versus the left!* So, when you read a paper, you have to carefully check which definition is being used, unless the vector space is real, in which case, linearity on the left and right both hold. Mamma mia!

2. Nagy, Page 198, Prob. 6.3.4 Note that the problem does NOT require the basis to be *orthonormal*. Because it is really boring to check all of them, only verify that $< p_0, p_3 >= 0$ and $< p_1, p_2 >= 0$.

3. Using the standard inner product on $\mathbb{R}^3$, namely $< x, y >= x^\top y$, apply the Gram Schmidt procedure by hand to the set of vectors

$$y_1 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, y_2 = \begin{bmatrix} 4 \\ 0 \\ -1 \end{bmatrix}, y_3 = \begin{bmatrix} -2 \\ 2 \\ 3 \end{bmatrix}$$

   Do not bother to make them orthonormal.

4. This problem is dealing with the **Matrix Inversion Lemma**. It is a very useful result for reducing the complexity of matrix inversions. We will use it when doing recursive least squares and in the Kalman filter.

   (a) Either look up a proof of the following fact and copy it down, or develop your own proof: Suppose that A, B, C and D are compatible[1] matrices. If A, C, and $(C^{-1} + DA^{-1}B)$ are each square and invertible, then $A + BCD$ is invertible and

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

   (b) In many important applications, the inverse of A may be already known or easy to compute. Here is a made up example, but it gets the point across: By hand, evaluate $(A + BCD)^{-1}$ when

$$A = \text{diag}([1,\ 0.5,\ 0.5,\ 1,\ 0.5]),\ B = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 3 \end{bmatrix}, C = 0.2, D = B^\top$$

[1] The sizes are such the matrix products and sum in $A + BCD$ make sense.

For Problems 5 and 6, download HW4_Problem5.mat and HW4_Problem6.mat from Content > MATLAB

5. Your objective is to

compute the derivative of $y(t)$, in a causal manner, that is, your estimate of the derivative at time t can only depend on measurements $y(\tau)$ for $\tau \leq t$. The file also contains the true derivative of the signal, called dy in the mat-file. The hints outline a solution to this problem using regression and a "moving window" of data points, but you do not have to follow the hints.

   (a) A *naive* estimate of $dy(t)/dt = \frac{y(t)-y(t-\Delta t)}{\Delta T}$. Compute this estimate first and plot it versus time t, along with the true derivative. Why it is called a *naive* estimate will be clear in the next problem. :) **Note:** In a realistic experimental situation, the true derivative would not be known. In this HW problem, it is provided in the data file for the purpose of evaluating your algorithms.

   (b) Use regression to compute an estimate of $\hat{y}_k(t)$, and from this estimate, compute $\widehat{\frac{dy_k}{dt}}(t)$, which you take as an estimate of $dy(t)/dt$ (see the hints). Plot your estimate versus the true derivative. Label your plots clearly using the `legend` command. **Remark:** Because the data is "clean" in this case, (no added noise), the naive scheme works great and you'll wonder why you are bothering with regression. The next problem settles this question.

6. Use your algorithm from

Prob. 5 to compute the derivative of $y(t)$, in a causal manner. The signal $y(t)$ in the data file has been corrupted by noise. Yikes!

   (a) Compute your estimate of $dy(t)/dt$ and the *naive* estimate. Plot both of them as before and label your plots clearly using the `legend` command.

   (b) Compute and report

$$\frac{1}{L}\sqrt{\sum_{k=1}^{L}\left(\dot{y}(t_k) - \widehat{\frac{dy_k}{dt}}(t_k)\right)^2}$$

   where L is the number of data points in your estimate of the derivative. It is OK to check with other classmates to see how your estimate compares to theirs. To be clear, your solution for the estimate of $\dot{y}(t)$ cannot use the provided values of the derivative.

Remark: The deviations in the data may not always come from "measurement" noise. In many sensors, the values of the outputs are quantized, that is, the values reported by the sensor have discrete levels, such as $2^{12} = 4096$ values for 360 degrees of rotation, and then maybe only 11 of the 12 bits are really significant, giving you only 2048 values. It may seem like $\frac{360}{2048} = 0.18$ degrees is an accurate reading until you try to compute derivatives of the signal.

8. **Def.** A norm $||\cdot||$ on a vector space $(\mathcal{X}, \mathbb{R})$ is said to be *strict*[2] when

$$||x+y|| = ||x|| + ||y||$$

holds if, and only if, there exists a non-negative constant α such that either $y = \alpha x$ or $x = \alpha y$. One then says that $(\mathcal{X}, \mathbb{R}, ||\cdot||)$ is strictly normed.

Prob. Suppose that $(\mathcal{X}, \mathbb{R}, ||\cdot||)$ is strictly normed. Let M be a subspace of $\mathcal{X}$, and suppose that $x \in X$ is such that $d(x, M) > 0$. Show that if there exists $m^* \in M$ such

$$||x - m^*|| = d(x, M) := \inf_{y \in M} ||x - y||,$$

then m^* is unique.

9. (Work any 2 of the 3 parts!) For simplicity, we'll do this problem on $\mathbb{R}^2$, but the results hold for $\mathbb{R}^n$. For each of the following norms, determine if it is strict in the sense of Prob. 8. We let $x = [x_1, x_2]^\top$

   (a) $||x||_1 = |x_1| + |x_2|$

   (b) $||x||_2 = \sqrt{(x_1)^2 + (x_2)^2}$

   (c) $||x||_\infty = \max\{|x_1|, |x_2|\}$

   To show something is <u>not</u> strictly normed, you just need to provide a counterexample, that is x and y that are not related by a non-negative scale factor and yet $||x+y|| = ||x|| + ||y||$.

[2] We always have the triangle inequality holding, namely $||x+y|| \leq ||x|| + ||y||$. When the norm is strict, we see that unless x and y are related by a non-negative constant, then the inequality is strict, namely $||x+y|| < ||x|| + ||y||$.

Hints

Hints: Prob. 1 Nothing exciting here. Just straightforward manipulations of complex numbers.

Hints: Prob. 4 You can multiply $(A + BCD)$ by the proposed inverse and check that you obtain the identity [it is enough to multiply on the left or the right, you do not have to check both, and you will find one of the two is easier to do than the other]. Alternatively, there are dozens of proofs on the web. You can choose a proof that seems to satisfy you and report that. You will not be faulted for your choice. Most of the results on the web are not stated quite as precisely as our HW problem, but that's life!

Hints: Prob. 5 One way to approach the problem is outlined below. In order to check that whatever you do is correct, it is best to check your work on data where you know the answer. Hence, create a test data set, such as $y(t) = \sin(t)$, and make sure that your estimate looks like $\cos(t)$, or let $y(t) = t^2$, and make sure that your derivative estimate is close to $2t$.

(a) Denote the time instances by $t_k = k\Delta T$, where ΔT is the sample interval.

(b) Define a *moving window of data* Y_k, which uses measurements at $t_{k-M+1}, \cdots, t_k$. The data block has $M \geq 2$ measurements in it, and M is fixed. It is called a moving window because the "window" shifts to the right when each new data point becomes available (imagine doing this in real time on a computer).

(c) Select your favorite set of functions $\{\varphi_1(t), \cdots, \varphi_N(t)\}$ for $N \geq 1$ and regress Y_k to obtain

$$\hat{y}_k(t) = \sum_{i=1}^{N} \alpha_i[k]\varphi_i(t).$$

The notation $\alpha_i[k]$ emphasizes that the coefficients depend on the data used in the regression, and the data do change with time.

(d) Obtain your estimate of $\frac{dy}{dt}(t)$ by differentiating $\hat{y}_k(t)$.

(e) If you follow to the letter the above outline, your regression matrix will change with k, and hence it will have to be recomputed at each time step. If instead, you always regress your favorite set of functions, with time shifted by t_k, namely, you use $\{\varphi_1(t-t_k), \cdots, \varphi_N(t-t_k)\}$ for $N \geq 1$ and regress $Y_k = A\alpha[k]$ to obtain

$$\widehat{y}_k(t) = \sum_{i=1}^{N} \alpha_i[k]\varphi_i(t - t_k)$$

your regressor matrix A will be the same at each time step. What will change is the data vector Y_k, which will change the coefficients $\alpha_i[k]$ at each time step.

(f) In my implementation, I have

$$\widehat{\frac{dy_k}{dt}}(t) = \sum_{i=1}^{N} \beta_i[k]\frac{d}{dt}\varphi_i(t - t_k)$$

and there is a fixed matrix B such that

$$\begin{bmatrix} \beta_1[k] \\ \vdots \\ \beta_N[k] \end{bmatrix} = BY_k.$$

In a real-time environment, such an implementation is very practical.

(g) You can also consult an old conference paper S. Diop, J.W. Grizzle, P.E. Moraal, and A. Stefanopoulou, "Interpolation and numerical differentiation for observer design", Proceedings of the American Control Conference, 1994, pp. 1329 - 1333. It is a simple result that has been rediscovered a few times, Lisha Chen, M Laffranchi, N.G. Tsagarakis, D.G., Caldwell, "A novel curve fitting based discrete velocity estimator for high performance motion control," IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2012, pp. 1060 - 1065.

(h) **Related work:**

- AM Dabroom and HK Khalil, "Discrete-time implementation of high-gain observers for numerical differentiation," International Journal of Control, Volume 72, Issue 17, 1999, pp. 1523-1537.
- Vasiljevic, Luma K and Khalil, Hassan K, "Error bounds in differentiation of noisy signals by high-gain observers," Systems & Control Letters, Vol. 57, No. 10, 2008, pp. 856–862.

Hints: Prob. 5 and 6 If you're using python instead of MATLAB, you can load the .mat files with the package scipy.io.loadmat (`https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.io.loadmat.html`).

Hints: Prob. 7 Compute the Gram matrix, as in lecture. It reduces <u>all</u> finite-dimensional least squares problems to a set of matrix equations. It's kind of amazing. No, it's totally amazing!

Hints: Prob. 8 No more hints given in office hours! Suppose both $m_1 \in M$ and $m_2 \in M$ satisfy $||x - m_i|| = d(x, M)$. The objective is to show that $m_1 = m_2$. Let $\gamma = d(x, M)$ and note that $\frac{m_1+m_2}{2} \in M$. Hence,

$$\gamma = \inf_{y \in M} ||x - y|| \leq ||x - \frac{m_1 + m_2}{2}|| = ||\frac{x - m_1}{2} + \frac{x - m_2}{2}|| \leq \frac{1}{2}||x - m_1|| + \frac{1}{2}||x - m_2|| = \frac{\gamma}{2} + \frac{\gamma}{2} = \gamma.$$

Think about what has to hold at each of the "less than or equal to signs", given the common bound on either end. And then go back to the definition of a strict norm. If you can show that $x - m_1 = x - m_2$, then you'll be done!

Remark: By definition of the infimum, for any point $m \in M$, $\inf_{y \in M} ||x - y|| \leq ||x - m||$. Because $\frac{m_1+m_2}{2}$ is a point in M, it follows that $\inf_{y \in M} ||x - y|| \leq ||x - \frac{m_1+m_2}{2}||$. The other inequality used is the triangle inequality.

Hints: Prob. 9 (a) and (c) are straightforward. Try a few vectors and you'll see what is going on. Part (b) will take a clever idea or a lot of brute force calculation. If you are pressed for time, skip (b) and just read the solutions. *The take home message is that how you decide to measure "error" (i.e., which norm you chose) can have a big effect on the solution of your approximation problem.*