HW # 04 Solutions: FAQ

1. How to keep track of all the rules?

   Answer: Build that google doc and share it with rob101staff umich.edu

2. The determinant has a lot of rules. How to keep them straight?

   Answer: See above! I guarantee you, they will link up and make sense if you put them all in one place.

3. How to remember that $(A \cdot B)^{-1} = B^{-1}A^{-1}$?
   Answer: You can remember this formula by how to check it is correct. Assume A and B are both $n \times n$ invertible matrices. From the matrix product rule,

   $$B^{-1}A^{-1}AB = B^{-1}(A^{-1}A)B = B^{-1}I_nB = B^{-1}B = I_n.$$

   This formula will appear frequently in other types of algebra you encountered.

4. How to remember that $(A \cdot B)^\top = B^\top A^\top$?

   Answer: Method 1 (from Prof. Gilbert Strang): When you wear socks and then shoes, to reverse the process you take off your shoes first and then your socks!

   Method 2: Let's look at the dimensions. Suppose that $A = n \times k$ and $B = k \times m$. Then the product $C = A \cdot B$ makes sense and is $n \times m$.

   We know that $C^\top$ is $m \times n$, $A^\top$ is $k \times n$ and $B^\top$ is $m \times k$.

   Hence, in general, the product $A^\top \cdot B^\top$ does not make sense because we cannot multiply $k \times n$ by $m \times k$. Of course, if A and B were square, it would always work, so if you forget which way they go, try a 1×3 and 3×2 for example, and check it!

   Method 3: We use Julia notation: $A = \left[\begin{array}{cccc} a_1^{\rm col} & a_2^{\rm col} & \cdots & a_m^{\rm col} \end{array}\right]$ and $B = \left[\begin{array}{c} b_1^{\rm row};\ b_2^{\rm row};\ \cdots;b_m^{\rm row} \end{array}\right]$. Then

   $$\begin{aligned}
   A \cdot B &= \sum_{k=1}^{m} a_k^{\rm col} \cdot b_k^{\rm row} \\
   (a_k^{\rm col})^\top &= \text{row} \\
   (b_1^{\rm row})^\top &= \text{column} \\
   A^\top &= \left[\ (a_1^{\rm col})^\top;\ (a_2^{\rm col})^\top;\ \cdots;\ a_m^{\rm col}\ \right] \\
   B^\top &= \left[\begin{array}{cccc} (b_1^{\rm row})^\top & (b_2^{\rm row})^\top & \cdots & (b_b^{\rm row})^\top \end{array}\right] \\
   B^\top \cdot A^\top &= \sum_{k=1}^{m} (b_k^{\rm row})^\top \cdot (a_k^{\rm col})^\top \text{ equals sum of the columns of B transpose times the rows of A transpose!}
   \end{aligned}$$

   Hence, if you can show that

   $$\left(a_k^{\rm col} \cdot b_k^{\rm row}\right)^\top = (b_k^{\rm row})^\top \cdot (a_k^{\rm col})^\top,$$

   then you are done. But

   $$\begin{aligned}
   \left[a_k^{\rm col} \cdot b_k^{\rm row}\right]_{ij} &= a_{ik}b_{kj} \\
   \left[\left(a_k^{\rm col} \cdot b_k^{\rm row}\right)^\top\right]_{ij} &= \left[a_k^{\rm col} \cdot b_k^{\rm row}\right]_{ji} = a_{jk}b_{ki}, \\
   \left[(b_k^{\rm row})^\top \cdot (a_k^{\rm col})^\top\right]_{ij} &= b_{ki}a_{jk} = \left[\left(a_k^{\rm col} \cdot b_k^{\rm row}\right)^\top\right]_{ij}
   \end{aligned}$$

   Method 4: watch this YouTube video: https://youtu.be/HgFUYepT7FM

5. Why must b be a linear combination of the columns of A?

   Answer:

   The following is more or less a proof: Suppose that $\bar{x}$ satisfies $A\bar{x} = b$, which is the same thing as saying $\bar{x}$ is a solution of $Ax = b$. Then, doing the indicated multiplication of $A\bar{x}$ via our now favorite method, the columns of A times the rows of the vector $\bar{x}$, which are its scalar entries $\bar{x}_i$, yields

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} \bar{x}_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} \bar{x}_2 + \cdots + \begin{bmatrix} a_{1m} \\ a_{2m} \\ \vdots \\ a_{nm} \end{bmatrix} \bar{x}_m = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{1}$$

   Exchanging the two sides of the equal sign and moving the scalars $\bar{x}_i$ to the front of the vectors give

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \bar{x}_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} + \bar{x}_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} + \cdots + \bar{x}_m \begin{bmatrix} a_{1m} \\ a_{2m} \\ \vdots \\ a_{nm} \end{bmatrix}; \tag{2}$$

   in other words, b is a linear combination of the columns of A. The other way around works as well: if we manage to write

$$b = c_1 a_1^{\mathrm{col}} + c_2 a_2^{\mathrm{col}} + \cdots c_m a_m^{\mathrm{col}}$$

   for some real numbers $c_i \in \mathbb{R}$, then $\bar{x} = \begin{bmatrix} c_1 & c_2 & \cdots & c_m \end{bmatrix}^\top$ satisfies $A\bar{x} = b$, and hence it is a solution to $Ax = b$.

   Recall that $a_j^{\mathrm{col}} := \begin{bmatrix} a_{ij} & a_{2j} & \cdots & a_{nj} \end{bmatrix}^\top, 1 \le j \le m$.

6. Existence of solutions? **Answer:** See above!

7. As the math gets more conceptually deep, it is hard to see how to apply it to real problems. **Answer:** Well, look at Chapter 8 and the in-class lecture material. They show you how very large systems of linear equations come about when you do regression problems, which is what you do when you want to build a model (that is, a function) to explain your data. We are living in the era of big data! The workforce needs people who understand how to work with big data sets. One aspect of ROB 101 is Mathematics at the Scale of Life!

   Here is another way to see problems where you can tie everything together:

   (a) $Ax = b$ has a solution if, and only if, b is a linear combination of the columns of A.

   (b) b is a linear combination of the columns of A if, and only if the matrix $A_e = [A \ \ b]$ has the same number of linearly independent columns as A has linearly independent columns.

   (c) Hence, counting linearly independent vectors in a set is an important problem. Look for this section of the textbook! And then look at the example following this Pro-Tip.

Pro Tip! Linear Combination or Not?

Fact: A vector $v_0 \in \mathbb{R}^n$ can be written as a linear combination of $\{v_1, \ldots, v_m\} \subset \mathbb{R}^n$ if, and only if, the set $\{v_0, v_1, \ldots, v_m\}$ has the same number of linearly independent vectors as $\{v_1, \ldots, v_m\}$.

Applying this to determining if a linear system of equations $Ax = b$ has a solution, we first define $A_\mathrm{e} := [A \;\; b]$ by appending b to the columns of A. Then we do the corresponding LDLT Factorizations

- $P \cdot \left(A^\top \cdot A\right) \cdot P^\top = L \cdot D \cdot L^\top$
- $P_\mathrm{e} \cdot \left(A_\mathrm{e}^\top \cdot A_\mathrm{e}\right) \cdot P_\mathrm{e}^\top = L_\mathrm{e} \cdot D_\mathrm{e} \cdot L_\mathrm{e}^\top$.

Fact: $Ax = b$ has a solution if, and only if, D and D_e have the same number of non-zero entries on their diagonals.

Why? We know that $Ax = b$ has a solution if, and only if, b is a linear combination of the columns of A. That is, A and $A_e := [A \;\; b]$ must have the same number of linearly independent columns.

(Optional Remark): You get the same result by defining $A_\mathrm{e} := [b \;\; A]$; in fact, you can insert b anywhere you want, even in the middle of the columns of A.

8. Why are identity matrices important?
   Answer: For the same reason number 1 is important. We need an identity element to make sense of more advanced concepts such as inverse of an element.

9. Why is the matrix inverse important?
   Answer 1: Without matrix inverses all of our calculations are irreversible! For example we would not be able to undo the multiplication of a matrix by applying its inverse. The lack of an inverse element would make life much harder. That's why invertible matrices are considered "nice" matrices.

   Answer 2: Let's riff on the above. $\bar{x}$ is a solution to the equation $\underbrace{A}_{n\times m}\underbrace{x}_{m\times 1} = \underbrace{b}_{n\times 1}$ if, and only if, $A\bar{x} = b$. Suppose there exists a matrix $\bar{A}$ such that
   $$A \cdot \bar{A} = I_n.$$
   Then, we note that $\bar{x} := \bar{A} \cdot b$ is a solution to our equation, because
   $$A\bar{x} = \underbrace{A \cdot \bar{A}}_{I_n} \cdot b = I_n \cdot b = b.$$

   Ooohhh, in the above, we did not even assume that A was square! We just said, if we can find a matrix such that when we multiply A on the right-hand side by that matrix we get the appropriately sized identity matrix, then we can use that matrix to compute a solution to our equation! That's pretty dope!

   A matrix such that $A \cdot \bar{A} = I_n$ is called a right-inverse of A while a matrix such that $\bar{A} \cdot A = I_m$ is called a left-inverse of A. In 99% of all undergraduate linear algebra courses for first year students, including ROB 101, one assumes the matrix is square, that is $n = m$, and then the right and left inverses become the same matrix, we simply call that matrix the matrix inverse of A, and we denote it A^{-1}. The key point, however, is that if the matrix inverse does exist, we can use it to determine a solution to our equation by $\bar{x} = A^{-1}b$.

   The above is 100% true and for a **pure mathematician**, the story stops there. As an engineer or as an applied mathematician or even a physicist, you will need one day to worry about how efficiently you compute a solution to $Ax = b$. Why? Because you will be solving HUGE problems or even modest problems, but they are in a for loop and if you are me, you must solve the equations super fast so that Cassie Blue knows where to go! We update our control loop on Cassie Blue every 0.5 milliseconds! In that case, it is much faster to solve $Ax = b$ via an LU Factorization (or the QR Factorization we are learning in Chapter 9) than to compute A^{-1}.

10. Why is it hard to compute a matrix inverse?
Answer: The general matrix inversion requires an advanced method such as LU decomposition. You know by now that the human brain is not as good as a computer to perform super fast repeated computations. While we understand how it is done, we leave it for the computer to invert the matrix for us. So our job is to come up with the algorithm, then computer can do the work when needed. On a more philosophical note, the computer is a labor and does not augment human intelligence. Because we provided the intelligent algorithm and programmed it for the computer. If one day the computer can solve a similar problem to augment our intelligence, who knows how far we can go. Artificial Intelligence and Robotics are heading that way!

11. Computation of matrix inverse seems unnecessarily complex and comes with little payoff. **Answer:** Hopefully, the above helps!

12. When to use the matrix transpose?
Answer 1: The short answer is when it makes sense. Sometimes we naturally use the transpose to make sense of an equation or operation. The mathematical way of expressing it can become more advanced and is more suitable for a math-oriented linear algebra course. Later when you learn the dot product of two vectors a and b, if A is a matrix, we can write

$$\langle Aa, b\rangle = \langle a, A^\top b\rangle.$$

Answer 2: Recall our Pro-Tip: $Ax = 0 \iff A^\top Ax = 0$. You need to use the matrix transpose here!

13. Concepts of linear independence and dependence, and how to check them. **Answer:** The more you become familiar with vector spaces, the more this notion makes sense. Your approach can be purely applied and think about a set of tools to make sure the problem is well-posed. Or you can learn about more abstract concepts and the role of linear independence. It all starts by thinking about matrices as a bunch of columns vectors stacked horizontally. That's a vector space! The rank-nullity theory is a fundamental part of linear algebra and here is a good part to learn it. But we don't emphasize on the theoretical proofs in this course. Once you are able to solve problems, the theory will make sense after hearing it.

14. Permutation matrices...seem hard to grasp. **Answer:** We basically want to rearrange some objects. For example, we have $Apple, Orange, Plum$ and want to arrange it as $Orange, Plum, Apple$. For matrices, these objects are their rows and columns. a clever way is to use another matrix and matrix multiplication to permute rows and columns of a matrix. This is clever because we can reuse our tools that are also programmable in a computer.

15. Why are large and small numbers in a matrix bad?

Answer: Computers use a finite number of zeros and ones to represent a number. Individual zeros and ones are called bits. You can use your bits to give you granularity as in representing a number with a lot of decimal places, such as representing π to 64 decimal places, or you can use bits to represent the range of a number, as $\pm 10^k$. All computations involve a trade off between granularity (accuracy) and range: large range means low granularity and low accuracy. In fact, you can basically summarize it as a conservation rule

$$\text{accuracy} \times \text{range} \;=\; \text{a constant}.$$

When accuracy is low, round off errors accumulate! Real computations in a computer are hard. Engineers, applied mathematicians, and physicists live in the world of real computations. I do not know of any undergraduate courses that really dig into this. Perhaps someone has a suggestion?

16. Why is there a theory-practice gap?

Answer: See above. Doing calculations with perfect arithmetic, as you did in High School, is different than doing calculations with with imperfect arithmetic, as you must do when you are an engineer, applied mathematician, or physicist, for example.

17. Struggling with $P \cdot A = L \cdot U$

Answer: If you try to run our standard LU Factorization Algorithm on the matrix

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix},$$

you would first define

$$C = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \text{ and } R = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

But then we have $\text{pivot} = C[1] = 0$, and to normalize, we'd have to divide by zero, which is a major no-no! However, if we swap the rows of A by pre-multiplying by

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

we end up with

$$P \cdot A = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}.$$

We can then define

$$C = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \text{ and } R = \begin{bmatrix} 2 & 3 \end{bmatrix}.$$

We then have $\text{pivot} = C[1] = 2$, and to normalize, we are no longer dividing by zero!

18. How to use LU Factorization of Solving $Ax = b$ **Answer:** Now, how to use LU Factorization **with row permutations** to solve $Ax = b$? Suppose that $P \cdot A = L \cdot U$. Then, multiplying both sides of $Ax = b$ by P gives

$$P \cdot Ax = P \cdot b.$$

Now, because $P \cdot A = L \cdot U$, we end up with

$$L \cdot Ux = P \cdot b.$$

Hence, we first solve

$$Ly = P \cdot b$$

using forward substitution (note that the rows of b have been swapped) and then we solve

$$Ux = y.$$

Ta da!

19. Visualizing 3D space? **Answer:** $\mathbb{R}^2$ is straightforward. It's a big jump to $\mathbb{R}^3$. Your professor almost failed out of engineering as a freshman because, back in the day, we had to do isoperimetric drawings, where one has to be able to look at a 3D object and then draw all of its side views in 2D in such a way that you uniquely define the object, to scale. What a nightmare. Fortunately, EEs did not need to be good at that. Therefore I became an EE!

Some of us are wired to move things around in 3D and keep track of things. Others, not so much. I am terrible at t. But it does not bother me, because who can do 4D? 127D? 10^8D However, I can do algebraic manipulation of vectors with thousands and thousands of components, without being able to picture them in my head. I apply the same reasoning to $\mathbb{R}^3$. I think Prof. Ghaffari is a pro at $\mathbb{R}^3$. You should hang out with him!

20. How to understand vectors in $\mathbb{R}^n$? **Answer:** Carefully? Ok, just being not funny, I know. See above. I treat them as lists of numbers, literally, that obey certain rules of vector addition and scalar times vector multiplication. In addition, I know how to transform vectors by multiplying them by matrices.