Narrative, Early Abstraction, and Matrix Factorizations in MATH 3260: Linear Algebra I

Timothy E. Faver

October 16, 2026

Narrative

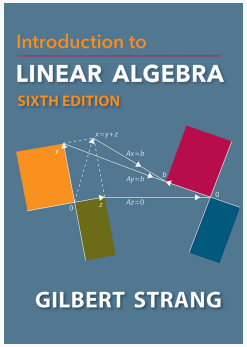
- Course tell stories.



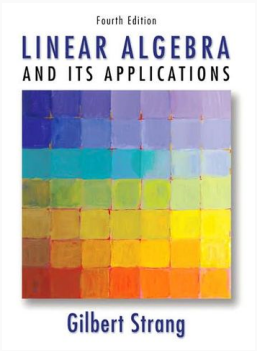
- What is the story of linear algebra?
- Arveson (*A Short Course on Spectral Theory*): "what might be called the fundamental problem of operator theory [is] to calculate spectra of specific operators on infinite-dimensional spaces."
- Maybe: the fundamental problem of linear algebra is to characterize the range of a linear operator.
- Maybe: the story of linear algebra is the story of **understanding** $A\mathbf{x} = \mathbf{b}$. Solving $\subsetneq$ understanding. (This presumes $A\mathbf{x} = \mathbf{b}$ is fundamentally worthwhile, regardless of "applications.")

Disclaimers

- This is the right story for me. Maybe not for you. That's okay!
- Is my story objectively better for students than another story? I don't know how to measure this. But I will talk about some data.
- **Student audience** in 3260 at KSU: 35 engineers + 0 to 1 Math.
- I am okay with 3260 sticking to finite-dimensional spaces, maybe just Euclidean spaces.
- I claim that if you do a search-and-replace ($\mathbb{R}^n \to \mathcal{V}$, $\mathbb{R}^m \to \mathcal{W}$; $A\mathbf{x} = \mathbf{b} \to \mathcal{T}v = w$; $\mathbf{v} \cdot \mathbf{w} \to \langle v, w \rangle$), you get a pretty rigorous finite-dimensional course.
- Very little is original to me. Mostly this originates in Strang's *Introduction to Linear Algebra* 6E.



$\neq$



Why?

- **Narrative** organizes the course and makes everything meaningful. (To me.) Why does something hard/abstract/weird matter? Because it helps us **understand** $A\mathbf{x} = \mathbf{b}$.

- **Early abstraction** introduces abstract concepts (null/column spaces, (in)dependence, pivot columns, rank) **early** as natural tools for asking and maybe answering meaningful questions about $A\mathbf{x} = \mathbf{b}$. There are **fewer surprises** later in the course.

- **Matrix factorizations** organize the narrative and the abstract concepts into the **big ideas**. Strang 6E: each big idea gets paired with a factorization—CR, LU, QR, diagonalization, SVD. (Strang 5E: pair big ideas with problems—square systems, nonsquare systems/least squares, eigenvalues, singular values.) And matrix factorizations are **hugely useful**.

- The plan: I will take you through *early* abstraction and content to at least the invertible matrix theorem. Pause for vocabulary. Maybe rank–nullity and orthogonality. **You** are the target audience.

How the narrative starts

- First day: solve a toy linear system and raise the fundamental questions of existence and uniqueness.

$$\begin{cases} x - 2y = 1 \\ 3x + 2y = 11 \end{cases} \iff \begin{cases} x = 3 \\ y = 1. \end{cases}$$

- Second day: recast in matrix-vector language, no more solving for weeks. To **understand** $A\mathbf{x}$, view it as a linear combination.

$$\begin{cases} x_1 - 2x_2 = 1 \\ 3x_1 + 2x_2 = 11 \end{cases} \iff x_1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + x_2 \begin{bmatrix} -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \end{bmatrix} \iff \begin{bmatrix} 1 & -2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \end{bmatrix}.$$

- **Slogans:** (1) *Matrices are static and dynamic.* They store data (a system's coefficients, factors in matrix factorizations) and they act on data (multiplying vectors, other matrices, doing elimination).

(2) *What things do defines what things are.* What matrices do is multiply: $A = B$ (static) $\iff$ $A\mathbf{v} = B\mathbf{v}$ for all $\mathbf{v}$ (dynamic).

- Dot products are great for small calculations by hand, less so for understanding. **To understand $A\mathbf{x}$, view it as a linear combination.**

Early abstraction starts with the column space

- The goal is now to understand as much as possible about $A\mathbf{x} = \mathbf{b}$ without having a formula/algorithm for solutions.
- **Analyst's creed in ODE:** *having a formula for something is not the same as understanding that thing.* Estimates are better than equalities. ($< > =$.)
- For what $\mathbf{b}$ *can* we solve $A\mathbf{x} = \mathbf{b}$? We want the **column space**

$$\mathbf{C}(A) = \{A\mathbf{v} \mid \mathbf{v} \in \mathbb{R}^n\}$$

to be as big as possible. The column space **controls existence.**

- What can go wrong? Look for patterns in simple examples:

$$\begin{bmatrix} 1 & -2 \\ 3 & -6 \end{bmatrix} \qquad \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 3 \\ 0 & 2 & 4 \\ 0 & 0 & 0 \end{bmatrix}$$

- It looks like $\mathbf{C}(A) \neq \mathbb{R}^m$ when there is **linear redundancy** in the columns. (Disclaimer: A has to be square.) Inefficient data!

Linear redundancy is linear dependence

- The columns of A are **dependent** if one column is a linear combination of the others. **Vocab:** "A is dependent", "the third column is dependent on the others"...
- **Conjecture:** if $A \in \mathbb{R}^{m \times m}$ has dependent columns, then $\mathbf{C}(A) \neq \mathbb{R}^m$.
- Strang's idea: this is an unwieldy definition of dependence because it requires finding a "guilty" column.
- Simple examples: rearrange the linear dependence relation into a nontrivial solution to $A\mathbf{x} = \mathbf{o}_m$.
- Here comes the **null space** $\mathbf{N}(A) = \{\mathbf{v} \in \mathbb{R}^n \mid A\mathbf{v} = \mathbf{o}_m\}$.
- Easy proofs: dependent columns $\iff$ $\mathbf{N}(A) \neq \{\mathbf{o}_n\}$ (null space is too big) $\iff$ if a solution to $A\mathbf{x} = \mathbf{b}$ exists, it's not unique. The null space **controls uniqueness**.
- Math miracle: the one problem $A\mathbf{x} = \mathbf{o}_m$ has something profound to say about all the problems $A\mathbf{x} = \mathbf{b}$.
- **Complain:** we still don't know how to solve $A\mathbf{x} = \mathbf{o}_m$.

So what data matters? The pivot columns

- The story so far: linearly redundant data appears to destroy existence and definitely destroys uniqueness.

- **"Removal" theorem:** the column space is unchanged when you delete "dependent" columns.

$$\mathbf{C}\left(\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & (2\mathbf{a}_1+3\mathbf{a}_2)\end{bmatrix}\right) = \mathbf{C}\left(\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}\right).$$

- Scan the matrix left to right and pull out the columns that are not linear combinations of their predecessors:

$$\begin{bmatrix}0 & 1 & 2 & 0 & 3\\ 0 & 0 & 0 & 1 & 4\\ 0 & 0 & 0 & 0 & 0\end{bmatrix} \qquad \begin{bmatrix}0 & 1 & 2 & 0 & 3\\ 0 & 0 & 0 & 1 & 4\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

- **Pivot columns**, dynamically: the first PC is the first nonzero column, the jth PC is the first column not in the span of the previous $j-1$ PC's. **Rank** $=$ number of PC's $=$ amount of meaningful data.

- Disclaimer: finding PC's $=$ checking linear combinations $=$ solving linear systems. All problems are toys, all numbers are easy.

How do we see meaningful data? Multiply to factor

- How can we represent A transparently in terms of its pivot columns? We factor integers into products of primes and polynomials into products of lower degree stuff...

- Let $C \in \mathbb{R}^{m\times r}$ be the matrix of pivot columns of $A \in \mathbb{R}^{m\times n}$ with $\operatorname{rank}(A) = r$. Can we find a matrix R and define the product CR so that $A = CR$?

$$A = \begin{bmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} \mathbf{a}_2 & \mathbf{a}_4 \end{bmatrix}, \quad \begin{cases} \mathbf{a}_1 = 0\mathbf{a}_2 + 0\mathbf{a}_4 \\ \mathbf{a}_2 = 1\mathbf{a}_2 + 0\mathbf{a}_4 \\ \mathbf{a}_3 = 2\mathbf{a}_2 + 0\mathbf{a}_4 \\ \mathbf{a}_4 = 0\mathbf{a}_2 + 1\mathbf{a}_4 \\ \mathbf{a}_5 = 3\mathbf{a}_2 + 4\mathbf{a}_4 \end{cases}$$

$$R = \begin{bmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix} \qquad \Longrightarrow \qquad \mathbf{a}_j = C\mathbf{r}_j.$$

- We get $A = CR$ if we put

$$C \begin{bmatrix} \mathbf{r}_1 & \cdots & \mathbf{r}_n \end{bmatrix} = \begin{bmatrix} C\mathbf{r}_1 & \cdots & C\mathbf{r}_n \end{bmatrix}.$$

Strang's CR-factorization

- $$\begin{bmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

- This is **static** matrix multiplication: matrix multiplication expresses useful data about a matrix.

- Look at R: there is an identity matrix.

$$\begin{bmatrix} 0 & 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 0 & 4 \end{bmatrix} P$$

- Any nonzero $A \in \mathbb{R}^{m \times n}$ factors (blocks! permutations!) as

$$A = CR = C \begin{bmatrix} I_r & F \end{bmatrix} P.$$

- Strang's *ILA* 5E introduced matrix multiplication in the context of encoding Gaussian elimination. Too much all at once?

It's time to solve stuff

- This is the end of early abstraction: we know column and null spaces, linear (in)dependence, pivot columns, rank, and a matrix factorization. We want an unambiguous, universal algorithm for figuring stuff out. Things will look more familiar now.

- The strategy comes from the example on the first day: the problem $A\mathbf{x} = \mathbf{b}$ is easier to solve when it's actually $U\mathbf{x} = \mathbf{c}$ with U upper-triangular and the diagonal entries of U are nonzero.

- How to turn A into U? This is **Gaussian elimination**. Gloriously, we can encode it with **dynamic** matrix multiplication: matrix multiplication acts on vectors and other matrices.

- Every step of elimination needs to be paired with an elementary matrix (elimination or permutation only here, no scaling needed). How do you find those matrices? *What things do defines what things are.* Look for linear combinations.

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \mapsto \begin{bmatrix} v_1 \\ v_2 - 3v_1 \end{bmatrix} = v_1 \begin{bmatrix} 1 \\ -3 \end{bmatrix} + v_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Only invert what you need to invert

- Gaussian elimination gives E such that EA is upper-triangular and $E =$ product of elimination and/or permutation matrices.

$$A\mathbf{x} = \mathbf{b} \Longrightarrow EA\mathbf{x} = E\mathbf{b} \iff U\mathbf{x} = \mathbf{c},\ U = EA,\ \mathbf{c} = E\mathbf{b}.$$

- The problem $EA\mathbf{x} = E\mathbf{b} \iff U\mathbf{x} = \mathbf{c}$ is easier to solve. Do we have

$$EA\mathbf{x} = E\mathbf{b} \Longrightarrow A\mathbf{x} = \mathbf{b}?$$

$$\Longrightarrow \Longrightarrow \Longleftarrow ?$$

- Of course: E needs to be **invertible**. Get this quickly: (1) elimination and permutation matrices are invertible (. . .what things do. . .) and (2) products of invertible matrices are invertible.

- **Do not waste time on inverting matrices in general with** $\begin{bmatrix} A & I_m \end{bmatrix} \to \begin{bmatrix} I_m & A^{-1} \end{bmatrix}$**.** Maybe do LU instead for actually solving $A\mathbf{x} = \mathbf{b}$. "[O]ther than in [the] simple 2×2 case, there is no valid reason ever to compute a determinant or a matrix inverse; all other algorithms for large scale systems and matrices rely on alternative, more efficient, and more accurate algorithms."—Calder & Olver's new *Linear Algebra, Data Science, and Machine Learning*

The invertible matrix theorem

- **Do** spend a little time inverting upper-triangular matrices via **Gauss–Jordan** ("upwards") elimination. This introduces scaling matrices. Elementary = elimination, permutation, scaling.

- Three passes at elimination: Gaussian (downwards) to get square $A \to U$, Gauss–Jordan (upwards) to get square $U \to I_m$, and (later) GJ on nonsquare A. "That algorithm is atypical of numerical linear algebra, exceptionally difficult to analyze, yet at the same time tediously familiar to every student."—Trefethen & Bau's *NLA*.

- Use some results about upper-triangular matrices to prove that all are equivalent:

(1) A is invertible.

(2) ($\exists +!$) $A\mathbf{x} = \mathbf{b}$ always has a unique solution.

(3) ($\exists$ only) $A\mathbf{x} = \mathbf{b}$ always has a solution.

(4) ($!$ only) If a solution to $A\mathbf{x} = \mathbf{b}$ exists, it is unique.

(5) (no linearly redundant data) The columns of A are independent.

Vocabulary: the journey so far

- Lake & Erik (Fa22 TC): students clearly struggle with vocabulary in 3260 and MATH 2345: Discrete Mathematics. How do we help?
- What I learned: my responsibility is to emphasize the study of vocabulary as a distinct technique (maybe new?) within the broader "math techniques" of the course. How to reinforce, assess, reward?
- 3260 in Sp23 (my first time): semester-long glossary project in which students selected and **re-**evaluated the 10 most important concepts in the course. The good: engagement, creativity, ownership, conversation. The less good: quiz and exam scores still showed significant vocab struggle, in-class questions, too.
- 4260 (Linear II) in Fa24 (first time): 4 **mean** vocab quizzes spaced throughout the term. Cumulative. Questions asked for **examples and nonexamples**, not just definitions.
- 3260 onwards from Sp25: weekly quizzes with cumulative vocab questions. Exams test vocab in "survey" questions (decoupled parts) and "scaffolded" questions (define the concept, then use it).

Vocabulary: assessment and support

- **Definition-type question:** What does it mean for a vector $\mathbf{v} \in \mathbb{R}^m$ to be a **linear combination** of the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^m$?
- **Example-type question:** Give an example of a 2**-dimensional subspace** of $\mathbb{R}^3$ and a basis for your subspace.
- **Nonexample-type question:** Give an example of a set $\mathcal{V}$ in $\mathbb{R}^p$ (your choice of $p \geq 2$) such that $\mathcal{V}$ is *not* a **subspace** of $\mathbb{R}^p$. Briefly explain why your $\mathcal{V}$ is not a subspace of $\mathbb{R}^p$.
- **Scaffolded-type question:** Part (i) asks for the definition of a subspace and part (iii) asks for a proof that $\mathbf{C}(A)$ is a subspace.
- Candidate concepts for quizzes and exams are clearly indicated daily in lecture notes. Special indication of nonexamples.
- I adapted the glossary project into an optional "glossary template" that students can fill in: concept, definition, examples, nonexamples, evolving role in the course, other stuff.
- I give a sample template filled in for Calc I-level **continuity**.

Vocabulary: outcomes

- Cumulative assessment (concepts from Week 2 on a quiz in Week 12) requires constant study. Vocabulary never stops being used.
- Weekly data scores give me some insight into where students are overall struggling or qualitative trends involving expression ("The matrix is dependent"). How do I help?
- Quality of vocabulary responses on scaffolded problems illuminates quality of computational/proof parts: failure to define "subspace" $\mapsto$ failure to prove that $\mathbf{C}(A)$ is a subspace.
- Memorizing definitions is the bare minimum of effort.
- Repeated assessment of the *same* concept can indicate growth.
- In-class discussions rarely get held up by "what does that word mean?" and focus more on "why does the reasoning work this way?"
- Individual diagnosis of struggle in office hours/email frequently boils down to lack of fluency with vocabulary.
- **None of this is really specific to MATH 3260—thanks, Lake & Erik!**

$A\mathbf{x} = \mathbf{b}$ in general: a renewed role of the RREF

- Simple examples suggest that existence or uniqueness always fails for $A\mathbf{x} = \mathbf{b}$ when A is nonsquare. How to prove this and quantify/qualify failure?

- Quantification is dimension and rank–nullity. Qualification is orthogonality (maybe later): $\mathbf{C}(A) = \mathbf{N}(A^{\mathsf{T}})^{\perp}$, range $=$kernel$^{\perp}$.

- Start with the simpler problem $A\mathbf{x} = \mathbf{o}_m$ for $A \in \mathbb{R}^{m\times n}$ and $r = \operatorname{rank}(A) \geq 1$ to get $\mathbf{N}(A)$ as a "special" (Strang) column space:

$$A = CR,\ R = \begin{bmatrix} I_r & F \end{bmatrix} P \qquad \Longrightarrow \qquad \mathbf{N}(A) = \mathbf{N}(R) = \mathbf{C}\left(P^{-1} \begin{bmatrix} -F \\ I_{n-r} \end{bmatrix} \right).$$

- If you know CR, you know $\mathbf{N}(A)$ exactly. How do you get CR?

- This is the RREF: do Gauss–Jordan elimination on any matrix to get

$$EA = R_{\mathrm{O}} = \begin{bmatrix} R \\ \mathrm{O} \end{bmatrix} \qquad \text{and} \qquad A\mathbf{x} = \mathbf{b} \iff R_{\mathrm{O}}\mathbf{x} = \mathbf{c} \text{ ("easier").}$$

- **The price of early abstraction:** proving that the columns of the RREF with leading 1's are its (dynamic!) pivot columns is hard.

Quantifying failure via dimension

- Rank measures size: the number of pivot columns spanning $\mathbf{C}(A)$ and expressing $\mathbf{N}(A)$ as a "special" column space.
- How are these two roles of rank "the same"? How are $\mathbf{C}(A)$ and $\mathbf{N}(A)$ "the same"? Here is a payoff of early abstraction.
- **Subspaces:** what $\mathbf{C}(A)$ and $\mathbf{N}(A)$ have in common is probably not their vectors but how their vectors interact dynamically via the fundamental linear operations of vector addition and scalar multiplication.
- **Bases:** $\mathbf{C}(A)$ and $\mathbf{N}(A)$ are spans of independent lists (pivot columns, "special" solutions). A basis is a unique coordinate system. Every subspace is a column space (useful?).
- **Dimension:** no basis is more efficient than another. Prove in the context of rank as a "threshold" for $\mathbf{C}(A)$: lists shorter than rank can't span, lists longer are dependent. . .
- Rank–nullity quantifies breadth of solvability vs. degrees of freedom in the precise language of subspace and dimension.

Orthogonality is all about solving the unsolvable problem

- The narrative so far is absolute: either $\mathbf{b} \in \mathbf{C}(A)$ and we can solve $A\mathbf{x} = \mathbf{b}$ (maybe with degrees of freedom) or $\mathbf{b} \notin \mathbf{C}(A)$ and we can't.

- Do we give up when $\mathbf{b} \notin \mathbf{C}(A)$? **No!** Replace
the **unsolvable** problem $A\mathbf{x} = \mathbf{b}$
with a **solvable** problem $A\widehat{\mathbf{x}} = \mathbf{p}$ for a good choice of $\mathbf{p} \in \mathbf{C}(A)$.

- Use the Euclidean norm (approximations), the dot product to (orthogonality), and the transpose (because examples) to conjecture every $\mathbf{b} \in \mathbb{R}^m$ can be written uniquely as $\mathbf{b} = \mathbf{p} + \mathbf{w}$ for $\mathbf{p} \in \mathbf{C}(A)$ and $\mathbf{w} \in \mathbf{N}(A^{\mathsf{T}})$, and $\mathbf{p}$ is the best approximation in $\mathbf{C}(A)$ to $\mathbf{b}$.

- Proving the conjecture requires properties of transposes (it "pops": $A\mathbf{v} \cdot \mathbf{w} = \mathbf{v} \cdot A^{\mathsf{T}}\mathbf{w}$), $\mathbf{C}(A) \perp \mathbf{N}(A^{\mathsf{T}})$, and projections.

- Along the way, get the **fundamental theorem of linear algebra** $\mathbb{R}^m = \mathbf{C}(A) \oplus \mathbf{N}(A^{\mathsf{T}})$ and $\mathbb{R}^n = \mathbf{N}(A) \oplus \mathbf{C}(A^{\mathsf{T}})$ and a **geometric characterization of the range** as $\mathbf{C}(A) = \mathbf{N}(A^{\mathsf{T}})^{\perp}$.

- Projections $A(A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}$ stink, so use **orthonormal** bases: the best. Introduce QR to turn least squares $A\widehat{\mathbf{x}} = \mathbf{p}$ into back-sub.

What's next?

- Some reorganization of the treatment of orthogonality: writing this talk made me refocus on least squares and approximation.
- Give me one more week in the semester: SVD time
- Two more weeks: abstract finite-dimensional vector spaces and matrix representations of linear operators. How do the matrix representations for

$$p(x) \mapsto p'(x) \qquad \text{and} \qquad p(x) \mapsto \int_0^x p(s)\, ds$$

on polynomial spaces reflect what we know about derivatives and integrals from calculus?
- -1 more weeks?!
- Can I bring more computational ideas to 3260 (like why A^{-1} is bad)? I should teach 3261...
- 4260: can I write the great American advanced linear algebra novel that brings infinite-dimensional analysis to KSU undergrads via "familiar" concepts from calculus?