Example 94. (warmup) Consider $A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$.

- What are the eigenspaces?
- What are A^{-1} and A^{100}?

Solution.

- $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is a 2-eigenvector, and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is a 3-eigenvector. In other words, the 2-eigenspace is $\operatorname{span}\left\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\}$ and the 3-eigenspace is $\operatorname{span}\left\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\}$.
- $A^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/3 \end{bmatrix}$ and $A^{100} = \begin{bmatrix} 2^{100} & 0 \\ 0 & 3^{100} \end{bmatrix}$

Comment. Algebraically, this looks like a very simple map. However, notice that it is not so easy to say what happens to, say, $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ geometrically. That is because two things are happening: part of that vector is scaled by 2, the other part is scaled by 3.

Example 95. If A has λ-eigenvector $\boldsymbol{v}$, then what can we say about A^2?

Solution. A^2 has λ^2-eigenvector $\boldsymbol{v}$.
[Indeed, $A^2\boldsymbol{v} = A(A\boldsymbol{v}) = A(\lambda\boldsymbol{v}) = \lambda A\boldsymbol{v} = \lambda^2\boldsymbol{v}$. This is even easier in words: multiplying $\boldsymbol{v}$ with A has the effect of scaling it by λ; hence, multiplying it with A^2 scales it by λ^2.]
Important comment. Similarly, A^{100} has λ^{100}-eigenvector $\boldsymbol{v}$.

Example 96. If a matrix A can be diagonalized as $A = PDP^{-1}$, what can we say about A^n?

Solution. First, note that $A^2 = (PDP^{-1})(PDP^{-1}) = PD^2P^{-1}$. Likewise, $A^n = PD^nP^{-1}$.

The point being that D^n is trivial to compute because D is diagonal.

In particular. $A^{-1} = PD^{-1}P^{-1}$

Example 97. (extra) If $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$, then what is A^n.

Solution. We compute directly that $A^2 = \begin{bmatrix} 2^2 & & \\ & 3^2 & \\ & & 4^2 \end{bmatrix}$. It then becomes obvious that $A^n = \begin{bmatrix} 2^n & & \\ & 3^n & \\ & & 4^n \end{bmatrix}$.

Comment. As done above, it is common to leave zero entries of a matrix blank to emphasize the structure of that matrix.

Example 98. Though they use different language, the following statements are equivalent:

A is not invertible
$\iff$ $A\boldsymbol{x} = \boldsymbol{0}$ has a solution besides $\boldsymbol{x} = \boldsymbol{0}$
$\iff$ $\dim \operatorname{null}(A) > 0$
$\iff$ $\det(A) = 0$
$\iff$ 0 is an eigenvalue of A

Comment. It is important that we are able to "talk" using the basic notions of linear algebra. If the above statements don't make perfect sense, please review or check with me.

Armin Straub
straub@southalabama.edu

Example 99. Let 3×3 be the matrix A of a projection onto a plane (containing the origin). What are the eigenspaces? Is A invertible, orthogonal, symmetric?

Of course, we already know a lot about projections. The point is to think about these properties from the perspective of eigenvalues and eigenvectors.

Solution. Let us write $W = \text{span}\{\boldsymbol{w}_1, \boldsymbol{w}_2\}$ for the plane we're projecting onto.

- The eigenvalues of A are $1, 1, 0$.
  The 1-eigenspace is W (2-dimensional), and the 0-eigenspace is $W^\perp$ (1-dimensional).
  [Make sure this makes sense!]

- A is not invertible (because 0 is an eigenvalue) and therefore also cannot be orthogonal.
  A is indeed symmetric. This is a bit more tricky to see but is a consequence of the eigenspaces being orthogonal and the eigenvalues real (see spectral theorem): we can therefore choose an orthonormal basis for the matrix P in the diagonalization $A = PDP^{-1}$, so that $A = PDP^T$. But the latter is symmetric because $(PDP^T)^T = (P^T)^T D^T P^T = PDP^T$.

Comment. This gives us another way to compute projection matrices: using the eigenvalues and eigenvectors, we can write down the matrices P, D for the diagonalization $A = PDP^{-1}$.

Comment. Why is there the condition that the plane we reflect through contains the origin? A linear map $\boldsymbol{x} \mapsto A\boldsymbol{x}$ given by a matrix A must have the property that $\mathbf{0} \mapsto \mathbf{0}$ (i.e. the origin is fixed). To talk about other kinds of projections, we would need to consider **affine maps** $\boldsymbol{x} \mapsto A\boldsymbol{x} + \boldsymbol{b}$.

How little we actually know!

Q: *How fast can we solve N linear equations in N unknowns?*

Estimated cost of Gaussian elimination:

$$\begin{bmatrix} \blacksquare & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \vdots & & & \vdots \\ 0 & * & * & \cdots & * \end{bmatrix}$$

- to create the zeros below the first pivot:
  $\Longrightarrow$ on the order of N^2 operations
- if there is N pivots total:
  $\Longrightarrow$ on the order of $N\cdot N^2 = N^3$ operations

- A more careful count places the cost at $\sim\frac{1}{3}N^3$ operations.
- For large N, it is only the N^3 that matters.
  It says that if $N \to 10N$ then we have to work 1000 times as hard.

That's not optimal! We can do better than Gaussian elimination:

- Strassen algorithm (1969): $N^{\log_2 7} = N^{2.807}$
- Coppersmith–Winograd algorithm (1990): $N^{2.375}$
- ... Stothers–Williams–Le Gall (2014): $N^{2.373}$ (If $N \to 10N$ then we have to work 229 times as hard.)

Is $N^{2+(\text{a tiny bit})}$ possible? **We don't know!** (People increasingly suspect so.) (Better than N^2 is impossible; why?)

Comment. The above algorithms actually are for computing matrix products. It can be shown that, if $M(N)$ is the cost for multiplying two $N\times N$ matrices, then $N\times N$ systems can also be solved for cost on the order of $M(N)$. In other words, we don't even know how costly it is to multiply two matrices.

Good news for applications:

- Matrices typically have lots of structure and zeros
  which makes solving so much faster.