

[Get started](#)[Open in app](#)

Jeremy Kun

2.6K Followers

About

Follow



Here's just a fraction of what you can do with linear algebra



Jeremy Kun Feb 26, 2015 · 4 min read

The next time someone wonders what the point of linear algebra is, send them here. I write a blog on [math and programming](#) and I see linear algebra applied to computer science all the time. Here's a short list that comprises a small fraction of the things you can do with linear algebra.

Ranking in search engines: The most high-profile use of linear algebra (whether or not you know it) was in the creation of Google. Their original ranking algorithm, which has since surely become far more complicated, used a lot of linear algebra to rank which webpages should show up first. More generally, any time you want to analyze a random walk in a network, you're likely to need some linear algebra. Here's a [series of blog posts I wrote](#) deriving the method.

Linear programming: The most widely used application of linear algebra is definitely optimization, and the most widely used kind of optimization is linear programming. You can optimize budgets, your diet, and your route to work using linear programming, and this only scratches the surface of the applications. [Here's a series](#) (still in progress) on the mathematics behind linear programming. The primary technique for solving them, called the simplex algorithm, is essentially a beefed up Gaussian elimination.

Error correcting codes: Another unseen but widespread use of linear algebra is in coding theory. The problem is to encode data in such a way that if the encoded data is tampered with a little bit, you can still recover the unencoded data. Such schemes are called error correcting codes, and the simplest ones encode data as vectors in a vector

space. Error correcting codes are used in DVDs to prevent scratches from ruining a movie. They're also used on deep space probes to transmit data back to Earth, and they allowed us to get the first ever close-up pictures of Saturn and Jupiter. Here's an article describing the simplest kind of error correcting code, the Hamming code.

Signal analysis: The field of signal analysis gives one massively useful tools for encoding, analyzing, and manipulating “signals” that can be audio, images, video, or things like x-rays and light refracting through a crystal. The simplest way to understand the Fourier transform is as a linear map that performs a change of basis. Fourier analysis has even been used to make art. Here's the first post in a series deriving Fourier analysis from scratch, although much of it can be abbreviated, skipped, or skimmed if you have a strong understanding of linear algebra. A discrete cousin of Fourier analysis has been part of many theoretical techniques in computer science as well.

Graphics: Pretty much all graphics innovation since computers have existed have come from video games and movies. The central part of graphics is projecting a three-dimensional scene onto a two-dimensional screen. Projection is already a linear map. On top of that, rotations, scaling, and perspective are all implemented and analyzed properly using linear algebra.

Facial recognition: A cool (but not the best) method for doing automated facial recognition uses a linear algebraic technique called *principal component analysis*. Essentially this is just finding a particularly good basis to represent a database of face images, and using eigenvectors (“eigenfaces”) to rebuild the images. Here is an article describing that method without prior knowledge of PCA, and a more general article showing PCA for any dataset. Here is a picture of what an “eigenface” might look like.



Prediction: The simplest models of prediction are linear models, and these are developed and understood with linear algebra. Here, for example is an article describing how to do [linear regression](#).

Community detection: The leading methods for detecting communities in networks of people (or any other kind of network) use a linear-algebraic tool called the spectrum of the network. Like ranking webpages, community detection techniques also rely on random walks. [Here is an overview](#) of some notions of community detection.

Greedy algorithms: Greedy algorithms are characterized by a kind of generalization of linear systems called a matroid. In other words, every problem solvable by a greedy algorithm can be represented as a matroid and every matroid can be optimized by a greedy algorithm. Understanding linear algebra is not a requirement to understand matroids, but it makes the process much easier. [Here is an article](#) proving what I just said.

Quantum computing: All of quantum computing is literally just linear algebra, as is general quantum mechanics. You can understand how quantum computers can break cryptosystems without any physics, as long as you understand linear algebra. [Here is the first](#) of a series of articles (in progress) doing this, and it gives you an idea of how linear algebra is the primary tool.

There are vastly more applications of linear algebra than I can list here, from cryptography to data analysis to medical imaging and beyond. But the point is clear: a strong foundation in linear algebra is all *kinds* of useful.

[Linear Algebra](#) [Mathematics](#) [Programming](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app



