3 LU factorization

Matrices can do neat things:

- Blur an image
- Traverse a graph
- Rotate geometry

Can we come up with a generic "undo" button for these things?

(... that does *not* depend on the application) (!)

To warm up, let's try this for matrices where it is super-easy.

Example: Upper triangular matrices

What do we do about more general matrices?

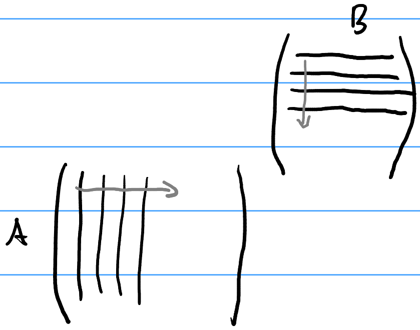
What's the difference between REF and an upper triangular matrix?

What happens if you don't just eliminate downward, but also upward?

So we could implement Gaussian elimination every time we would like

w

"store" the work we've put in, to reuse it with another right-hand side b?



So how do you represent an elimination step as a matrix?

Are elimination matrices invertible?

With enough elimination matrices, we should be able to arrive at REF...

What happens if we combine many elimination matrices like that?

We could rearrange that relationship to get a factorization of A!

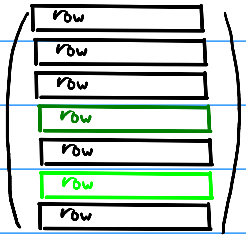
Does an LU factorization help us solve $Ax = b$?

So is LU/Gaussian elimination bulletproof?

So is our process just too stupid to find the LU factorization?

How are we going to fix this mess?

How do we swap rows in matrix notation?



What's the inverse of a permutation matrix?

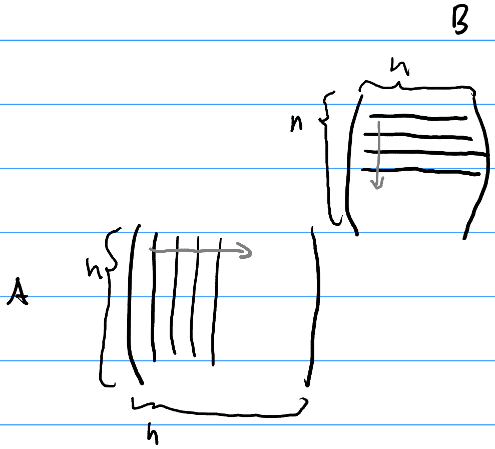
How do we combine partial pivoting with the elimination matrices?

That has made quite a mess of our LU factorization, right?

So... how do we sort out this mess?

So... how do we sort out this mess? (cont'd)

Let's talk about computational cost. What is the asymptotic cost of multiplying two n x n matrices?



So how expensive is LU factorization?

Can LU deal with non-square matrices?