Overview

- CN: cost
- CN: apps (CA)
- CN for interpolation

Q: - ill vs. well
- P mat

border line "well" → $10^{-10} \leq 10^{5} \cdot 10^{-15}$

rel. out err, cond #, rel. inp. err

ill ⇒ $10^{0} \leq 10^{5}$ 10^{-4}

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 6 & 1 \\ 1 & \partial & \partial \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 7 \\ \partial \end{pmatrix}$$

Computational Cost

- What is the computational cost of multiplying two $n \times n$ matrices?

$$O(n^3)$$

- What is the computational cost of carrying out LU factorization on an $n \times n$ matrix?

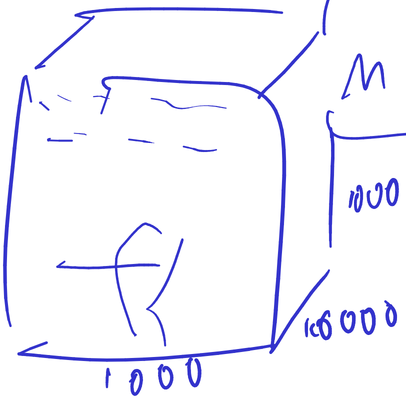
$O(n^2)$

$n \cdot \cancel{O(n^3)}$

$$\underbrace{M\,P\,M\,P\,M \cdot P}_{n} \cdot A$$

$\rightsquigarrow$ LU not really $O(n^4)$

$\rightsquigarrow$ LU actually: $O(n^3)$



M. any matrix : $O(n^2)$

P. any matrix : $O(n^2)$

Demo: Complexity of Mat-Mat multiplication and LU

n goes up by 10^6

$$1000 \times 1000 \rightsquigarrow 1s$$

$$10^9 \times 10^9 \rightsquigarrow (10^6)^3 \cdot 1s$$

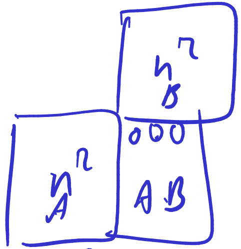
More cost concerns

- What's the cost of solving $A\boldsymbol{x} = \boldsymbol{b}$?

- What's the cost of solving $A\boldsymbol{x} = \boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$?

- What's the cost of finding A^{-1}?

Cost: Worrying about the Constant, BLAS



$O(n^3)$ really means

$$\alpha \cdot n^3 + \beta \cdot n^2 + \gamma \cdot n + \delta.$$

n^2 data $\to$ n^3 computation

All the non-leading and constants terms swept under the rug. But: at least the leading constant ultimately matters.

lots of opportunity for data reuse

Getting that constant to be small is surprisingly hard, even for something deceptively simple such as matrix-matrix multiplication.

opportunity for cache mgmt.

Idea: Rely on library implementation: BLAS (Fortran)

Level 1	$\boldsymbol{z} = \alpha \boldsymbol{x} + \boldsymbol{y}$	vector-vector operations $O(n)$ `?axpy`
Level 2	$\boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{y}$	matrix-vector operations $O(n^2)$ `?gemv`
Level 3	$C = A\,B + \beta C$	matrix-matrix operations $O(n^3)$ `?gemm`

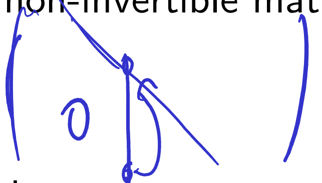
BLAS

LAPACK: Implements 'higher-end' things (such as LU) using BLAS
Special matrix formats can also help save const significantly, e.g.

- banded
- sparse

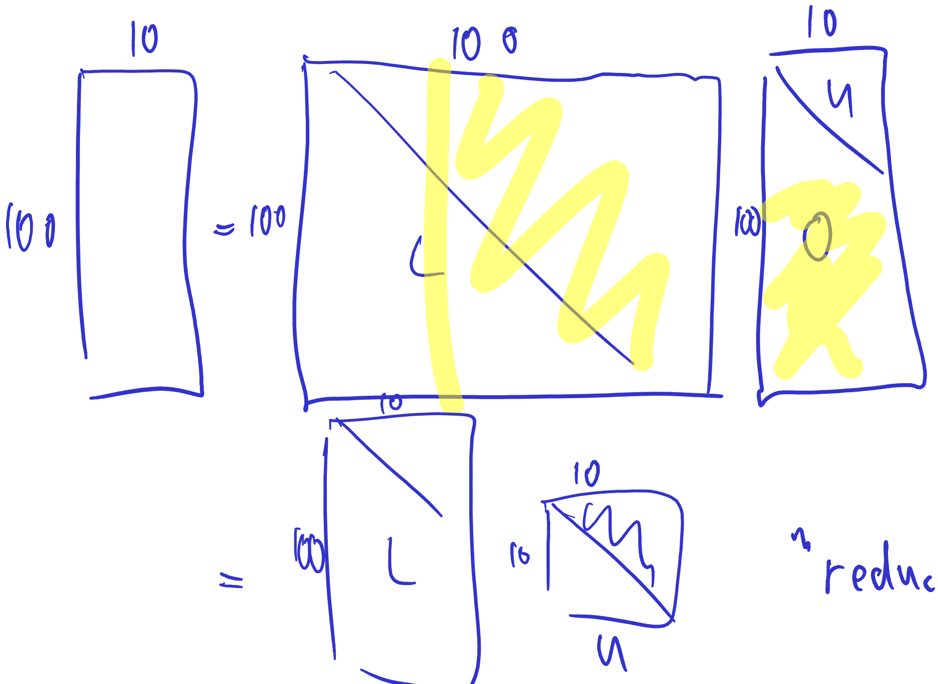
LU: Special cases

- What happens if we feed a non-invertible matrix to LU?



PA = LU

- What happens if we feed LU an $m \times n$ non-square matrices?



"reduced LU"

In-class activity: LU factorization 2

9 LU: Applications

9.1 Linear Algebra Applications

Solve a Linear System

- LU factorization gives us

$$PA = LU, \quad \rightarrow \quad A = P^T L U$$

so that P is a permutation matrix, L is lower triangular, U is upper triangular. How does that help solve a linear system $A\boldsymbol{x} = \boldsymbol{b}$?

$$Ax = b$$

$$P^T L U x = b$$

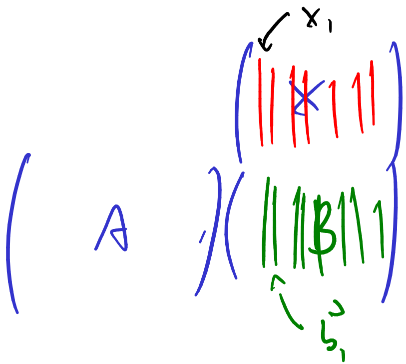
$$L U x = P b$$

$$L y = P b \quad \text{fw subst}$$

$$U x = y \quad \text{bw subst.}$$

Solve a Matrix Equation

- Suppose we want to solve $AX = B$.
  A and B are given, X is unknown.
  (Assume: square and have same size) How can we do that using LU?



$A\vec{x}_1 = \vec{b}_1$

$A\vec{x}_2 = \vec{b}_2$

↳ solve column-by-column

no: solve n linear systems: $O(n^4)$?

yes: ① calculate $L\,U$: $O(n^3)$

② $n \times$ fw/bw subst: $n \cdot O(n^2) = O(n^3)$

Compute an Inverse

- Suppose we want to compute the inverse A^{-1} of a matrix A. How do we do that using LU?

solve $AX = I$

- What's the computational cost of doing so?

$O(n^3)$

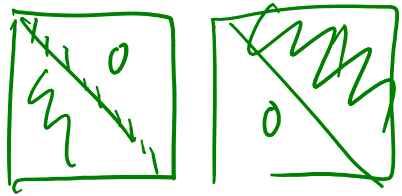
Find the Determinant of a Matrix

- How can we find the determinant of a matrix using LU?

$$PA = LU \rightsquigarrow A = P^T L U$$

$$\det(A) = \det(P^T) \cdot \det(L) \cdot \det(U)$$

$$\pm 1$$



$$1$$

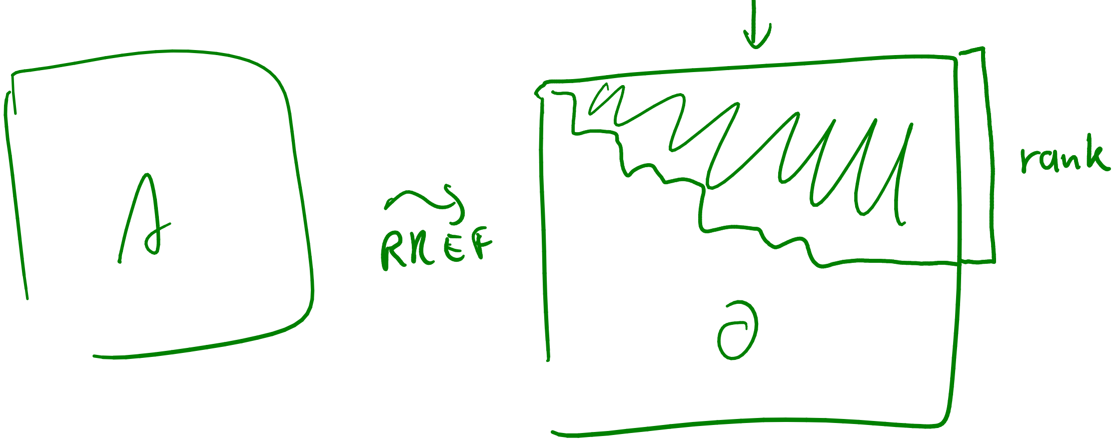
$$u_{11} \cdot u_{22} \cdot \cdots u_{nn}$$

$$O(n^3)$$

Find Row Echelon Form... if we can?

Is that U? No!

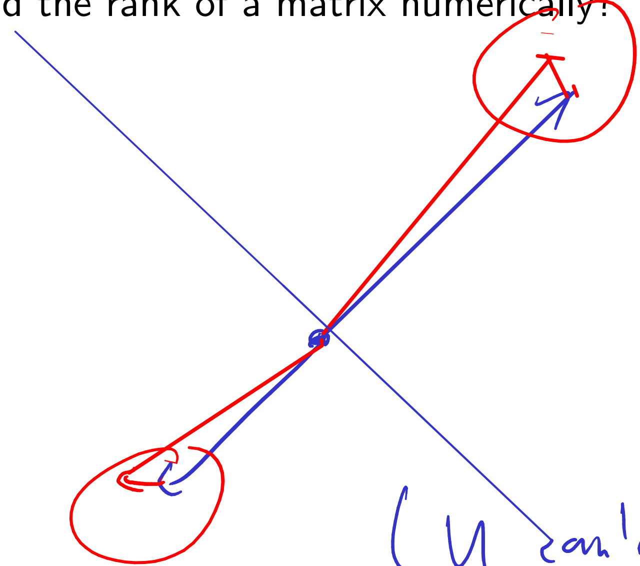
- The factor U in pivoted LU looks like it is in upper echelon form. Is it?



$\text{rank}(A) = \dim(\text{colspace}) = \dim(\text{rowspace})$

Finding the Rank of a Matrix Numerically... if we can?

- Can we find the rank of a matrix numerically?



To ask about rank, we need a tolerance

LU can't do tolerances
can't do RREF
$\Big\} \to$ postpone

9.2 Interpolation

points functions

$$\begin{pmatrix} x_1^0 & \cdots & x_1^5 \\ \vdots & & \vdots \\ x_n^0 & & x_n^5 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_5 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Recap: Interpolation

Starting point: Looking for a linear combination of functions φ_i to hit given data points (x_i, y_i).

Interpolation becomes solving the linear system:

$$y_i = f(x_i) = \sum_{j=0}^{N_{\text{func}}} \alpha_j \underbrace{\varphi_j(x_i)}_{V_{ij}} \qquad \leftrightarrow \qquad V\boldsymbol{\alpha} = \boldsymbol{y}.$$

$$V = \begin{pmatrix} \varphi_1(x_1) & \cdots & \\ \vdots & & \\ \varphi_1(x_n) & & \varphi_n(x_n) \end{pmatrix}$$

Want unique answer: Pick $N_{\text{func}} = N \to V$ square.

V is called the (generalized) Vandermonde matrix.

$$V\vec{\alpha} = \vec{y}$$

Main lesson:

$$V(\text{coefficients}) = (\text{values at nodes}).$$

Rethinking Interpolation

We have so far always used monomials $(1, x, x^2, x^3, \ldots)$ and equispaced points for interpolation. It turns out that this has *significant problems*.

Demo: Monomial interpolation

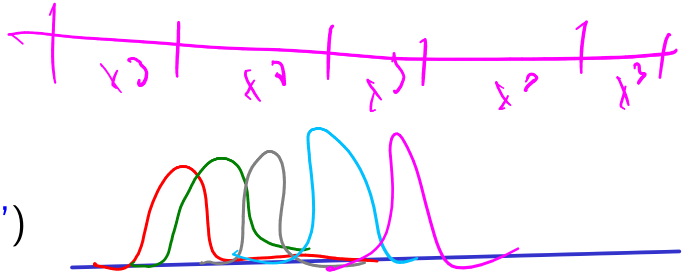
Demo: Choice of Nodes for Polynomial Interpolation

Interpolation: Choosing Basis Function and Nodes

Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

- Monomials $1, x, x^2, x^3, x^4, \ldots$
- Functions that make $V = I$ $\rightarrow$ 'Lagrange basis'
- Functions that make V triangular $\rightarrow$ 'Newton basis'
- Splines (piecewise polynomials)
- Orthogonal polynomials
- Sines and cosines
- 'Bumps' ('Radial Basis Functions')



Ideas for nodes:

- Equispaced
- 'Edge-Clustered' (so-called Chebyshev/Gauss/... nodes)

Better Conditioning: Orthogonal Polynomials

- What caused monomials to have a terribly conditioned Vandermonde?
- What's a way to make sure two vectors are *not* like that?
- But polynomials are functions!

- But how can I practically compute the Legendre polynomials?