$$\begin{pmatrix} & & \\ & & \\ 1 & & \\ & & \\ & & \end{pmatrix}\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

$P[3,1] \rightsquigarrow$ first row $\to$ third row

$$M_2 P_2 M_1 P_1 A = U \qquad L_1 = P_2 M_1^{-1} P_2^{-1}$$

$$A = P_1^{-1} M_1^{-1} P_2^{-1} M_2^{-1} U \qquad L_2 = M_2^{-1}$$

$$\stackrel{?}{=} P_1 P_2 \, L_1 L_2 U$$

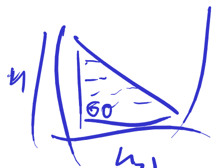
$$= \overbrace{P_1 P_2}^{P} \underbrace{P_2 M_1^{-1} P_2^{-1}}_{L_1} \underbrace{M_2^{-1}}_{L_2} U = PLU$$

$O(n^3)$ $O(\cancel{n^4})$ $O(n^3)$

Demo: Complexity of Mat-Mat multiplication and LU

In-class activity: Pivoting and Cost

$O(n^3)$ $O(n^2)$ $O(n^2)$



More cost concerns

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

$A = LU$

$L(\underbrace{Ux}_{y}) = b$

$O(n^3) \leftarrow LU$ + Fw subst + Bw subst

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

$\underbrace{LU + \underbrace{\text{Fw subst} + \text{Bw subst}}_{n \text{ times} \quad O(n^2)}}_{O(n^3)}$

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

$$A x_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \cdots \quad A x_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \leftarrow \text{solving w/ } n \text{ RHS}$$

$O(n^3)$

$$A^{-1} = \begin{pmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{pmatrix}$$

Cost: Worrying about the Constant, BLAS

$O(n^3)$ really means

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

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

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

Idea: Rely on library implementation: BLAS (Fortran)

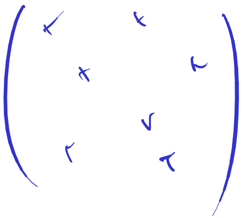
Level 1	$\boldsymbol{z} = \alpha \boldsymbol{x} + \boldsymbol{y}$	vector-vector operations $O(n)$ `?axpy`
Level 2	$\boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{y}$	matrix-vector operations $O(n^2)$ `?gemv`
Level 3	$C = A\,B + \beta C$	matrix-matrix operations $O(n^3)$ `?gemm`, `?trsm`

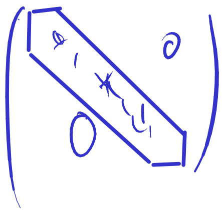
LAPACK: Implements 'higher-end' things (such as LU) using BLAS

Special matrix formats can also help save const significantly, e.g.

- banded $\quad n\ ba^{n}$
- sparse (see hw0)

$\underbrace{\text{dge}}\text{mm}$



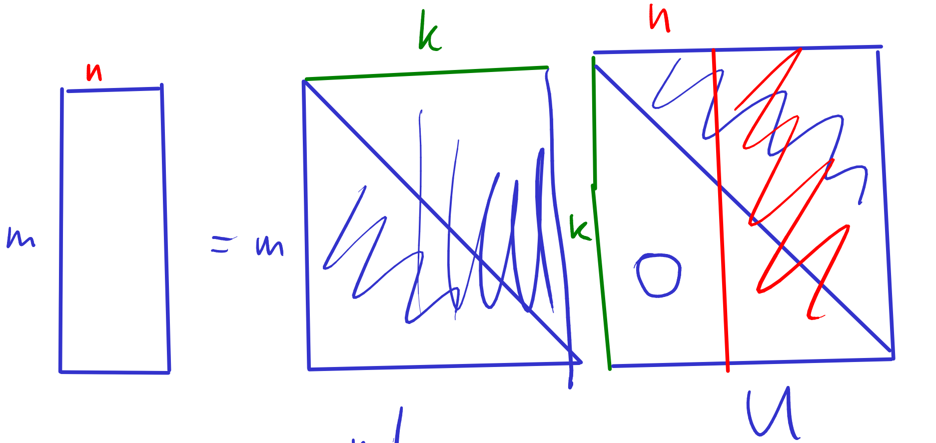


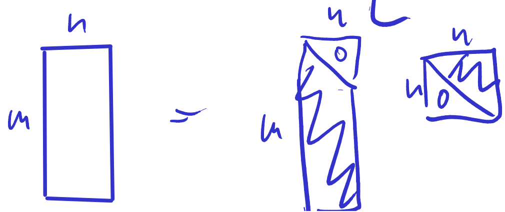
LU: Special cases

- 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"

Changing matrices

$Ax = b_1, \quad Ax = b_2$

- Seen: Cheap to re-solve if RHS changes. (Able to keep the expensive bit, the LU factorization) What if the *matrix* changes?

$$\hat{A} = A + \vec{u}\vec{v}^T$$

Sherman-Morrison

Solve $\hat{A}x = b$

1. Solve $Ay = b \rightsquigarrow y = A^{-1}b$
2. solve $Az = u \rightsquigarrow z = A^{-1}u$
3. $y - \dfrac{zv^Ty}{1 + v^Tz}$

Demo: Sherman-Morrison

3 Linear Least Squares

3.1 Introduction

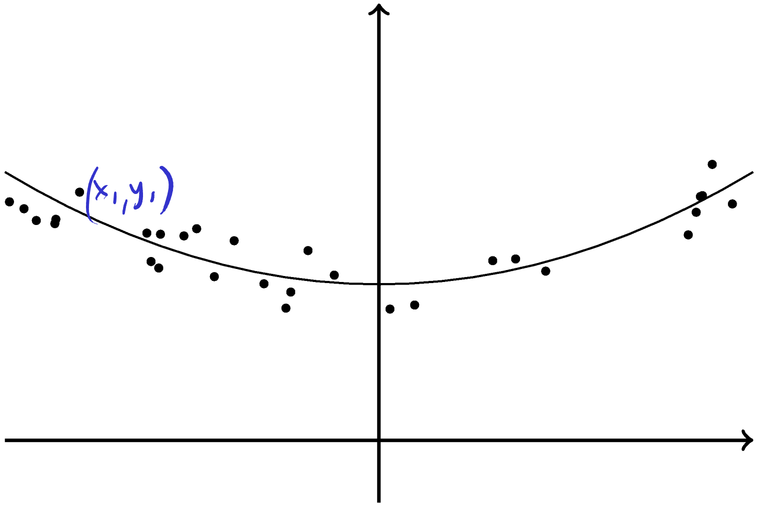
What about non-square systems?

Specifically, what about linear systems with 'tall and skinny' matrices? (A: $m \times n$ with $m > n$) (aka overdetermined linear systems)

- Specifically, any hope that we will solve those exactly?

$$\begin{pmatrix} \\ \\ A \\ \\ \\ \end{pmatrix} x \approx \begin{pmatrix} \\ b \\ \\ \end{pmatrix}$$

Example: Data Fitting



Have data: (x_i, y_i) and model:

$$y(x) = \overset{\alpha}{a} + \overset{\beta}{b}x + \overset{\gamma}{c}x^2$$

- Find data that best fit model:

$$\alpha + \beta x_1 + \gamma x_1^2 = y_1$$

$$\alpha + \beta x_2 + \gamma x_2^2 = y_2$$

$$|\alpha + \beta x_1 + \gamma x_1^2 - y_1|^2$$

$$+ |\alpha + \beta x_2 + \gamma x_2^2 - y_2|^2$$

$$\vdots$$

$$+ |\alpha + \beta x_n + \gamma x_n^2 - y_n|^2 \longrightarrow \min_{\alpha, \beta, \gamma}!$$

$$\|A\vec{a} - \vec{y}\|_2^2 \rightarrow \min_{\vec{a} = (\alpha, \beta, \gamma)}!$$

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & \vdots \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

Vandermonde matrices

$$f_1(x) = \alpha + \beta x + \gamma x^2$$ ← "linear"

$$f_2(x) = \underbrace{\exp(\alpha)}_{\tilde{\alpha}} + \beta x + \gamma x^2$$ ← "nonlinear in 1 coeff."

$$f_3(x) = \alpha + \exp(\beta x + \gamma x^2)$$ ← genuinely nonlinear

can be rewritten as $\|A\vec{a} - b\|_2^2$

cannot be rewritten as $\|A\vec{a} - b\|_2^2$

Demo: Interactive Polynomial Fitting