CSC338 Lecture 3

Today, we'll start talking about a new class of computational problems.

Solving a system of linear equations. $A\underline{x} = \underline{b}$.

A: $n \times n$, $\underline{x}$: $n \times 1$, $\underline{b}$: $n \times 1$

We wish to solve for $\underline{x}$ given A and $\underline{b}$

- Develop an efficient algorithm (today)
- Develop an efficient and stable algorithm
- Determine when the problem is well-conditioned

} next week.

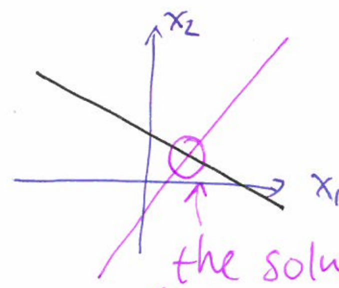
If $A \in \mathbb{R}^{m \times n}$ rectangular and

- $m > n \Rightarrow$ over-determined system — more equations than unknowns
- $m < n \Rightarrow$ under-determined system — fewer equations than unknowns

eg// $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ $\quad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ $\quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ unknown.

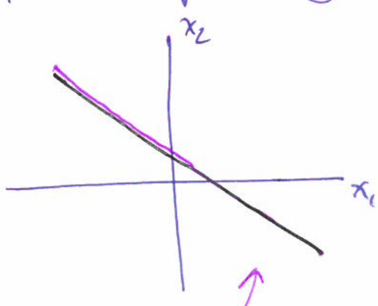
then $\underline{x}$ represents the intersection of 2 lines.

$$A\underline{x} = \underline{b} \iff \begin{array}{l} a_{11}x_1 + a_{12} \cdot x_2 = b_1 \\ a_{21}x_1 + a_{22} \cdot x_2 = b_2 \end{array}$$

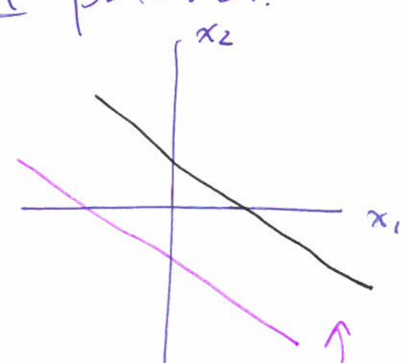


the solution we're looking for!

Two lines intersect at a unique point if they are not parallel.



intersect everywhere



intersect nowhere

Def The matrix A is non-singular iff there exists a matrix A^{-1} such that $AA^{-1} = A^{-1}A = I = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots & \\ & & & 1 \end{bmatrix}$

identity matrix

(i.e. A is invertible)

Equivalent Def:
- $\det(A) \neq 0$
- $\text{rank}(A) = n$ $\quad A \in \mathbb{R}^{n\times n}$
- for any vector $\underline{v} \in \mathbb{R}^n$, $\underline{v} \neq \underline{0}$, $A\underline{v} \neq \underline{0}$

eg// $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ is singular.

$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \underline{x} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$ has ∞ solutions. $\qquad \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \underline{x} = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$ has no solution

If A is nonsingular, then, $A\underline{x} = \underline{b}$ has a unique solution. the system.

Ⓠ How do we solve $A\underline{x} = \underline{b}$ $\quad A \in \mathbb{R}^{n\times n}$ nonsingular?

$\Rightarrow$ Why not compute A^{-1}, then set $\underline{x} = A^{-1}\underline{b}$?

↳ But how do we compute A^{-1}? (Assume n is large)

We need solve $AY = I$ which means solving

$Y = \begin{bmatrix} | & | & & | \\ y_1 & y_2 & \cdots & y_n \\ | & | & & | \end{bmatrix}$ $\qquad A\underline{y_i} = \underline{e_i} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow$ i^{th} row $\quad$ i^{th} column of the identity matrix.

for each $i \in \{1, 2, \ldots, n\}$

we're back to solving systems of linear equations!

To answer Ⓠ, start by thinking of situations where $A\underline{x} = \underline{b}$ is easy to solve. Then maybe convert

general problem $\Rightarrow$ easy to solve problem

When is $A\underline{x} = \underline{b}$ easy to solve?

$\Rightarrow$ when A is <u>diagonal</u> $a_{ij} = 0$ if $i \neq j$

eg// $A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix}$ $\underline{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ $\underline{x} = \begin{bmatrix} 1/3 \\ 2/4 \\ 3/5 \end{bmatrix}$.

In general, $a_{ii} x_i = b_i \Rightarrow \boxed{x_i = \dfrac{b_i}{a_{ii}}}$ n×n system require n flops to solve

$\Rightarrow$ when A is non-singular <u>lower triangular</u> $\begin{cases} a_{ij} = 0 \text{ if } i < j \\ a_{ii} \neq 0 \end{cases}$

eg// $A = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 4 & 0 \\ 2 & 2 & 5 \end{bmatrix}$ $\underline{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

$3x_1 = 1 \Rightarrow x_1 = \frac{1}{3}$

$2(\frac{1}{3}) + 4x_2 = 2 \Rightarrow x_2 = \frac{1}{3}$

$2(\frac{1}{3}) + 2(\frac{1}{3}) + 5x_3 = 3 \Rightarrow x_3 = \frac{1}{3}$

$\underline{x} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$.

In general, use <u>forward substitution</u> algorithm for a lower triangular matrix A.

$$x_1 = \frac{b_1}{a_{11}}$$

$$x_2 = \frac{b_2 - x_1 a_{21}}{a_{22}}$$

$$\vdots$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j}{a_{ii}}$$

$\Rightarrow$ when A is nonsingular <u>upper triangular</u>.

Use algorithm called <u>backward substitution</u>

eg// $A = \begin{bmatrix} 5 & 2 & 2 \\ 0 & 4 & 2 \\ 0 & 0 & 3 \end{bmatrix}$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}$$

Idea: transform a system $A\underline{x} = \underline{b}$ into an upper-triangular system with the same solution. ...using elementary row operations.

$\Rightarrow$ Gauss Elimination

Example system

$$
\begin{array}{c}
R_1\\ R_2\\ R_3\\ R_4
\end{array}
\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{A}
\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\underline{x}}
=
\underbrace{\begin{bmatrix} 1 \\ 1 \\ -1 \\ 3 \end{bmatrix}}_{\underline{b}}
$$

$$
\begin{array}{r}
-({}^{2\times}2x_1 + {}^{2\times}x_2 + {}^{2\times}x_3 = {}^{2\times}1) \\
4x_1 + 3x_2 + 3x_3 + x_4 = 1 \\
\hline
x_2 + x_3 + x_4 = -1
\end{array}
$$

Step 1 Use elementary row operations to zero out values below a_{11}

$$
\begin{array}{l}
\\ R_2 \leftarrow R_2 - 2\cdot R_1 \\ R_3 \leftarrow R_3 - 4\cdot R_1 \\ R_4 \leftarrow R_4 - 3\cdot R_1
\end{array}
\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{bmatrix}}_{A'}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\underbrace{\begin{bmatrix} 1 \\ -1 \\ -5 \\ 0 \end{bmatrix}}_{b'}
$$

$$A' = M_1 \cdot A$$
$$\underline{b'} = M_1 \underline{b}$$

$$M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix}$$

Step 2 zero out below the next diagonal.

$$
\begin{array}{l}
\\ \\ R_3 \leftarrow R_3 - 3R_2 \\ R_4 \leftarrow R_4 - 4R_2
\end{array}
\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}}_{A''}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\underbrace{\begin{bmatrix} 1 \\ -1 \\ -2 \\ 4 \end{bmatrix}}_{b''}
$$

$$A'' = M_2 \cdot M_1 \cdot A$$
$$\underline{b''} = M_2 \cdot M_1 \underline{b}$$

$$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & -4 & 0 & 1 \end{bmatrix}$$

Step 3

$$
\begin{array}{l}
\\ \\ \\ R_4 \leftarrow R_4 - \frac{2}{2} R_3
\end{array}
\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{A'''}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\underbrace{\begin{bmatrix} 1 \\ -1 \\ -2 \\ 6 \end{bmatrix}}_{b'''}
$$

$$A''' = M_3 M_2 M_1 A$$
$$\underline{b'''} = M_3 M_2 M_1 \underline{b}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$\Rightarrow$ use backward substitution to solve the system

$$(M_3 M_2 M_1 A)\,\underline{x} = (M_3 M_2 M_1 \underline{b})$$

(resume 4:10pm)

The matrices M_1, M_2, M_3 are called <u>elementary elimination matrices</u>.

<u>Gauss Elimination</u> Algorithm Pseudocode

loop over columns $k = \{1, 2, \ldots, (n-1)\}$

  # eliminate below a_{kk}.

  loop over rows $i = \{k+1, k+2, \ldots, n\}$

    $m = \dfrac{a_{ik}}{a_{kk}}$ # the multiplier

    # $R_i \leftarrow R_i - m R_k$.

    loop over $j = \{i+1, i+2, \ldots, n\}$

      $a_{ij} = a_{ij} - m \cdot a_{ik}$.

(we'll write actual python code in tutorial)

<u>Properties of Elementary Elimination Matrices</u>

1. M_k is lower triangular, have unit diagonals, and is non-singular.

2. Can write $M_k = I - \underline{m}\,\underline{e}_k^T$ for some

$$m = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1} \\ m_{k+2} \\ \vdots \\ m_n \end{bmatrix} \qquad e_k = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\leftarrow$ k^{th} column of the identity matrix.

3. $M_k^{-1} = I + \underline{m}\,\underline{e}_k^T$ (homework 3)

4. If we have $M_k = I - \underline{m}\,\underline{e}_k^T$, $M_j = I - \underline{t}\,\underline{e}_j^T$ then

$$M_k M_j = (I - \underline{m}\,\underline{e}_k^T)(I - \underline{t}\,\underline{e}_j^T)$$

$$= \underbrace{I - \underline{m}\,\underline{e}_k^T - \underline{t}\,\underline{e}_j^T}_{} + \underbrace{\underline{m}\,\underline{e}_k^T\,\underline{t}\,\underline{e}_j^T}_{=0}$$

for k > j ~~because k≠j~~

$\Rightarrow$ Products are "unions"

$\Rightarrow M_k M_j$ is lower triangular.

$\Rightarrow \prod_{\ell} M_\ell$ is also lower triangular.

Combining these properties:

$$\underbrace{(M_3 M_2 M_1)}_{\text{lower triangular}} A = \underbrace{A'''}_{\text{upper triangular}}$$

$$\Rightarrow \quad A = \underbrace{(M_3 M_2 M_1)^{-1}}_{\text{lower triangular}} \underbrace{A'''}_{\text{upper triangular}}$$

$$= L \quad U.$$

LU Factorization

If we keep track of the elementary elimination matrices while running Gauss Elimination, we can factor $A = LU$ into an upper triangular and a lower-triangular component.

If we have a factorization $A = LU$, then we can solve $A\underline{x} = \underline{b}$ by solving $LU\underline{x} = \underline{b}$

① Solve $L\underline{y} = \underline{b}$ using forward substitution

② Solve $U\underline{x} = \underline{y}$ using backward substitution.

LU factorization is useful when we want to solve $A\underline{x} = \underline{b}$ for many $\underline{b}$'s. (e.g// inverting a matrix)