02.01 systems of equations: direct

why stop at one?

1 gaussian elimination

naive gaussian elimination

example 01

consider system of equations

$$\begin{aligned} 2x - 2y - z &= -2 \\ 4x + y - 2z &= 1 \\ -2x + y - z &= -3. \end{aligned}$$

three equations and three unknowns $\Rightarrow$ mosh those equations to solve for z; use z to solve for y; use z, y to solve for x.

in an orderly, pre-algorithm kind of way,

$$Ax = b \Rightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{\text{more writing}} \sim \overbrace{\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & | & b_1 \\ a_{21} & a_{22} & a_{23} & | & b_2 \\ a_{31} & a_{32} & a_{33} & | & b_3 \end{bmatrix}}_{\text{less writing} = \text{yay}}}^{\text{tableau}}$$

where a_{ij}, b_i represents elements at row i and column j.

to "solve for z then y then x", get the structure to look like this,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & | & b_1 \\ 0 & a'_{22} & a'_{23} & | & b'_2 \\ 0 & 0 & a''_{33} & | & b''_3 \end{bmatrix}$$

where prime $'$ and double-prime $''$ represents element after modification by **forward elimination**.

for the given system of equations,

$$\begin{bmatrix} 2 & -2 & -1 \\ 4 & 1 & -2 \\ -2 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 4 & 1 & -2 & | & 1 \\ -2 & 1 & -1 & | & -1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & ? & ? & | & ? \\ 0 & 0 & ? & | & ? \end{bmatrix}$$

$$\begin{aligned} &1.\ \text{subtract row 1 from row 2} && \Rightarrow \text{new [row 2]} = \text{[row 2]} - m \times \text{[row 1], where } m = \frac{a_{21}}{a_{11}} = \frac{4}{2} = 2 \\ & && \Rightarrow \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 4-2(2) & 1-2(-2) & -2-2(-1) & | & 1-2(-2) \\ -2 & 1 & -1 & | & -1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ -2 & 1 & -1 & | & -1 \end{bmatrix} \\ &2.\ \text{subtract row 1 from row 3} && \Rightarrow \text{new [row 3]} = \text{[row 3]} - m \times \text{[row 1], where } m = \frac{a_{31}}{a_{11}} = \frac{-2}{2} = -1 \\ & && \Rightarrow \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ -2+1(2) & 1+1(-2) & -1+1(-1) & | & -1+1(-2) \end{bmatrix} = \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ 0 & -1 & -2 & | & -3 \end{bmatrix} \\ &3.\ \text{subtract row 2 from row 3} && \Rightarrow \text{new [row 3]} = \text{[row 3]} - m \times \text{[row 2], where } m = \frac{a_{32}}{a_{22}} = \frac{-1}{5} = -\frac{1}{5} \\ & && \Rightarrow \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ 0 & -1+\frac{1}{5}(5) & -2+\frac{1}{5}(0) & | & -3+\frac{1}{5}(5) \end{bmatrix} = \begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ 0 & 0 & -2 & | & -2 \end{bmatrix} \end{aligned}$$

huzzah, upper triangular matrix. ✓

forward elimination

consider relevant operations,

1. swap one equation for another;
2. add or subtract multiple of one equation from another; and
3. multiply an equation by a nonzero constant.

to zero below the diagonal, consider that the columns zeroed are $j = 1$ to $n - 1$, where $n = 3$ number of equations, number of unknowns, dimension of $A_{n \times n}, b_n$.

```
for j = 1 : n-1
  # eliminate column j
end
```

for each j, the rows to zero below the diagonal are $i = j + 1$ to n.

```
for j = 1 : n-1
  # eliminate column j
  for i = j+1 : n
    # eliminate entry a(i,j)
end
```

and whats applied to one element in a row (eg, a_{ij}) is applied to all elements in that row -- including b_i.

algorithm, forward elimination

```
for j = 1 : n-1
  if abs(a(j,j))<eps; error('zero pivot encountered'); end
  for i = j+1 : n
    mult = a(i,j)/a(j,j);
    for k = j+1:n
      a(i,k) = a(i,k) - mult*a(j,k);
    end
    b(i) = b(i) - mult*b(j);
  end
end
```

oc this requires that the diagonal not be zero. ie, the diagonal contains the **pivots** of coefficient matrix A.

backward substitution

example 01, continued

$$\begin{bmatrix} 2 & -2 & -1 & | & -2 \\ 0 & 5 & 0 & | & 5 \\ 0 & 0 & -2 & | & -2 \end{bmatrix} \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & | & b_1 \\ 0 & a'_{22} & a'_{23} & | & b'_2 \\ 0 & 0 & a''_{33} & | & b''_3 \end{bmatrix} \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix}$$

to solve for $[x, y, z]^T$,

$$\begin{aligned} 1.\quad & z = \frac{b''_3}{a''_{33}} \\ 2.\quad & y = \frac{b'_2 - a'_{23} \cdot z}{a'_{22}} \\ 3.\quad & x = \frac{b'_1 - a'_{13} \cdot z - a'_{12} \cdot y}{a'_{11}} \end{aligned}$$

$$\Downarrow$$

$$[x, y, z]^T = [\frac{1}{2}, 1, 1]^T$$

no surprises there. more generally, for any n...

algorithm, backward substitution

```
for i = n : -1 : 1
  for j = i+1 : n
    b(i) = b(i) - a(i,j)*x(j);
  end
  x(i) = b(i)/a(i,i);
end
```

operation counts

consider matrix of coefficients A wrt forward elimination. cost in operations stacks like this:

$$\begin{bmatrix} 0 & & & & & & \\ 2n+1 & 0 & & & & & \\ 2n+1 & 2(n-1)+1 & 0 & & & & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & 0 & & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & \dots & 2(3)+1 & 0 & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & \dots & 2(3)+1 & 2(2)+1 & 0 \end{bmatrix}$$

where n is dimension of $A_{n\times n}, b_n$ and each cell contains the cost in operations to zero its value.

eg, for $a'_{21} = 0 = a_{21} - a_{11} \cdot \frac{a_{21}}{a_{11}} \sim 3$ operations but also consider every cell in that same row that gets refactored by the same m. so $2n+1$ where "2" is the subtraction and multiplication, "n" is the number of cells in that row that need attention (includes b_i) and "$+1$" is the cost of the single division required to evaluate m for that row.

remember, for any positive integer n,

1. $1+2+3+\cdots+n = n(n+1)/2$ and
2. $1^2+2^2+3^2+\cdots+n^2 = n(n+1)(2n+1)/6$.

in case of homework...

to sum,

$$\begin{aligned} \sum_{j=1}^{n-1}\sum_{i=1}^{j} 2(j+1)+1 &= \sum_{j=1}^{n-1} 2j(j+1)+j \\ &= 2\sum_{j=1}^{n-1} j^2 + 3\sum_{j=1}^{n-1} j = 2\frac{(n-1)n(2n-1)}{6} + 3\frac{(n-1)n}{2} \\ &= (n-1)n\left[\frac{2n-1}{3}+\frac{3}{2}\right] = \frac{n(n-1)(4n+7)}{6} \\ &= \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n. \end{aligned}$$

the cost of back substitution is $n^2 \sim \mathcal{O}(n^2)$ vs $\mathcal{O}(n^3)$ for forward elimination.

2 LU factorization

matrix form of guassian elimination

LU factorization splits coefficient matrix A into **lower triangular** and **upper triangular** matrices such that

$$[L][U] = [A]$$

where entries of $L_{m\times n}$ satisfy $l_{ij} = 0$ for $i < j$ and entries of $U_{m\times n}$ statisy $u_{ij} = 0$ for $i > j$.

example 04

$$LU = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} = A.$$

example 05

consider matrix of coefficients,

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix}$$

apply forward elimination on A to get upper triangular U; however, store the m used in each step in L. ie,

$$\begin{aligned}
&0.\ \text{start with } L = I, U = A \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} \\
&1.\ \text{subtract row 1 from row 2} \quad \Rightarrow \text{new [row 2]} = \text{[row 2]} - m \times \text{[row 1], where } m = \frac{a_{21}}{a_{11}} = \frac{2}{1} = 2 \\
&\qquad\qquad\qquad\qquad\qquad\quad \Rightarrow l_{ij} = m \text{ where } i = \text{ [row to zero]} = 2, j = \text{[row to zero with]} = 1 \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ -3 & 1 & 1 \end{bmatrix} \\
&2.\ \text{subtract row 1 from row 3} \quad \Rightarrow \text{new [row 3]} = \text{[row 3]} - m \times \text{[row 1], where } m = \frac{a_{31}}{a_{11}} = \frac{-3}{1} = -3 \\
&\qquad\qquad\qquad\qquad\qquad\quad \Rightarrow l_{ij} = m \text{ where } i = \text{ [row to zero]} = 3, j = \text{[row to zero with]} = 1 \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 7 & -2 \end{bmatrix} \\
&3.\ \text{subtract row 2 from row 3} \quad \Rightarrow \text{new [row 3]} = \text{[row 3]} - m \times \text{[row 2], where } m = \frac{a_{32}}{a_{22}} = \frac{7}{-3} = -\frac{7}{3} \\
&\qquad\qquad\qquad\qquad\qquad\quad \Rightarrow l_{ij} = m \text{ where } i = \text{ [row to zero]} = 3, j = \text{[row to zero with]} = 2 \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = LU. \\
&\quad \Rightarrow LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} = A. \checkmark
\end{aligned}$$

back substitution with LU factorization

$Ax = b \Rightarrow LUx = b$. let $c = Ux$. then back substitution becomes

a) solve $Lc = b$ for c;
b) solve $Ux = c$ for x.

complexity of LU factorization

why do this? bc classical guassian elimination includes vector b_n in forward elimination, the more expensive part of the process. if n is large, that cost is significant.

3 error

beware poorly conditioned coefficient matrices; beware swamping.

error magnification and condition number

definition 03 infinity norm of $\vec{x}$, $||x||_\infty = \max|x_i|, i = 1, \ldots, n$.

definition 04 let x_a be approximate solution of linear system $Ax = b$. then **residual** $\vec{r} = b - Ax_a$. **backward error** $= ||b - Ax_a||_\infty$ and **forward error** $= ||x - x_a||_\infty$ $\sim$ bc this is root-finding.

example 11

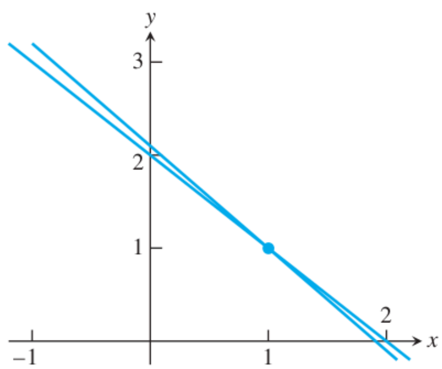
calculate forward and backwards errors for approximate solution $x_a = [-1, 3.0001]^T$ to system

$$\begin{aligned} x_1 + x_2 &= 2 \\ 1.0001x_1 + x_2 &= 2.0001. \end{aligned}$$

by gauss elimination, $x^* = [1, 1]^T$. so

$$b - Ax_x = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix} = \begin{bmatrix} -0.0001 \\ 0.0001 \end{bmatrix} \Rightarrow ||b - Ax_a||_\infty = 0.0001$$

$$x - x_a = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ -3.0001 \end{bmatrix} = \begin{bmatrix} 2 \\ -2.0001 \end{bmatrix} \Rightarrow ||x - x_a||_\infty = 2.0001$$



lol

[ ] ↳ 2 cells hidden

usw

that sort of thing happens when the approximation and the solution exist on nearly parallel lines.

also, the relative errors,

$$\text{relative backward} = \frac{||r||_\infty}{||b||_\infty}, \text{relative forward} = \frac{||x - x_a||_\infty}{||x||_\infty}.$$

error magnification $= \frac{\text{relative forward}}{\text{relative backward}} = \frac{\frac{||x-x_a||_\infty}{||x||_\infty}}{\frac{||r||_\infty}{||b||_\infty}}$.

for example 11, error magnifcation $= \frac{\frac{2.0001}{1}}{\frac{0.0001}{2.0001}} \approx \frac{200\%}{0.005\%} = 40004.0001 =$ LOL.

definition 05 condition number of square matrix A, **cond**(A) is maximum possible error magnification factor for solving $Ax = b$ over all RHS b.

matrix norm of $A_{n \times n}$: $||A||_\infty =$ maximum absolute row sum. ie, total absolute values of each row, assign maximum of these n numbers to be norm of A.

theorem 06 condition number of $n \times n$ matrix A is

$$\text{cond}(A) = ||A|| \cdot ||A^{-1}||.$$

for example 11,

$$\begin{aligned} ||A|| &= 2.0001 \\ A^{-1} &= \begin{bmatrix} -10000 & 10000 \\ 10001 & -10000 \end{bmatrix} \\ &\Downarrow \\ ||A^{-1}|| &= 20001 \\ &\Downarrow \\ \text{cond}(A) &= (2.0001)(20001) = 40004.0001 = \text{error magnification}. \end{aligned}$$

swamping

example 13

consider system of equations

$$\begin{aligned} 10^{-20}x_1 + x_2 &= 1 \\ x_1 + 2x_2 &= 4. \end{aligned}$$

a) human flywheel

$$\begin{bmatrix} 10^{-20} & 1 & | & 1 \\ 1 & 2 & | & 4 \end{bmatrix} \xrightarrow{\text{subtract } 10^{20} \times \text{row 1 from row 2}} \begin{bmatrix} 10^{-20} & 1 & | & 1 \\ 0 & 2-10^{20} & | & 4-10^{20} \end{bmatrix}$$

$$\Downarrow$$

$$\left.\begin{aligned} (2-10^{20})x_2 = 4-10^{20} &\quad \Rightarrow x_2 = \tfrac{4-10^{20}}{2-10^{20}} \\ 10^{-20}x_1 + \tfrac{4-10^{20}}{2-10^{20}} = 1 &\quad \Rightarrow x_1 = 10^{20}\left(1 - \tfrac{4-10^{20}}{2-10^{20}}\right) = \tfrac{-2\times 10^20}{2-10^{20}} \end{aligned}\right\} \quad \Rightarrow \quad [x_1, x_2] \approx [2, 1].$$

b) breaking the machine

with IEEE double precision,

$$\begin{bmatrix} 10^{-20} & 1 & | & 1 \\ 1 & 2 & | & 4 \end{bmatrix} \to \begin{bmatrix} 10^{-20} & 1 & | & 1 \\ 0 & 2-10^{20} & | & 4-10^{20} \end{bmatrix} \xrightarrow{\text{2,4 vs } 10^{20}\text{, lost to rounding}} \begin{bmatrix} 10^{-20} & 1 & | & 1 \\ 0 & -10^{20} & | & -10^{20} \end{bmatrix}$$

$$\Downarrow$$

$$\left.\begin{aligned} -10^{20}x_2 = -10^{20} &\quad \Rightarrow x_2 = 1 \\ 10^{-20}x_1 + 1 = 1 &\quad \Rightarrow x_1 = 0 \end{aligned}\right\} \quad \Rightarrow \quad [x_1, x_2] = [0, 1].$$

c) compensating

with IEEE double precision and row exchange,

$$\begin{bmatrix} 1 & 2 & | & 4 \\ 10^{-20} & 1 & | & 1 \end{bmatrix} \xrightarrow{\text{subtract } 10^{-20} \times \text{ row 1 from row 2}} \begin{bmatrix} 1 & 2 & | & 4 \\ 0 & 1-2\times 10^{-20} & | & 1-4\times 10^{-20} \end{bmatrix} \xrightarrow{\text{rounding}} \begin{bmatrix} 1 & 2 & | & 4 \\ 0 & 1 & | & 1 \end{bmatrix}$$

$$\Downarrow$$

$$\left.\begin{aligned} x_1 + 2x_2 = 4 &\quad \Rightarrow x_1 = 2 \\ x_2 = 1 &\quad \Rightarrow x_2 = 1 \end{aligned}\right\} \quad \Rightarrow \quad [x_1, x_2] = [2, 1] \text{ to about 16 significant digits.}$$

ie, for gauss elimination, multipliers should be kept as small to avoid swamping.

4 PA=LU factorization

gauss so far has been "naive" bc it doesnt do much about zeros on the diagonal and theres that swamping thing. moving stuff around helps.

partial pivoting

partial pivoting refers to comparing numbers before each elimination step and swapping the row with the larger first entry towards the top.

ie, select pth row, where

$$|a_{p1}| \geq |a_{i1}|$$

for all $1 \leq i \leq n$ and exchange rows 1 and p. then use the new a_{11} to eliminate column 1 as usual. ie,

$$m_{i1} = \frac{a_{i1}}{a_{11}} \text{ and } |m_{i1}| \leq 1.$$

apply same to every pivot during the algorithm. when deciding the second pivot, a_{22} will be compared with entries lower than it,

$$|a_{p2}| \geq |a_{i2}|$$

for all $2 \leq i \leq n$ and exchange rows 2 and p. then use the new a_{22} to eliminate column 2 as usual.

und so weiter.

example 15

apply gaussian elimination with partial pivoting to system

$$\begin{aligned} x_1 - x_2 + 3x_3 &= -3 \\ -x_1 - 2x_3 &= 1 \\ 2x_1 + 2x_2 + 4x_3 &= 0. \end{aligned}$$

$$\left[\begin{array}{ccc|c} 1 & -1 & 3 & -3 \\ -1 & 0 & -2 & 1 \\ 2 & 2 & 4 & 0 \end{array}\right] \xrightarrow{\text{swap row 1 with row 3}} \left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ -1 & 0 & -2 & 1 \\ 1 & -1 & 3 & -3 \end{array}\right]$$

$$\xrightarrow{\text{subtract } -\frac{1}{2}\times\text{row 1 with row 2}} \left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & -1 & 3 & -3 \end{array}\right]$$

$$\xrightarrow{\text{subtract } \frac{1}{2}\times\text{row 1 with row 3}} \left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & 1 & -3 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & 1 & -3 \end{array}\right] \xrightarrow{\text{swap row 2 with row 3}} \left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ 0 & -2 & 1 & -3 \\ 0 & 1 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{\text{subtract } -\frac{1}{2}\times\text{row 2 with row 3}} \left[\begin{array}{ccc|c} 2 & 2 & 4 & 0 \\ 0 & -2 & 1 & -3 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{array}\right]$$

note: all three multipliers are less than 1 in absolute value.

$$\left.\begin{aligned} \frac{1}{2}x_3 &= -\frac{1}{2} \\ -2x_2 + x_3 &= -3 \\ 2x_1 + 2x_2 + 4x_3 &= 0. \end{aligned}\right\} \Rightarrow x = [1, 1, -1]^T.$$

permuation matrices

permutation matrix is $n \times n$ matrix of all zeros except for a single 1 in every row and column.

theorem 08 fundamental theorem of permutation matrices

let P be $n \times n$ permutation matrix formed by a particular set of row exchanges applied to the identity matrix. then for any $n \times n$ matrix A, PA is the matrix obtained by applied exactly the same set of row exchanges to A.

usw

eg, permutation matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

is formed by exchanging rows 2 and 3 of the identity matrix. applying it to another matrix is the same as applying the same row swaps,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ g & h & i \\ d & e & f \end{bmatrix}.$$

PA=LU factorization

the sum of gaussian elimination, hooray.

example 16

find PA=LU factorization of matrix

$$A = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \qquad \Rightarrow PA = LU$$

$$IA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \xrightarrow{\text{swap row 1 with row 2}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -4 \\ 2 & 1 & 5 \\ 1 & 3 & 1 \end{bmatrix} \sim P_1A = A_1$$

$$\xrightarrow{\text{mod row 2 with row 1}} \begin{bmatrix} 4 & 4 & -4 \\ \frac{1}{2} & -1 & 7 \\ 1 & 3 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 1 & 3 & 1 \end{bmatrix} \sim L_1U_1$$

$$\xrightarrow{\text{mod row 3 with row 1}} \begin{bmatrix} 4 & 4 & -4 \\ \frac{1}{2} & -1 & 7 \\ \frac{1}{4} & 2 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 0 & 2 & 2 \end{bmatrix} \sim L_2U_2$$

$$P_1A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \xrightarrow{\text{swap row 2 with row 3}} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -4 \\ 1 & 3 & 1 \\ 2 & 1 & 5 \end{bmatrix} \sim P_2A = PA = A_2$$

$$L_2U_2 = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & -4 \\ 0 & -1 & 7 \\ 0 & 2 & 2 \end{bmatrix} \sim \begin{bmatrix} 4 & 4 & -4 \\ \frac{1}{2} & -1 & 7 \\ \frac{1}{4} & 2 & 2 \end{bmatrix} \xrightarrow{\text{swap row 2 with row 3}} \begin{bmatrix} 4 & 4 & -4 \\ \frac{1}{4} & 2 & 2 \\ \frac{1}{2} & -1 & 7 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & -1 & 7 \end{bmatrix} \sim L_3U_3$$

$$\xrightarrow{\text{mod row 3 with row 2}} \begin{bmatrix} 4 & 4 & -4 \\ \frac{1}{4} & 2 & 2 \\ \frac{1}{2} & -\frac{1}{2} & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix} \sim L_4U_4 = LU$$

usw

the LU factorization for PA=LU now becomes,

$$\left.\begin{aligned} PAx &= Pb \\ LUx &= Pb \end{aligned}\right\} \xrightarrow{solve} \begin{aligned} &1. \quad Lc = Pb \text{ for } c \\ &2. \quad Ux = c \text{ for } x. \end{aligned}$$

example 17

use PA=LU factorization to solve system $Ax = b$, where

$$A = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix}.$$

1. solve $Lc = Pb$,

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 5 \end{bmatrix}$$

starting at the top,

$$\begin{aligned} c_1 &= 0 \\ \tfrac{1}{4}(0) + c_2 &= 6 \Rightarrow c_2 = 6 \\ \tfrac{1}{2}(0) - \tfrac{1}{2}(6) + c_3 &= 5 \Rightarrow c_3 = 8 \end{aligned} \quad \Rightarrow \quad c = [0, 6, 8]^T.$$

2. solve $Ux = c$,

$$\begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 8 \end{bmatrix}$$

starting at the bottom,

$$\begin{aligned} 8x_3 &= 8 \Rightarrow x_3 = 1 \\ 2x_2 + 2(1) &= 6 \Rightarrow x_2 = 2 \\ 4x_1 + 4(2) - 4(1) &= 0 \Rightarrow x_1 = -1 \end{aligned} \quad \Rightarrow \quad x = [-1, 2, 1]^T.$$

resources

- *big O refresher*: @wiki @brightside (4 minutes!)