Lecture 6: Row echelon and Gaussian elimination.

[§RREF of B]

Last time:

$$\left\{\begin{array}{c}\text{Linear sys}\\ \text{with } n \text{ vars}\\ \text{and } m \text{ eqns}\end{array}\right\} \underset{\mathcal{LS}}{\overset{\text{Augmented}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\text{matrix}}}}} \mathrm{Mat}_{m\times(n+1)}(\mathbb{R})$$

$$\begin{aligned} 3x_1 + 2x_2 &= 1 \\ -2x_1 - x_2 &= 0 \end{aligned} \longleftrightarrow \begin{pmatrix} 3 & 2 & 1 \\ -2 & -1 & 0 \end{pmatrix}$$

Row ops: ① $R_i \longleftrightarrow R_j$ ② Scale row by $c \neq 0$.
③ Replace R_j with $cR_i + R_j$.

Thm: If M and N are row equivalent, then $\mathcal{LS}(M)$ and $\mathcal{LS}(N)$ have the same sol'ns.

Reduced Row Echelon Form: A matrix where

① Rows of zeros are at the bottom

② The first nonzero entry in any row is a 1 (leading 1).

③ A leading 1 is the only nonzero entry in its column.

④ Suppose (i,j) and (s,t) are indices of leading 1's. If $i > s$ then $j > t$.

Ex:

$$\begin{pmatrix} \boxed{1} & 2 & 0 & 1 & 0 & 2 \\ 0 & 0 & \boxed{1} & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & \boxed{1} & 4 \end{pmatrix} = M$$

$(s,t) = (2,3)$ $\quad$ $(i,j) = (3,5)$

Non ex:

$$\begin{pmatrix} 2 & 2 & 3 \\ 0 & 0 & 0 \\ 0 & \boxed{1} & 4 \\ \boxed{1} & 0 & 5 \end{pmatrix}$$

This violates all 4 conditions!

Why useful: Can easily solve the linear system for the vars assoc to the leading 1's.

Ex: $\mathcal{LS}(M) =$

$$\begin{aligned} x_1 + 2x_2 + x_4 &= 2 \\ x_3 + 3x_4 &= 3 \\ x_5 &= 4 \end{aligned}$$

So

$$\left.\begin{aligned} x_1 &= 2 - 2x_2 - x_4 \\ x_3 &= 3 - 3x_4 \\ x_5 &= 4 \end{aligned}\right\}$$

Key: No x_1, x_3, or x_5 over here, because of 3.

Thus all sol'n are

"where" "in"

$$\{(2-2s-t,\ s,\ 3-3t,\ t,\ 4) \mid s,t \in \mathbb{R}\}$$

Thm: Any matrix M is row equivalent to one in reduced row echelon form.

Gauss-Jordan Elimination: M an $m \times n$ matrix.

Denote rows by R_i and columns by C_j.

0. Set $r=0$, $j=0$.
1. If $j \geq n$ stop and return current matrix. Otherwise, set $j=j+1$.
2. If C_j is 0 below row r, go to 1.
3. Set $r=r+1$. Arrange that j^{th} entry of R_r is nonzero by swapping R_r with one $R_{r+1}, \ldots, R_m$ if necessary.
4. Scale R_r so that j^{th} entry is 1.
5. For i in $[1, \ldots, m]$, $i \neq r$ clear the j^{th} entry of R_i by setting $R_i = -cR_r + R_i$ where c is the j^{th} entry of R_i.
6. Go to 1.

Ex: $\overset{j=1,\ r=1}{\begin{pmatrix} 0 & 4 & 6 & 8 \\ 2 & 0 & -2 & 4 \\ -3 & 0 & 3 & 5 \end{pmatrix}} \xrightarrow{R_1 \leftrightarrow R_1} \begin{pmatrix} 2 & 0 & -2 & 4 \\ 0 & 4 & 6 & 8 \\ -3 & 0 & 3 & 5 \end{pmatrix}$

$\xrightarrow{\frac{1}{2}R_1} \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 4 & 6 & 8 \\ -3 & 0 & 3 & 5 \end{pmatrix} \xrightarrow{3R_1+R_3} \overset{j=2,\ r=2}{\begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 4 & 6 & 8 \\ 0 & 0 & 0 & 11 \end{pmatrix}}$

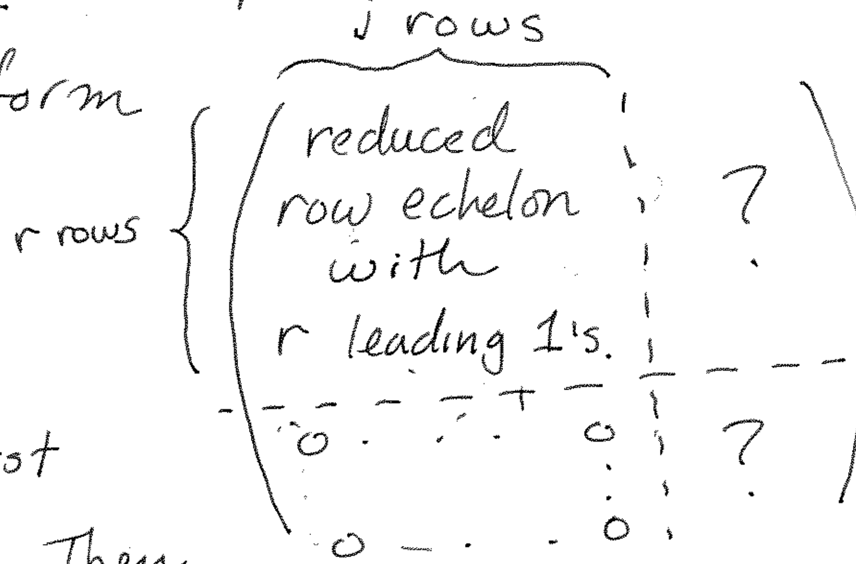
$\xrightarrow{\frac{1}{4}R_2} \overset{j=3,\ r=2}{\begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & 3/2 & 2 \\ 0 & 0 & 0 & 11 \end{pmatrix}} \longrightarrow \overset{j=4,\ r=3}{\begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & 3/2 & 2 \\ 0 & 0 & 0 & 11 \end{pmatrix}}$

$\xrightarrow{\frac{1}{11}R_3} \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & 3/2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{-2R_3+R_1} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 3/2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

$\xrightarrow{-2R_3+R_2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 3/2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ Done!

Proof algorithm works: First note that it always terminates as we can visit step ① at most n times. The proof that the final matrix is in red. row echelon form is inductive.

Claim: When we arrive at step 1 the matrix has the form

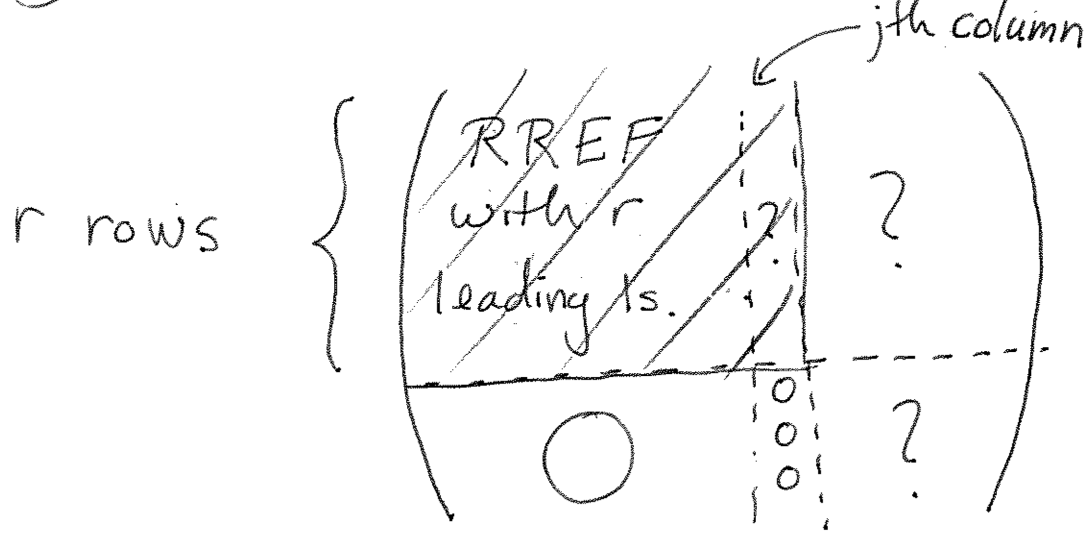


Base case: At the first visit, $r = 0$ and $j = 0$. Then the matrix cond holds vacuously.

Inductive step: Assume claim holds at step 1, show holds next time we return to step 1.

Two cases:

(a) We return via 2. In this case, we have



The new matrix is still in RREF

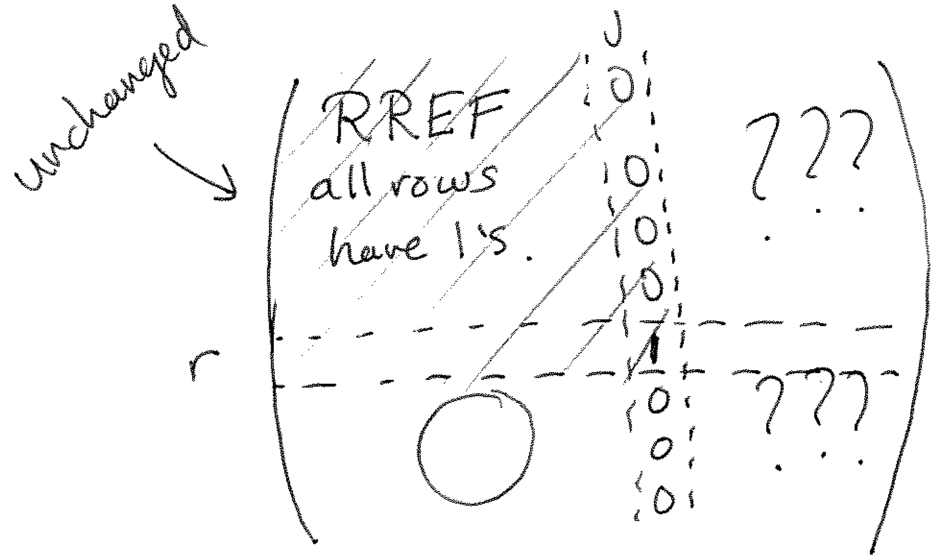
<u>because</u> the smaller matrix had r leading 1s.

in RREF

(Compare: $\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 0 & 9 \end{array}\right)$ } Not in RREF)

(b) We return via step (6). In this case

have incremented both r and j



and the new (r, j) submatrix is indeed

in RREF with r leading 1's.

So by induction the claim holds for all visits

to step (1), including the last one, which

means the returned matrix is in RREF form. ▨