# Number of Solutions

The most basic question you can ask about a system of equations is: how many solutions does it have?

Recall: The pivots of a matrix are the positions of the first nonzero entries of each row after putting the matrix into REF (using row operations).

The rank of a matrix is the number of pivots

Eg: from last time:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 2 & -3 & 2 & 14 \\ 3 & 1 & -1 & -2 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -7 & -4 & 2 \\ 0 & 0 & -\frac{50}{7} & -\frac{150}{7} \end{array}\right]$$

$$\begin{aligned} x_1+2x_2+3x_3&=6 \\ 2x_1-3x_2+2x_3&=14 \\ 3x_1+x_2-x_3&=-2 \end{aligned} \quad \rightsquigarrow \quad \begin{aligned} x_1+2x_2+3x_3&=6 \\ -7x_2-4x_3&=2 \\ -\tfrac{50}{7}x_3&=-\tfrac{150}{7} \end{aligned}$$

How many solutions? 1

Find the (only) solution using back-substitution.

Eg: from last time:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 9 & -1 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

$$\begin{aligned} x_1+2x_2+3x_3&=1 \\ 4x_1+5x_2+6x_3&=0 \\ 7x_1+8x_2+9x_3&=-1 \end{aligned} \quad \hookrightarrow \quad \begin{aligned} x_1+2x_2+3x_3&=1 \\ -3x_2+6x_3&=4 \\ 0&=0 \end{aligned}$$

How many solutions? $\infty$

We can choose any value for $x_3$: no pivot in $3^{rd}$ col.

**Q**: What is the difference between the previous two examples? (In terms of pivots.)

**Eg**: from last time:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 9 & 0 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & 4 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

$$\begin{aligned} x_1+2x_2+3x_3&=1 \\ 4x_1+5x_2+6x_3&=0 \\ 7x_1+8x_2+9x_3&=-1 \end{aligned} \quad \hookrightarrow \quad \begin{aligned} x_1+2x_2+3x_3&=1 \\ -3x_2+6x_3&=4 \\ 0&=1 \end{aligned}$$

How many solutions? $0$

**Q**: What is the difference between the previous three examples? (In terms of pivots.)

**Def**: A **pivot column** of a matrix is a column with a pivot position.

$\rightarrow$ Again, the pivots are in the REF matrix!

**Fact:** Let $A$ be the augmented matrix for a system of equations.

(1) If every column except the last is a pivot column, then the system has one solution.

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -7 & -4 & 2 \\ 0 & 0 & -\frac{50}{7} & -\frac{150}{7} \end{array}\right]$$

= pivot column
= not a pivot column

($\infty$) If the last column and some other column are not pivot columns, then there are infinitely many solutions.

can choose any value for $x_3$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

= pivot column
= not a pivot column

(0) If the last column is a pivot column, then there are zero solutions.

$$0 = 1 \rightarrow \left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & 4 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

= pivot column
= not a pivot column

**Def:** A system is consistent if it has at least 1 solution (so 1 or $\infty$). It is inconsistent otherwise.

# Gaussian Elimination

This is how a computer solves systems of linear equations using elimination. Almost all questions in this class will reduce to this procedure! (The interesting part is how they do so.)

**Def:** Two matrices are row equivalent if you can get from one to the other using row operations.

**NB:** If augmented matrices are row equivalent then they have the same solution sets.

**Algorithm** (Gaussian Elimination/row reduction):

Input: Any matrix

Output: A row-equivalent matrix in REF.

Procedure:

(1a) If the first nonzero column has a zero entry at the top row swap so that the top entry is nonzero.

$$\begin{bmatrix} 0 & 4 & 3 & 2 \\ 1 & 1 & -1 & 3 \\ 2 & -3 & -6 & -3 \end{bmatrix} \xrightarrow{R_1 \leftrightarrow R_2} \begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 2 & -3 & -6 & -3 \end{bmatrix}$$

This is now the first pivot position.

(1b) Perform row replacements to clear all entries below the first pivot.

$$\begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 2 & -3 & -6 & -3 \end{bmatrix} \xrightarrow{R_3 -= 2R_1} \begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & -5 & -4 & -9 \end{bmatrix}$$

Now ignore the row & column with the first pivot and recurse into the submatrix below and to the right:

$$\begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & -5 & -4 & -9 \end{bmatrix}$$

(2a) If the first nonzero column has a zero entry at the top row swap so that the top entry is nonzero.

$$\begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & -5 & -4 & -9 \end{bmatrix}$$

second pivot

(Not applicable to this matrix)

(2b) Perform row replacements to clear all entries below the first pivot.

$$\begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & -5 & -4 & -9 \end{bmatrix} \xrightarrow{R_3 += \frac{5}{4}R_2} \begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & 0 & -1/4 & -13/2 \end{bmatrix}$$

Doesn't mess up the 1st column!

etc. (recurse)

In our example, the recursion has terminated:

$$\begin{bmatrix} 1 & 1 & -1 & -3 \\ 0 & 4 & 3 & 2 \\ 0 & 0 & -1/4 & -13/2 \end{bmatrix}$$ is in REF!

Eg: $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 7 & 3 \\ 0 & 0 & 1 & 1 \end{bmatrix} \xrightarrow[(1b)]{R_2 -= 2R_1} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 1 & 1 \end{bmatrix}$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

This submatrix has no pivot in the first column! The first nonzero column is the second.

$$\xrightarrow[(2b)]{R_3 -= R_2} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix}$$

This is in REF: $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix}$

Important: If you want to apply this algorithm to an augmented matrix, just delete the augmentation line (pretend it's not augmented).

Demo: Gauss-Jordan slideshow

Use Rabinoff's Reliable Row Reducer on the HW!

# Jordan Substitution

This is the back-substitution procedure.

It is necessary when you have $\infty$ solutions.

It puts a matrix into the following form:

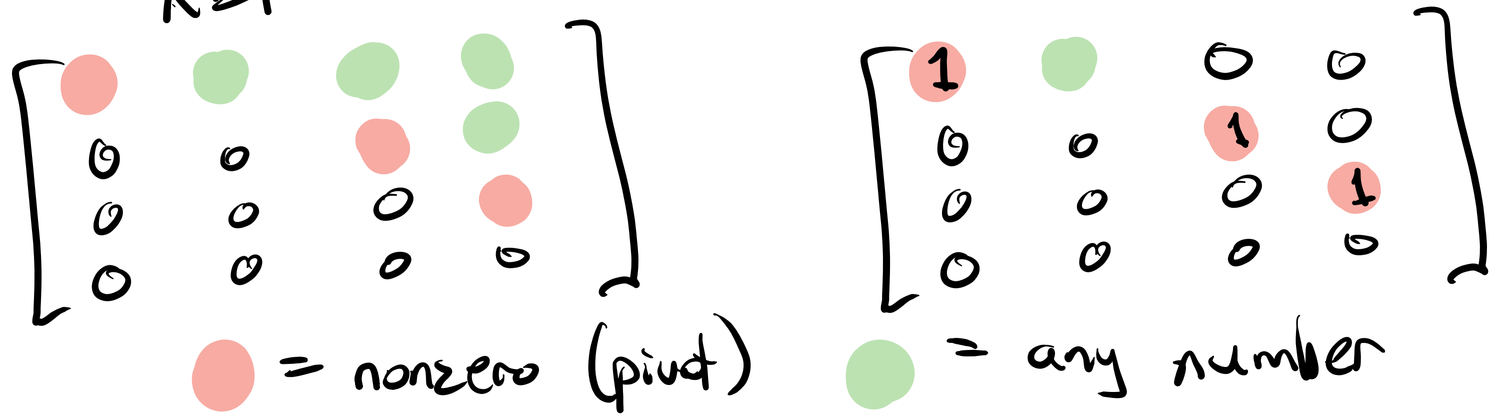
**Def:** A matrix is in reduced row echelon form (RREF) if:

(1–2) It is in REF

(3) All pivots are equal to 1.

(4) A pivot is the only nonzero entry in its column.

REF RREF



Eg: $\left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -7 & -4 & 2 \\ 0 & 0 & -\frac{50}{7} & -\frac{150}{7} \end{array}\right]$ is in REF. How to put into RREF?

Do back substitution!

**Row Operations** | **Back-Substitution**

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -7 & -4 & 2 \\ 0 & 0 & -\frac{50}{7} & -\frac{150}{7} \end{array}\right]$$

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 6 \\ -7x_2 - 4x_3 &= 2 \\ -\tfrac{50}{7}x_3 &= -\tfrac{150}{7} \end{aligned}$$

(scale so this is 1)

$R_3 \times = -\frac{7}{50}$ } solve for $x_3$

$$\left[\begin{array}{cccc} 1 & 2 & 3 & 6 \\ 0 & -7 & -4 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 6 \\ -7x_2 - 4x_3 &= 2 \\ x_3 &= 3 \end{aligned}$$

(kill these)

$R_1 -= 3R_3$
$R_2 += 4R_3$ } substitute $x_3 = 3$ into $R_1$ & $R_2$ then move the constants to the RHS

$$\left[\begin{array}{cccc} 1 & 2 & 0 & -3 \\ 0 & -7 & 0 & 14 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

$$\begin{aligned} x_1 + 2x_2 &= -3 \\ -7x_2 &= 14 \\ x_3 &= 3 \end{aligned}$$

(scale so this is 1) $R_2 \div = -7$ } solve for $x_2$

$$\begin{bmatrix} 1 & 2 & 0 & -3 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \qquad \begin{aligned} x_1 + 2x_2 \quad &= -3 \\ x_2 \quad &= -2 \\ x_3 &= 3 \end{aligned}$$

(kill this)

$R_1 -= 2R_2$ substitute $x_2 = -2$ into $R_1$ then move the constants to the RHS

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \qquad \begin{aligned} x_1 \quad &= 1 \\ x_2 \quad &= -2 \\ x_3 &= 3 \end{aligned}$$

This is in RREF:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}\right] \rightsquigarrow \begin{aligned} x_1 &= 1 \\ x_2 &= -2 \\ x_3 &= 3 \end{aligned}$$

Solved ✓

**Upshot:** Jordan substitution is exactly back-substitution.

**Demo:** Gauss-Jordan slideshow, cont'd

**Algorithm (Jordan Substitution):**

**Input:** A matrix in REF

**Output:** The row-equivalent matrix in RREF.

**Procedure:**

Loop, starting at the last pivot:

(a) Scale the pivot row so the pivot $=1$.

(b) Use row replacements to kill the entries above that pivot.

"theorem"

**Thm:** The RREF of a matrix is unique.

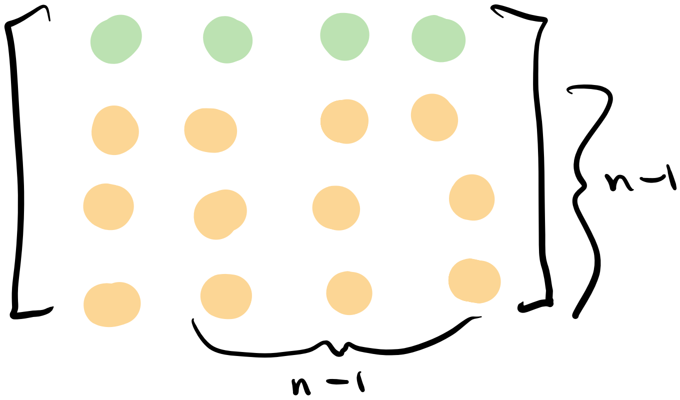
In other words, if you start with a matrix, do any legal row operations at all, and end with a matrix in RREF, then it's the same matrix that Gauss-Jordan will produce.

↳ Gaussian elimination + Jordan substitution.

# Computational Complexity

How much computer time does Gauss-Jordan take?

Gaussian Elimination on an $n \times n$ matrix takes:




**Step 1:** Each row replacement requires $n-1$ multiplications & $n-1$ additions. (no computation in 1st col: just write "0") Do for $n-1$ lower rows:

$$\begin{array}{rl} (n-1)(n-1) & \text{mult} \\ (n-1)(n-1) & \text{add} \\ \hline 2(n-1)^2 & \end{array}$$

**flops** = floating point operations


**Step 2:** Each row replacement requires $n-2$ multiplications & $n-2$ additions. Must do this for $n-2$ remaining rows

$$\begin{array}{rl} (n-2)(n-2) & \text{mult} \\ +\ (n-2)(n-2) & \text{add} \\ \hline 2(n-2)^2 & \text{flops} \end{array}$$

etc.

$$\text{Total: } 2[\overbrace{(n-1)^2+(n-2)^2+\cdots+1^2}^{\text{pyramidal number}}]$$

$$= 2\cdot\frac{n(n-1)(2n-1)}{6} \approx \frac{2}{3}n^3 \text{ flops}$$

## Back-Substitution

$\bullet\, x_n = \bullet$ — 1 mult = 1 flop

$\bullet\, x_{n-1} + \bullet\, x_n = \bullet$ — 2 mult, 1 add = 3 flops

(substitute $x_n$, $\times\bullet$, subtract, $\div\bullet$)

$\bullet\, x_{n-2} + \bullet\, x_{n-1} + \bullet\, x_n = \bullet$ — 3 mult, 2 add = 5 flops

(substitute $x_n$ & $x_{n-1}$, $\times\bullet$, $\times\bullet$, subtract, $\div\bullet$)

$\bullet\, x_1 + \cdots + \bullet\, x_n = \bullet$ — $n$ mult, $(n-1)$ add $= 2n-1$ flops

**Total:** $1+3+5+\cdots+(2n-1) = n^2$ flops

**NB:** $\frac{2}{3}n^3$ is a lot more than $n^2$!

For a $1000\times1000$ matrix, $\frac{2}{3}n^3 \approx \frac{2}{3}$ gigaflops but $n^2 = 1$ megaflop. If we want to solve $Ax=b$ for 1000 values of $b$, doing elimination each time takes $\frac{2}{3}$ teraflops!