Writing Systems of Linear Equations

Here's a system of 2 equations in 3 variables.

System of Equations

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 2 \\ 2x_1 + 3x_2 + 2x_3 = 6 \end{cases}$$

We're going to use matrices and vectors to solve it. Here are 3 more equivalent ways to write it:

Matrix Equation $Ax = b$

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

A $\quad$ x $\quad$ b

- A is the coefficient matrix: it contains the coefficients of the variables
  - → It is an $m \times n$ matrix
    - m = # equations $(=2)$
    - n = # variables $(=3)$
- b contains the constants on the right of the $=$
  - → It is a vector in $\mathbb{R}^m$

- x contains the variables
  → it is a vector in $\mathbb{R}^n$

If you expand out the product you get

$$\begin{pmatrix} 1x_1 + 2x_2 + 3x_3 \\ 2x_1 + 3x_2 + 2x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

which is the same as our original system of eq^{n}s.

Vector equation $x_1v_1 + x_2v_2 + \cdots + x_nv_n = b$

$$x_1\begin{pmatrix} 1 \\ 2 \end{pmatrix} + x_2\begin{pmatrix} 2 \\ 3 \end{pmatrix} + x_3\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

- The vectors $\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ are the columns of the coefficient matrix A.
  → They are vectors in $\mathbb{R}^m$
- The b vector $\begin{pmatrix} 2 \\ 6 \end{pmatrix}$ is the same.
- The variables are now the weights in a linear combination of $\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

If you expand out $Ax = b$ by columns, you get this vector equation:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \overset{\text{def}}{=} x_1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} + x_2 \begin{pmatrix} 2 \\ 3 \end{pmatrix} + x_3 \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

Augmented Matrix $(A|b)$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \end{array}\right)$$

This is just a shorthand way of recording all of the numbers in A & b. The augmentation line reminds you which is A & which is b.

It is easy to translate between these four representations of a system of equations.

$$\begin{cases} 1x_1 + 2x_2 + 3x_3 = 2 \\ 2x_1 + 3x_2 + 2x_3 = 6 \end{cases} \quad \rightleftharpoons \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \end{pmatrix} x = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \end{array}\right) \quad \rightleftharpoons \quad \begin{pmatrix} 1 \\ 2 \end{pmatrix} x_1 + \begin{pmatrix} 2 \\ 3 \end{pmatrix} x_2 + \begin{pmatrix} 3 \\ 2 \end{pmatrix} x_3 = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

Row Operations

We are going to manipulate our system of equations to make it simpler as easier to solve.

Eg: $x_1 + 2x_2 = -1$, $2x_1 + 3x_2 = -1$

"Subtract 2x the 1st eqn from the 2nd"

$R_2 -= 2R_1$

$x_1 + 2x_2 = -1$

$-x_2 = 1$

"Multiply the 2nd eqn by -1"

$R_2 *= -1$

$x_1 + 2x_2 = -1$

$x_2 = -1$

$R_1 -= 2R_2$

$x_1 = 1$

$x_2 = -1$

There are 3 kinds of valid manipulations, called row operations. They really only change the numbers, not the variables, so it's easier to apply them to (augmented) matrices — there's less writing.

$$1x_1 + 2x_2 = -1$$
$$2x_1 + 3x_2 = -1$$
$$\rightsquigarrow \left(\begin{array}{cc|c} 1 & 2 & -1 \\ 2 & 3 & -1 \end{array}\right)$$

Row ops also make sense for any (non-augmented) matrix.

Row Operations:

(1) **Row Replacement**: replace R_i by $R_i + c\,R_j$ (any number c)

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \end{array}\right) \xrightarrow{R_2 \to -2R_1} \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \end{array}\right)$$

(2) **Row Swap**: interchange R_i & R_j

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \end{array}\right) \xrightarrow{R_1 \leftrightarrow R_2} \left(\begin{array}{ccc|c} 2 & 3 & 2 & 6 \\ 1 & 2 & 3 & 2 \end{array}\right)$$

(3) **Row Scale**: multiply R_i by a nonzero number

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \end{array}\right) \xrightarrow{R_1 \times 2} \left(\begin{array}{ccc|c} 2 & 4 & 6 & 4 \\ 2 & 3 & 2 & 6 \end{array}\right)$$

Note 1: If (x_1, x_2, x_3) is a solution of the corresponding system of equations **before** doing a row operation, then it is still a solution **after**.

Eg: $x_1 = 1 \quad x_2 = 2 \quad x_3 = -1$ is a solution of the system

$$\begin{array}{l} x_1 + 2x_2 + 3x_3 = 2 \\ 2x_1 + 3x_2 + 2x_3 = 6 \end{array} \xrightarrow{R_2 \to -2R_1} \begin{array}{l} x_1 + 2x_2 + 3x_3 = 2 \\ -x_2 - 4x_3 = 2 \end{array}$$

$$\begin{array}{l} 1 + 2(2) + 3(-1) = 2 \\ 2(1) + 3(2) + 2(-1) = 6 \end{array} \xrightarrow[\text{sides of } R_2]{\text{subtract 4 from both}} \begin{array}{l} 1 + 2(2) + 3(-1) = 2 \\ -2 - 4(-1) = 2 \end{array}$$

Note 2: All of the row operations are **reversible**: you can undo it with another row operation.

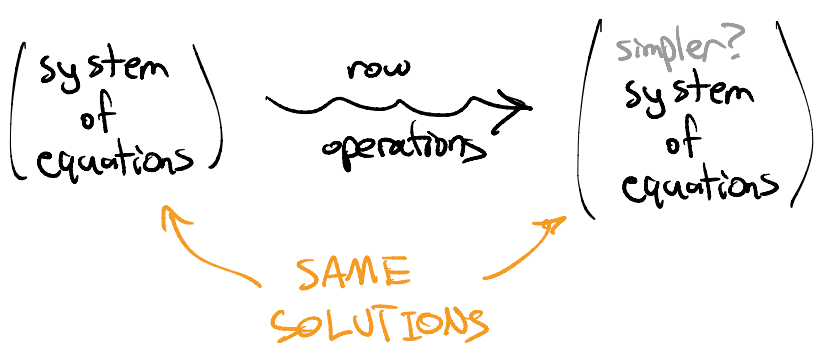
Eg: How do you undo these row operations?

- $R_1 -= 2R_2$ $\quad R_1 += 2R_2$
- $R_1 \hookleftarrow R_2$ $\quad R_1 \hookleftarrow R_2$
- $R_1 \times= 2$ $\quad R_1 \div= 2$

Note 1 + Note 2 imply that

> Row operations don't change the solution set of a system of equations!

In other words, (x_1, x_2, x_3) is a solution of our system of equations **before** doing a row op $\iff$ it is a solution **after**.



Solving Systems of Equations using Elimination

How do we make our equations simpler using row ops?

Idea: we want to eliminate some variables from some equations.

Eg:

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

augmented matrix $\rightsquigarrow$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 2 & 3 & 2 & 6 \\ 3 & 1 & -1 & 6 \end{array}\right)$$

$R_2 -= 2R_1$ $\rightsquigarrow$

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

$R_3 -= 3R_1$ $\rightsquigarrow$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \\ 0 & -5 & -10 & 0 \end{array}\right)$$

Now x_1 has been eliminated from R_2 & R_3: it doesn't appear! This system of equations is

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 2 \\ -x_2 - 4x_3 &= 2 \\ -5x_2 - 10x_3 &= 0 \end{aligned}$$

$\leftarrow$ 2 equations in 2 variables!

We can keep going to eliminate x_2 from R_3:

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \\ 0 & -5 & -10 & 0 \end{array}\right) \xrightarrow{R_3 - = 5R_2} \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \\ 0 & 0 & 10 & -10 \end{array}\right)$$

Now we have only one variable in R_3!

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 2 \\ -x_2 - 4x_3 &= 2 \\ \boxed{10x_3 = -10} &\leftarrow \text{1 equation in 1 variable!} \end{aligned}$$

At this point we have isolated x_3. There is only one possible solution: $x_3 = -1$.

Now we substitute $x_3 = -1$ into R_1 & R_2:

$$\begin{aligned} x_1 + 2x_2 + 3(-1) &= 2 \\ -x_2 - 4(-1) &= 2 \end{aligned} \quad \rightsquigarrow \quad \begin{aligned} x_1 + 2x_2 &= 5 \\ -x_2 &= -2 \end{aligned}$$

We have isolated x_2: the only solution is $x_2 = 2$.

Now we substitute $x_2 = 2$ into R_1:

$$x_1 + 2(2) = 5 \rightsquigarrow x_1 = 1$$

We have isolated x_1: the only solution is $x_1 = 1$.

So the system has a unique solution

$$x_1 = 1 \qquad x_2 = 2 \qquad x_3 = -1.$$

Check:

$1(1) + 2(2) + 3(-1) = 2$

$2(1) + 3(2) + 2(-1) = 6$

$3(1) + 1(2) - 1(-1) = 6$ ✓

Summary:

- There were 2 main steps.
  - **Elimination**: eliminate variables from eqns
  - **Substitution**: isolate each variable to solve for it then substitute that value into the other equations

- There was a **unique** solution because all 3 variables could be isolated.
  (more on this later)

We'll have different algorithms for these.

Elimination ⇝ Gaussian Elimination
Substitution ⇝ Jordan Substitution

It's important to keep them distinct, since

> Elimination is much slower than Substitution

(More on this later too.)

When combined we call it Gauss-Jordan Elimination.

Row Echelon Forms

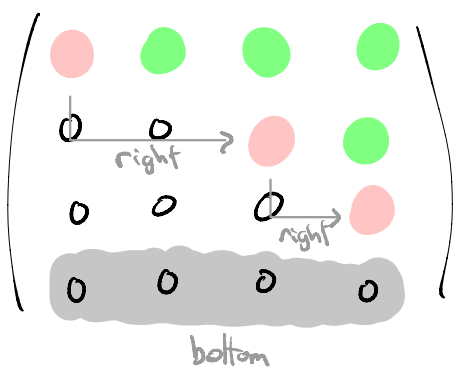
What does it mean for elimination to be "done"?
When do these algorithms terminate?

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

(1) The first nonzero entry of each row is to the **right** of the first nonzero entry of the row above it.

(2) All zero rows are at the **bottom**.

Picture:



= nonzero

= any number

In REF:

$$\begin{pmatrix} 1 & 2 & -1 & 4 \\ 0 & 0 & 3 & 12 \end{pmatrix} \qquad \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \\ 0 & 0 & 10 & -10 \end{array}\right)$$

Not in REF:

$$\begin{pmatrix} 1 & 2 & -1 & 4 \\ \cancel{3} & 0 & 0 & 12 \end{pmatrix} \qquad \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 0 & 10 & -10 \end{array}\right)$$

Important: When deciding if an augmented matrix is in REF, **ignore the augmentation line.**

Idea: If an augmented matrix is in REF, then there is nothing left to eliminate!

Each nonzero entry has already been used to eliminate that variable from the equations after it.

Def: The pivot positions (pivots) of a matrix are the positions of the 1st nonzero entries in each row after you put it into REF (using row operations).

$$\begin{pmatrix} 1 & 2 & -1 & 4 \\ 0 & 0 & 3 & 12 \end{pmatrix} \qquad \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ 0 & -1 & -4 & 2 \\ 0 & 0 & 10 & -10 \end{array}\right)$$

● = pivots

If REF is the output of elimination, what is the output of substitution?

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

(1-2) It is in REF. (So RREF is also REF)

(3) All pivots are $= 1$.

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

REF: $\begin{pmatrix} ● & ● & ● & ● \\ 0 & 0 & ● & ● \\ 0 & 0 & 0 & ● \\ 0 & 0 & 0 & 0 \end{pmatrix}$ RREF: $\begin{pmatrix} 1 & ● & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

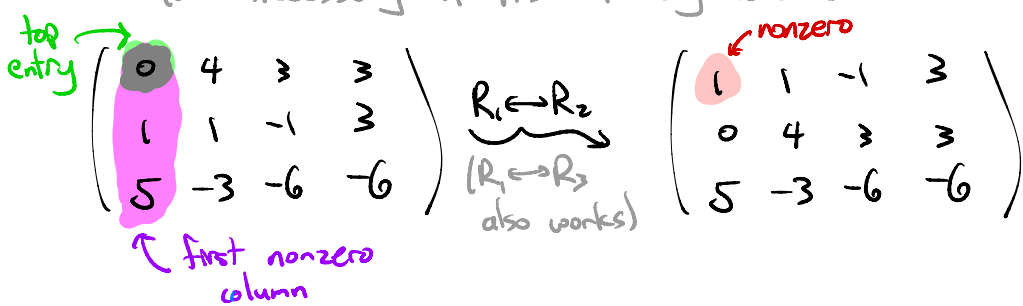
Gaussian Elimination Algorithm

This is how you would program a computer to systematically put a matrix into REF using row ops. (It also works fine for solving systems of eq^{n}s by hand.)

Gaussian Elimination: Start with any matrix.

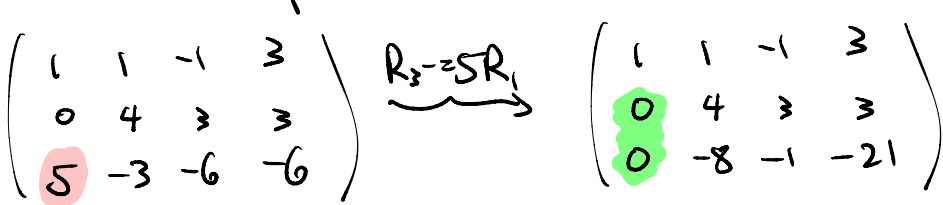
(1a) Perform a row swap so that the top entry of the first nonzero column is nonzero.

(Not necessary if it's already nonzero.)

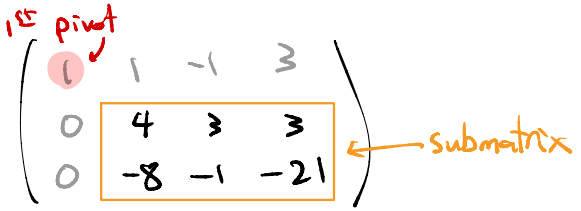


This is the first pivot position.

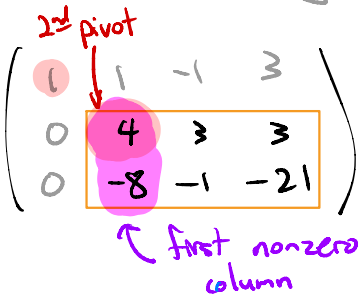
(1b) Perform row replacements to eliminate all entries below the first pivot.



Now recurse into the submatrix to the right & below the 1st pivot & ignore everything else.



(2a) Perform a row swap so that the top entry of the first nonzero column of the submatrix is nonzero.
(Not necessary if it's already nonzero.)



(Not needed for this matrix, but you could do $R_2 \leftrightarrow R_3$.)

This is the second pivot position.

(2b) Perform row replacements to eliminate all entries below the second pivot.

Now recurse into the submatrix to the right & below the 2nd pivot & ignore everything else.

(3a) etc...

In our example, the algorithm terminates after (2b):

$$\begin{pmatrix} 1 & 1 & -1 & 3 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 5 & -15 \end{pmatrix}$$ is in REF!

Jordan Substitution Algorithm

This is the substitution procedure: it takes a matrix in REF and performs row operations to produce a matrix in RREF.

Jordan Substitution: Start with a matrix in REF.

Loop, starting at the last pivot:

(a) Perform a row scale so the pivot is $= 1$

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

This exactly corresponds to the substitution procedure!

Eg: $\left(\begin{array}{ccc|c} 1 & 1 & -1 & 3 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 5 & -15 \end{array}\right)$ is in REF

Row operations

$$\left(\begin{array}{ccc|c} 1 & 1 & -1 & 3 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 5 & -15 \end{array}\right)$$

↑ last pivot

(a) $R_3 \div = 5$

← kill these

$$\left(\begin{array}{ccc|c} 1 & 1 & -1 & 3 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 1 & -3 \end{array}\right)$$

(b) $R_1 \mathrel{+}= R_3$, $R_2 \mathrel{-}= 3R_3$

$$\left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 4 & 0 & 12 \\ 0 & 0 & 1 & -3 \end{array}\right)$$

↑ next-last pivot

(a) $R_2 \div = 4$

← kill this

$$\left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -3 \end{array}\right)$$

(b) $R_1 \mathrel{-}= R_2$

Substitution

$$\begin{aligned} x_1 + x_2 - x_3 &= 3 \\ 4x_2 + 3x_3 &= 3 \\ 5x_3 &= -15 \end{aligned}$$

↓ solve for x_3

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

↓ substitute $x_3 = -3$ into R1 & R2

$$\begin{aligned} x_1 + x_2 &= 0 \\ 4x_2 &= 12 \\ x_3 &= -3 \end{aligned}$$

↓ solve for x_2

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

↓ substitute $x_2 = 3$ in R_1

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -3 \end{array}\right)$$

$x_1 = -3$

$x_2 = 3$

$x_3 = -3$

This is in RREF:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -3 \end{array}\right)$$

This is solved!

$x_1 = -3$

$x_2 = 3$

$x_3 = -3$

(Starting matrix)

$$\left(\begin{array}{ccc|c} 0 & 4 & 3 & 3 \\ 1 & 1 & -1 & 3 \\ 5 & -3 & -6 & -6 \end{array}\right)$$

Check:

$0(-3) + 4(3) + 3(-3) = 3$

$1(-3) + 1(3) - 1(-3) = 3$

$5(-3) - 3(3) - 6(-3) = -6$ ✓

Important: If you want to apply these algorithms to an augmented matrix, just delete the augmentation line.

(It's only there to remind you that the matrix contains the coefficient matrix & the b vector.)

[DEMO]: Gauss–Jordan Slideshow

Use Rabinoff's Reliable Row Reducer to practice doing elimination without worrying about making arithmetic errors.

Once you've mastered the procedure, use the Sage cell on the course webpage:

Sage Cell

Want to do a quick calculation? This is a Sage Cell running python and SymPy.

```
1 from sympy import *
2 A = Matrix([[1,2,3],[4,5,6],[7,8,9]])
3 pprint(A.rref(pivots=False))
```

Evaluate

If you do both algorithms:

Gaussian Elimination + Jordan Substitution
= Gauss-Jordan Elimination

then you transform any matrix to RREF!

Idea: Solving a system of linear equations $\Leftrightarrow$ Putting a matrix into RREF using row ops

"Theorem"

Thm: The RREF of a matrix is unique: if you transform your matrix to RREF using any sequence of row ops, then you will always end up with the same matrix.

This is not true for REF: a different choice of pivots in step (a) will give different REFs.

NB: Jordan substitution doesn't change the pivots. So the Thm implies the pivot positions are also unique!

Warning: There might be a more clever sequence of row ops that also transforms your matrix to (R)REF. This is great if you only care about the (R)REF of your matrix, but sometimes you need to do the algorithm(s) as written! (LU decompositions, for instance.)