LU-Factorization/Decomposition:

However, what if our 'b' values were to change?

Augmented Matrix ⇒ Upper Triangular Matrix ⇒ Solve for unknowns

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Through tracking our scaling factors when performing Gauss Elimination we can find the solution to this 'A' matrix no matter what values are put in the 'b' (or constant) matrix

$$A \cdot \vec{x} = \vec{b}$$

After Gauss elimination

upper-triagular matrix ← $U \cdot \vec{x} = \vec{c}$ → new constant vector

$$LU \cdot \vec{x} = L\vec{c}$$

A

$L \cdot \vec{c} = \vec{b}$ ← Through this we can easily solve for '$\vec{c}$' → (Forward substitution)

$U \cdot \vec{x} = \vec{c}$ ← then plug $\vec{c}$ in this equation to get the unknown variables '$\vec{x}$' (Backward substitution)

- In Summary, LU factorization is similar to Gauss Elimination, however, it is much more versatile and much faster computation wise (If more than one constant vector $\vec{b}$ is needed to be solved for).

Partial pivoting

If $|a_{i,i}| < |a_{j,i}|$ then switch our rows i & j

Solve the following SLE using LU Decomposition

$$\underset{A}{\begin{bmatrix} 1 & 3 & 2 \\ 3 & 2 & 6 \\ 2 & 4 & 8 \end{bmatrix}} \underset{\vec{x}}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}} = \underset{\vec{b}}{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}}$$

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 3 & 2 \\ 0 & 1 & 0 & 3 & 2 & 6 \\ 0 & 0 & 1 & 2 & 4 & 8 \end{array}\right] \begin{matrix} \\ \text{II} - 3\text{I} \\ \text{III} - 2\text{I} \end{matrix}$$

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 3 & 2 \\ 3 & 1 & 0 & 0 & -7 & 0 \\ 2 & 0 & 1 & 0 & -2 & 4 \end{array}\right] \begin{matrix} \\ \\ \text{III} - \frac{2}{7}\text{II} \end{matrix}$$

$$\longrightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & \frac{2}{7} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 1 & 3 & 2 \\ 0 & -7 & 0 \\ 0 & 0 & 4 \end{bmatrix}}_{U}$$

$L \cdot \vec{C} = \vec{b} \longrightarrow$ **Forward substitution**

$$C_1 = 1$$

$$C_2 = 2 - 3 = -1$$

$$C_3 = 3 - 2(1) - \frac{2}{7}(-1)$$

$$\Rightarrow \boxed{C_3 = \frac{9}{7} \simeq 1.2857}$$

$U \cdot \vec{x} = \vec{C} \longrightarrow$ **Backward substitution**

$$\begin{bmatrix} 1 & 3 & 2 \\ 0 & -7 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 9/7 \end{bmatrix}$$

$$4x_3 = \frac{9}{7} \Rightarrow \boxed{x_3 = \frac{9}{28}}$$

$$-7x_2 = -1 \Rightarrow \boxed{x_2 = \frac{1}{7}}$$

$$x_1 = 1 - 3x_2 - 2x_3 = 1 - \frac{3}{7} - \frac{9}{14} = 1 - \frac{15}{14}$$

$$\Rightarrow \boxed{x_1 = -\frac{1}{14}}$$

Iterative Vs. Direct Numerical Methods:

- **Direct:** Ways of solving systems of linear equations through a finite # of steps.
  Ex: Gaussian Elimination (LU Factorization/Decomposion)

System of Linear Equations → Augmented Matrix → Upper-Triangular Matrix → Solution

After a certain # of steps, we arrive at a final solution

- **Iterative:** Do <u>not</u> necessarily stop after a certain # of steps! Will continue to iterate until a sufficient amount of error is acheived.

Ex: Bisection Method, Newton's Method, and Jacobi Iteration.

☆ These need an initial guess to begin the solving!

Direct	Iterative
• Can't stop after a certain # of steps	• We can choose to stop whenever we want. It's just a tradeoff between Accuracy and computation time!

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

↓ all to zero, then backward sub. to solve.

What are iterative methods and can they be used to solve systems of linear equations?

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

- We could use direct numerical methods to solve this, However, Direct numerical methods are not ideal for very large SLE and in non-linear equations direct methods don't work.

How do we develop an iterative methode to solve the matrix above?

$$\begin{bmatrix} \sum_{j=1}^{n} a_{1,j} \cdot x_j \\ \sum_{j=1}^{n} a_{2,j} \cdot x_j \\ \sum_{j=1}^{n} a_{3,j} \cdot x_j \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$a_{1,1} \cdot x_1 + a_{1,2} \cdot x_2 + a_{1,3} \cdot x_3$

constant · variable = linear equation

$b_1 - a_{1,2} \cdot x_2 - a_{1,3} x_3 = a_{1,1} \cdot x_1$

Then divide by $a_{1,1}$!

* Our goal is to find some x-values that will give us the proper b-values!

For a single row which we will call i: $\sum_{j=1}^{n} a_{i,j} \cdot x_j = b_i$

$$x_i = \frac{b_i - \sum_{j=1, j \neq i}^{n} a_{i,j} \cdot x_j}{a_{i,i}}$$

← We took $a_{i,i} \cdot x_i$ out of the summation, think of this as taking and isolating say $a_{1,1} \cdot x_1$ above.

- We have no way of solving this problem currently as we still don't know our x-values that satisfy our equation.

Our guess number.

$$x_i^{(K+1)} = \frac{b_i - \sum_{j=1, j \neq i}^{n} a_{i,j} \cdot x_j^{(K)}}{a_{i,i}}$$

← so we are using our old x-values guesses to find our new one!

These are the n-equations that we use to iterate.

$$x_i^{(K+1)} = x_i^{(K)} + \frac{b_i - \sum_{j=1}^{n} a_{i,j} \cdot x_j^{(K)}}{a_{i,i}}$$

This was solved through $\ominus$ the $x_i^{(K)}$

$$x_i^{(K+1)} = x_i^{(K)} + \frac{b_i - A_i \, x^{(K)}}{a_{i,i}}$$

Remember: $A \cdot x = b \rightarrow$ so $\rightarrow A_i \cdot x^{(K)} = b_i$ is the b-value based on our guesses for the x-values! Therefore, it's basically a Remainder term.

- Iterative methods are ways to solve our matrix (so find the x-values) without performing all the steps in direct numerical methods!

$b_i - A_i \cdot x^{(K)} =$ "Remainder" (i.e. how far off we were from the real x-values! Thus, our goal is to turn this to zero! as close as we could get!

as K↑ then $R^{(K)} \rightarrow 0$

(as # of iterations↑)

Just Note: Direct method gives an EXACT answer, where, iterative methods are APPROXIMATE!

Why did we go through all the work above?

↳ Iterative methods can be quicker for large zero matrices, which can be a problem for direct numerical methods.

The problem is we can never be certain about the number of iterations that are required.

↳ ∴ Only Perform when you have no choice or you're certain that the matrix converges.

Note:

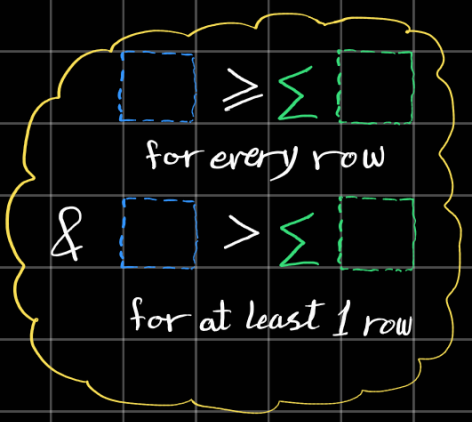
1: ↓ iterations (K) needed if good/educated guesses are used initially.

2: Convergence is guarenteed if matrix is Diagonally dominant.

What is a Diagonally Dominant Matrix?

Condition of a diagonally dominant matrix: $|a_{i,i}| \geq \sum_{j=1, j\neq i}^{n} a_{i,j}$ for all rows with at least one row being strictly greater than (>)!

- This is important as we are guarenteed using iterative methods if our square matrix of interest is Diagonally Dominant.



$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

* To ensure no mistakes are made, strictly focus on one row at a time!

Performing Jacobi Iteration (by hand) example:

↳ Requirements:
1. Square Matrix
2. Linear
3. Ideally Diagonally Dominant → Ensure convergence
   → If not, we either need a better starting guess, or it won't converge.

$$\left[\begin{array}{ccc|c} 9 & 2 & 3 & 7 \\ 1 & 12 & 9 & 2 \\ 4 & 6 & 14 & 1 \end{array}\right]$$

check for diagonally dominance:

- is $9 \geq |2|+|3|$ | is $12 \geq |1|+|9|$ | is $14 \geq |4|+|6|$
  $9 \geq 5$ yes | $12 \geq 10$ yes | $14 \geq 10$ yes

∴ Matrix is diagonally dominant

1. Assume an initial guess for the x-vector:
   $x_1^0 = 0,\ x_2^0 = 0,\ x_3^0 = 0$
2. Rewrite system of equations to solve for an x-value.

$$9x_1 + 2x_2 + 3x_3 = 7 \longrightarrow x_1^{k+1} = \frac{7 - [2x_2^k + 3x_3^k]}{9}$$

$$x_1 + 12x_2 + 9x_3 = 2 \longrightarrow x_2^{k+1} = \frac{2 - [x_1^k + 9x_3^k]}{12}$$

$$4x_1 + 6x_2 + 14x_3 = 1 \longrightarrow x_3^{k+1} = \frac{1 - [4x_1^k + 6x_2^k]}{14}$$

	0	1	2	3	4	
x_1	0	$\frac{7}{9}$	$\frac{271}{378}$ ≈ 0.717	$\frac{5723}{6804}$ ≈ 0.841	$\frac{4481}{5832}$ ≈ 0.76835	≈ 0.826
x_2	0	$\frac{1}{6}$	$\frac{73}{1512}$ ≈ 0.0483	$\frac{1241}{4536}$ ≈ 0.2736	$\frac{4949}{23328}$ ≈ 0.21215	≈ 0.3172
x_3	0	$\frac{1}{14}$	$-\frac{2}{9} = -0.\overline{2}$	$-\frac{233}{1512}$ ≈ -0.1541	≈ -0.286145	≈ -0.239

- Validate we are converging:

$$0 \leq 7 - 9x_1 - 2x_2 - 3x_3$$
$$0 \leq 2 - 1x_1 - 12x_2 - 9x_3$$
$$0 \leq 1 - 4x_1 - 6x_2 - 14x_3$$

iteration	Error in Equation 1
1	$= 7 - 9(0.\overline{777}) - 2(0.1\overline{66}) - 3(0.714) = -0.5476$
2	$= 7 - 9(0.7169) - 2(0.04828) - 3(-0.\overline{222} = 1.1177$
3	$= -0.65498$
4	$= 0.5140$
5	$= -0.3515$
6	$= 0.2648$
7	$= -0.1839$

if we kept going earlier

Converging (error is decreasing!)

Gauss-Seidel Method: An iterative numetrical methode that can aide us in solving Linear systems of equations!

↳ very similar to Jacobi Methode.

Jacobi Method	vs.	Gauss-Seidel Method
solve for all new x-values before implementing		use newly found x-values imediatly.

Gauss-Seidel procedure:

① Make a guess for all values in our systems x-matrix

$x_1^0 = 0$
$x_2^0 = 0$
$x_3^0 = 0$
$\vdots$

We typically use a zero matrix as an initial guess, unless we have additional information about the problem.

Better starting guess = faster and more likely to converge

② Write all independant equations to solve for different x-values.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \longrightarrow a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1$$

$$x_1 = \frac{b_1 - [a_{12} \cdot x_2 + a_{13} \cdot x_3]}{a_{11}}$$

* Repeat for all independent equations

③ Create a summary table

variable	iter. 0	iter. 1	iter. 2	...
x_1	$x_1^0 = 0$			
x_2	$x_2^0 = 0$			
x_3	$x_3^0 = 0$			

④ Solve for iteration 1:

$$x_1^1 = \frac{b_1 - [a_{12} \cdot x_2^0 + a_{13} \cdot x_3^0]}{a_{11}}$$

$$x_2^1 = \frac{b_2 - [a_{21} \cdot x_1^1 + a_{23} \cdot x_3^0]}{a_{22}}$$

$$x_3^1 = \frac{b_3 - [a_{31} \cdot x_1^1 + a_{32} \cdot x_2^1]}{a_{33}}$$

★ Always using our most recent value of x_i to solve for the next $x_i^{(i+1)}$ value

⑤ Repeat step 4 for iteration 2 using same method!

⑥ Keep iterating until the difference between the Left hand side of your independent equations = the right hand side of your independent equations

$$\text{Absolute value}\left[\begin{array}{l}\text{LHS of Eqn}\\ -\\ \text{RHS of Eqn}\end{array}\right] \leq \text{acceptable Error}$$