CSCC37 Floating Point Number Notes

1. Introduction:

- Can the following numbers be represented exactly on a computer:

a) $\pi \rightarrow$ No, because it goes on forever.
b) $\sqrt{2} \rightarrow$ No, because it goes on forever
c) $1/10 \rightarrow$ No, because there are conversion issues.

2. Representation of Non-negative Integers:

- Decimal (Base 10) System:
- $(350)_{10} = 3\times10^2 + 5\times10^1 + 0\times10^0$

- Binary (Base 2) System:
- $(10101110)_2 = 1\times2^8 + 0\times2^7 + 1\times2^6 + 0\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$
$= (350)_{10}$

- Hexadecimal (Base 16) System:
- Has 0, 1, ..., 9, A, B, C, D, E, F
- $(8FA)_{16} = 8\times16^2 + F\times16^1 + A\times16^0$

F is the 15th digit, so we substitute 15 for it

A is the 10th digit so we substitute 10 for it.

$= 8\times16^2 + 15\times16^1 + 10\times16^0$
$= (2298)_{10}$

- Given a base b system, where $b>0$ and $b \in Z$, suppose $x = (d_n d_{n-1} \ldots d_0)_b$. Then, $x = d_n \cdot b^n + d_{n-1} b^{n-1} + \ldots + d_0 b^0$, where $x \geq 0$, $x \in N$, $0 \leq d_i < b$, $i = 0, 1, \ldots n$.

- Converting from base b to base 10:
- Suppose we have $x = (d_n d_{n-1} \ldots d_0)_b$. To convert x to base 10, simply do $d_n b^n + d_{n-1} b^{n-1} + \ldots + d_0 b^0$.

E.g. 1 Convert $(235)_8$ to base 10.

Soln:

$$(235)_8 = 2 \cdot 8^2 + 3 \cdot 8^1 + 5 \cdot 8^0$$
$$= 128 + 24 + 5$$
$$= (157)_{10}$$

E.g. 2 Convert $(1011)_2$ to base 10.

Soln:

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 8 + 0 + 2 + 1$$
$$= (11)_{10}$$

- Converting from base 10 to base b:

E.g. 3 Convert $(350)_{10}$ to base 2

Soln:

Numerator	Denominator	Quotient	Remainder
350	2	175	0
175	2	87	1
87	2	43	1
43	2	21	1
21	2	10	1
10	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

Read bottom to up

$(350)_{10} = (101011110)_2$

- This method/technique works for any base.
- We stop when the quotient = 0. (It will always reach 0 at some point.)
- For the conversion, write the values in the remainder column from bottom to top.
- This conversion is safe except for overflow.

E.g. 4 Convert $(110)_{10}$ to base 16.

Soln:

Numerator	Denominator	Quotient	Remainder
110	16	6	14
6	16	0	6

Since the 14th digit in hexadecimal is E,

$(110)_{10} = (6E)_{16}$

E.g. 5 Convert $(140)_{10}$ to base 8.

Soln:

Numerator	Denominator	Quotient	Remainder
140	8	17	4
17	8	2	1
2	8	0	2

$(140)_{10} = (214)_8$

3. Representation of Reals:

- If $x \in R$, then $x = \pm(x_I . x_F)_b$ where

x_I = Integral part and x_F = Fractional part.

$x = \pm(x_I . x_F)_b$

$= \pm(d_n \ldots d_0 . d_{-1} d_{-2} \ldots)_b$

Note: The sign, +/−, is 1 bit, 0 or 1.

Note: x_I is a non-negative integer.

Note: x_F can be infinite.

E.g. $(0.77\ldots)_{10} = (0.\overline{7})_{10}$

$= 7 \times 10^{-1} + 7 \times 10^{-2} + \ldots$

- For a binary system, suppose we have $(.0001100011\ldots)_2$. This is equivalent to $(.0\overline{0011})_2$ which equals to $0\times2^{-1} + 0\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + 1\times2^{-5} + \ldots$

- Given a base b system, suppose we have $X_F = (.d_{-1}d_{-2}\ldots)_b$. Then:

$$X_F = d_{-1}b^{-1} + d_{-2}b^{-2} + \ldots$$

$$= \sum_{i=1}^{\infty} d_{-i}\,b^{-i}$$

Note: A terminating binary fraction with n digits always has a terminating decimal representation.

However, in base 3, this isn't true.

E.g. $(0.1)_3 = 1\times3^{-1}$

$= \frac{1}{3}$ ← Does not terminate

- Converting reals in base 10 to base b:

E.g. 6 Convert $(.625)_{10}$ to base 2.

Soln:

Multiplier	Base	Product	Integral		Fraction
.625	2	1.25	1	Read top down	0.25
0.25	2	0.5	0		0.5
0.5	2	1.0	1	↓	0

$(0.625)_{10} = (.101)_2$

- We stop when the fraction column hits 0.

Note: Sometimes, it may never hit 0.

- We read the integral column top down to get the corresponding base b value.

Note: A terminating decimal fraction may not have a terminating binary fraction.

E.g. 7 Convert $(.1)_{10}$ to base 2

Soln:

Multiplier	Base	Product	Integral	Fraction
0.1	2	0.2	0	0.2
0.2	2	0.4	0	0.4
0.4	2	0.8	0	0.8
0.8	2	1.6	1	0.6
0.6	2	1.2	1	0.2
0.2	2	0.4	0	0.4
0.4	2	0.8	0	0.8
0.8	2	1.6	1	0.6
0.6	2	1.2	1	0.2

Repeats

The cycle is 0011.

Hence, $(0.1)_{10} = (.0\overline{0011})_2$.

Hence, $(0.1)_{10}$ converts to a non-terminating binary fraction.

Recall: In the introduction section (pg 1) we said that 0.1 cannot be represented exactly on a computer. This is why.

E.g. 8 Convert $(72.6)_{10}$ to base 3

Soln:

To solve this, we need to split the integral and fraction parts.

$(72)_{10}$ to base 3:

Numerator	Denominator	Quotient	Remainder
72	3	24	0
24	3	8	0
8	3	2	2
2	3	0	2

$(72)_{10} = (2200)_3$

$(.6)_{10}$ to base 3:

Multiplier	Base	Product	Integral	Fraction
0.6	3	1.8	1	0.8
0.8	3	2.4	2	0.4
0.4	3	1.2	1	0.2
0.2	3	0.6	0	0.6

I will stop here as it will repeat.

$(0.6)_{10} = (.\overline{1210})_3$

$(72.6)_{10} = (2200.\overline{1210})_3$

E.g. 9 Convert $(86.4)_{10}$ to base 4

Soln:

$(86)_{10}$ to base 4:

Numerator	Denominator	Quotient	Remainder
86	4	21	2
21	4	5	1
5	4	1	1
1	4	0	1

$(86)_{10} = (1112)_4$

$(.4)_{10}$ to base 4:

Multiplier	Base	Product	Integral	Fraction
0.4	4	1.6	1	0.6
0.6	4	2.4	2	0.4

I will stop here as it is clear that it will repeat.

$(0.4)_{10} = (.\overline{12})_4$

Hence, $(86.4)_{10} = (1112.\overline{12})_4$

4. Machine Representation of Reals:

- Real numbers are represented in computers as **Floating Point Numbers (FPN)**.

- A FPN x in base b has the form:

$x = (F)_b \cdot b^{(e)_b}$, where F is the fraction part and has the form $F = \pm (.\, d_1 d_2 \ldots d_t)_b$ and e is the exponent and has the form $e = \pm (c_s c_{s-1} \ldots c_1)_b$.

I.e.
$F = \pm (.\, d_1 d_2 \ldots d_t)_b$ is the **fraction part**.
$e = \pm (c_s c_{s-1} \ldots c_1)_b$ is the **exponent**.

Note: Another term for F is the **mantissa**.

- On a 64-bit computer, 1 bit is used for the sign, 11 bits are used for the exponent and 52 bits are used for the mantissa.

- A FPN is **normalized** if $d_1 \neq 0$ unless $d_1 = d_2 = \ldots = d_t = 0$.
I.e. A FPN is normalized if $d_1 \neq 0$ unless $d_i = 0$ $\forall i = 0, 1, \ldots, t$.

- There's 2 reasons why we want normalized FPNs:
1. Uniqueness
2. Storage efficiency / Better storage

E.g. 10 Suppose we are using a 32-bit computer where 1 bit is used for the sign, 8 bits are used for the exponent and 23 bits are used for the mantissa. Represent $(-53.5)_{10}$ in base 2 and write the FPN equivalent.

Soln:

Converting $(53.5)_{10}$ to base 2:

a) Converting $(53)_{10}$ to base 2:

Numerator	Denominator	Quotient	Remainder
53	2	26	1
26	2	13	0
13	2	6	1
6	2	3	0
3	2	1	1
1	2	0	1

Hence $(53)_{10} = 110101$

b) Converting $(0.5)_{10}$ to base 2:

Multiplier	Base	Product	Integral	Fraction
0.5	2	1.0	1	0

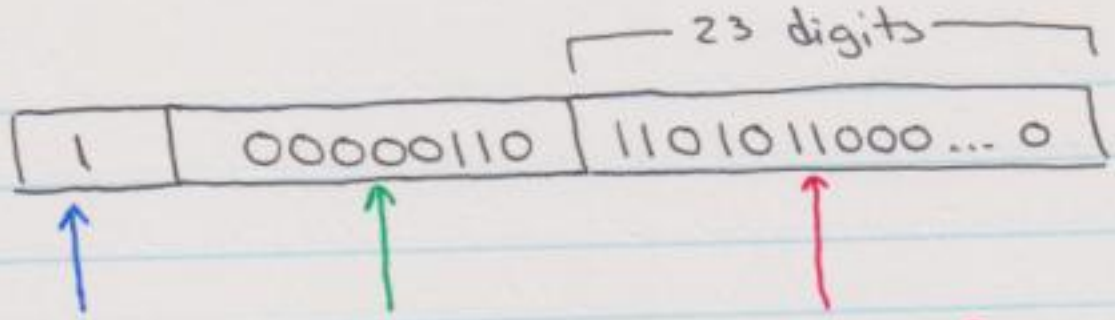
Hence, $(0.5)_{10} = (0.1)_2$.

$\therefore (53.5)_{10} = (110101.1)_2 \rightarrow (-53.5)_{10} = (-110101.1)_2$

Writing the FPN equivalent of $(-110101.1)_2$:

$(-110101.1)_2 \rightarrow (-0.1101011)_2 \cdot 2^{(110)_2}$

Note: We had to shift the decimal point 6 spots to the left, so we multiply (-0.1101011) by 2^6, but 6 is in base 10, so we have to convert it to base 2, which is 110.



- Significant digits of a non-zero FPN are the digits following and including the first non-zero digit.
Note: In a normalized FPN, all digits of the mantissa are significant.

- For the purposes of our course, the absolute value of the mantissa is always fractional.
I.e. $0 \leq |\text{mantissa}| < 1$

- The length of the exponent is also limited. It can only have s digits at most.

- The value of the exponent is bound by $-M$ and M where $M = \underbrace{(aa\ldots a)_b}_{s}$ where $a = b-1$.

I.e. $-M \leq e \leq M$ where $M = \underbrace{(aa\ldots a)_b}_{s}$ and $a = b-1$.

Consider example 10. In that example, $b=2$ and $s=8$.
In that example, $e \leq 11111111$ and $-11111111 \leq e$.
I.e. $-11111111 \leq e \leq 11111111$

- The absolute value of the largest FPN defined by this system is

$$(aa\ldots a)_b \cdot b^{\overbrace{(aa\ldots a)_b}^{s}}$$

(a repeated t times)

where $a = b-1$.

Suppose $b=2$, $s=3$, $t=4$. The largest FPN is

$$(1111)_2 \cdot 2^{(111)_2}.$$

- The smallest non-zero, normalized FPN is

$$(.10\ldots0)_b \cdot b^{-\overbrace{(a\ldots a)_b}^{s}}$$

(t digits)

- The smallest non-zero FPN is $(.00\ldots1)_b \cdot b^{-\overbrace{(a\ldots a)_b}^{s}}$.

(t digits)

Non-normalized FPNs allow us to get very close to 0.

- $R_b(t,s)$ denotes the set of all FPN with base b, t digit mantissa and s digit exponent.

- Overflow or underflow occurs whenever a non-zero FPN with abs value outside the ranges must be stored on a computer.

When the number gets too close to 0, it's underflowing. When the number gets too large, in the positive or negative direction, it's overflowing.

E.g. 11 Suppose we have 8-bit signed ints. The range of representable ints start at -128 and ends at 127.

If we do 127+1, it causes an overflow as the value is outside of the given range.

Likewise, if we do -128-1, it causes an overflow.

Now, suppose that the exponent part can represent values from -127 to 127. Then, any number with abs value less than 2^{-127} may cause underflow.

- The number of normalized FPN is $2(B-1)B^{t-1}(U-L+1)+1$ where
  - The 2 is used for the 2 possible signs.
  - The $(B-1)$ is used to represent the number of possibile values the first digit can have. Recall in a normalized FPN, the leading digit can't be 0.
  - The B^{t-1} is used to represent the number of possible values each digit after the first can have.
  - The $U-L+1$ is used to represent the number of values the exponent can have. Note: $U=M$, $L=-M$
  - The +1 is used in case the number is 0.

- The smallest positive normalized FPN, also called the underflow level, is B^{-M}, which has a 1 for the first digit of the mantissa and 0 elsewhere. Earlier, we defined this to be $(.10\ldots0)_b \cdot b^{-(a\ldots0)}$.
This is equal to $1 \times b^{-(a\ldots0)}$
$= b^{-(a\ldots0)}$
$= b^{-M}$

- The largest FPN, also known as the **overflow level**, is $B^{M+1}(1-B^{-t})$. Earlier, we defined this to be $(.a_1 \ldots a)_b \cdot b^{(a_1 \ldots a)_b}$.

- $R_b(t,s)$ is finite while $\mathbb{R}$ is infinite. Furthermore, $\mathbb{R}$ is compact while $R_b(t,s)$ isn't. This means that between any 2 real numbers, there is an infinite number of reals in between.

- A real number $x = \pm (X_I . X_F)_b$
$= \pm (d_n d_{n-1} \ldots d_0 . d_{-1} d_{-2} \ldots)_b$

can be represented in $R_b(t,s)$ by the following algo:

1. **Normalize the mantissa:**
   - Shift the decimal point to the left of d_n.
   - $x = \pm (d_n d_{n-1} \ldots d_0 . d_{-1} \ldots)_b \rightarrow \pm (. D_1 D_2 \ldots)_b \cdot b^{n+1}$

2. **Round or Chop off the Mantissa:**
   a) Chopping: Chops after t digits of the mantissa.
   b) Rounding: Chop off after digit t then round D_t up if $D_{t+1} \geq b/2$ and down if $D_{t+1} < b/2$.

Note: Another, possibly more efficient, technique of rounding is to add $b/2$ to digit D_{t+1} and then chop after D_t.

We convert x to FPN with the following notation:
$x \in \mathbb{R} \rightarrow FL(x) \in R_b(t,s)$.

E.g. Consider $FL(2/3) \in R_{10}(2,4)$.

It equals $\begin{cases} 0.66, & \text{if chopped} \\ 0.67, & \text{if rounded} \end{cases}$

- Round off error is the difference between $x \in \mathbb{R}$ and $FL(x) \in R_b(t,s)$.

It is usually measured relative to x as $\frac{x - FL(x)}{x} = d$

or $FL(x) = x(1-d)$ where d is the relative round off.

Absolute Error (AE) $= x - FL(x)$
Relative Error (RE) $= \frac{x - FL(x)}{x}$

We can bound d independently of x.
$d < b^{1-t}$ for chopped, normalized FPNs.
$|d| < \frac{b^{1-t}}{2}$ for rounding normalized FPNs.

5. Machine Arithmetic:

- Let $x, y \in R$ and $FL(x), FL(y) \in R_b(t,s)$.
Consider $o \in \{+, -, \times, /\}$. I.e. o is an operation.
$x o y \approx FL(FL(x) o FL(y))$

E.g. 12 In $R_{10}(2,4)$, let $x = 2$ and $y = 0.0000058$.
Suppose we do $x+y$.

$$x+y = FL(FL(x) + FL(y))$$
$$= FL(0.20 \cdot 10^1 + 0.58 \cdot 10^{-5})$$
$$= FL(0.20000058 \cdot 10^1)$$
$$= 0.20 \cdot 10^1$$

6. Machine Precision / Machine Epsilon:

- Defined as the smallest non-normalized FPN eps s.t. $1 + eps > 1$. Eps is referred to as machine epsilon.

- $eps = \begin{cases} b^{1-t}, & \text{if chopping} \\ \frac{b^{1-t}}{2}, & \text{if rounding} \end{cases}$

Hence, $0 \leq d \leq eps$ for chopping and
$|d| \leq eps$ for rounding

E.g. 13
Let $o \in \{+, -, \times, /\}$
Let $x, y \in \mathbb{R}$ and $FL(x) = x(1-d)$, $FL(y) = y(1-d)$
Find/Calculate the error bound for $x \cdot y$.

Soln:

$$\begin{aligned} x \cdot y &= FL(FL(x) \cdot FL(y)) \\ &= FL(x(1-dx) \cdot y(1-dy)) \\ &= x(1-dx) \cdot y(1-dy) \cdot (1-dxy) \\ &= xy\,((1-dx)(1-dy)(1-dxy)) \\ &= xy\,((1-dx-dy+dxdy)(1-dxy)) \\ &= xy\,(1-dxy-dx+dxdxy-dy+dydxy+dxdy-dxdydxy) \\ &\approx xy(1-dx-dy-dxy) \\ &= xy(1-d.) \end{aligned}$$

Since $|dx| \leq eps$ and $|dy| \leq eps$ and $|dxy| \leq eps$,
$|d.| \leq 3 \cdot eps$. Note that this is the worst case scenario.

This is a good approximation because since each d_i is a small fraction, multiplying them with each other will result in values so small, they are much lower than eps. Hence, we can remove all instances of 2 or more d_i's times each other.

E.g. 14

Wt $x, y \in \mathbb{R}$ and $FL(x) = x(1-d)$, $FL(y) = y(1-d)$.
Calculate the error bound for $x+y$.

Soln:

$$\begin{aligned} x+y &= FL(FL(x) + FL(y)) \\ &= [x(1-dx) + y(1-dy)](1-dxy) \\ &= x(1-dx)(1-dxy) + y(1-dy)(1-dxy) \\ &= x(1-dxy-dx+dxdxy) + y(1-dxy-dy+dydxy) \\ &\approx x(1-dxy-dx) + y(1-dxy-dy) \\ &= (x+y)\left(1 - \frac{x(dx+dxy)}{x+y} - \frac{y(dy+dxy)}{x+y}\right) \\ &= (x+y)(1-d+) \end{aligned}$$

$$|d+| \leq \left|\frac{x}{x+y}\right| \cdot 2eps + \left|\frac{y}{x+y}\right| \cdot 2eps$$

$$= \frac{|x| + |y|}{|x+y|} \cdot 2eps$$

$$|d+| \leq \begin{cases} 2eps, & \text{when } x \text{ and } y \text{ have the same sign} \\ 2eps \cdot \frac{|x-y|}{|x+y|}, & \text{when } x \text{ and } y \text{ have the opposite signs.} \end{cases}$$

Note: As x approaches $-y$, the error bound approaches infinity. This is called **subtractive cancellation.**

E.g. 15 Consider $R_{10}(3,1)$ with rounding.
Compute $a^2 - 2ab + b^2$ with $a = 15.6$ and $b = 15.7$.

Soln:

$FL(a) = 0.156 \cdot 10^2$

$FL(b) = 0.157 \cdot 10^2$

$FL(a^2) = FL(243.36) = +(0.243 \cdot 10^3)$
$FL(2ab) = FL(489.84) = +(0.490 \cdot 10^3)$
$FL(b^2) = FL(246.49) = +(0.246 \cdot 10^3)$

$$\begin{aligned} FL(a^2 - 2ab + b^2) &= FL(243 - 490 + 246) \\ &= FL(-1) \\ &= -(0.100 \cdot 10^1) \end{aligned}$$

This is an issue because $a^2 - 2ab + b^2 = (a-b)^2$, which is positive. This is an example of subtractive cancellation.

7. Stability of Formulae:

- Used for algorithms.
- An algorithm is **stable** if the result it produces is relatively insensitive to perturbations resulting from approximations made during the computation.
- One way to deal with unstable algorithms is to use a different, more stable algorithm.

E.g. 16 Consider $1 - \cos x$.
When x approaches 0, $\cos x$ approaches 1 and the formula suffers from subtractive cancellation.
We can use an alternative formula.

$$1 - \cos x \left(\frac{1 + \cos x}{1 + \cos x} \right)$$

$$= \frac{1 - \cos^2 x}{1 + \cos x}$$

$$= \frac{\sin^2 x}{1+\cos x}$$

This new formula is fine when x approaches 0 but not fine when x approaches π as $\cos(\pi) = -1$.
The original formula is fine when x approaches π.
Hence, depending on the value of x, choose a formula that doesn't suffer from subtractive cancellation.

$$\text{use} \begin{cases} 1-\cos x, & \text{if } x \text{ approaches } \pi \\ \frac{\sin^2 x}{1+\cos x}, & \text{if } x \text{ approaches } 0 \end{cases}$$

8. Condition of Functions:

- Used for functions.
- A function is **well-conditioned** if a given relative change in the input data causes a reasonably proportinate relative change in the solution.
- A function is **ill-conditioned** if the relative change in the solution can be much larger than that in the input data.
- The conditiong of a function can be calculated as

$$\text{Cond}(f) = \frac{|\text{relative change in soln}|}{|\text{relative change in input data}|}$$

$$\simeq \frac{|(f(\hat{x}) - f(x))/f(x)|}{|(\hat{x}-x)/x|}, \text{ where } \hat{x} \text{ is a point near } x$$

$$\simeq \frac{|x f'(\hat{x})|}{|f(x)|}$$

$$\approx \frac{|x f'(x)|}{|f(x)|}$$

To calculate the conditioning of a function, we'll use

$$\text{cond}(f) = \frac{|x f'(x)|}{|f(x)|}$$

- A function, f, is ill-conditioned if cond(f) is much larger than 1.

E.g. 17 Let $f(x) = \sqrt{x}$. Calculate cond(f).

Soln:

$$\text{cond}(f) = \frac{|x f'(x)|}{|f(x)|}$$

$$= \frac{|x (\sqrt{x})'|}{|\sqrt{x}|}$$

$$= \frac{|x|}{|2(\sqrt{x})(\sqrt{x})|}$$

$$= \frac{|x|}{|2x|}$$

$$= \frac{1}{2} \leftarrow \text{well conditioned}$$

E.g. 18 Let $f(x) = \frac{10}{1-x^2}$. Calculate cond(f).

Soln:

$$\text{cond}(f) = \frac{|x f'(x)|}{|f(x)|}$$

$$= \frac{\left| x \cdot \frac{20x}{(1-x^2)^2} \right|}{\left| \frac{10}{1-x^2} \right|}$$

$$= \frac{|2x^2|}{|1-x^2|} \leftarrow \text{Ill conditioned when } |x| \approx 1$$

- If algo A is stable, then function f is well-defined. However, if f is well-defined, there's a chance we can't find a stable algo to compute it.

FPN

CSCC37 Examples

1. Convert $(101110)_2$ to decimal.

Soln:

$1\times2^5+0\times2^4+1\times2^3+1\times2^2+1\times2^1+0\times2^0$
$=32+8+4+2$
$=(46)_{10}$

2. Convert $(46)_{10}$ to binary.

Soln:

Numerator	Denominator	Quotient	Remainder
46	2	23	0
23	2	11	1
11	2	5	1
5	2	2	1
2	2	1	0
1	2	0	1

Reading the remainder col from bottom to top, we get 101110.

3. Convert $(0.5)_{10}$ to binary.

Soln:

Multiplier	Base	Product	Integral	Fraction
0.5	2	1.0	1	0

Hence, $(0.5)_{10} = (0.1)_2$.

4. Convert $(0.75)_{10}$ to binary.

Soln:

Multiplier	Base	Product	Integral	Fraction
0.75	2	1.5	1.0	0.5
0.5	2	1.0	1.0	0

We read the integral col top to bottom.
Hence, $(0.75)_{10} = (0.11)_2$.

5. Convert $(5.875)_{10}$ to binary.

Soln:
We need to split 5.875 into 5 and 0.875 and convert each part individually and then combine the results.

Converting $(5)_{10}$ to binary:

Numerator	Denominator	Quotient	Remainder
5	2	2	1
2	2	1	0
1	2	0	1

Hence, $(5)_{10} = (101)_2$

Converting $(0.875)_{10}$ to binary:

Multiplier	Base	Product	Integral	Fraction
0.875	2	1.75	1	0.75
0.75	2	1.5	1	0.5
0.5	2	1.0	1	0

Hence, $(0.875)_{10} = (.111)_2$

Putting it all together, $(5.875)_{10} = (101.111)_2$.

6. Convert $(10.125)_{10}$ to binary.

Soln:

Converting $(10)_{10}$ to binary

Numerator	Denominator	Quotient	Remainder
10	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

Hence, $(10)_{10} = (1010)_2$.

Converting $(.125)_{10}$ to binary

Multiplier	Base	Product	Integral	Fraction
0.125	2	0.25	0	0.25
0.25	2	0.5	0	0.5
0.5	2	1.0	1	0

Hence, $(0.125)_{10} = (0.001)_2$

Putting it together, $(10.125)_{10} = (1010.001)_2$.

CSCC37 Linear Systems Notes

1. Linear Algebra Review:

a) Terminology:

Let A be a mxn matrix:

- This means that A has m rows and n colns.
- If m=n, then A is a **square matrix**.

Let B be a nxn matrix:

- B is a square matrix.
- B is said to be **singular** if it has 1 of the following equivalent properties:

1. B has no inverse.
2. $\det(B) = 0$
3. $B\bar{z} = \bar{0}$ for some vector $\bar{z} \neq \bar{0}$

Otherwise, B is **non-singular**. If B is non-singular, then B^{-1} exists and the system $B\bar{x} = \bar{b}$ always has a unique soln $\bar{x} = B^{-1}\bar{b}$ regardless of the value of $\bar{b}$. If B is singular, it will either have no solns or infinitely many solns.

- The **main diagonal** of B is the values $B_{11}, B_{22}, \ldots, B_{nn}$.

E.g. Let $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

The main diagonal is circled in red.

General Terminology:

- The **transpose** of matrix $A_{m,n}$, denoted as A^T, is created when you switch the row and coln indices of each element in A.

E.g. $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ $A^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$

If A is a mxn matrix, A^T is a nxm matrix

To create A^T from A, write the rows of A as the cols of A^T.

A square matrix whose transpose is equal to itself is a **symmetric matrix.**
I.e. If $A^T = A$, then A is a symmetric matrix.

A square matrix whose transpose is equal to its negative is a **skew-symmetric matrix.**
I.e. If $A^T = -A$, then A is a skew-symmetric matrix.

- The **identity matrix**, denoted as I, is a square matrix with 1's along the main diagonal and 0 elsewhere.
E.g. $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ are identity matrices.

Note: The identity matrix is a symmetric matrix.
Note: The product of 2 inverse matrices is always the identity matrix.
I.e Let $B = A^{-1}$. Then, $AB = BA = I$

- A **lower triangular matrix** is a square matrix if all entries above the main diagonal is 0.
E.g. $A = \begin{bmatrix} 1 & 0 \\ -3 & 2 \end{bmatrix}$ is a lower triangular matrix.

- An **upper triangular matrix** is a square matrix if all entries below the main diagonal is 0.
E.g. $A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$ is an upper triangular matrix.

- A **permutation matrix**, denoted as P, is a square matrix having exactly one 1 in each row and coln and 0 elsewhere. It is used if you want to swap 2 rows.

E.g. Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Suppose you want to swap rows 1 and 2.
Your permutation matrix, P, would be $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

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

E.g. Let $B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Suppose you want to swap rows 1 and 3.
Your permutation matrix P would be $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$

Now suppose you want to swap rows 2 and 3 of the original matrix. Your P would be $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{bmatrix}$$

Suppose you want to swap rows i and j, $i \neq j$.
To create/determine your permutation matrix, start with the identity matrix. Then, move the 1 at position (i,i) to (i,j) and move the 1 at position (j,j) to position (j,i).

In the first example, when I wanted to swap rows 1 and 2, i=1 and j=2. The 1 at (1,1) got moved to (1,2) and the 1 at (2,2) got moved to (2,1).

b) Calculations:
Matrix Addition and Subtraction:
- Two matrices, A and B can only be added or subtracted if they have the same number of rows and colns.
- $A \pm B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \pm \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \dots & b_{mn} \end{bmatrix}$

E.g. 1 Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 3 & 5 \\ 1 & 5 \end{bmatrix}$

Find A+B and A-B

Soln:

$A+B = \begin{bmatrix} 1+3 & 2+5 \\ 3+1 & 4+5 \end{bmatrix} = \begin{bmatrix} 4 & 7 \\ 4 & 9 \end{bmatrix}$

$A-B = \begin{bmatrix} 1-3 & 2-5 \\ 3-1 & 4-5 \end{bmatrix} = \begin{bmatrix} -2 & -3 \\ 2 & -1 \end{bmatrix}$

Matrix Multiplication:

- We can only multiply 2 matrices, A and B, if the number of cols of A = the num of rows of B. The resulting matrix will have the same number of rows as A and the same number of cols as B.

E.g. 2 Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, $B = \begin{bmatrix} 3 & 1 & 2 \\ 5 & 2 & 3 \end{bmatrix}$

Find $A \times B$

Soln:

$$A \times B = \begin{bmatrix} (1)(3)+(2)(5) & (1)(1)+(2)(2) & (1)(2)+(2)(3) \\ (3)(3)+(4)(5) & (3)(1)+(4)(2) & (3)(2)+(4)(3) \\ (5)(3)+(6)(5) & (5)(1)+(6)(2) & (5)(2)+(6)(3) \end{bmatrix}$$

$$= \begin{bmatrix} 13 & 5 & 8 \\ 29 & 11 & 18 \\ \uparrow & 17 & 28 \\ 45 & & \end{bmatrix}$$

2. Linear Systems:

- $A\bar{x} = \bar{b}$, where $A \in R^{n \times n}$, $\bar{x}, \bar{b} \in R^n$

We're given A and $\bar{b}$ and have to solve for $\bar{x}$.

- General Soln Technique:

1. Reduce the problem to an equivalent one that's easier to solve.
2. Solve the reduced problem.

E.g. 3 Solve $3x_1 - 5x_2 = 1$
$6x_1 - 7x_2 = 5$

Soln:

$$\begin{bmatrix} 3 & -5 \\ 6 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{matrix} r_1 \\ r_2 \end{matrix}$$

Do $r_2 - 2r_1$

$$\begin{bmatrix} 3 & -5 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$3x_2 = 3 \Rightarrow x_2 = 1$
$6x_1 - 7x_2 = 5$
$6x_1 - 7 = 5$
$6x_1 = 12$
$x_1 = 2$

$x_1 = 2, \; x_2 = 1$

Now, we'll generalize this to n unknowns and n eqns.
Let matrix $A = [a_{ij}]$ where a_{11} is the top left element.

Eqn 1: $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$
Eqn 2: $a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$
$\vdots$
Eqn n: $a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$

We will use the following steps to reduce this system to triangular form.

Recall: An upper triangular matrix is a square matrix with 0's below the main diagonal.

Step 1:

- Assume that $a_{11} \neq 0$
- Multiply Eqn 1 by $\frac{a_{21}}{a_{11}}$ and subtract from Eqn 2.
- Multiply Eqn 1 by $\frac{a_{31}}{a_{11}}$ and subtract from Eqn 3.
- Repeat for all remaining rows.

I.e. Multiply Eqn 1 by $\frac{a_{i1}}{a_{11}}$ and subtract from Eqn i

where $4 \leq i \leq n$.

- We now have an equivalent system where X_1 has been eliminated from eqns 2 to n.

I.e Now we have:

$$\text{Eqn } 1: \quad a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$
$$\text{Eqn } \hat{2}: \quad 0 + \hat{a}_{22}x_2 + \dots + \hat{a}_{2n}x_n = \hat{b}_2$$
$$\vdots \qquad \vdots$$
$$\text{Eqn } \hat{n}: \quad 0 + \hat{a}_{n2}x_2 + \dots + \hat{a}_{nn}x_n = \hat{b}_n$$

Note: The small hat, ^, is used to note that these values changed.

Step 2:

- Assume that $\hat{a}_{22} \neq 0$.
- Multiply eqn 2 by $\frac{\hat{a}_{32}}{\hat{a}_{22}}$ and subtract from Eqn 3.
- Repeat for all remaining rows.
- We now have an equivalent system where X_2 has been eliminated from eqns 3 to n.

Repeat this pattern up to and including eqn n-1. Afterwards, we will have an upper triangular system.

I.e.

Eqn 1: $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$

Eqn $\hat{2}$: $0 + \hat{a}_{22}x_2 + \dots + \hat{a}_{2n}x_n = \hat{b}_2$

$\vdots$ $\quad\vdots\quad$ $\vdots$

Eqn $\tilde{n}$: $0 + 0 + \dots + \tilde{a}_{nn}x_n = \tilde{b}_n$

- Another way of looking at this is using vector notation. ($A\bar{x} = \bar{b}$)

Step 1: Eliminate the first coln of A using a_{11}.

$L_1 A\bar{x} = L_1\bar{b}$, where $L_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & & \\ \vdots & & \ddots & \\ -\frac{a_{n1}}{a_{11}} & & & 1 \end{bmatrix}$

Note: L_1 is very similar to the identity matrix except the first coln is filled with multipliers used in the first step.

Step 2: Eliminate the second coln of L_1A using $\hat{a}_{22}$.

$L_2(L_1A)\bar{x} = L_2(L_1\bar{b})$ where $L_2 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & -\frac{\hat{a}_{32}}{\hat{a}_{22}} & & \\ 0 & \vdots & & 1 \\ & -\frac{\hat{a}_{n2}}{\hat{a}_{22}} & & \end{bmatrix}$

We continue until we have $L_{n-1}L_{n-2}\dots L_2L_1A\bar{x} = L_{n-1}L_{n-2}\dots L_2L_1\bar{b}$.

We let $L_{n-1}L_{n-2}\dots L_1A = U$ where U is an upper triangular matrix. This becomes very easy to solve.

E.g. 4 Solve the following system of eqns using the technique we just learned.

$$2x_1 + 4x_2 - 2x_3 = 2$$
$$4x_1 + 9x_2 - 3x_3 = 8$$
$$-2x_1 - 3x_2 + 7x_3 = 10$$

Soln:

$$\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ \uparrow \\ 10 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$L_1(A) = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

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

$$L_1\bar{B} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 8 \\ \uparrow \\ 10 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

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

$$L_2(L_1 \bar{B}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}$$

We now have

$$\begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}$$

This makes finding $\bar{x}$ much easier.

$$\bar{x} = \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$$

3. LU Factorization:

- We have $L_{n-1} L_{n-2} \dots L_1 A = U \leftrightarrow A = \overbrace{L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}}^{L} U$

Lemma 1: If L_i is a Gauss Transformation, then L_i^{-1} exists and is also a Gauss Transformation.

Lemma 2: If L_i and L_j are Gauss Transformations and $j > i$, then $L_i L_j = L_i + L_j - I$

- $A = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} U \leftrightarrow A = LU$
- Using $A = LU$ to solve $A\bar{x} = \bar{b}$, we can convert $A\bar{x} = b$ into $(LU)\bar{x} = \bar{b}$. Then, let $\bar{d} = U\bar{x}$ where $L\bar{d}$ is a lower triangular matrix. We now have $L\bar{d} = \bar{b}$.

I.e.

$A\bar{x} = \bar{b}$

$\leftrightarrow LU\bar{x} = \bar{b}$

$\leftrightarrow L\bar{d} = \bar{b}$ where $\bar{d} = U\bar{x}$.

while $L\bar{d}$ is a lower triangular matrix and $U\bar{x}$ is an upper triangular matrix.

- We use LU factorization because if we have the same coefficient matrix A but different RHS, we can use the same LU.

Eg. 5

Let $A = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -2 & 4 & 1 \end{bmatrix}$

Find L_1 and L_2

Soln:

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -2 & 4 & 1 \end{bmatrix}$$

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

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 1 \\ 0 & 3 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix} \leftarrow U$$

Recall: $L_{n-1} L_{n-2} \dots L_1 A = U$

Now, let's show Lemma 2.

Recall that $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$ and that $L_i L_j = L_i + L_j - I$, $j > i$.

To compute L_i^{-1}, simply take L_i and toggle/switch the sign of the multipliers.

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \quad L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \rightarrow \qquad L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$L = L_1^{-1} \cdot L_2^{-1}$$

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

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

Another way to compute $L_1^{-1} \cdot L_2^{-1}$ is $L_1^{-1} + L_2^{-1} - I$

$$L_1^{-1} + L_2^{-1} - I = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$ Same as above

Now that we have L and U, lets see what LU is.

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

↑ A

E.g. 6 Given $A = \begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix}$

find L and U.

Soln:

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3/4 & 0 & 1 \end{bmatrix}$$

$$L_1(A) = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -3/4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 2 & 9 \\ 0 & 8 & -1/2 \\ 0 & \frac{22}{4} & 9/4 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{11}{16} & 1 \end{bmatrix}$$

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{11}{16} & 1 \end{bmatrix} \begin{bmatrix} 8 & 2 & 9 \\ 0 & 8 & -1/2 \\ 0 & \frac{22}{4} & 9/4 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 2 & 9 \\ 0 & 8 & -1/2 \\ 0 & 0 & \frac{83}{32} \end{bmatrix} \leftarrow U$$

$$L = L_1^{-1} + L_2^{-1} - I$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/4 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 11/16 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/4 & 11/16 & 1 \end{bmatrix}$$

4. GE with Pivoting:

- If you go back to page 7, you'll see that we have "Assume that $a_{11} \neq 0$" and "Assume that $\hat{a}_{22} \neq 0$." But what happens if at some step, $a_{ii} = 0$?

E.g. $A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 3 \end{bmatrix}$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 3 \\ 0 & 1 & 2 \end{bmatrix}$$ ← Notice that we can no longer divide by $\hat{a}_{22}$ as it is 0.

A possible Soln, in this case, is to swap rows 2 and 3.

We now have $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{bmatrix}$

- In general, if $a_{ii} = 0$, go down coln i, starting from row i+1, and find a suitable row to swap with row i.
Note: If we want to find a row to swap with row i, we can only choose rows that are below row i. That is, we can only choose row j to swap with row i if $j > i$.

- A similar problem arises/occurs if one of the elements along the main diagonal is very small.
This is a more common scenario than the previous one.

E.g. Let $LA = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 10^{-16} & 3 \\ 0 & 1 & 2 \end{bmatrix}$

$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -10^{16} & 1 \end{bmatrix}$

$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -10^{16} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 10^{-16} & 3 \\ 0 & 1 & 2 \end{bmatrix}$

$= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 10^{-16} & 3 \\ 0 & 0 & 2 - 3 \cdot 10^{16} \end{bmatrix}$

This term will cause a lot of issues.

A possible soln is to go down the coln, starting from the very small element, and to swap rows where the second row has a bigger element in that coln. This is called **partial row pivoting**.
For this example, swap rows 2 and 3.

E.g. 7 Solve $\begin{bmatrix} 2 & 6 & 6 \\ 3 & 5 & 12 \\ 6 & 6 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$

using row pivoting.

Soln:

Step 1: We want the biggest value in col 1 to be on the main diagonal. Hence, we swap row 1 and 3.

$P_1 A\bar{x} = P_1 \bar{b}$, where $P_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ **Recall:** P_1 is a permutation matrix.

$P_1 A = \begin{bmatrix} 6 & 6 & 12 \\ 3 & 5 & 12 \\ 2 & 6 & 6 \end{bmatrix}$ $\quad P_1 \bar{b} = \begin{bmatrix} 30 \\ 25 \\ 20 \end{bmatrix}$

Step 2: Find L_1

$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/3 & 0 & 1 \end{bmatrix}$

$L_1 (P_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 3 & 5 & 12 \\ 2 & 6 & 6 \end{bmatrix}$

$= \begin{bmatrix} 6 & 6 & 12 \\ 0 & 2 & 6 \\ 0 & 4 & 2 \end{bmatrix}$

$L_1 (P_1 \bar{b}) = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 25 \\ 20 \end{bmatrix} = \begin{bmatrix} 30 \\ 10 \\ 10 \end{bmatrix}$

Step 3: We want the biggest value in col 2, starting, from row 2 to be on the main diagonal. Hence, we swap rows 2 and 3.

$P_2 L_1 P_1 A\bar{x} = P_2 L_1 P_1 \bar{b}$, where $P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$P_2 L_1 P_1 A = \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}$

$P_2 L_1 P_1 \bar{b} = \begin{bmatrix} 30 \\ 10 \\ 10 \end{bmatrix}$

Step 4: Find L_2

$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/2 & 1 \end{bmatrix}$

$L_2 P_2 L_1 P_1 A = \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix}$

$L_2 P_2 L_1 P_1 \bar{b} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$

Now, we have $\begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$ $\bar{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

- We have $L_2 P_2 L_1 P_1 A = U$.

$$L_2 P_2 L_1 P_1 A \longleftrightarrow L_2 P_2 L_1 P_2 P_2 P_1 A \quad (1)$$
$$\longleftrightarrow L_2 \hat{L}_1 P_2 P_1 A \quad (2)$$

Note: The inverse of a permutation matrix is itself. Hence, that's why we can multiply $L_2 P_2 L_1 P_1 A$ by $P_2 \cdot P_2$ in (1).

Note: $\hat{L}_1 = P_2 L_1 P_2$ is a modified Gauss Transformation.

E.g. 8 Let $P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ and $L_1 = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & 0 & 1 \end{bmatrix}$

Solve $P_2 L_1 P_2$

Soln:

$$P_2 L_1 P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ \ell_{31} & 0 & 1 \\ \ell_{21} & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ \ell_{31} & 1 & 0 \\ \ell_{21} & 0 & 1 \end{bmatrix}$$

Note: When you pre-multiply a matrix with a permutation matrix, you switch 2 rows. However, when you post-multiply a matrix with a permutation matrix, you switch 2 columns.

I.e. $PL \rightarrow$ Changes 2 rows of L.
$LP \rightarrow$ Changes 2 cols of L.

Hence, $\hat{L}_1$ is L_1 with its 2 multipliers swapped.

Now we have $L_2 \hat{L}_1 P_2 P_1 A = U$

$\Leftrightarrow P_2 P_1 A = \hat{L}_1^{-1} \hat{L}_2^{-1} U$

$\Leftrightarrow PA = LU$

where $P = P_2 P_1$ and $L = \hat{L}_1^{-1} L_2^{-1}$

- Now, we have to solve $A\bar{x} = \bar{b}$ given $PA = LU$.

$A\bar{x} = \bar{b}$

$\Leftrightarrow PA\bar{x} = P\bar{b}$

$\Leftrightarrow LU\bar{x} = \hat{b}$ where $\hat{b} = P\bar{b}$

Let $\bar{d} = U\bar{x}$.

Hence, we solve:

1. $L\bar{d} = \hat{b}$ for $\bar{d}$ (Forward Solve)
2. $U\bar{x} = \bar{d}$ for $\bar{x}$ (Backward Solve)

- What happens if at the k^{th} step, one element along the main diagonal, a_{kk}, and everything below it is 0?

I.e. $L_{k-1} \dots L_1 A =$

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

$\uparrow k^{th}$ column

Remember that our goal is to make every element in the k^{th} coln under k 0, so we just continue.

However, this will result in a matrix U with a 0 for one of the entries along the main diagonal. Then, we will have a singular matrix U. If U is singular, when we do $U\bar{x} = \bar{d}$, we could have either 0 solns or infinitely many solns

E.g. $U\bar{x} = \bar{d}$, where $U = \begin{bmatrix} 2 & 5 & 4 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix}$

$$\begin{bmatrix} 2 & 5 & 4 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

So, we have $x_3 = d_2$

$2x_3 = d_3$

If d_3 is twice d_2, then x_2 is a free variable and we have infinitely many soln.

If d_3 is not twice d_2, then we have no soln.

Note: While it's possible for U to be a singular matrix, L cannot be a singular matrix.

- Now suppose that at the k^{th} step, if all elements below a_{kk} and the element a_{kk} have a magnitude of $\leq$ eps. $\max_{1 \leq j \leq k} |u_{jj}|$. We call this **numerical singularity** or **near singularity**.

5. Complexity of GE:

- Let A be a nxn matrix.
- We will count additional/multiplication pairs, i.e. mx+b, as **Floating Point operation** or **FLOP**.

a) Computing the complexity of LU Factorization:

- Our first step is zeroing out the first col after a_{11}. Hence, we have $(n-1)^2$ FLOPs.

- Our second step is zeroing out the second col after a_{22}. Hence, we have $(n-2)^2$ FLOPs.

⋮

- Our last step is zeroing out the $(n-1)^{th}$ col after $a_{(n-1)(n-1)}$. Hence, we have $(n-(n-1))^2$ or 1 FLOP.

- In total, we have $(n-1)^2 + (n-2)^2 + \dots + 1$ FLOPs.

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

However, since we go up to $(n-1)^2$, we have $\frac{n(n-1)(2n)}{6}$ or $\frac{n^3}{3} + O(n^2)$ FLOPs for computing the complexity of LU Factorization.

b) Computing the Complexity of Forward and Backward Solve:

- Consider forward solve:

$$\begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & 0 & \\ \ell_{31} & 1 & & \\ \vdots & & \ddots & \\ \ell_{n1} & \cdots & & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}$$

Recall forward solve is $Ld = b$ where L is a lower triangular matrix.

$d_1 = b_1$ ← No flops

$d_2 = b_2 - \ell_{21} d_1$ ← 1 flop

$d_3 = \underbrace{b_3 - \ell_{31} d_1}_{1 \text{ flop}} - \underbrace{\ell_{32} d_2}_{2^{nd} \text{ flop}}$ ← 2 flops

$\vdots$

$d_n = (n-1)$ flops

Hence, we have $0 + 1 + 2 + \ldots + (n-1)$ FLOPS.

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

However, since we go up to $(n-1)$, we get $\frac{n(n-1)}{2}$

or $\frac{n^2}{2} + O(n)$ FLOPs.

- Backward Solve is similar, resulting in another $\frac{n^2}{2} + O(n)$ FLOPs.

- In total, for forward and backward solve, the complexity is $n^2 + O(n)$ FLOPs.

6. Round off Error:

- Recall that we have $PA = LU$ computed in a floating point system (FPS).

- Because of machine round off error, we actually get $\hat{P}(A+E) = \hat{L}\hat{U}$ where $\hat{P}$, $\hat{L}$, $\hat{u}$ are the computed factors and E is the error that occurs during factorization process.

- We hope that E is small relative to A.

- Now, solving $A\bar{x} = \bar{b}$ becomes $(A+E)\hat{\bar{x}} = \bar{b}$ where $\hat{\bar{x}}$ is the computed soln.

- Equivalently, let $E\hat{\bar{x}} = \bar{r}$.
Then, $(A+E)\hat{\bar{x}} = \bar{b}$
$\iff A\hat{\bar{x}} + \bar{r} = \bar{b}$
$\iff \bar{r} = \bar{b} - A\hat{\bar{x}}$, where $\bar{r}$ is the residual. We would like $\bar{r}$ to be $\bar{0}$.

- If we use row partial pivot, we can show that
a) $\|E\| \leq k \cdot eps \cdot \|A\|$ where k is not too large and depends on n.
b) $\|\bar{r}\| \lessapprox k \cdot eps \cdot \|\bar{b}\| \iff \frac{\|\bar{r}\|}{\|\bar{b}\|} \lessapprox k \cdot eps$

$\frac{\|\bar{r}\|}{\|\bar{b}\|}$ is called the **relative residual**.

Note: This does not mean that $\|\bar{x} - \hat{\bar{x}}\|$ or $\frac{\|\bar{x} - \hat{\bar{x}}\|}{\|\bar{x}\|}$ is small.

$\|\bar{x} - \hat{\bar{x}}\|$ is called the **absolute error**.

$\frac{\|\bar{x} - \hat{\bar{x}}\|}{\|\bar{x}\|}$ is called the **relative error**.

$\bar{x}$ is the true solution.

E.g. 9. Given $\begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.656 \end{bmatrix} \bar{x} = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix}$ where

$\bar{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and 2 computed solns $\hat{\bar{x}}_1 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}$

and $\hat{\bar{x}}_2 = \begin{bmatrix} 0.341 \\ -0.087 \end{bmatrix}$, find $\bar{r}_1$ and $\bar{r}_2$.

$\bar{r}_1 = \bar{b} - A\hat{\bar{x}}_1$
$= \begin{bmatrix} -0.001243 \\ -0.001572 \end{bmatrix}$

$\bar{r}_2 = \bar{b} - A\hat{\bar{x}}_2$
$= \begin{bmatrix} -0.000001 \\ 0 \end{bmatrix}$

We see that $\frac{\|\bar{r}_2\|}{\|\bar{b}\|}$ is much smaller than $\frac{\|\bar{r}_1\|}{\|\bar{b}\|}$,
yet we see that $\hat{\bar{x}}_2$ is a terrible soln.

Furthermore, why is $\frac{\|\bar{x} - \hat{\bar{x}}_1\|}{\|\bar{x}\|}$ so much smaller than $\frac{\|\bar{x} - \hat{\bar{x}}_2\|}{\|\bar{x}\|}$?

- We need a relationship between relative error and relative residual.

Note: A small residual does not always mean a small error.

We have 2 eqns to start off:

1. $A\hat{\bar{x}} = \bar{b} - \bar{r}$
2. $A\bar{x} = \bar{b}$

We will now subtract (1) from (2) We get:

$A(\bar{x} - \hat{\bar{x}}) = \bar{r}$ (3)

Rearranging (3) by multiplying both sides by A^{-1} gets us:

$\bar{x} - \hat{\bar{x}} = A^{-1}\bar{r}$ (4)

Taking the norm of both sides of (4) gets us:

$$\|\bar{x} - \hat{\bar{x}}\| = \|A^{-1}\bar{r}\| \leq \|A^{-1}\|\,\|\bar{r}\| \quad (5)$$

Taking the norm of $\bar{b} = A\bar{x}$, we get

$$\|\bar{b}\| = \|A\bar{x}\| \leq \|A\|\,\|\bar{x}\| \quad (6)$$

Combining (5) and (6), we get

$$\frac{\|\bar{x} - \hat{\bar{x}}\|}{\|A\|\,\|\bar{x}\|} \leq \frac{\|A^{-1}\|\,\|\bar{r}\|}{\|\bar{b}\|}$$

$$\underbrace{\frac{\|\bar{x} - \hat{\bar{x}}\|}{\|\bar{x}\|}}_{\text{Relative error}} \leq \frac{\|A\|\,\|A^{-1}\|\,\|\bar{r}\|}{\|\bar{b}\|}$$

$$\text{Relative error} = \underbrace{\|A\|\,\|A^{-1}\|}_{\text{Cond}(A)}\ \underbrace{\frac{\|\bar{r}\|}{\|\bar{b}\|}}_{\text{Relative residue}}$$

Note: $1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\|\,\|A^{-1}\| = \text{Cond}(A)$

Cond(A) $= \|A\|\,\|A^{-1}\|$. Cond(A) ≥ 1 always.

If Cond(A) is very large, the problem is poorly conditioned and small relative residuals do not mean small relative errors.

If Cond(A) is not too large, the problem is well conditioned and a small relative residual is a reliable indicator of small relative error.

Note: Conditioning is a continuous spectrum. How large is "very large" depends on context.

Going back to example 9, $A = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.656 \end{bmatrix}$.

Hence, $A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} 0.656 & -0.563 \\ -0.913 & 0.780 \end{bmatrix}$

$$= 10^6 \begin{bmatrix} 0.656 & -0.563 \\ -0.913 & 0.780 \end{bmatrix}$$

Now, let's find Cond(A).

$\|A\|_\infty = 1.572$

$\|A^{-1}\|_\infty = 1.693 \cdot 10^6$

$\rightarrow$ $\text{Cond}_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty = 2.66 \cdot 10^6$

$\frac{\|\bar{x} - \bar{\hat{x}}\|}{\|\bar{x}\|} \leq 2.66 \times 10^6 \frac{\|\bar{r}\|}{\|\bar{b}\|}$, meaning that the relative error in $\bar{x}$ could be as big as $2.66 \cdot 10^6$ times the relative residual.

This means that A is a poorly conditioned matrix and relative residual is not a reliable indicator of relative error.

7. Iterative Refinement:

- One way to improve $\bar{\hat{x}}$ is to improve the mantissa length.
- Suppose that you've already solved $(A+E)\bar{\hat{x}} = \bar{b}$ and you want to solve $A\bar{x} = \bar{b}$.

$$(A+E)\bar{\hat{x}} = \bar{b} \iff A\bar{\hat{x}} + \bar{r} = \bar{b}$$
$$\iff A\bar{\hat{x}} = \bar{b} - \bar{r}$$

Now we have

$A\bar{x} = \bar{b}$ (1)

$A\bar{\hat{x}} = \bar{b} - \bar{r}$ (2)

If we do (1) - (2), we get $A(\bar{x} - \bar{\hat{x}}) = \bar{r}$.
Let $\bar{z} = \bar{x} - \bar{\hat{x}}$

Now, I'll solve $A\bar{z} = \bar{r}$.
Furthermore, $\bar{x} = \bar{\hat{x}} + \bar{z}$. However, this is a fallacy because we can't get $\bar{z}$. Instead, we get $\bar{\hat{z}}$.

- Algorithim:

1. Compute $\hat{x}^{(0)}$ by solving $A\bar{x} = \bar{b}$ in a FPS.
2. For $i = 0, 1, 2, \ldots$ until the soln is good enough:
3. Compute $\bar{r}^{(i)} = b - A\bar{\hat{x}}^{(i)}$
4. Solve $A\bar{z}^{(i)} = \bar{r}^{(i)}$ for some $\bar{z}^{(i)}$
5. Update $\bar{\hat{x}}^{(i+1)} = \bar{\hat{x}}^{(i)} + \bar{\hat{z}}^{(i)}$

8. Vector Norms:

- Let $\bar{x} \in R^n$. Then:

a. $\|\bar{x}\|_1 = \sum_{i=1}^{n} |x_i|$

b) $\|\bar{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}$

c) $\|\bar{x}\|_\infty = \max_{1 \le i \le n} (|x_i|)$

In general, for $p > 0$, $\|\bar{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$.

- Properties:

a) $\|\bar{x}\| > 0$, if $\bar{x} \neq \bar{0}$

b) $\|a\bar{x}\| = |a| \|\bar{x}\|$ for any scalar a.

c) $\|\bar{x}+\bar{y}\| \le \|\bar{x}\| + \|\bar{y}\|$ Triangle Inequality

d) $\|\bar{x}\|_1 \ge \|\bar{x}\|_2 \ge \|\bar{x}\|_\infty$

e) $\|\bar{x}\|_1 \le \sqrt{n} \|\bar{x}\|_2$

f) $\|\bar{x}\|_2 \le \sqrt{n} \|\bar{x}\|_\infty$

g) $\|\bar{x}\|_1 \le n \|\bar{x}\|_\infty$

E.g. 10 Let $\bar{x} = [3, 5, -7, 8]$

$$\|\bar{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

$$= 3+5+7+8$$

$$= 23$$

$$\|\bar{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$$
$$= (3^2 + 5^2 + 7^2 + 8^2)^{1/2}$$
$$= \sqrt{147}$$

$$\|\bar{x}\|_\infty = 8$$

9. Matrix Norms:

- Let $A \in \mathbb{R}^{n \times m}$ (I.e. A is a nxm matrix).

- $\|A\|_1 = \max_{1 \le j \le m} \left(\sum_{i=1}^{n} |a_{ij}|\right)$
  = Max absolute col sum

- $\|A\|_\infty = \max_{1 \le i \le n} \left(\sum_{j=1}^{m} |a_{ij}|\right)$
  = Max absolute row sum

E.g. || Let $A = \begin{bmatrix} 1 & -7 \\ -2 & -3 \end{bmatrix}$. Find $\|A\|_1$ and $\|A\|_\infty$

Soln:

$\|A\|_1 = \max(|1| + |-2|, |-7| + |-3|)$
$= \max(3, 10)$
$= 10$

$\|A\|_\infty = \max(|1| + |-7|, |-2| + |-3|)$
$= \max(8, 5)$
$= 8$

— Properties:

a) $\|A\| > 0$, if $A \neq 0$

b) $\|\alpha A\| = |\alpha| \, \|A\|$ for any scalar α

c) $\|A+B\| \leq \|A\| + \|B\|$

d) $\|AB\| \leq \|A\| \cdot \|B\|$

e) $\|A\bar{x}\| \leq \|A\| \cdot \|\bar{x}\|$ for any vector $\bar{x}$.

f) In general, $\|A\| = \max_{\bar{x} \neq \bar{0}} \frac{\|A\bar{x}\|}{\|\bar{x}\|}$

g) $\|A\|_2 = \sigma_{max}(A)$, where $\sigma_{max}(A)$ is the largest singular value of matrix A.

h) $\|A\|_2 \leq \|A\|_F$, where $\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$

and is called the **Frobenius norm**.

10. Tutorial Examples:

E.g. 12 Given $A = \begin{bmatrix} 2 & 6 & 6 \\ 3 & 5 & 12 \\ 6 & 6 & 12 \end{bmatrix}$ and $\bar{b} = \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$

use LU factorization to solve $A\bar{x} = \bar{b}$.

Soln:

We want $A = LU$.

Then, $A\bar{x} = \bar{b} \iff (LU)\bar{x} = \bar{b}$

$\iff L\bar{d} = b$, where $\bar{d} = U\bar{x}$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -3/2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}$$

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -3/2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 \\ 3 & 5 & 12 \\ 6 & 6 & 12 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 6 & 6 \\ 0 & -4 & 3 \\ 0 & -12 & -6 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}$$

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 \\ 0 & -4 & 3 \\ 0 & -12 & -6 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 6 & 6 \\ 0 & -4 & 3 \\ 0 & 0 & -15 \end{bmatrix} \leftarrow U$$

$$U = \begin{bmatrix} 2 & 6 & 6 \\ 0 & -4 & 3 \\ 0 & 0 & -15 \end{bmatrix}$$

$$L = L_1^{-1} \cdot L_2^{-1}$$

$$= L_1^{-1} + L_2^{-1} - I$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 3/2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 3/2 & 1 & 0 \\ 3 & 3 & 1 \end{bmatrix}$$

We now have L and U.
I will define $\bar{d} = U\bar{x}$.

$L\bar{d} = \bar{b}$, solve for $\bar{d}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3/2 & 1 & 0 \\ 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$$

$d_1 = 20$

$\frac{3}{2} d_1 + d_2 = 25 \rightarrow 30 + d_2 = 25 \rightarrow d_2 = -5$

$3d_1 + 3d_2 + d_3 = 30 \rightarrow 60 + (-15) + d_3 = 30 \rightarrow d_3 = -15$

$$\bar{d} = \begin{bmatrix} 20 \\ -5 \\ -15 \end{bmatrix}$$

Now, solve for $\bar{x}$ in $U\bar{x} = \bar{d}$

$$\begin{bmatrix} 2 & 6 & 6 \\ 0 & -4 & 3 \\ 0 & 0 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 20 \\ -5 \\ -15 \end{bmatrix}$$

$x_3 = 1$

$-4x_2 + 3x_3 = -5 \rightarrow -4x_2 + 3 = -5 \rightarrow -4x_2 = -8 \rightarrow x_2 = 2$

$2x_1 + 6x_2 + 6x_3 = 20 \rightarrow 2x_1 + 12 + 6 = 20 \rightarrow 2x_1 = 2 \rightarrow x_1 = 1$

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Note: I previously did this example on page 17 as example 7. You can see that we got the same soln.

E.g. 13 Do the same example as example 12, but this time, also do pivoting.

Soln:
Recall that we want the elements along the main diagonal, the **pivot**, to be the largest value in that column where the entries are chosen from and below the pivot.

I.e. The position of a pivot is a_{kk}, $1 \le k \le n$. we want a_{kk} to be the largest value in col k starting from row k and going down.

Looking at the first col of A, we see that the largest value of col 1 starting from row 1 is the 6 on row 3. Hence, we swap rows 1 and 3.

$$P_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$P_1 A = \begin{bmatrix} 6 & 6 & 12 \\ 3 & 5 & 12 \\ 2 & 6 & 6 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/3 & 0 & 1 \end{bmatrix}$$

$$L_1(P_1 A) = \begin{bmatrix} 6 & 6 & 12 \\ 0 & 2 & 6 \\ 0 & 4 & 2 \end{bmatrix}$$

Now, looking at col 2 of $L_1(P_1 A)$, we see that the highest value of col 2 starting from row 2 is 4. So, we swap rows 2 and 3.

$$P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$P_2(L_1 P_1 A) = \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

$$L_2(P_2 L_1 P_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix} \leftarrow U$$

$L_2 P_2 L_1 P_1 A \leftrightarrow L_2 \underbrace{P_2 L_1 P_2}_{\tilde{L}_1} P_2 P_1 A$, because $P_i \cdot P_i = I$

$$\tilde{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}$$

Recall: If you do $P_2 M$, you switch the 2nd and 3rd rows of M. If you do MP_2, you switch the 2nd and 3rd cols of M.

$$L_2 \tilde{L}_1 P_2 P_1 A = U$$

$$\Leftrightarrow \underbrace{P_2 P_1}_{P} A = \underbrace{\tilde{L}_1^{-1} L_2^{-1}}_{L} U$$

$$PA = LU$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix}$$

Recall that we started off with $A\bar{x} = \bar{b}$.
Now, we have $PA\bar{x} = P\bar{b}$, where $P = P_2 P_1$

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

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

$$P\bar{b} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$$

$$= \begin{bmatrix} 30 \\ 20 \\ 25 \end{bmatrix}$$

$L U\bar{x} = P\bar{b}$
Let $\bar{d} = U\bar{x}$
I will solve $L\bar{d} = P\bar{b}$ for $\bar{d}$.
Then, I will solve $U\bar{x} = \bar{d}$ for $\bar{x}$.

$$L\bar{d} = P\bar{b}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 20 \\ 25 \end{bmatrix}$$

$d_1 = 30$

$\frac{d_1}{3} + d_2 = 20 \rightarrow 10 + d_2 = 20 \rightarrow d_2 = 10$

$\frac{d_1}{2} + \frac{d_2}{2} + d_3 = 25 \rightarrow 15 + 5 + d_3 = 25 \rightarrow d_3 = 5$

$$\bar{d} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$$

$U\bar{x} = \bar{d}$

$$\begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$$

$x_3 = 1$

$4x_2 + 2x_3 = 10 \rightarrow 4x_2 + 2 = 10 \rightarrow 4x_2 = 8 \rightarrow x_2 = 2$

$6x_1 + 6x_2 + 12x_3 = 30 \rightarrow 6x_1 + 12 + 12 = 30 \rightarrow 6x_1 = 6 \rightarrow x_1 = 1$

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

which is what we got in the previous 2 tries.

LU Factorization Notes

1. LU Factorization Without Pivoting:

- Want to solve $A\bar{x} = \bar{b}$.
- We can $L_{n-1} L_{n-2} \dots L_1 A = U \leftrightarrow A = \underbrace{L_1^{-1} \dots L_{n-1}^{-1}}_{L} U$

where L is a lower triangular matrix and U is an upper triangular matrix.

- Now we have $LU\bar{x} = \bar{b}$.
- Let $U\bar{x} = \bar{d}$
- Now, we solve $L\bar{d} = \bar{b}$ for $\bar{d}$ (forward substitution) and $U\bar{x} = \bar{d}$ for $\bar{x}$ (backward substitution).
- Note: Even if A is non-singular, we may not always be able to use this strategy.
- To compute L_i^{-1}, simply take L_i and switch the sign of the multipliers.
- If L_i is a Gauss Transformation, then L_i^{-1} exists and is also a Gauss Transformation.
- If L_i and L_j are Gauss Transformations, and $j > i$, then $L_i L_j = L_i + L_j - I$

E.g.1 Given $A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix}$, find L and U.

Soln

$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}$

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix}$$

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

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$$

$$L_2 (L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix}$$

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

$$U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix}$$ ← Upper Triangular Matrix

$$L = L_1^{-1} \cdot L_2^{-1}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}$$ ← Lower Triangular Matrix

Another way to compute L in this case is

$$L = L_1^{-1} + L_2^{-1} - I$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}$$

Now, let's see what LU equals to.

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix}}_{A}$$

E.g. 2 Given $A = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & 3 \\ 2 & 3 & 1 \end{bmatrix}$ and $\bar{b} = \begin{bmatrix} 4 \\ -6 \\ 7 \end{bmatrix}$

Solve using LU factorization.

Soln:

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & 3 \\ 2 & 3 & 1 \end{bmatrix}$$

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

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 1 \end{bmatrix}$$

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & -3 & 4 \\ 0 & 1 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & -1 \\ 0 & -3 & 4 \\ 0 & 0 & \frac{13}{3} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 1 & -1 \\ 0 & -3 & 4 \\ 0 & 0 & \frac{13}{3} \end{bmatrix}$$

$$L = L_1^{-1} + L_2^{-1} - I$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -\frac{1}{3} & 1 \end{bmatrix}$$

$LU\bar{x} = \bar{b}$
Let $U\bar{x} = \bar{d}$

Now, we solve $L\bar{d} = \bar{b}$ for $\bar{d}$ and then $U\bar{x} = \bar{d}$ for $\bar{x}$.

$$L\bar{d} = \bar{b}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -6 \\ 7 \end{bmatrix}$$

$d_1 = 4$
$d_1 + d_2 = -6$
$4 + d_2 = -6$
$d_2 = -10$
$2d_1 - \frac{d_2}{3} + d_3 = 7$

$6d_1 - d_2 + 3d_3 = 21$
$24 - (-10) + 3d_3 = 21$
$34 + 3d_3 = 21$
$3d_3 = -13$
$d_3 = \frac{-13}{3}$

$$\bar{d} = \begin{bmatrix} 4 \\ -10 \\ -\frac{13}{3} \end{bmatrix}$$

Now, we'll solve $U\bar{x} = \bar{d}$ for $\bar{x}$.

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & -3 & 4 \\ 0 & 0 & \frac{13}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ -\frac{13}{3} \end{bmatrix}$$

$x_3 = -1$
$-3x_2 + 4x_3 = -10$
$-3x_2 - 4 = -10$
$-3x_2 = -6$
$x_2 = 2$
$x_1 + x_2 + x_3 = 4$
$x_1 + 2 + 1 = 4$
$x_1 + 3 = 4$
$x_1 = 1$

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} \leftarrow 1$$

2. LU Factorization with Pivoting:
- When we do LU factorization with pivoting, we want the biggest value for each pivot where the value is in the same column as the pivot and is either the pivot or below the pivot.

E.g. Take $\begin{bmatrix} 1 & \overset{3}{} & 3 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}$

Look at the second column. The pivot is 1. However, it's the smallest value in that column. We want to replace it with the biggest value in that column s.t. the value is the pivot or below the pivot. In this example, it's 2. We ignore 3 because 3 is above the pivot.

- When we swap/switch rows, we need to multiply by a permutation matrix, P.

- Now, we have $L_{m-1}P_{m-1}\dots L_2P_2L_1P_1A = U$

$$\leftrightarrow L_{m-1}\hat{L}_{m-2}\dots\hat{L}_1P_{m-1}\dots P_1A = U$$

$$\leftrightarrow \underbrace{P_{m-1}\dots P_1}_{P}A = \underbrace{\hat{L}_1^{-1}\hat{L}_2^{-1}\dots L_{m-1}^{-1}}_{L}U$$

$$\leftrightarrow PA = LU$$

Originally, we had $A\bar{x} = \bar{b}$.
Now, we have $PA\bar{x} = P\bar{b}$

$$\leftrightarrow LU\bar{x} = P\bar{b}$$

Let $U\bar{x} = \bar{d}$
We solve $\underbrace{L\bar{d} = P\bar{b} \text{ for } \bar{d}}$ and $\underbrace{U\bar{x} = \bar{d} \text{ for } \bar{x}}$.

Forward Solve — Backward Solve

E.g. 3 Solve $\begin{bmatrix} 2 & 6 & 6 \\ 3 & 5 & 12 \\ 6 & 6 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$

using LU factorization with pivoting.

Soln:

Step 1: Since we want the pivot to be the biggest value in the col at or below the pivot, we need to switch rows 1 and 3.

$$P_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$P_1 A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 \\ 3 & 5 & 12 \\ 6 & 6 & 12 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 6 & 12 \\ 3 & 5 & 12 \\ 2 & 6 & 6 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}$$

$$L_1(P_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 3 & 5 & 12 \\ 2 & 6 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 6 & 12 \\ 0 & 2 & 6 \\ 0 & 4 & 2 \end{bmatrix}$$

Step 2: Now, we switch the 2nd and 3rd rows so that the pivot is 4 instead of 2.

$$P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$P_2(L_1P_1A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 0 & 2 & 6 \\ 0 & 4 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

$$L_2(P_2L_1P_1A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix}$$

$$U = \begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix}$$

Step 3: Right now we have $L_2 P_2 L_1 P_1 A$. We want the L's together before the P's. I.e. We want $L_2 \hat{L}_1 P_2 P_1 A$. To do this, we'll multiply $L_2 P_2 L_1 P_1 A$ by $P_2 \cdot P_2$ at a specific spot.

$L_2 P_2 L_1 P_2 P_2 P_1 A$

Note: The inverse of any permutation matrix is itself. So, when we do $P_i \cdot P_i$, we get I.

Note: When we pre-multiply by a permutation matrix, we swap rows. When we post-multiply by a permutation matrix, we swap columns.

$$P_2 L_1 P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 0 & 1 \\ -\frac{1}{2} & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \leftarrow \hat{L}_1$$

Note: $\hat{L}_1$ is just L_1 with its multipliers switched.

$L_2 \hat{L}_1 P_2 P_1 A = U$

$\underbrace{P_2 P_1 A}_{P} = \underbrace{\hat{L}_1^{-1} L_2^{-1}}_{L} U$

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

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

$$L = L_1^{-1} \cdot L_2^{-1}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$

Now, instead of $A\bar{x} = \bar{b}$, we have $PA\bar{x} = P\bar{b}$

$\leftrightarrow LU\bar{x} = P\bar{b}$

$$P\bar{b} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 20 \\ 25 \\ 30 \end{bmatrix}$$

$$= \begin{bmatrix} 30 \\ 20 \\ 25 \end{bmatrix}$$

$LU\bar{x} = P\bar{b}$

Let $U\bar{x} = \bar{d}$

We solve $L\bar{d} = P\bar{b}$ for $\bar{d}$, and $U\bar{x} = \bar{d}$ for $\bar{x}$.

Forward Solve — Backward Solve

$$L\bar{d} = P\bar{b}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 20 \\ 25 \end{bmatrix}$$

$d_1 = 30$

$\frac{d_1}{3} + d_2 = 20$

$10 + d_2 = 20$

$d_2 = 10$

$\frac{d_1}{2} + \frac{d_2}{2} + d_3 = 25$

$15 + 5 + d_3 = 25$

$d_3 = 5$

$$\bar{d} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$$

$$U\bar{x} = \bar{d}$$

$$\begin{bmatrix} 6 & 6 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 10 \\ 5 \end{bmatrix}$$

$5x_3 = 5$

$x_3 = 1$

$4x_2 + 2x_3 = 10$

$4x_2 + 2 = 10$

$4x_2 = 8$

$x_2 = 2$

$6x_1 + 6x_2 + 12x_3 = 30$

$6x_1 + 12 + 12 = 30$

$6x_1 = 6$

$x_1 = 1$

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Non-Linear Methods

1. Introduction:

- Let $F: R - R$. We want to solve for x s.t. $F(x) = 0$. x is called a **root**.
- Examples of non-linear eqns:

1. $2x^2 + 7 = 0$
2. $x - e^{-x} = 0$ **Transcendental Eqn**

- Typically, we can't find a closed form or analytical soln to solve non-linear eqns. However, we can find iterative methods that generate an approximation.

I.e. let $k = 0, 1, 2, \ldots$ As $k \to \infty$, $\hat{x}_k \to \tilde{x}$, where $\hat{x}_k$ is the approximation.

2. Fixed Point Methods (FPM):

- $F(\tilde{x}) = 0$ is called a **root finding problem.**
- $\tilde{x} = g(\tilde{x})$ is called a **fixed point problem.**
- $F(\tilde{x}) = 0$ is equivalent to $\tilde{x} = g(\tilde{x})$.

E.g. $\underbrace{x - e^{-x} = 0}_{\text{Root Finding problem}} \iff \underbrace{x = e^{-x}}_{\text{Fixed point problem}}$

- We can let $g(\tilde{x}) = \tilde{x} - F(\tilde{x})$ or let $g(\tilde{x}) = \tilde{x} - h(\tilde{x})F(\tilde{x})$ where $h(\tilde{x})$ is an auxiliary function.
- $g(\tilde{x}) = \tilde{x} - F(\tilde{x})$ is called the **first form.**
- $g(\tilde{x}) = \tilde{x} - h(\tilde{x})F(\tilde{x})$ is called the **second form.**

- If we use the first form, then $F(\tilde{x})=0$ is always equivalent to $\tilde{x}=g(\tilde{x})$.

Proof:
LHS: $F(\tilde{x})=0$
RHS: $\tilde{x}=g(\tilde{x})$
$= \tilde{x}-F(\tilde{x})$
$0=-F(\tilde{x})$
$F(\tilde{x})=0$
Hence, LHS = RHS
- If we use the second form, if $F(\tilde{x})=0$, then $\tilde{x}=g(\tilde{x})$. However, we could have $\tilde{x}=g(\tilde{x})$ but $F(\tilde{x})\neq 0$. This situation occurs if $h(\tilde{x})=0$. Hence, the two equations aren't equivalent. Furthermore, after we find a fixed point, we need to check if it is a root.
- The advantage of the second form is that there's flexibility in designing $g(\tilde{x})$, to make iteration converge faster.

3. Fixed Point Iteration (FPI):

- Start with an approximate soln $\hat{x}_0$ then iterate $\hat{x}_{k+1}=g(\hat{x}_k)$, $k=0,1,2,\ldots$ until convergence or failure.
- E.g. 1 Let $F(x)=x^2+2x-3$. We know that the roots are 1 and -3.

Consider the FPI $x_{k+1}=x_k+\dfrac{(x_k)^2+2x_k-3}{(x_k)^2-5}$

for the fixed point problem $x=g(x)$

$$=x+\frac{x^2+2x-3}{x^2-5}$$

We see that this is the second form of the fixed point problem where $h(x) = \frac{-1}{x^2-5}$.

Since $h(x) \neq 0, \forall x \in R$, this means that we don't need to check if a fixed point is a root.

If we start with $\hat{x}_0 = -5$, then $\hat{x}_k$'s approach -3.
If we start with $\hat{x}_0 = 5$, then $\hat{x}_k$'s do not converge.
If we start with $\hat{x}_0 = 0$, then $\hat{x}_k$'s converge to 1.

Hence, we can see that depending on $\hat{x}_0$, the FPI may converge to some fixed point or may not converge.

4. Fixed Point Theorem (FPT):

- If there's an interval $[a,b]$ s.t.

1. $g(x) \in [a,b] \quad \forall x \in [a,b]$
2. $\|g'(x)\| \leq L < 1 \quad \forall x \in [a,b]$

then $g(x)$ has a unique fixed point in $[a,b]$.

Proof:

Note: This proof has 3 parts but we were only shown part 1. The other parts were left for assignments.

Start with any initial guess, $\hat{x}_0 \in [a,b]$, and iterate.
$\hat{x}_{k+1} = g(\hat{x}_k), \quad k = 0, 1, 2, \ldots$
Then, all $\hat{x}_k \in [a,b]$.

Furthermore, $x_{k+1} - x_k = g(x_k) - g(x_{k-1})$
$= g'(n_k)(x_k - x_{k-1})$
for some $n_k \in [x_{k-1}, x_k]$
$\subset [a,b]$
We know this by the Mean Value Theorem (MVT)

Therefore, $|x_{k+1} - x_k| \leq |g'(n_k)(x_k - x_{k-1})|$
$= L|x_k - x_{k-1}|$
Then, $|x_k - x_{k-1}| \leq \dots \leq L^k |x_1 - x_0|$
Since we know that $L<1$, $|x_k - x_{k-1}| \to 0$ as $k \to \infty$.

This means that x_k converges to some point $\tilde{x} \in [a,b]$.

To complete the proof, we have to show 2 things:
1. $\tilde{x}$ is a fixed point. I.e. $\tilde{x} = g(\tilde{x})$.
2. $\tilde{x}$ is unique.

5. Rate of Convergence:
- Def: If $\lim_{\tilde{x}_k \to \tilde{x}} \frac{|\tilde{x} - x_{k+1}|}{|\tilde{x} - x_k|^p} = c \neq 0$, then we have the p-th order convergence to fixed point $\tilde{x}$.

E.g. 2 This example will show the importance of p. Consider the table of absolute errors of iterates, $|\tilde{x} - x_k|$, below.

k	P=1, C=1/2	P=2, C=1
0	10^{-1}	10^{-1}
1	$5 \cdot 10^{-2}$	10^{-2}
2	$2.5 \cdot 10^{-2}$	10^{-4}
3	$1.25 \cdot 10^{-2}$	10^{-8}
4	$6.125 \cdot 10^{-3}$	10^{-16}

We start with 10^{-1} for both systems.
For system 1, P=1, C=1/2, we get
$|\tilde{x} - x_{k+1}| = \frac{|\tilde{x} - x_k|}{2}$. Hence, with each iteration, we divide by 2.

For system 2, P=2, C=1, we get
$|\tilde{x} - x_{k+1}| = (|\tilde{x} - x_k|)^2$. Hence, with each iteration, we square.

Notice that despite having C=1, the third column converges much faster. This is because p=2 in the third column.

6. Rate of Convergence Thm:

- For the FPI $x_{k+1} = g(x_k)$, if $g'(\tilde{x})$, $g''(\tilde{x})$, ..., $g^{(p-1)}(\tilde{x}) = 0$ but $g^p(\tilde{x}) \neq 0$, then we have **p-th order convergence.**

Proof:

$$x_{k+1} = g(x_k)$$

$$= g(\tilde{x} + (x_k - \tilde{x}))$$

$$= g(\tilde{x}) + g(x_k - \tilde{x})g'(\tilde{x}) + \frac{(x_k - \tilde{x})^2}{2!} g''(\tilde{x})$$

$$+ \ldots + \frac{(x_k - \tilde{x})^{p-1}}{(p-1)!} g^{(p-1)}(\tilde{x}) + \underbrace{\frac{(x_k - \tilde{x})^p}{p!} g^p(n_k)}_{\text{Remainder Term}}$$

Taylor Series

If we have $g'(\tilde{x})$, $g''(\tilde{x})$, ..., $g^{(p-1)}(\tilde{x}) = 0$, then we get

$$x_{k+1} = g(\tilde{x}) + \frac{(x_k - \tilde{x})^p}{p!} g^p(n_k)$$

Recall that $g(\tilde{x}) = \tilde{x}$.

$$x_{k+1} = \tilde{x} + \frac{(x_k - \tilde{x})^p}{p!} g^p(n_k)$$

Rearranging the eqn, we get

$$\frac{x_{k+1} - \tilde{x}}{(x_k - \tilde{x})^p} = \frac{1}{p!} g^p(n_k)$$

As $k \to \infty$, $x_k \to \tilde{x}$, $n_k \in [\tilde{x}, x_k] \to \tilde{x}$.
We can rewrite this as

$$\lim_{x_k \to \tilde{x}} \frac{|x_{k+1} - \tilde{x}|}{|x_k - \tilde{x}|^p} = \frac{1}{p!} g^p(\tilde{x})$$

We see that if the p^{th} derivative of $g(\tilde{x})$ is not zero, we get p^{th} order convergence.

We can see that by using the second form of FPI, we can pick a $h(x)$ s.t. the p^{th} derivative of g is not zero.

7. Newton's Method:
   - Formula: $x_{k+1} = x_k - \frac{F(x)}{F'(x)}$

This is the second form with $h(x) = \frac{1}{F'(x)}$.

- Suppose that $F(\tilde{x}) = 0$ and $F'(\tilde{x}) \neq 0$.

$$g'(x) = 1 - \left(\frac{F'(x)\,F'(x) - F(x)\,F''(x)}{(F'(x))^2} \right)$$

$$= \frac{(F'(x))^2 - (F'(x))^2 + F(x)F''(x)}{(F'(x))^2}$$

$$= \frac{F(x)F''(x)}{(F'(x))^2}$$

$$= 0$$

By the rate of convergence thm, Newton's Method has at least quadratic convergence for any function F.

- Geometric Interpretation of NM:
We want to solve $F(x)=0$ at an initial guess X_k which is an approximate/model to $F(x)$ by a linear polynomial $p(x)$ that satisfies the conditions:

1. $p(X_k) = F(X_k)$
2. $p'(X_k) = F'(X_k)$

} $p(x)$ is the tangent line to $F(x)$.

$$p_k(x) = F(X_k) + (x - X_k) F'(X_k)$$

Then, X_{k+1} is a root of $p_k(x)$.

I.e. $p_k(X_{k+1}) = 0 \rightarrow F(X_k) + (X_{k+1} - X_k) F'(X_k) = 0$

$$\rightarrow X_{k+1} = X_k - \frac{F(X_k)}{F'(X_k)}$$

Note: NM doesn't always converge.

8. Secant Method:

- Take NM, $X_{k+1} = X_k - \frac{F(X_k)}{F'(X_k)}$

and approximate $F'(X_k)$ with $\frac{F(X_k) - F(X_{k-1})}{X_k - X_{k-1}}$

Now, the formula is $X_{k+1} = X_k - \frac{F(X_k)(X_k - X_{k-1})}{F(X_k) - F(X_{k-1})}$

- The secant method is not a FPI.
I.e. It can't be expressed as $x_{k+1} = g(x_k)$.
This is because there are 2 x_{k-1} terms.

- We can't directly use FCT or RCT to analyze this method. However, with some adjustments, we can prove that $p = \frac{1+\sqrt{5}}{2}$.

This is called **Superlinear convergence.**

Note: NM has a $p = 2$.

- NM requires 2 function evaluations per iteration as F and F' aren't the same. The secant method only requires 1 function evaluation as we only need to compute $F(x_k)$. $F(x_{k-1})$ has already been computed in the previous step.
The effective rate of convergence takes into account cost per iteration as well as speed. Hence, the secant method is effectively faster than Newton's Method.

Note: The secant method doesn't always converge.

9. Bisection Method:
- Often, if we use NM/SM, we will start off with the wrong initial guess and won't get convergence.

- The Bisection Method will always guarantee convergence, but it's much slower.

- We need to find an $a<b$ s.t. $F(a) \leq 0 \leq F(b)$ or $F(b) \leq 0 \leq F(a)$. This means that there is at least 1 root in $[a,b]$.

Assume $F(a) \leq 0 \leq F(b)$
Loop until $b-a$ is small enough.
Let $m = \frac{a+b}{2}$

If $F(m) \leq 0$, let $a=m$, else $b=m$
Repeat for the interval $[m,b]$ or $[a,m]$.

Then, we have linear rate of convergence with $p=1$, $C=1/2$, and with guaranteed convergence.

10. Hybrid Method:

- Combines slow, reliable methods with faster ones that require a more accurate guess.
E.g. Bisection + Newton

11. System of Non-Linear Eqn:

- Problem: Solve $F(\bar{x}) = \bar{0}$
- Newton's Method can be extended for a system of non-linear eqns.

$\bar{x}_{k+1} = \bar{x}_k - \frac{F(\bar{x}_k)}{F'(\bar{x}_k)}$ where F' is the Jacobian Matrix of F

When we divide by a matrix, we multiply by its inverse.

Hence, we get: $\overline{x_{k+1}} = \overline{x_k} - (F'(\overline{x_k}))^{-1} F(\overline{x_k})$ or

$$\underbrace{(F'(\overline{x_k}))(\overline{x_{k+1}} - \overline{x_k}) = -F(\overline{x_k})}$$

This is simply in the form of $A\bar{x} = \bar{b}$.

This is very expensive. We can use a pseudo NM by holding the Jacobian matrix fixed for a few iterations. This means that we can reuse the $PA = LU$ factorization. This is alright so long as we are still converging.

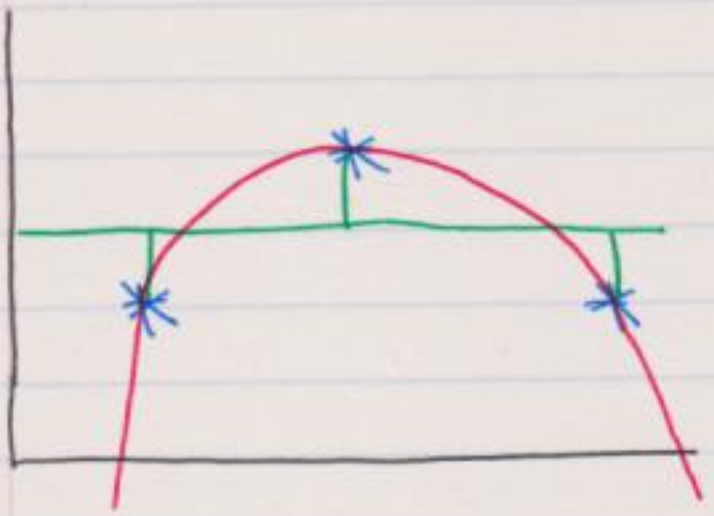
CSCC37 Approximation / Interpolation Notes

Introduction:

- With **approximation**, the line does not go through all the points on the graph.

- With **interpolation**, the line goes through the points on the graph.

E.g.



The red line is interpolation.

The green line is approximation.

- Truncated Taylor Series:

$$p(x) = F(a) + F'(a)(x-a) + \dots + \frac{F^{(n)}(a)}{n!}(x-a)^n$$

Only has the first n+1 terms

This is polynomial because of the $(x-a)^i$, $i = 0, 1, \dots$

The error $e(x) = p(x) - F(x)$

$$= \frac{F^{(n+1)}(\eta)}{(n+1)!}(x-a)^{n+1}$$

- Some other approximations are:

a) Interpolation: Find a polynomial p s.t. $p(x_i) = F(x_i)$, $i = 0, 1, 2, \ldots$
F is the function we're trying to approximate. It could simply be a set of data.

b) Least Squares: Find a polynomial p s.t. $p(x)$ minimize $\|F-p\|_2 = \left(\int_a^b (F(x)-p(x))^2 \, dx\right)^{1/2}$

Other norms we can use for least squares are:

i) $\|F-p\|_\infty = \max_{a \le x \le b} |F(x)-p(x)|$

ii) $\|F-p\|_1 = \int_a^b |F(x)-p(x)| \, dx$

Note: If you want to approximate a function around a given point and you have access to derivatives of the function, then you may want to use a Taylor expansion. If you want to approximate a function on an interval where you can access some function values but not derivatives, you can use an interpolation polynomial.

Polynomial Interpolation:
- Consider P_n, which is the set of polynomials of degree $\le n$. This is a function space and requires the basis of $n+1$ functions. The most common basis is the monomial basis, which is $\{x^i, i = 0, 1, 2, \ldots\}$.

Weierstrass' Thm:

- If a function F is continuous on an interval $[a,b]$, then for any $\varepsilon > 0$, $\exists p_\varepsilon$ s.t. $\|F - p_\varepsilon\| < \varepsilon$.
- This means that for any continuous function on a closed interval $[a,b]$, there exists some polynomial that is as close to it as it can be.

Numerical Methods For Polynomical Interpolation:

1. Vandermonde Thm:

- Also known as Method of Undetermined Coefficients.
- Thm: For any sets $\{x_i, i=0,1,\dots n\}$ and $\{y_i, i=0,1,\dots,n\}$, for distinct x_i's and undistinct y_i's, $\exists$ a unique polynomial $P(x) \in P_n$ s.t. $P(x_i) = y_i$, $i=0,1,\dots n$.
- Proof:

If $P(x)$ exists, then it must be possible to write it as

$$P(x) = \sum_{i=0}^{n} a_i x^i$$

This can be converted into a matrix problem with $P(x_i) = y_i$, $i=0,1,2,\dots n$.

We can solve for the a_i's using the Vandermonde Matrix.

$$\begin{bmatrix} (x_0)^0 & (x_0)^1 & (x_0)^2 & \cdots & (x_0)^n \\ (x_1)^0 & (x_1)^1 & (x_1)^2 & \cdots & (x_1)^n \\ \vdots & \vdots & \vdots & & \vdots \\ (x_n)^0 & (x_n)^1 & (x_n)^2 & \cdots & (x_n)^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Vandermonde Matrix

The question now becomes "Is the Vandermonde Matrix non-singular?"

The Vandermonde matrix is non-singular because all the columns are linearly independent.

- The Vandermonde Theorem proves existence but does not lead to the best algorithm. It can be poorly conditioned.
- Gives the monomial basis.

2. Lagrange Basis:
- For a simple interpolation problem $P(x_i) = y_i$, $i = 0, 1, \ldots n$, consider the basis

$$l_i = \prod_{\substack{j=0 \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \quad \text{for } i = 0, 1, 2, \ldots, n$$

$$= \left(\frac{x - x_0}{x_i - x_0}\right) \cdots \left(\frac{x - x_{i-1}}{x_i - x_{i-1}}\right)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right) \cdots \left(\frac{x - x_n}{x_i - x_n}\right)$$

Notice that we skipped $\frac{x - x_i}{x_i - x_i}$.

- $l_i(x) \in P_n$
- Consider $l_i(x_j)$.

If $j=i$, we get $\prod_{\substack{j=i \\ j\neq i}}^{n} \frac{x_i - x_j}{x_i - x_j} = 1$

$j=i$
$x_j = x_i$

I.e. $l_i(x_j), j=i, = \prod_{\substack{j=i \\ j\neq i}}^{n} \frac{x_i - x_j}{x_i - x_j}$

$$= \left(\frac{x_i - x_0}{x_i - x_0}\right) \cdots \left(\frac{x_i - x_{i-1}}{x_i - x_{i-1}}\right) \left(\frac{x_i - x_{i+1}}{x_i - x_{i+1}}\right) \cdots \left(\frac{x_i - x_n}{x_i - x_n}\right)$$

$$= 1$$

If $j \neq i$, we get $l_i(x_j), j \neq i = 0$.

$$\prod_{\substack{j\neq i \\ j\neq i}}^{n} \frac{x_j - x_j}{x_i - x_j} = 0$$

Expanding the product above, we get

$$\left(\frac{x_j - x_0}{x_i - x_0}\right) \cdots \left(\frac{x_j - x_{i-1}}{x_i - x_{i-1}}\right)\left(\frac{x_j - x_{i+1}}{x_i - x_{i+1}}\right) \cdots \left(\frac{x_j - x_n}{x_i - x_n}\right)$$

One of these products will be 0 as $0 \leq j \leq n$, and $j \neq i$. Hence, the entire product will be 0.

To summarize, $l_i(x_j) = \begin{cases} 1, & \text{if } i=j \\ 0, & \text{if } i \neq j \end{cases}$

- The lagrange polynomical is free to construct, but very expensive to evaluate at non-interpolation points.

- With the basis function, we can write out the interpolating polynomial for free.

$$P(x) = \sum_{i=0}^{n} l_i(x) y_i$$

Furthermore, $P(x_i) = y_i$, for $i=0, 1, \dots, n$ because

$$P(x_i) = \sum_{i=0}^{n} \underbrace{l_i(x_i) y_i}$$

⇓

Equals to 1, as stated previously

3. Newton Basis:

- Also called Divided Differences.
- For a simple interpolation $P(x_i) = y_i$, $i=0,1,\dots n$, we look for an interpolating of the form

$$P(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

Converting into a matrix, we get

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & & 0 \\ \vdots & & & & \vdots \\ 1 & x_n - x_0 & \cdots & \cdots & \prod_{i=0}^{n-1} (x_n - x_i) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

This is a lower triangular matrix, meaning that no factorization is involved.

$$a_0 = y_0$$

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

$$a_2 = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}$$

} Divided Differences

- Divided Differences: $Y[x_i] = y(x_i) = y_i$

$$Y[x_{i+k}, \ldots, x_i] = \frac{Y[x_{i+k}, \ldots, x_{i+1}] - Y[x_{i+k-1}, \ldots, x_i]}{x_{i+k} - x_i}$$

E.g. $$Y[x_2, x_1, x_0] = \frac{Y[x_2, x_1] - Y[x_1, x_0]}{x_2 - x_0}$$

- Newton's Polynomial: $p(x) = Y[x_0] + (x-x_0)Y[x_1, x_0] + \dots + (x-x_0)(x-x_1)\dots(x-x_{n-1})y[x_n, \dots, x_0]$

Then, $p(x) \in P_n$ and $p(x_i) = y_i$, $i = 0, 1, 2, \dots, n$

E.g. Find a $P \in P_3$ s.t. $P(0)=1$, $P(1)=3$, $P(2)=9$, $P(3)=25$

Soln:

X	$Y[x_i]$	$Y[x_{i+1}, x_i]$	$Y[x_{i+2}, \dots, x_i]$	$Y[x_{i+3}, \dots, x_i]$
0	1	$\frac{3-1}{1-0} = 2$	$\frac{6-2}{2-0} = 2$	$\frac{5-2}{3-0} = 1$
1	3	$\frac{9-3}{2-1} = 6$	$\frac{16-6}{2-1} = 5$	
2	9	$\frac{25-9}{3-2} = 16$		
3	25			

$$P(x) = y[x_0] + (x-x_0)y[x_1, x_0] + (x-x_0)(x-x_1)y[x_2, x_1, x_0] + (x-x_0)(x-x_1)(x-x_2)y[x_3, \ldots, x_0]$$

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

Read coefficients from top of triangle.

- How are divided differences and derivatives related?

Consider $y[x_1, x_0] = \frac{y(x_1) - y(x_0)}{x_1 - x_0}$

$$\lim_{x_1 \to x_0} y[x_1, x_0] = \lim_{x_1 \to x_0} \frac{y(x_1) - y(x_0)}{x_1 - x_0}$$

$= y'(0)$, provided that $y'(x_0)$ exists

Consider $y[x_2, x_1, x_0] = \frac{y[x_2, x_1] - y[x_1, x_0]}{x_2 - x_0}$

$$\lim_{\substack{x_2 \to x_0 \\ x_1 \to x_0}} y[x_2, x_1, x_0] = \frac{y''(x_0)}{2!}$$

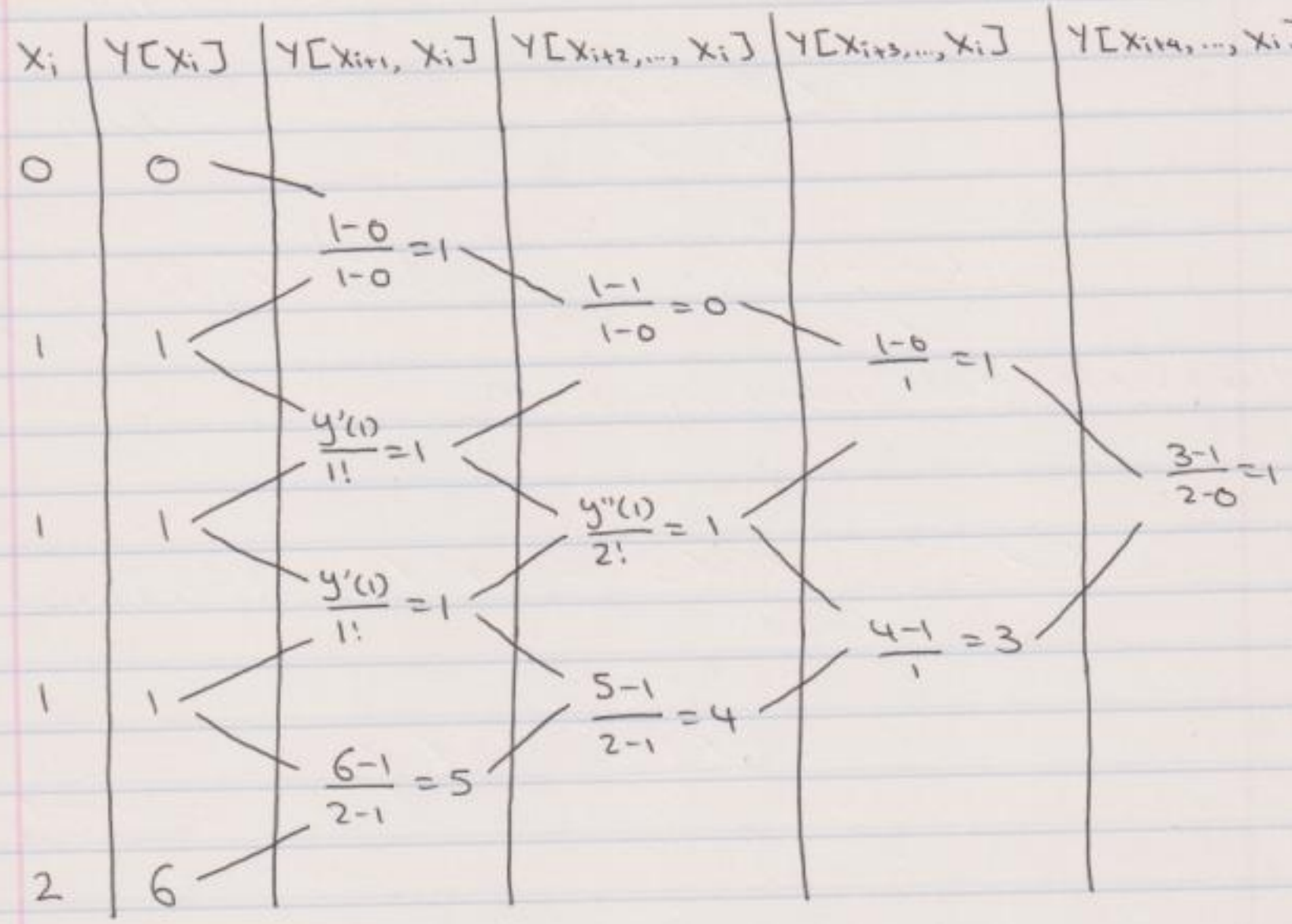
In general, we can show that

$$\lim_{\substack{x_k \to x_0 \\ x_{k-1} \to x_0 \\ \vdots \\ x_1 \to x_0}} y[x_k, \ldots, x_0] = \frac{y^{(k)}(x_0)}{k!}$$

– How does this help with osculatory interpolation, which is interpolation with derivatives?

E.g. Find $P \in P_4$ s.t. $P(0)=0$, $P(1)=1$, $P'(1)=1$, $P''(1)=2$ and $P(2)=6$.

Soln:



$$P(x) = Y[0] + xY[1,0] + x(x-1)Y[1,1,0] + x(x-1)^2 Y[1,1,1,0] + x(x-1)^3 Y[2,1,1,1,0]$$

$$= 0 + x + x(x-1)^2 + x(x-1)^3$$

← Read the coefficients from top of triangle.

Error in Polynomial Interpolation:

- $E(x) = \underbrace{Y(x)}_{\text{Underlying Function}} - \underbrace{P(x)}_{\text{Interpolating Polynomial}}$

- For a simple interpolation $P(x_i) = y_i$, $i = 0, 1, 2, \ldots, n$ we can show that $E(x) = \frac{y^{(n+1)}}{(n+1)!}(\xi) \prod_{i=0}^{n} (x - x_i)$

where $\xi \in \text{span}\{x_0, \ldots, x_n, x\}$
$= [\min\{x_0, \ldots, x_n, x\}, \max\{x_0, \ldots, x_n, x\}]$