Week 1 May 6

Wednesday, May 10, 2023 5:09 PM

1. Consider the irrational number Pi the ratio of a circle's circumference to its diameter?
    1. How can you compute Pi via a digital computer? *Hint: what is the series expansion for Pi?*

Series Expansion for π:

$$\frac{1}{1+w} = 1 - w + w^2 - w^3 + w^4 - \ldots = \sum_{n=0}^{k} (-1)^n (w)^n = \sum_{n=0}^{k} (-w)^n$$

for $w = x^2$

$$\frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + x^8 - \ldots = \sum_{n=0}^{k} (-w^2)^n$$

Integral both sides

$$\arctan y = y - \frac{y^3}{3} + \frac{y^5}{5} + \frac{y^7}{7}$$

for $y = 1$

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} \ldots = \sum_{n=0}^{k} (-1)^n \frac{1}{2n+1}$$

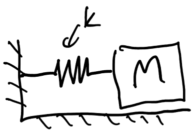
$$\pi = 4 \sum_{n=0}^{k} (-1) \frac{1}{2n+1}$$

Steps:
- raise to the power
- Add
- divide
- multiply by (-1)
- Sum

    2. How can you compute Pi via an analog computer? *Hint: what are the differential equations for sinusoidal motion? How can this be used to compute Pi?*

analogue computers use physical phenomena (e.g. electrical, mechanical)

One such mechanical system is a mass on a spring



assuming frictionless

Hooke's Law

$F = -kx$

$a = -\frac{k}{m} x$

$\Rightarrow x'' = -\frac{k}{m} x$ ← second order homogeneous eq

$x'' + \frac{k}{m} x = 0$

$\Rightarrow c^2 + \frac{k}{m} = 0$

$\frac{k}{m} \geq 0 \therefore$ complex roots

$c = \pm \sqrt{-\frac{k}{m}}$

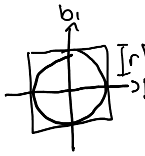
$c = \pm \sqrt{\frac{k}{m}}\, i$

$$x = e^{0t} \left[C_1 \cos\left(\sqrt{\tfrac{k}{m}}\, t\right) + C_2 \sin\left(\sqrt{\tfrac{k}{m}}\, t\right) \right]$$

$$x = C \sin(\omega t + \varphi)$$

Steps:
- measure
- multiply
- add

    3. How can you compute Pi with the use of a random number generator? *Hint: what is the ratio the area of a square to that of an inscribed circle? If you draw two random numbers, how can you tell if they define a coordinate within a square or an inscribed circle?*



Ir With uniform random distribution, generate numbers $b_1, b_2 \in \mathbb{R}$

$b_1 \in \{-r, r\}$
$b_2 \in \{-r, r\}$

$b = (b_1, b_2) \quad b \in \mathbb{R}^2$

b is 'in the circle' if $\sqrt{b_1^2 - b_2^2} \leq r$

b is 'out the circle' if $\sqrt{b_1^2 - b_2^2} > r$

Take a random b n times

$$\frac{\text{Area of circle}}{\text{Area of square}} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}$$

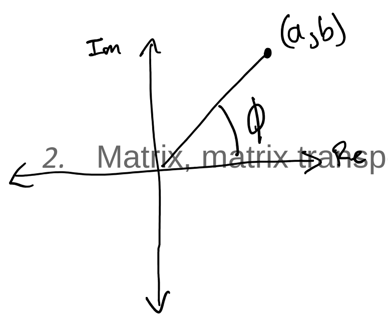
$$\frac{\pi}{4} \simeq \frac{\#\text{ Points in circle}}{\#\text{ points out circle} + \#\text{ points in circle}} \rightarrow \pi \simeq 4 \times \frac{\#\text{ Points in circle}}{\#\text{ points out circle} + \#\text{ points in circle}}$$

Steps: · generate b_1, b_2
· square, subtract, sqrt, compare
· sum
· Add
· Divide

4. For each of the above algorithms, analyze them in terms of representation of information, time (i.e., number of steps) to compute the value (i.e., execute the algorithm), the quality of the result, how to improve quality of result, and the primitive operations required.

method above

2. Review the following elementary concepts:
   1. Complex numbers, their representation, visualization, magnitude and phase



can be represented as:

(a, b)

a + bi

$\|z\| \angle \arg(z)$

magnitude:

$\|z\| = \sqrt{a^2 + b^2}$

phase:

$\arg(z) = \tan^{-1}\left(\frac{b}{a}\right)$

   2. Matrix, matrix transpose, matrix inverse, unitary matrix

$A \in \mathbb{R}^{3\times 3}$

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

$A^{-1} \cdot A = I$

A^{-1} · must be square
· $\det(A) \neq 0$ } non-singular i.e. invertable

↳ $\det(B \in \mathbb{R}^{2\times 2}) = \det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$

$$\begin{bmatrix} \cdots \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

" $\cdot \det(A) \neq 0$) invertable

↳ $\det(B \in \mathbb{R}^{2\times 2}) = \det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$
$= ad - bc$

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

$$\det(A) = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Unitary matrix
- Square matrix
- Maintains vector length/norm, orthogonality

Satisfies $A^H = A^{-1} \Rightarrow A^H A = I$
(Conjugate Transpose) (Inverse)

e.g. $U = \frac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix}$

3. Vector space, vector norms

$\bar{U} = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$

$U^H = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$

multiply & show $U^H U = I$

Vector Space properties
1. Closure under vector space
2. Closure under scalar multiplication
3. Associativity of addition
4. Commutativity of addition
5. Identity element of addition
6. Existance of additive Inverse
7. Distributivity
8. Compatibility of scalar multiplications

norm: $V \in \mathbb{Z}^{n\times 1}$

$\|V\| = \sqrt{V_1^2 + V_2^2 + \dots V_n^2}$

4. Boolean algebra

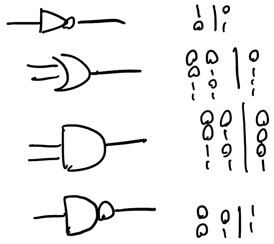
logic operations (AND, OR, NOT, NAND, NOR, XOR, etc.);

'1' - On/active
'0' - off/inactive

truth tables;

$\bar{x}$ not
$x+y$ or
xy and
$\bar{x}$ ... implication

Not
OR
And
Nand



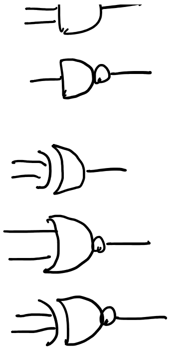
truth tables;

xy and

$\bar{x}+y$ implication

$x \oplus y$ exclusive or

$x \equiv y$ equivalence



And	1	0	0
	1	1	1
Nand	0	0	1
	0	1	1
	1	0	1
	1	1	0
XOR	0	0	0
	0	1	1
	1	0	1
	1	1	0
NOR	0	0	1
	0	1	0
	1	0	0
	1	1	0
XNOR	0	0	1
	0	1	0
	1	0	0
	1	1	1

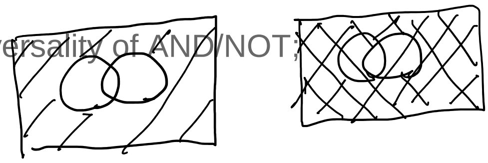
DeMorgan's rule;

x	y	$\bar{x}$	$x+y$	xy	$x \Rightarrow y$	$x \oplus y$	$x \equiv y$
0	0	1	0	0	1	0	1
0	1	1	1	0	1	1	0
1	0	0	1	0	0	1	0
1	1	0	1	1	1	0	1

parity and XORs;

$$(A \cup B)' = A' \cap B'$$

universality of AND/NOT;



Parity - Check for errors during data transmission/storage
- Odd parity - make total # of ones odd (chc. parity)
- Even parity - make total # of ones even " "

universality of NAND;

XORs - Output the inverse of input

- functional completeness
  - ↳ any logical or arithmetic operation can be expressed as a combination of only And & not operations

sum-of-product;

- functional Completeness
  - ↳ same as above

product-of-sum

- equivilant to logical AND function

3. Let **x** be a vector in R^N, and let U be a unitary matrix that can operate on **x**. What is the 2-norm of **x**? What is the 2-norm of U**x**?

$f(A,B) = A'B' + A'B + AB' + AB$

output true for a specific input

- equivilant to logical OR function

$f(A,B) = (A+B)(A+B')(A'+B)(A'+B')$

output false for a specific input

4. Code search() that takes an input list and a keying function f as input, and behaves as described in the lecture.

$x \in \mathbb{R}^{N\times 1}$

$||x||^2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}$

$||Ux||^2 = ||x||^2$

5. Code sort() that takes an input list and a two-input compare function f as input, and behaves as described in any computer algorithms resource.

```
def search(input_list, f):
    for x in input_list
        if f(x):
            return x
    return
```

6. Let u be a 5-bit signed binary word. Design the function sqrt(u), that returns the square root of u to two bits of precision (and full dynamic range). Show the truth table and the logic equations.

Truth Table:

U	Binary	Sqrt(u)	Sqrt(u) Binary
0	00000	0	00
1	00001	1	01
2	00010	1	01
3	00011	1	01
4	00100	2	10

- just doing Bubble sort

```
def sort(input_list, f):
    length = len(input_list)
    for i in range(length - 1):
```

2	00010	1	01
3	00011	1	01
4	00100	2	10
5	00101	2	10
6	00110	2	10
7	00111	3	11
8	01000	3	11
9	01001	3	11
10	01010	3	11
11	01011	3	11
12	01100	3	11
13	01101	4	11
14	01110	4	11
15	01111	4	11

```
length = len(input_list)
for i in range(length -1):
    for j in range(0, length -i-1):
        if f(input_list[j], input_list[j+1]):
            swap(input_list[j], input_list[j+1])
return input_list
```

Logic eqn

$Y_1 =$ if U_1 or U_2 or U_3 or U_4 are true $\rightarrow Y_1 =$ true

$\Rightarrow$ only cases here are U_2 U_3

$$\boxed{Y_1 = U_2 + U_3}$$

Y_0 all except

- all zero $\rightarrow$ NAND
- $\bar{U}_0 \bar{U}_1 U_2 \bar{U}_3$
- $U_0 \bar{U}_1 U_2 \bar{U}_3$
- $\bar{U}_0 U_1 U_2 \bar{U}_3$

$$\Rightarrow Y_0 = (\bar{U}_0 \bar{U}_1 U_2 \bar{U}_3) \cdot (U_0 \bar{U}_1 U_2 \bar{U}_3) \cdot (\bar{U}_0 U_1 U_2 \bar{U}_3) \cdot (U_0 U_1 U_2 U_3)$$

Logic Eqns:

$U_4 U_3 U_2 U_1 U_0$ $Y_1 Y_0$

$Y_0 =$