i=0

j=0

pic[0] [0]
pic[0] [1]

0000

possible cells

11

EW S N
8 7 6 5 4 3 2 1 0

Prop		
0	No-slip- obstacle →	10
1	free slip Obstacle →	100
2	outflow obstacle →	1000
3	Inflow obstacle →	10000
4	Fluid →	0 0 0 0 0 0 0 0 1

Now I need to check for these 4 typed obstacle further divided into some types acc. to flags / ie pressure, water on which direct.

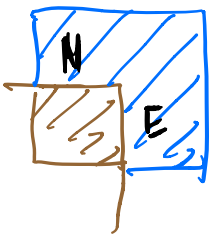
① → 0 0 1 0 (to the South Water)

② → 0 0 0 1 (to the north Water) bottom wall

③ → 1 0 0 0 (to the east water)

④ → 0 1 0 0 (to the West Water)

⑤ → 1 0 0 1



	8	7	6	5	4	3	2	1	0
South water =	0	0	1	0	0	0	0	0	0
+ north water =	0	0	0	1	0	0	0	0	0
+ east water =	1	0	0	0	0	0	0	0	0
+ West water =	0	1	0	0	0	0	0	0	0

N (OR) E

S (OR) E

S (OR) W

N (OR) W

2^8

— Iterate overall loop elements (Not Fluid)

① if (flg (i)(j) ≠ 1)
flag [i] [j] =

flag [i] [j] (→ as value)

+ (flag [i+1] [j]) * 1 (East cell)

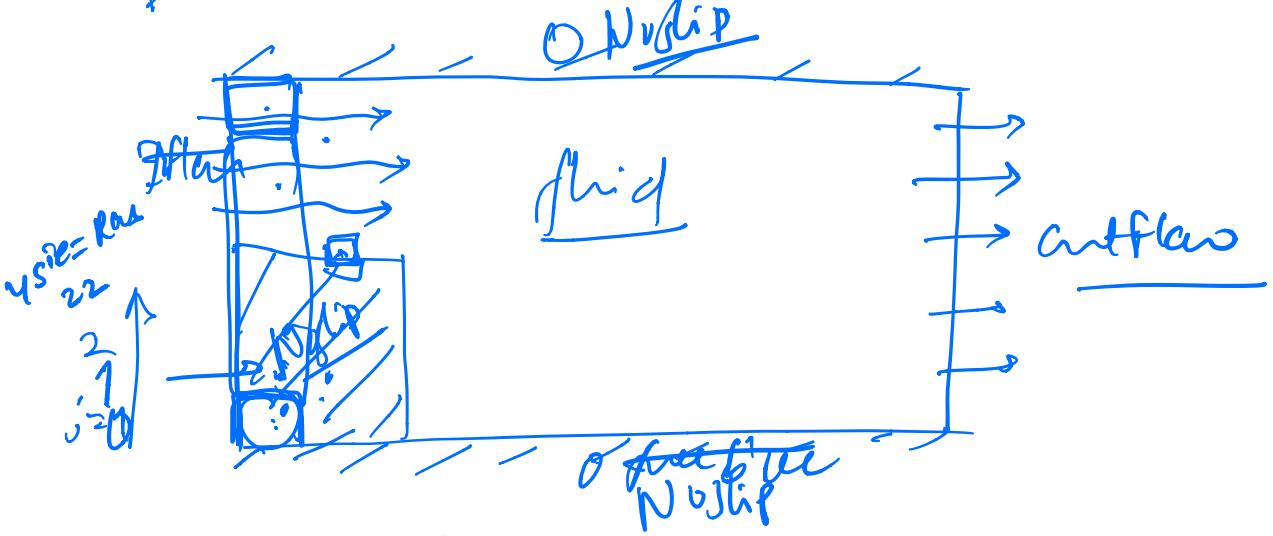
⑥ → 1 0 1 0

⑦ → 0 1 1 0

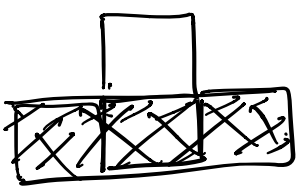
⑧ → 0 1 0 1

+ (flag[i-1][j]) + 1( )
+ (flag[i][j+1]) + 1( )
+ (flag[i][j-1]) + 1( )

256
+
266



eg.



if flag[i][j] == 4
 dum[i][j] = 1
else dum[i][j] = 0

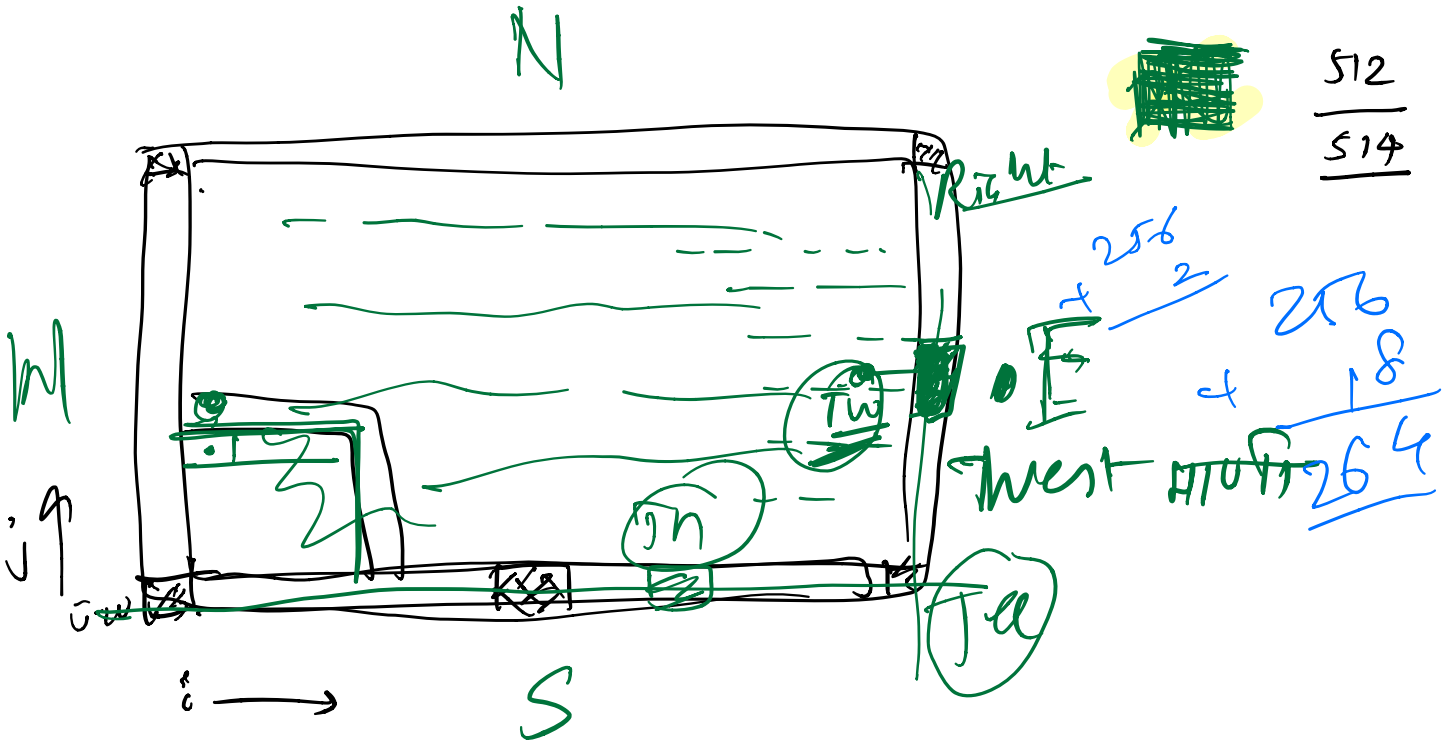
DJT

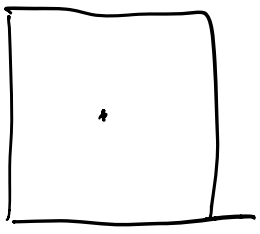
```
{
    if (flag[i+1][j] == 4
     flag[i][j] = flg[i][j] + East cell.
    else if (flag[i-1][j] == 4
     flag[i][j] = flag[i][j] + west cell.
    else if
}
```

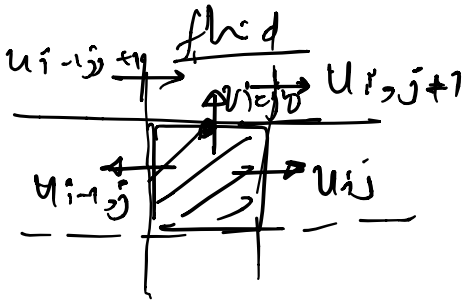
[i+1][j] East
[i-1][j] West
[i][j+1] North
[i][j-1] South

Possible values of Flags

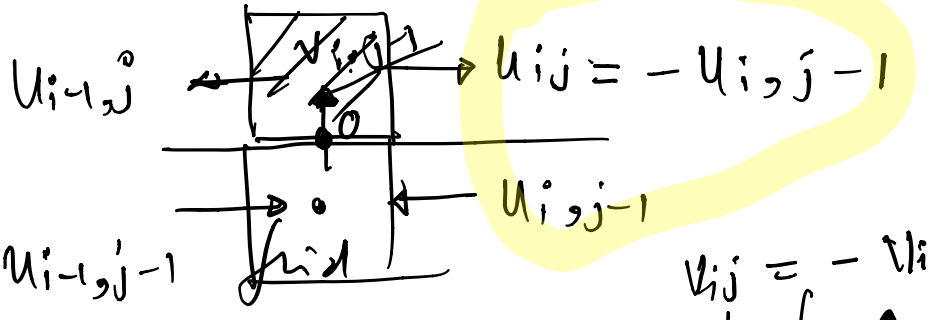
	E W I S N C I O F N F (512 256 128 64 32 16 8 4 2 1)	Decimal		This Prob.
	0 0 0 0 0 0 0 0 0 1	1	→ Fluid (Not want anywhere.)	
	0 0 0 0 0 0 0 0 1 0	2	→ only obstacle (Inner obstacles)	
No slip-obstacles	1 0 0 0 0 0 0 0 1 0	514	→ Water to East with No slip condition →	
	0 1 0 0 0 0 0 0 1 0	258	→ West related	
	0 0 1 0 0 0 0 0 1 0	130	→ water to South with No slip condition →	Top
	0 0 0 1 0 0 0 0 1 0	66	→ water to North with No slip cond^n. →	Bottom
	1 0 0 1 0 0 0 0 1 0	578	→ Water to North-East with No slip condition →	one
	1 0 1 0 0 0 0 0 1 0	642	→ water to South-East →	corner cell
	0 1 1 0 0 0 0 0 1 0	386		
	0 1 0 1 0 0 0 0 1 0	322		
Free slip obstacles	1 0 0 0 0 0 0 1 0 0	516		
	0 1 0 0 0 0 0 1 0 0	260		
	0 0 1 0 0 0 0 1 0 0	132		
	0 0 0 1 0 0 0 1 0 0	68		
	1 0 0 1 0 0 0 1 0 0	580		
	1 0 1 0 0 0 0 1 0 0	644		
	0 1 1 0 0 0 0 1 0 0	388		
	0 1 0 1 0 0 0 1 0 0	324		
Outflow condition	1 0 0 0 0 0 1 0 0 0	520		
	0 1 0 0 0 0 1 0 0 0	264	→ Water to West with outflow condition →	Right side
	0 0 1 0 0 0 1 0 0 0	136		
	0 0 0 1 0 0 1 0 0 0	72		
	1 0 0 1 0 0 1 0 0 0	584		
	1 0 1 0 0 0 1 0 0 0	648		
	0 1 1 0 0 0 1 0 0 0	392		
	0 1 0 1 0 0 1 0 0 0	328		
Inflow condition	1 0 0 0 0 1 0 0 0 0	528	→ Water to East with Inflow condition →	left side
	0 1 0 0 0 1 0 0 0 0	272		
	0 0 1 0 0 1 0 0 0 0	144		
	0 0 0 1 0 1 0 0 0 0	80		
	1 0 0 1 0 1 0 0 0 0	592		
	1 0 1 0 0 1 0 0 0 0	656		
	0 1 1 0 0 1 0 0 0 0	400		
	0 1 0 1 0 1 0 0 0 0	336		
Coupling Boundary	1 0 0 0 1 0 0 0 0 0	544		
	0 1 0 0 1 0 0 0 0 0	288		
	0 0 1 0 1 0 0 0 0 0	160		
	0 0 0 1 1 0 0 0 0 0	96		
	1 0 0 1 1 0 0 0 0 0	608		
	1 0 1 0 1 0 0 0 0 0	670		
	0 1 1 0 1 0 0 0 0 0	416		
	0 1 0 1 1 0 0 0 0 0	352		


N
W
E
S
TW
TN
TE
j
i

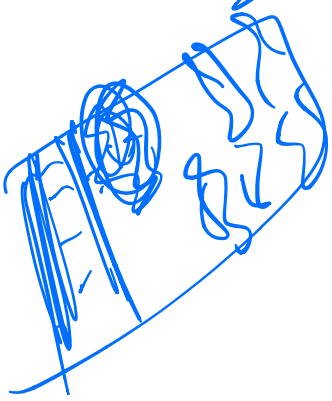


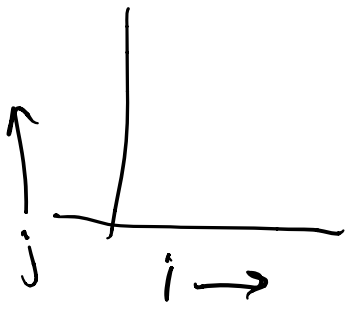

fluid
Ui-1,j+1
Ui,j+1
Ui-1,j
Uij

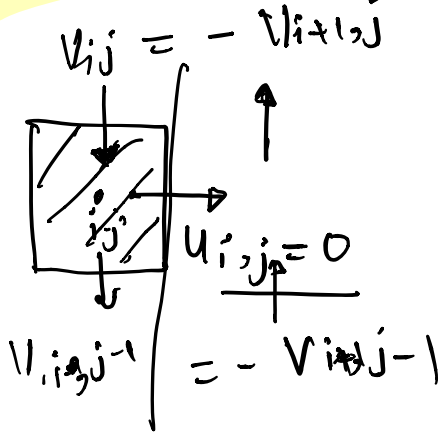
BS


Ui-1,j
Ui,j = - Ui,j-1
Ui,j-1
Ui-1,j-1

BN



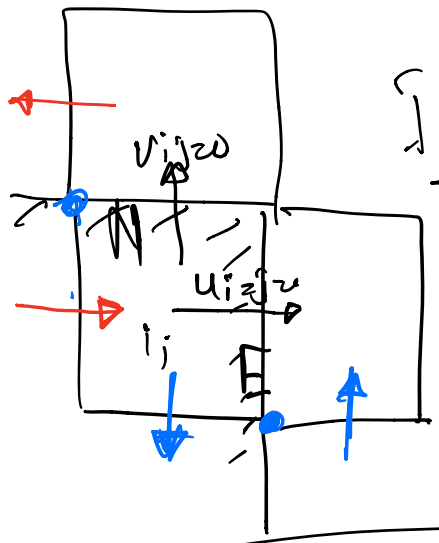

j
i


Vij = - Vi+1,j
Ui,j = 0
Ui,j-1 = - Vi+1,j-1

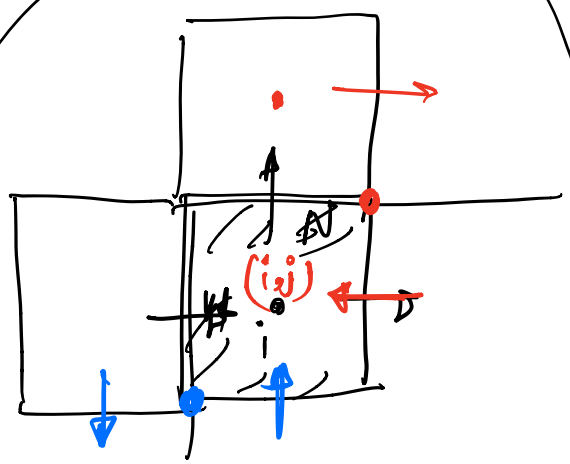
BO

East

NE



NW

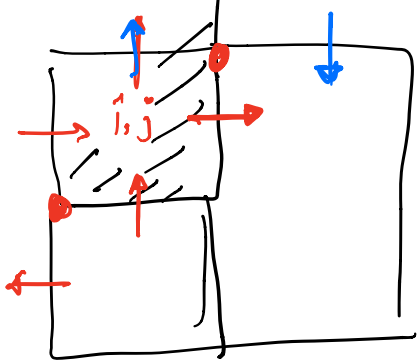


$v_{ij} = u_{i-1,j} = 0$

$v_{ij} = -v_{i,j+1}$

$v_{i,j-1} = -v_{i-1,j-1}$

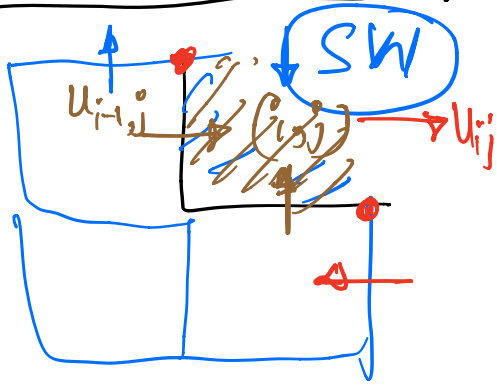
SE



$u_{ij} = 0$, $u_{i-1,j} = -u_{i-1,j-1}$

$v_{i,j-1} = 0$ $v_{ij} = -v_{i+1,j}$

SW



$u_{i-1,j} = 0$

$v_{i,j-1} = 0$

$u_{ij} = -u_{i,j-1}$

$v_{ij} = -v_{i-1,j}$

1) Calculate dt → Addition of Pr
2) Temp(T) matrix initialization in Main and addition in init.c
3) addition of extra parameters in (Th, Tc, β) in read parameter.
4) addⁿ of parameter in Dat problem file
5) adding Temp. boundary conditions in boundcal.c
6) adding calculate_Temp in UVP.c
7) Solⁿ of every eqⁿ.

→ Read the problem from user

- Squash - small commits
- Merge request / discuss.

(10 * 1) - 20

10 - 10

or

12

20 - 20

t = 0
t = 10
t = 20
t = 30
t = 40

0
12
16
18
20

$10 - (12 / 19)$

$(1 \cdot 32 - 9)$

22

24

30

32

0.

0

1

1

1

1

10.001

10.0.0

$h = 100$

$t/10$ $10 = 10.045$

$0.045 < 0.1$

$0.008 <$

For DAT file

Temp independent Puller

Pr = 1
beta = 0
allT = 0

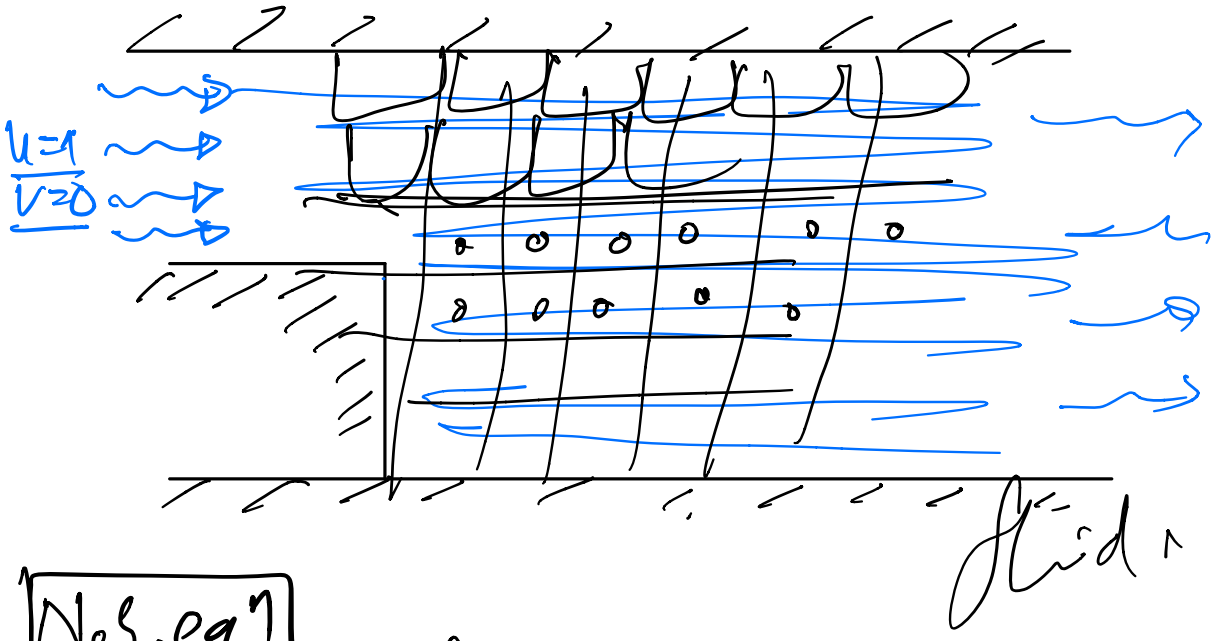
$$\frac{\partial u}{\partial t} = \left[-\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + (1-\beta T) g_x \right.$$
$$\left. \underbrace{-\frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y}} \right]$$
$$\Downarrow$$

$$\frac{u_{i,j}^{n+1} - u_{ij}^{n}}{dt} =$$

$$u_{ij}^{n+1} = dt\left[-\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u + \partial^2 u}{\partial x^2 \; \partial y^2} \right) + (1-\beta T) g_x \right.$$
$$\left. -\frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} \right] + u_{ij}^{n}$$

$$= dt\left(-\frac{\partial p}{\partial x} \right) + \frac{dt}{Re}\left(\frac{\partial^2 u + \partial^2 u}{\partial x^2 \; \partial y^2} \right)$$

$$u_{ij}^{n+1} = \underbrace{f_{ij}^{n+1}}_{F_{ij}} + \frac{\delta t}{\delta x}\left(p_{i+1,j}^{n+1} - p_{ij}^{n+1}\right)$$

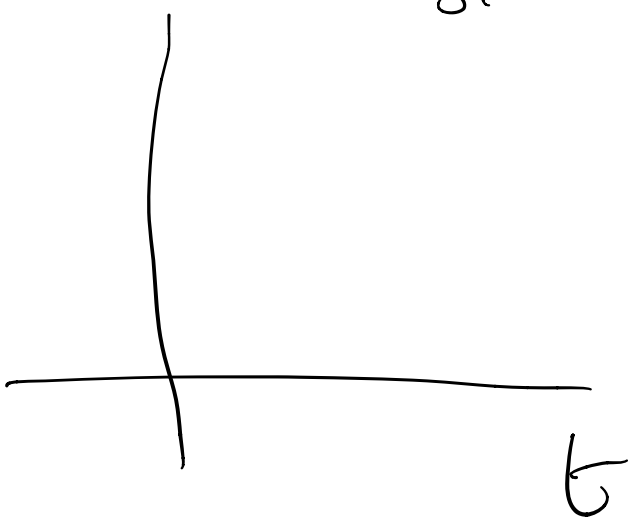


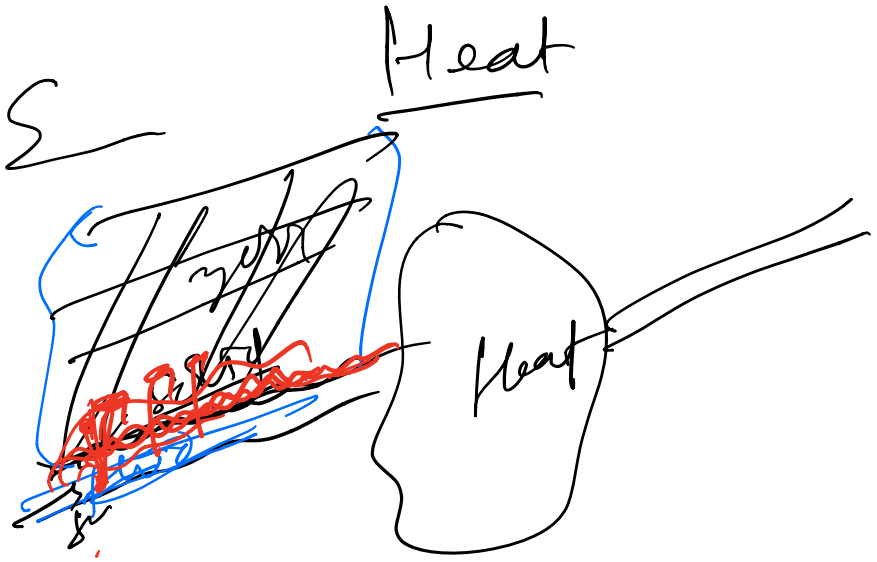
No S. eqn

① u, no
v no
continuity

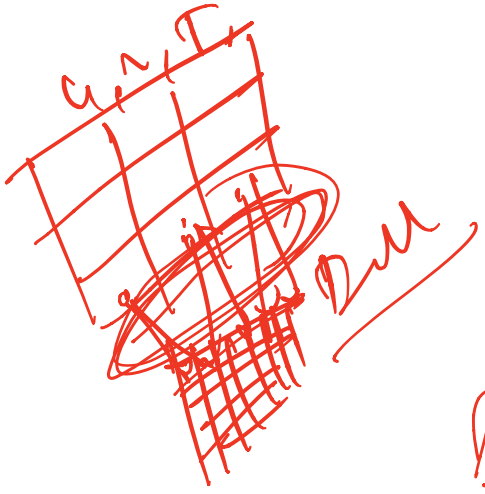
solid Dynamics

St

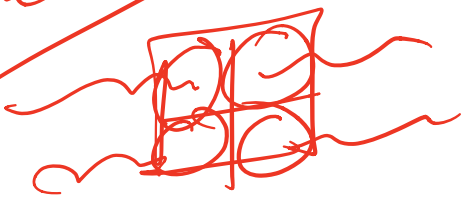


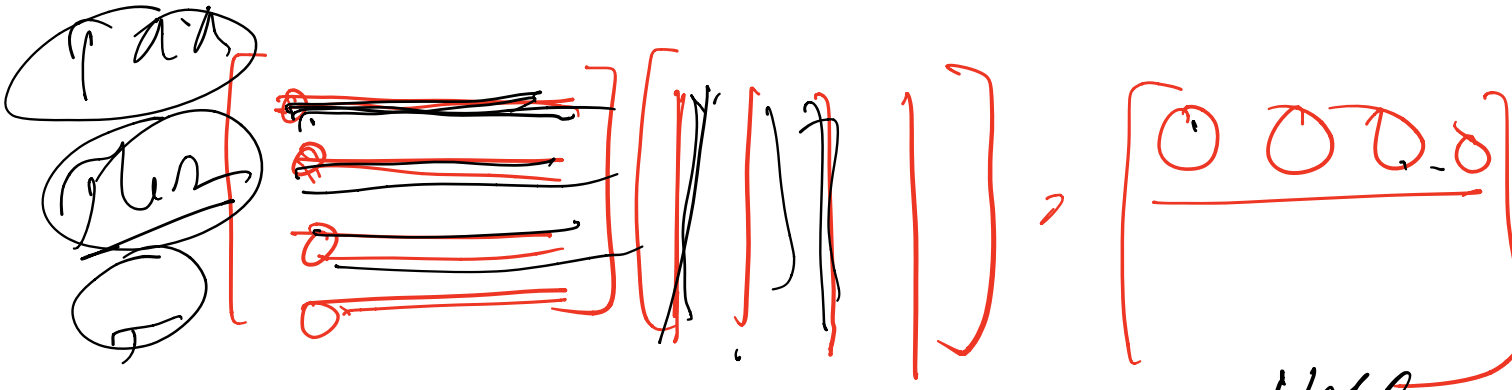


FSI - Fluid structure Interaction



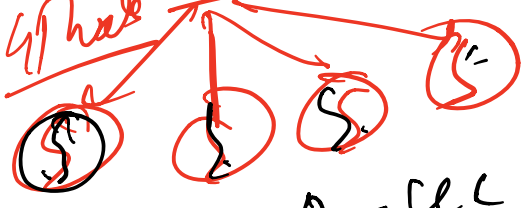
Parallel Pre CICE



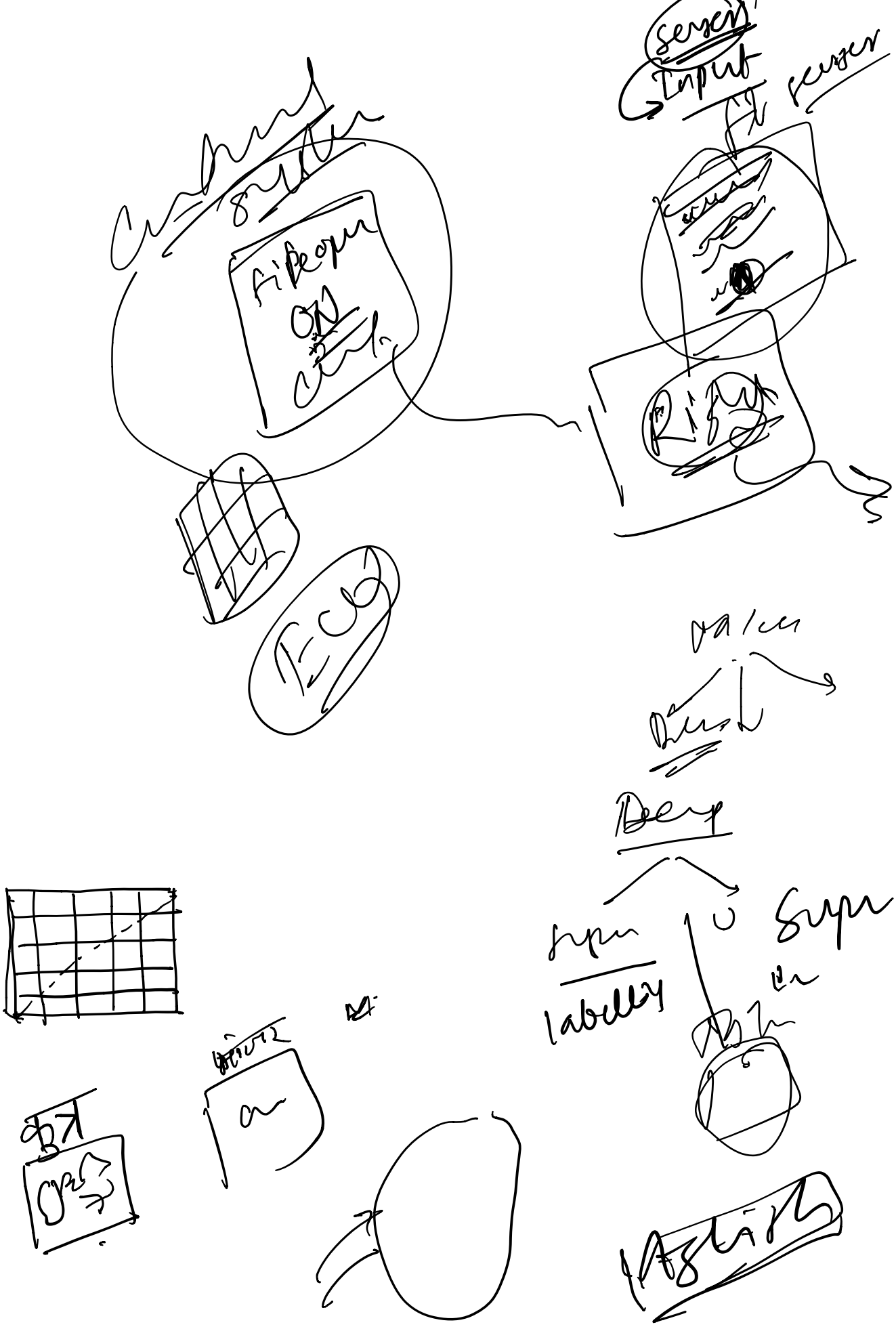


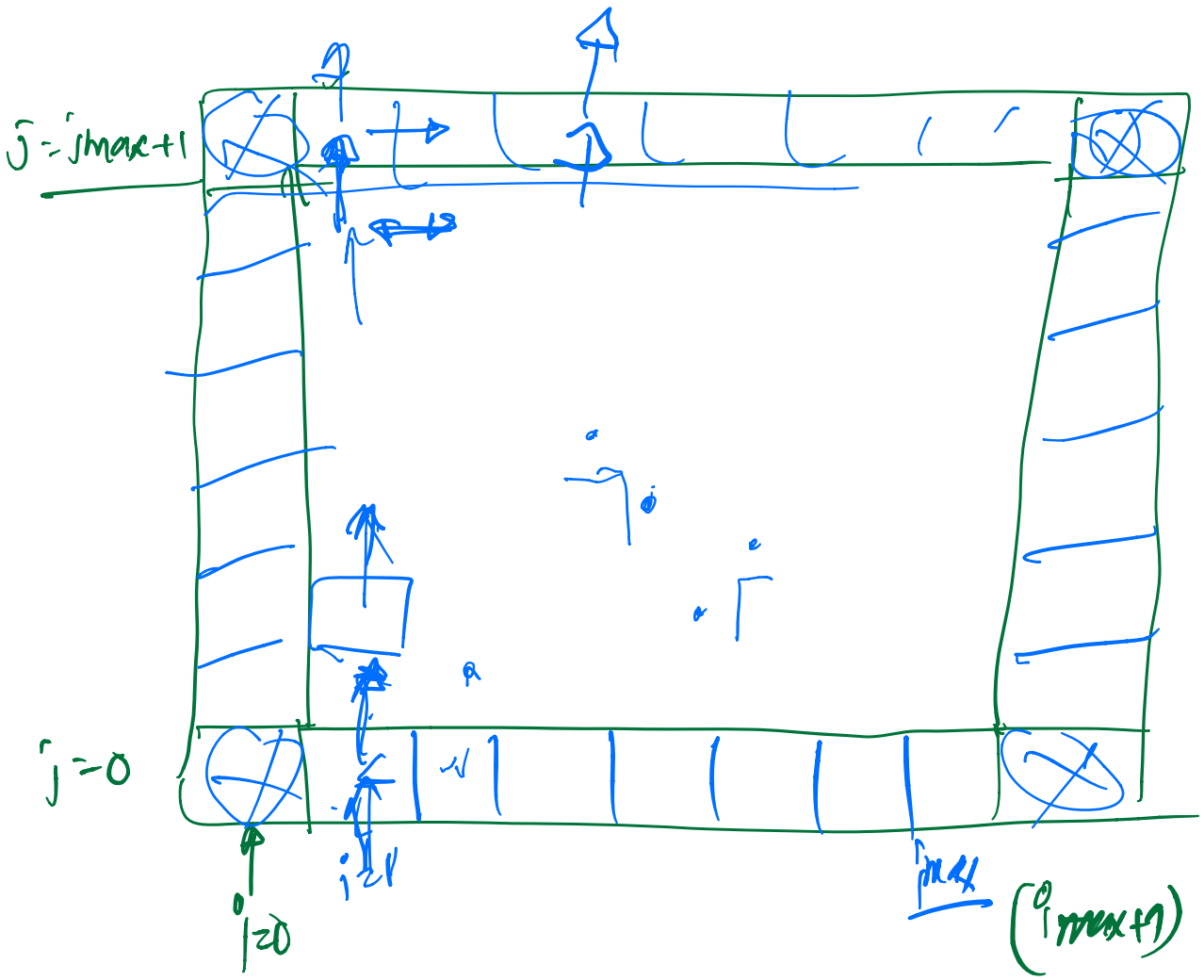
Core

Computer, CPU, Chip



4 sec




j = jmax+1
j=0
i=0
(imax+1)