Navier - Stokes Equation:

$\nabla \cdot u = 0$ ⟶ divergence of velocity = 0 ✳ ✳

not possible

mass conserved

$\rho \frac{Du}{Dt} = -\nabla p + \mu \nabla^2 u + \rho F$

gravity, walls, winds

high P ↓ low P

for incompressible, Newtonian fluids

acceleration depends upon - pressure gradient - viscosity - external forces.

↑ viscosity = higher friction between fluid particles which means velocity in fluid in one part will more easily spread out to the surrounding areas

They give only specification, not implementation

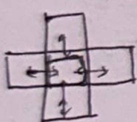
Representation:

#* (eulerian)

particles that move around (lagrangian) / grid of stationary regions with attributes like velocity, density, temperature, that represent avg of above prop. regarding particles that would be in each grid square.

Diffusion:

⟶ n't dependent upon velocities in the fluid



when attributes of each part of the fluid, spreads into surrounding area.

actual calculation involves making each squares value gradually become average value of the squares surrounding it.

Diffusion of density:

$d(x,y)$ - density of square at x,y

$s(x,y)$ - average

$$= \frac{d(x+1,y) + d(x-1,y) + d(x,y+1) + d(x,y-1)}{4}$$

we gradually want to make

$d(x,y) \rightarrow s(x,y)$

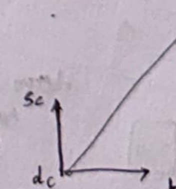
k = amount of change

$\square_c$ = value from current iteration (known)

$\square_n$ = from next iteration (unknown)

$d_n = d_c + k(s_c - d_c)$

(linear interpolation)



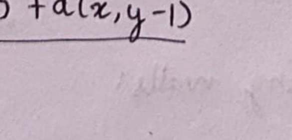
problem when $k > 1$

up & down unexpected jump in values

so

$d_c = d_n - k(s_n - d_n)$

$d_n = \frac{d_c + k s_n}{1+k}$ → hyperbolic equation



→ problem s_n?

until converge to true values

assume values of unknown as 0 & go on doing iterations

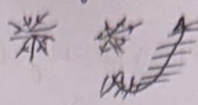
→ expand - find solut^n using iterative solver - gauss - seidel method

Advection: movement of attributes, following it's velocity.

Curl

Divergence → means mass created / destroyed

means



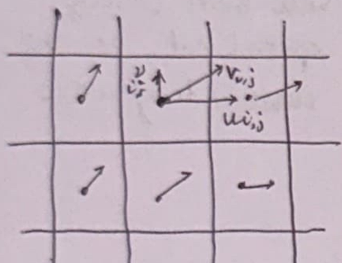
vector field of fluid = curl-free + divergence-free

↳ we calculate this & subtract from (vector field of fluid)

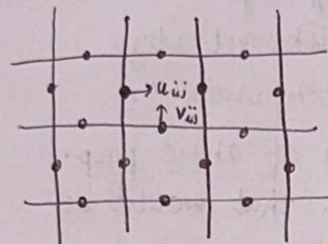
↳ we want this (divergence-free)

Fluid as a velocity field on a grid:

$v = \begin{bmatrix} u \\ v \end{bmatrix}$



collocated grid



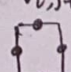
staggered grid
we store velocity in different locatn:
- horizontal components stored at center of vertical cell faces
- vertical in horizontal cell faces
- using this we can see how much fluid flows from one cell to it's neighbour

Fluid sim simulation:

1. modify velocity values. eg. add gravity
2. make fluid incompressible (projection)
3. move the velocity field (advection)

1. Update velocity
for all i, j
$v_{i,j} \leftarrow v_{i,j} + \Delta t \cdot g$

2. Divergence (total outflow)



$d \leftarrow u_{i+1,j} - u_{i,j} + v_{i,j+1} - v_{i,j}$

if d = +ve = outflow
d = -ve = inflow
zero = incompressible

★ how to force incompressibility
- fluid can't change only one velocity
- it can change all by same amount

$u_{ij} \leftarrow u_{ij} + d/4$; $u_{i+1,j} \leftarrow u_{i+1,j} - d/4$;

$v_{ij} \leftarrow v_{ij} + d/4$; $v_{i,j+1} \leftarrow v_{i,j+1} - d/4$

Measuring pressure

for all i,j
$p_{ij} \leftarrow 0$
for n iterations
for all i,j
$p_{ij} \leftarrow p_{ij} + \frac{d}{s} \cdot \frac{\rho h}{\Delta t}$ (ρ → density, h → grid spacing)

Gauss Seidel → very ↑ iteratn
so multiply d by o
1 < o < 2
1.9 = o - nice value

this thing in
for n iterations
for all i,j
// code

How do we do it:
new variable s
= 0 for ~~walls~~ blocked cells
1 for else

$s_{i+1,j}$ $\boxed{s_{i-1}}$

$s \leftarrow s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1}$

$u_{ij} \leftarrow u_{ij} + d \times \frac{s_{i-1,j}}{s}$

$u_{i+1,j} \leftarrow u_{i+1,j} - d \times \frac{s_{i+1,j}}{s}$

$v_{ij} \leftarrow v_{ij} + d \times \frac{s_{i,j-1}}{s}$

$v_{i,j+1} \leftarrow v_{i,j+1} - d \times \frac{s_{i,j+1}}{s}$

Handling walls:

$\cancel{u_{ij}}$

$u_{i+1,j} \leftarrow u_{i+1,j} - d/3$

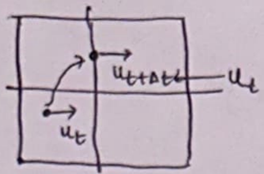
$v_{ij} \leftarrow v_{ij} + d/3$

$v_{i,j+1} \leftarrow v_{i,j+1} - d/3$

$u_{ij} = 0$ for walls
$\neq 0$ for moving objects

3. Advection:

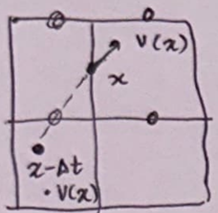
Semi Lagrangian Advection

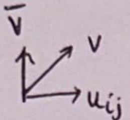


- which fluid particle moved to the location where u is stored
- set new velocity

$$u_{t+\Delta t} \leftarrow u_t$$

How to know previous locat^n:





4°

$$\bar{v} = \frac{(v_{ij} + v_{i,j+1} + v_{i-1,j} + v_{i-1,j+1})}{4}$$