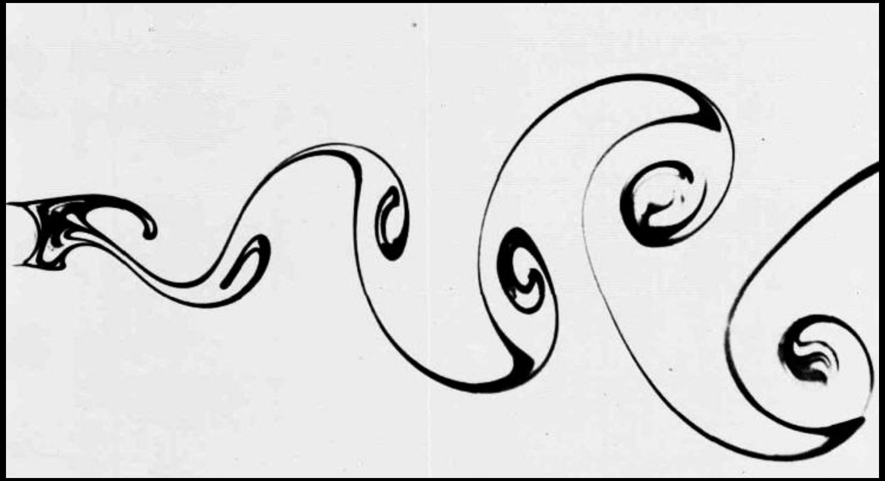
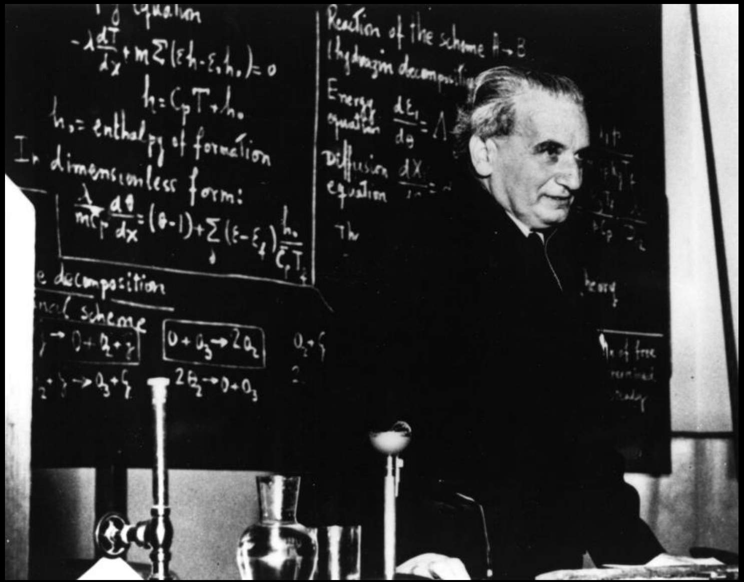
Computer modeling of physical phenomena

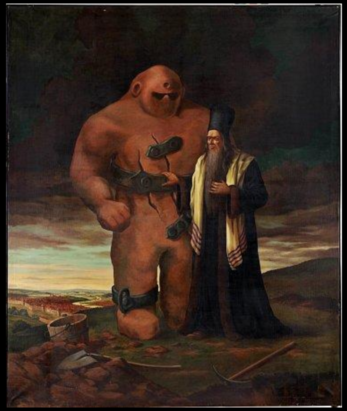


Lab VIII: Vortex shedding

Theodore von Kármán...



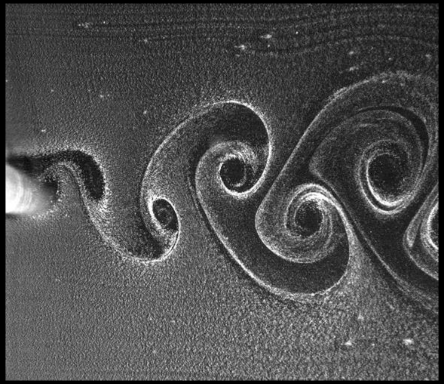
Theodore von Kármán
(Tódor Kármán), 1889-1963



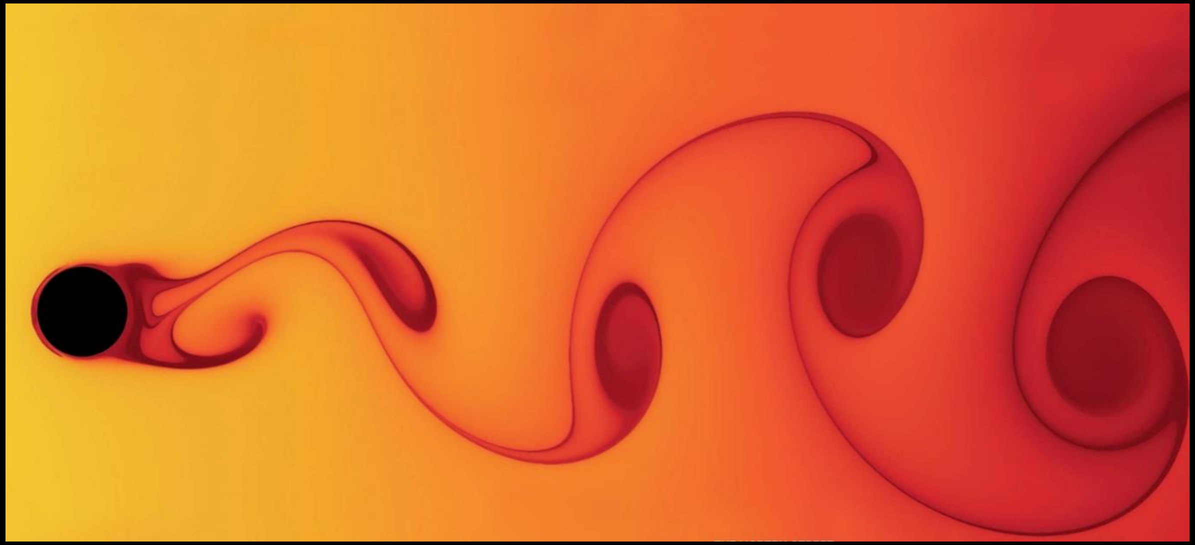
Judah Löwe ben Bezalel,
(1520 – 1609) and his Golem

..and his street

KTH sculpture

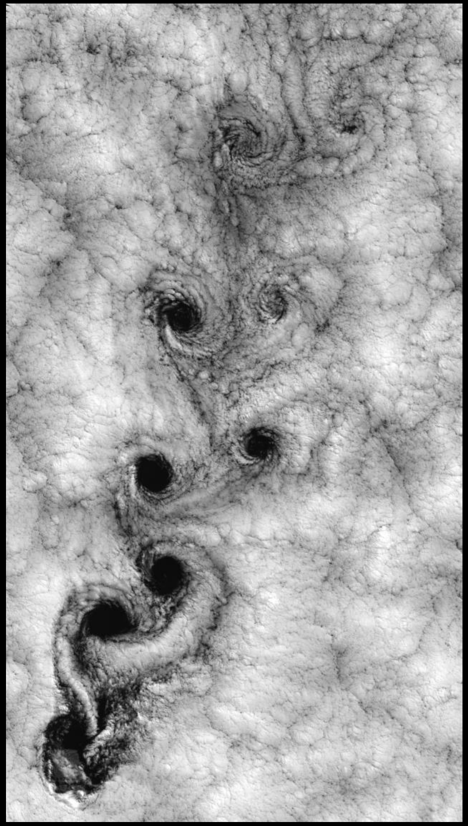


experiment: 6mm cylinder in water



simulations

Often observed behind islands

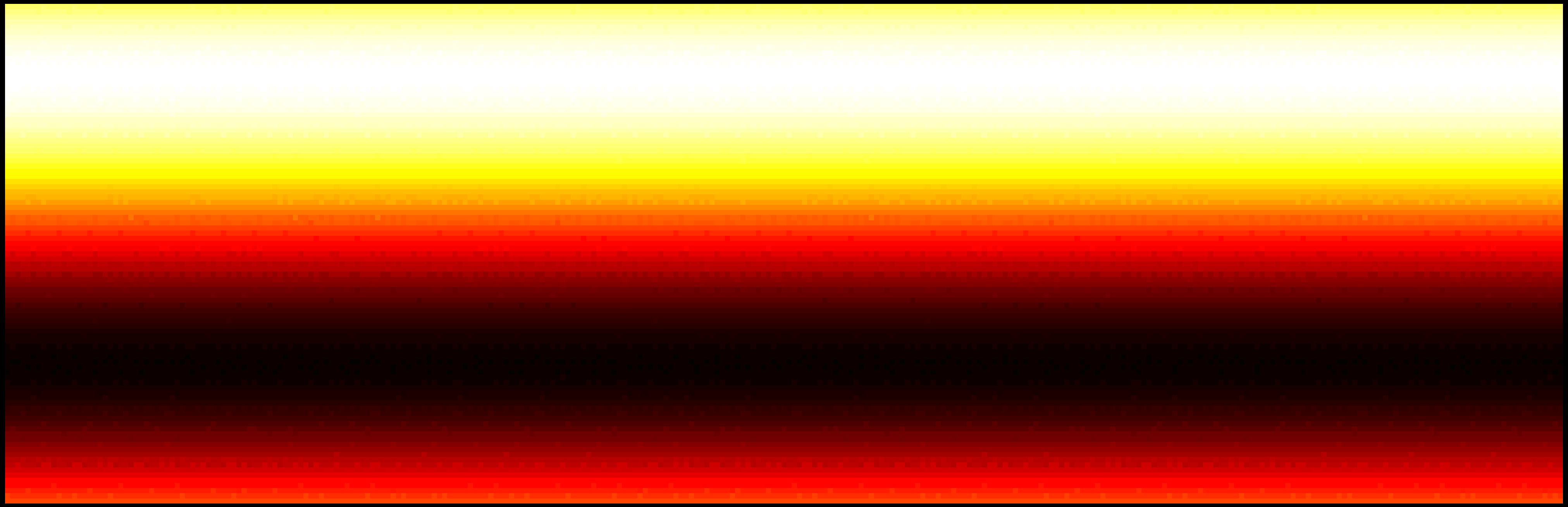


Juan Fernandez Islands

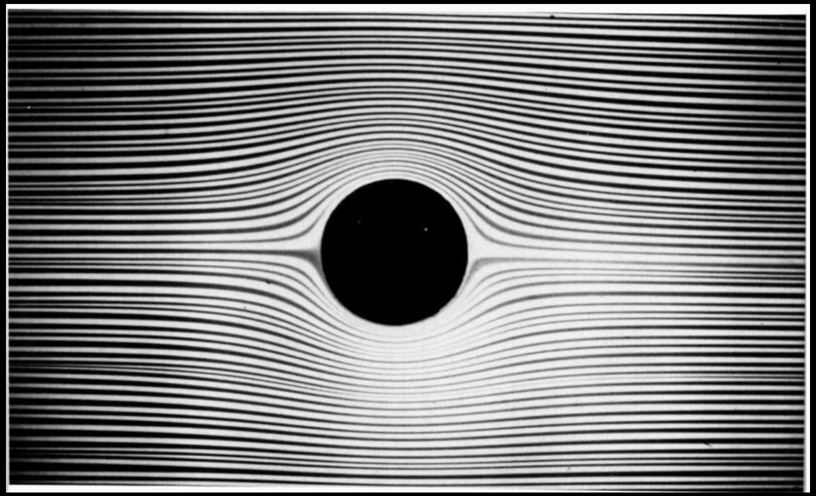
Guadalupe Islands

Cape Verde Islands

Vortex street video

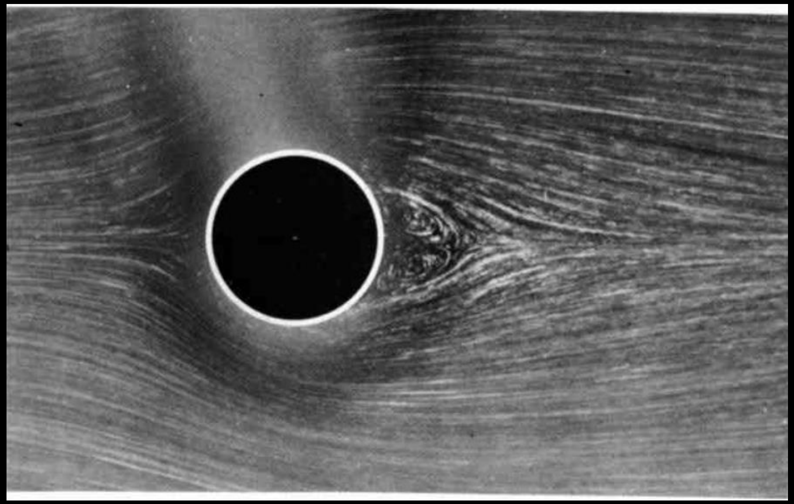


Flow past a cylinder



Re = 0.2

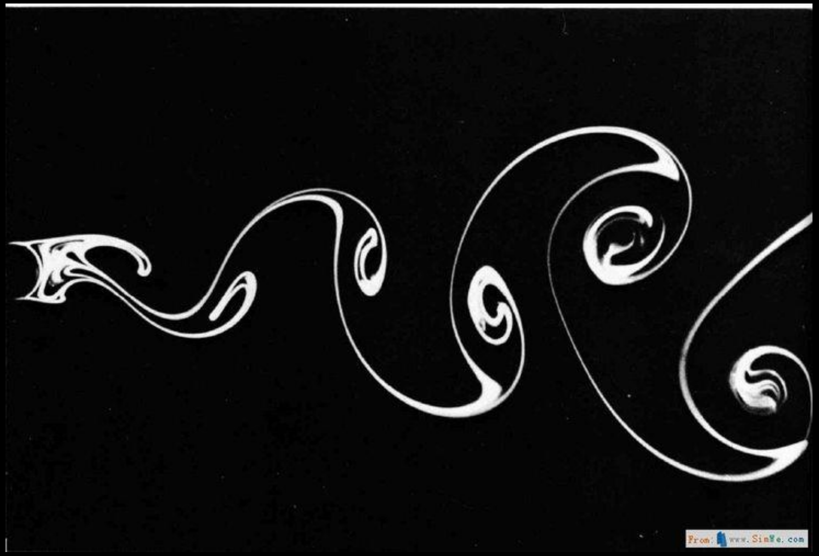
Re = 2.7



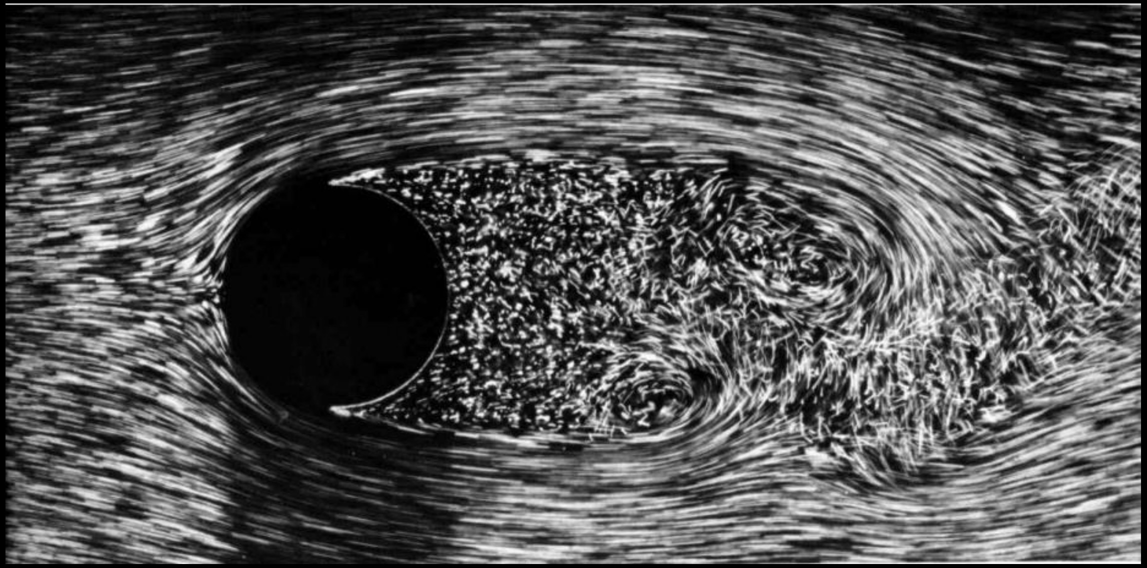
Re = 13

Re = 26

Flow past a cylinder

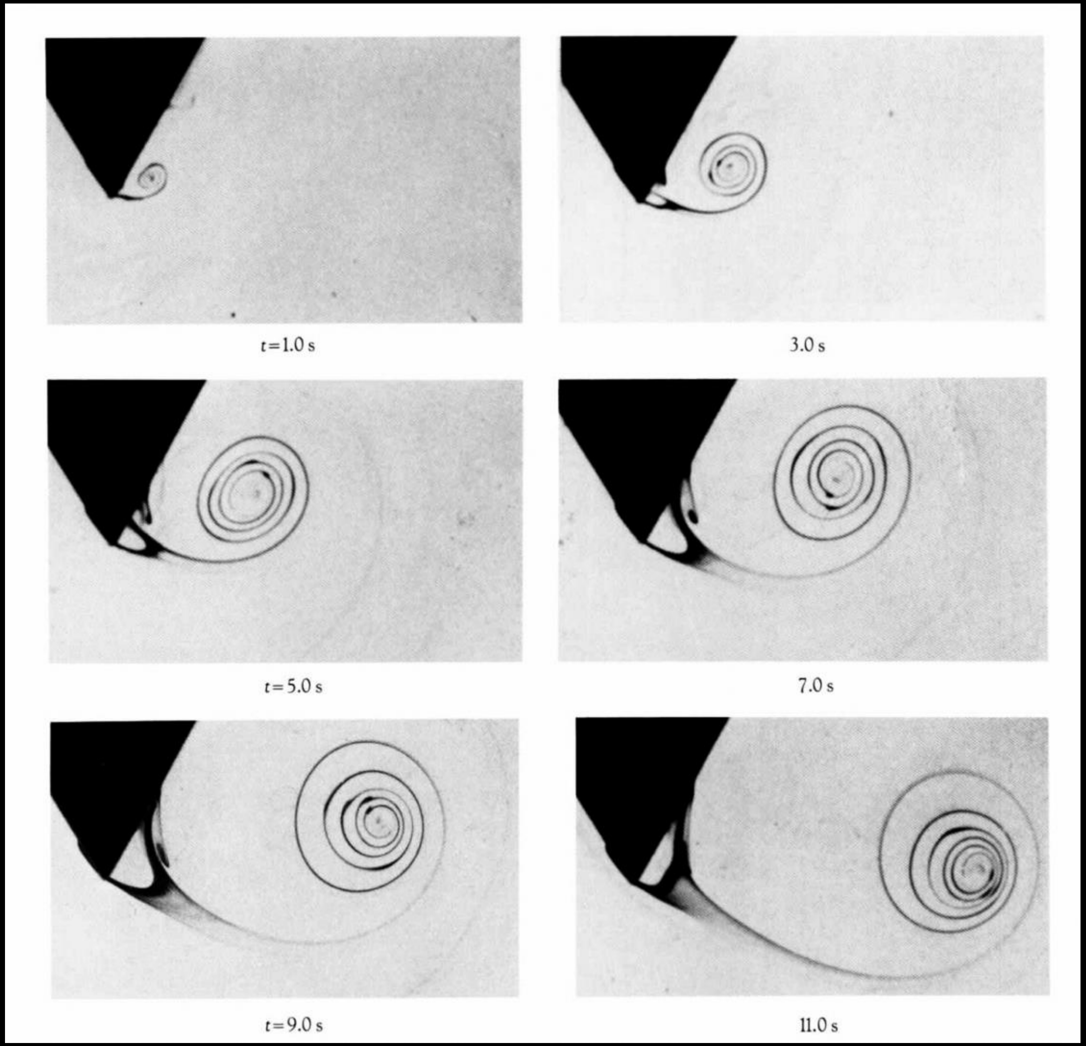


Re = 140

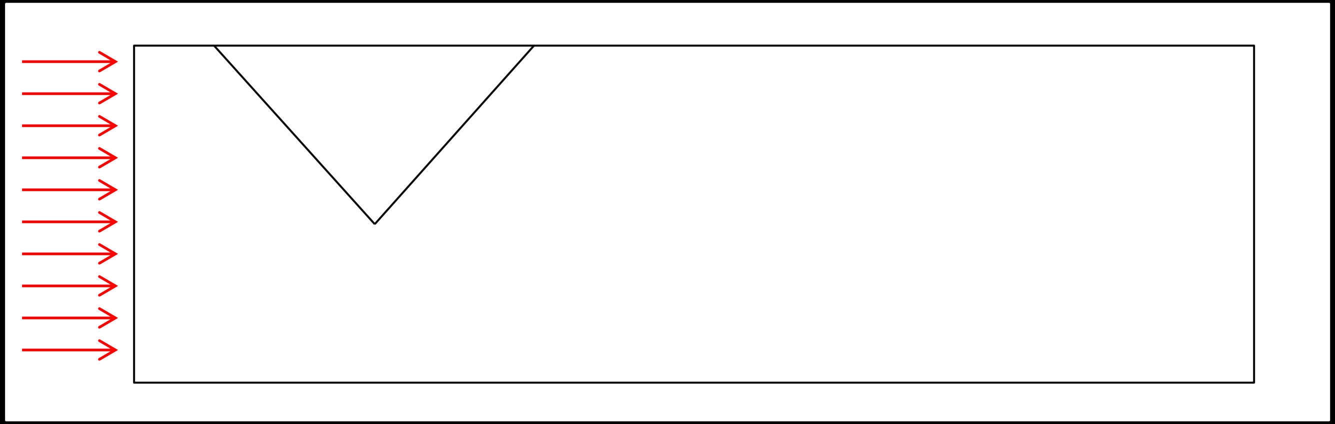


Re = 2000

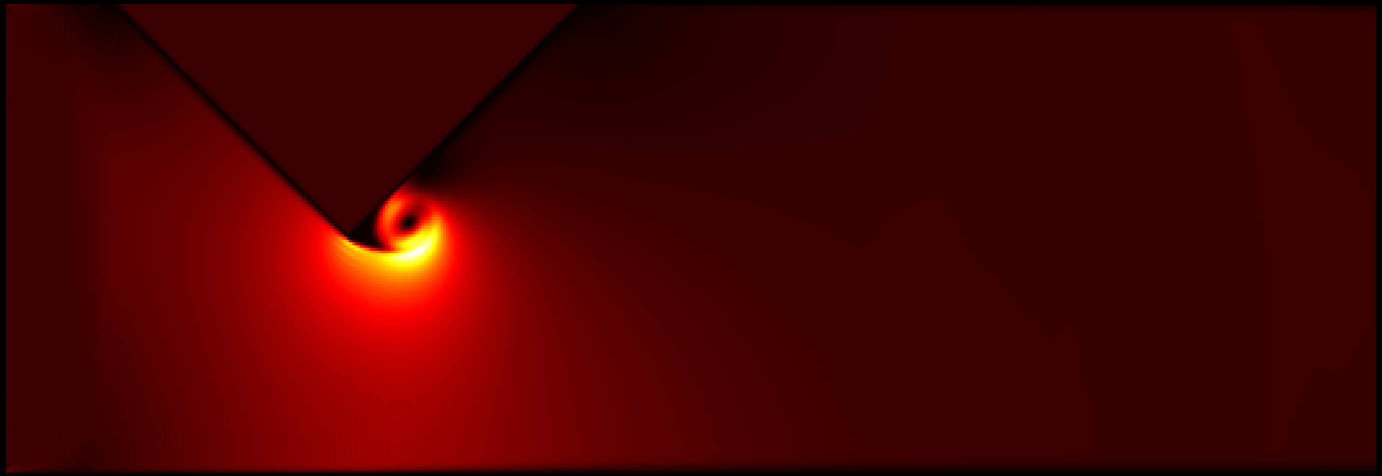
Vortex shedding behind a wedge



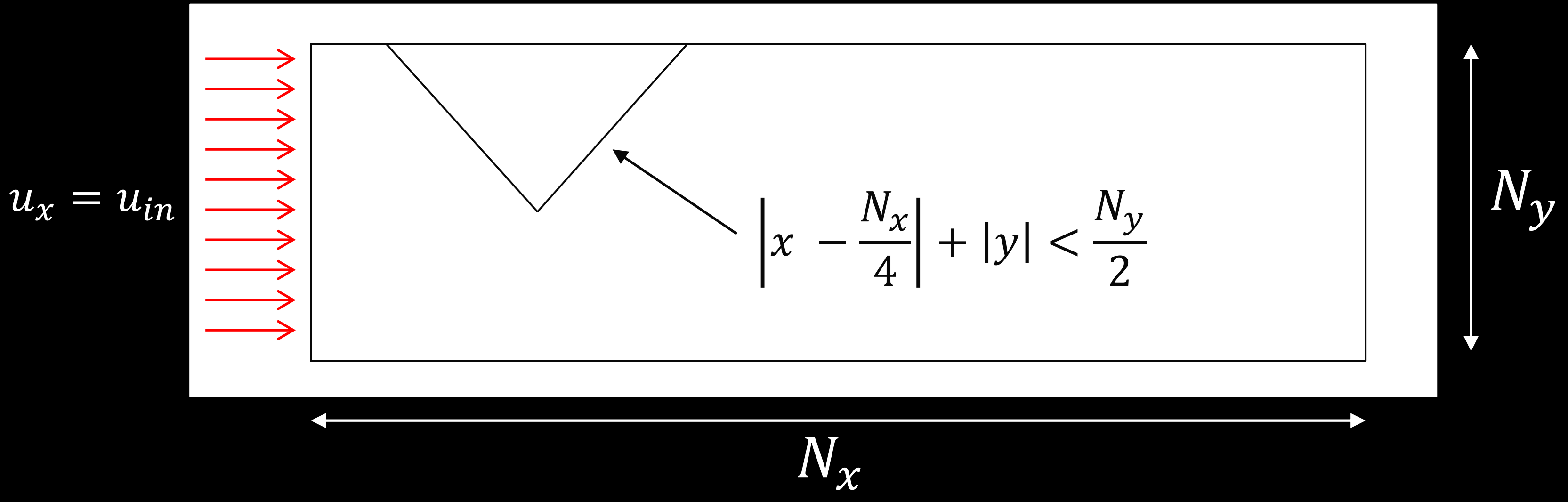
Your task



Simulate vortex shedding behind an obstacle using LBM.



Your task



$$u_{in} = 0.04$$ velocity in lattice units

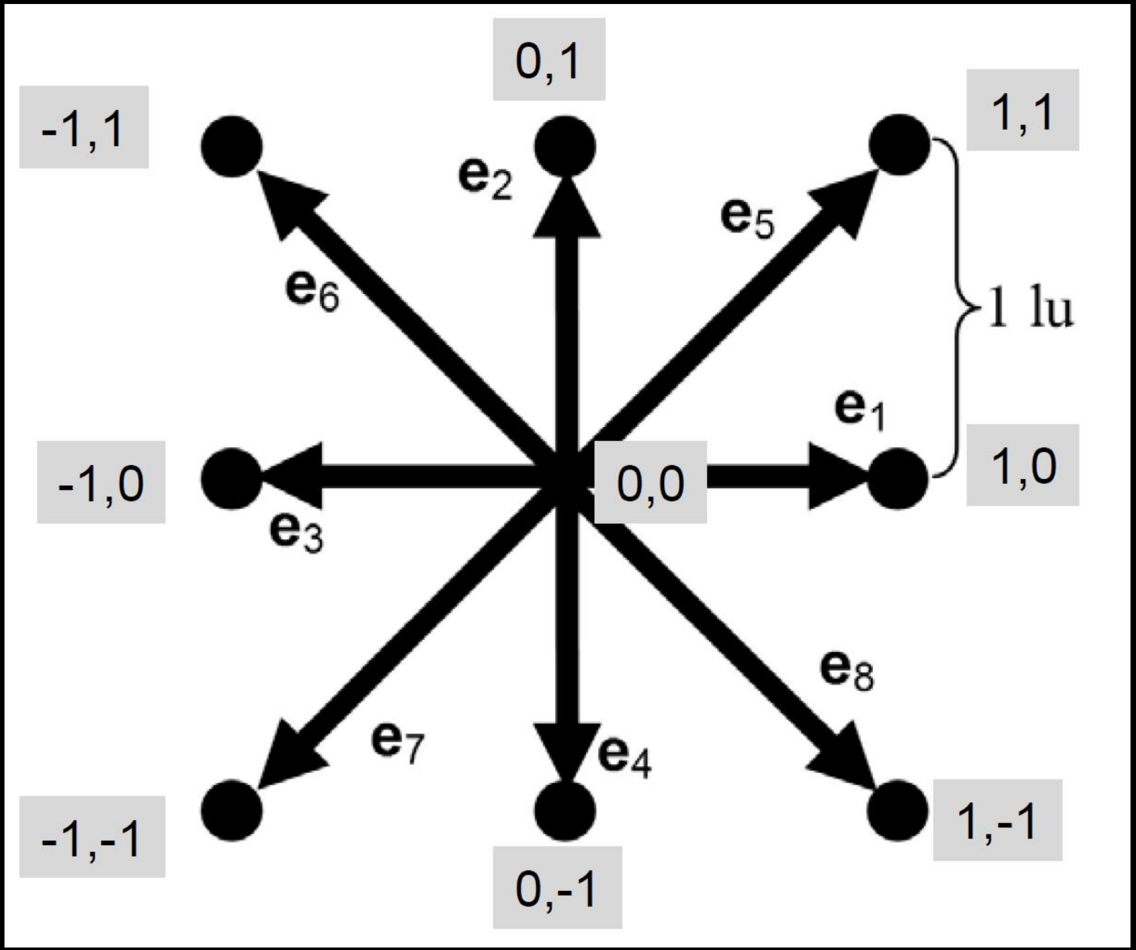
$$Re = 1000$$ Reynolds number

$$N_x = 520$$
$$N_y = 180$$
system size

$$\nu_{LB} = \frac{u_{in}(N_y/2)}{Re}$$ viscosity

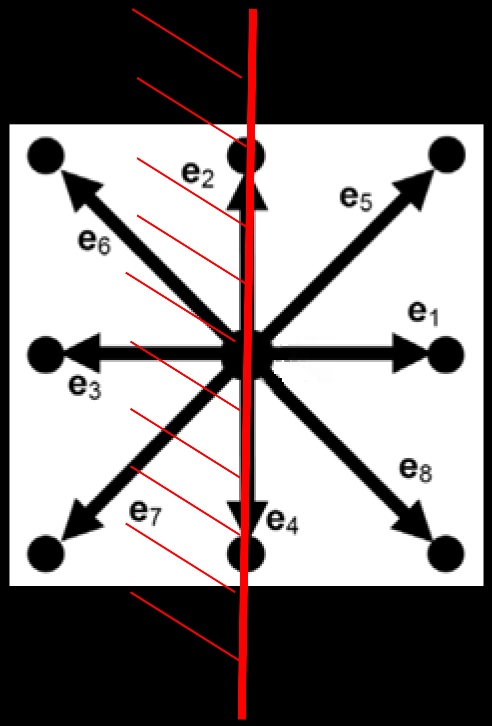
$$\tau = 3\nu_{LB} + \frac{1}{2}$$ relaxation time

Velocity vectors



e = [[0, 0], [1, 0], [0, 1], [-1, 0], [0, -1], [1, 1], [-1, 1], [-1, -1], [1, -1]]

Inlet boundary conditions



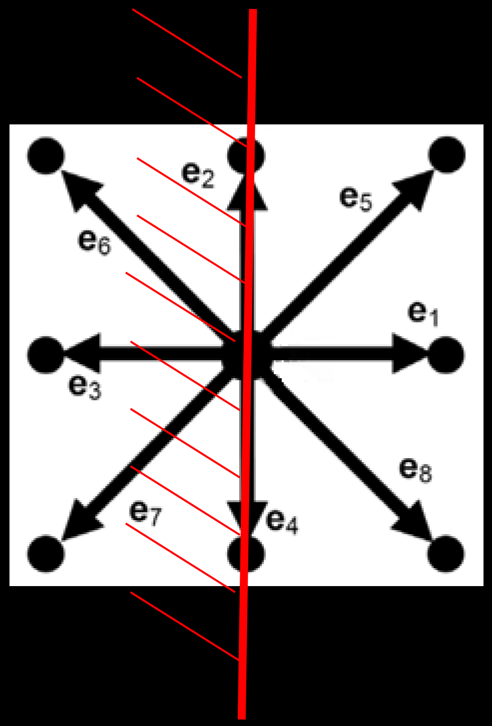
- the populations f_1, f_5, f_8 are unknown after streaming
- additionally, the inlet density (ρ) is unknown – we need 4 eqs.

$$\rho = \sum_{a=0}^{8} f_a \qquad \mathbf{u}_0 = \frac{1}{\rho}\sum_{a=0}^{8} f_a\,\mathbf{e}_a$$

along x: $$\rho u_0 = f_1 - f_3 + f_5 - f_6 - f_7 + f_8$$

subtracting both: $$\rho = \frac{2(f_3 + f_6 + f_7) + (f_0 + f_2 + f_4)}{1 - u_0}$$

Inlet boundary conditions



As for the unknown populations, f_1, f_5, f_8, one can simply reduce them to their equilibrium values:

$$f_1 = f_1^{eq}$$
$$f_5 = f_5^{eq}$$
$$f_8 = f_8^{eq}$$

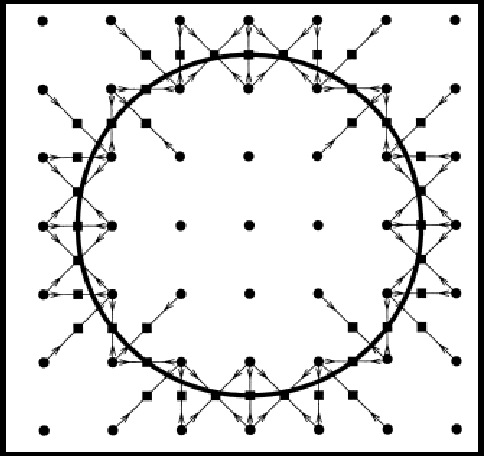
Other boundaries

upper & lower boundaries - reflecting (bounce-back)
right boundary - outflow condition:

$$f_3(N_x, y) = f_3(N_x - 1, y)$$
$$f_6(N_x, y) = f_6(N_x - 1, y)$$
$$f_7(N_x, y) = f_7(N_x - 1, y)$$

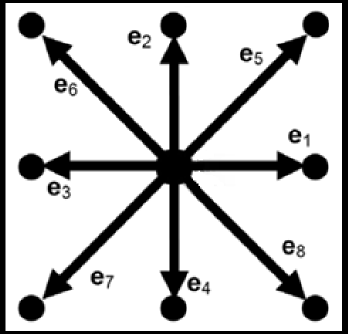
Collisions with the cylinder

Reverse the velocities of particles that hit the solid.



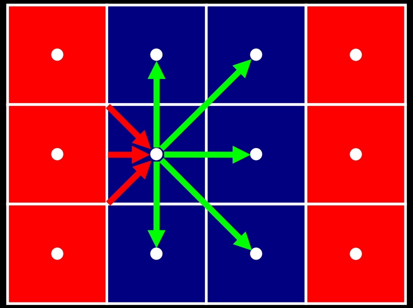
Compute the reverse indices of the directions:

e: [0, 1, 2, 3, 4, 5, 6, 7, 8] -> [0, 3, 4, 1, 2, 7, 8, 5, 6]



Apply the bounce-back rules in the solid:

$$f_i^{col}(obstacle) = f_{reverse(i)}(obstacle)$$



Initialization

Initialize velocity to be equal $\mathbf{u}_0$ everywhere. This corresponds to a slight perturbation in a uniform inflow, which helps to break the symmetry of the flow and produce the first vortex behind the wedge.

$$\mathbf{u}_0 = u_{in}[1 + \epsilon \sin(\frac{y}{2\,\pi\left(N_y - 1\right)})]\ \mathbf{e}_\mathbf{x}$$

$\epsilon = 0.0001$ velocity perturbation

Calculate equilibrium distribution f^{eq} based on this velocity and unit density $\rho_0 = 1$. Use it as initial distribution (f).

Algorithm

1. Calculate the density and equilibrium distribution on the inlet.

$$\rho(x=0) = \frac{2(f_3+f_6+f_7)+(f_0+f_2+f_4)}{1-u_0}$$

$$f_i^{eq}(x=0) = W_i\,\rho(x=0)\,[1+3(\mathbf{e}_i\cdot\mathbf{u}_0)+\frac{9}{2}(\mathbf{e}_i\cdot\mathbf{u}_0)^2-\frac{3}{2}\mathrm{u}_0^2]$$

2. Apply the boundary conditions for the inlet and outlet.

$$f_{1,5,8} = f_{1,5,8}^{eq}$$

$$f_{3,6,7}(N_x,y) = f_{3,6,7}(N_x-1,y)$$

3. Recalculate density and equilibrium distribution everywhere.

$$\rho = \sum_{a=0}^{8} f_a$$

$$f_i^{eq} = W_i\rho[1+3(\mathbf{e}_i\cdot\mathbf{u})+\frac{9}{2}(\mathbf{e}_i\cdot\mathbf{u})^2-\frac{3}{2}\mathrm{u}^2]$$

VERTE!

Algorithm

4. Calculate the collision step.

$$f_i^{col} = f_i - (f_i - f_i^{eq})/\tau \qquad \tau = 3\nu_{LB} + \frac{1}{2}$$

5. Replace parts of the distribution function after collision that correspond to fluid-solid boundaries with bounce-back condition.

$$f_i^{col}(obstacle) = f_{reverse(i)}(obstacle)$$

6. Calculate streaming step.

$$f_i = stream(f_i^{col}, direction = e_i)$$

7. Use distribution function after streaming as the initial one for the next iteration (go back to 1).

Tips

- You *really* need loops over the directions array – you don't want to rewrite every step of the algorithm 9 times!

  e = [[0, 0], [1, 0], [0, 1], [-1, 0], [0, -1], [1, 1], [-1, 1], [-1, -1], [1, -1]]

- Get friendly with slices – they make life so much better in this task!

$$f^{col}[i,:,:] = \ \ldots$$

- Streaming is extremely easy! Just use 2D rolls in the right directions.

- An array of reverse directions is very handy for collisions and not too hard to implement.

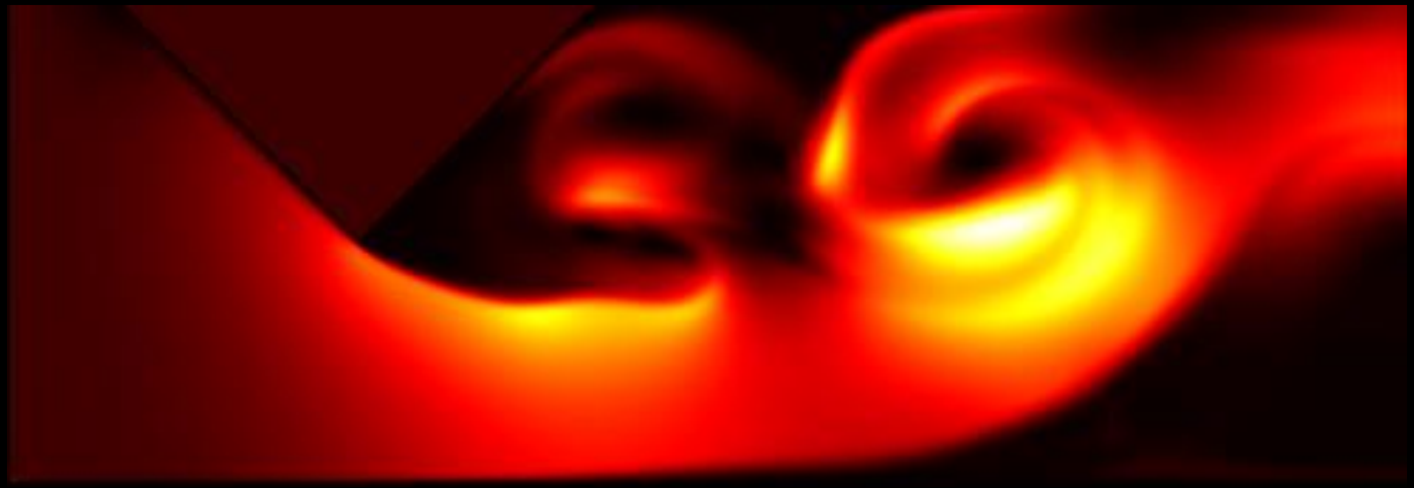
- To get the coordinates of the wedge, you can play with numpy fromfunction.

  wedge = np.fromfunction(lambda x, y: conditions for x, y, (size))

- This simulation takes it time to run, but you should get worried if it spends way more than 0.1s per iteration (you can figure how much time 20000 iterations would then take) ;)

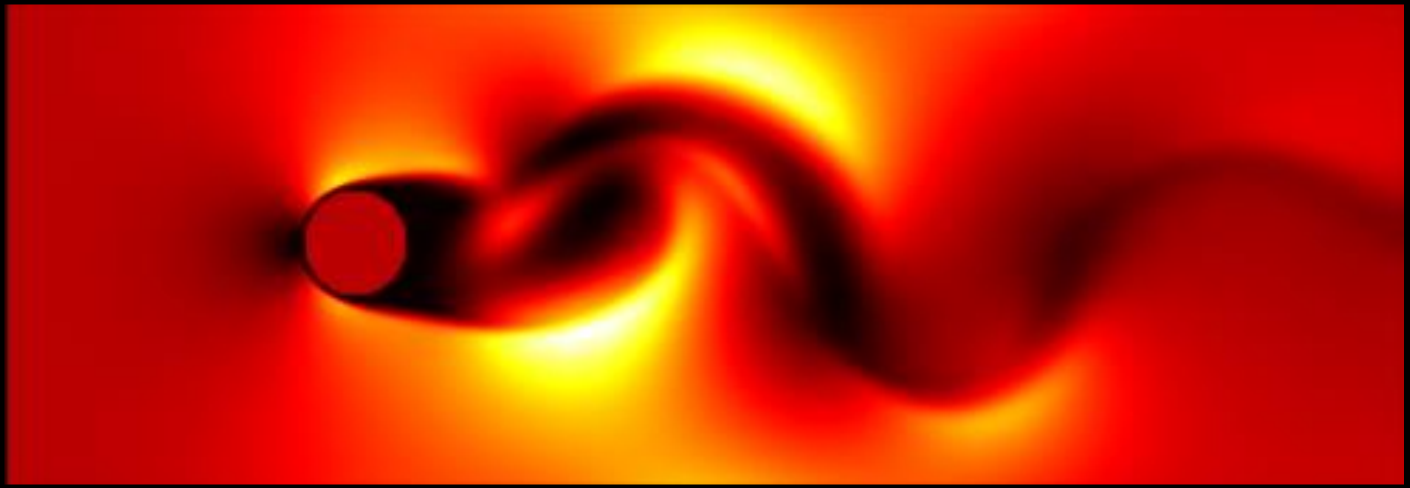
Visualization

Run the simulation for at least 20000 iterations. Plot the velocity magnitude ($|\mathbf{u}|^2$) every 100 steps. Use imshow (with colormap 'hot' it looks *really* cool). Prepare animation illustrating the evolution of the flow.

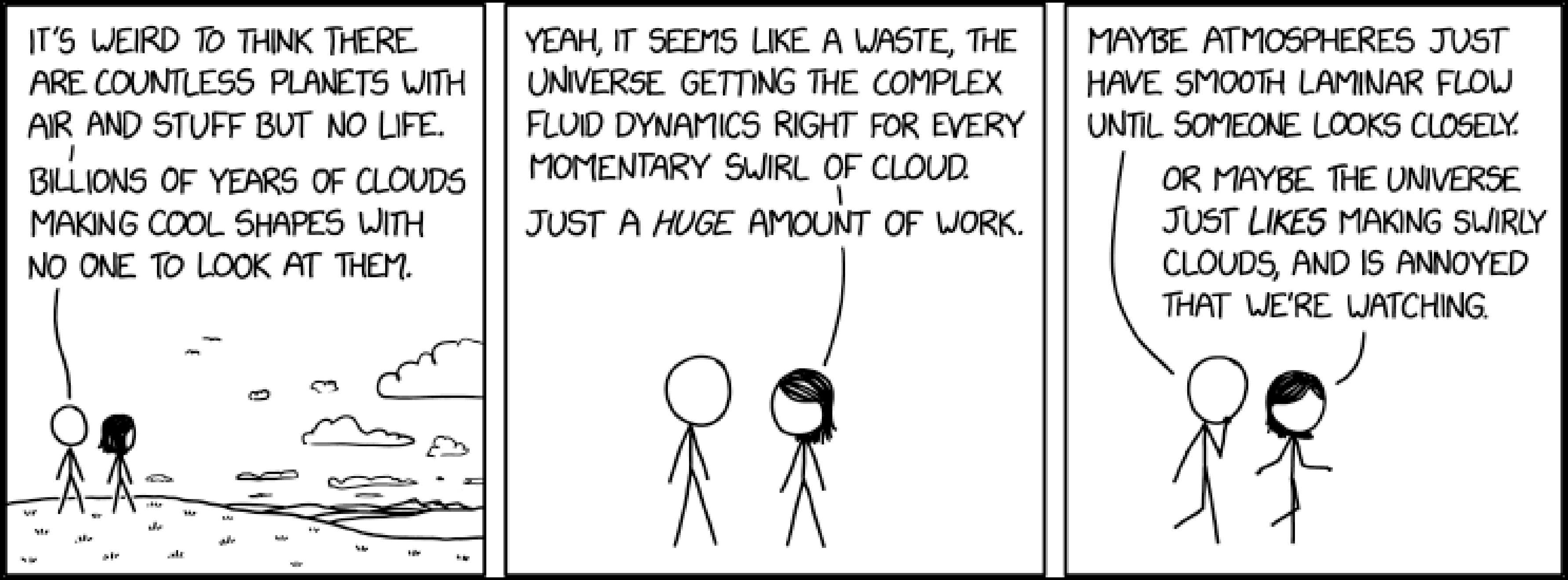


Extra task

Check how the system behaves at smaller and larger Reynolds numbers. Try to find values corresponding to different regimes (creeping flow, attached vortices, vortex trail).



PS If you think the cylinder vortex street looks cooler, it’s extremely easy to switch to it (you may do it instead of the above extra task) – just change the parameters of the obstacle and the bounce-back condition on upper & lower boundary to PBC. Enjoy the beautiful flow patterns!



- Why did you get into fluid dynamics?
- Well, SOME planet has to have the coolest clouds, odds are it's not ours, and rockets are slow.