How to write a Fire Simulator

Matthias Müller, Ten Minute Physics

For the code and the demo see:

www.matthiasmueller.info/tenMinutePhysics

Based on Eulerian Fluid Simulator

See tutorial 17: How to write a Eulerian fluid simulator

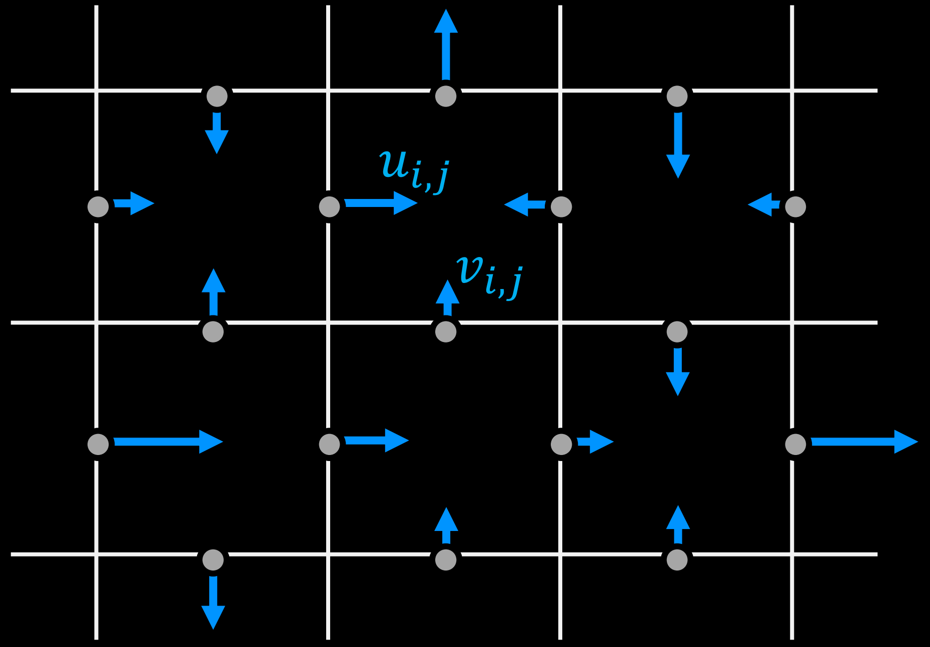
- A fluid is a liquid or a gas

tutorial 17: passive flow

today: active, burning

Fluid as a Velocity Field on a Grid

- Velocity is a 2d vector $\mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix}$



staggered grid

Simulation Overview

1. Modify velocities (add gravity, external forces)
2. Make the fluid incompressible
3. Advect velocity and smoke density fields

Adding Gravitational Acceleration

for all i, j

$$v_{i,j} \leftarrow v_{i,j} + \Delta t \cdot g$$

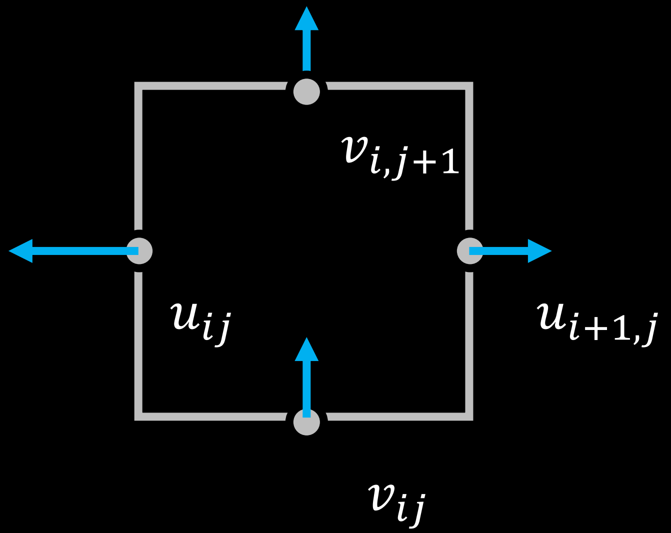
Δt : time step size

g: gravitational acceleration ($\sim 9.81 \frac{m}{s^2}$)

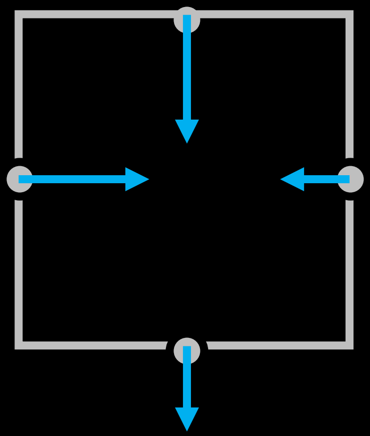
Make Fluid Incompressible

For each cell:

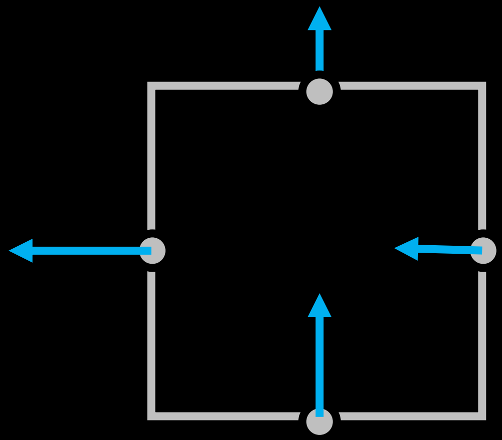
Compute total outflow (divergence): $d \leftarrow u_{i+1,j} - u_{i,j} + v_{i,j+1} - v_{i,j}$



- Positive
- Too much outflow

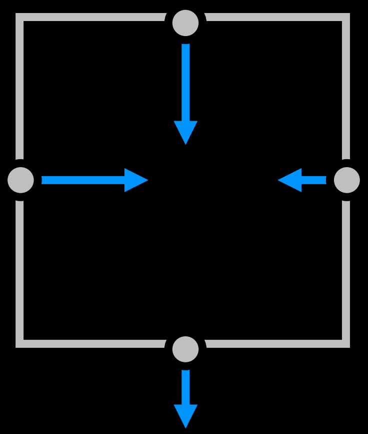


- Negative
- Too much inflow

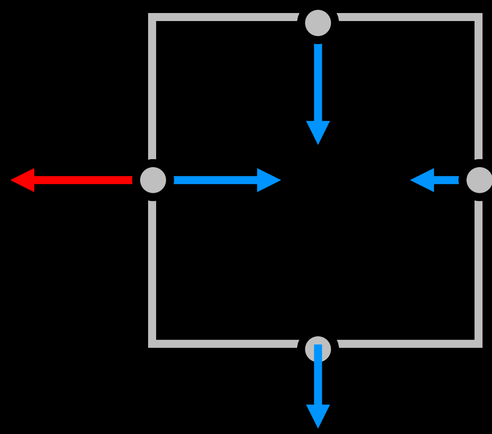


- Zero
- Incompressible

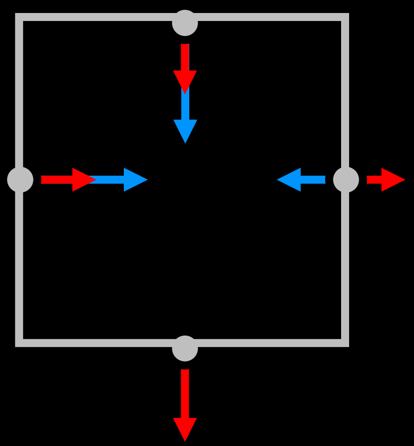
Forcing Incompressibility



Too much inflow



Changing one velocity
A fluid cannot do that!

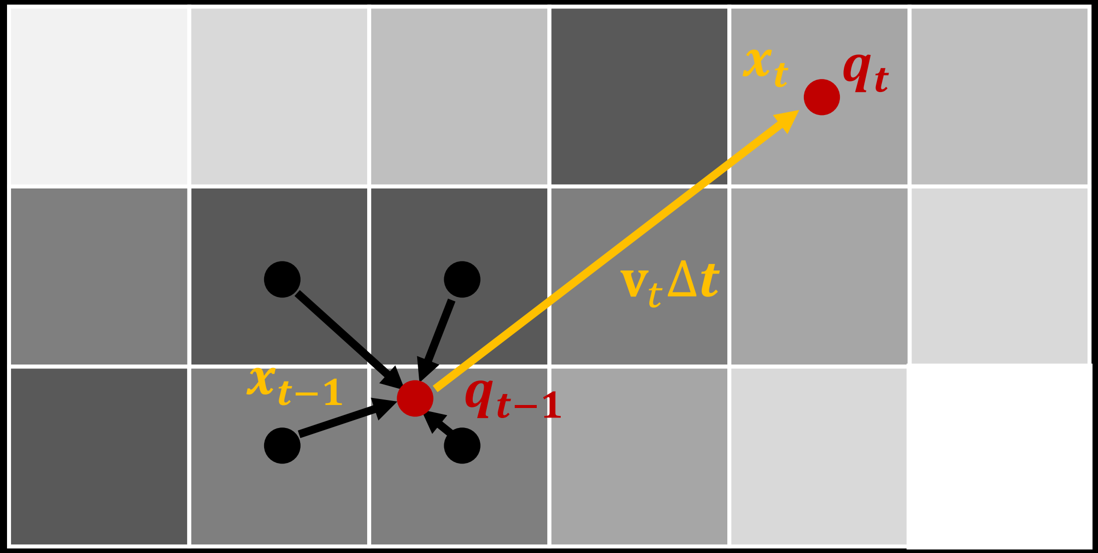


Push all velocities outward
by the same amount

- Global solution: multiple iterations through all cells
- Considering boundary conditions (see tutorial 17)

Advection

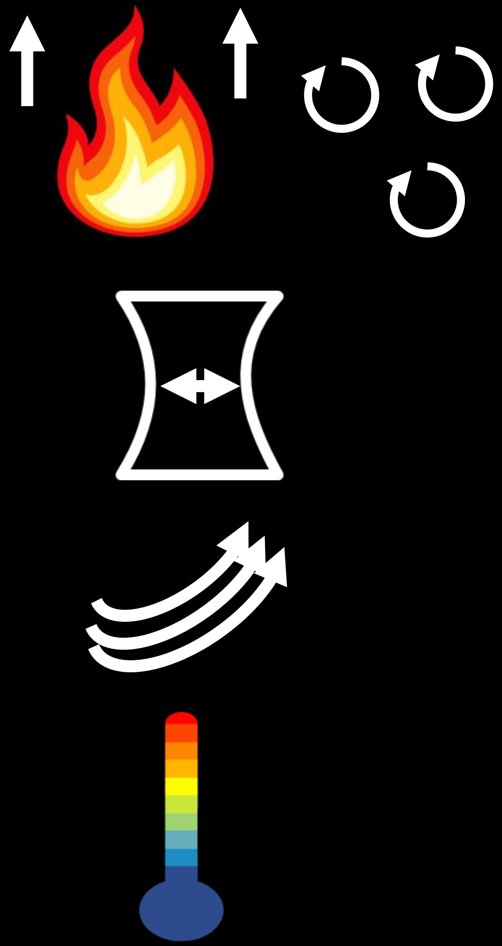
- Drive quantities (smoke density, velocities) along the velocity field
- Goal: compute the value of a quantity at q at position $\mathbf{x}_t$



- The previous position is $\mathbf{x}_{t-1} = \mathbf{x}_t - \mathbf{v}_t \cdot \Delta t$ (v = velocity, Δt = time step size)
- Not necessarily at the cell or face center!
- Compute the value as the weighted average of values around $\mathbf{x}_{t-1}$.

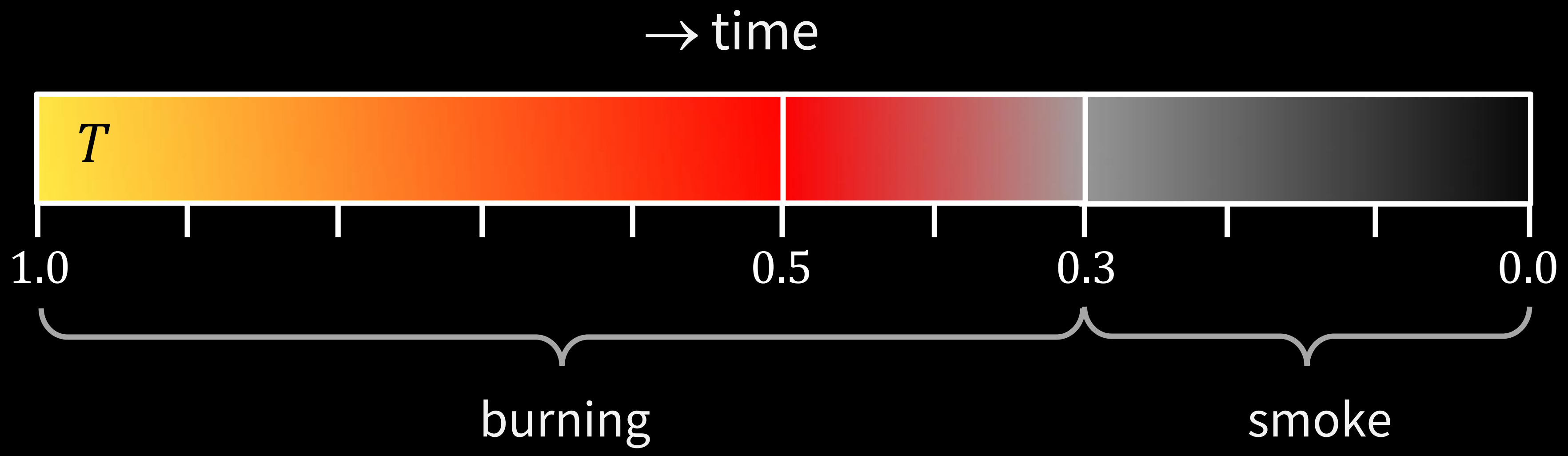
Fire Simulation Overview

- Modify velocities (add lift forces, turbulence)
- Make the fluid incompressible
- Advect velocity and temperature fields
- Modify temperatures (burning, cooling)

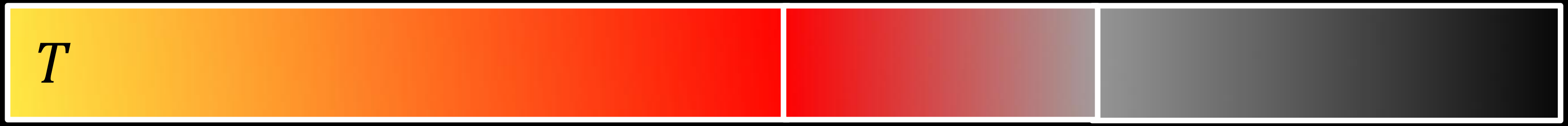


Simplified Physics

- Fuel, Temperature, Smoke → One normalized temperature field [0,1]



Fire Simulation



At every time step

- Initialize $T \leftarrow 1$ at fire sources
- Decrease T over time: $T \leftarrow \max(T - r \cdot \Delta t,\ 0)$. (Two different cooling rates for fire and smoke)
- Advect T along the fluid velocity
- Influence on velocity

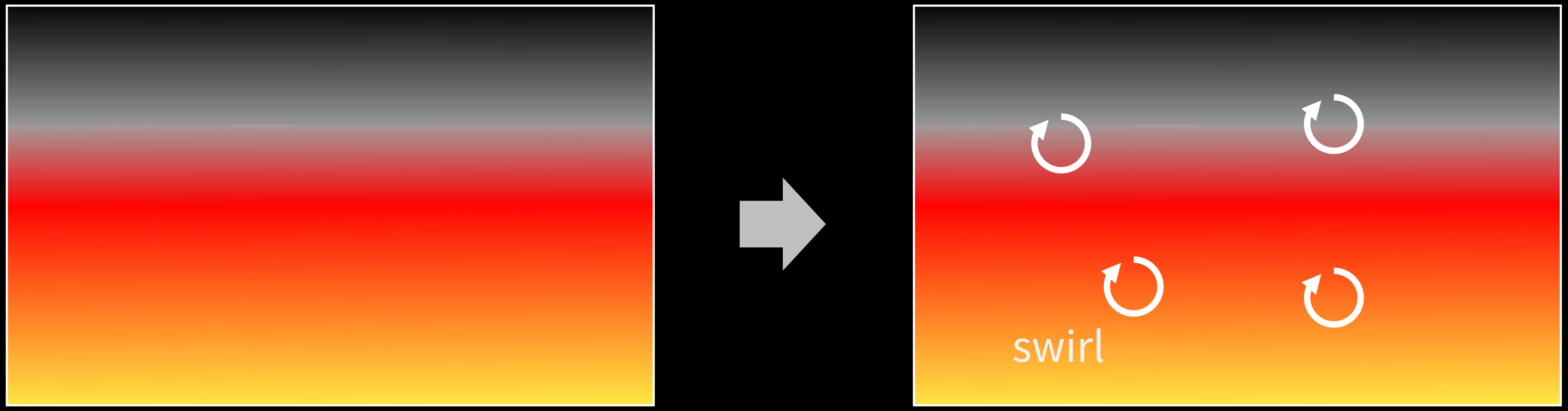
$$v_{\text{target}} \leftarrow v_{\text{lift}} \cdot T$$

$$v \leftarrow v + a \cdot \left(v_{\text{target}} - v\right) \cdot \Delta t$$

- Tune parameter v_{lift} and acceleration a

Adding Turbulence

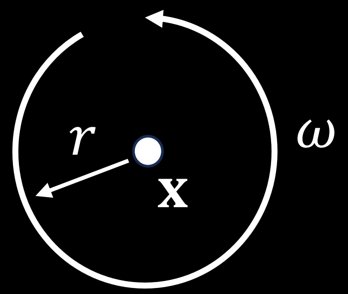
- A burning floor simulation:

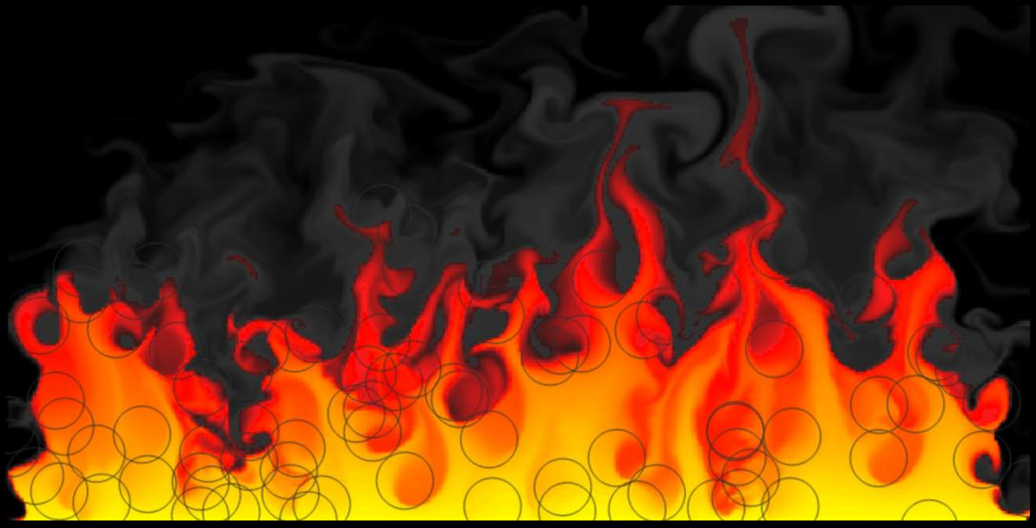


- We need external disturbance, turbulence!
- Adding swirls to simulate external disturbances and enhancing turbulence

Swirls

- A swirl has a position $\mathbf{x}$, a radius r, an angular velocity ω and an age





- Swirls are created with a given probability at fire source cells
- Advected with the velocity field
- Deleted when their maximum age is reached

Influence of Swirls on Velocities

- Velocity update

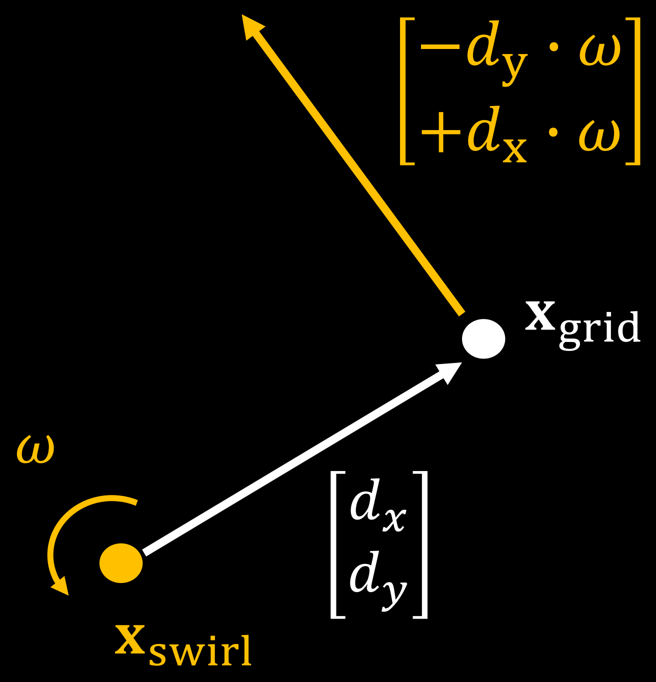
  Let $\mathbf{d} = \mathbf{x}_{\text{grid}} - \mathbf{x}_{\text{swirl}}$ and d the length of $\mathbf{d}$

  Then we have:

$$u_{\text{grid}} \leftarrow u_{\text{grid}} + \left(u_{\text{swirl}} - d_{\text{y}} \cdot \omega - u_{\text{grid}}\right) \cdot k(d)$$

$$v_{\text{grid}} \leftarrow v_{\text{grid}} + \left(v_{\text{swirl}} + d_{\text{x}} \cdot \omega - v_{\text{grid}}\right) \cdot k(d)$$

- The equations pull the grid velocities toward the swirl velocity at the grid node



Kernel

$$u_{\text{grid}} \leftarrow u_{\text{grid}} + \left(u_{\text{swirl}} - d_{\text{y}} \cdot \omega - u_{\text{grid}}\right) \cdot k(d)$$

$$v_{\text{grid}} \leftarrow v_{\text{grid}} + \left(v_{\text{swirl}} + d_{\text{x}} \cdot \omega - v_{\text{grid}}\right) \cdot k(d)$$

- The strength is defined by a kernel function $k(d)$ with $k(d)\ \epsilon\ [0,1]$ and $k(d) = 0$ if $d \geq r_{\text{swirl}}$

- I use the simple kernel: