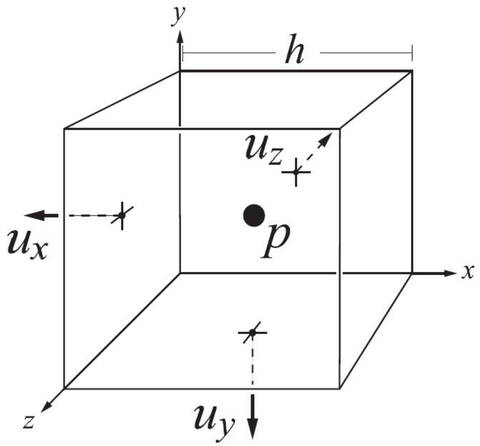
Fluid simulation

Jack van der Drift & Marjolein Zwerver

MAC grid

- Fluid Flow for the rest of us – Cline, Cardon and Egbert
- Divide the space into grid cells
- For each cell
  - Store velocity components on corresponding faces
  - Store pressure at cell center
- Use marker particles



Algorithm

1. Calculate time step Δt
2. Update cell states
3. Advance the velocity field
   a. Update fluid velocities
   b. Apply external forces
   c. Calculate pressure
   d. Apply pressure
   e. Update air velocities
   f. Set solid cell velocities
4. Move marker particles

1. Calculate timestep Δt

- Loop over all velocity components to get max velocity
- $\Delta t = 1/v_{max}$
- Clamped between 0.01 and 0.1

2. Update cell states

- Cell state matrix which keeps track of the state (water/air) for each cell
- If a cell contains a marker particle it is marked as a water cell

3a. Backwards particle trace

- For each face:
  - Get interpolated velocity in x, y and z direction
  - Trace particle backwards:
    - $pos_{prev} = pos_{old} - vel * \Delta t$
  - New velocity at face is the interpolated velocity at pos_{prev}

3b. Apply external forces

- Just gravity in our case
- $vel_y = vel_y - 9.81 * \Delta t$
- Apply only to faces that border fluid

3c. Calculate pressure

- Only for fluid cells
- A * P = B
  - Matrix A
    - Diagonal: -(number of non-solid neighbors)
    - Rest: 1 if neighbors, 0 otherwise
  - Adjusted divergence B: $\rho/\Delta t * \nabla u$
- Jacobi method to solve

3d. Apply pressure

- Only applied to velocity components that border fluid cells
- $\nabla p(x, y, z) = p(x, y, z) - p(x - 1, y, z)$
- $u_x = u_x - \Delta t/\rho * \nabla p(x, y, z)$

3e. Update air velocities

- Only for a 1 cell buffer zone
- Interpolated from bordering water cells

Set solid cell velocities

- Velocities that point into a wall are set to 0

Move marker particles

- Velocity for each particle calculated by interpolating between face values
- $pos_{new} = pos_{old} + vel * \Delta t$

Demo