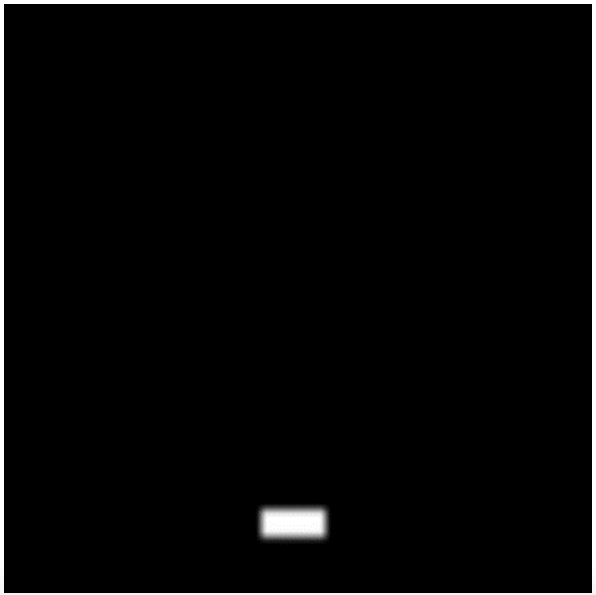
Physically-Based Simulation Exercise Session

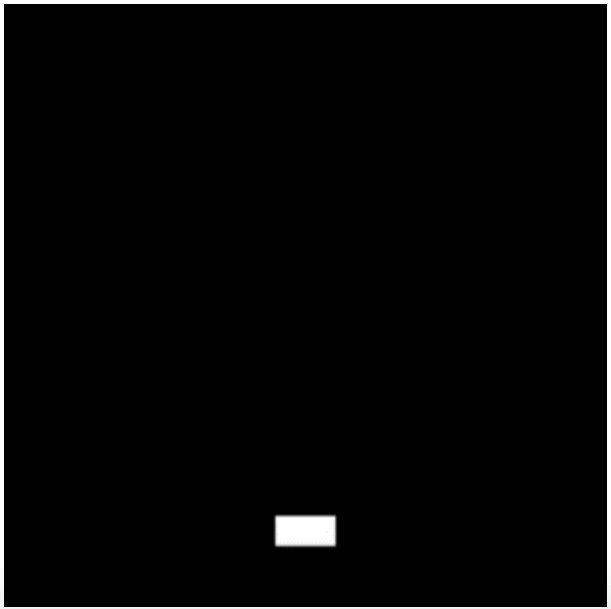
Exercise 3

Byungsoo Kim, kimby@inf.ethz.ch

Exercise Overview



grid: 64 x 64



grid: 200 x 200

Exercise Overview

- *Complete* an Eulerian 2D fluid solver with semi-Lagrangian advection

1) Solve for pressure

$$\nabla \cdot \nabla p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^n$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \frac{1}{\rho} \nabla p$$

2) Update velocities with pressure gradient

3) Semi-Lagrangian Advection!

Problem 1: Pressure Computation

- Very (very very) simple Poisson solver

$$p_{i,j}^{n+1} = \frac{\Delta x^2 d_{i,j} + \overbrace{p_{i-1,j}^{n+1} + p_{i,j-1}^{n+1}}^{\text{already updated}} + \overbrace{p_{i+1,j}^{n} + p_{i,j+1}^{n}}^{\text{not yet updated}}}{4}$$

- d is the right-hand side (divergence)
  - indifferent for the solver, already computed by framework
- Gauss-Seidel method
  - in-place substitution, no temporary buffers
- Residual: average L2-norm per grid cell
  - $r = \sqrt{\sum_{i,j} \left| d_{i,j} - A p_{i,j}^{n+1} \right|^2} / \#\ cells$

Problem 2: Velocity Update

- Compute pressure gradient with central differences for each velocity component

$$u_{i+½,j}^{n+1} = u_{i+½,j} - \Delta t \frac{1}{\rho} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

$$v_{i,j+½}^{n+1} = v_{i,j+½} - \Delta t \frac{1}{\rho} \frac{p_{i,j+1} - p_{i,j}}{\Delta x}$$

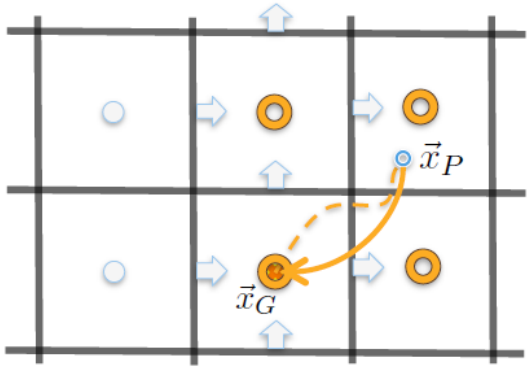
$$\nabla \cdot \nabla p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^n$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \frac{1}{\rho} \nabla p$$

2)

Problem 3: Semi-Lagrangian Advection

- Advect pressure and velocities
  - Different advections for different velocity components
- MAC grid!
- Bilinear interpolation
  - Look it up…



$$\vec{x}_P = \vec{x}_G - \Delta t\, \vec{u}_G$$

More Tips and Comments

- Indexing nightmare!
  - Note comments about indexing
- The total grid length is 1.0
  - Same unit for the velocity as well
- Temporary buffers for advection
- Grid values stored in an `Array2d`
  - Access through `()` operator:

```
Array2d a(4, 4); a(0, 1) = 2.0;
```

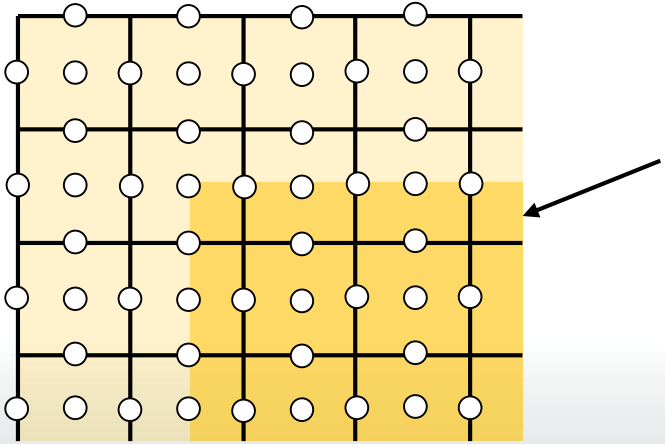
  - Reading size through `size()`:

```
int sizeX = a.size(0);
int sizeY = a.size(1);
```

More Tips and Comments

- Boundaries are handled by the framework
  - Do not update a value at the boundary
  - Do not read a value out of bounds (important for advection)

MAC grid



Position limits for backwards tracing

Project Plan

Due: November 8

- Presentation
  - **3 mins! (check the guideline plan on webpage)**
  - Submit by email to **kimby@inf.ethz.ch**
- Motivation
  - Simulation scenario
  - Inspirational image or video
- Technical background
  - Simulation methods you plan to use
  - Related work (references)
- Milestones/Schedule
  - Minimal, desired, and optimal targets (with dates)