Physically-Based Simulation
Final Presentation:
Dam Overflow

Group 21
Ryan Ammann, Paul Ochs
Fluid Implicit Particle Method

- Hybrid Method
- Uses grid based computation for solving pressure equations to keep the simulation incompressible
- Uses particle based computation for “advection” since this is way cheaper than for fully grid based

\[
u_p^{new} = \alpha \cdot lerp(u_{grid}^{new}, x_p) + (1 - \alpha)\left[u_p^{old} + lerp(\Delta u_{grid}, x_p)\right]
\]

PIC contribution

Flip contribution
One Simulation Step

1. Do particle to grid transfer
   - `compute_velocityfield();`
   - `save_velocities();`

2. Add forces and boundaries
   - `add_forces();`
   - `add_boundaries();`
   - `classify_cells();`

3. Solve pressure
   - `solve_poisson();`
   - `apply_pressure();`

4. Do grid to particle transfer
   - `update_particle_velocities();`

5. Get suitable timestep
   - `double stable_dt = cfl_timestep();`
   - `int substeps = std::ceil(m_dt / stable_dt);`
   - `for (int i = 0; i < substeps; i++)`
     - `advect_particles(stable_dt);`
Implementation
Stability and Optimization

- 2D implementation fairly stable, assuming “correct” damping and pressure
- 3D implementation more volatile
- Adaptive time-stepping

\[ u_p^{new} = \alpha \cdot lerp(u_{grid}^{new}, x_p) + (1 - \alpha)[u_p^{old} + lerp(\Delta u_{grid}, x_p)] \]

- Some small attempts at parallelization using OpenMP
- Simple compiler flags (-O3, march = native …) yielded best speedup
In Conclusion