Lecture 20:

Introduction to Fluid Simulation

Computer Graphics and Imaging
UC Berkeley CS184/284A
Example: Fluids

Macklin and Müller, Position Based Fluids TOG 2013
Example: Fluids

Macklin and Müller, Position Based Fluids TOG 2013
Problem Setup

Lagrangian Formulation

• Where in space did this material move to?
• Commonly used for solid materials

Eulerian Formulation

• What material is at this location in space?
• Commonly used for fluids
  • Why: Because fluids don’t remember their shape
Problem Discretization

Grids

- Store quantities on a grid
- Fluid move “through” grid
- Scales reasonably well to large systems
- Surface tracking is challenging

Particles

- Fluid defined by locations of particles
- Inter-particle forces create fluid behavior
- Scaling to large systems not simple
- Surface tracking less difficult

Many popular methods combine grids and particles
# Fluid Grid

Store Fluid State On Grid

- Velocity
- Pressure
- Density

### Staggered Grid

- Bilinear interpolation
- Seems odd at first
- Very useful
- Non-staggered produces unstable checkerboard
Fluid Grid

Store Fluid State On Grid

- Velocity
- Pressure
- Density

Staggered Grid

- Bilinear interpolation
- Seems odd at first
- Very useful
- Non-staggered produces unstable checkerboard
Fluid Grid

Store Fluid State On Grid
- Velocity
- Pressure
- Density

Staggered Grid
- Bilinear interpolation
- Seems odd at first
- Very useful
- Non-staggered produces unstable checkerboard
Fluid Grid

2D Staggered Grid

\[ u = u(x, y) \]
\[ p = p(x, y) \]

3D Staggered Grid

\[ u = u(x, y, z) \]
\[ p = p(x, y, z) \]
Fluid Grid

2D Staggered Grid

\[ u = \begin{bmatrix} u \\ v \end{bmatrix} \]

3D Staggered Grid
CHAPTER 2. OVERVIEW OF NUMERICAL SIMULATION

Figure 2.1: The two-dimensional MAC grid.

Figure 2.2: One cell from the three-dimensional MAC grid.

\[ \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \mathbf{u}_x = \mathbf{u} = \text{BiLinear}(\cdot, \cdot, \cdot, \cdot, \cdot) \]
Fluid Grid

\[ u = \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ u_x = u = \text{BiLinear}(\cdot, \cdot, \cdot, \cdot, \cdot) \]

\[ u_y = v = \text{BiLinear}(\cdot, \cdot, \cdot, \cdot, \cdot) \]
Vector Fields

\[ \mathbf{v} = \mathbf{v}(x, y) \]

\[ p = p(x, y) \]

**Gradient:**
Direction of greatest change

\[
\text{grad}(p(x, y)) = \nabla p \big|_{x,y} = \begin{bmatrix}
\frac{\partial p}{\partial x} \\
\frac{\partial p}{\partial y}
\end{bmatrix}
\]
Vector Fields

**Gradient:**

Direction of greatest change

\[
\text{grad}(p(x, y)) = \nabla p \bigg|_{x,y} = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix}
\]

\[
\text{grad}(p) = \nabla p
\]

The \( \nabla \) is a differential operator, like \( \frac{\partial}{\partial x} \), but a vector

\[
\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}
\]

\[
v = v(x, y) \\
p = p(x, y)
\]
Vector Fields

\[ \mathbf{v} = \mathbf{v}(x, y) \]
\[ p = p(x, y) \]

\[ \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \]

Gradient:
Direction of greatest change

\[ \text{grad}(p) = \nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \]

In 3D, cell centers and faces

Ren Ng, James O’Brien
Vector Fields

\[ \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \]

\[ \mathbf{v} = \mathbf{v}(x, y) \]

\[ p = p(x, y) \]

Gradient:
Direction of greatest change

\[ \text{grad}(p) = \nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \]

Divergence:
Net flow in or out of region

\[ \text{div}(\mathbf{u}) = \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \]

In 3D, cell centers and faces

CS184/284A

Ren Ng, James O’Brien
Vector Fields

Gradient:
Direction of greatest change

\[
\nabla p = \nabla \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix}
\]

Divergence:
Net flow in or out of region

\[
\text{div}(u) = \nabla \cdot u = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}
\]

Curl:
Circulation around point

\[
\text{curl}(u) = \nabla \times u = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}
\]

In 3D, cell faces and edges
Vector Fields

\[ \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \]

\[ \mathbf{v} = \mathbf{v}(x, y) \]

\[ p = p(x, y) \]

Gradient: Direction of greatest change

\[ \text{grad}(p) = \nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \]

Divergence: Net flow in or out of region

\[ \text{div}(\mathbf{u}) = \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \]

Curl: Circulation around point

\[ \text{curl}(\mathbf{u}) = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \]

In 3D, curl is vector stored at edges

CS184/284A

Ren Ng, James O’Brien
Vector Fields

\[ \nabla^2 = \nabla \cdot \nabla = \left[ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{array} \right] \cdot \left[ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{array} \right] = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \]

Divergence:
Net flow in or out of region
\[ \text{div}(\mathbf{u}) = \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \]

Curl:
Circulation around point
\[ \text{curl}(\mathbf{u}) = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \]

Gradient:
Direction of greatest change
\[ \text{grad}(p) = \nabla p = \left[ \begin{array}{c} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{array} \right] \]
Vector Fields

Laplacian:
Difference from the neighborhood average

\[ \nabla^2 = \nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \]

Divergence:
Net flow in or out of region

\[ \text{div}(\mathbf{u}) = \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \]

Curl:
Circulation around point

\[ \text{curl}(\mathbf{u}) = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \]

Directional Derivative:
How a quantity changes as point of observation moves

\[ (\mathbf{u} \cdot \nabla) = \begin{pmatrix} u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y} \end{pmatrix} \]

Gradient:
Direction of greatest change

\[ \text{grad}(p) = \nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \]

CS184/284A

Ren Ng, James O’Brien

“Vector Analysis” - lots of fun math
Navier–Stokes Equations (N-SE)

\[ \frac{\partial \mathbf{u}}{\partial t} = - (\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{\nabla p}{\rho} + \frac{\nu}{\rho} \nabla^2 \mathbf{u} + \frac{f_f}{\rho} \]

- Change in fluid velocity
- Advection
- Pressure
- Viscosity
- Field forces (e.g.: gravity)

\( \rho \) is density
\( \nu \) is viscosity
Bad Solver

- Store velocity ($u$) and density ($\rho$) on staggered grid
- Compute pressure ($p$) as function of density
- Use N-SE to update velocities
- Update densities $\dot{\rho} \propto -(u \cdot \nabla)\rho + \nabla \cdot (u\rho)$
- Repeat until end of simulation

Problem: Pressure waves move fast so this explicit method must use very small timesteps or go unstable.

Problem: Advection term also limits time step based on speed of fluid. (Bulk speed of fluid is generally less than wave speed.)
Incompressible Fluids

Replace pressure forces with constraints

- No more pressure waves
- This is another projection method!

Divergence is net in-/out-flow

- Constrain divergence to be zero by projection
  - $\nabla \cdot \mathbf{u} = 0$

Split advection term off from the rest of N-SE and use semi-Lagrangian advection.

“Stable Fluids” by Jos Stam, SIGGRAPH 99
Incompressible Fluids

Separate problems terms from the rest:

\[ \Delta u = \Delta t \left( -\frac{(u \cdot \nabla)u}{\rho} + \frac{\nabla p}{\rho} + \frac{\nu}{\rho} \nabla^2 u + \frac{f_f}{\rho} \right) \]

\[ \Delta u = \Delta t \left( \Delta u_a + \Delta u_p + \frac{\nu}{\rho} \nabla^2 u + \frac{f_f}{\rho} \right) \]

\[ u^* = u^t + \Delta t \left( \frac{\nu}{\rho} \nabla^2 u + \frac{f_f}{\rho} \right) \]

Unprojected and unadvected new velocities
Incompressible Fluids

Separate problems terms from the rest:

\[ u^* = u^t + \Delta t \left( \frac{\nu}{\rho} \nabla^2 u + \frac{f_f}{\rho} \right) \]

In general we will have \( \nabla \cdot u^* \neq 0 \)

Use pressure to correct this:

\[ \nabla \cdot \left( u^* + \Delta u_p \right) = \nabla \cdot u^* + \nabla \cdot \Delta u_p = 0 \]

\[ \Delta u_p = -\Delta t \frac{\nabla p}{\rho} \]

\[ \nabla \cdot u^* = \Delta t \nabla \cdot \frac{\nabla p}{\rho} \]
Incompressible Fluids

Separate problems terms from the rest:

\[ u^* = u^t + \Delta t \left( \frac{\nu}{\rho} \nabla^2 u + \frac{f_f}{\rho} \right) \]

\[ \nabla \cdot u^* = \Delta t \nabla \cdot \frac{\nabla p}{\rho} \]

\[ \frac{\Delta t \nabla^2}{\rho} p = \nabla \cdot u^* \]

Solve for pressure.

Density is now constant, so it can move past the divergence operator.

\[ A x = b \]
Incompressible Fluids

Add pressure correction to get projected, but not advected, velocities:

\[ u^+ = u^* - \frac{\Delta t \nabla^2 p}{\rho} \]

Solving for pressure

- Successive over-relaxation
  - Easy to understand and implement, but slow
- Pre-conditions conjugate gradient
  - Widely used, reasonably fast
- [Modified] Incomplete Cholesky for preconditioned
- Other problem-specific methods
Incompressible Fluids

Add pressure correction to get projected, but not advected, velocities:

\[ u^+ = u^* - \frac{\Delta t \nabla^2 p}{\rho} \]

No pressure projection

With pressure projection
Incompressible Fluids

Add pressure correction to get projected, but not advected, velocities:

\[ u^+ = u^* - \frac{\Delta t \nabla^2 p}{\rho} \]
Semi-Lagrangian Advection

(A method of characteristics)

Instead of using 2nd order advection term, pick up the values and move them!

• For each location
  • Track backward through grid for $\Delta t$
  • Interpolate value
  • Copy to new location

Note: This works for other quantities besides velocity.

Note: Vector values should be rotated based on flow, but most people don’t do this.

Note: Backtrace is done in one or more substeps.
Semi-Lagrangian Advection

Final velocity is:

\[ u^{t+\Delta t} = \text{advect} \left( u^* - \frac{\Delta t \nabla^2}{\rho} p \right) \]

Unconditionally stable

Large steps introduce extra damping

- Viscosity term often omitted as unwanted
Stable Fluids

Demo by Amanda Ghassaei

https://apps.amandaghassaei.com/gpu-io/examples/fluid/

Things to notice:

• In pressure view you can see grid cells
• You don’t see them when simulation is rendered!
• Note how much damping there is
• Note how pressure changes as cursor is moved
Viscoelasticity

Ideal Elastic Solids

- Deformations are recoverable
Viscoelasticity

Ideal Fluids

- Deformations are permanent
Viscoelasticity

Viscoelastic Fluids

1. Elastic deformation until a limit (yield point)

2. Some permanent (plastic) deformation after the limit
Viscoelasticity

Creep

- Plastic deformation occurs over time
Viscoelasticity

Strategy

• Start with an Eulerian based fluid simulation

• Add elastic behavior:
  – Elastic forces
  – Plastic yielding
  – Creep
Viscoelasticity

Fluid Simulation

- **Navier-Stokes**
  (Incompressible)

\[
\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{\nabla p}{\rho} + \frac{\mu \nu}{\rho} \nabla^2 u + \frac{f}{\rho}
\]

\[
\nabla \cdot u = 0
\]
Viscoelasticity

Viscoelastic Fluid Simulation

- **Modified Navier-Stokes**
  (Incompressible)

\[
\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{\nabla p}{\rho} + \frac{\mu_v}{\rho} \nabla^2 \mathbf{u} + \frac{f}{\rho} + \frac{\mu_c \nabla \cdot \epsilon}{\rho}
\]

\[
\nabla \cdot \mathbf{u} = 0
\]

- **Need to update strain \( \epsilon \)**
Viscoelasticity

Decomposing Strain

- Strain measures total deformation
- Plastic strain: permanent deformation
- Elastic strain: recoverable (temporary) deformation

\[ \varepsilon_{\text{Tot}} = \varepsilon_{\text{Elc}} + \varepsilon_{\text{Plc}} \]
Viscoelasticity

Rate of Strain

- No deformation function available in Eulerian formulation
- Instead compute strain-rate and integrate

\[ \dot{\varepsilon}^{\text{Tot}} = \frac{1}{2} \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \]

\[ \varepsilon^{\text{Tot}} = \varepsilon_0^{\text{Tot}} + \int_0^t \dot{\varepsilon}^{\text{Tot}} \, dt \]
Viscoelasticity

Strain Update

Case 1: \[ \| \varepsilon^{Elc} \| \leq \gamma \]

\[ \dot{\varepsilon}^{Tot} = \dot{\varepsilon}^{Elc} + \dot{\varepsilon}^{Plc} \]

\[ \dot{\varepsilon}^{Plc} = 0 \]

\[ \Rightarrow \dot{\varepsilon}^{Elc} = \dot{\varepsilon}^{Tot} \]

\( \gamma \): Plastic Yield Point

Extra material for 284A

Ren Ng, James O’Brien
Viscoelasticity

Putting It Together

\[ \dot{\varepsilon}_\text{Elc} = \begin{cases} \dot{\varepsilon}_\text{Tot} \\ \dot{\varepsilon}_\text{Tot} - \alpha \frac{\varepsilon_\text{Elc}}{\|\varepsilon_\text{Elc}\|} (\|\varepsilon_\text{Elc}\| - \gamma) \end{cases} \quad : \quad \|\varepsilon_\text{Elc}\| \leq \gamma \\
\dot{\varepsilon}_\text{Tot} - \alpha \frac{\varepsilon_\text{Elc}}{\|\varepsilon_\text{Elc}\|} (\|\varepsilon_\text{Elc}\| - \gamma) \quad : \quad \|\varepsilon_\text{Elc}\| > \gamma
\]

where:

\[ \dot{\varepsilon}_\text{Tot} = \frac{1}{2} \left( \nabla u + (\nabla u)^T \right) \]
Viscoelasticity

Effects of $\alpha$ and $\gamma$

$\gamma = 0.0 \quad \alpha = 25$

$\gamma = 0.1 \quad \alpha = 25$

$\gamma = 0.5 \quad \alpha = 25$

$\gamma = 0.0 \quad \alpha = 500$

$\gamma = 0.1 \quad \alpha = 500$

$\gamma = 0.5 \quad \alpha = 500$
Viscoelasticity

Effects of $\alpha$ and $\gamma$

$\gamma = 0.0 \quad \alpha = 25$

$\gamma = 0.1 \quad \alpha = 25$

$\gamma = 0.5 \quad \alpha = 25$

$\gamma = 0.0 \quad \alpha = 500$

$\gamma = 0.1 \quad \alpha = 500$

$\gamma = 0.5 \quad \alpha = 500$
Viscoelasticity

Drip Examples

Goktekin, Bargteil, O'Brien
Other Effects: Melting

Clausen, Wicke, Shewchuk, O’Brien

CS184/284A

Ren Ng, James O’Brien
Other Effects: Melting

Clausen, Wicke, Shewchuk, O'Brien
Other Effects: Strain Heating

Clausen, Wicke, Shewchuk, O'Brien
Other Effects: Strain Heating
Other Effects: Strain Heating

Clausen, Wicke, Shewchuk, O'Brien
Other Methods

Particle-based fluids

- SPH
  - Many variations
- FLIP/PIC

Mesh-based fluids

- Eulerian Meshes
- Lagrangian Meshes
Other Methods

Lagrangian Mesh Fluids

Clausen, Wicke, Shewchuk, O’Brien
Other Methods

Lagrangian Mesh Fluids

Clausen, Wicke, Shewchuk, O’Brien
Still to come: Surface Tracking and Collisions
Acknowledgments

This slide set contain contributions from:

- Kayvon Fatahalian
- David Forsyth
- Pat Hanrahan
- Angjoo Kanazawa
- Steve Marschner
- Ren Ng
- James F. O’Brien
- Mark Pauly