Modeling, Simulating and Rendering Fluids

Thanks to Ron Fediw et al, Jos Stam, Henrik Jensen, Ryan
Applications

• Mostly Hollywood
  – Shrek
  – Antz
  – Terminator 3
  – Many others…

• Games

• Engineering
Animating Fluids is Hard...

• Too complex to animate by hand
  – Surface is changing very quickly
  – Lots of small details

• Need automatic simulations
Ad-Hoc Methods

- Some simple algorithms exist for special cases
  - Mostly waves

- What about water glass?

- Too much work to come up with empirical algorithms for each case
Physically-Based Approach

• Borrow techniques from *Fluid Dynamics*
  – Long history. Goes back to Newton…
  – Equations that describe fluid motion

• Use numerical methods to approximate fluid equations, simulating fluid motion
  – Like mass-spring systems
What do we mean by ‘Fluid’?

- liquids or gases

Mathematically:
- A vector field $\mathbf{u}$ (represents the fluid velocity)
- A scalar field $p$ (represents the fluid pressure)
- fluid density ($d$) and fluid viscosity ($v$)
Vector Fields

• 2D Scalar function:
  – \( f(x,y) = z \)
  – \( z \) is a scalar value

• 2D Vector function:
  – \( u(x,y) = v \)
  – \( v \) is a vector value
    • \( v = (x', y') \)

• The set of values \( u(x,y) = v \) is called a vector field
Fluid Velocity == Vector Field

• Can model a fluid as a vector field $\mathbf{u}(x,y)$
  – $\mathbf{u}$ is the *velocity* of the fluid at $(x,y)$
  – Velocity is different at each point in fluid!

• Need to compute *change in vector field*
Particles carry Velocities

- Particle Simulation:
  - Track particle positions $x = (x,y)$
  - Numerically Integrate: change in position

- Fluid Simulation:
  - Track fluid velocities $u = (u,v)$ at all points $x$ in some fluid volume $D$
  - Numerically Integrate: change in velocity
Some Math

- **Del Operator:**

\[
\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)
\]

- **Laplacian Operator:**

\[
\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}
\]

- **Gradient:**

\[
\nabla p = \left( \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right)
\]
More Math

- **Vector Gradient:**
  \[ \nabla \mathbf{u} = (\nabla u, \nabla v, \nabla w) \]

- **Divergence:**
  \[ \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \]

- **Directional Derivative:**
  \[ \mathbf{u} \cdot \nabla = u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z} \]
Navier-Stokes Fluid Dynamics

- Velocity field \( \mathbf{u} \), Pressure field \( p \)
  - Viscosity \( \nu \), density \( d \) (constants)
  - External force \( \mathbf{f} \)

Navier-Stokes Equation:

\[
\frac{\partial \mathbf{u}}{\partial t} = - (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{d} \nabla p + \mathbf{f}
\]

Mass Conservation Condition:

\[
\nabla \cdot \mathbf{u} = 0
\]
Navier-Stokes Equation

Derived from momentum conservation condition

4 Components:
- Advection/Convection
- Diffusion (damping)
- Pressure
- External force (gravity, etc)

\[
\frac{\partial u}{\partial t} = - (u \cdot \nabla) u + v \nabla^2 u - \frac{1}{d} \nabla p + f
\]
Mass Conservation Condition

- Velocity field $u$ has zero divergence
  - Net mass change of any sub-region is 0
  - Flow in $==$ flow out
  - Incompressible fluid

- Comes from continuum assumption

$$ \nabla \cdot u = 0 $$
Change in Velocity

\[
\frac{\partial u}{\partial t} = - (u \cdot \nabla) u + \nabla \cdot (v \nabla u) - \frac{1}{d} \nabla p + f
\]

• Derivative of velocity with respect to time

• Change in velocity, or acceleration
  – So this equation models acceleration of fluids
Advection Term

Change in Velocity

\[-(u \cdot \nabla) u + \nabla \cdot (v \nabla u) - \frac{1}{d} \nabla p + f\]

- Advection term
  - Force exerted on a particle of fluid by the other particles of fluid surrounding it
  - How the fluid “pushes itself around”
Diffusion Term

\[
\text{Change in Velocity} = \text{Advection} + \nabla \cdot (\nu \nabla \mathbf{u}) - \frac{1}{d} \nabla p + \mathbf{f}
\]

- Viscosity constant \( \nu \) controls velocity diffusion
- Essentially, this term describes how fluid motion is damped
- Highly viscous fluids stick together
  - Like maple syrup
- Low-viscosity fluids flow freely
  - Gases have low viscosity
Weather: Advection & Diffusion

• “Jet-Stream”
Pressure Term

\[
\text{Change in Velocity} = \text{Advection} + \text{Diffusion} - \frac{1}{d} \nabla p + f
\]

- **Pressure follows a diffusion process**
  - Fluid moves from high-pressure areas to low-pressure areas

- **Moving == velocity**
  - So fluid moves in direction of largest change in pressure
  - This direction is the gradient
Weather: Pressure

• “Fronts” are the boundaries between regions of air with different pressure…

• “High Pressure Zones” will diffuse into “Low Pressure Zones”
Body Force

\[
\text{Change in Velocity} = \text{Advection} + \text{Diffusion} - \text{Pressure} + f
\]

- Body force term represents external forces that act on the fluid
  - Gravity
  - Wind
  - Etc…
Summary

\[
\frac{\partial \mathbf{u}}{\partial t} = - (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla \cdot (\nu \nabla \mathbf{u}) - \frac{1}{d} \nabla p + \mathbf{f}
\]

\[\text{Change in Velocity} = \text{Advection} + \text{Diffusion} - \text{Pressure} + \mathbf{f}\]

• Add mass conservation (1 liter in == 1 liter out) constraint:
  \[\nabla \cdot \mathbf{u} = 0\]

• Need to simulate these equations…
Incompressible Euler Equations

\[
\frac{\partial u}{\partial t} = - (u \cdot \nabla) u + f \\
\nabla \cdot u = 0
\]

self-advection forces

incompressible

(Navier-Stokes without viscosity)
Additional Equations

\[
\frac{\partial \rho}{\partial t} = - (\mathbf{u} \cdot \nabla) \rho + S
\]

temperature

\[
\frac{\partial T}{\partial t} = - (\mathbf{u} \cdot \nabla) T + H
\]

\[
f = -\alpha \, \rho \, \mathbf{z} + \beta \, (T - T_{\text{amb}}) \, \mathbf{z}
\]
Discretization
Algorithm

t = 0

add forces → self-advect → project

\[ t = t + dt \]
Step 1 – Add Force

- Assume change in force is small during timestep
- Just do a basic forward-Euler step

\[ \mathbf{w}_1(x) = \mathbf{w}_0(x) + \Delta t \mathbf{f}(x, t) \]

- *Note: \( \mathbf{f} \) is actually an acceleration?*
Step 2 - Advection
Method of Characteristics

- \( p \) is called the *characteristic*
  - Partial streamline of velocity field \( u \)
  - Can show \( u \) does not vary along streamline

- Determine \( p \) by tracing backwards

\[
w_2(x) = w_1(p(x, -\Delta t))
\]

- Unconditionally stable
  - Maximum value of \( w_2 \) is never greater than maximum value of \( w_1 \)
Self-Advection

Semi-Lagrangian solver (Courant, Issacson & Rees 1952)
Self-Advection

For each u-component...
Self-Advection

Trace backward through the field
Self-Advection

Interpolate from neighbors
Self-Advection

Set interpolated value in new grid
Self-Advection

Repeat for all u-nodes
Self-Advection

Similar for v-nodes
Self-Advection

V_{max} > V_{max}

Adveected velocity field
Enforcing Zero Divergence

Pressure and Velocity fields related

- Say we have velocity field \( w \) with non-zero divergence

- Can decompose into \( w = u + \nabla p \)
  - Helmholtz-Hodge Decomposition
  - \( u \) has zero divergence

- Define operator \( P \) that takes \( w \) to \( u \):

\[
    u = Pw = w - \nabla p
\]

- Apply \( P \) to Navier-Stokes Equation:

\[
    \frac{\partial u}{\partial t} = P \left( - (u \cdot \nabla) u + v\nabla^2 u + f \right)
\]

(Used facts that \( P_{uu} = u \) and \( P\nabla p = 0 \))
Operator P

- Need to find $\nabla p$
- Implicit definition:
  \[
  \nabla \cdot w = \nabla \cdot u + \nabla \cdot \nabla p
  \]
  \[
  \nabla \cdot w = \nabla^2 p
  \]

- Poisson equation for scalar field $p$
  - Neumann boundary condition $\frac{\partial p}{\partial n} = 0$

- Sparse linear system when discretized
Adding Viscosity – Diffusion

- **Standard diffusion equation**

\[
\frac{\partial w_2}{\partial t} = \nu \nabla^2 w_2
\]

- **Use implicit method:**

\[
w_3 - \Delta t \frac{\partial w_3}{\partial t} = w_2
\]

\[
(I - \nu \Delta t \nabla^2) w_3 = w_2
\]

- **Sparse linear system**
Step 4 - Projection

- Enforces mass-conservation condition \( \nabla \cdot \mathbf{u} = 0 \)

- Poisson Problem:
  \[
  \nabla^2 q = \nabla \cdot \mathbf{w}_3 \quad \mathbf{w}_4 = \mathbf{w}_3 - \nabla q
  \]

- Discretize \( q \) using central differences
  - Sparse linear system
  - Maybe banded diagonal…

- Relaxation methods too inaccurate
  - Method of characteristics more precise for divergence-free field
Solving the System

- Need to calculate: \( \frac{\partial \mathbf{u}}{\partial t} = P \left(- (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}\right) \)

- Start with initial state \( \mathbf{w}_0 = \mathbf{u}(x, t) \)

- Calculate new velocity fields

  add force \( \rightarrow \)  advect \( \rightarrow \)  diffuse \( \rightarrow \)  project

  \( \mathbf{w}_0 \rightarrow \mathbf{w}_1 \rightarrow \mathbf{w}_2 \rightarrow \mathbf{w}_3 \rightarrow \mathbf{w}_4 \)

- New state: \( \mathbf{u}(x, t + \Delta t) = \mathbf{w}_4 \)
Vorticity Confinement

Basic idea:

Add energy lost as an external force. Avoid very quick dissipation.

“Vorticity Confinement” force preserves swirling nature of fluids.
Vorticity Confinement

\[ \omega = \nabla \times \mathbf{u} \]
Vorticity Confinement

\[ N = \frac{\eta}{|\eta|} \quad \eta = \nabla|\omega| \]
Vorticity Confinement

\[ \mathbf{f} = \epsilon \ h \ (\mathbf{N} \times \omega) \]
Videos