Natural Phenomena

Water, Gas, Fire

CMPT 466
Computer Animation
Torsten Möller

Reading

- Chapter 5 of Parents book
- Lots of different SIGGRAPH papers
- Osher, Fedkiw, “Level Set Methods and Dynamic Implicit Surfaces”
- Lichtenbelt et al, “Introduction To Volume Rendering”

References


Goals

• Handle viscous liquids in computer animation
• interact with graphics primitives
• enough detail, realistic looking behavior, efficient solution

Kind of Flows!

• Fluid flows
  – Breaking waves, rapidly running stream, atmospheric clouds
  – Or …

Water, Gas & Clouds, Fire

• Heuristic methods – particle based
  – fast
  – mostly for games
• physics based modeling
  – based on Navier-Stokes equations
  – computationally expensive
  – extremely realistic

Fluid Phenomena

• Grid-based methods
  – partition into individual cells
• Particle based methods
  – representative particles “move” tracked through space
• Hybrid methods
  – ocean waves - grids
  – spray - particles
Basics

- Particles and Fields
  - Velocity field \( V(x,t) : 3D \) vector, space\( (x) \), time\( (t) \)
- Eulerian – sit and observe
  - Field quantities
- Lagrangian – move with the flow 😊
  - Quantities pertaining to individual particles

Still Waters and Small Amplitude Waves

- Model using sinusoids
- radial symmetric vs. linear waves (from source)
  
  \[ h_s(s) = A \cos\left( \frac{2\pi}{L} \right) \]
- \( L \) = wavelength, \( A \) = amplitude
- Time-varying height of a single point
  
  \[ h_t(t) = A \cos\left( \frac{2\pi}{T} \right) \]
- \( T \) = period of wave
- Combined 2D function:
  
  \[ h_{p,s}(s,t) = A \cos\left( 2\pi \left( \frac{s}{L} + \frac{t}{T} \right) \right) \]

Still Waters and Small Amplitude Waves (2)

- Typically use bump mapping for quick rendering
- compute derivative of \( h_p \)

  \[ \frac{d}{ds} h_p(s,t) = -A \frac{2\pi}{L} \sin\left( 2\pi \left( \frac{s}{L} + \frac{t}{T} \right) \right) \]

Anatomy of Waves

- \( C \) - propagation speed
- \( L \) - wavelength
- \( T \) - (time) length of period
- \( s \) - distance from source point

\[ f(s,t) = A \cos\left( 2\pi \frac{s-Ct}{L} \right) \]
Anatomy of Waves (2)

• Steepness of a wave $S = H/L$
  – S small - sinusoidal
  – S large - cresting/cycloid
• idealized water
• average orbital speed:
  $Q = \pi H/T = \pi HC/L = \pi SC$

Anatomy of Waves (3)

• Airy model - common simplification of CFD (computational fluid dynamics):
  $C = \sqrt{\frac{g}{\kappa \tanh(\kappa d)}} = \sqrt{\frac{gL}{2\pi \tanh\left(\frac{2\pi d}{L}\right)}}$

  • d - water depth
  • g - gravity acceleration
  • $\kappa$ - wave frequency

Anatomy of Waves (4)

$$C = \sqrt{\frac{g}{\kappa \tanh(\kappa d)}} = \sqrt{\frac{gL}{2\pi \tanh\left(\frac{2\pi d}{L}\right)}}$$

• as $d$ gets big $\tanh(\kappa d) \to 1$, hence
  $C = \sqrt{\frac{gL}{2\pi}}$

• as $d$ gets small, $\tanh(\kappa d) \to \kappa d$, hence
  $C = \sqrt{gd}$

Anatomy of Waves (5)

• As wave approaches shallow water
  • C & L decrease
  • T remains the same,
  • A & H remain the same
  • hence Q remains the same
  • hence waves tend to break
**Transport of water**

- Approx to Navier-Stokes
- \( \frac{\partial u}{\partial t} + u \cdot \frac{\partial u}{\partial x} + g \cdot \frac{\partial h}{\partial x} = 0 \)
- Change in height + spatial change in amount of water = 0
  \( \frac{\partial d}{\partial t} + \frac{\partial}{\partial x} (v \cdot d) = 0 \)

**CFD- Navier-Stokes Equations**

- \( u \) - is the kinematic viscosity,
- \( u \) is the velocity of the fluid parcel,
- \( p \) is the pressure, and
- \( \rho \) is the fluid density
  \( \frac{\partial}{\partial t} \rho + \text{div}(\rho u) = 0 \)
  \( \frac{du}{dt} = -(u \cdot \nabla)u - \frac{1}{\rho} \nabla p + \nu \nabla^2 u + f \)

**CFD (2)**

- Incompressible flow -
  \( \nabla \cdot u = 0 \)
- Euler’s equation is simplified without the viscosity term
  \( \frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{\nabla P}{\rho} = 0 \)

**Conservation of Mass**

- Mass = \( \int_{\Omega_t} \rho(x,0)dx = \int_{\Omega_t} \rho(x,t)dx \)
  \( \rho \) is density
  Transport theorem
  \( d \int_{\Omega_t} \rho(x,t)dx = \int_{\Omega_t} \left\{ \frac{\partial}{\partial t} \rho + \text{div}(\rho u) \right\}(x,t)dx = 0 \)
  Integrand vanishes
  \( \rho \) is constant for incompressible fluids
  \( \text{div} u = 0 \)
The Basis

\[ \frac{d}{dt} \int_{\Omega} f(x,t) dx = \int_{\Omega} \left( \frac{\partial}{\partial t} f + \text{div}(fu) \right) (x,t) dx = 0 \]

Conservation of Momentum

momentum = \( \int_{\Omega} \rho(x,t) u(x,t) dx \)

change in momentum = \( \sum \text{acting forces} \)

body forces: \( \int_{\Omega} \rho(x,t) f(x,t) dx \)

surface forces: \( \int_{\partial \Omega} \sigma(x,t) nds \)

\( \sigma \): stress tensor \( n \): normal

\[ \frac{d}{dt} \int_{\Omega} \rho(x,t) u(x,t) dx = \int_{\Omega} \rho(x,t) f(x,t) dx + \int_{\partial \Omega} \sigma(x,t) nds \]

Solving the equations

Basic Approach

1. Create a tentative velocity field.
   a. Finite differences
   b. Semi-Lagrangian method (Stable Fluids [Stam 1999])

2. Ensure that the velocity field is divergence free:
   a. Adjust pressure and update velocities
   b. Projection method

Conservation of Momentum

\[ x = \Phi(c,t) \]
\[ u(x,t) = \frac{\partial}{\partial t} \Phi(c,t) \]
Tentative Velocity Field

  \[ u_{i+1/2,j,k} = u_{i,j,k} + \delta t \left( \frac{1}{\delta x} \left[ (u_{i,j+1/2,k}^2 - (u_{i,j-1/2,k})^2 \right) + \frac{1}{\delta y} \left[ (u_{i+1/2,j-1/2,k} - (u_{i+1/2,j+1/2,k}) \right] + \frac{1}{\delta z} \left[ (u_{i+1/2,j,k-1/2} - (u_{i+1/2,j,k+1/2}) \right] + g \cdot \right) \\
  + (1/\delta x)(\rho_{i+1/2,j,k} - \rho_{i,j,k}) + \left( \frac{\nu}{\delta x^2} \right)(u_{i+1/2,j,k}) \\
  - 2u_{i+1/2,j,k} + u_{i-1/2,j,k}) + \left( \frac{\nu}{\delta y^2} \right)(u_{i+1/2,j+1,k} \\
  - 2u_{i+1/2,j,k} + u_{i+1/2,j+1,k}) + \left( \frac{\nu}{\delta z^2} \right)(u_{i+1/2,j+1,k}) \\
  - 2u_{i+1/2,j,k} + u_{i+1/2,j+1,k}) \right] \]

Limits on time step
- **CFL conditions** – don’t move more than a single cell in one time step
  \[ |u_{\text{max}}| \Delta t < \Delta x, \quad |v_{\text{max}}| \Delta t < \Delta y \]
- Diffusion term
  \[ 2\Delta t \nu < \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1} \]

Tentative Velocity Field

Stable Fluids Method
1. Add forces: \[ \tilde{u}_1(x) = u_0(x) + \Delta t \cdot f(x) \]
2. Advection
3. Diffusion

Advection
Finite differences is unstable for large \( \Delta t \).
Solution: trace velocities back in time. Guarantees that the velocities will never blow up.

\[ \tilde{u}_2(x) = \tilde{u}_1(p(x,-\Delta t)) \]
Rendering of Results

- Typically marching cubes of a surface
- Particle rendering

Tracking the Free Surface

- The movement of the free surface is not explicit in the Navier-Stokes equations.
- Three methods for tracking the free surface:
  1. Marker and cell (MAC) method
  2. Front tracking
  3. Particle level set method

MAC

- Due to [Harlow and Welch 1965].
- Track massless marker particles to determine where the free surface is located.
- Markers are transported according to the velocity field.
- Cells with markers are *fluid cells*. Fluid cells bordering empty cells are *surface cells*.
- There are boundary conditions that must be satisfied at the surface.
- Extended by [Chen et al. 1997] to track particles only near the surface.

- Problems:
- Can lead to mass dissipation, especially with stable fluid style advection.
- No straight forward way to extract a smooth surface.
Front Tracking

- Proposed by [Foster and Fedkiw 2001]
- Front tracking uses a combination of a level set and particles to track the surface.
- The particles are used to define an implicit function. An isocontour of this function represents the liquid surface.
- The isocontour yields a smoother surface than particles alone.

Front Tracking

- Using the level set method, the isocontour can be evolved directly over time by using the fluid velocities.
- Particles and level set evolution have complementary strengths and weaknesses
  - Level set evolution suffers volume loss
  - Particles can cause visual artifacts
  - Level sets are always smooth.
  - Particles retain details.

Front Tracking

- Combine the two techniques by giving particles more weight in areas of high curvature. Particles escaping the level set are rendered directly as splashing droplets.

Particle Level Set Method

- Extrapolated velocities at the surface give more realistic motion.
Algorithm – Foster & Fedikw

I. Model the static environment as a voxel grid
II. Model the liquid volume using a combination of particles and an implicit surface.

For each simulation time step

I. Update the velocity field using finite differences combined with a semi-Lagrangian method
II. Apply velocity constraints due to moving objects
III. Enforce incompressibility by solving a linear system
IV. Update the position of the liquid volume (particles and implicit surface) using the new velocity field

Voxel Environment

• A rectangular grid of voxels
• Each cell
  – Pressure variable at its center
  – Shares a velocity variable with each of its adjacent neighbors
  – Velocity is defined at the center of the face
  – Tagged empty or filled with an impermeable static object

Liquid Representation

• The actual distribution of liquid
  – Combination of inertia-less particles and a dynamic iso-contour (implicit surface)
  – Particles provide detail where the surface starts to splash
  – Iso-contour provides a smooth surface

Using Particles

• Particles
  – Placed into the grid according to some initial liquid distribution
  – Positions evolve over time by simple convection
  – Particle velocity: from velocity grid using trilinear interpolation
  – Moved according to
  \[
  \frac{dx_p}{dt} = v_x
  \]
Particles

- **Drawback**
  - Not easy to extract a smooth polygonal description of the actual liquid surface
  - Possible
    - Connecting all the particles together into triangles
    - Suffer from connectivity, smooth surface, temporal aliasing

Isocontour

- **Isocontour**
  - an implicit function
    \[ \phi_p(x) = \sqrt{(x_i - x_{m})^2 + (x_j - x_{pj})^2 + (x_k - x_{pk})^2} - r \]
  - Surface: the spherical shell around \( x_p \) where \( \phi_p(x) = 0 \)
  - use a marching cubes algorithm to tessellate the iso-contour with polygons

Dynamic Level Set

- **Tracking the surface of a volume of liquid**
  - Track and move particles in the velocity field
    - Adding extra particles when the surface becomes too sparsely resolved
    - Removing them as the surface folds, or “splashed” back over itself
  - The level set method
    - Evolve \( \Phi \) directly over time using the liquid velocity field \( u \)

Dynamic Level Set

- **Update \( \Phi \)**
  - a convection term
  - could solve using semi-Lagrangian method
  - too inaccurate since this represents the mass evolution
Hybrid Surface Model

- **motive**
  - Particles evolution: fully Lagrangian approach
  - Level set evolution: fully Eulerian approach
  - Level set is always smooth, particles retain detail
- **A novel combination**
  - At each time step, evolve the particles and the level set $\Phi$
  - Use the updated value of the level set function to decide how to treat each particle

Control

- **Animation control**
  - Difficulty in arbitrary aesthetic level of control over a physics-based animation
  - Swirl, mix, splash (local control)
  - Foster and Metaxas (CGI ’97)
    - Directly manipulate the body force term, $g$

Results

- **Interaction between a thick liquid and a hand animated character**

More …

- Fedkiw’s results
- [http://graphics.stanford.edu/~fedkiw](http://graphics.stanford.edu/~fedkiw)