
SIGGRAPHASIA2011
HONG KONG
The 4th ACM SIGGRAPH Conference and Exhibition
on Computer Graphics and Interactive Techniques in Asia
Conference: 12-15 December | Exhibition: 13-15 December
Hong Kong Convention and Exhibition Centre
Sponsored by ACM SIGGRAPH
www.SIGGRAPH.org/ASIA2011

Realtime Water Simulation on GPU

Nuttapong Chentanez
NVIDIA Research

Overview

- Approaches to realtime water simulation
- Hybrid shallow water solver + particles
- Hybrid 3D tall cell water solver + particles
- Future

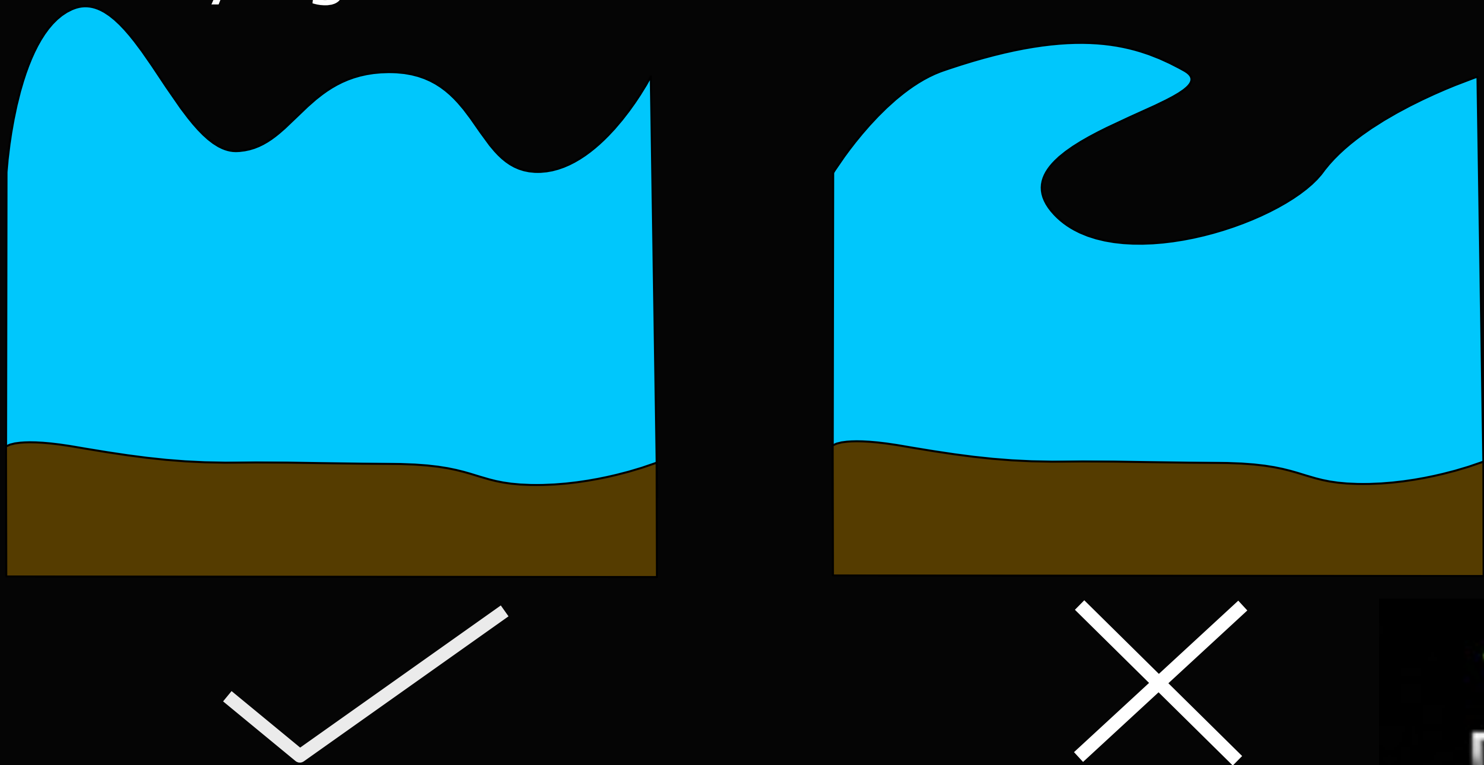
Realtime Water Simulation

“2D”

“3D”

"2D" Simulations

- Water represented by height above an underlying terrain



“2D” Simulations

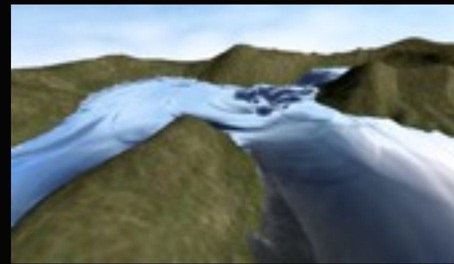
- Grid

NVIDIA DirectX 11, Island Demo

FFT

Hilko et al. 09

Wave sim
Pipe sim



Brodtkorb A. R. et al. 11,

Shallow water sim

- Particle

Yuksel et. al. 07

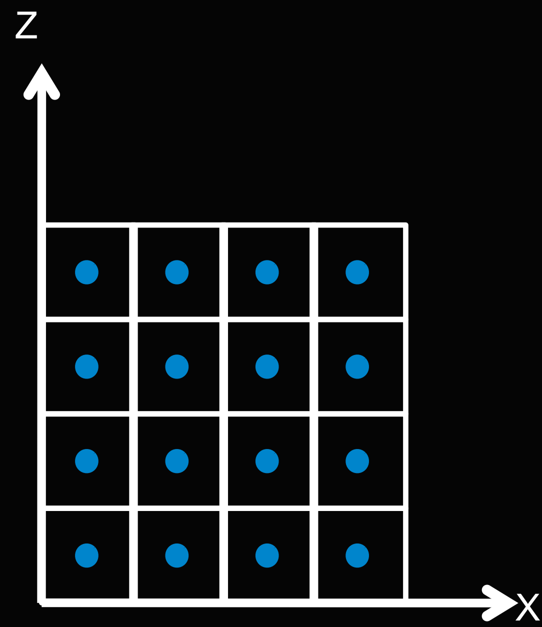
Wave Particle

Solenthaler et al. 11

Shallow water SPH

“2D” grid

- Water depth (and terrain height) stored in 2D array
- Water depth is updated in each time step



FFT

- Fast Fourier Transform (FFT)
- Represent waves as sum of sinusoids
- Wave length, speed, amplitude from statistical models
- Update height and derivatives in frequency domain
- Use iFFT to transform back to spatial domain for rendering

FFT

NVIDIA DirectX 11, Island Demo

FFT

- Pros
  - Fast
  - Great results for ocean wave, open water
- Cons
  - No interaction with objects
  - No boundary

Wave equation / Pipe model

- Wave equation
  - Assumptions: Water surface is a height field, velocity constant vertically, water is shallow, pressure gradient is vertical, ignore non-linear terms
  - Discretize temporally and spatially
  - Result in water height stored in 2D array + update rules

Wave equation / Pipe model

Hilko et al. 09, “Real-Time Open Water Scenes with Interacting Objects”

Wave equation / Pipe model

- Pipe model
  - Water heights stored in 2D array
  - Neighbors are connected by pipe
  - Flow rate in pipes updated by heights
  - Heights changed by flow rate

SIGGRAPHASIA2011 HONG KONG

Wave equation / Pipe model

Stava et al. 08, “Interactive Terrain Modeling Using Hydraulic Erosion”

NVIDIA.

SIGGRAPH ASIA 2011 HONG KONG

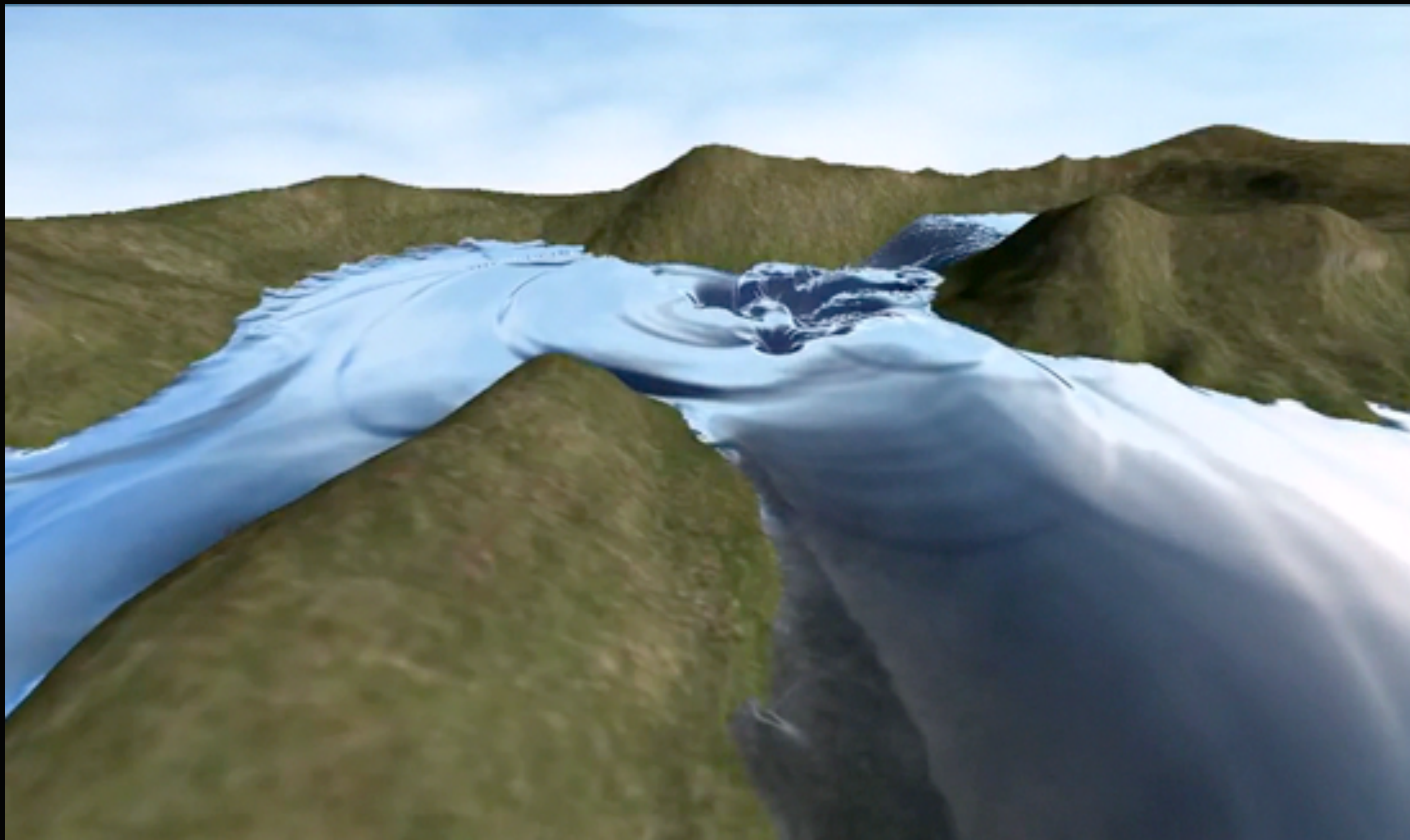
Wave equation / Pipe model

- Pros
  - Still fast
  - Interaction with objects
  - Boundary
- Cons
  - No vortices
  - No large flow
  - Not unconditionally stable

Shallow water equation

- Assumptions: Water surface is a height field, velocity is constant vertically, water is shallow, pressure gradient is vertical, with non-linear term
- Discretize temporally and spatially
- Result in water height + velocity stored in 2D array + update rules

Shallow water equation



Brodtkorb A. R. et al. 11, “Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation”

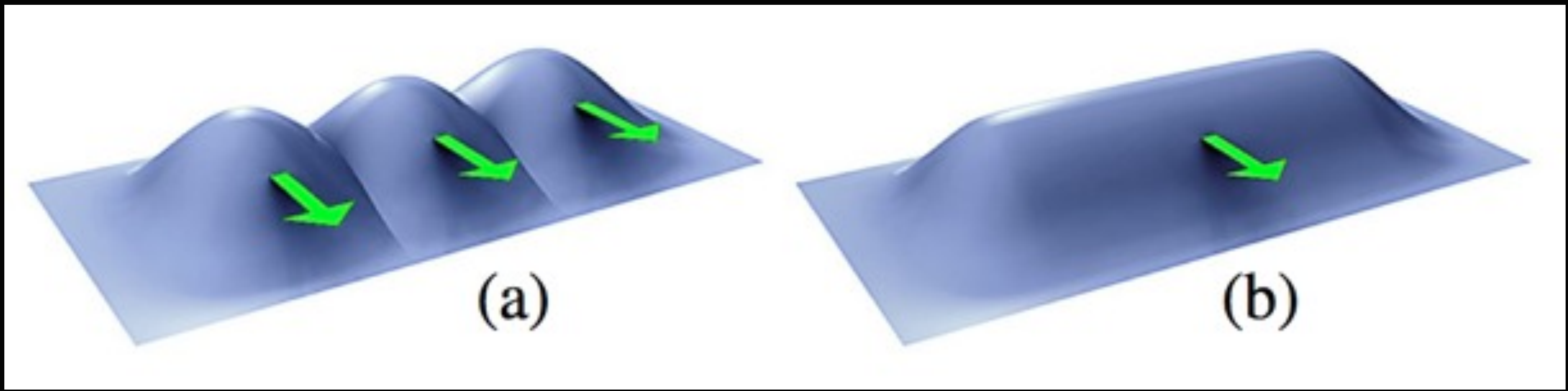
Shallow water equation

- Pros
  - Still fast
  - Interaction with objects
  - Boundary
  - Vortices

- Cons
  - Not unconditionally stable
  - No splash, foam, spray

Wave Particles

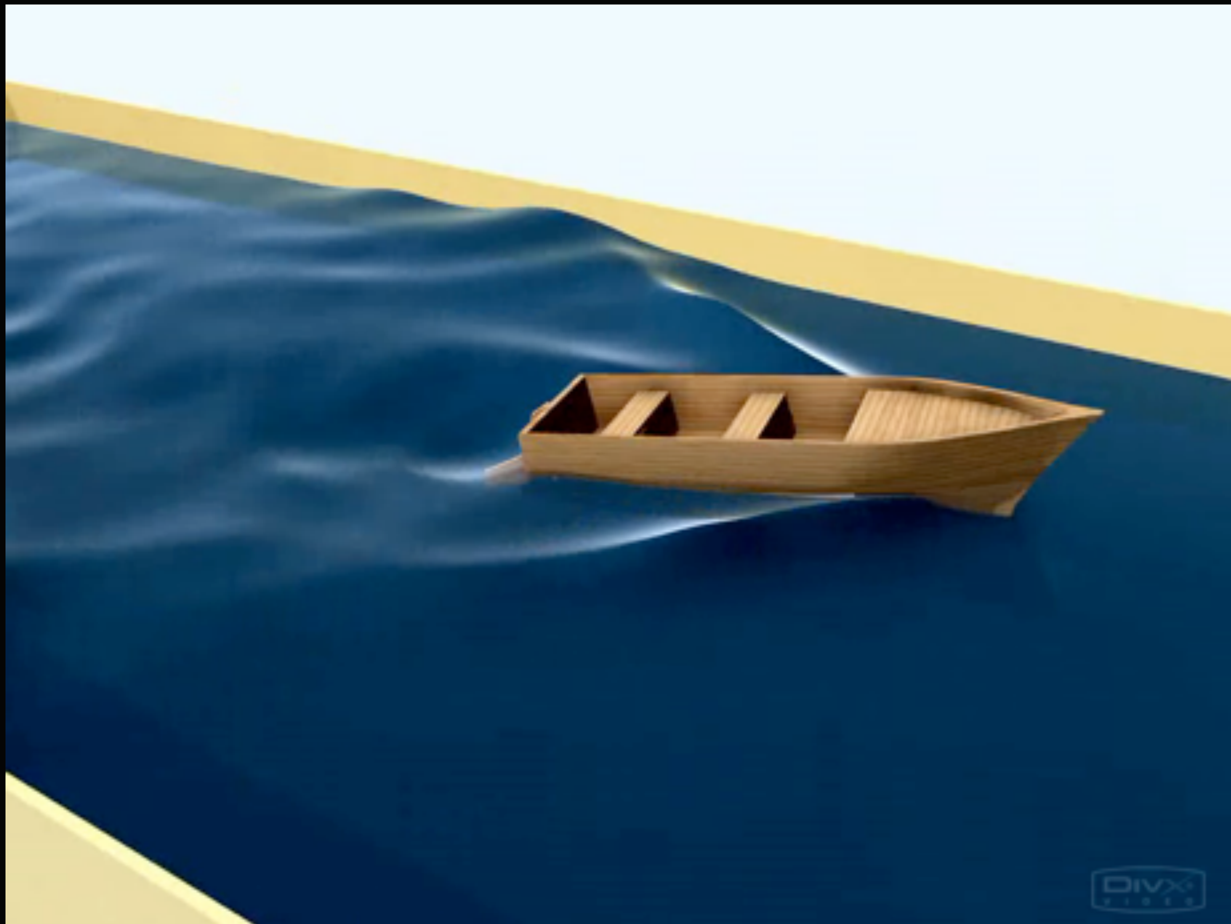
- Particle based wave simulation
- Each particle stores a waveform
- Particles form wave front



Yuksel et. al. 07, “Wave Particles”

- Either bounce off or leave domain boundary

Wave Particles



Yuksel et. al. 07, "Wave Particles"

Wave Particles

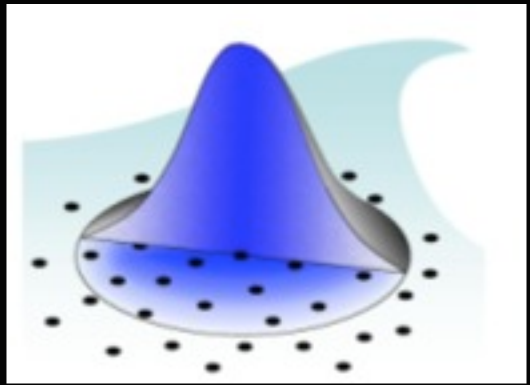
- Pros
  - Interaction with objects
  - Open boundary is easy
  - Unconditionally stable

- Cons
  - Still require grid for rendering
  - No vortices
  - No large flow

SIGGRAPHASIA2011 HONG KONG

SPH Shallow Water Simulation

- Use Smoothed Particles Hydrodynamics (SPH) to solve shallow water equation
- Particles store mass and velocity
- Kernels are centered around particles



SPHERIC - SPH European Research Interest Community

- Volume computed by summing kernel values
- Density = Mass / Volume interpreted as height

SPH Shallow Water Simulation

Solenthaler et al. 11, "SPH Based Shallow Water Simulation"

SPH Shallow Water Simulation

- Pros
  - Interaction with objects
  - Open boundary is easy
  - Vortices and Flow
- Cons
  - Still require grid for rendering
  - Not unconditionally stable
  - Still no 3D effect!

SIGGRAPHASIA2011 HONG KONG

"2D" Simulations

- Generally fast
- Interaction with solids
- Used in many games

- But no 3D effects
    - Can get away with good procedural approaches!

NVIDIA DirectX 11, Island Demo
FFT

Hilko et al.
Wave sim
Pipe sim

Brodtkorb A. R. et al. 11,
Shallow water sim

Yuksel et. al.
Wave Particle

Solenthaler et al. 11
Shallow water SPH

NVIDIA

“3D” Simulations

- Grid

Long B. and Reinhard E.

Discrete Sine/Cosine Transform

Keenan C. et al. 2007

Regular Grid

- Particles

NVIDIA GF100 Fluid Demo

NVIDIA PhysX Fluid Demo

SPH

NVIDIA

SIGGRAPHASIA2011 HONG KONG

3D grid

- Water states stored in 3D array
  - Velocity
  - Distance to surface
  - Density
  - etc.

- States are updated in each time step

NVIDIA

Discrete Sine/Cosine Transform

- Use cosine and sine transform
  - Instead of FFT
  - To be able to enforce boundary condition
- Do physics in frequency domain
- Transform back to spatial domain

Discrete Sine/Cosine Transform

Long B. and Reinhard E. 09 ,"Real-Time Fluid Simulation Using Sine/Cosine Transforms"

Discrete Sine/Cosine Transform

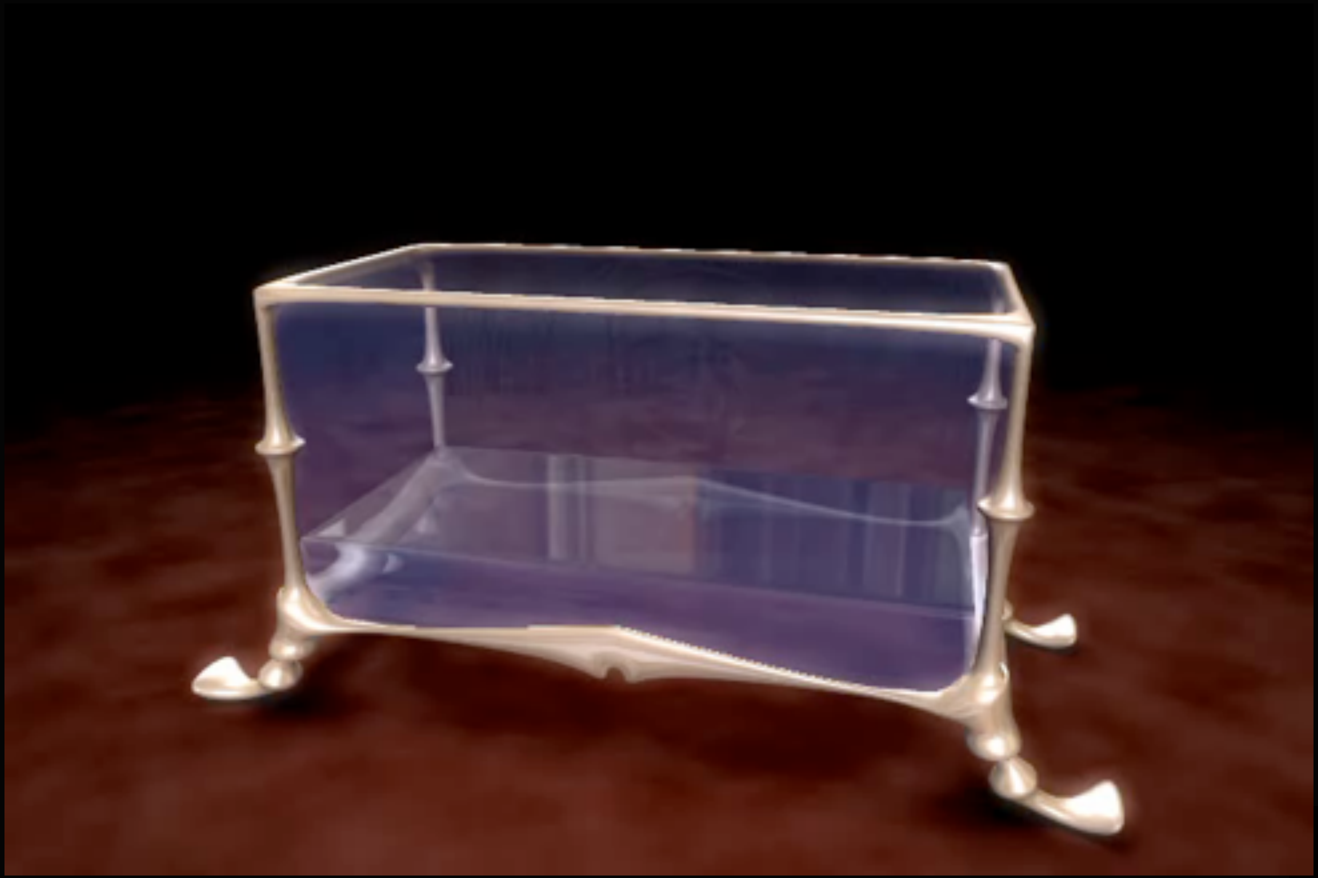
- Pros
  - Relatively fast
  - Unconditionally stable

- Cons
  - No interaction with object
  - Box shape domain
  - No small scale details for coarse grid

Regular grid

- Commonly used in offline production simulation
- Store water states in dense 3D grid
- Solve fluid dynamics PDE by discretizing spatially and temporally
- States in the next time step determined by state in the current time step and external forces
- "Brute Force"

Regular Grid



Keenan C. et al. 2007, “Real Time Simulation and Rendering of 3D Fluids”

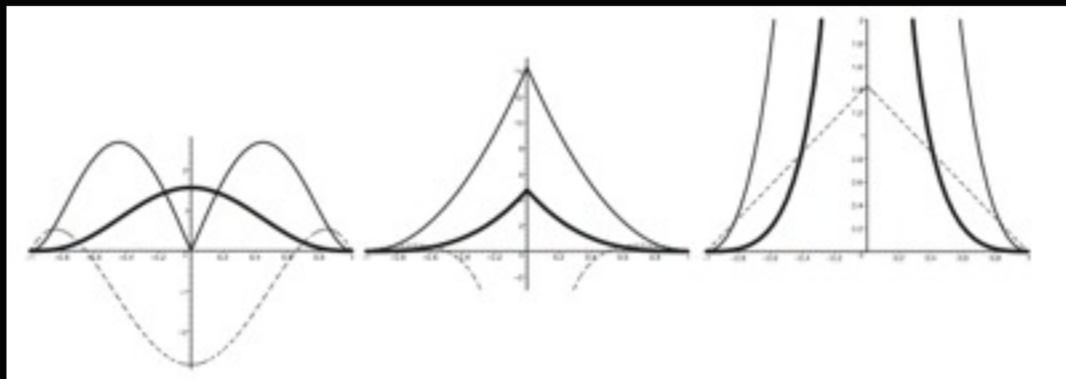
Regular Grid

- Pros
  - Good result
  - Unconditionally stable
- Cons
  - Box shape domain
  - Very computationally intensive
  - Mass loss
  - No small scale details for coarse grid

SIGGRAPH ASIA 2011 HONG KONG

SPH Simulation

- Use Smoothed Particles Hydrodynamics (SPH) to solve fluid dynamic PDE
- Particles store mass, velocity,
- Kernels are centered around particles

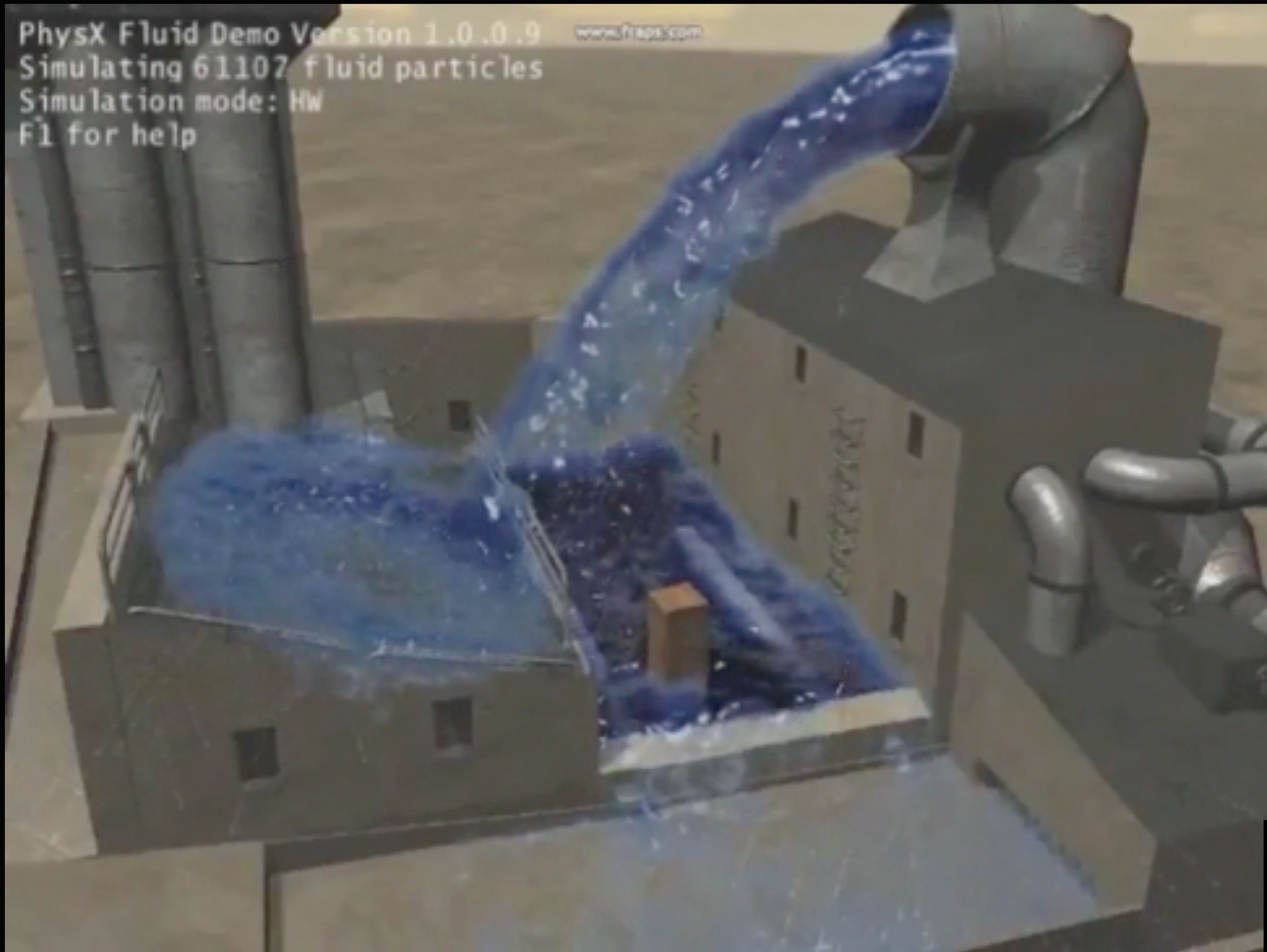


Mueller et al. 03, "Particle-Based Fluid Simulation for Interactive Applications"

- Reconstruct surface from particle or render particles directly

SIGGRAPHASIA2011
HONG KONG

SPH Simulation



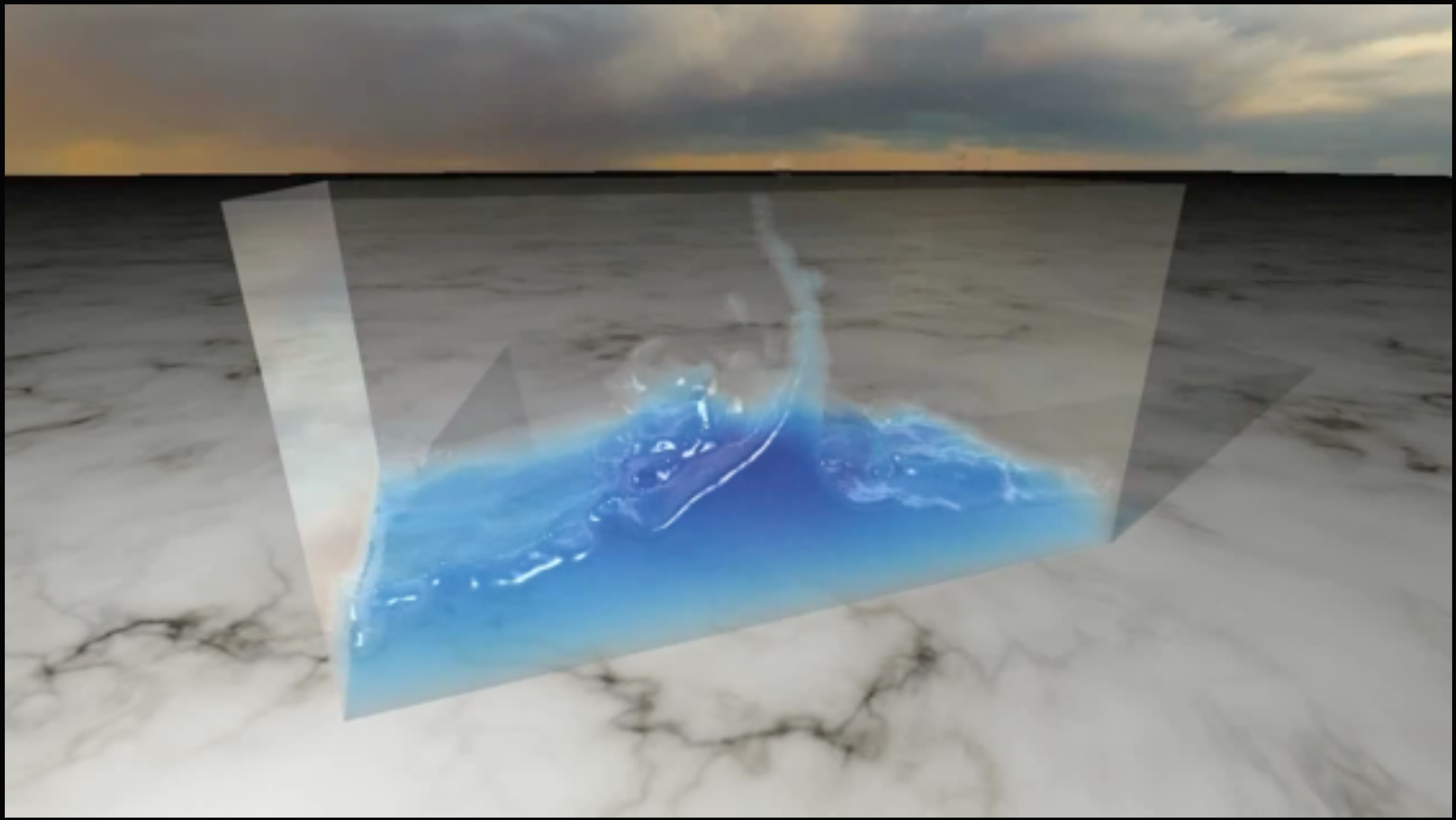
NVIDIA PhysX Fluid Demo

NVIDIA

Sponsored by ACM SIGGRAPH

www.SIGGRAPH.org/ASIA2011

SPH Simulation



NVIDIA GF100 Fluid Demo

SIGGRAPH ASIA 2011 HONG KONG

SPH Simulation

- Pros
  - Arbitrary domain
  - Interaction with object
  - No mass loss
- Cons
  - Noisy surface
  - Not unconditionally stable

NVIDIA

"3D" Simulations

- At high resolution, produce great results
  - Widely used in movie industry

- Can't afford to do large scene with small scale details

Long B.

Discrete Sine/Cosine Transform

Keenan C.

Regular Grid

NVIDIA GF100 Fluid

NVIDIA PhysX Fluid

SPH

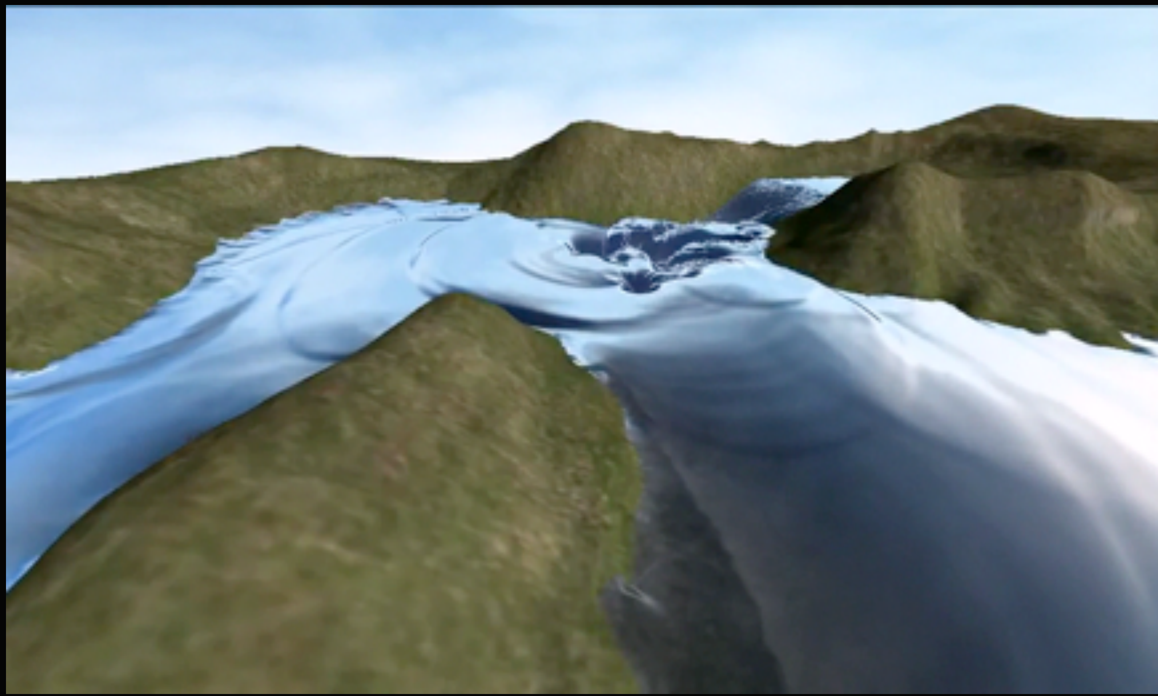
SIGGRAPH ASIA 2011 HONG KONG

Overview

- Approaches to realtime water simulation
- Hybrid shallow water solver + particles
- Hybrid 3D tall cell water solver + particles
- Future

Shallow water equation

- Missing
  - Splashes
  - Sprays
  - Small Waves
  - Foams
- Use particles!



Brodtkorb A. R. et al. 11, "Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation"

Shallow Water Solver + Particles

- Large bodies of water
  - Pond, River, Beach, Open Ocean
- Small scale details
  - Splashes, Sprays, Small Waves, Foams

Shallow Water Solver + Particles

Chentanez N. and Mueller M. 2010, “Real-time Simulation of Large Bodies of Water with Small Scale Details”

Shallow Water Equations

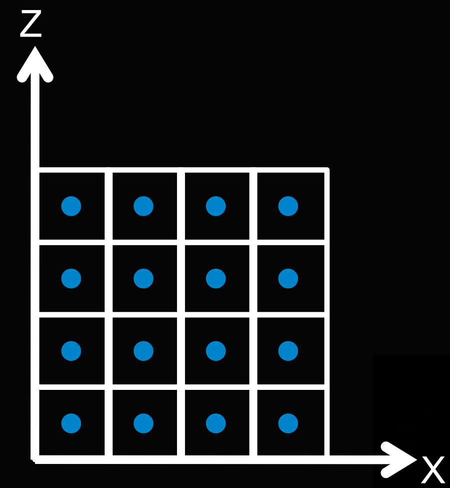
- Simplify from 3D Fluid Equation to 2D
  - Water depth, h
  - 2D velocity, v
  - Terrain height, H

$$\frac{Dh}{Dt} = -h\nabla \cdot \mathbf{v}$$

$$\frac{D\mathbf{v}}{Dt} = -g\nabla\eta + \mathbf{a}^{ext}$$

$$\eta = h + H$$

- Discretized with staggered grid
  - Cell center • :
    - Store h and H
  - X–Face | and Z–Face — :
    - x and z component of v



SIGGRAPHASIA2011 HONG KONG

Particles Simulation

- Particles sources
  - Waterfalls
    - Terrain height discontinuity, create spray and splash

  - Breaking waves
    - When wave about to overturn, create spray and splash

NVIDIA

SIGGRAPHASIA2011 HONG KONG

Particles Simulation

- Particles sources
  - Falling splash
    - Create spray and foam

  - Solid interaction
    - Create spray and splash

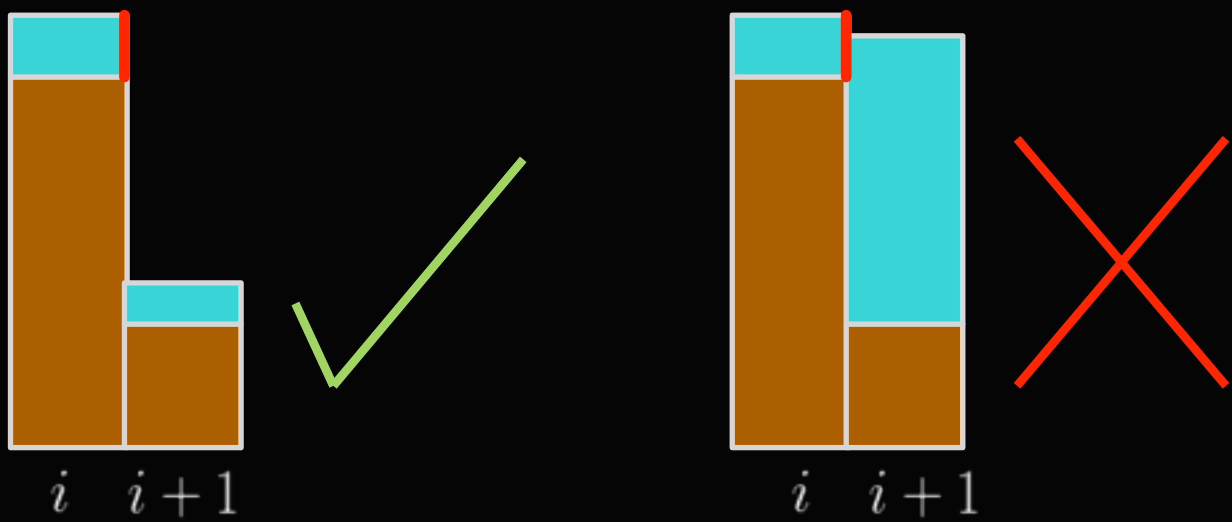
NVIDIA

Sponsored by ACM SIGGRAPH

www.SIGGRAPH.org/ASIA2011

Waterfall

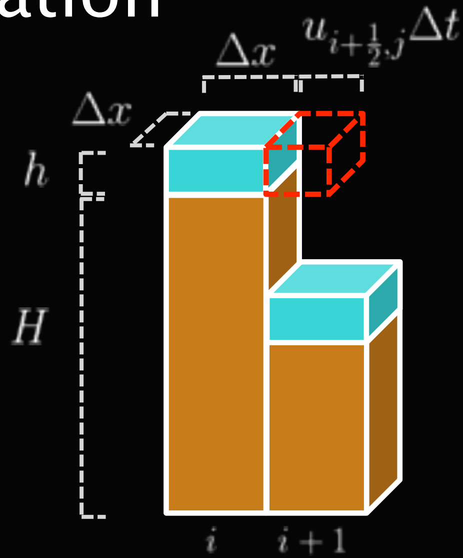
- We treat a face as a waterfall face if
  - Terrain height change is greater than a threshold and
  - Water height in the lower cell has not yet reached the terrain height in the higher cell



SIGGRAPHASIA2011 HONG KONG

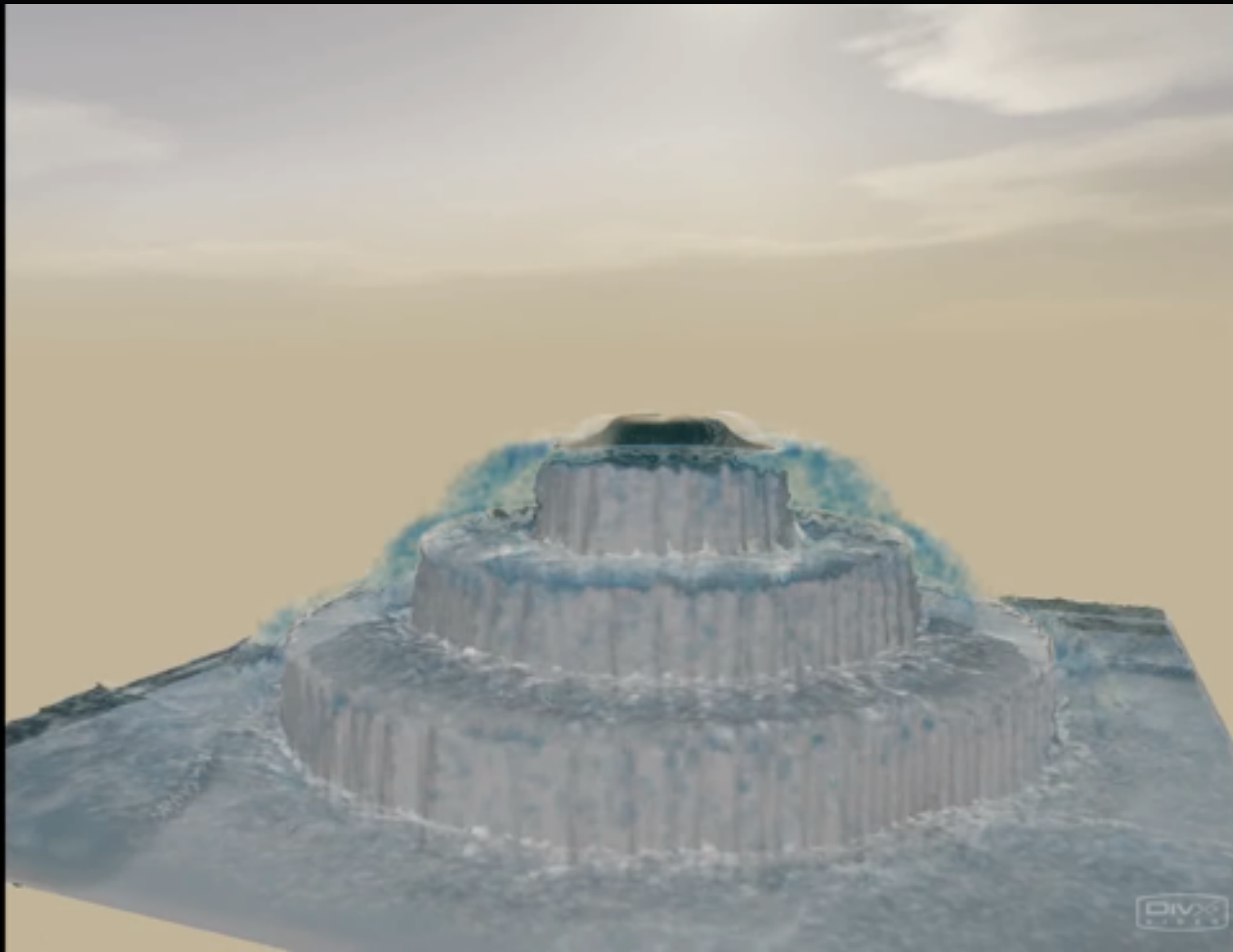
Waterfall

- Particles generation



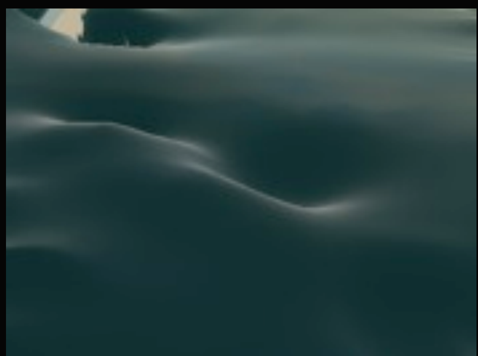
- Sample uniformly within red dotted box
  - Total mass should be the same as mass flow across the face
- Velocity found by interpolation
- Jitter initial position and velocity

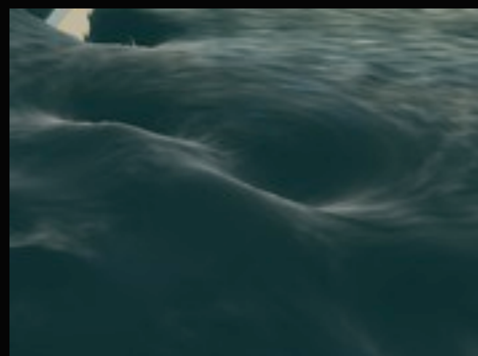
Waterfall



Adding small scale waves

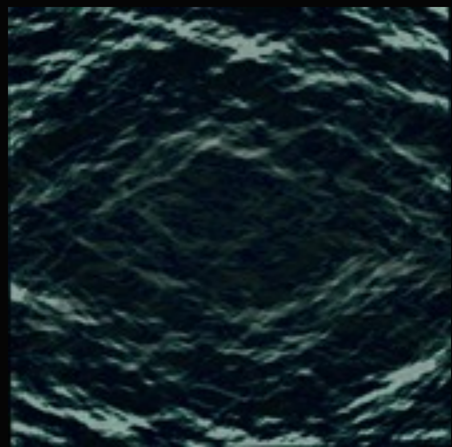
- Simulation cannot resolve waves with wave length $< \Delta x$
- Decreasing Δx may not be an option
- Still want small waves with the following properties
  - Advected with the velocity field
  - Not distorted excessively over time
  - Disappear if being stretched too much
  - Cheap to compute





Adding small scale waves

- Algorithm
  - Generate texture using FFT simulation



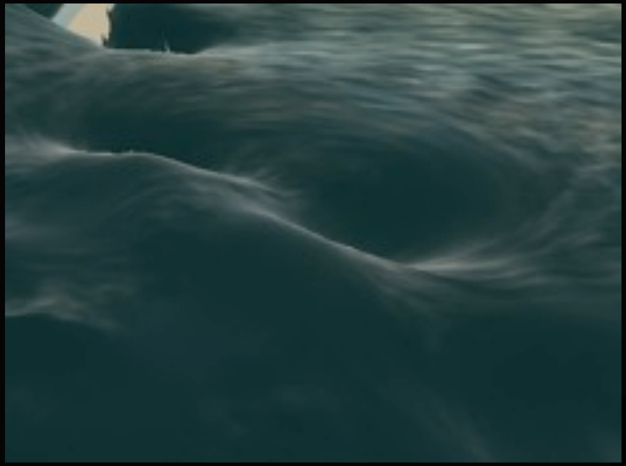
  - Advect 3 set of texture coordinates
  - Fetch texture and blend to get displacement map
  - Regenerate after some time

SIGGRAPHASIA2011 HONG KONG

Adding small scale waves

- So far, waves will never disappear
  - Wave persist even when being stretched a lot
  - Can have lava-like look
  - Need to suppress in region with too much stretch

Without suppressant



With suppressant

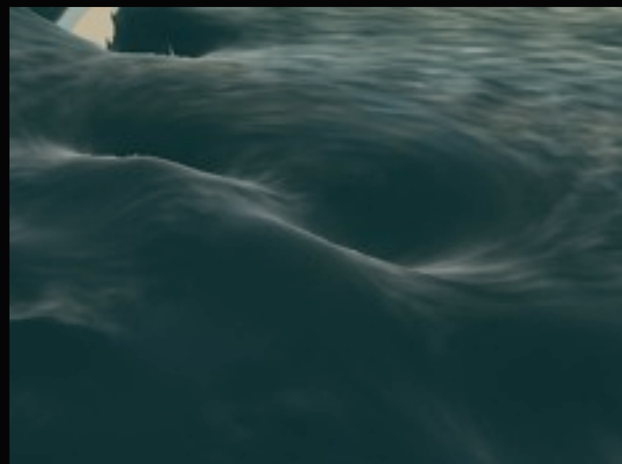
NVIDIA

SIGGRAPH ASIA 2011 HONG KONG

Adding small scale waves

- Measure of deformation
  - Use maximum eigenvalue of the Green Strain of the texture coordinates μ
- Modulate the final displacement
  - With an exponential decay $e^{-\Omega\mu}$

Without suppressant



With suppressant

SIGGRAPH ASIA 2011 HONG KONG

Adding small scale waves

NVIDIA

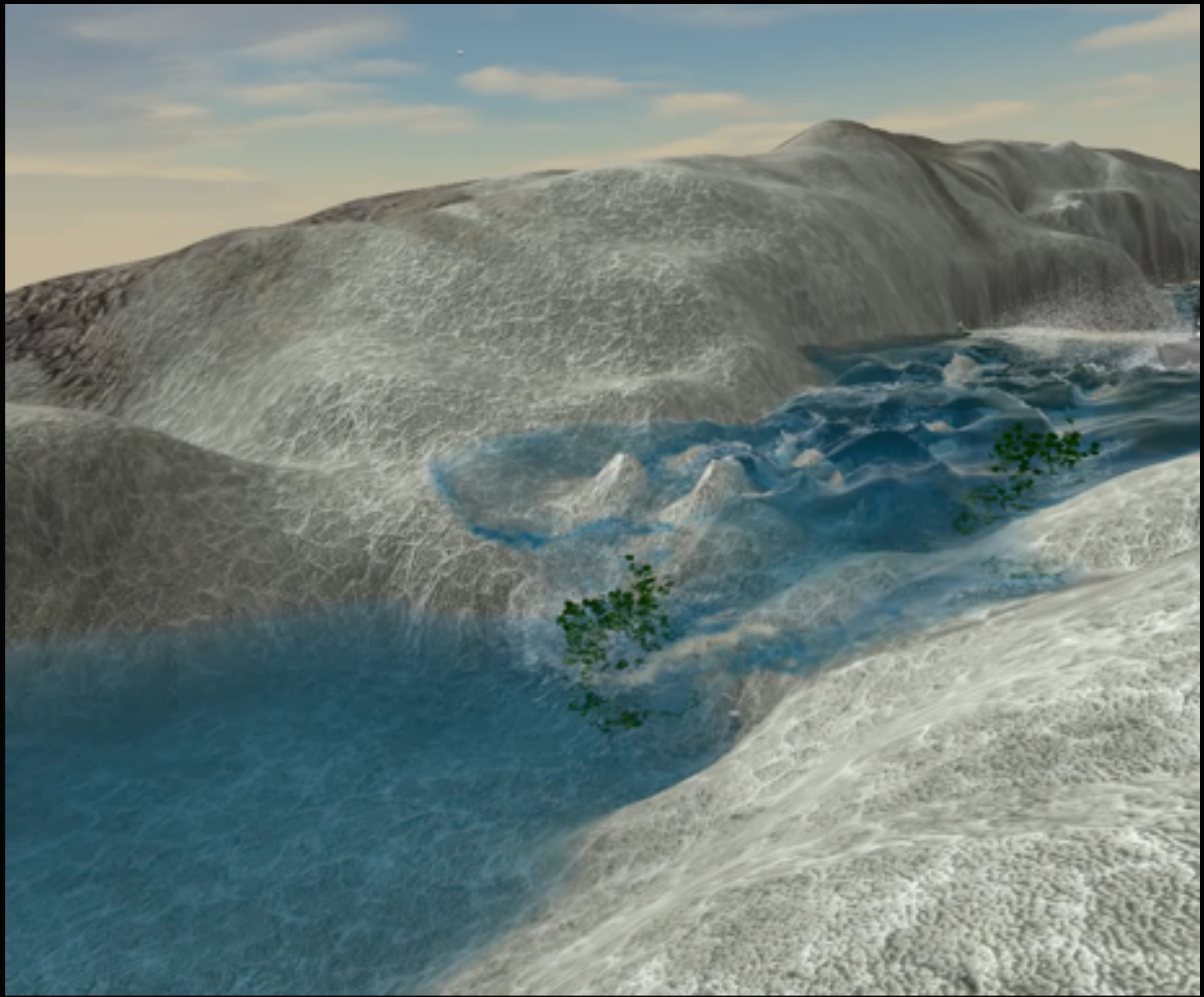
More results

Beach

128x128 grid
220K particles

SIGGRAPHASIA2011 HONG KONG

More results



Overview

- Approaches to realtime water simulation
- Hybrid shallow water solver + particles
- Hybrid 3D tall cell water solver + particles
- Future

Grid based 3D water simulation

- Small domain
- Computation increase with volume of water
- Also want small scale details
  - Splash
  - Foam
  - Spray

Keenan C. et al. 2007, "Real Time Simulation and Rendering of 3D Fluids"

3D Tall Cell Water Solver + Particles

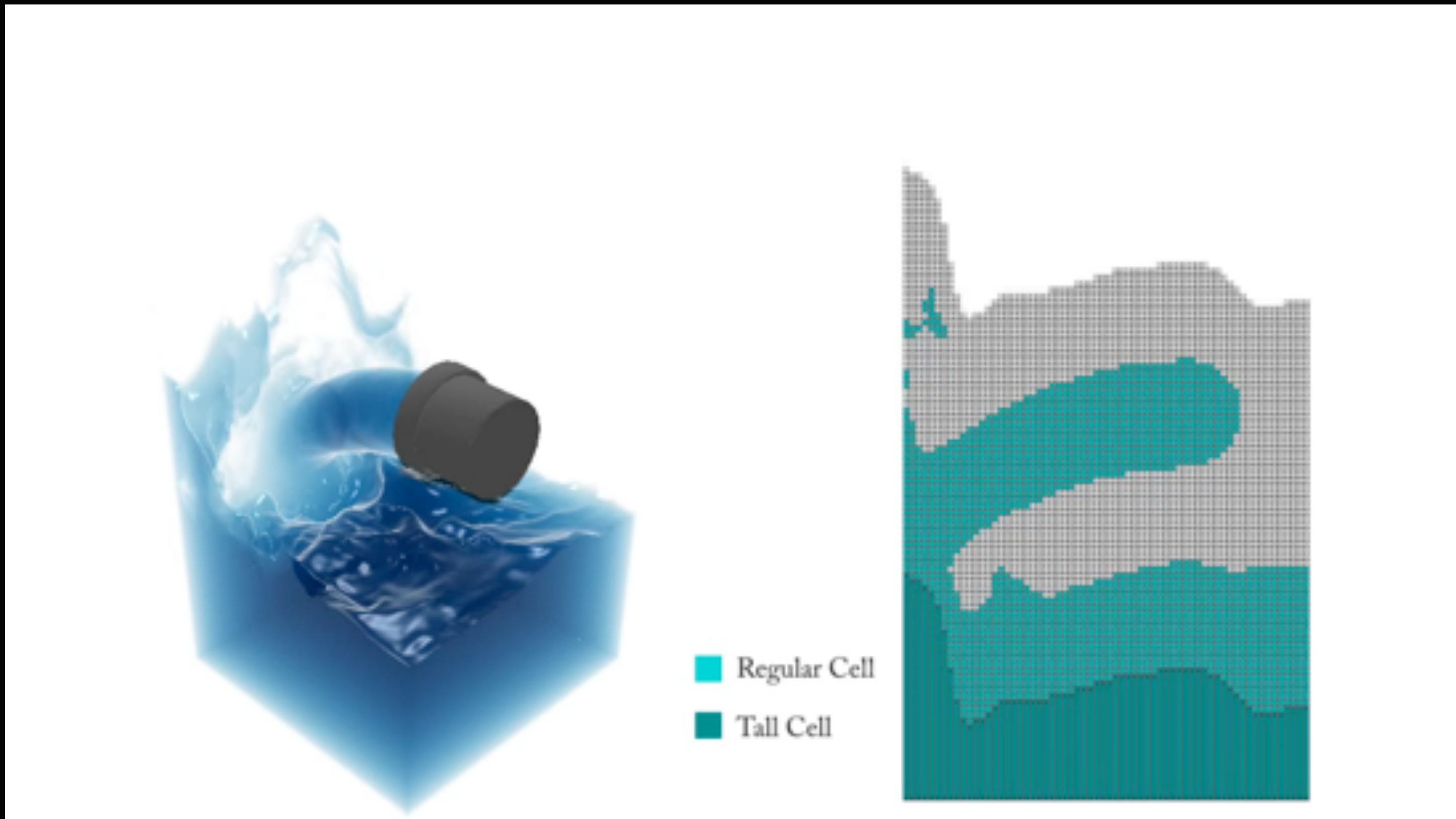
Flood

Spray particle generation
Flat steady state on arbitrary terrain

Grid: 256 x (32+2) x 64
100K Particles

Chentanez N. and Mueller M. 2011, "Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid"

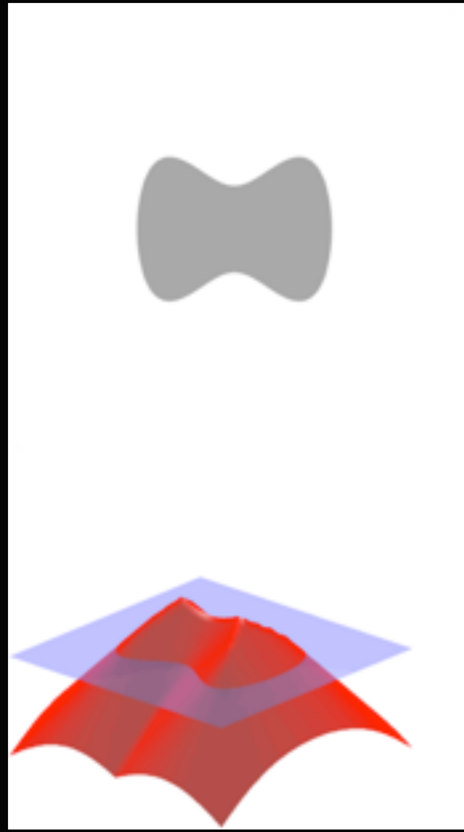
3D Tall Cell Water Solver + Particles



SIGGRAPHASIA2011 HONG KONG

3D Tall Cell Water Solver

- States of water
  - u – Velocity field
  - ϕ – Level Set (Signed distance function)
    - Positive inside water
    - Negative outside water
    - Zero on surface



http://en.wikipedia.org/wiki/File:Signed_distance2.png

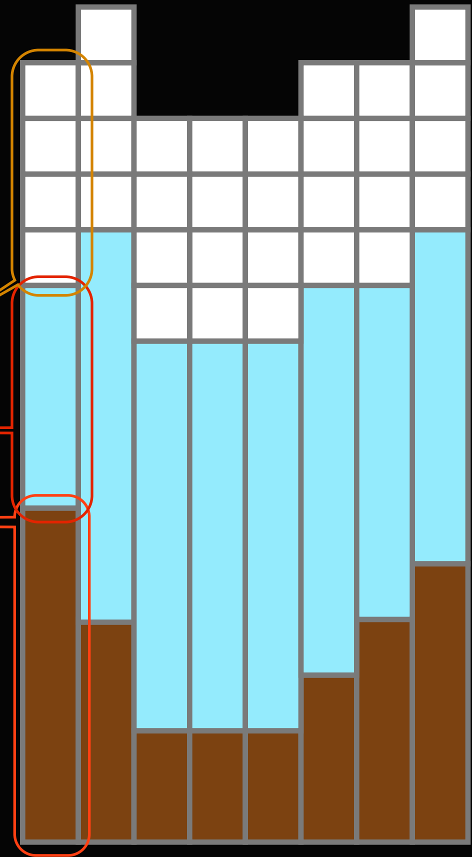
- Store states on grid points
  - Interpolate to get value everywhere
- Simulation == Rules to update these states

NVIDIA

SIGGRAPH ASIA 2011 HONG KONG

Discretization

- Tall cell grid
  - Each column consist of
    - Constant number of regular cells
    - One tall cell
    - Terrain

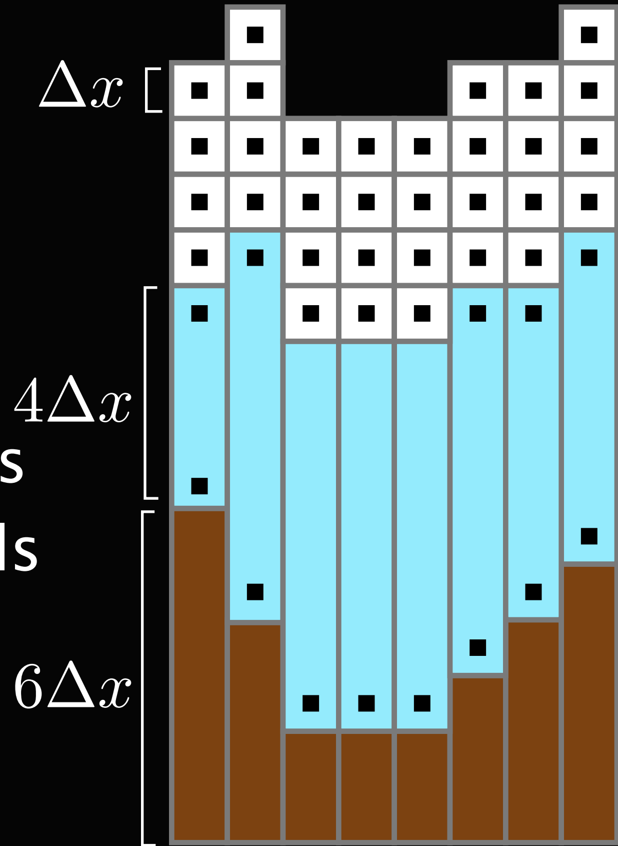


NVIDIA

SIGGRAPHASIA2011 HONG KONG

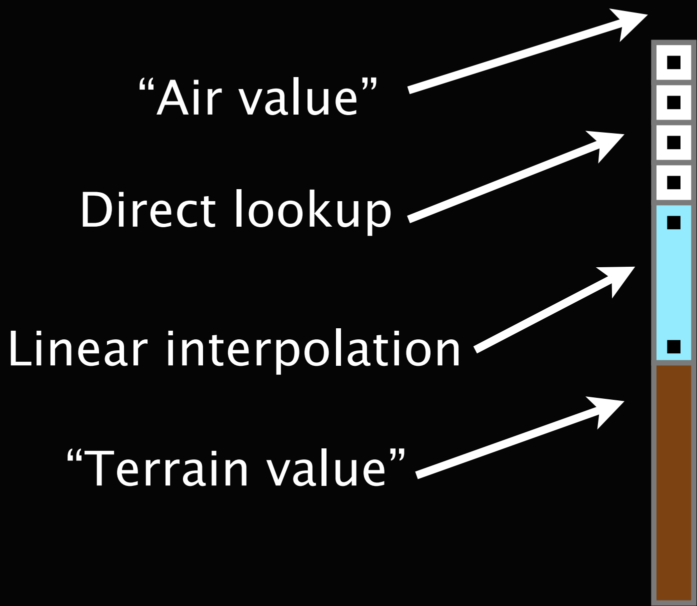
Discretization

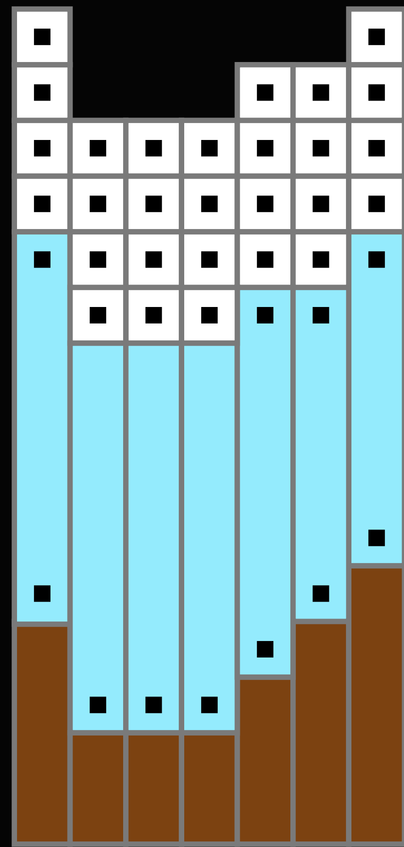
- Tall cell grid
  - Heights are multiple of Δx
  - Physical quantities u, ϕ
    - At cell center for regular cells
    - At top and bottom of tall cells



Discretization

- Tall cell grid
  - Quantity q at world position $(x\Delta x, y\Delta x, z\Delta x)$ denoted by $q(x, y, z)$
    - Hide tall cell structure of the grid

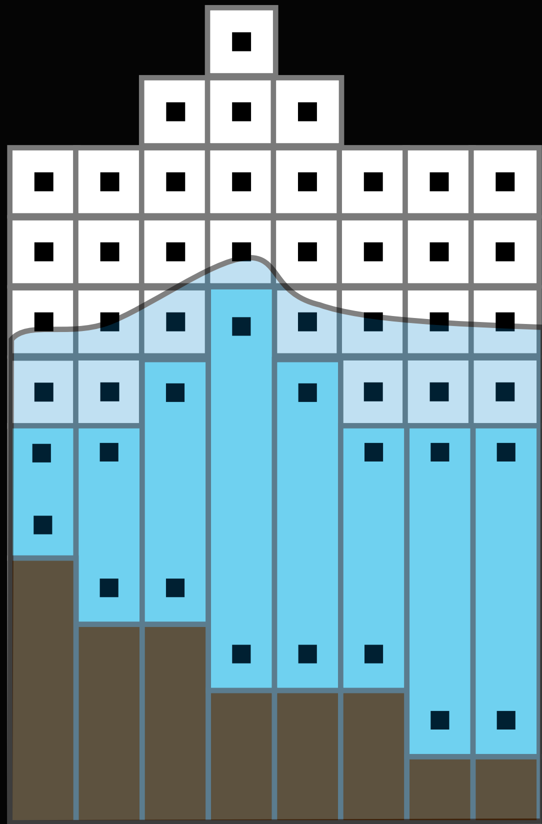




SIGGRAPHASIA2011 HONG KONG

Time integration

Extrapolate u to air

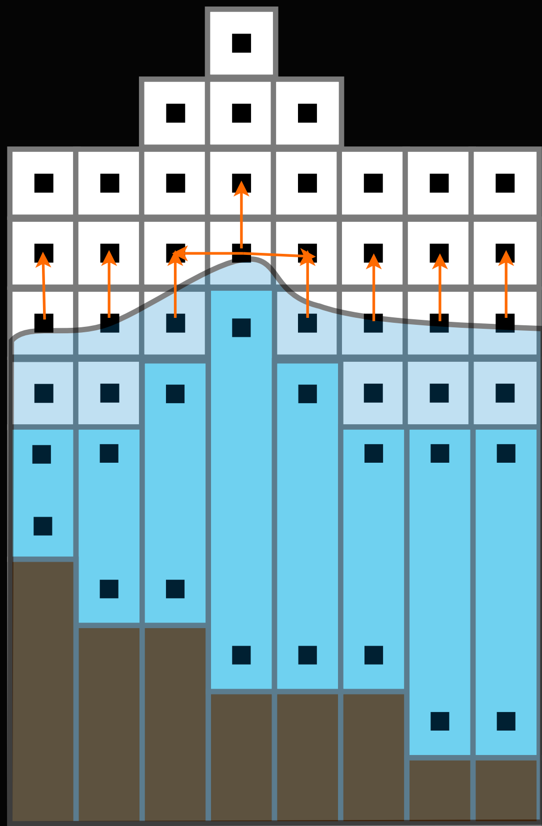


NVIDIA

SIGGRAPHASIA2011 HONG KONG

Time integration

Extrapolate u to air

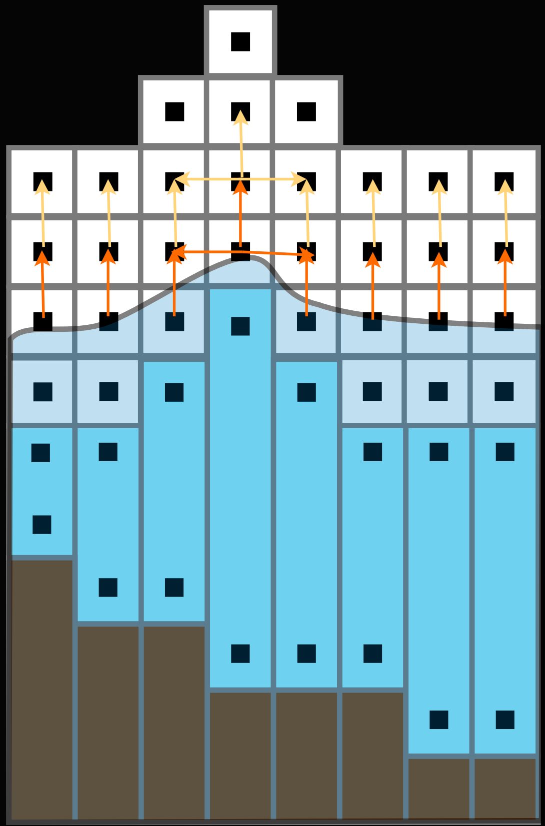


NVIDIA

SIGGRAPHASIA2011 HONG KONG

Time integration

Extrapolate u to air

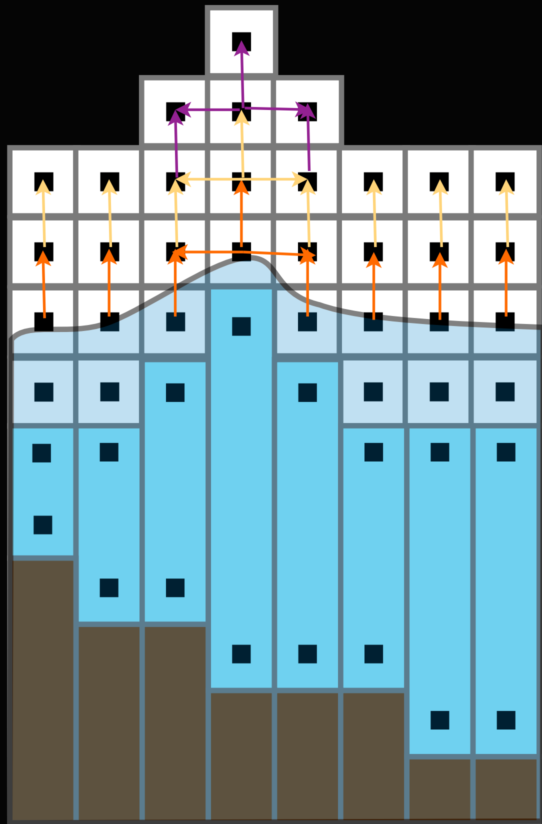


NVIDIA

SIGGRAPHASIA2011 HONG KONG

Time integration

Extrapolate u to air



NVIDIA

SIGGRAPHASIA2011 HONG KONG

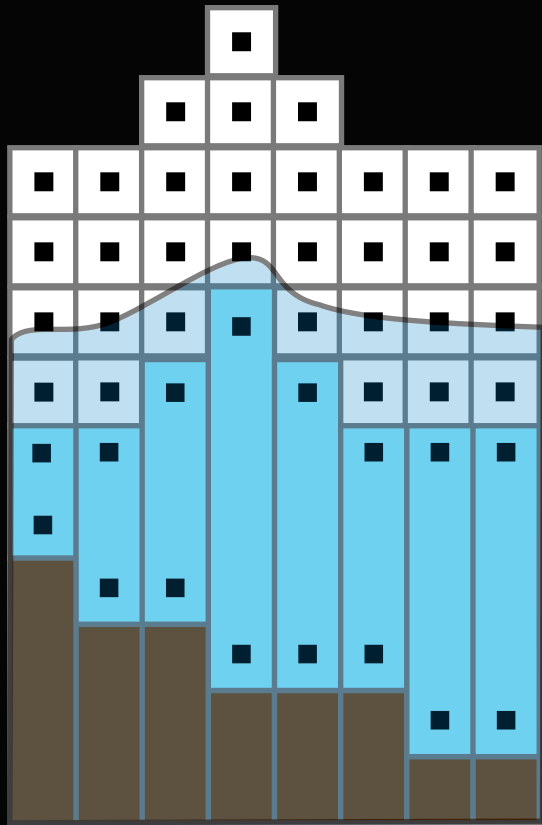
Time integration

Extrapolate u to air

↓

Make ϕ a signed distance function

Because ϕ will no longer be a signed distance function, as we update the states



NVIDIA

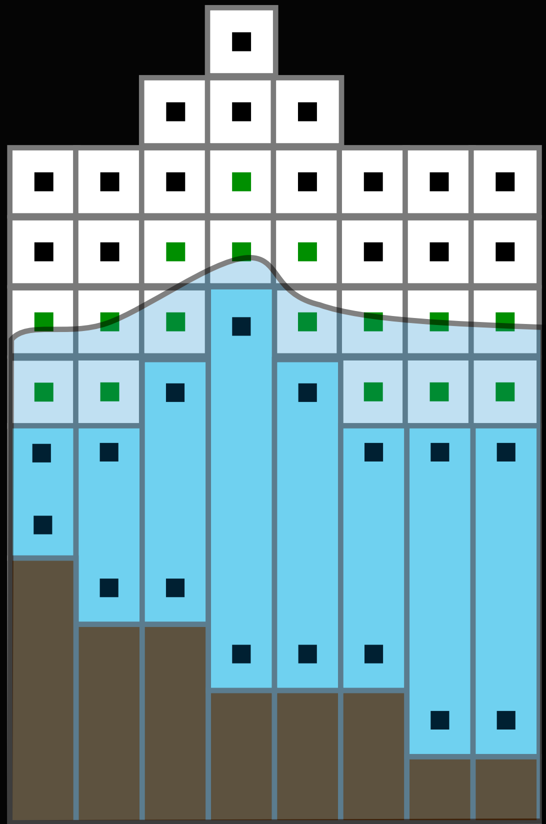
SIGGRAPHASIA2011
HONG KONG

Time integration

Extrapolate u to air

↓

Make ϕ a signed distance function



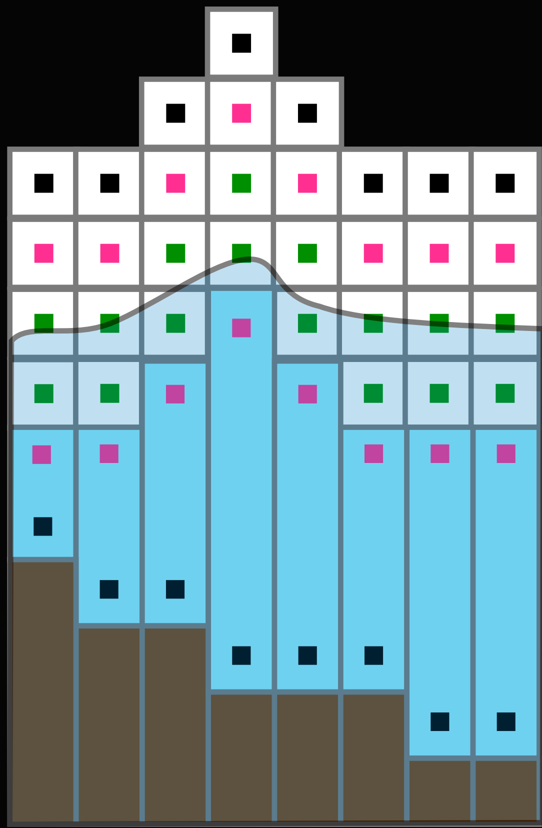
SIGGRAPHASIA2011 HONG KONG

Time integration

Extrapolate u to air

↓

Make ϕ a signed distance function

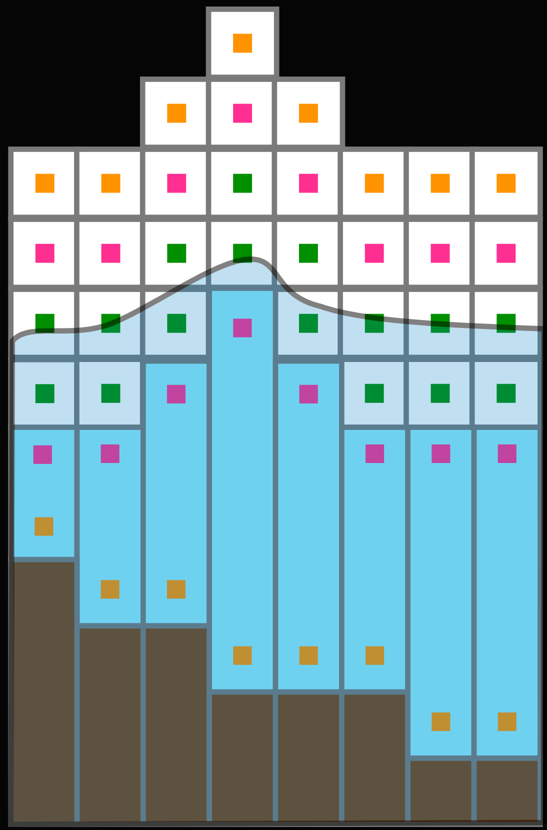


Time integration

Extrapolate u to air

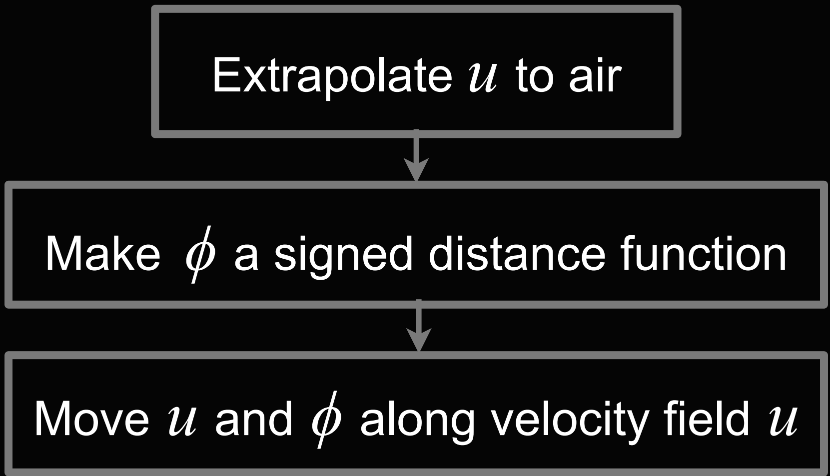
↓

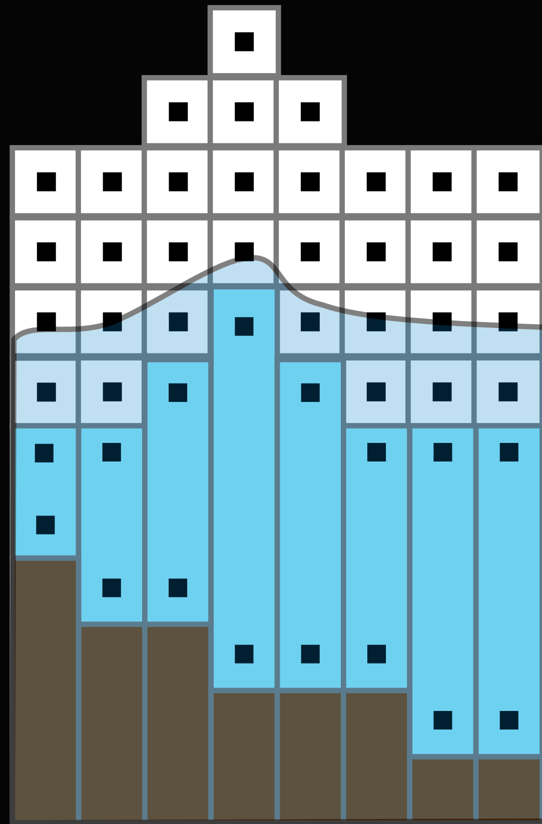
Make ϕ a signed distance function



SIGGRAPHASIA2011 HONG KONG

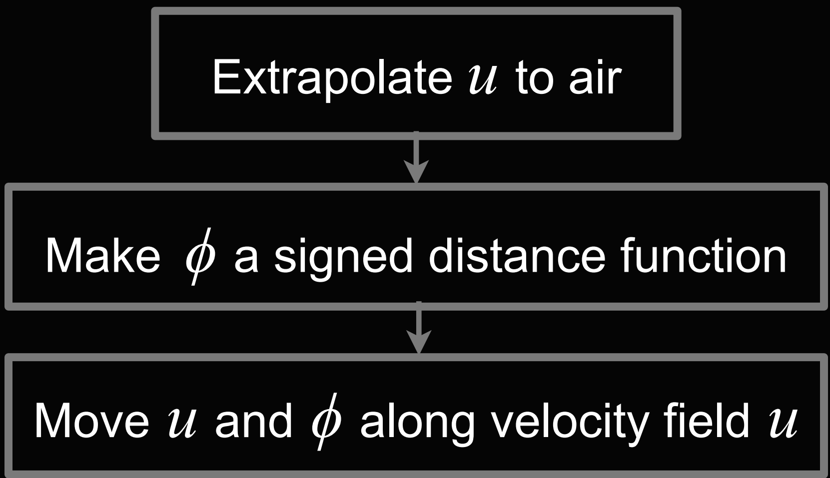
Time integration

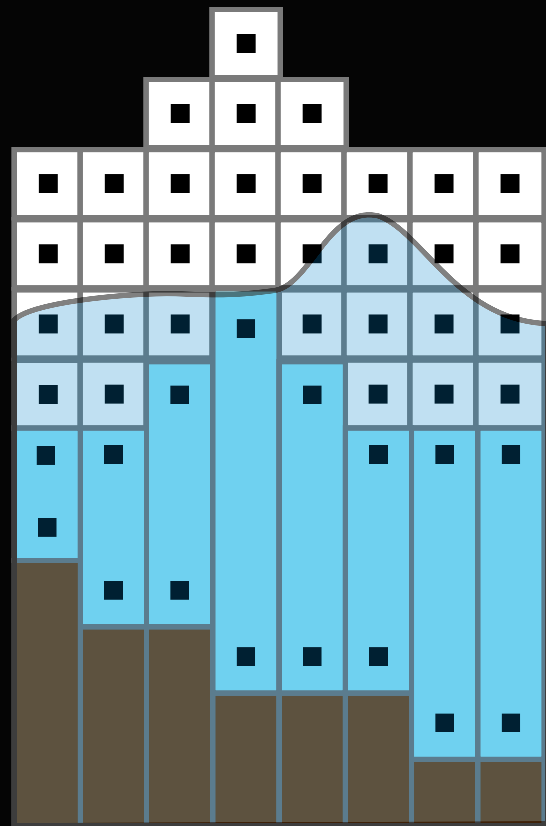




SIGGRAPHASIA2011 HONG KONG

Time integration





Time integration

Extrapolate u to air

↓

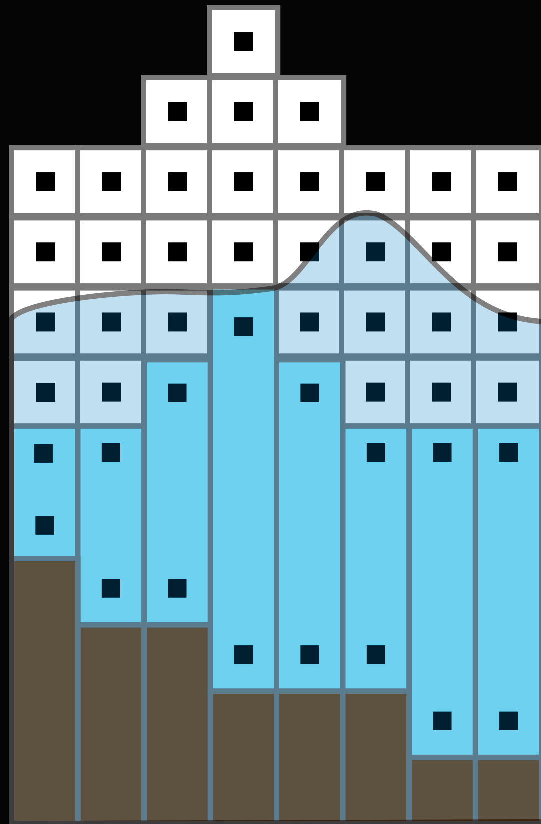
Make ϕ a signed distance function

↓

Move u and ϕ along velocity field u

↓

Adjust tall cell heights



Time integration

Extrapolate u to air

↓

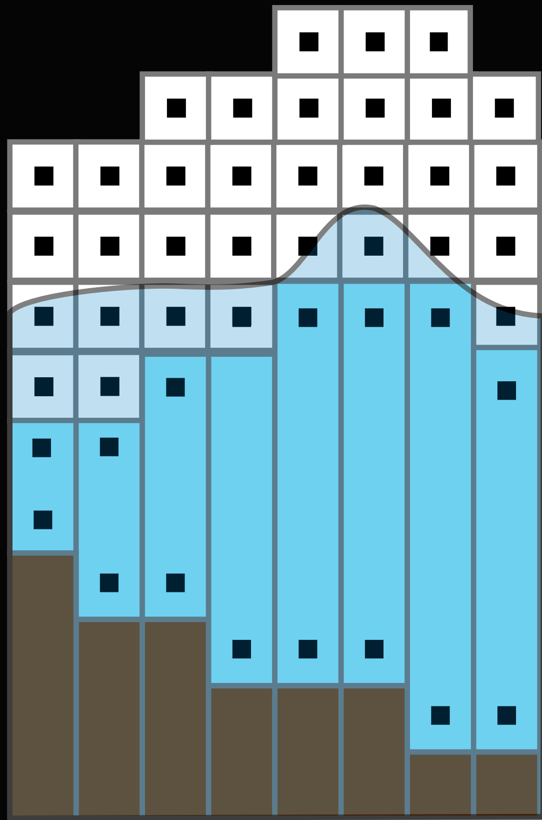
Make ϕ a signed distance function

↓

Move u and ϕ along velocity field u

↓

Adjust tall cell heights



Time integration

Extrapolate u to air

↓

Make ϕ a signed distance function

↓

Move u and ϕ along velocity field u

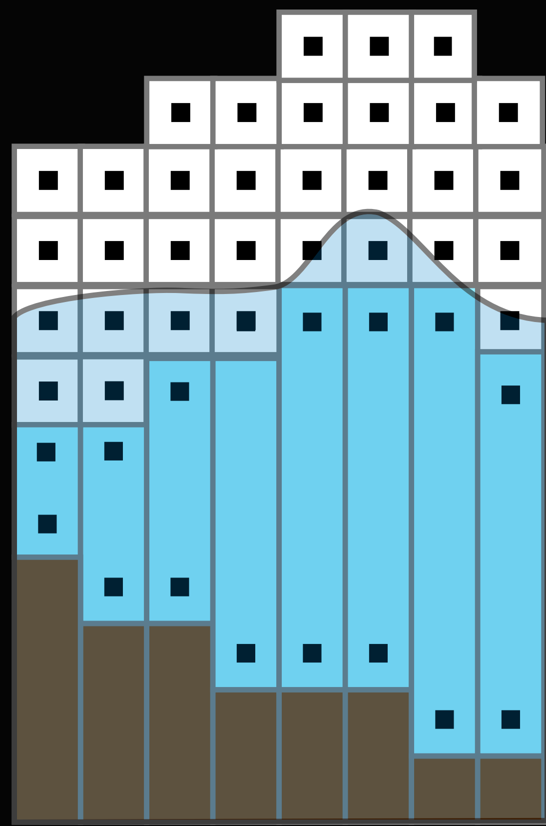
↓

Adjust tall cell heights

↓

Make u divergence free

The most difficult and time consuming step



Particles

- Spray
  - Seed particles inside grid cells whose ϕ are positive but small (near water surface)
  - Move them along velocity field u
  - After we update ϕ, for each particle
    - Check if ϕ at the current location is negative (outside water)
      - If so, generate spray particles
      - otherwise, ignore
  - Move ballistically
- Foam
  - Generate when spray particle falls into water
  - Move with u, projected to water surface

SIGGRAPHASIA2011 HONG KONG

More result

Chentanez N. and Mueller M. 2011, “Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid”

NVIDIA

Future

- Hybrid 3D + 2D + Particles + Procedural?
  - 3D + Particles near camera
  - 2D + Particles further away
  - 2D even further
  - Procedural far from camera

SIGGRAPHASIA2011 HONG KONG

Future

Thuerey N. et, al, 2006 “Animation of Open water Phenomena with coupled Shallow Water and Free Surface Simulation”

NVIDIA

Future

Thuerey N. et, al, 2006 “Animation of Open water Phenomena with coupled Shallow Water and Free Surface Simulation”

Future

- Hybrid 3D + 2D + Particles + Procedural?
  - 3D + Particles near camera
  - 2D + Particles further away
  - 2D even further
  - Procedural far from camera
- Dynamic LOD
  - Best quality within budget

Q&A

Thank you very much!

References

- Tessendof SIGGRAPH Course 99, “Simulating Ocean Wave”
- Enright et al. SIGGRAPH 02, “Animation and Rendering of Complex Water Surfaces”
- Mueller et al. SCA 03, “Particle-Based Fluid Simulation for Interactive Applications”
- Thuerey N. et, al, SCA 06 “Animation of Open water Phenomena with coupled Shallow Water and Free Surface Simulation”
- Keenan C. et al. GPU Gems III 07, “Real Time Simulation and Rendering of 3D Fluids”
- Yuksel et. al. SIGGRAPH 07, “Wave Particles”
- Stava et al. SCA 08, “Interactive Terrain Modeling Using Hydraulic Erosion”
- Hilko et al. Eurographics workshop on natural phenomena 09, “Real-Time Open Water Scenes with Interacting Objects”
- Long B. and Reinhard E. I3D 09 ,”Real-Time Fluid Simulation Using Sine/Cosine Transforms”
- Chentanez N. and Mueller M. SCA 10, “Real-time Simulation of Large Bodies of Water with Small Scale Details”
- Brodtkorb A. R. et al. Computer & Fluid 11, “Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation”
- Chentanez N. and Mueller M. SIGGRAPH 11, “Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid

NVIDIA

Math

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

SIGGRAPH ASIA 2011 HONG KONG

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

Change in velocity

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

NVIDIA

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

Change in velocity

Velocity moves itself

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

Change in velocity

Velocity moves itself

External force such as gravity

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

SIGGRAPHASIA2011 HONG KONG

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

Change in velocity — Velocity moves itself — External force such as gravity — Pressure gradient

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

Change in velocity; Velocity moves itself; External force such as gravity; Pressure gradient

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

- Subject to $\nabla \cdot \mathbf{u} = 0$

Incompressibility
– What come in must go out

- Inside region $\phi < 0$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

Change in velocity

Velocity moves itself

External force such as gravity

Pressure gradient

- Subject to

$$\nabla \cdot \mathbf{u} = 0$$

Incompressibility
– What come in must go out

- Inside region $\phi < 0$

Implicit function that represent water body

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

NVIDIA

3D Tall Cell Water Solver

- Solve Inviscid Euler Equations

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{\mathbf{f}}{\rho} - \frac{\nabla p}{\rho}$$

Change in velocity

Velocity moves itself

External force such as gravity

Pressure gradient

- Subject to

$$\nabla \cdot \mathbf{u} = 0$$

Incompressibility
– What come in must go out

- Inside region $\phi < 0$

Implicit function that represent water body

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla \phi$$

Implicit function moved by water velocity