Simulating Water and Smoke Using an Octree Data Structure

A paper by Frank Lossaso, Frederic Gibou, Ron Fedkiw, February 2004

Presented by Venkat Krishnaraj

Overview

- The Motivation
- Background Review - Octree
- The Octree Method
- Water and Smoke Simulation
- Mesh Refinement and Simulation – Problems and Solutions
- The Refinement Criteria
- Results

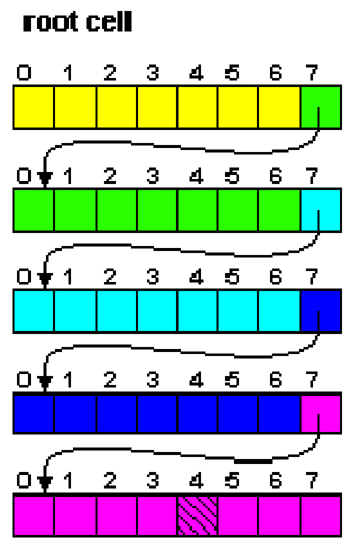
The Motivation

The Motivation

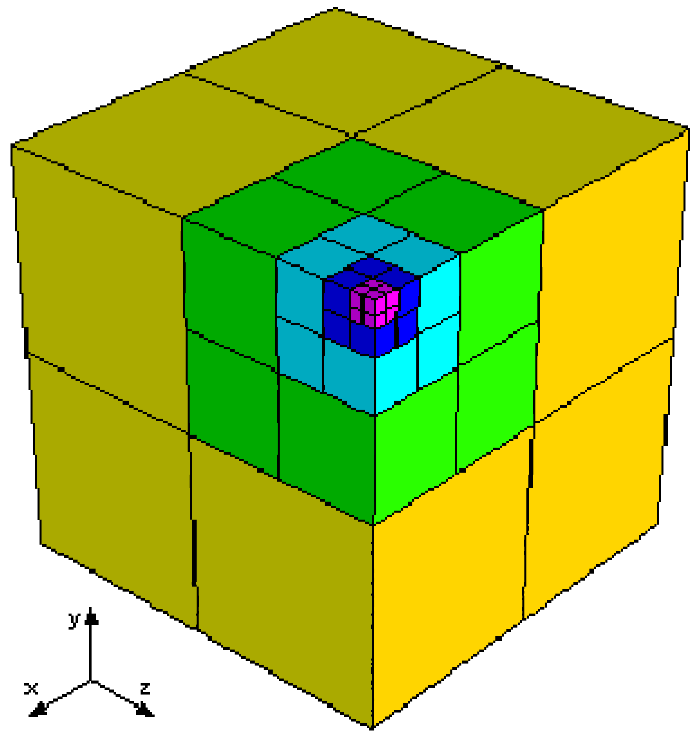
- We want to capture small details
- However, we cannot increase grid resolution (for computational efficiency reasons)
- Hence, we want to use an Adaptive Mesh Refinement strategy
- Refinement criteria : Where things are sort of interesting

Background Review - Octree

Octrees (Quadtrees): Axis aligned, regularly spaced planes cut space into cubes (squares)



		0	1	2	3	4
Z	31	1	1	1	1	1
Y	30	1	1	1	1	0
X	30	1	1	1	1	0
		7	7	7	7	4



http://nilo.stolte.free.fr/octree.html

Background Review - Octree

Types of Octree

Restricted Octree

A condition or constraint is specified while growing the tree

Unrestricted Octree

No such conditions

Motivation : A balanced tree (or to optimize calculations)

The Octree Method

- More grids at visually interesting region
- [Popinet 2003] has the first octree implementation
- This paper extends that (basically to make it more efficient via subtle assumptions)

The Octree Method [Popinet 2003]

- Standard Spatial Discretization
- Each cell C has a direct neighbour at the same level in each direction => 6 neighbours
- Each of them is accessed through a face
- In addition, the paper defines mixed cells as cells cut by an embedded boundary
- Add restriction to simplify the calculations

The Octree Method [Popinet 2003]

- Constraints Added:
- The levels of direct neighboring cells cannot differ by more than one
- The levels of the diagonally neighboring cells can not differ by more than one
- All the cells directly neighboring a mixed cell must be at the same level
- Last constraint is most restrictive

The Octree Method [Popinet 2003]

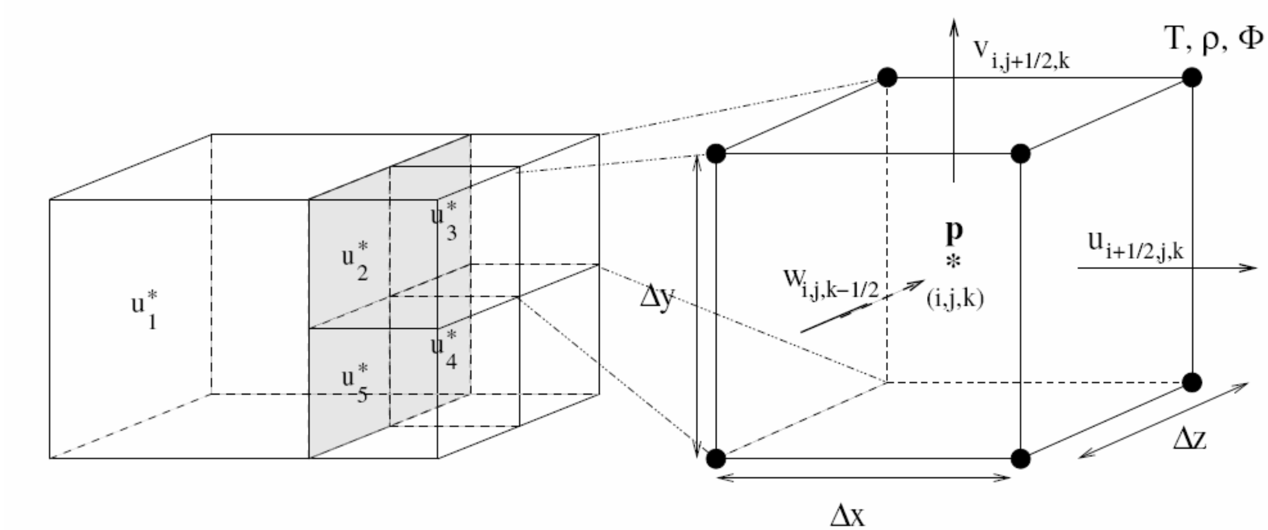
Images

The Octree Method – This paper

- In [Popinet 2003], the constraints were added to simplify calculations of the gradient and flux
- However, this paper relaxes those constraints and allows unrestricted octrees
- Also, allows the modeling of the liquid interface

The Octree Method – This paper

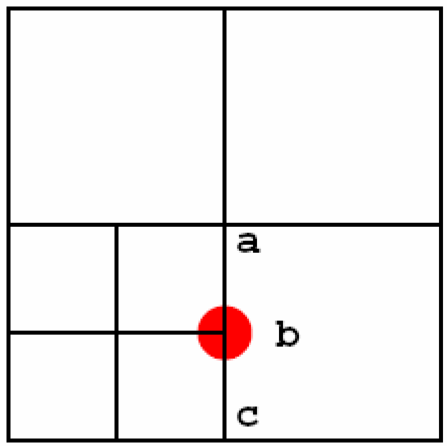
- Original MAC grid : Scalars (pressure, density and viscosity) defined at cell centers, and velocities at cell faces
- All scalars except pressure are stored at the node corners (since interpolations are harder with cell centered data)



The Octree Method – This paper

- While coarsening, nodal values are deleted or unchanged and the values for the faces are the average of the old valuesDuring Refining, values for the new face centers / edges is got by simple interpolation
- For an edge, its an average of the two neighbors, and for a face, its an average of the 4 corners from the surrounding cells
- The Evil T-joint

The Octree Method – T Joint



- T-Joints are problematic because we need to ensure that calculations done at b are an average of only a & c
- Basically, even during traversals, T-joints need to be handled as a special case
- This paper is no different, where, T-joints are detected earlier and handled specially

Water and Smoke Simulation

- Determine cell contents depending upon the tracking method (Hybrid Level Set)
- Set up boundary conditions for surface cells
- Compute velocities for the cells without using the pressure term (Navier-Stokes Equations)
- Perform pressure iteration for all full cells (Conservation of mass)
- Compute velocities for surface cells (divergence free velocity field)
- Update positions

- Ignore viscosity
- Determine cell contents depending upon the tracking method
- Set up boundary conditions for surface cells
- Compute velocities for the cells without using the pressure term (Navier-Stokes Equations)
- Perform pressure iteration for all full cells (Conservation of mass)
- Compute velocities for surface cells (divergence free velocity field)
- **Compute buoyant forces**
- **Vorticity Confinement**

Water and Smoke Simulation [Fedkiw, Stam, Jensen]

- Smoke Simulation Equations

- Bouyant Forces

Water and Smoke Simulation [Fedkiw, Stam, Jensen]

- Vorticity Confinement

- Analogous to the Hybrid Level Set which uses curvature and particle distance to track the surface

Mesh Refinement and Simulation

- The Divergence Equation
- Green’s theorem
- The new form of the equation

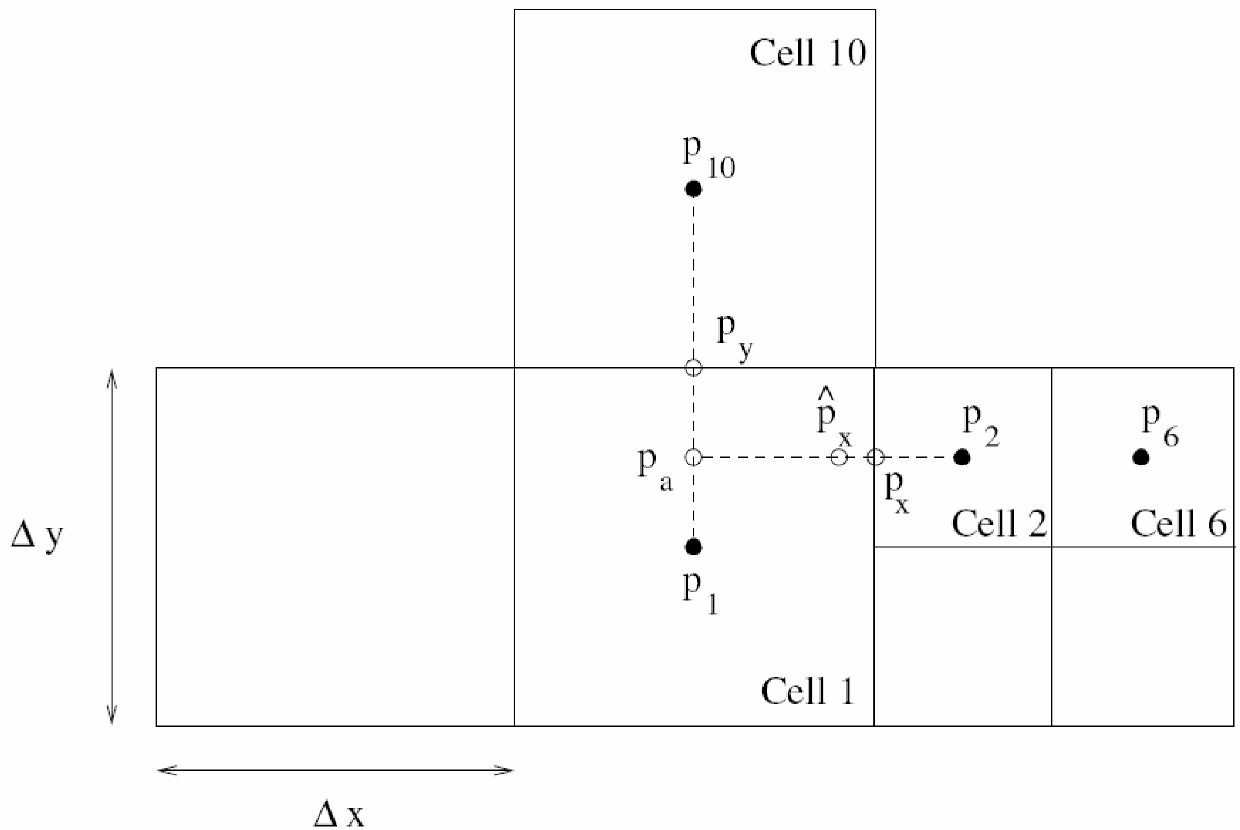
Mesh Refinement and Simulation

- The Divergence Equation
- Green’s theorem
- The new form of the equation

Mesh Refinement and Simulation

- We need to make subtle changes to the algorithm to enable it to use Octrees
- Mostly, these involve traversing the octrees and averaging operations that need to be performed as we move from cell-center components to face components
- T-junction handling
- Careful discretizing of pressure

Mesh Refinement and Simulation



Mesh Refinement and Simulation

- There exists many choices to discretize deltap
- However, we choose the one which when plugged into a Poisson Solver yields a symmetric matrix which is invertible
- Basically, a Poisson Solver can be used to solve second order differential equations

Mesh Refinement and Simulation

- The choice of discretizing deltap is not physically based
- However, results seem accurate
- Speed tradeoff is large, and since the resulting matrix is symmetric, we can perform efficient matrix inversion (PCG or Preconditioned Conjugate Gradient)
- The preconditioner used in this paper is the LU Cholesky factorization

Mesh Refinement Criteria

- Smoke
  - High Vorticity
  - Band of density values (0.1 to 0.3)
  - Near objects
- Water
  - Where the isocontour of the level-set evaluates to 0 (or surface)
  - Biased towards the side containing the liquid

- Water simulation now uses a 2^{nd} order Runge-Kutta for the particles

Results

- Symmetric formulation reduces simulation time to 25% of the original values
- Reasonable computational costs on a 1024x1024x1024 grid

Results

- Dominant errors were due to the first order advection scheme
- 20 iterations to converge to an accuracy of machine precision

Questions?