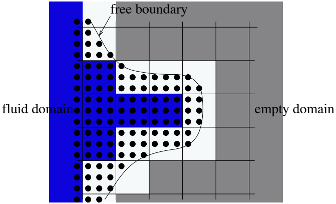
Free Surface Flows

M. Farahani, E. Wolter, A. Hahn

03.02.2014

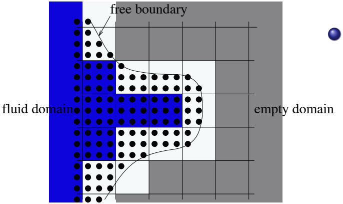
Application

Theory



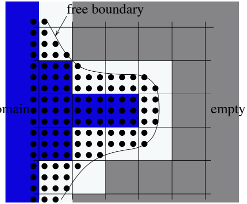
- the stress tensor: $\sigma = (-P + \lambda div\vec{u})\, I + 2\mu\delta$

Theory



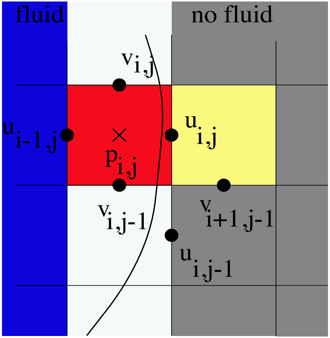
- the stress tensor: $\sigma = (-P + \lambda div\vec{u})\, I + 2\mu\delta$
- $P + \frac{2}{Re}\left(n_x n_x \frac{\partial u}{\partial x} + n_x n_y \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + n_y n_y \frac{\partial v}{\partial y}\right) = K\kappa$

Theory



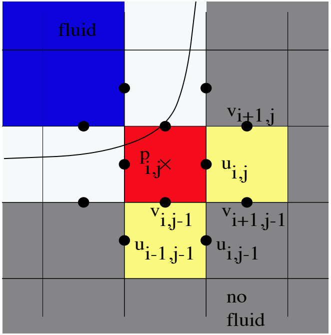
- the stress tensor: $\sigma = (-P + \lambda div\vec{u})\, I + 2\mu\delta$
- $P + \frac{2}{Re}\left(n_x n_x \frac{\partial u}{\partial x} + n_x n_y\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + n_y n_y \frac{\partial v}{\partial y}\right) = K\kappa$
- $2n_x m_x \frac{\partial u}{\partial x} + (n_x m_y + n_y m_x)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + 2n_y m_y \frac{\partial v}{\partial y} = 0$

One empty neighbor



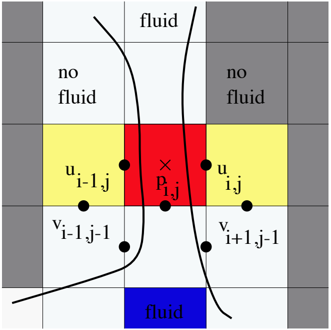
- free boundary lie almost parallel to the grid lines
- $n_y \& m_x = 0 \quad Rn_x \& m_y = 0$
- $P = \frac{2}{Re}\frac{\partial u}{\partial x}$
- $\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0$
- using continuity equation

Two empty neighbor-common corner



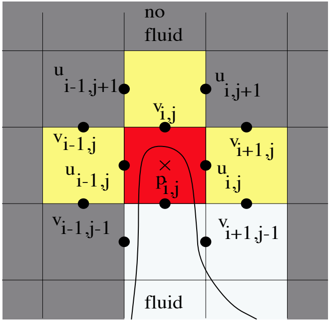
- $n_y = m_x = n_x = m_y$
- $P = \pm \frac{1}{Re} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial x} \right)$
- $\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} = 0$

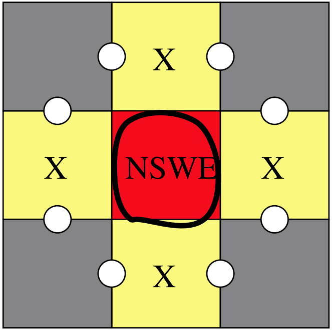
Two empty neighbor-opposite side



- $u_{i,j}^{new} = u_{i,j}^{old} + \delta t g_x$
- $u_{i-1,j}^{new} = u_{i-1,j}^{old} + \delta t g_x$
- $v_{i,j}^{new} = v_{i,j}^{old} + \delta t g_y$
- $v_{i,j-1}^{new} = v_{i,j-1}^{old} + \delta t g_y$

Three empty neighbor





Particle and ParticleTracer

- **Particle(real x, real y, int type)**
  Has some functions which can detect its position on the grid
- **ParticleTracer(StaggeredGrid *grid)**
  Has a vector of particles
  - **void markCells()**
  - **void fillCell(int i, int j, int numParticles, int type)**
  - **void addRectangle(real x1, real y1, real x2, real y2, int type)**
  - **void addCircle(real x, real y, real r, int type)**
  - **void advanceParticles(real const dt)**

Types and StaggeredGrid

- **Types.hh**:
  - **flag EMPTY**
- **StaggeredGrid.cc**:
  - **int ppc_**
  - **bool isEmpty(const int x, const int y)**
  - **void setCellToEmpty(int x, int y)**
  - **void refreshEmpty()**

FluidSimulator

- **FluidSimulator.cc**:
  - **real rectX1_particle_, rectX2_particle_ , ...**
  - **real circR_particle_, circX_particle_, ...**
  - **void set_UVP_surface(int i, int j , const real &dt, bool compP)**
  - **void one_empty_neighbour(int i , int j , const real &dt, bool compP)**
  - ...
  - **four_empty_neighbour(int i , int j , const real &dt, bool compP)**
  - **void refreshEmpty()**

Main while-loop

```
while (n <= nrOfTimeSteps)
{
...  determineNextDT(safetyfac_);
     particle_tracer_.markCells();
     set_UVP_surface(dt_, true);
     computeFG();
     composeRHS();
     solv().solve(grid_);
     updateVelocities();
     refreshBoundaries();
     set_UVP_surface(dt_, false);
...  particle_tracer_.advanceParticles(dt_);
}
```

Examples

- The Breaking Dam - Outflow
- The Breaking Dam - Freeslip
- The Splash of a Liquid Drop

The Breaking Dam

```
imax = 50,           jmax = 20,
xlength = 10.0,      ylength = 4.0,
tau = 0.5,           delt = 0.04,        t_end = 5.0,
eps = 0.001,         omg = 1.7,
gamma = 0.5,         itermax = 500,
GX = 0.0,            GY = -1.0,          Re = 10.0,
UI = 0.0,            VI = 0.0,           PI = 0.0,
ppc=16,

wW = free,           wE=out,
wS = free,           wN=out
```

The Splash of a Liquid Drop

```
imax = 40,          jmax = 30,
xlength = 8.0,      ylength = 6.0,
tau = 0.2,          delt = 0.01,     t_end = 10.0,
eps = 0.001,        omg = 1.7,
gamma = 0.5,        itermax = 500,
GX = 0.0,           GY = -1.0,       Re = 40.0,
UI = 0.0,           VI = 0.0,        PI = 0.0,
ppc=16,

wW = free,          wE=free,
wS = free,          wN=out
```