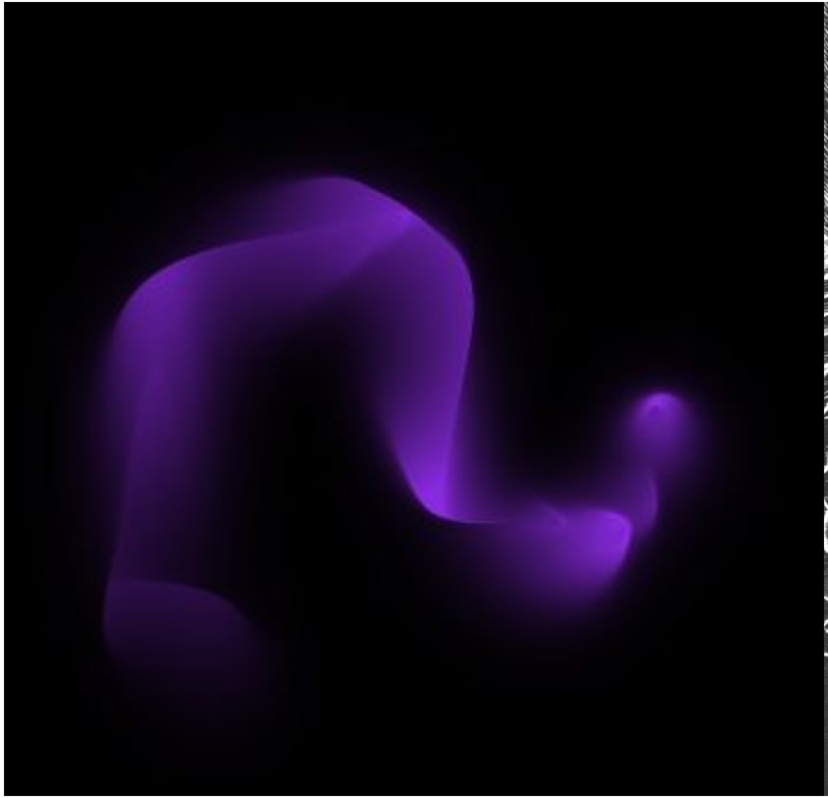
正波紋, Hamon

2D Stable Fluid Simulation

106062216 馮謙

106062120 史孟玄

オーバードライブ



Notations

Operator	Definition	Finite Difference Form
Gradient	$\nabla p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}\right)$	$\frac{p_{i+1,j} - p_{i-1,j}}{2\delta x}, \frac{p_{i,j+1} - p_{i,j-1}}{2\delta y}$
Divergence	$\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$	$\frac{u_{i+1,j} - u_{i-1,j}}{2\delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\delta y}$
Laplacian	$\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$	$\frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{(\delta x)^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{(\delta y)^2}$

The Navier-Stokes Equations

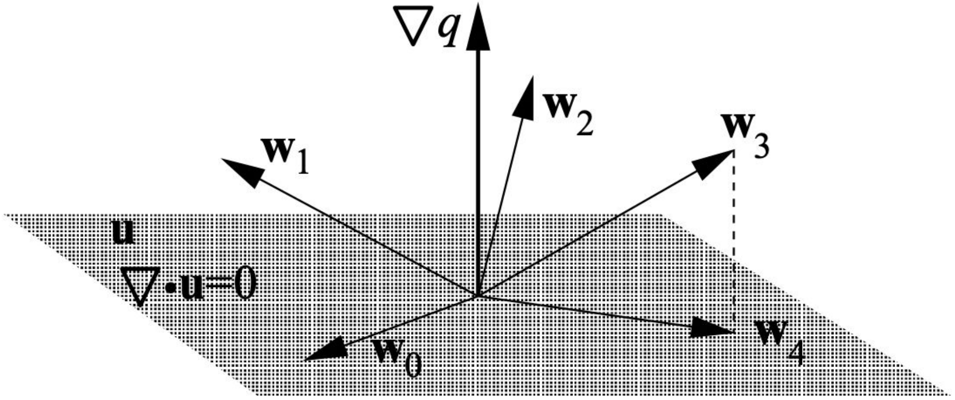
$$
\begin{aligned}
\nabla \cdot \mathbf{u} &= 0 \\
\frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f},
\end{aligned}
$$

The Helmholtz-Hodge Decomposition

$$\mathbf{w} = \mathbf{u} + \nabla q,$$

1. Where u has zero divergence, and q is a scalar field

2. Any vector field can be decomposed into a divergence-free field and a gradient filed



First Realization

We get a nonzeror divergence vector field ***w*** by computing “advect”, “diffusion”, and “force”.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

and then we can subtract the gradient of pressure to get the divergence-free vector field ***u***

$$\mathbf{u} = \mathbf{P}\mathbf{w} = \mathbf{w} - \nabla q.$$

Second Realization

The theorem also leads to a method for computing the pressure field. If we apply the divergence operator to both sides:

$$\nabla \cdot \mathbf{w} = \nabla \cdot (\mathbf{u} + \nabla p) = \nabla \cdot \mathbf{u} + \nabla^2 p .$$

but the term [del x u] equals to 0, so we get the following equation, which is also known as the Poisson equation.

$$\nabla^2 p = \nabla \cdot \mathbf{w},$$

Pipeline

1. Compute “advect”, “diffusion”, and “force” to get w
2. Apply divergence operator to solve gradient of p
3. Finally, w subtrats gradinet p to get u

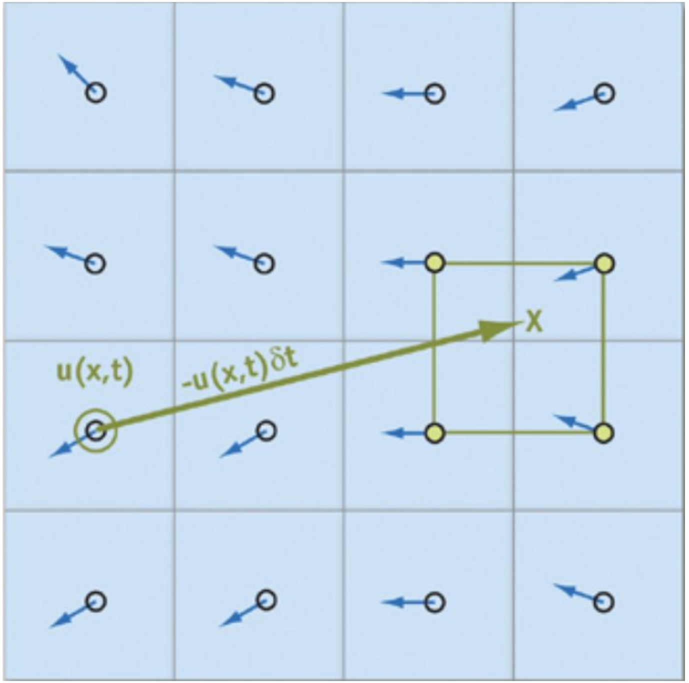
Force

To apply force on the vector field is easy, we simply add the force vector on the gridblock and the scale it with a scalar.

$$F\delta t \exp\left(\frac{\left(x - x_p\right)^2 + \left(y - y_p\right)^2}{r}\right).$$

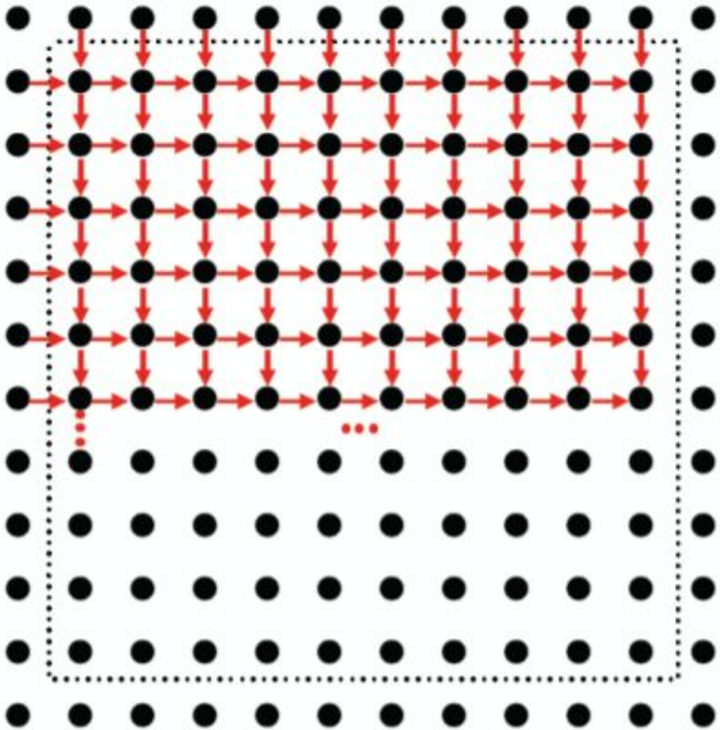
Advect

1. Advection is the process by which a fluid's velocity transports itself and other quantities in the fluid
2. However, directly compute position of quantitied in the fluid leads to unstable process
3. Thus, we follow the velocity field “back in time” to get the velocity of the grid



Diffusion and Pressure

Diffusion and pressure can be solve iteratively, which can be solved by applying Gauss-Siedel algorithm or Jacobi method



Parallelism

We compute each pixel on cuda core individually

use cudaMallocManaged(), cudaMemPrefetchAsync() for GPU memory manage

Performance improve

with only CPU : 10+-fps, 150 * 150 will reach max CPU load

with GPU : 100+-fps, 400* 400 run smoothly