Dartmouth COSC 89/189 Computational Methods for Physical Systems, Fall 2019

Instructor: Bo Zhu (bo.zhu@dartmouth.edu)

Course Notes 4: Grid-based Fluid

We are solving the inviscid and incompressible Navier-Stokes equations on a uniform grid:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla \cdot \nabla \vec{u} + \vec{g}$$

$$\nabla \cdot \vec{u} = 0,$$

with $\vec{u}$ as velocity, ρ as density, ν as kinematic viscosity, and p as pressure.

We consider our simulation as a 2D problem, with constant unit density ($\rho = 1$), zero viscosity (ν=0), and zero body force ($\vec{g} = 0$). Typically, we call the Navier-Stokes equations without viscosity as the Euler's equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p$$

$$\nabla \cdot \vec{u} = 0,$$

To solve the problem numerically, we discretize the spatial domain by a uniform Cartesian grid with $N_x \times N_y$ grid nodes. Each grid cell has the size of Δx. The data samples for both velocity (2D vector) and pressure (1D scalar) are stored on grid nodes. The Euler's equation can be discretized in the temporal domain as:

$$(\vec{u}^{n+1} - \vec{u}^{n})/\Delta t + \vec{u} \cdot \nabla \vec{u} = -\nabla p$$

$$\nabla \cdot \vec{u}^{n+1} = 0,$$

We split the first equation into two parts by introducing an intermediate velocity $\vec{u}^*$:

$$\vec{u}^* = \vec{u}^n - \Delta t(\vec{u}^n \cdot \nabla \vec{u}^n)$$

$$\vec{u}^{n+1} = \vec{u}^* - \nabla p \Delta t$$

We call the first equation the "advection" step and the second equation the "projection" step. Intuitively, advection step evolves the current velocity field and the projection step guarantees that the obtained new velocity field is divergence free. The divergence-free condition is satisfied by substituting the projection equation into the divergence free condition for time $n + 1$, where we will get

$$\nabla \cdot \vec{u}^{n+1} = \nabla \cdot (\vec{u}^* - \nabla p \Delta t) = 0,$$

which can be further written as,

$$\nabla \cdot \nabla \hat{p} = \nabla \cdot \vec{u}^*$$

This is a Poisson equation with the pressure as the unknowns and the intermediate velocity divergence as the right hand side. Notice that we absorb the Δt into $\hat{p}$ to simplify the equation, that is, $\hat{p} = p\Delta t$. We will get the pressure values by solving this Poisson equation, and then we substitute the pressure values back into the projection equation to get the divergence-free velocity field for the next time step.

Advection We use the semi-Lagrangian scheme for advection. The idea is to back-trace a virtual particle from the current position to interpolate a velocity value which will flow into this position in the next time step. Given a position $\vec{x}$ and its velocity $\vec{u}$ in time n, the velocity in time step $n+1$ is calculated as:

$$\vec{u}^*(\vec{x}) = Interpolate(\vec{u}^n, \vec{x} - \vec{u}^n(\vec{x})\Delta t\ )$$

We use $*$ instead of $n+1$ because the updated velocity is not the final result but just an intermediate variable which will be further processed in the projection step. We use the bi-linear interpolation scheme on a 2D grid to implement the interpolation function. Notice that the advection is a Jacobi process which means the input field for interpolation won't change during the updates of the different nodes.

For better numerical accuracy, we choose the velocity in the location by back-tracing with a half time step, instead of directly using the velocity in the location of $\vec{x}$. This can be written as:

$$\vec{u}^{n+\frac{1}{2}}(\vec{x}) = Interpolate(\vec{u}^n, \vec{x} - \vec{u}^n(\vec{x})\frac{\Delta t}{2}\ )$$

$$\vec{u}^*(\vec{x}) = Interpolate(\vec{u}^n, \vec{x} - \vec{u}^{n+\frac{1}{2}}(\vec{x})\Delta t\ )$$

Projection Given a grid node (i,j), its gradient, divergence, curl, and its Laplacian can be approximated by its four neighbors and itself as:

$$(\nabla p)_{i,j} = \frac{1}{2\Delta x}\begin{pmatrix} p_{i+1,j} - p_{i-1,j} \\ p_{i,j+1} - p_{i,j-1} \end{pmatrix}$$

$$(\nabla \cdot \vec{u})_{i,j} = \frac{1}{2\Delta x}(u_{i+1,j} - u_{i-1,j} + v_{i,j+1} - v_{i,j-1})$$

$$(\nabla \times \vec{u})_{i,j} = \frac{1}{2\Delta x}(v_{i+1,j} - v_{i-1,j} - u_{i,j+1} + u_{i,j-1})$$

$$(\nabla \cdot \nabla p)_{i,j} = \frac{1}{\Delta x^2}(p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j})$$

with $\vec{u} = (u,v)^T$ as a 2D vector and p as a scalar.
We take three steps for projection: calculating the velocity divergence for the rhs of the Poisson equation, solving the Poisson equation for pressure, and enforcing the divergence-free condition by subtracting the pressure gradient. We do all the three steps in a matrix-free way. First, the velocity divergence on each grid node is calculated as:

$$(\nabla \cdot \vec{u})_{i,j} = \frac{1}{2\Delta x}(u_{i+1,j} - u_{i-1,j} + v_{i,j+1} - v_{i,j-1})$$

Second, to solve the Poisson equation $\nabla \cdot \nabla \hat{p} = \nabla \cdot \vec{u}^*$, we update the pressure on each grid node iteratively by employing the Gauss-Seidel smoothing operator:

$$\hat{p}_{i,j} = [-(\nabla \cdot \vec{u}^*)_{i,j} + \frac{1}{\Delta x^2}(\hat{p}_{i-1,j} + \hat{p}_{i+1,j} + \hat{p}_{i,j-1} + \hat{p}_{i,j+1})]/(\frac{4}{\Delta x^2})$$

After a fixed number of iterations, we believe the pressure is accurate enough to get an approximately divergence free velocity. The last step for velocity projection is implemented as:

$$\vec{u}_{i,j}^{n+1} = \vec{u}_{i,j}^{*} - (\nabla\hat{p})_{i,j}$$

Notice that there should have a Δt with the pressure gradient in the original equation, but since we use the modified pressure $\hat{p}$ instead of p we do not need it in the correction step.

Vorticity Confinement In order to preserve the vorticity details in the flow field, which were vastly damped out due to the interpolations happened in every advection step, we introduce an artificial vorticity confinement force as:

$$\vec{f}_{vc} = \epsilon\Delta x(\vec{N} \times \vec{\omega})$$

with

$$\vec{\omega} = \nabla \times \vec{u}$$

and

$$\vec{N} = \frac{\nabla|\omega|}{\|\nabla|\omega|\|}$$

The goal of this force is to enhance the local rotational motion of the fluid by adding a force along the direction of the local rotation ($\vec{N}$). Here vorticity field $\vec{\omega}$ is defined as the curl of the velocity field, and ϵ is a parameter that can be adjusted to control the confinement effect.

The force is added to the velocity stored on each grid node as

$$\vec{u}_{i,j}^{**} = \vec{u}_{i,j}^{*} + \vec{f}_{i,j}\Delta t$$

We omit the mass inverse term in the equation by assuming $\rho = 1$.