Numerical simulation of the Navier-Stokes equations with an image-based initial condition

Charbel Z. J. MAMLANKOU

1- General Overview

We consider the Navier-Stokes system defined by:

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

where $\mathbf{u}$ is the fluid velocity vector, p the pressure, ρ the density, ν the kinematic viscosity, and $\mathbf{f}$ the external forces. We use the model of **Jos Stam** ("Stable Fluids") by decomposing the resolution of the equations into several sequential steps. To preserve a perpetual motion dynamic in the fluid, external forces are added to the velocity field. Jos's decomposition allows us to consider the following equation to sequentially determine this velocity:

$$\mathbf{u}^* = \mathbf{u} + \Delta t \cdot \mathbf{f}$$

We use random forces (randomly generated) in our case. The objective is to achieve fluid movement starting from an initial condition based on an image. Therefore, it is appropriate to introduce a diffusive model to propel the velocity diffusion over time and keep the fluid in motion. This is where the heat equation comes into play, modeled by:

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \nabla^2 \mathbf{u}$$

An implicit FDM scheme was used for its discretization to ensure numerical stability. One of the crucial steps in this simulation is the **projection for incompressibility**. This step ensures that the velocity field satisfies the incompressibility condition $\nabla \cdot \mathbf{u} = 0$. A **velocity field advection** was also considered, corresponding to the nonlinear term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ in the Navier-Stokes equations.

The density (image intensity) is treated as a passive scalar transported by the fluid. Its dynamics are governed by (**advection/diffusion**):

$$\frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla)\rho = \kappa \nabla^2 \rho$$

where κ is the diffusion coefficient. These equations are solved similarly to those of the velocity field.

2- Discrete Formulation

The domain is discretized on a cellular grid (finite difference method). For a scalar field ϕ, the Laplacian operator is approximated by:

$$\nabla^2 \phi_{i,j} \approx \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{h^2}$$

where h represents the spatial step. The divergence is approximated by:

$$\nabla \cdot \mathbf{u}_{i,j} \approx \frac{u_{i+1,j} - u_{i-1,j} + v_{i,j+1} - v_{i,j-1}}{2h}$$

We then use an implicit Euler scheme for diffusion (to ensure stability) and a semi-Lagrangian scheme for advection.

We use periodic boundary conditions, and for a scalar field like density, values are mirrored at the edges:

$$\phi_{0,j} = \phi_{1,j}, \phi_{n+1,j} = \phi_{n,j}, \phi_{i,0} = \phi_{i,1}, \phi_{i,n+1} = \phi_{i,n}$$

For the normal components of velocity, conditions are inverted to ensure no flux crosses the boundaries.

Note: In our application, the image is treated as an initial density field. Its grayscale levels are normalized between 0 and 1 to form the initial condition. The pixel intensity then follows the fluid transport equations, creating an image "mixing" effect over time.

The code for this simulation is available on the Github repository:

$$https://github.com/MCDev30/NS\_CFD$$