Technical Report

Toffalini Leonardo

FIRST DRAFT

1 Notation

In this report we are going to heavily rely on differential operators, as this project is about solving a partial differential equation. Thus, before delving into the details of the main topic at hand, we shall quickly remind the reader of the notations that the following sections rely on. For a more complete introduction to the notations, we refer the reader to consult Besenyei, Komornik, Simon: Parciális differenciál egyenletek 4.5.

Let ∇ (formally) represent the $(\partial_1, \partial_2, ..., \partial_n)$ vector, where n is always deduced by the context the operator is used in.

With the help of this vector we can define the gradient of a field as

$$\nabla u = (\partial_1 u, \partial_2 u, ..., \partial_n u).$$

Moreover, as ∇ is a vector, we can apply vector operations to it to get different operations, such as

$$u \cdot \nabla = \sum_{i=1}^{n} u_i \partial_i$$

or

$$(u \cdot \nabla) u = \sum_{i=1}^{n} u_i \partial_i u.$$

The Δ operator, called the **Laplace operator**, sometimes written as $\nabla \cdot \nabla$ or ∇^2, acts as follows:

$$\nabla \cdot \nabla u = \Delta u = \sum_{i=1}^{n} (\partial_i u)^2.$$

2 The equations of fluids

The advective form of the **Navier-Stokes equations** for an incompressible fluid with uniform viscosity are as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \Delta \boldsymbol{u} - \frac{1}{\rho} \nabla p + \frac{1}{\rho} \boldsymbol{f}. \tag{1}$$

where

- $\boldsymbol{u}$ is the velocity vector field.
- $\boldsymbol{f}$ is the external forces.
- ρ is the scalar density field.
- p is the pressure field.
- ν is the kinematic viscosity.

We will not go into the details of deriving the above equations, instead we will just take it as granted that they truly formulate the evolving velocity of a viscous incompressible fluid.

The intrigued reader may find satisfaction in exploring the derivation of the above equations in [1].

For some, that have not yet encountered the differential operators used in the above formulation, we give a short summary:

Explanation of the different parts of Equation 1:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \overbrace{(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}}^{\text{Advection}} = \underbrace{\nu \Delta \boldsymbol{u}}_{\text{Diffusion}} \quad \overbrace{-\frac{1}{\rho}\nabla p}^{\text{Internal source}} \quad + \quad \underbrace{\frac{1}{\rho}\boldsymbol{f}}_{\text{External source}} \quad .$$

In broad strokes the parts can be described as follows:

1. Advection – How the velocity moves.
2. Diffusion – How the velocity spreads out.
3. Internal source – How the velocity points towards parts of lesser pressure.
4. External source – How the velocity is changed subject to external intervention, like a fan blowing air.

3 Equations for fluid simulation

It is best to mention here, in this section, that the described method will not be exact. This, however, will not be to our detriment, as our aim here is to show interesting and realistic visuals as opposed to precise measurements useful for engineering efforts.

For our purposes we will rearrange Equation 1 such that we have only the time derivate of the velocity on the left and omit the internal force part, as we will recover this part later. So we arrive at the equation which we will solve:

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nu \Delta \boldsymbol{u} + \frac{1}{\rho}\boldsymbol{f}.$$

The evolving of the velocity field in and of itself is not that interesting to see, because in real life we do not see the velocity field, we only see it's effect. To provide a more stimulating visual experience we must let the velocity field act upon a density field and in term visualize said density field. We present the equation for the density field

$$\frac{\partial \rho}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\rho + \kappa \Delta \rho + S,$$

where κ is the diffusion coefficient of the fluid and S is a scalar field of external sources.

The astute reader might find this equation quite similar to that of the velocity field, which is by no means a coincidence as the same broad strokes apply to the density field too, which are: advection, diffusion, and external sources. The only thing we must be careful of, is that the density is a scalar field, in contrast to the velocity, which is a vector field.

The partial differential equation by itself does not result in a unique solution for us to find, for this we must also specify a boundary condition. In our endeavor to simulate a fluid, we will work with the Dirichlet-boundary condition, which prescribes that the values on the boundary must vanish. We can formulate the previous statement as follows:

$$\begin{aligned} u|_{\partial\Omega} &= 0 \\ \rho|_{\partial\Omega} &= 0, \end{aligned}$$

where Ω represents the domain on which we are searching for the solution.

4 Simulating fluids

The equations presented in the previous sections hold on the entire n-dimensional space in which the fluid resides. However, when numerically solving a partial differential equation one often discretizes the space into small rectangles and only bothers to calculate the solution on these rectangles. For our purposes we will confine ourselves to the 2-dimensional plane, but everything mentioned hereafter can be easily extended to higher dimensions.

We will make another simplification, which is to only consider a rectangular domain. This makes the calculations easier to handle as matrix operations, however, one is free to extend the domain to more complex shapes by exercising proper caution when handling the boundary. Figure 1 shows a schematic diagram of how one must imagine a grid on a rectangular domain. Notice, how we introduced a 0th and an $(N+1)$th row and column to handle the boundary, this way it is clear that we will be simulating the fluid on an $N \times N$ grid on the inside.

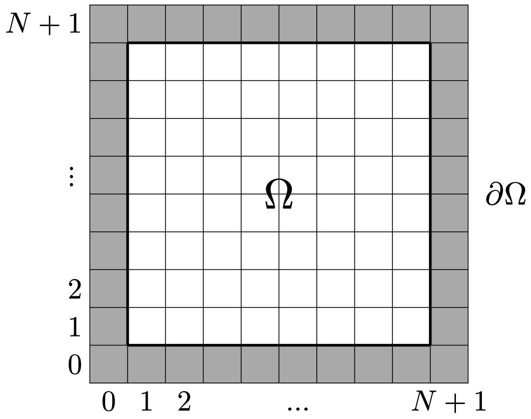

Figure 1: Discretization of the rectangular domain.

5 Moving densities

In our simulation, we will solve the parts of the simplified Navier-Stokes equations for the density field one by one. Starting with the simplest, adding additional density, then diffusion, and lastly advection. The simulation steps can be seen in Figure 2, where one most imagine applying the steps in order from left to right, and repeating the last three steps, while the simulation is running.

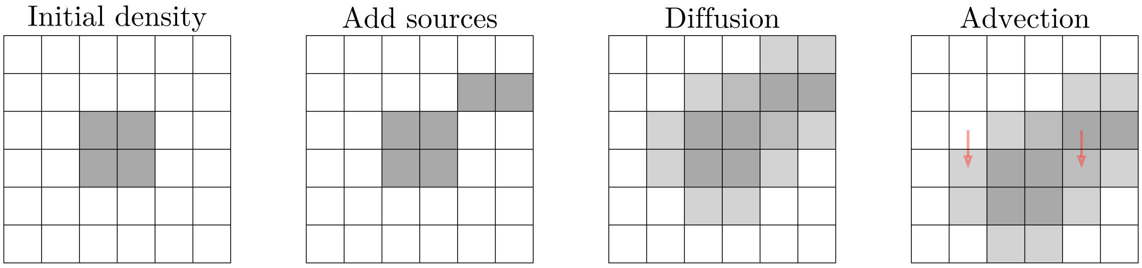

Figure 2: Rough outline of the density solver

5.1 Adding sources

Possibly the simplest step of the simulation is to add new sources to the density field, or to the velocity field for that matter. For a rectangular grid this step can be simplified to a simple matrix addition $\rho_h + S_h$, where ρ_h is the density field on the discretized domain, and S_h is the matrix containing the sources at each grid cell.

5.2 Diffusion

The diffusion step boils down to solving the simple Poisson equation

$$\frac{\partial \rho}{\partial t} = \kappa \Delta \rho.$$

To solve this, we are going to employ the most intuitive method, known as the finite difference method, where we think of the density moving outwards from each cell to each of it's four neighbors, and density flowing in to it from it's neighbors. Figure 4 aids in visualizing the density exchange between the neighboring cells.

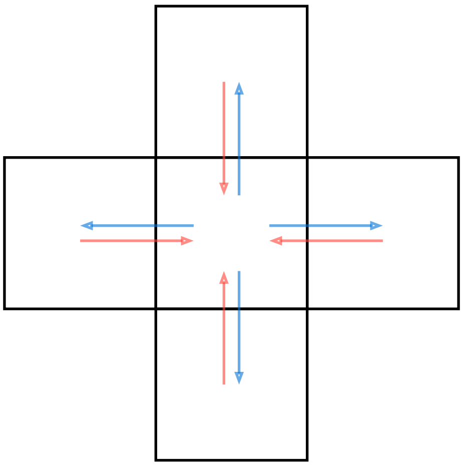

Figure 3: Five point stencil intuition

In numerical methods for partial differential equations theory this method is often called a five point stencil finite difference method.

In a more mathematical sense, what this method does is it approximates the Laplacian Δu with two second order finite difference schemes, formally

$$\left(\Delta_h \rho_h\right)_{i,j} = \frac{\rho_{i+1,j} + \rho_{i-1,j} + \rho_{i,j+1} + \rho_{i,j-1} - 4\rho_{i,j}}{h^2},$$

where h is the mesh fineness, that is $h = \frac{1}{N}$.

The above equations for i and j indices define a linear system of equations $A_h \rho_h = f_h$, where A_h is the discretization of the Laplacian and f_h is the right hand side of the original Poisson equation restricted on the h-fine grid.

To solve a linear system of equations one can solve it exactly with various approaches, however, this will not suffice for us, as all the exact methods are too slow for our needs. One can prove sufficient properties of the discretization matrix that imply that an iterative method, such as Gauss–Seidel will converge rapidly to the exact solution, saving us precious time at the cost of exactness. We present an illustration of the A_h discretization matrix in Figure 4

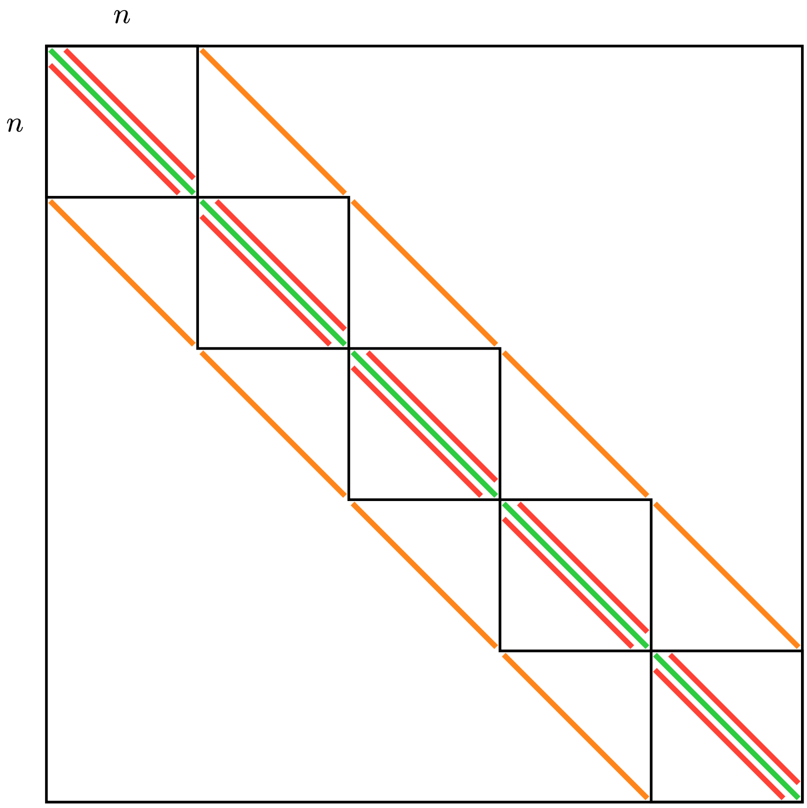

Figure 4: Five point stencil matrix, where green $= 4/h^2$, red $= -1/h^2$, orange $= -1/h^2$

Let us denote the Kronecker product of two matrices as $A \otimes B$ and let $B = \text{tridiag}(-1, 2, -1)$. Then, the above discretization matrix of the five point stencil an be achieved with the following succinct formula: $I \otimes B + B \otimes I$.

However, this matrix is so sparse, that we need not even construct it, as we can just solve the resulting linear system of equations with an iterative method without constructing the full matrix.

For a more extensive treatment of the subject, the reader is advised to consult section 2.2 of [2].

5.3 Advection

Make it clear that this is the first crucial idea of the method. The idea is that instead of the solving the advection equation as a partial differential equation, we trace a path backwards to where a fluid particle could have come from.

The problem with solving the advection equation is that it is dependent on the velocity vector, in contrast to the diffusion step where it was only dependent on the previous state of the density field.

After tracing back the possible locations where fluid particles could have come from we might get a particle that came from not the exact center of a grid. Remember, that we established that we shall think of the fluid as point masses centered at the middle of the grid cells. If a particle came from not the dead center then we must somehow give meaning to it too. In this case we will take the linear interpolation of the four closes neighbors of where the particle came from.

Fluid simulations methods that solve a partial differential equation on a discretized space are called Lagrangian, whereas methods that simulate fluids as a collection of particles are called Eulerian. For this reason this method is sometimes called semi-Lagrangian.

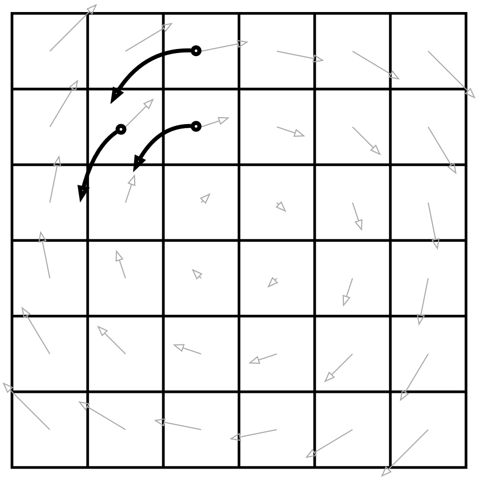

Figure 5: Tracing back the particle path along the velocity field

6 Evolving velocities

Recall, that the velocity equation is almost the same as the density equation.

This is where the second novel idea comes into play. Remember in the second section we mentioned that we shall omit a part of the equation to make it simpler and handle it later, now is the time to do so. The part we left out made sure that the velocity field was divergence free, meaning that it was mass conserving. This is intuitive about fluids, that a fluid can not just fluid outward from a single point, if some fluid flows out from a point, then an equal amount must flow into said point.

Since we did not take care to hold the divergence free property during the diffusion and advection steps we quite possible end up with a velocity field which has non zero divergence. To combat this we rely on a result from vector calculus which states that a vector field can be decomposed as a sum of a field with no divergence and one which is the gradient of a scalar potential. This result is called the Helmholtz–Hodge decomposition.

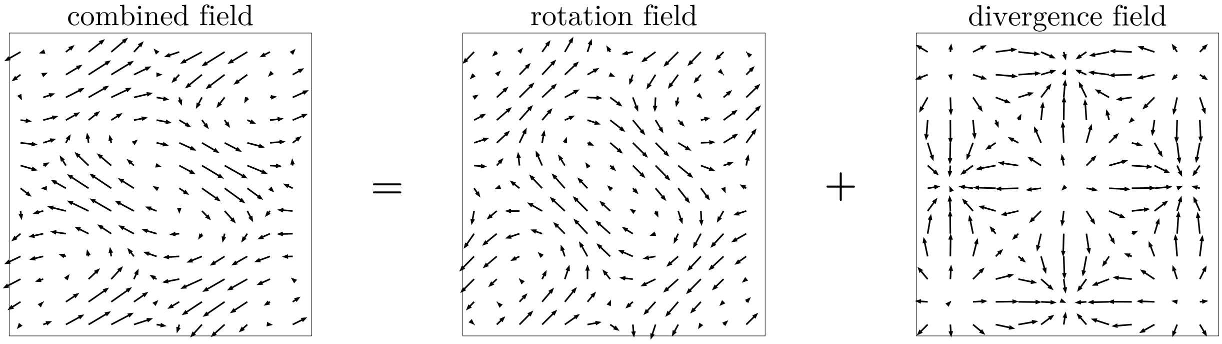

Figure 6: Helmholtz–Hodge decomposition

The Helmholtz–Hodge decomposition states that any vector field $\boldsymbol{w}$ can be uniquely decomposed into the sum of a divergence field and a rotation field, more concisely

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

where $\nabla \cdot (\boldsymbol{u}) = 0$, and q is a scalar field. Figure 6 illustrates the idea of the decomposition.

Formally taking the dot product with the ∇ operator of both sides, we get

$$\begin{aligned}\nabla \cdot \boldsymbol{w} &= \nabla \cdot \boldsymbol{u} + \nabla \cdot \nabla q \\ \nabla \cdot \boldsymbol{w} &= 0 + \nabla \cdot \nabla q \\ \nabla \cdot \boldsymbol{w} &= \Delta q\end{aligned}$$

The relation between $\boldsymbol{w}$ and q we just derived is a simple Poisson equation for q, which can be solved with the finite difference method we outlined in the previous section. After solving for q, we can extract $\boldsymbol{u}$ as

$$\boldsymbol{u} = \boldsymbol{w} - \nabla q.$$

With this result in our hands we can finally resolve the mass conserving property of the velocity of the simulated fluid by decomposing the resulting field after the last step into a divergence free field.

One can then imagine the simulation steps as follows:

$$u_1 \xrightarrow{\text{add source}} u_2 \xrightarrow{\text{diffusion}} u_3 \xrightarrow{\text{advection}} u_4 \xrightarrow{\text{projection}} u_5$$

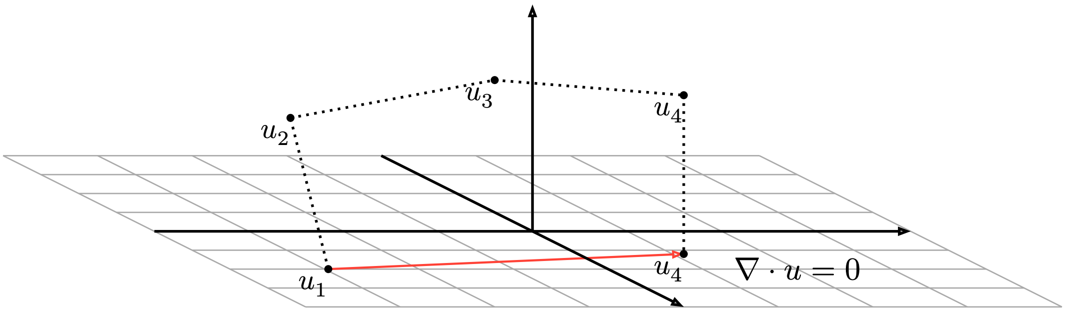

Figure 7: Illustrative steps of the simulation

7 Appendix

All the code can be found at https://github.com/leonardo-toffalini/viscous

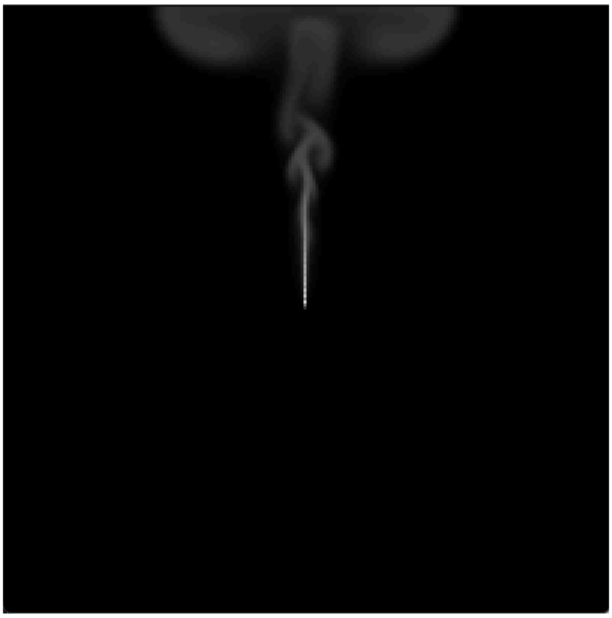

Figure 8: Screenshot from the fluid simulation of a predefined scene that aims to represent smoke emitting from the tip of a cigarette.

Bibliography

[1] A. J. Chorin, J. E. Marsden, and J. E. Marsden, A mathematical introduction to fluid mechanics, vol. 168. Springer, 1990.

[2] R. H. János Karátson, “Numerical Methods for Elliptic Partial Differential Equations.” [Online]. Available: https://kajkaat.web.elte.hu/pdnmell-ang-2023-I.pdf