Stable Fluids

by

Lukas Polthier, Johannes von Lindheim

based on Stam, 1999

Scientific Visualization, winter 15/16

Supervisor: Prof. Dr. Konrad Polthier

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Berlin, den 9. April 2016

Lukas Polthier
Johannes von Lindheim
Institut für Mathematik
Arnimallee 6
Freie Universität Berlin
14195 Berlin
`info@lukas-polthier.de`
`jovoli@gmx.de`

Table of Contents

0 Abstract

1 Introduction

2 Mathematical Modelling

2.1 Basic Equations

Let $\Omega \subset \mathbb{R}^2$ be the domain of interest, e.g. $\Omega = (0,1)^2$. Now let $u \in C^2(\mathbb{R} \times \Omega, \mathbb{R}^2)$ denote the velocity vector field and $p \in C^2(\mathbb{R} \times \Omega, \mathbb{R})$ denote the pressure field. Both fields depend on the time $t \in \mathbb{R}$ and the position in space $x \in \Omega$. The evolution of these fields is given by the Navier-Stokes equations

$$\text{div } u = 0 \tag{2.1}$$

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{1}{\rho}\nabla p + \nu \Delta u + f, \tag{2.2}$$

where ν, ρ are constants that determine the viscosity of the fluid and the density respectively. In $\mathbb{R}^2$, equation 2.2 can be written out as

$$\begin{pmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial u_2}{\partial t} \end{pmatrix} = -\begin{pmatrix} u_1 \frac{\partial u_1}{\partial x_1} + u_2 \frac{\partial u_1}{\partial x_2} \\ u_1 \frac{\partial u_2}{\partial x_1} + u_2 \frac{\partial u_2}{\partial x_2} \end{pmatrix} - \frac{1}{\rho}\begin{pmatrix} \frac{\partial p}{\partial x_1} \\ \frac{\partial p}{\partial x_2} \end{pmatrix} + \nu \begin{pmatrix} \frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_1}{\partial x_2^2} \\ \frac{\partial^2 u_2}{\partial x_1^2} + \frac{\partial^2 u_2}{\partial x_2^2} \end{pmatrix} + \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}.$$

The pressure and velocity field that appears in the Navier-Stokes equations, are related. By combining equation 2.1 and 2.2 we obtain a singe equation as follows.

By the Helmholtz-Hodge decomposition theorem we have that any vector field $w = u + \nabla q + \text{res}$ uniquely decomposes into a divergence-free part u, a gradient field ∇q and a residual term depending on the genus of the surface. In our case, the residual term vanishes.

Let $P : C^2(\Omega, \mathbb{R}^2) \to \{f \in C^2(\Omega, \mathbb{R}^2), \text{div} f = 0\}$ denote the projection operator onto the divergence free part. Obviously, the operator P is implicitly defined by

$$\text{div } w = \Delta q. \tag{2.3}$$

With Neumann boundary condition ($\frac{\partial q}{\partial n} = 0$ on $\partial\Omega$, n is the outward normal), equation 2.3 is a Poisson equation. Let q denote the solution, then P is defined by $Pw = w - \nabla q$. If we now apply P to both sides of 2.2, the Navier-Stokes equation compress into our fundamental equation 2.4.

$$\frac{\partial u}{\partial t} = P\left(-(u \cdot \nabla)u - \frac{1}{\rho}\nabla p + \nu \Delta u + f\right) \tag{2.4}$$

2.2 Method of Solution and Discretization

Equation 2.4 consists of four parts, the **add force** term f, the **advection** term $(u \cdot \nabla)u$, the **diffusion** term $\nu \Delta u$ and the **projection** operator P. The equation is solved from an initial state $u^0 = u(0, x)$. Both time and space are discretized with time step ΔT and some equidistant grid points of distance $h = \frac{1}{n}$. Each of the four terms in equation 2.4 is applied successively to the initial state $u^0 \in C^2(\Omega, \mathbb{R})$. The general procedure is

$$u^0 \xrightarrow{\text{add force}} u^1 \xrightarrow{\text{advect}} u^2 \xrightarrow{\text{diffuse}} u^3 \xrightarrow{\text{project}} u^4$$

The solution at time $t + \Delta t$ is then given by $u(x, t + \Delta t) = u^4(x)$.

2.2.1 Add force

The add force step incorporates additional force by the user, or buoyancy force due to uplift of lighter gases and downlift of heavier gases.

$$u^1(x) = u^0(x) + \Delta t \; f(t, x)$$

The buoyancy force is computed using Archimedes' principle. In a simplified approach, heaviness is equal to the density of the smoke. After computing the average temperature of the fluid, the upward force is determined for each pixel separately depending on the difference with respect to the average temperature.

2.2.2 Advect

The advect step accounts for the advection or convection of the fluid itself, i.e. this step lets the fluid „flow" a little. The advection step is fundamental to this particular fluid solver. Thats why the solver is called „Stable"Fluids, as this solver will never blow up, independent of the size of the time step Δt.
The method can be understood intuitively: All particles in the fluid are moved by the velocity of the fluid itself. To obtain the velocity at the point x at time $t+\Delta t$ we backtrace the the point x through the velocity field at time t. This defines a path $p : (-\delta, \delta) \times \Omega \to \Omega$ corresponding to a streamline of the fluid. The velocity $u^2(x)$ is the set to be the velocity of $u^1(p(-\Delta t, x))$ at the previous time step:

$$u^2(x) = u^1(p(-\Delta t, x)).$$

[tba] include a figure that illustrates the approach. [tba] [tba] include a comparison to other solvers of the advect step. [tba]

2.2.3 Diffuse

This step solves the diffusion of the fluid itself, i.e. the „friction" between parts of the fluid with different velocity. This effect is equivalent to the diffusion equation

$$\frac{\partial u^2}{\partial t} = \nu \Delta u^2. \tag{2.5}$$

The most straightforward way would be to discretize the Laplacian and solve the resulting sparse linear system. However, this approach is unstable when the viscosity is large. For our implicit approach we proceed as follows, by approximating $\frac{\partial u}{\partial t}$ with the backward difference quotient:

$$\frac{u(t+\Delta t,x)-u(x,t)}{\Delta t} = \nu\Delta u(t-\Delta t,x)$$

finally, this gives

$$(I-\nu\Delta t)u^3(x) = u^2(x). \tag{2.6}$$

We now discretize 2.6 using a finite difference method and obtain

$$\left(\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} - \begin{bmatrix} -4 & 1 & & 1 & & \\ 1 & \ddots & \ddots & & \ddots & \\ & \ddots & \ddots & \ddots & & 1 \\ 1 & & \ddots & \ddots & \ddots & \\ & \ddots & & \ddots & \ddots & 1 \\ & & 1 & & 1 & -4 \end{bmatrix} \nu\frac{\Delta t}{h^2}\right)u^3 = u^2. \tag{2.7}$$

The resulting square matrix has $n*m$ rows and columns, where n,m denote the number of pixels in the x- and y-axis respectively. Solving such a system can be done efficiently by iterative schemes, e.g. Gauß-Seidel.

2.2.4 Project

The last step makes the vectorfield mass preserving, i.e. divergence free. We already discussed in the derivation of our fundamental equation that the projection is obtained by solving

$$\text{div } u = \Delta q.$$

When discretized, this equation becomes

$$\frac{1}{2h}\begin{bmatrix} 0 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix} u_1^3 + \frac{1}{2h}\begin{bmatrix} 0 & & 1 & \\ & \ddots & & \ddots \\ & & \ddots & 1 \\ 1 & & & \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix} u_2^3 = \begin{bmatrix} -4 & 1 & & 1 & & \\ 1 & \ddots & \ddots & & \ddots & \\ & \ddots & \ddots & \ddots & & 1 \\ 1 & & \ddots & \ddots & \ddots & \\ & \ddots & & \ddots & \ddots & 1 \\ & & 1 & & 1 & -4 \end{bmatrix} q. \tag{2.8}$$

As in equation 2.7 we need to solve a sparse linear system.

2.3 Moving substances through the Fluid

Our solver enables us to compute the ambient fluid. However, we need to visualize the vector field. A substance, that is injected in the fluid and does not interact with it, will be advected by the vector field and diffuse at the same time. Let $d \in C^2(\mathbb{R}^+ \times \Omega, \mathbb{R})$ denote the density field of such a substance. The evolution of this scalar field is given by

$$\frac{\partial d}{\partial t} = -u \cdot \nabla a + \kappa \Delta - \alpha a + S$$

where κ is the diffusion constant, α is the dissipation term and S is a source term. The dissipation term, which will be dropped in our modelling, describes the effect that kinetic energy is converted in thermal energy.

2.4 Vorticity Confinement

$$\begin{pmatrix} g & f & f \\ g & f & f \\ g & f & f \end{pmatrix}$$

3 Implementation

4 Results

5 Extensions

6 Outlook

Reference List

[1] H. Alt: “Lineare Funktionalanalysis”, fifth edition, Springer, 2006.

[2] T. Apel: “Variationsrechnung”, script, Institut für Mathematik und Bauinformatik Fakultät für Bauingenieurwesen und Umweltwissenschaften, Universität der Bundeswehr München, 2013.

[3] C. Carathéodory: “Gesammelte Mathematische Schriften”, second volume, C.H. Beck’sche Verlagsbuchhandlung München, 1955.

[4] R. Coleman: “A Detailed Dnalysis of the Brachistochrone Problem”, <hal-00446767v2>, 2012. https://hal.archives-ouvertes.fr/hal-00446767v2

[5] E. A. Coddington, N. Levinson: “Theory of Ordinary Differential Equations”, first edition, McGraw-Hill, 1955.

[6] L. Euler: “Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrici latissimio sensu accepti”, Lausanne, Geneva, 1774.

[7] W. Forst, D. Hoffmann: “Gewöhnliche Differentialgleichungen: Theorie und Praxis”, second edition, Springer Spektrum, 2013.

[8] O. Forster: “Analysis 2”, eighth edition, Vieweg+Teubner, 2008.

[9] O. Forster: “Analysis 3”, seventh edition, Springer Spektrum, 2012.

[10] P. Funk: “Variationsrechnung und ihre Anwendung in der Technik”, second edition, Springer, 1970.

[11] I. M. Gelfand, S. V. Fomin: “Calculus of Variations”, Rev. Engl. ed. / Transl. and ed. by Richard A. Silverman, Englewood Cliffs, N.J. : Prentice-Hall, 1963.

[12] W. G. Kelley, A. C. Peterson: “ The Theory of Differential Equations”, second edition, Springer, 2010.

[13] R. Kornhuber, C. Schütte: “Computerorientierte Mathematik”, script, Freie Universität Berlin, 2011.

[14] P. V. Negrón-Marrero, B. L. Santiago-Figueroa: “The Nonlinear Brachistochrone Problem with Friction”, University of Puerto Rico, 2005. http://mate.uprh.edu/~urmaa/reports/brach.pdf

[15] H. J. Oberle: “Variationsrechnung u. Optimale Steuerung”, http://www.math.uni-hamburg.de/home/oberle/skripte/varopt.html, 24.09.2015.

[16] C. Rousseau, Y. Saint-Aubin: “Calculus of Variations and Applications”, Springer Science+Business Media, LLC 2008. https://math.berkeley.edu/~strain/170.S13/cov.pdf

[17] A. Sasane: “Calculus of Variations and Optimal Control”, University of Virginia, U.S.A, 2004. https://math.berkeley.edu/~strain/170.S13/cov.optimal.control.pdf

[18] J. Stoer: “Numerische Mathematik 1”, ninth edition, Springer, 2005.

[19] J. Stoer, R. Bulirsch: “Numerische Mathematik 2”, fifth edition, Springer, 2005.

[20] W. Walter: “Ordinary Differential Equations”, Transl. by Russel Thompson, Springer 1998.