SECOND EDITION

Fluid Simulation for Computer Graphics

Robert Bridson

CRC Press
Taylor & Francis Group

AN A K PETERS BOOK

SECOND EDITION

Fluid Simulation for Computer Graphics

Robert Bridson

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN A K PETERS BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed at CPI on sustainably sourced paper.
Version Date: 20150810

International Standard Book Number-13: 978-1-4822-3283-7 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

SECOND EDITION

Fluid Simulation for Computer Graphics

[For my wife, Rowena.]

Preface

Seven years have now passed since I wrote the first edition of this book. My aim is still not to provide a full survey of the field, but instead a practical introduction to writing fluid solvers. I have tried to distill my knowledge of the research field and my experience in the visual effects industry to hit on what I think now are the most important points, giving enough motivation that hopefully it is clear how and why things work. I hope nobody will be upset if I missed their research: I make no claim to properly overview the field, but am just carving out a path I believe is useful.

Compared to the first edition there is plenty of new material, for example new chapters on level sets and vortex methods. The ordering of topics has changed to make more sense when read the first time through, and I have upgraded several parts according to my experience. I still assume the reader has no background in fluid dynamics, and not much in the way of numerical methods, but a comfort with vector calculus, ordinary differential equations, and the standard graphics mix of linear algebra and geometry is necessary.

Previously I thanked Ron Fedkiw, who introduced me to graphics and fluids; my coauthors and students (many more now!); Marcus Nordenstam with whom I wrote several important fluid solvers including Naiad and now Bifrost; Jim Hourihan, Matthia Müller-Fischer, Eran Guendelman, and Alice Peters (of A K Peters) who all helped in the process of turning ideas and enthusiasm into the first edition. To these I would also add Wei-Pai Tang who got me started in numerical methods; the University of British Columbia Computer Science Department; Michael Nielsen; and the staff at Taylor and Francis who have made this second edition possible. Above all I would like to thank my family, especially my wife, for supporting me through the late nights, stress, one-sided conversations, and all the rest that accompany writing a book.

Robert Bridson
April 2015

Contents

Part I

The Basics

1

The Equations of Fluids

Fluids surround us, from the air we breathe to the oceans covering two thirds of the Earth, and are at the heart of some of the most beautiful and impressive phenomena we know. From water splashing, to fire and smoke swirling, fluids have become an important part of computer graphics. This book aims to cover the basics of simulating these effects for animation. So let's jump right in with the fundamental equations governing their motion.

Most fluid flow of interest in animation is governed by the famous *incompressible Navier-Stokes equations*, a set of partial differential equations that are supposed to hold throughout the fluid. The equations are usually written as

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \nu \nabla \cdot \nabla \vec{u}, \tag{1.1}$$

$$\nabla \cdot \vec{u} = 0. \tag{1.2}$$

These may appear pretty complicated at first glance! We'll soon break them down into easy-to-understand parts (and in Appendix B provide a more rigorous explanation), but first let's begin by defining what each symbol means.

1.1 Symbols

The letter $\vec{u}$ is traditionally used in fluid mechanics for the velocity of the fluid. Why not $\vec{v}$? It's hard to say, but it fits another useful convention to call the three components of 3D velocity (u, v, w), just as the three components of position $\vec{x}$ are often taken to be (x, y, z).

The Greek letter ρ stands for the density of the fluid. For water, this is roughly 1000 kg/m^3, and for air in usual sea-level conditions this is roughly 1.3 kg/m^3, a ratio of about $700 : 1$.

It's worth emphasizing right away my insistence on using real units (meters, kilograms, etc.): long experience has shown me that it is well worth keeping all quantities in a solver implicitly in SI units, rather than just set to arbitrary values. It is tempting when starting to program a new solver to just fill in unit-less values like 1 for physical quantities such as density or to drop them altogether from expressions, whether you're operating from a quick-and-dirty just-make-it-work point of view or a more mathematically founded non-dimensionalization rationale[1]. However, this often comes back to haunt you when simulations don't quite look right, or need to be resized, or adjusted in other ways where it's not clear which of a plethora of nonphysical parameters need to be tweaked. We'll discuss the ramifications of this in algorithm design as well throughout the book.

The letter p stands for **pressure**, the force per unit area that the fluid exerts on anything.

The letter $\vec{g}$ is the familiar acceleration due to gravity, usually $(0, -9.81, 0)$ m/s^2. Now is a good time to mention that in this book we'll take as a convention that the y-axis is pointing vertically upwards, and the x- and z-axes are horizontal. We should add that in animation, additional control accelerations (to make the fluid behave in some desired way) might be added on top of gravity — we'll lump all of these into the one symbol $\vec{g}$. More generally, people call these **body forces**, because they are applied throughout the whole body of fluid, not just on the surfaces.

The Greek letter ν is technically called the **kinematic viscosity**. It measures how viscous the fluid is. Fluids like molasses have high viscosity, and fluids like mercury have low viscosity: it measures how much the fluid resists deforming while it flows (or more intuitively, how difficult it is to stir).

1.2 The Momentum Equation

The first differential equation (1.1), which is actually three in one wrapped up as a vector equation, is called the **momentum equation**. This really is good old Newton's equation $\vec{F} = m\vec{a}$ in disguise. It tells us how the fluid accelerates due to the forces acting on it. We'll try to break this down before moving onto the second differential equation (1.2), which is called the **incompressibility condition**.

Let's first imagine we are simulating a fluid using a particle system (later in the book we will actually use this as a practical method, but for now let's just use it as a thought experiment). Each particle might represent a little blob of fluid. It would have a mass m, a volume V, and a velocity $\vec{u}$. To integrate the system forward in time all we need is to

[1]Non-dimensionalization is a strategy for mathematically simplifying physical equations by rewriting all quantities as ratios to characteristic values of the problem at the hand, like the usual density of the fluid and the width of the container, rather than using SI units. This can reduce the number of constants that appear in the equations to the minimal set that matter, making some analysis much easier.

figure out what the forces acting on each particle are: $\vec{F} = m\vec{a}$ then tells us how the particle accelerates, from which we get its motion. We'll write the acceleration of the particle in slightly odd notation (which we'll later relate to the momentum equation above):

$$\vec{a} \equiv \frac{D\vec{u}}{Dt}.$$

The big D derivative notation is called the **material derivative** (more on this later). Newton's law is now

$$m\frac{D\vec{u}}{Dt} = \vec{F}.$$

So what are the forces acting on the particle? The simplest is of course gravity: $m\vec{g}$. However, it gets interesting when we consider how the rest of the fluid also exerts force: how the particle interacts with other particles nearby.

The first of the fluid forces is pressure. High-pressure regions push on lower-pressure regions. Note that what we really care about is the net force on the particle: for example, if the pressure is equal in every direction there's going to be a net force of zero and no acceleration due to pressure. We only see an effect on the fluid particle when there is an imbalance, i.e. higher pressure on one side of the particle than on the other side, resulting in a force pointing away from the high pressure and toward the low pressure. In the appendices we show how to rigorously derive this, but for now let's just point out that the simplest way to measure the imbalance in pressure at the position of the particle is simply to take the negative gradient of pressure: $-\nabla p$. (Recall from calculus that the gradient is in the direction of "steepest ascent," thus the negative gradient points away from high-pressure regions toward low-pressure regions.) We'll need to integrate this over the volume of our blob of fluid to get the pressure force. As a simple approximation, we'll just multiply by the volume V. You might be asking yourself, but what is the pressure? We'll skip over this until later, when we talk about incompressibility, but for now you can think of it being whatever it takes to keep the fluid at constant volume.

The other fluid force is due to viscosity. A viscous fluid tries to resist deforming. Later we will derive this in more depth, but for now let's intuitively develop this as a force that tries to make our particle move at the average velocity of the nearby particles, i.e., that tries to minimize differences in velocity between nearby bits of fluid. You may remember from image processing, digital geometry processing, the physics of diffusion or heat dissipation, or many other domains, that the differential operator that measures how far a quantity is from the average around it is the

Laplacian $\nabla \cdot \nabla$. (Now is a good time to mention that there is a quick review of vector calculus in the appendices, including differential operators like the Laplacian.) This will provide our viscous force then, once we've integrated it over the volume of the blob. We'll use the **dynamic viscosity coefficient**, which is denoted with the Greek letter μ (dynamic means we're getting a *force* out of it; the kinematic viscosity from before is used to get an *acceleration* instead). I'll note here that near the surface of a liquid (where there isn't a complete neighborhood around the blob) and for fluids with variable viscosity, this term ends up being a little more complicated; see Chapter 10 for more details.

Putting it all together, here's how a blob of fluid moves:

$$m\frac{D\vec{u}}{Dt} = m\vec{g} - V\nabla p + V\mu\nabla \cdot \nabla\vec{u}.$$

Obviously we're making errors when we approximate a fluid with a small finite number of particles. We will take the limit then as our number of particles goes to infinity and the size of each blob goes to zero. Of course, this clearly makes a different sort of error, as real fluids are in fact composed of a (very large) finite number of molecules! But this limit, which we call the **continuum model**, has the advantages of mathematical conciseness and independence from the exact number of blobs, and has been shown experimentally to be in extraordinarily close agreement with reality in a vast range of scenarios. However, taking the continuum limit does pose a problem in our particle equation, because the mass m and volume V of the particle must then go to zero, and we are left with nothing meaningful. We can fix this by first dividing the equation by the volume, and then taking the limit. Remembering m/V is just the fluid density ρ, we get

$$\rho\frac{D\vec{u}}{Dt} = \rho\vec{g} - \nabla p + \mu\nabla \cdot \nabla\vec{u}.$$

Looking familiar? We'll divide by the density and rearrange the terms a bit to get

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g} + \frac{\mu}{\rho}\nabla \cdot \nabla\vec{u}.$$

To simplify things even a little more, we'll define the kinematic viscosity as $\nu = \mu/\rho$ to get

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g} + \nu\nabla \cdot \nabla\vec{u}.$$

We've almost made it back to the momentum equation! In fact this form, using the material derivative D/Dt, is actually more important to us in computer graphics and will guide us in solving the equation numerically. But we still will want to understand what the material derivative is and how it relates back to the traditional form of the momentum equation. For that, we'll need to understand the difference between the **Lagrangian** and **Eulerian** viewpoints.

1.3 Lagrangian and Eulerian Viewpoints

When we think about a continuum (like a fluid or a deformable solid) moving, there are two approaches to tracking this motion: the Lagrangian viewpoint and the Eulerian viewpoint.

The Lagrangian approach, named after the French mathematician Lagrange, is what you're probably most familiar with. It treats the continuum just like a particle system. Each point in the fluid or solid is labeled as a separate particle, with a position $\vec{x}$ and a velocity $\vec{u}$. You could even think of each particle as being one molecule of the fluid. Nothing too special here! Solids are almost always simulated in a Lagrangian way, with a discrete set of particles usually connected up in a mesh.

The Eulerian approach, named after the Swiss mathematician Euler, takes a different tactic that's usually used for fluids. Instead of tracking each particle, we instead look at fixed points in space and see how measurements of fluid quantities, such as density, velocity, temperature, etc., at those points change in time. The fluid is probably flowing past those points, contributing one sort of change: for example, as a warm fluid moves past followed by a cold fluid, the temperature at the fixed point in space will decrease—even though the temperature of any individual particle in the fluid is not changing! In addition the fluid variables can be changing in the fluid, contributing the other sort of change that might be measured at a fixed point: for example, the temperature measured at a fixed point in space will decrease as the fluid everywhere cools off.

One way to think of the two viewpoints is in doing a weather report. In the Lagrangian viewpoint you're in a balloon floating along with the wind, measuring the pressure and temperature and humidity, etc., of the air that's flowing alongside you. In the Eulerian viewpoint you're stuck on the ground, measuring the pressure and temperature and humidity, etc., of the air that's flowing past. Both measurements can create a graph of how conditions are changing, but the graphs can be completely different as they are measuring the rate of change in fundamentally different ways.

Numerically, the Lagrangian viewpoint corresponds to a particle system, with or without a mesh connecting up the particles, and the Eulerian viewpoint corresponds to using a fixed grid that doesn't change in space even as the fluid flows through it.

It might seem the Eulerian approach is unnecessarily complicated: why not just stick with Lagrangian particle systems? Indeed, there are schemes, such as vortex methods (see, e.g., [YUM86, GLG95, AN05, PK05]) and smoothed particle hydrodynamics (SPH) (see, e.g., [DC96,MCG03,PTB$^+$03]) that do this. However, even these rely on the Eulerian-derived equations for forces in the fluid, and in this book we will largely stick with Eulerian methods for a few reasons:

- It's easier to analytically work with the spatial derivatives like the pressure gradient and viscosity term in the Eulerian viewpoint.
- It's much easier to numerically approximate those spatial derivatives on a fixed Eulerian mesh than on a cloud of arbitrarily moving particles.

The key to connecting the two viewpoints is the material derivative. We'll start with a Lagrangian description: there are particles with positions $\vec{x}$ and velocities $\vec{u}$. Let's look at a generic quantity we'll call q: each particle has a value for q. (Quantity q might be density, or velocity, or temperature, or many other things.) In particular, the function $q(t, \vec{x})$ tells us the value of q at time t for the particle that happens to be at position $\vec{x}$: this is an Eulerian variable since it's a function of space, not of particles. So how fast is q changing for the particle whose position is given by $\vec{x}(t)$ as a function of time, i.e., the Lagrangian question? Just take the total derivative (a.k.a. the Chain Rule):

$$\begin{aligned}\frac{d}{dt}q(t,\vec{x}(t)) &= \frac{\partial q}{\partial t} + \nabla q \cdot \frac{d\vec{x}}{dt}\\ &= \frac{\partial q}{\partial t} + \nabla q \cdot \vec{u}\\ &\equiv \frac{Dq}{Dt}.\end{aligned}$$

This is the material derivative!

Let's review the two terms that go into the material derivative. The first is $\partial q/\partial t$, which is just how fast q is changing at that fixed point in space, an Eulerian measurement. The second term, $\nabla q \cdot \vec{u}$, is correcting for how much of that change is due just to differences in the fluid flowing past (e.g., the temperature changing because hot air is being replaced by cold air, not because the temperature of any molecule is changing).

Just for completeness, let's write out the material derivative in full, with all the partial derivatives:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} + w\frac{\partial q}{\partial z}.$$

Obviously in 2D, we can just get rid of the w- and z-term.