

Control of Newtonian Fluids With Minimum Force Impact Using the Navier-Stokes Equations

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Patrick Fürst

Matrikelnummer 0927543

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Prof. Michael Wimmer, Károly Zsolnai MSc.

Wien, 3. Oktober 2014

Patrick Fürst

Prof. Michael Wimmer, Károly
Zsolnai MSc.

Control of Newtonian Fluids With Minimum Force Impact Using the Navier-Stokes Equations

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Patrick Fürst

Registration Number 0927543

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Prof. Michael Wimmer, Károly Zsolnai MSc.

Vienna, 3rd October, 2014

Patrick Fürst

Prof. Michael Wimmer, Károly
Zsolnai MSc.

Erklärung zur Verfassung der Arbeit

Patrick Fürst
Torggelgasse 18, 6971 Hard, Austria

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 3. Oktober 2014

Patrick Fürst

Acknowledgements

I would like to express my gratitude to my advisor Károly Zsolnai for the continuous support of my thesis, for his patience and motivation. I could not have imagined having a better advisor for my thesis. Also, I want to thank my family very much for supporting me in every way. Last but not least, I want to thank Teresa Sutter for proofreading my thesis and for the discussions we had about writing my thesis.

Kurzfassung

Diese Arbeit stellt eine neue Herangehensweise vor, um Rauch mit minimaler Krafteinwirkung zu kontrollieren, mit dem Ziel, natürlichere visuelle Resultate zu erhalten. Der erste Teil besteht aus einer Einführung zu Flüssigkeitssimulationen und Flüssigkeitskontrolle, gefolgt von einer Auflistung vorhergehender Forschungsarbeiten im Bereich numerische Strömungsmechanik, speziell im Bereich Computer Graphik. Danach werden die Navier-Stokes Gleichungen vorgestellt sowie eine kurze Übersicht, mit welchen Verfahren diese gelöst werden können, gegeben. Der letzte Teil beschreibt unseren Ansatz zur Kontrolle von Rauch mit „biased diffusion“ und „long-range force“ und die daraus entstandenen Resultate werden gezeigt.

Von der numerischen Lösung wurde ein Kriterium abgeleitet, mit welchem der Algorithmus entscheidet, ob Diffusion oder Kraft angewendet wird, um den Rauch zu verteilen. Die Resultate zeigen, dass der Rauch die Zieldichte früher erreicht und die Bewegung weniger heftig ist, was wiederum zu natürlicheren Ergebnissen beiträgt.

Unser Algorithmus wurde in der Open Source Animations-Software Blender implementiert und gibt dem Benutzer Zugang zu den Kontrollparametern.

Abstract

This thesis introduces a novel approach to control smoke towards a target density distribution with minimal force impact to reduce unnatural behavior and unconvincing visual results. The first part consists of an introduction to fluid simulations and fluid control followed by an exploration of previous research in the field of Computational fluid dynamics, especially in Computer Graphics. After that, the Navier-Stokes equations are introduced and a short overview on how to solve them is given. The last part describes our approach to controlling smoke based on biased diffusion and long-range force and shows the results of this research.

Based on a criterion, which has emerged from the numerical solution of diffusion, the algorithm decides whether to apply forces or use diffusion to distribute the smoke resulting in a great reduction of forces applied to the smoke. Results show that the smoke reaches the target density faster and the motion is much less furious, which contributes to more natural results.

Our algorithm is implemented in the open source animation software Blender and gives the artist access to smoke control parameters.

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
List of Figures	xiv
List of Tables	xiv
List of Algorithms	xv
1 Introduction	1
2 Related Work And Previous Work	3
3 The Navier-Stokes Equations - Theoretical Background	5
3.1 The Navier-Stokes Equations	5
3.2 Numerical Solution To the Navier-Stokes Equations	9
4 Smoke Control	13
4.1 Method For Smoke Control	13
4.2 Reducing Forces To A Minimum	16
5 Results	19
6 Conclusions and Discussion	23
Bibliography	25

List of Figures

1.1	Test rendering of the implemented algorithm with sphere as the target shape.	2
3.1	Relationship of shear rate and shear stress of different fluid types.	7
3.2	On the left side the Eulerian approach and on the right side the Lagrangian approach are shown.	9
4.1	"Top row: images from a real horse running sequence which serves as an evolving target density function; Bottom row: images from a controlled smoke simulation approximating both the shape and motion of the running horse" [SY02]	14
4.2	Left: more than half of the neighbours reached the target density in which case long-range force is applied to the current grid point. Right: a case where most of the neighbours did not reach the target density in which case the long-range force field is omitted as diffusion takes care of the density distribution.	15
5.1	From left to right. 1: Diffusion without boundaries. 2: Diffusion with boundaries. 3: Biased diffusion and long-range force. 4: Biased diffusion and reduced long-range force.	19
5.2	Top row: simulation with the algorithm based on Shi and Yu's approach [SY02], average calculation time per frame: 651ms ; Bottom row: simulation with reduced force approach, average calculation time per frame: 397ms.	20
5.3	Top row: Grid resolution 64^3 . Target shape: Stanford Bunny; Bottom row: Grid resolution 64^3 with Wavelet Turbulence, 4 divisions. Target shape: Stanford Bunny.	21
5.4	Grid resolution 64^3 with Wavelet Turbulence, 4 divisions. Target shape: Letter Y. .	21

List of Tables

5.1	Shows the different calculation time between our approach and the simplified version of Shi and Yu's approach using only the long-range force field. Note that the complete version of their algorithm takes 5-8 minutes per frame. The time was measured for the smoke control part of the algorithm and all values are average values per frame.	21
-----	--	----

List of Algorithms

4.1	Biased diffusion	17
-----	----------------------------	----

Introduction

Fluid simulations have become very popular in the motion picture industry. Nowadays, smoke and water are simulated and added to movies in post-processing steps rather than costly scene setups being made to capture the right shot. This is not only due to the fact that computing power got cheaper and faster, but physically correct algorithms were developed which present results where it is hardly possible to distinguish them from reality.

The Abyss [1989], directed by James Cameron, was one of the first movies which was a combination of real footage and rendered computer graphic images. The producers used a kind of fluid simulation to bring water into specific shapes. In Terminator 2 - Judgment Day [1991], also directed by James Cameron, the next step of fluid control was introduced. The Terminator T-1000 was made of a "mimetic poly-alloy" (liquid metal) structure which can transform into almost any shape it touches. What seemed to be impressive effects at the time, was not physically correct and took many hours to simulate, not to mention the development of such animations.

Since then, much research has been done in the field of fluid simulation and nowadays a lot of 3D animation software come with an integrated physically correct fluid simulation module. However, fluid simulations can still be very time consuming because of their huge system of equations which has to be solved. An extended overview on related work in the research field of fluid simulation as well as fluid control is provided in chapter 2. In chapter 3 the Navier-Stokes equations and how they can be solved efficiently is discussed.

In the case of fluid control, the main issue is that it is physically impossible to shape fluids. That means we can only achieve physically plausible results which have to be tested visually. According to the Navier-Stokes equations, external forces can be added in any number. This is used for example to add wind to a scene which would blow smoke in one direction. These forces have to be applied with much care because it can yield unnatural behavior very fast. While there are attempts to use these forces to shape fluids [SY02], in this thesis we present an approach to reduce the applied forces to a minimum and let the advection and diffusion help us reach a defined target shape (see Fig. 1.1). The overall concept of how the fluid control algorithm is designed is explained in more detail in chapter 4, which is followed by our approach on how to

reduce the forces to a minimum while still preserving the intended distribution. The differences between the old algorithm and ours as well as the results are presented in chapter 5.

Besides the motion picture industry, fluid simulations are widely used in the areas of engineering and physics, for example, to validate the design of new cars or airplanes by performing wind tunnel tests with a computer software or to forecast and prevent possible catastrophes such as flooding and tsunamis. However, fluid control is not of vital importance in those areas because of the fluid's unnatural behavior. It is more of an extension of the fluid simulation to help artists realise the visions and images they have in their minds.

In this paper smoke is used as a representative for simulations based on the Navier-Stokes equations although it is not limited to smoke but to common gas-like phenomena.

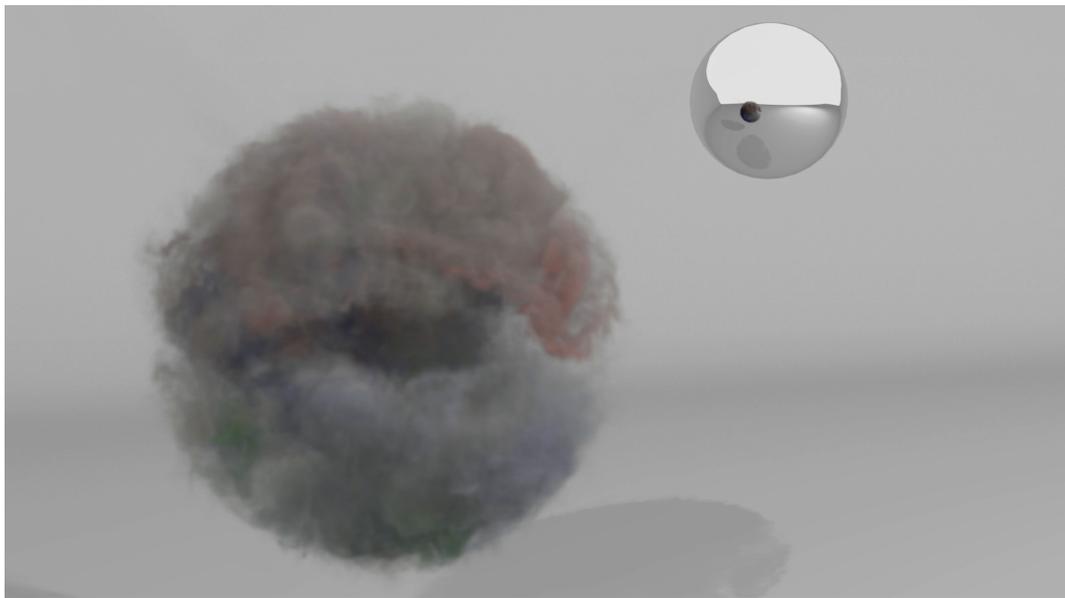


Figure 1.1: Test rendering of the implemented algorithm with sphere as the target shape.

Related Work And Previous Work

20 years ago modelling of smoke was done using procedural textures and the density was animated directly without modelling its velocity [EP90].

Later the equations of fluid dynamics were simulated directly resulting in a more natural way. Because of the lack of computing power back then, the simulations had to be solved on a relatively coarse grid resulting in bad resolutions and images.

Foster and Metaxas were the first to use the full Navier-Stokes equations in three dimensions to model both water [FM96] and gases [FM97], producing convincing fluid flows on relatively coarse grids. Solving the coupling of pressure and velocity, which is the key for realistic animation of fluids, was essential. Also, they decoupled the temporal resolution with the spatial resolution which results in better scaling and more freedom for the artist, because there is no need to understand the underlying system of equations in order to initiate it or change parameters to get the desired results [FM96]. Though one problem with their model is that it uses an explicit solver, which gets unstable for large time steps and limits the speed and interactivity.

The first unconditional stable algorithm "which still produces complex fluid-like flows" [Sta99] was introduced by Stam. It uses a combination of a semi-Lagrangian advection scheme and implicit solvers. This algorithm is easy to implement and allows the users to interact in real time with the simulation which is achieved by using much larger time steps. The down side is that this model suffers from too much numerical dissipation resulting in rapidly damping the flow. While it works rather well for graphical applications as the animator applies external forces to it and thus keeps it alive, it is not accurate enough for most engineering applications. For a detailed step by step explanation on how to implement this algorithm, see Real-Time Fluid Dynamics for Games [Sta03].

To reduce the numerical dissipation Fedkiw et al. [FSJ01] implemented a technique called vorticity confinement. As mentioned before, external forces can be applied by any number, so "the basic idea is to inject the energy lost due to numerical dissipation back into the fluid using a forcing term" [FSJ01], resulting in adding back the rotational and turbulent structure of the fluid field on the coarse grid.

Still, the resolution of the grid has to be high enough to produce visually acceptable results. If features of the fluid smoke are smaller than the coarse grid they get lost. So the grid must be refined in some way, which results in a linear increase in memory use and a greater than linear increase in the running time.

Kim et al. presented "a novel wavelet method for the simulation of fluids at high spatial resolution", which "enables large- and small-scale detail to be edited separately, allowing high-resolution detail to be added as a post-processing step" [KTJG08]. This approach allows artists and animators to simulate the Navier-Stokes equations on a low refined mesh and add the high frequency details later on as a post-processing step, without changing the overall behaviour of the fluid. The post-processing step only requires the velocity field of an existing fluid simulation as input and can be processed highly parallel. Though it does not "produce the results of explicit high-resolution simulations" [KTJG08], it uses much less memory and takes a lot less time.

Much less attention is given to smoke control. Initial methods only allowed users to define velocity values on a specific grid cell to control the flow of the smoke. This approach is not really suited for artists, since it requires a long process of trial and error to get the desired results and is far from smoke control or smoke shaping.

Treuille et al. [TMPS03] developed a control algorithm with which it is possible to shape smoke to a wanted target shape based on user-specified keyframes. These keyframes consist of a smoke density field and velocity field which are used to calculate specific external forces to reach the smokes density in each keyframe. This control algorithm presented a "significant advance over previous methods of controlling fluids through direct manipulation of simulation parameters" [TMPS03], although it was still very computationally intense and suffered from various drawbacks.

Inspired by the work of Treuille et al. [TMPS03], Fattal et al. [FL04] developed a so-called target-driven algorithm. The simulation is controlled by target objects which attract the smoke and drive it in the right direction. This algorithm can be implemented at a lower cost compared to a standard smoke simulation, since it does not approximate the target optimally and doesn't guarantee that the target density is ever reached before switching to the next one.

A similar algorithm was introduced by Shi et al. [SY02]. A target object is modeled as volumetric density function and again an external force field, including long-range and short-range force fields, is carefully designed to match the smoke density with the target density. Additionally, a nonuniform, biased diffusion equation is introduced to enforce high-resolution details in the target density.

The Navier-Stokes Equations - Theoretical Background

The following section gives an overview of the theoretical background of this thesis. First, the Navier-Stokes equations for incompressible, homogeneous Newtonian fluids is introduced and described in more detail. Next, we describe a solution to these equations proposed by Fedkiw et al. in Visual Simulation of Smoke [FSJ01] and the last part provides a short overview of the algorithm.

Following mathematical conventions, bold letters are used for vector types and non-bold letters are used for scalar types. All mathematical descriptions are valid either in 2D or 3D space.

3.1 The Navier-Stokes Equations

Named after Claude Louis Marie Henri Navier and George Gabriel Stokes, who were both physicists and mathematicians, the equations provide a mathematical model of motion of fluids and are of very importance in the field of fluid dynamics, hence they have been studied by many researchers.

The Navier-Stokes equations for an incompressible, homogeneous Newtonian fluid is described by a set of two equations. First, the momentum equation (3.1) and second, the equation of conservation of mass (3.2).

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \quad (3.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3.2)$$

3.1.1 The Momentum Equation

The first of the Navier-Stokes equations (3.1) is derived from Newton's second law of motion, the momentum equation (3.3).

$$\mathbf{F} = m\mathbf{a} \quad (3.3)$$

It states that the force \mathbf{F} of an object is equal to the mass m of the same object multiplied by the acceleration \mathbf{a} of that object.

In order to describe the force through density and velocity, the following substitutions are made: First, the mass is substituted for density, which is possible because we are operating in constant volume and mass is conserved. Second, the acceleration is described as the derivative of velocity in space and time. After applying the chain rule, we get a similar equation which describes the force through a density field and a velocity field (3.4).

$$\mathbf{F} = \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) \quad (3.4)$$

The left hand side of the previous equation (3.4) describes the forces acting on the fluid. There are many different forces acting on a fluid, some are always present such as the pressure p , some are added from the "outside", called external forces \mathbf{f} and some depend on the flow-properties of a fluid. These flow properties are described through a general term, the stress tensor $\nabla \cdot T$.

After substituting the general force term \mathbf{F} with the forces acting on a fluid, the result is the most general form of the first Navier-Stokes equation (3.5).

$$-\nabla p + \nabla \cdot T + \mathbf{f} = \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) \quad (3.5)$$

These single parts of the equation can be described as follows:

Pressure Term: $-\nabla p$

The pressure also known as the volumetric stress tensor is a normal stress, meaning it acts normal to the surface of the fluid. Regions of higher pressure push to regions with lower pressure and the direction can be determined by calculating the gradient field of the pressure, so in every point of the grid cell the direction of the pressure is known. The pressure change itself can be directly caused by external sources.

Stress Tensor As Viscous Stress:

For a Newtonian fluid the stress tensor describes the viscosity of the fluid and has to be stated more precisely, resulting in viscous stress.

Viscosity describes the resistance against deformation and different types of fluids have different viscosity. For example, honey has a much higher viscosity than water. Assuming it is a Newtonian fluid a more accurate term for the stress tensor can be achieved. Newtonian fluids obey Newton's law of viscosity, which states that there is a linear relation between shear rate and shear stress (not like in non-Newtonian fluids) [Bat00]. Thus the stress tensor can be described as

the velocity gradient multiplied by the viscosity μ , whereas the viscosity μ is a constant value depending on the fluid. Higher viscosity values counteracts more against fluid movements than lower values. Figure 3.1 shows different fluid types and their relationship of shear stress and shear rate.

$$\nabla \cdot T = \mu \nabla^2 \mathbf{u} \quad (3.6)$$

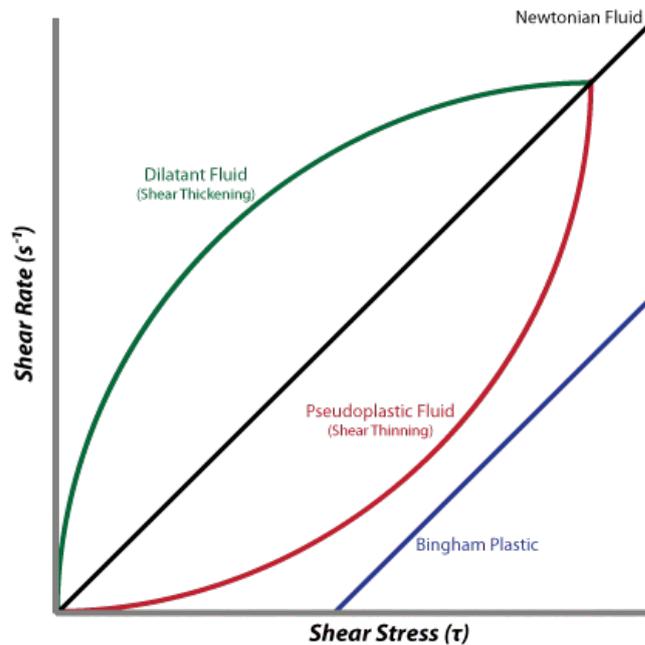


Figure 3.1: Relationship of shear rate and shear stress of different fluid types.

Image courtesy of Chucklingcanuck. [Chu14]

This part of the equation is also known as diffusion and can be described as exchange of density between neighbouring cells if there is a density difference. As a consequence, parts with higher density even out parts with lower density resulting in an equilibrium state.

Advection: $(\mathbf{u} \cdot \nabla)\mathbf{u}$

Advection describes the movement of quantities in the fluid and also the movement of the velocity field itself. This results in the mathematical formulation as directional derivative of the velocity field [Zso12]. This term makes the Navier-Stokes equations non-linear and thus difficult to solve.

3.1.2 Conservation Of Mass

Conservation of mass means that the total amount of mass in a system must remain constant over time if nothing is added or removed.

Derived from the general continuity equation, which describes the transport of conserved quantity in a volume, which in case of fluids is density ρ , we get the differential form of conservation of mass.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.7)$$

The first term describes the change rate of density over time and the second the change rate of density in space.

For incompressible fluids where the density within a volume is constant we can simplify it further. The derivative of the density becomes zero and by dividing through the constant density ρ we get:

$$\nabla \cdot \mathbf{u} = 0 \quad (3.8)$$

This is the incompressibility condition and means that the divergence of the velocity field is zero everywhere.

As of now, the first Navier-Stokes equation is derived from Newtons second law of motion and the incompressibility condition is introduced. These equations describe the motion of complex fluids, for which the next step is to find numerical solutions.

As we describe in the next part, it is not straight forward to find a numerical solution which is stable and can be used in every part of Computational Fluid Dynamics. Though in this field much research and mathematical investigation has been done, there are still unanswered questions. In fact, the existence of a "smooth solution" for every initial condition is not yet proven and is one of the greatest unsolved problems in physics. This is called the Navier-Stokes existence and smoothness problem.

The Clay Mathematics Institute states the Navier-Stokes equations as one of the seven Millennium Prize Problems and allocated \$1 million to the solution of each problem [Ins14]. The official description of the problem can be found on the website of the Clay Mathematics Institute ¹.

3.1.3 Simplified Navier-Stokes Equation

Before we describe the numerical solution we apply some simplifications to the first Navier-Stokes equation, which are specifically introduced to simulate gas-like effects like smoke or fire. Fedwik et al. [FSJ01] use a gas model which has the following properties:

- It is inviscid (zero viscosity),
- it is incompressible,
- it has constant density.

The viscosity can be ignored because "in gases especially on coarse grids" the "numerical dissipation dominates physical viscosity and molecular diffusion." [FSJ01]. With these assumptions we obtain the incompressible Euler equations (3.9) and (3.10) which look almost the same

¹<http://www.claymath.org/sites/default/files/navierstokes.pdf> [Accessed 01-June-2014]",

as the Navier-Stokes equations, except that the diffusion term is zero and the constant density is arbitrarily set to one. Later on, when introducing smoke control we reintroduce the diffusion term in a derived version called biased-diffusion.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \mathbf{f} \quad (3.9)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3.10)$$

3.2 Numerical Solution To the Navier-Stokes Equations

There are two different approaches to obtain a numerical solution of the Navier-Stokes equations, the Eulerian and the Lagrangian approach. In order to describe fluid movement we have to specify characteristic values such as velocity and density of certain points in the fluid. We can do that either by defining these values at fixed points in the volume of a fluid, the Eulerian method (see left side of Figure 3.2), or we can define small fluid material volumes like a particle system and track the characteristic values for every "particle", the Lagrangian approach (see right side of Figure 3.2).

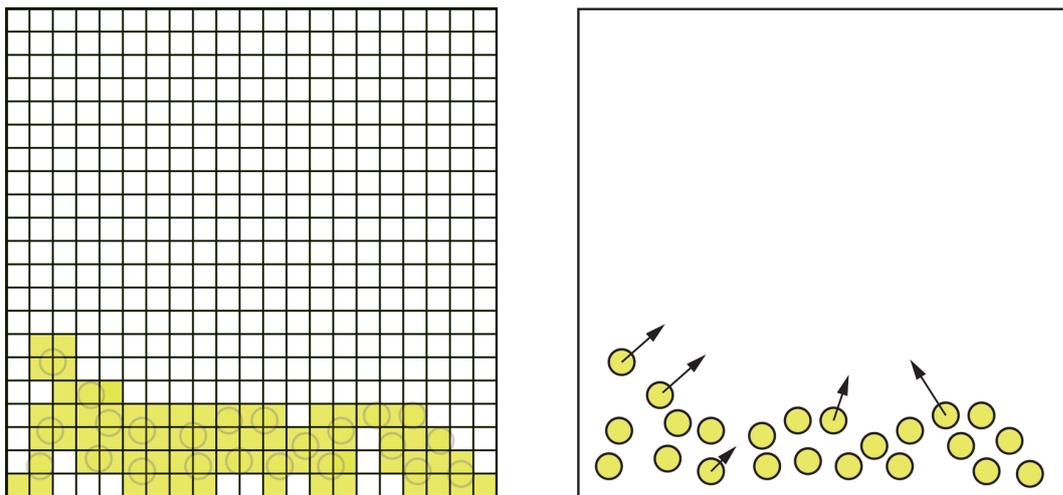


Figure 3.2: On the left side the Eulerian approach and on the right side the Lagrangian approach are shown.

Though both methods are widely used "almost all of the theory in fluid mechanics has been developed in the Eulerian system" [Pri06]. As Price et al. [Pri06] mentioned, this is due to the fact that fluid properties in fluid dynamics are required to be known at fixed positions rather than a moving material volume. In engineering, for example, one wants to know the temperature of a material at a specific point or the pressure at a valve.

These described methods are used to convert the characteristic values from continuous space in an equivalent discrete space which is called discretisation. This is done to bring differential equations in a computer understandable form and numerically solve these with approximations like finite difference, finite element or finite volume methods. As one can imagine when describing the fluid on grid points it is only an approximation of the real fluid and thus the quality highly depends on the resolution of the grid. So the higher the resolution the better the result, but also the higher the time to calculate each step. For example, if we want to double the resolution of a 3D volume without adding optimisation techniques the calculation time is eight times higher.

The aim in computer graphics is to generate visually pleasing results in a reasonable amount time, while in other areas of Computational Fluid Dynamics such as aviation it is important to calculate the most accurate solutions possible. On the one hand, some simplifications to the Navier-Stokes equations are applied in order to speed up calculations, on the other hand a bigger time step between each calculation step is used.

However, when using methods like finite difference the time steps have to be sufficiently small in order to get a stable simulation. As previously mentioned, Stam [Sta99] introduced a different solution using a semi-Lagrangian integration scheme followed by a pressure-Poisson equation resulting in an unconditional stable algorithm. Although this approach suffers from too much numerical dissipation, it still gives visually pleasing results. This leads us to the paper Visual simulation of smoke [FSJ01], also mentioned before, where a force is introduced to counteract this numerical dissipation. This is the basis for our implementation of smoke control. The basic algorithm for solving the Navier-Stokes equations is described in the following sections.

3.2.1 Helmholtz-Hodge Decomposition And the Final Equation To Solve

The fluids movement is described through velocity and pressure. Both vary in space and time and are related to each other. In this part, the Helmholtz-Hodge Decomposition is introduced and how equation (3.1) and equation (3.2) can be combined to obtain a single equation for the velocity.

"The key advantage of using the Helmholtz-Hodge Decomposition in this context is the decoupling of pressure and velocity fields. If the computation preserves orthogonality of the decomposition, any error in one of the terms is not reflected in the other. This procedure is more efficient than solving a coupled system of NSEs for velocity and pressure." [BNPB13]

The Helmholtz-Hodge Decomposition, named after Hermann von Helmholtz, states that any vector field \mathbf{w} can be decomposed in its divergence-free and curl-free components:

$$\mathbf{w} = \mathbf{v} + \nabla q \quad (3.11)$$

where \mathbf{v} is the divergence-free (incompressible) vector field satisfying $\nabla \cdot \mathbf{v} = 0$ and q is a scalar field [BNPB13].

In terms of the numerical solution of the Navier-Stokes equations, the velocity field has to conserve mass (incompressibility condition) (3.2) after each calculation step. This is achieved by first calculating an intermediate velocity field \mathbf{u}_{t+1} from \mathbf{u}_t , where t indicates time and the divergence-free velocity field \mathbf{u}_t is the result of the previous time step. Because \mathbf{u}_{t+1} is not divergence-free anymore, it has to be projected to its divergence-free component as a next step. The projection step with can be written as $\mathbf{v} = \mathbf{P}(\mathbf{w})$, where \mathbf{w} is any vector field, \mathbf{v} its

divergence-free component and \mathbf{P} the projection operator. Because \mathbf{P} satisfies $\mathbf{P}(\nabla q) = 0$ and \mathbf{v} is divergence-free ($\mathbf{P}(\mathbf{v}) = \mathbf{v}$), $\mathbf{v} = \mathbf{P}(\mathbf{w})$ is true.

"The operator is in fact defined implicitly by multiplying both sides of" the equation (3.11) with the del operator " ∇ " [Sta99]:

$$\nabla \cdot \mathbf{w} = \nabla^2 q \quad (3.12)$$

Remember $\nabla \cdot \mathbf{v} = 0$.

This is the previously bespoken Poisson equation of the scalar field q and leads to the calculation of the projected vector field \mathbf{v} :

$$\mathbf{v} = \mathbf{P}\mathbf{w} = \mathbf{w} - \nabla q \quad (3.13)$$

If this operator is applied on both sides of the momentum equation (3.1) the equation of the conservation of mass (3.2) (impressionability condition) is maintained.

$$\mathbf{P} \left(\underbrace{\left(\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\rho} \nabla p \right)}_{\mathbf{u} + \nabla q} \right) = \mathbf{P} \left(\underbrace{-(\mathbf{u} \cdot \nabla) \mathbf{u} + \mu \nabla^2 \mathbf{u} + \mathbf{f}}_{\mathbf{w}} \right) \quad (3.14)$$

With the above described fact this equation simplifies to:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{P} \left(-(\mathbf{u} \cdot \nabla) \mathbf{u} + \mu \nabla^2 \mathbf{u} + \mathbf{f} \right) \quad (3.15)$$

For the sake of completeness we applied the projection step to the more complex equations instead of the Euler equations (3.9) and (3.10).

This is the final form of the Navier-Stokes equation from which Stam [Sta99] developed a stable fluid solver. In the following section we roughly describe the steps of the algorithm but refer to the original papers [FSJ01], [Sta99] and [Sta03] for more detailed descriptions.

3.2.2 The Algorithm

Before the first step can be executed the initial state has to be defined. The simplest and, in most cases of Computer Graphics, sufficient one is an empty vector field for velocity and empty scalar fields for both density and pressure. The basic algorithm consists of 4 different steps.

1. Add forces to the velocity field.
This includes external forces from interactions with the fluid simulation, wind-like effects as well as vorticity confinement or buoyancy.
2. Calculate diffusion of both the velocity and the density field.

3. Project the velocity field to its divergence-free part.
This also includes solving the Poisson equation described in the previous part.
4. Calculate advection of both the velocity and the density field.

3.2.3 Sparse Systems of Linear Equations

All previously described steps, beside the first one, involve solving systems of linear equations. There are many numerical methods for this problem like the Jacobi method, the Gauss-Seidel method, the conjugate gradient and the multigrid method to name a few. All of these are not exact solvers and result only in an approximation of the real solution. Hence, all of these have different characteristics regarding calculation time and convergence. Choosing the right method depends again on which area of Computational Fluid Dynamics the simulation is applied to.

A detailed exploration of this topic would be out of scope of this thesis.

Smoke Control

As already mentioned, it is not possible to control the shape of smoke physically correctly. Hence, there is no mathematical plausible description of this problem which leads to results being assessed subjectively, for example, by comparing two different simulations visually and observing which one behaves more naturally. Although the Navier-Stokes equations allow us to add as many forces as we want to the simulation this results in unnatural behaviour and unconvincing visual results. As a consequence, the force field has to be designed with much care and it is desirable to add only as many forces as needed to steer the density in the desired direction.

In the following, two methods, long-range force and biased diffusion, are introduced and it is explained how they contribute to shape smoke. Additionally, we introduce altered methods on the basis of the previous arguments, which reduce the force to a minimum while still preserving the desired behavior. Again the goal is to simplify the work for artists and preserve the freedom to create any shape they can imagine.

4.1 Method For Smoke Control

The approach from Shi and Yu [SY02](Figure 4.1) involves the introduction of a new density field, the target density, an altered diffusion function called biased diffusion and a method to carefully create a force field.

The target density is "a volume density function for the target object" which is used to calculate a new force called long-range force "to drive the smoke so that the density function of the smoke closely matches the target density function" [SY02]. The target density field can be defined freely by the artist and has no restrictions regarding shapes.

Second, biased diffusion is introduced in order to be able to move the smoke density at higher-resolution detail. Again, it uses the target density field to control the amount of diffusion applied.

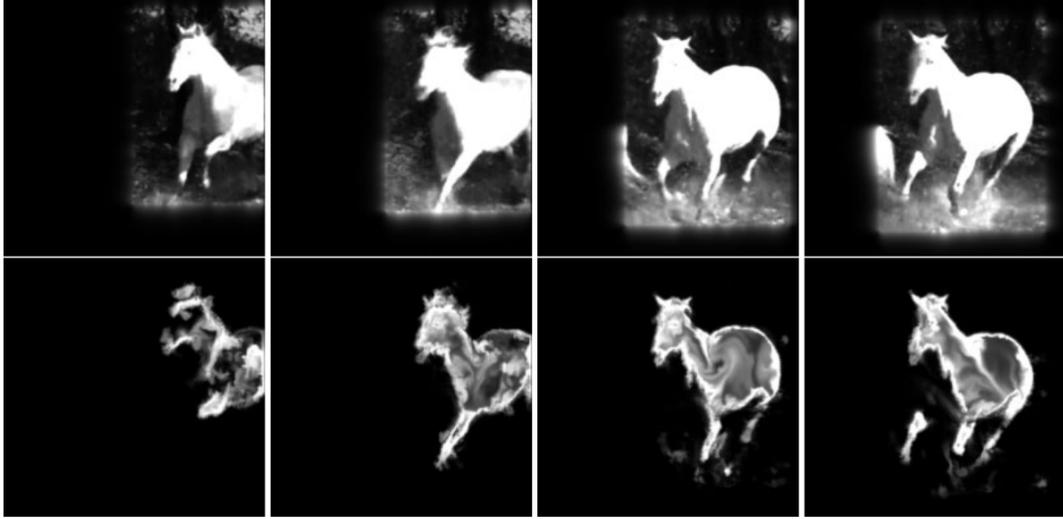


Figure 4.1: "Top row: images from a real horse running sequence which serves as an evolving target density function; Bottom row: images from a controlled smoke simulation approximating both the shape and motion of the running horse" [SY02]

Image courtesy of Shi and Yu. [SY02]

4.1.1 Long Range Force

The long-range force is applied to regions with excess density and the direction points to locations of regions with lower density. This results in transportation of density to these regions. Like other physical forces it decreases as the distance of these two points increases.

The long-range force field is described by the following formula,

$$\mathbf{F}_i^L = \sum_j \frac{[\rho_j - \rho_j^t]^+ \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^\alpha} \quad (4.1)$$

whereas ρ_j describes the current smoke density at grid point j , ρ_j^t describes the target density at grid point j , \mathbf{r}_{ij} is a vector from point i to j and α is the fall-off value.

It would be too costly ($O(ij)$) to calculate every interaction between i and all other grid points. As Zsolnai pointed out, "every sensible choice of α will make the force decay in an at least quadratic manner" [Zso12], which allows us to restrict the distance between i and j to a maximum value. This value depends on the grid resolution and should be adapted when the resolution changes.

4.1.2 Biased Diffusion

Since smoke does not have much diffusion it also has to be taken into account that there is no excessive amount applied. As previously mentioned, the diffusion step was omitted completely because for the same reason. Here, we reintroduce the diffusion part in order to help create the fine details in regions with little density. The altered version, called biased diffusion, only applies

diffusion where there is a difference in smoke density and target density and no diffusion if they are the same.

Introduced by Shi and Yu [SY02], the original diffusion of the density is changed to the following form

$$\mu \nabla^2 \rho \Rightarrow \mu \nabla^2 [\rho^t - \rho] \quad (4.2)$$

which converges to zero if the density field becomes exactly the same as the target density field.

In our approach we changed it to

$$\mu_t(\rho, \rho^t) \nabla^2 [\rho] \quad \text{with} \quad \mu_t \in [0, \mu], \quad (4.3)$$

where now the viscosity coefficient depends on the density and target density field and also converges to zero if the density field becomes exactly the same as the target density field. This is much simpler to implement with different solvers for linear systems with the drawback of having a new field for the viscosity coefficients.

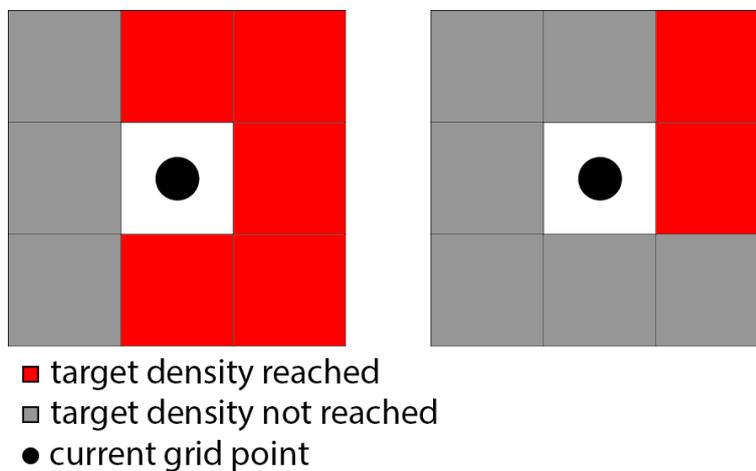


Figure 4.2: Left: more than half of the neighbours reached the target density in which case long-range force is applied to the current grid point. Right: a case where most of the neighbours did not reach the target density in which case the long-range force field is omitted as diffusion takes care of the density distribution.

4.2 Reducing Forces To A Minimum

As described before, excessive use of the long-range external force field results in unnatural behaviour and unconvincing visual results. This is the reason why we introduce an approach to reduce the force to a minimum and prefer biased diffusion to help reach the desired density distribution. It is often the case that excess density in a cell needs to be transferred to many neighbouring cells. Our key idea is that in these cases, we can control the smoke volume without using the long-range field by using the natural, diffusion motions as a substitute. Figure 4.2 shows the two cases of our criterion in 2D, which can be applied to 3D easily.

4.2.1 Criterion

In order to obtain a criterion on how to decide when to use long-range force and when to use biased diffusion only, we look at the following example:

Imagine that a small volume in the centre of the simulation domain is filled with density. If only diffusion is applied it diffuses in all directions in the same way and the density would even out over time. There would be no need for an additional force if the target density has the same shape as our simulation domain.

From this example we derive a criterion in order to decide when to use long-range force. If n is the number of neighbouring grid points of point j which are suitable for diffusion and m is a constant value, then long-range force is applied to grid point j if the following condition is true:

$$n \leq m \quad \text{with} \quad m \in [0, n_{max}] \quad (4.4)$$

where

$$n_{max} \text{ is the amount of neighbouring cells} \quad (4.5)$$

Smooth and convincing results can be achieved if the value of m is $\frac{n_{max}}{2}$.

Finally, we have to define whether a neighbouring point is suitable for diffusion. As described above, there has to be a density difference for diffusion and we apply the restriction that the cell must not reach the target density.

In the following section the algorithm, as described here, is shown:

Algorithm 4.1: Biased diffusion

Data: G all grid points, N_i all neighbours of current point i , D density field, D_t target density field, V viscosity field, F long-range force field

```
1  $i \leftarrow$  current grid point;
2  $n \leftarrow 0$ ;
3  $m \leftarrow \frac{N_i}{2}$ ;
4  $\mu \leftarrow$  viscosity coefficient;
5 for  $i \in G$  do
6   // calculate amount of possible diffusion neighbours;
7   for  $x \in N_i$  do
8     if  $|D[x] - D_t[x]| > 0$  then
9        $n++$ ;
10    end
11  end
12  // decide if we use long-range force;
13  if  $n \leq m$  then
14     $F[i] = true$ ;
15  end
16  // calculate dynamic viscosity coefficient;
17   $V[i] = \mu * \max(0, \min(|D_t[i] - D[i]|, 1))$ ;
18 end
19 // solve diffusion;
20 SolveDiffusion( $V, D$ );
```

Results

In order to compare our approach with normal biased diffusion and long-range force we created a basic test scene. This scene shows the distribution of smoke in a Y-target shape without any additional forces like wind, buoyancy or others. The smoke is only driven by diffusion and long-range force. The resolution of the simulation grid of this scene is 64^3 and in order to make it more realistic we used the wavelet turbulence method introduced by [KTJG08], which does not interfere with the actual smoke control algorithm.

The smoke control was then simulated with four different algorithms (Figure 5.1) in order to highlight every single step. The first rendered image shows normal diffusion of smoke without any forces applied. The second one shows the same but with boundaries applied to the Y-target shape. The third scene uses biased diffusion and long-range force in order to distribute the smoke within the target density. The fourth scene uses our optimised biased diffusion and long-range force algorithm.



Figure 5.1: From left to right. 1: Diffusion without boundaries. 2: Diffusion with boundaries. 3: Biased diffusion and long-range force. 4: Biased diffusion and reduced long-range force.

The first thing that we noticed is that even when applying less force the target density is reached faster. This is due to the fact that the force is only applied where it is needed to steer the density into a desired direction. Figure 5.2 shows the difference between Shi and Yu’s approach and ours.

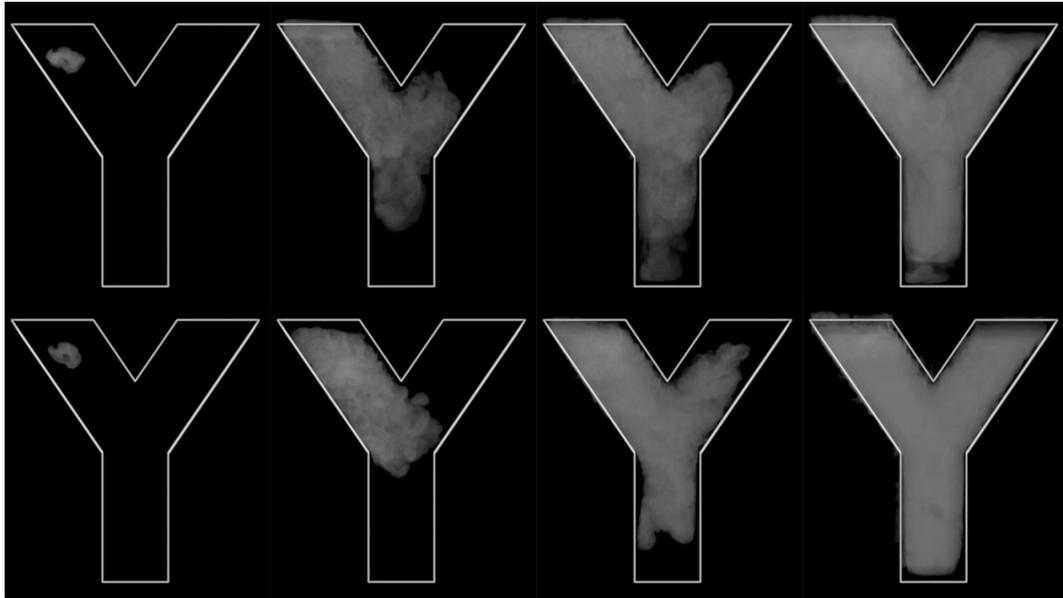


Figure 5.2: Top row: simulation with the algorithm based on Shi and Yu’s approach [SY02], average calculation time per frame: 651ms ; Bottom row: simulation with reduced force approach, average calculation time per frame: 397ms.

The second thing we noticed is that with our approach much less furious smoke movements are noticed. The reason for this is that the old approach applies forces in all directions resulting in forces counteracting with each other and creating furious smoke movements. In the rendered video sequence, which is part of this thesis, the difference can be seen very clearly.

After highlighting the difference of the algorithms Figure 5.3 and 5.4 show different target shapes and Table 5.1 shows the different calculation time of our control algorithm per frame compared to Shi and Yu’s approach [SY02]. The time was measured on a computer with Intel Core i7-2700QM CPU @ 2.40GHz (8 logical CPUs), 8 GB DDR3 RAM and an AMD Radeon HD 6700M 1GB. Although the calculation time is not of very importance here it shows that the calculation of the control part is speed up too.

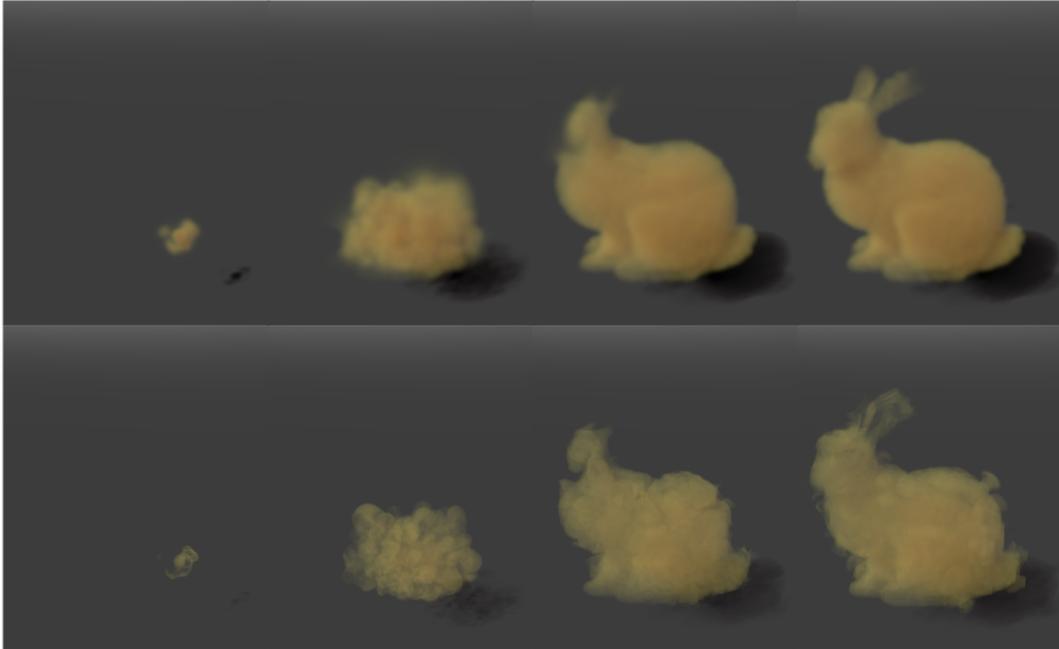


Figure 5.3: Top row: Grid resolution 64^3 . Target shape: Stanford Bunny; Bottom row: Grid resolution 64^3 with Wavelet Turbulence, 4 divisions. Target shape: Stanford Bunny.

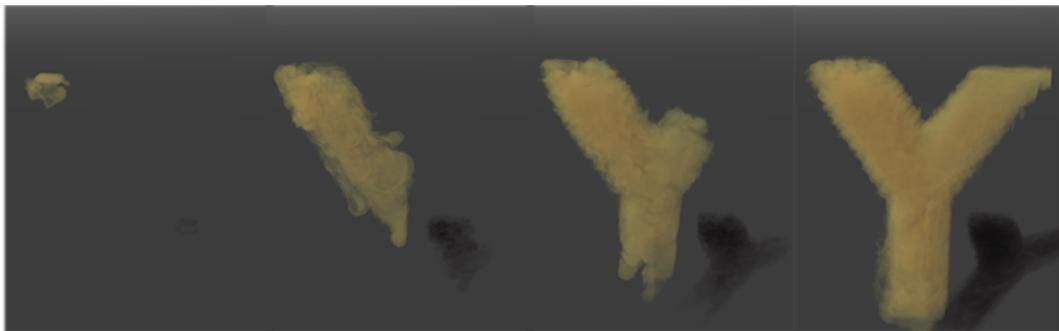


Figure 5.4: Grid resolution 64^3 with Wavelet Turbulence, 4 divisions. Target shape: Letter Y.

Algorithm	Bunny 32^3	Bunny 64^3	Y-Shape 32^3	Y-Shape 64^3
Our	114ms	975ms	45ms	397ms
[SY02]	222ms	1192ms	92ms	651ms

Table 5.1: Shows the different calculation time between our approach and the simplified version of Shi and Yu’s approach using only the long-range force field. Note that the complete version of their algorithm takes 5-8 minutes per frame. The time was measured for the smoke control part of the algorithm and all values are average values per frame.

Conclusions and Discussion

We demonstrated an approach for smoke control with minimal force impact in order to get better natural behaviour and visual results. Our approach is derived from smoke control methods introduced by Shi and Yu [SY02]. Our method uses the long-range force field and biased diffusion combined together with a novel decision rule that restricts the use of long-range forces where natural diffusion does not help the convergence towards a chosen target distribution.

As mentioned, physically correct smoke control is not possible and leaves a wide variety of implementations with different approaches to be experimented with. We showed that our approach contributes to a natural overall behaviour to reach the target density and that it contributes to more realistic smoke motion during the control process.

We note that values for parameters like long-range force strength or long-range force distance, presented in equation 4.1, greatly depend on the resolution of the grid, the target shape and also on the target density. This leaves an artist to find out the right values and try different combinations of them in order to get the desired results.

It would be also desirable to remove the hard boundaries from the target density and design the force field so it keeps the smoke inside without suddenly stopping it at the boundaries. While this can be implemented in 2D with an approach that uses the direction of the tangent on the boundary position to keep the smoke "around", the same approach is not possible in 3D.

The implementation of the algorithm in the open source animation software Blender features the creation of target shapes from any mesh and all necessary parameters for smoke control are accessible through the user interface.

Bibliography

- [Bat00] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press, 2000.
- [BNPB13] Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The helmholtz-hodge decomposition; a survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1386–1404, August 2013.
- [Chu14] Chucklingcanuck. Rheology of time independent fluids, 2014. [Accessed 22-September-2014].
- [EP90] David S Ebert and Richard E Parent. Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. *Computer Graphics*, 24(4), 1990.
- [FL04] Raanan Fattal and Dani Lischinski. Target-driven smoke animation. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 441–448. ACM, 2004.
- [FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58(5):471–483, 1996.
- [FM97] Nick Foster and Dimitris Metaxas. Modeling the motion of a hot, turbulent gas. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 181–188. ACM Press/Addison-Wesley Publishing Co., 1997.
- [FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM, 2001.
- [Ins14] Clay Mathematics Institute. Navier-stokes equation. <http://www.claymath.org/millennium-problems/navier-stokes-equation>, 2014. [Accessed 01-June-2014].
- [KTJG08] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. In *ACM Transactions on Graphics (TOG)*, volume 27, page 50. ACM, 2008.

- [Pri06] James F Price. Lagrangian and eulerian representations of fluid flow: Kinematics and the equations of motion. *Woods Hole Oceanographic Institution, Woods Hole, MA*, 2543, 2006.
- [Sta99] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.
- [Sta03] Jos Stam. Real-time fluid dynamics for games. In *Proceedings of the game developer conference*, volume 18, page 25, 2003.
- [SY02] Lin Shi and Yizhou Yu. Object modeling and animation with smoke. Technical report, Citeseer, 2002.
- [TMPS03] Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. Keyframe control of smoke simulations. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 716–723. ACM, 2003.
- [Zso12] Károly Zsolnai. Real time simulation and control of newtonian fluids using the navier-stokes equations. 2012.