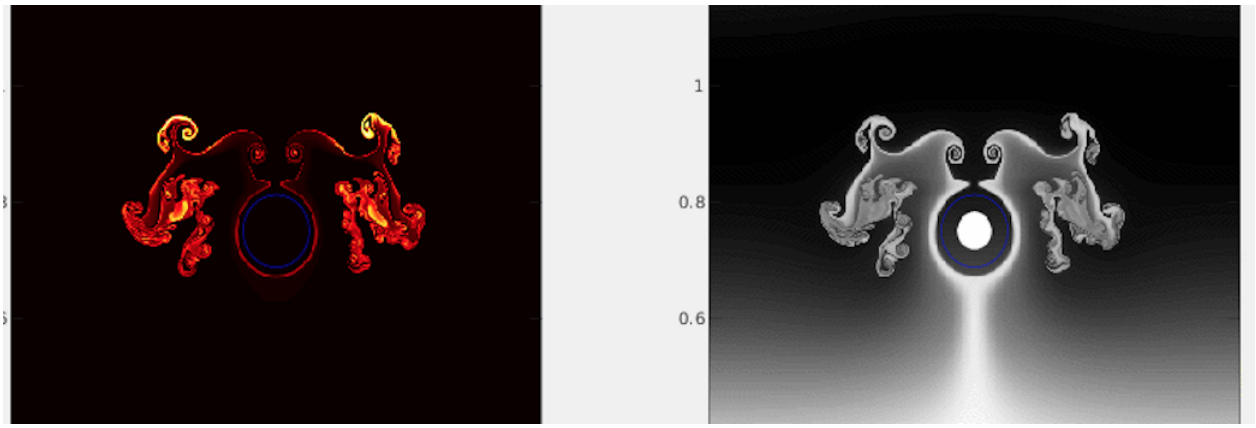
Characteristic Mapping Method for Smoke Simulation with the Euler Equations

PHYS 396 Project Report



Written By: Noah LeFrançois (260706235)

Supervised By: Prof. Jean-Christophe Nave

Working With: Xi Yuan Yin, Ph.D. Candidate

December 18, 2019

Abstract

The goal of this project was the extension of the characteristic mapping method [Mercier and Nave, 2013] [1], [Yin et al., 2019] [2], for the advection of arbitrary sets to a numerical simulation of smoke dynamics in two dimensions, building upon initial work on the problem completed during a summer project by the authors. This simulation takes initial sets defining the temperature and density at all locations and advects these quantities according to the resultant buoyancy forces while obeying the incompressible Navier-Stokes equations. Solid obstacles were defined using a level-set function and two methods were explored to ensure the fluid flows around these obstacles. The first attempted to reduce the normal velocity iteratively using multiplication by a smooth indicator function, while the second modifies the map to prevent points on the boundary from leaving it. The resultant violation of volume conservation is corrected by solving a heat equation and extracting the resultant heat flux to give a correction transport map and restore the map's conservation of volume. By using a combination of fine and coarse grids respectively to decouple the advection and representation steps, computational costs have been reduced while avoiding the numerical dissipation issues faced by many other approaches to this problem. We show the viability of the characteristic mapping method for producing realistic simulations using our physical model for buoyancy, solid-fluid interfaces, and artificial diffusion-like smoothing. This work lays the groundwork for an extension to three dimensions for applications in computer graphics.

1 Introduction

The realistic modelling of natural fluid behaviour is a challenging area of numerical analysis, with a variety of applications from aerospace engineering to nuclear physics. One application which has developed rapidly over the past two decades is the use of fluid simulations for visual effects in movies and video games. In particular, the behaviour of smoke involves multi-scale turbulence, and the accurate representation of the very fine vorticity which is quickly produced is a significant hurdle in this area. As a result, most methods for simulating this phenomenon face a trade-off between either a very fine grid which limits computational speed, or a faster, coarser grid which produces increased numerical viscosity and dissipation

of energy.

One of the first physics-based approaches to this problem was presented in 1999 by [Stam] [3] where an unconditionally stable scheme was used to produce visually realistic flows with short computational time by taking large time steps. While this new approach opened a new avenue for development, the new physics-based methods suffered from excessive numerical dissipation due to the loss of energy on scales smaller than the grid size which acts as a low-pass frequency filter. In 2001, the technique of vorticity confinement was presented by [Fedkiw et al.] [4], working around this problem by injecting vorticity back into the flow to account for the lost energy.

Methods such as FLIP presented by [Zhu and Bridson] [5] in 2005 and Eulerian energy-preserving integrators developed by [Mullen et al.] [6] in 2009 present two approaches to overcoming this challenge. The FLIP method effectively eliminates numerical viscosity but is unable to reduce numerical loss of energy, whereas the fully Eulerian energy-preserving integrators approach suffers from no energy loss but is unable to avoid artificial viscosity. [Wrenninge et al.] [7] proposed a method using a dual Eulerian/Lagrangian approach applying the FLIP technique in 2011. This was able to preserve fine details at low computational cost by decoupling the smoke density resolution from that of the vorticity. Based on the hypothesis that capturing fine details in the smoke field has a greater effect on perceived visual quality than small-scale fluid vorticity, this method reduces overall computational costs for a given simulation resolution by performing vorticity time-stepping on a coarser grid than is used for density advection. Another Eulerian approach, which investigates the coupling of solid-fluid systems where objects are immersed in smoke, was investigated by [Teng et al.] [8] in 2016.

More recently, [Zehnder et al.] [9] obtained greatly reduced energy and vorticity diffusion in 2018 by replacing the conventional projection process used after each advection step to enforce divergence-free conditions in the flow with a reflection operator halfway through each step. This produces significant reduction in energy loss. In exchange it has the same effective computational cost as an advection-projection solver with half the time-step size.

In this work, we present an approach which decouples the advection and representation steps using the characteristic mapping method [Mercier and Nave] [1]. This allows the selection of a coarse grid for the advection steps and only needs to access our finer, more expensive

representation grid after a number of steps when that coarse grid becomes too distorted to accurately represent our flow. To sample the vorticity, temperature and density on a representation grid at a given step, we use the resultant backwards map to trace characteristic curves back from each grid point to its location in the continuous scalar fields representing the initial conditions. In this way, the problem of numerical energy dissipation at each step due to loss of sub-grid scale features is avoided as the initial scalar functions to which the backwards map references at each step are untouched and not dependent upon the resolution used at intermediary steps. At any given step we can obtain arbitrarily fine-scale resolution by sampling the backwards map onto a finer grid. By applying this method to the solution of the incompressible Euler equations with a buoyancy forcing term, realistic smoke behaviour retaining fine-scale features can be produced without the prohibitive computational costs of obtaining similar results with other methods.

The remainder of this paper will be organized as follows. In section 2 we lay out the theoretical formulation of our model, in section 3 we perform a number of simulations and illustrate the behaviour of our model, in section 4 we discuss these simulation results, and in section 5 we summarize the ongoing work and directions for future development.

2 Theory

2.1 Equations of Fluid Flow

In this work we treat our smoke as an inert fluid suspended in and being passively advected by the movement of an inviscid, incompressible fluid of constant density, which we can suppose to be air. The smoke possesses a density and a temperature which depend upon location, while the air possesses a velocity at each location. In two dimensions we define the air velocity $\vec{u} = (u,v)$ and the resultant vorticity $\vec{\omega} = \nabla \times \vec{u}$. Under these assumptions, the fluid vorticity at any location obeys the incompressible Euler equations:

$$\nabla \cdot \vec{u} = 0 \tag{1a}$$

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{u} \cdot \nabla \vec{\omega} = 0 \tag{1b}$$

In our fluid of constant density, equation (1a) enforces conservation of mass by requiring a divergence-free flow, while equation (1b) is an advection equation enforcing conservation of momentum. In addition to the flow of vorticity in our air defined by equations (1a) and (1b), we also let the two scalar quantities defining our smoke (density, ρ, and temperature, T) be passively advected by the velocity field:

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = 0 \tag{2a}$$

$$\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho = 0 \tag{2b}$$

2.2 Forcing Terms

In order to create physically realistic simulations of smoke, the predetermined velocity fields originally used by [Mercier and Nave] [1] are replaced in this work by a vertical forcing term, $\vec{f}$, modeling the buoyancy of a gas with non-uniform temperature and density. This simple buoyancy model produces an upward force on the air where the local temperature is greater than the ambient temperature T_{Amb}, while providing a downward force where the local density is greater than the ambient density ρ_{Amb} [Fedkiw et al.] [4]:

$$f_y = b \cdot (T(x, y) - T_{Amb}) - a \cdot (\rho(x, y) - \rho_{Amb}) \tag{3}$$

These forces influence the movement of our suspension fluid by causing changes in the vertical velocity component due to the distribution of temperature and density in our smoke. The momentum-conserving Eq. (1b) is thus adapted to allow the injection of momentum by the addition of the curl of our forcing term, giving Eq. (4) for the forced advection of air vorticity:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{u} \cdot \nabla \vec{\omega} = \nabla \times \vec{f} \tag{4}$$

2.3 Characteristic Mapping Method

The characteristic mapping method allows the decoupling of the advection and representation steps required for the advection of a set, which greatly improves computational efficiency and is highly parallelizable while supporting the advection of multiple sets simultaneously under the same flow [Mercier and Nave] [1], [Yin et al.] [2] (a property we exploit for the passive advection of T and ρ in equation (2)). The Gradient-Augmented Level Set method [Nave et al., 2010] [10] is used to calculate the footpoints of characteristic curves. We compute a backwards map χ which solves the IVP:

$$\frac{d}{dt}\vec{x}(t) = \vec{u}(\vec{x}(t), t), \vec{x}(0) = \vec{x}_0 \tag{5}$$

corresponding to the Euler advection problem. This map is the solution to the advection equation and we can write:

$$\frac{\partial \chi}{\partial t} + \vec{u} \cdot \nabla \chi = 0 \tag{6}$$

The map χ is known to be a diffeomorphism, and using it we can calculate any advected quantity, such as the vorticity, at a given location using Eq. (7a), where ω_0 is the vorticity field at t=0. Then the velocity is calculated according to Eq. (7b):

$$\omega(\vec{x}, t) = \omega_0(\chi_B(\vec{x}, t)) \tag{7a}$$

$$\vec{u}(\vec{x}, t) = -\nabla^{\perp}\Delta^{-1}\omega(\vec{x}, t) \tag{7b}$$

Eq. (7b) can be broken down into two steps, first obtaining the stream function Ψ by solving the Laplace equation using a Fourier transform. Then the 2-dimensional curl is calculated using a Hermitian interpolant:

$$\Psi^n = \Delta^{-1}\omega(\vec{x}, t) \tag{8a}$$

$$\vec{u}^n = -\nabla^{\perp} H[\Psi^n] \tag{8b}$$

This velocity is then used to transport the temperature, density, and vorticity according to the advection equations (2a), (2b) and (4).

This advection step is computed on a relatively coarse grid; a more costly fine grid is only accessed to update the full backwards map once the coarse grid becomes too distorted to effectively represent the fluid flow. After this representation step updates the full backwards map χ_B by composing it with the current advection map χ_i, we reset χ_i to the identity map and continue the advection process. This alternating process continues until the simulation is terminated, repeatedly updating χ_B by 'remapping' when the flow is too distorted.

2.4 Reinitialization of the Map

In section 1 we discussed the ability of the characteristic mapping method to resolve flow features at arbitrary scales resulting from its backwards mapping to the initial temperature and density fields stored as continuous functions. This behaviour was further examined by [Yin et al.], demonstrating the ability to zoom in to examine very fine details of a simulated vortex. While providing theoretically infinite flow resolution, eventually the continued composition of many nested remappings becomes increasingly expensive and the simulation slows proportionately. However, for many computer graphics applications this infinite resolution is not an essential feature; it is sufficient to choose a maximum resolution for the simulation which we do not need to exceed in order to produce visually realistic results. By placing an upper limit upon resolution, we effectively introduce a tune-able diffusion effect at the scale of our minimum cell width, as features smaller than this will be smoothed by re-sampling onto our finest allowed grid. This simplification allows us to reinitialize our scalar fields after a chosen number of remappings, essentially restarting our simulation from the new set of 'initial conditions' (ω_0, T_0, ρ_0) produced by sampling the current fields onto a grid of our chosen maximum resolution. In essence, this splits the full simulation up to final time τ_f into a number of successive sub-simulations each of which takes its initial conditions from the final conditions of its predecessor. As a result, the number of remappings which must be stored during any given sub-simulation is reduced and the simulation as a whole is accelerated while retaining the chosen maximum resolution.

2.5 Obstacles

Solid obstacles are represented by a distance function ϕ. The interior of the obstacle is defined as the region where $\phi(x, y)<0$ and its boundary is the zero-level-set with unit-normal defined as $n_\phi = \nabla\phi$. In order to place this stationary obstacle in our simulation, a technique for preventing the flow of smoke into the interior of the obstacle is required. Two approaches were explored in this work.

Our initial approach enforced the boundary by projecting the velocity at each time step in order to remove the normal component. To do so, n_ϕ was multiplied by a smoothed indicator function at the boundary and this process was iterated repeatedly until the normal component had been sufficiently reduced. This method was not deemed satisfactory as it eventually generated very large vorticity at the boundary which led to a blow-up of the residual normal component after the iterative projection and crashing of the program.

The second method considered allows the map to flow at each time step and then checks for points which have entered the obstacle interior ($\phi(x, y)<0$). Any such points are then pulled back along their trajectory to be placed back on the boundary in order to rectify any obstacle penetration. One issue is that this method violates the volume preservation of our flow; in other words the determinant of the Jacobian of χ_B, $det(J\chi_B)$, is no longer equal to 1. By solving a heat equation (Eq. 9) in which $\varrho_0 = det(J\chi_B)$ is used as the density field, a characteristic map Φ is found which moves $\varrho \to 1$ as $t' \to \infty$:

$$\partial_{t'}\varrho = \Delta\varrho \tag{9a}$$

$$\varrho(x, t' = 0) = \varrho_0 \tag{9b}$$

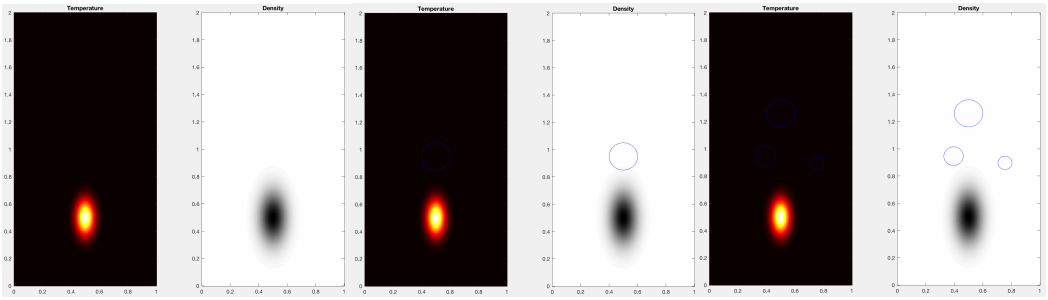
The final step of this penalty method is to compose Φ with χ_B in order to produce the final updated map, i.e. $\chi_B \leftarrow \chi_B \circ \Phi$.

Compared to the first approach which attempts to reduce n_ϕ but does not guarantee convergence to a solution without penetration, this method offers the advantage of penalizing the blow-up of n_ϕ and converging to a much better approximation of a solid, penetration-free boundary in the limit of increasingly fine grids. However, further work is required to

examine the error introduced in the average fluid velocity by the omission of a constant term in the vorticity when taking the derivative in Eq. 5b. This leads to an average velocity of zero in the reference frame of the obstacle since it behaves as a massless object and is therefore accelerated to the reference frame of the fluid. In order to correct this error and keep the object frame from accelerating, future investigations can examine methods to add the omitted vorticity term back into the fluid so that the average fluid velocity is not lost.

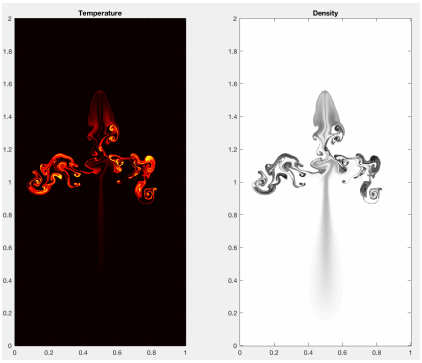
3 Results

To test our methods, a 2D smoke simulation was run with and without a circular obstacle. As seen in figure 1, the initial temperature and density fields were each defined as an ellipse with intensity inversely proportional to radius. Three tests were performed: free flow without an obstacle, flow around a single obstacle, and flow around/between 3 obstacles. These tests were run using our first obstacle-handling method of velocity projection as well as using our second method of volume-preserving penalization.



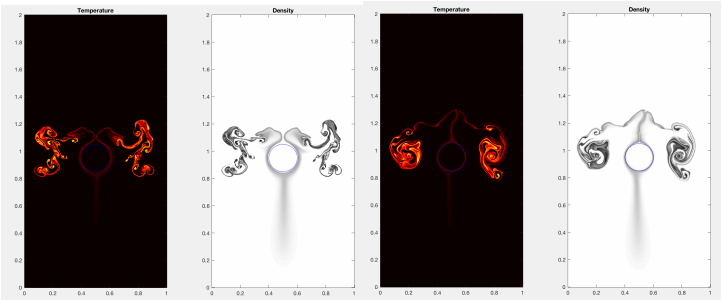
(a) No obstacle (b) Single Obstacle (c) Multiple Obstacles

Figure 1: Screenshots of our three test initial conditions. Obstacles can be seen as a blue circular outline in the temperature map, and a white area inside a blue circle in the density map.



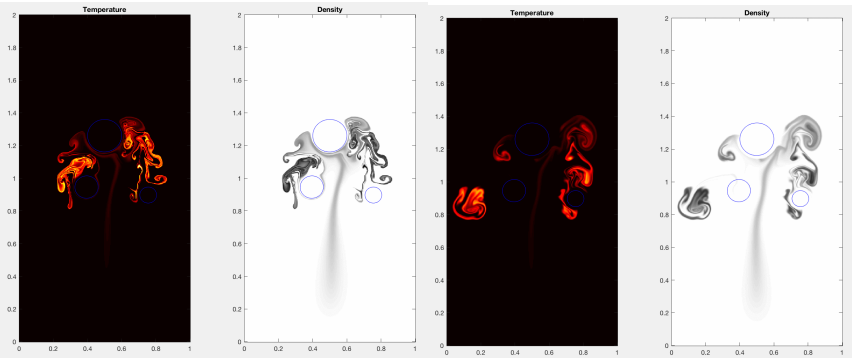
(a)

Figure 2: Screenshot showing the test without an obstacle, with initial conditions seen in Fig.1a



(a) Velocity Projection Method (b) Penalization Method

Figure 3: Screenshots showing the test with a single obstacle, with initial conditions seen in Fig.1b



(a) Velocity Projection Method (b) Penalization Method

Figure 4: Screenshots showing the test with multiple obstacles, with initial conditions seen in Fig.1c

4 Discussion

Traditional methods project the discretized temperature and density fields onto a fixed representation grid after each advection step, therefore having a fixed resolution based on the chosen grid size. As a result, information which is not retained when sub-grid size features of the flow are projected onto a grid at the first time-step is permanently lost. This numerical dissipation will be felt due to inaccuracy in the starting point of the second time-step, since the flow that is to be advected in that step has lost energy compared to the true solution. Much of the literature relating to smoke simulation for graphics applications is devoted to techniques for reducing this dissipation or artificially re-injecting energy to counteract losses. Our method, on the other hand, stores the initial fields as continuous functions which can be projected onto a grid of any arbitrary resolution using the backwards map. Importantly, the projection of our field functions onto a finite grid does not change them as they remain stored as functions ready for use again at the next time-step.

This key difference means that the level of detail obtainable at a given time is not constrained by the resolution with which preceding time steps are represented, since the initial fields to which the pullback coordinates reference remain unchanged throughout the simulation. An important consequence of this behaviour, as explored by [Mercier and Nave] [1], is the use of a dynamic grid size which can be made finer or coarser according to the flow complexity. Following the flow complexity improves efficiency by only using the required resolution at any given time instead of fixing the resolution of all steps at the maximum required value throughout the entire simulation.

A major question which remains is how to deal with the constant vorticity term which is omitted when calculating the velocity at each step; this omission affects the average velocity of the fluid relative to the obstacle reference frame. As a result flows observed from the obstacle reference frame can lose the vertical acceleration which the buoyancy terms should be providing. Some initial approaches to this problem have been attempted by averaging the initial forcing term magnitude over the domain and adding this average vertical acceleration in artificially. Early testing of this approach shows better-looking results which continue to accelerate upwards after passing the obstacle. However, this very heuristic technique is still

being developed and it remains to be investigated which of our two approaches to handling obstacle boundaries is most suited for adaptation to the issue.

5 Conclusions

While this project has demonstrated the feasibility of the characteristic mapping method as a technique for visually believable and computationally efficient simulation of smoke flow in two dimensions with obstacles, the eventual goal of this work is the extension of our model to three dimensions. This will enable the application of our method to visual effects in areas such as film and video game graphics. Once these 3D simulations are implemented, we have also implemented and tested a basic ray tracing algorithm for rendering our results based upon the methods of [Fedkiw et al.] [4] and [Rasmussen et al.] [11]. Alternatively, a commercial animation software could be used for rendering. This will produce more comprehensible visual outputs than the 2D heat map representation used in this work, enabling better examination of our simulation quality during testing.

Through the production of our 2D work, we have selected a physical model and a number of numerical techniques which we believe can be adapted to 3D. For example, our treatment of obstacle boundaries by either projecting the vorticity to enforce parallel flow or a volume-preservation penalty method are promising candidates for use in 3D. However, some aspects of our model do not seem to lend themselves well to the addition of a third dimension, namely our treatment of the forcing term in our advection problem. In 3D, a vorticity-stretching term is introduced which will require a different approach to solving this equation. In conclusion, this method shows promise but requires significant further efforts to reach the eventual results of which it appears to be capable.

References

[1] Olivier Mercier and Jean-Christophe Nave, *The Characteristic Mapping Method for the Linear Advection of Arbitrary Sets* (September 2013) 1, 2, 4, 5, 10

[2] Xi-Yuan Yin, Olivier Mercier, Badal Yadav, Kai Schneider, Jean-Christophe Nave, *A Characteristic Mapping Method for the Two-Dimensional Incompressible Euler Equations* (2019) 1, 5

[3] Jos Stam, *Stable Fluids*, SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pages 121–128 August (1999) 2

[4] Ronald Fedkiw, Jos Stam and Henrik Wann Jensen, *Visual Simulation of Smoke* (2001) 2, 4, 11

[5] Yongning Zhu, Robert Bridson, *Animating Sand as a Fluid*, ACM Transactions on Graphics 24, 3 965–972 (2005) 2

[6] Patrick Mullen, Keenan Crane, Dmitry Pavlov, Yiying Tong, and Mathieu Desbrun, *Energy-preserving Integrators for Fluid Animation*, ACM Trans. Graph. 28, 3, Article 38 (2009) 2

[7] Magnus Wrenninge, Henrik Fält, Chris Allen, Stephen Marshall, *Capturing Thin Features in Smoke Simulations*, Sony Pictures Imageworks (2011) 2

[8] Yun Teng, David I.W. Levin, Theodore Kim, *Eulerian Solid-Fluid Coupling*, Pixar Online Library (2016) 2

[9] Jonas Zehnder, Rahul Narain, Bernard Thomaszewski, *An Advection-Reflection Solver for Detail Preserving Fluid Simulation*, ACM Trans. Graph. 37, 4, Article 85 (2018) 2

[10] Jean-Christophe Nave, Rodolfo Ruben Rosales, and Benjamin Seibold, *A gradient- augmented level set method with an optimally local, coherent advection scheme*, Journal of Computational Physics **229**, no. 10, 3802–3827 (2010) 5

[11] Nick Rasmussen, Duc Quang Nguyen, Willi Geiger and Ronald Fedkiw, *Smoke Simulation for Large-Scale Phenomena* (2003) 11