The Moving Contact Line in a Shallow Water Model

by

Albert C. Wong

B.Sc., The University of British Columbia, 2003

AN ESSAY SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Computer Science)

The University Of British Columbia

April, 2006

Abstract

This essay presents a method for coupling a level set method with a shallow water model allowing us to track the contact line of the fluid in the shallow water model. A finite volume method is used to conserve the volume of the fluid while semi-Lagrangian advection is used to improve the stability of our model. Additionally, a fast sweeping method is used to update the level set over time while the velocity field is extrapolated outside of the fluid using a partial differential equation approach.

Table of Contents

List of Figures

Acknowledgements

Firstly, I would like to thank my supervisor, Robert Bridson, for all his help and patience with me. Without his guidance, I would not have learned as much as I have over the last few years. I wish him all the best.

I would also like to thank all the students in the lab and professors who have given me so much to think about during my years at UBC.

Lastly, I would like to thank my family and friends for their encouragement and support during my difficult times.

Chapter 1

Introduction

Shallow water models have been used in the past to model thin layer fluid flow in computer graphics [6, 7]. In addition, the shallow water equations are often used in the areas of atmospheric sciences and oceanography for large-scale global modelling. One problem shallow water models face when simultaing wetting and drying is that it is difficult to keep track of the contact line (the intersection between the air, fluid, and the ground). We propose using a level set method for tracking the contact line in a shallow water model.

1.1 Overview

In Chapter 2, we formulate the shallow water model that we used. Chapter 3 discusses level set methods and how we incorporate level sets to track the contact line in our shallow water model. We conclude with some results of our method and discuss possible future work in Chapter 4.

Chapter 2

Shallow Water Model

The shallow water equations are based on the full Navier-Stokes equations for incompressible flow. We present a derivation of the shallow water model that we use in this chapter.

2.1 Navier-Stokes Equations

The Navier-Stokes equations are a set of non-linear equations derived from Newton's second law of motion that describe the motion of a fluid particle at any point in space and time. For an incompressible fluid in three dimensions where the only external force is gravity, the equations are

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = g_x - \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right), \quad (2.1)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = g_y - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right), \quad (2.2)$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = g_z - \frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right), \quad (2.3)$$

where u, v, and w are the velocities in the x, y, and z directions respectively, g is the gravity, ρ is the density, p is the local pressure, and ν is the kinematic viscosity of the fluid. The Navier-Stokes equations are often applied along with the continuity equation, which is derived from the principle of conservation of mass:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \quad (2.4)$$

Simplifying the Navier-Stokes equations and the continuity equation using vector notation, we have

$$\rho\left(\frac{\partial}{\partial t} + \vec{U}\cdot\nabla\right)\vec{U} \;=\; -\nabla p + \mu\nabla^2\vec{U} + \rho\vec{g}, \quad (2.5)$$

$$\nabla \cdot \vec{u} \;=\; 0, \tag{2.6}$$

where $\vec{U}$ and $\vec{g}$ are the velocity and gravity vectors respectively and $\mu = \rho\nu$. There are many fluid simulation models such as those in [3, 4, 8] based on the Navier-Stokes equations. Many interesting phenomena can be simulated such as water being poured into a glass or objects dropping into a body of fluid. The main drawback of such simulation models is that the algorithms tend to take $\mathcal{O}(N^3)$ time to run where N is the number of grid subintervals along a single dimension.

2.2 Shallow Water Equations

The shallow water equations describe flows of thin layers of fluid and are derived from the Navier-Stokes equations by making several simplifying assumptions. For a column of fluid, let h be the height of the fluid and A be the base area of the fluid. Then the total mass of the column of fluid is

$$m = hA\rho, \tag{2.7}$$

and the force on the base of the column due to gravity is

$$f = mg. \tag{2.8}$$

Combining (2.7) and (2.8), we see that the pressure at the base of the column is

$$p = \frac{f}{A} = hg\rho \tag{2.9}$$

so that the pressure gradient is then

$$\nabla p = g\rho\nabla h. \tag{2.10}$$

Hence, assuming that the acceleration along the z direction is negligible, the Navier-Stokes equation simplifies to

$$\rho\left(\frac{\partial}{\partial t} + \vec{U}\cdot\nabla\right)\vec{U} = -g\rho\nabla h + \nu\nabla^2\vec{U} + \vec{g}. \tag{2.11}$$

where velocity changes in the z direction are zero. Thus, the equation simplifies to

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right) = -g\rho\frac{\partial h}{\partial x}-\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \quad (2.12)$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right) = -g\rho\frac{\partial h}{\partial y}-\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right). \quad (2.13)$$

To derive the equation for the height, we integrate the continuity equation along the z direction to get

$$\int_0^h\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)dz=\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)h+w_h-w_0=0. \quad (2.14)$$

Now, $w_0=0$ since w_0 is the vertical speed at the bottom while $w_h=\frac{dh}{dt}$ since w_h is the vertical velocity at the fluid's surface. Simplifying, we get

$$\frac{\partial h}{\partial t}+u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y}=-\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)h. \quad (2.15)$$

Finally, by assuming the fluid is inviscid where $\|\mu\nabla^2\vec{U}\|_2 \ll \|\nabla p\|_2$, the shallow water equations become

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y} = -g\frac{\partial h}{\partial x}, \quad (2.16)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y} = -g\frac{\partial h}{\partial y}, \quad (2.17)$$

$$\frac{\partial h}{\partial t}+u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y} = -\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)h. \quad (2.18)$$

Fluid simulation models based on the shallow water equations are able to model thin layer flows of fluid [6]. The complexity of these models tend to be $\mathcal{O}(N^2)$ since we need only keep track of the depth-average horizontal velocities and heights on an $N \times N$ grid. Often, instead of using a regular grid that stores the height and velocities at the same grid point, often a staggered grid is used for shallow water simulations. In our shallow water model, we use the Marker-and-Cell (MAC) grid [5] with a third-order TVD Runge-Kutta scheme [11].

2.2.1 Semi-Lagrangian Method

While a finite difference scheme using central difference approximations with zero flow along the boundaries can be used to model the shallow water equations, a more stable method is the semi-Lagrangian method first proposed in [2]. The semi-Lagrangian method is a hybrid between Eulerian and Lagrangian methods. In an Eulerian approach, one stays in a fixed point in space as the fluid evolves. This approach maintains the regularity of a grid since points stay fixed but requires small time steps to remain stable. In a Lagrangian approach, one follows a particle of fluid as the fluid evolves. This approach requires less restrictive time steps but particles tend to become irregularly spaced over time. The semi-Lagrangian method attempts to get the best of both worlds: the regularity of Eulerian methods and the stability of Lagrangian methods. The idea is to choose a different set of particles in each time step such that the set of particles arrive at the points of a Cartesian grid at the end of the time step. This consists of backward tracing of the characteristic information from the points on the Cartesian grid to arbitrary points in space followed by interpolating the values of the backward-traced points from data on the grid points. A review of semi-Lagrangian methods can be found in [12].

For the semi-Lagrangian formulation of the shallow water equations, we first replace the substansial derivatives in the shallow water equations with ordinary derivatives

$$\frac{du}{dt} = -g\frac{\partial h}{\partial x}, \tag{2.19}$$

$$\frac{dv}{dt} = -g\frac{\partial h}{\partial y}, \tag{2.20}$$

$$\frac{dh}{dt} = -\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) h, \tag{2.21}$$

where the ordinary derivatives are estimated using forward differences to track a particle located at grid point (i, j)

$$\frac{du}{dt} = \frac{u(t+\Delta t) - \tilde{u}(t)}{\Delta t} + \mathcal{O}(\Delta t), \tag{2.22}$$

$$\frac{dv}{dt} = \frac{v(t+\Delta t) - \tilde{v}(t)}{\Delta t} + \mathcal{O}(\Delta t), \tag{2.23}$$

$$\frac{dh}{dt} = \frac{h(t+\Delta t) - \tilde{h}(t)}{\Delta t} + \mathcal{O}(\Delta t). \tag{2.24}$$

$\tilde{h}$, $\tilde{u}$, and $\tilde{v}$ are calculated by backward-tracing from grid point (i, j) to the point of departure. Let $\vec{\alpha}$ be the displacement from the grid point to the backward-traced point, $\vec{x}_{i,j}$ be the location of the grid point (i, j), and $\vec{U}$ be the velocity field. To find the location of the backward-traced point, we solve

$$\vec{\alpha} = \Delta t \vec{U}(\vec{x} - \vec{\alpha}, t) \tag{2.25}$$

iteratively using $\Delta t \vec{U}(\vec{x}_{i,j}, t)$ as an initial guess. We then estimate the values of $\tilde{h}$, $\tilde{u}$, and $\tilde{v}$ at the backward-traced point using bilinear interpolation.

2.2.2 Finite Volume Method

For shallow water simulations, one drawback of using semi-Lagrangian advection and finite difference methods on the heightfield is that these methods do not conserve the volume of the fluid. For certain cases, the volume of the fluid can change significantly over time. One solution is to use a finite volume method [13] for computing the heightfield. In the finite volume method, each grid has a control volume Ω surrounding it. Using the divergence theorem, the volume integral of a partial differential equation containing a divergence term is transformed into a surface integral. The terms in the surface integral are then evaluated as fluxes across the surface of the control volume $\partial\Omega$. Using the finite volume method, the change in depth $D = h - b$ of the fluid at a grid point is then

$$\frac{\partial}{\partial t} \int_{\Omega} D d\Omega = - \oint_{\partial\Omega} ((D \cdot u)dy + (D \cdot v)dx) \tag{2.26}$$

where b is the elevation of the ground at the grid point. The volume of the fluid is conserved by ensuring that the fluxes between adjacent control volumes is balanced. In our approach, the fluxes are calculated using upwind differencing.

Chapter 3

Level Set Method

Level set methods are commonly used to track the interface of a fluid in numerical simulations. Simulations of fluid involving merging and pinching of fluids are easily handled using level set methods. Additionally, level sets are able to perform operations such as approximate distance and direction to the surface from any point in space in an efficent manner. This chapter provides background information on level set methods (A more detailed look at level set methods can be found in [10]).

3.1 Implicit Functions

An implicit interface representation is the isocontour of some function $\phi : \mathcal{R}^n \rightarrow \mathcal{R}$. More formally, we define an implicit surface as the set of points $\vec{x} \in \partial\Omega \subset \mathcal{R}^n$ satisfying

$$\phi(\vec{x}) = c \tag{3.1}$$

for some constant c which is typically chosen to be zero. While ϕ is a function defined on $\mathcal{R}^n$, the interface is only of dimension $n-1$. For closed interfaces, an interface defines a surface in $\mathcal{R}^n$ that separates the space into distinct subdomains with interior and exterior regions. We define $\Omega^- = \{\vec{x} \,|\, \phi(\vec{x}) < c\}$ as the interior region and $\Omega^+ = \{\vec{x} \,|\, \phi(\vec{x}) > c\}$ as the exterior region. Since the gradient $\nabla\phi$ is perpendicular to the isocontours of ϕ, the unit normal for ϕ is

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{3.2}$$

3.2 Signed Distance Functions

A distance function $d(\vec{x})$ is defined as

$$d(\vec{x}) = \min_{\vec{x_I} \in \partial\Omega} (|\vec{x} - \vec{x_I}|). \tag{3.3}$$

A signed distance function is an implicit function ϕ defined as

$$\phi(\vec{x}) = \begin{cases} d(\vec{x}) & : \quad \vec{x} \in \Omega^+ \\ 0 & : \quad \vec{x} \in \partial\Omega \\ -d(\vec{x}) & : \quad \vec{x} \in \Omega^- \end{cases} \tag{3.4}$$

which implies that $|\nabla\phi| = 1$. Hence, the unit normal for the signed distance function ϕ is

$$\vec{N} = |\nabla\phi|. \tag{3.5}$$

3.3 Level Set Equation

Introduced by Osher and Sethian [9], the level set method is an Eulerian method for tracking interfaces as they evolve in time. The idea behind level set methods is to represent the interface with an implicit function and use the level set equation to evolve the interface. The level set equation is given by

$$\frac{\partial\phi}{\partial t} + \vec{U} \cdot \nabla\phi = 0 \tag{3.6}$$

where $\vec{U}$ is the velocity field. Often, ϕ is initialized to be the signed distance function for the interface while samples of the signed distance function are stored on a fixed computational grid.

The contact line where the ground, fluid, and air intersect in the shallow water model can be tracked using a signed distance function that is evolved over time by semi-Lagrangian advection of the level set equation. In the finite volume method, we also set the flux to be zero for any control volume whose grid points lie outside of the contact line in order to prevent the fluid from travelling faster than the contact line. One problem with using the signed distance function in a level set method is that as the interface evolves, ϕ will no longer be a signed distance function. Two major strategies for keeping ϕ to be approximately equal to signed distance are reinitialization and marching [10].

3.4 Signed Distance Computation

Zhao [15] presents a fast sweeping algorithm for computing the numerical solution of the Eikonal equation

$$|\nabla u(\vec{x})| = f(\vec{x}),\ \vec{x} \in \mathcal{R}^n \tag{3.7}$$

with boundary condition, $u(\vec{x}) = \beta(\vec{x})$, $\vec{x} \in \Gamma \subset \mathcal{R}^n$. It is shown that the algorithm requires four iterations in 2 dimensions and requires $\mathcal{O}(N^2)$ running time. The distance function $d(\vec{x})$ is the exact solution of the Eikonal equation with $f(\vec{x}) = 1$. Using the fast sweeping method from [15], the signed distance function $\phi(\vec{x})$ for the contact line is computed as follows:

Discretization. Using a Godunov upwind differencing scheme, the interior grid points are discretized by

$$[\max((\phi_{i,j} - \phi_{x\min}), 0)]^2 + [\max((\phi_{i,j} - \phi_{y\min}), 0)]^2 = (\Delta x)^2 \tag{3.8}$$

where $\phi_{x\min} = \min(\phi_{i+1,j}, \phi_{i-1,j})$ and $\phi_{y\min} = \min(\phi_{i,j+1}, \phi_{i,j-1})$. One sided differences are used at the boundaries of the grid.

Initialization. Exact values for ϕ are assigned at grid points near the interface and remain fixed throughout the computation. Large positive values are assigned to the unfixed grid points outside of the fluid while the grid points on the interior of the fluid remain unchanged.

Gauss-Seidel iterations with alternating sweeping orderings. For unfixed grid points that are outside the interface, we compute the solution $\overline{u}$ from the current values of its neighbours $u_{i\pm1,j}$ and $u_{i,j\pm1}$. $u_{i,j}$ is then updated to be $u_{i,j}^{\text{new}} = \min(u_{i,j}^{\text{old}}, \overline{u})$. The entire domain is swept with four alternating orderings for four iterations:

$$\begin{aligned}
&(1)\ i = 1:N \quad , \quad j = 1:N,\\
&(2)\ i = N:1 \quad , \quad j = 1:N,\\
&(3)\ i = N:1 \quad , \quad j = N:1,\\
&(4)\ i = 1:N \quad , \quad j = N:1.
\end{aligned}$$

The solution to $\overline{u}$ is given by

$$\overline{u} = \begin{cases} \min(a,b) + \Delta x & , \quad |a-b| \geq \Delta x, \\ \frac{a+b+\sqrt{2(\Delta x)^2 - (a-b)^2}}{2} & , \quad |a-b| < \Delta x, \end{cases} \tag{3.9}$$

where $a = u_{x\min}$ and $b = u_{y\min}$.

Computing the interior points. To compute the grid points on the interior of the fluid, we negate the values of the fixed grid points, set the interior grid points to be large positive values, and perform the Gauss-Seidel iterations with alternating sweepings on the interior grid points. Finally, we negate the values of the fixed grid points and the interior grid points to obtain the approximated signed distance function.

3.5 Extrapolation

In order to advect grid points outside of the fluid, we must extend the velocity field from inside the fluid outwards. One way of accomplishing this is by extrapolation. We use a method for multidimensional extrapolation based on a partial differential equation approach discussed in [1]. To perform constant extrapolation of the velocity field, we solve the PDEs to steady state

$$\frac{\partial u}{\partial t} + H(\phi)\vec{N} \cdot \nabla u = 0, \tag{3.10}$$

$$\frac{\partial v}{\partial t} + H(\phi)\vec{N} \cdot \nabla v = 0, \tag{3.11}$$

where ϕ is the signed distance function, $\vec{N} = \nabla\phi$ is the outward normal direction from the contact line, and H is the Heaveside function

$$H(\phi) = \begin{cases} 1 & \text{if} \quad \phi > 0, \\ 0 & \text{if} \quad \phi \leq 0. \end{cases} \tag{3.12}$$

Once the PDEs are in a steady state, we have $\vec{N} \cdot \nabla u = 0$ and $\vec{N} \cdot \nabla v = 0$ which implies that u and v are constant along the outward normal direction $\vec{N}$. One numerical approach to approximating the solution to the PDEs is to use a fast sweeping method similar to the one in [15] but using only the values from points that are closer to the contact line.

Chapter 4

Conclusions

We have implemented the shallow water model in 1 and 2 dimensions using finite volumes to calculate the heightfield, semi-Lagrangian advection and finite differences to calculate the velocities, and a level set to track the contact line. Figures 4.2 - 4.4 show results for several 1-dimensional shallow water models while figure 4.5 show the results for our model. Figure 4.1 shows the initial heightfield for all the models. In comparison with the other models, our model did not suffer from any loss in the volume of fluid and high frequency oscillations were not visible near the contact line although the semi-Lagrangian advection caused some numerical diffusion. In the cases where the fluid flooded the surface of the ground, the level set managed to track the contact line.

4.1 Future Work

While coupling the level set with the shallow water equations allowed us to track the contact line as the fluid flooded a region, our model does not handle the drying of a region. In order to do this, a better method for coupling the level set with the finite volume method must be researched. Such a model may be modified to incorporate surface tension and contact line forces in order to simulate water drops on surfaces. While Wang et al. [14] proposed a method for simulating water drops on a surface that can handle more types of phenomena involving droplets, simulations typically takes 5–8 days to simulate on a single Pentium Xeon 2.8GHz machine. A shallow water model of droplets would require significantly less time to simulate.

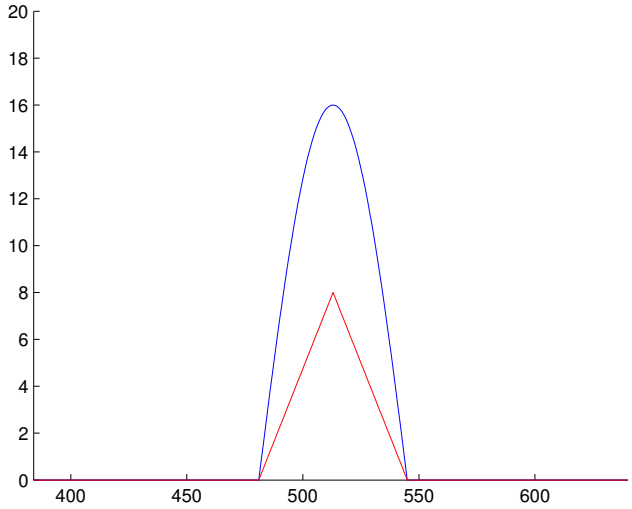

Figure 4.1: $t = 0.0$, Initial Heightfield. The blue line represents the surface of the fluid while the red line represents the ground.

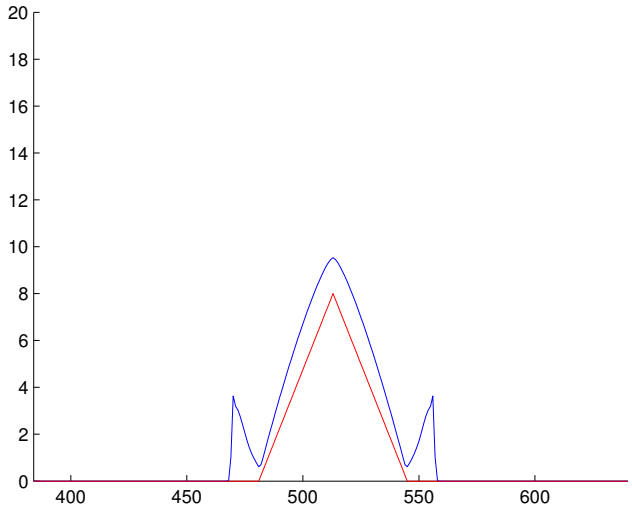

Figure 4.2: $t = 3.0$, Finite Differences with Semi-Lagrangian Advection. After 3.0 seconds of simulation, the volume of the fluid decreased by roughly 60%. High frequency oscillations were visible near the contact line.

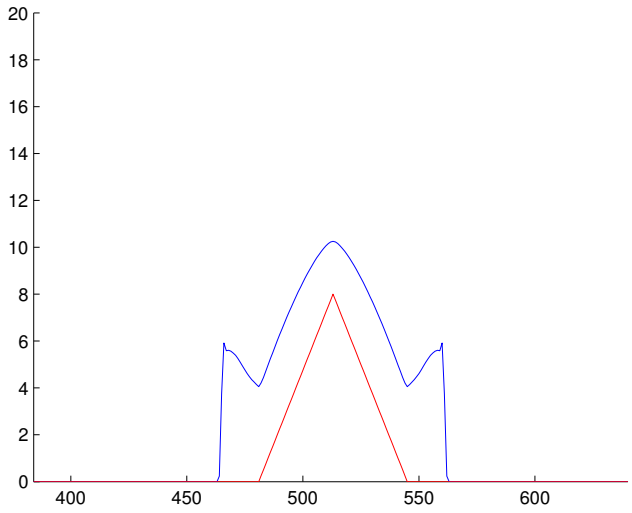

Figure 4.3: t = 3.0, Finite Volume Method. No change in the volume of fluid was observed. High frequency oscillations were visible near the contact line.

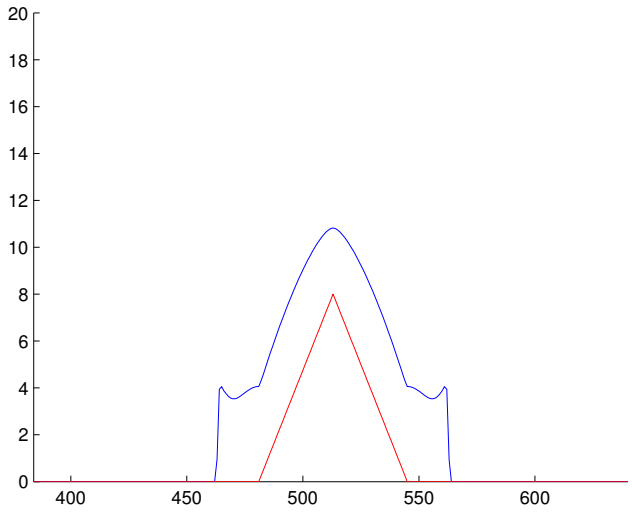

Figure 4.4: $t = 3.0$, Finite Volume Method with Semi-Lagrangian Advection. No change in the volume of fluid was observed. High frequency oscillations were less visible near the contact line.

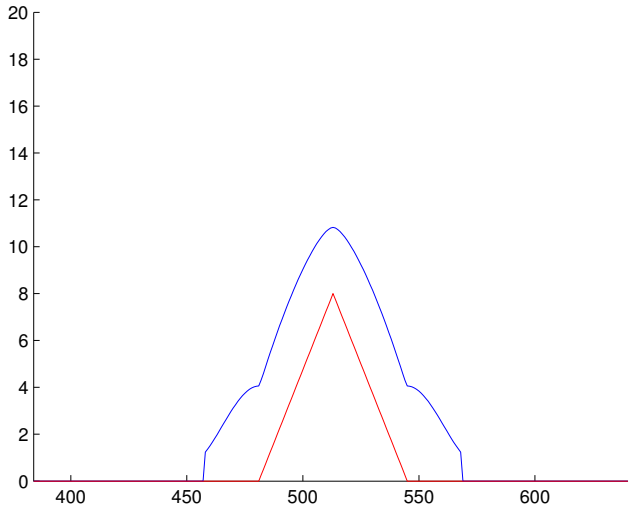

Figure 4.5: $t = 3.0$, Finite Volume Method with Semi-Lagrangian Advection and Level Set. No change in the volume of fluid was observed. No high frequency oscillations were visible near the contact line.

Bibliography

[1] Tariq D. Aslam. A partial differential equation approach to multidimensional extrapolation. *Journal of Computational Physics*, 193(1):349–355, 2004.

[2] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite difference. In *Communications on Pure and Applied Mathematics 5*, pages 243–255, 1952.

[3] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces, 2002.

[4] Nick Foster and Ronald Fedkiw. Practical animations of liquids. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 23–30. ACM Press / ACM SIGGRAPH, 2001.

[5] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flows of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, December 1965.

[6] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH 90*, pages 49–57, 1990.

[7] Anita T. Layton and Michiel van de Panne. A numerically efficient and stable algorithm for animating water waves. *The Visual Computer*, 18(1):41–53, 2002.

[8] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure, 2004.

[9] S. Osher and J. Sethian. Fronts propagating with with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[10] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2003.

[11] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1987.

[12] A. Staniforth and J. Côté. Semi–lagrangian integration schemes for atmospheric models – a review. *Monthly Weather Review*, 119(9):2206–2223, September 1991.

[13] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: The finite volume method*. Pearson Prentice Hall, 1995.

[14] Huamin Wang, Peter J. Mucha, and Greg Turk. Water drops on surfaces. *ACM Trans. Graph.*, 24(3):921–929, 2005.

[15] Hong Kai Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627.