A Fully Implicit Ghost Fluid Method on an Unstructured Mesh

by

Kyle Perline

Clarkson University

A Fully Implicit Ghost Fluid Method on an Unstructured Mesh

A Thesis Proposal by

Kyle Perline

Department of Mechanical and Aeronautical Engineering

Mentor: Dr. Brian Helenbrook

March 5, 2011

Introduction

Fluid motion is governed by some of the most basic laws of physics. These principles are encapsulated in the Navier-Stokes and continuity equations, which state Newton's three laws of motion must be obeyed and mass must be conserved. However, analytical solutions exist only for elementary textbook scenarios. As a result, the field of Computational Fluid Dynamics (CFD) is devoted to creating numerical methods that solve these equations. A subfield of CFD is specifically concerned with solutions to two-phase flow problems, which are characterized by the presence of two different immiscible fluids. These solutions include not only the velocity and pressure fields but in particular the motion of the interface separating the two fluids.

A CFD method that can accurately model two-phase flows and the interface motion has many applications. A common two-phase flow example is the waves around the hull of a moving ship and the resulting wake; the body of water and the air can be considered to be immiscible, and their interface is the water's surface [1]. A more esoteric illustration is the modeling of a star collapse, as there are regions containing gases or plasmas with different physical characteristics [2]. Another practical application is improving the efficiency of combustion engines. In a process called atomization, the liquid fuel is dispersed in the combustion chamber as small droplets. At the fuel injector tip interfacial instabilities between the fuel surface and the air determine the volume of these drops. The drops' size plays a large role in their combustion and therefore the performance of the engine overall. Therefore, a method of accurately simulating these fuel injectors and the atomization process could result in more efficient engines [3, 4].

While there are clearly uses for modeling two-phase flows, the development of an accurate method has proved challenging. The primary issue is that numerical methods are not well adapted to handing the jump in the fluids' physical parameters, such as pressure, at the interface. To cope with this problem, methods have been developed that smear out the interface over a small width. While this technique does allow one to obtain solutions to two-phase flows, this smearing reduces the accuracy. The Ghost Fluid Method (GFM) is a relatively new technique that avoids this problem. This method explicitly captures the jump conditions at the interface through the use of a ghost mesh and a Taylor series expansion of the physical parameters [3, 5]. The end result is a method that handles the discontinuities while maintaining the interface as an infinitely thin boundary between the two fluids.

All of the GFM techniques applied to date have been on structured meshes and use explicit time advancement methods [3, 6]. Structured meshes impose a severe tradeoff between the fluid interface resolution and the time required to compute the solution. The mesh size determines this resolution, where a more refined mesh yields more information. Increased detail is desired along the fluid interface in two-phase flows, but is often unnecessary elsewhere. A structured mesh has a uniform refinement, meaning the level of detail is the same everywhere, which requires that the interface is only as detailed as the whole fluid. This would not be a problem, except that a higher resolution solution takes much more time to compute. On structured meshes explicit time advancement methods are computationally cheap. Explicit methods can march the solution forward in time, where the solution at each time step is completely determined by the preceding one. To ensure accurate results the time step is limited by the

most refined area of the mesh; the more highly refined any portion is, the smaller each time step can be. In a structured mesh the refinement is uniform, and so the solution in one portion of the mesh is not slowed due to increased refinement in another. So, while these two techniques allow for an accurate solution, it is computationally expensive to obtain a detailed solution.

The goal of this thesis is to implement an accurate and computationally inexpensive two dimensional GFM. This will be achieved by implementing an implicitly advanced GFM on an unstructured mesh. An unstructured mesh will allow a coarse mesh to be used over the majority of the domain and a much finer mesh near the interface. In this way the resolution of the interface can be increased without significantly increasing the computational time. The time step in an implicit scheme, as it is not dependent upon the most refined mesh area, can be larger than that of an explicit method on an unstructured mesh. A drawback, however, is that the solution at the next time step is dependent upon both the preceding solution and itself. So, the solution at the next time step is iterated until it stops changing and reaches a steady state. This method is more complex than explicit methods but this increased computational expense is outweighed by the larger time steps. This combination of an unstructured mesh with an implicit scheme will yield a relatively computationally inexpensive method that can explicitly handle the interface on a fine local scale.

Background

The fluid flow will be analyzed using the Finite Element Method (FEM), which is a technique for numerically solving partial differential equations. In FEM the analytical equations governing fluid motion are replaced by their approximation. The function values are only determined at a finite number of points called nodes, which together form the discretization of the domain. Since differential equations can be approximated by difference quotients (the limit of a difference quotient is in fact the derivative), it is possible to generate accurate, but not exact, solutions to these equations by only using the nodal function values. While all numerical methods that approximate PDEs are created from difference quotients, they do not produce equally accurate solutions. For example, Euler's method for solving ordinary differential equations is only first order accurate, whereas the Runge-Kutta method is generally fourth order accurate.

There are many ways to discretize these PDEs. For a one dimensional model, the discretization is limited to either uniform meshes, in which all nodes are equidistant, or nonuniform meshes. In two dimensions, the nodes are generally aligned according to either a rectangular or triangular grid. Again, in both cases the distance between nodes connected by an edge can vary, called unstructured, or remain constant, called structured. These discretizations each have benefits and disadvantages. Nonuniform and unstructured meshes require a more general technique for handling difference quotients and so may be more computationally expensive, but they can be applied more easily to irregular domain shapes.

Since all finite element numerical methods are derived from the same idea, they all have the same limitation. These methods are sensitive to the function's derivative and so work best when the

nodal values approximate a continuous function. Even though the numerical methods can technically be applied to discontinuous functions, they will yield inaccurate solutions. This drawback is especially problematic when two-phase flows are being analyzed. In these flows the fluids' pressure, density, and viscosity are not continuous across the whole domain; these variables are continuous in each fluid separately, but at the interface there is a jump discontinuity. This jump, initially represented in the numerical approximation by a single large gradient between two connected nodes, cannot be maintained by numerical methods. As this flow is evolved, the magnitude of this gradient is decreased and spread over the width of multiple nodes. This smearing decreases the accuracy of the solution. [7]

The GFM is not a numerical method in the sense described above as it does not directly solve the PDEs. Instead, it removes this jump discontinuity by creating two single-phase fluids from the one two-phase fluid [5]. Each fluid in the two-phase flow is extended into a ghost mesh so that the two one-phase fluids, now comprised of both the real and ghost mesh, each cover the whole domain. By construction of this ghost mesh these two fluids only contain continuous functions without jumps; therefore, they can then both be evolved using any conventional FEM. After each time step the one-phase fluids are recombined to assemble the two-phase fluid. This recombination requires the location of the interface, but the two individual one-phase fluids, which now spread over the whole domain, no longer contain this information. So, the level-set method is used to track the interface. This method can, very roughly, be thought of as way for determining the distance between any point in the domain and the interface. Since all points on the interface have a distance of zero from themselves, the location of this zero-level contour is the interface itself.

An implementation of any GFM requires three components: a FEM that can solve one-phase flows based upon the governing equations; a level-set method that can be applied in conjunction with the FEM to track the interface; and a technique for creating these ghost fluid meshes. These components are detailed below.

Governing Equations

This model will be restricted to two-phase incompressible and viscous two dimensional flows. Therefore, the governing equations for these fluids will be the incompressible Navier-Stokes equations and the continuity equation. The Navier-Stokes equations are written compactly as

To examine them more explicitly they can be expanded out into

Here, *u* and *v* are the x and y velocity components respectively, *p* is the pressure, is the density, is the dynamic viscosity, and *f* represents body forces. Both viscosity and density are taken to be constant over the domain of a single fluid. Of course, their numerical values will likely be different in the two fluids.

The continuity equation is also required to model a fluid. This is given as

—

The term is the divergence of the product of density and velocity. This equation states that the net mass entering a given region is equivalent to the change of the enclosed mass.

Level-Set Formulations

The level set method is a technique used for tracking the interface in a two-phase fluid. A function assigns a scalar value to every point in the domain. The first fluid resides wherever this function is positive and the second fluid where it is negative. It then follows that the zero level contour of , meaning all locations where is zero, is the interface of the two fluids.

The level set function is advected using the velocity field of the fluid. This is accomplished by supposing that every particle in the fluid is assigned its own scalar value. The function can then be written

Here, *x(t)* and *y(t)* denote the location of the fluid particle at time *t*. Due to the no-slip boundary condition, the particles at the interface remain at the interface, implying that their values will always be zero. This is equivalent to stating that their level set value time rate of change is zero as well [8]. This can be written and simplified as

—

—

Even though only the particles at the interface were examined in the derivation, this formula applies to the whole fluid. is the velocity of the fluid, and so, as expected, the evolution of the level set function depends upon the fluid motion. This velocity field will be determined at each time step by the FEM that solves the one-phase flows.

Due to the approximations inherent in numerical methods, it is desirable but not necessary to have the level set function be a signed distance function from the zero level. This simply states that given an interface, and therefore the zero level of , all other values of the level set function are determined by the shortest signed distance from the point in question to the zero level. When the level set function has this characteristic more accurate results are obtained.

However, even if the level set function begins as a signed distance function it will lose this characteristic as it is evolved. Since the only relevant information contained in the level set function is its zero level contour it is acceptable to reinitialize the function. The reinitialization process maintains the same zero level but changes the scalar values at other points so that it becomes a signed distance function again. Many reinitialization techniques solve the following Hamilton-Jacobi equation in pseudo time [9,10]. Essentially, the level set function will reach steady state in pseudo time when its unit gradient is one everywhere. This function is

___

In this equation is the pseudo time variable and S can be the sign function. An unfortunate result of this process is that if it is completely solved to steady state the interface will tend to smooth out. To prevent this smoothing the above equation is only iterated a few times.

Many times it is necessary to know the normal vector to the interface. In general the unit vector in the direction of the gradient is found with

____

where is the normal vector to a given level set contour [9].

Ghost Fluid Method

The Ghost Fluid Method functions by extending each of the two fluids across the whole domain into ghost cells. At every location on the domain there will then exist two sets of data points. One data set will correspond to the real fluid at that location and contains its physical parameters of pressure, density, viscosity, and velocity. The second data set states the estimates of these same parameters of the other fluid were it to be at that location, and so has therefore been termed a ghost cell. Two single-phase flows can then be created by grouping the real and ghost cells corresponding to the same fluid together [5].

The primary concern of any GFM then is the method of estimating the ghost cells' values. This technique is based off of two observations. First, all variables within a single phase fluid must be continuous. Therefore, since two single phase fluids will be formed using the ghost cells it is required that all variables in each of these two resulting fluids are continuous across the whole domain. Also, the ghost cells' values must in some way depend upon the values of the real fluid at their respective locations. While a valid manner of computing the ghost cells' values based upon the first observation is simply repeating the values of the real fluid at the interface into the ghost cells, this would essentially decouple the two real fluids.

There are three cases that must be dealt with when extrapolating the variables. The first two are trivial and deal with the variables that are continuous across the interface and those that are specifically fluid dependent. Ghost cell values of variables that are continuous across the whole domain

are simply set to the real fluid value at the respective location. In general, the only variable continuous across the interface is the velocity. Secondly, in incompressible fluids density and viscosity are constant across each individual fluid; therefore the ghost fluid values are set as the real values from the fluid being extrapolated.

The last case deals with discontinuous variables, which are estimated using a Taylor series expansion around the interface. This Taylor series estimates the jump in the variable around the interface, and can be written as

— ——— —

In the above formula denotes the jump in the variable , the interface location is given by , and — is the jump of the derivative at the interface. By calculating this jump the ghost fluid values can be determined by the real fluid values [3]. The process of determining this jump is shown graphically in the diagram below.

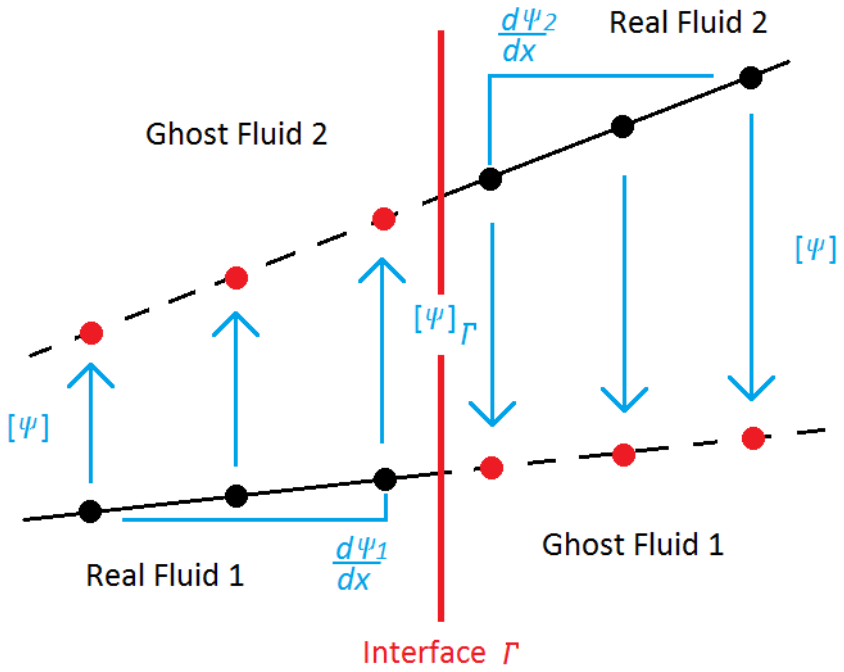

Figure 1. Determining variable jumps across the interface

In the figure above the interface is shown as the red vertical line, and the real fluid variables are black dots. Fluid 1 exists to the left of the interface and fluid 2 to the right, and so the variable jump at the interface is given as . The jump in the derivative, — , is the difference —— —— ; the numbers 1 and 2 denote which real fluid the derivative is calculated in. The variable jump at another location, , is shown as the blue line pointing from the real fluid values, and can be calculated using

the above Taylor expansion. Once this jump is known, the ghost value is found by subtracting the jump at a specific location from the corresponding real fluid value. The two single-phase fluids are then the union of the Real Fluid 1 and Ghost Fluid 1 data points and the union of the Real Fluid 2 and Ghost Fluid 2 data points. As seen in Figure 1 the variable is now continuous across these single-phase fluids.

For incompressible fluids only pressure will be discontinuous across the interface, and so only expressions for the pressure derivatives are required. The first derivative can be found with

While the second derivative is

Again, the subscripts 1 and 2 denote which fluid the pressure is evaluated in, and is the jump in the pressure at the interface. These definitions of the derivatives can then be used to determine the remaining values of the ghost nodes [3].

Once the ghost meshes have been completed the two-phase fluid can be separated into the two one-phase fluids which are solved independently. The last step in the GFM is reassembling the two-phase fluid after each time step. The level-set function is advected using the real two-phase flow velocity values, and its final shape tells where the interface is. The two-phase fluid can be constructed by selecting values from the appropriate single-phase fluid as determined by this information.

Research Methodology

The first step to creating a two-dimensional GFM on an unstructured mesh is developing a simplified model. Progress has already been made towards this goal, as an explicit ghost fluid technique has been implemented on a uniform one dimensional mesh. Instead of the Navier-Stokes equations the Euler equations were used to govern the fluid so that a more basic GFM than the one described above could be used. This base method will serve as the first building block for more complex models. All of the individual features that comprise this model can be altered and improved individually without significantly affecting the whole model. In this way increasingly complex models can be created from the previous one until it has attained or implemented all desired features.

Preliminary results demonstrate that the method does succeed in preserving the sharp interface. Figure 1 below shows how the variables , *u*, and *E*, or density, velocity, and energy, are

discontinuous across the interface at an x-position of 32. Without the use of the GFM the interface is smeared over five to ten percent of the domain, while there is clear jump discontinuity at a single location with the GFM.

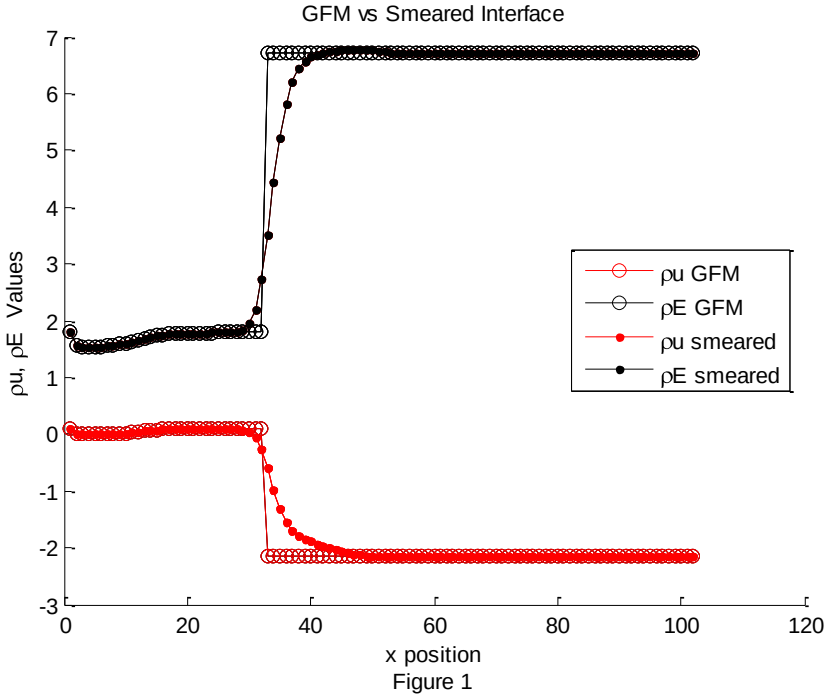

Figure 1

There are two extensions for this one-dimensional base method. The first is to model a contact discontinuity, in which the pressure and velocity are continuous across the interface but the entropy is not. This method will follow the procedure outlined in [5], in which the ghost fluid entropy values are simply copied from the real fluid values across the interface. Implementing this model requires establishing different initial conditions and changing the manner in which the variables are extrapolated. The second extension will use an artificial compressibility method, in which the fluid is governed by the Navier-Stokes and modified continuity equations instead of the Euler equations. Since this method more closely resembles the final goal of this thesis than the contact discontinuity model, an additional extension will be to then implement this on a nonuniform mesh.

These three models will complete the one-dimensional tests. A two-dimensional GFM will then be implemented within a C++ program developed by Dr. Brian Helenbrook that uses the Finite Element Method to discretize and solve the incompressible Navier-Stokes equations. The initial GFM will again be simplified by using a more basic technique of determining the ghost fluid values, similar to that in the contact discontinuity model. Due to the increased complexity of the two-dimensional method, it will also need to be more accurately checked against non-GFM methods. This will be accomplished by comparing solutions from both methods to ensure that they reasonably match. After the simplified model works correctly, the final stage will be the implementation of the more accurate procedure using the Taylor series to estimate the ghost values as described in the background section.

Lastly, the performance of the final GFM will need to be quantified. The accuracy of the solutions will be tested in two ways. The GFM solutions will be compared against the solutions yielded from a smeared-out method, and convergence rates of this method will be examined. Additionally, two comparisons will be made for the run-times required to determine the solution to a given fluids problem. These comparisons will be the GFM against a smeared-out method, and the fully implicit GFM against an explicit GFM. Gathering this performance data will conclude the thesis research.

Timeline

There are four main steps to completing this thesis, which are: completing the simplified one dimensional models; implementing an implicit GFM on an unstructured two-dimensional mesh; collecting and analyzing data; and finally writing the thesis. The timeline for the completion of these steps is shown below. The tasks shown on the right will be completed by the end of the listed month.

March, 2011	Finish 1D methods – model a contact discontinuity with the Euler equations and create an GF artificial compressibility method
April, 2011	Begin work on the two-dimensional model within the C++ program; Start writing thesis
June, 2011	Implement a GFM with a simplified ghost fluid value extrapolation technique
September, 2011	Improve the extrapolation technique by using the Taylor series expansion method
November, 2011	Test the final GFM; Begin data collection comparing the accuracy and run-times of the final model against smeared methods and other explicit GFMs
February, 2012	Complete data collection and analysis
March, 2012	Finish writing thesis

References

[1] Sussman, M.; Smith, K.; Hussaini, M.; Ohta, M. & Zhi-Wei, R. A sharp interface method for incompressible two-phase flows *Journal of Computational Physics, Elsevier,* 2007*, 221*, 469-505

[2] Benson, D. Computational methods in Lagrangian and Eulerian hydrocodes *Computer methods in Applied mechanics and Engineering, Elsevier,* 1992*, 99*, 235-394

[3] Desjardins, O.; Moureau, V. & Pitsch, H., An accurate conservative level set/ghost fluid method for simulating turbulent atomization *Journal of computational physics, Elsevier,* 2008*, 227*, 8395-8416

[4] Scardovelli, R. & Zaleski, S. Direct numerical simulation of free-surface and interfacial flow *Annual Review of Fluid Mechanics, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA,* 1999*, 31*, 567-603

[5] Fedkiw, R.; Aslam, T.; Merriman, B. & Osher, S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method) *Journal of Computational Physics, Elsevier,* 1999*, 152*, 457-492

[6]Moureau, V. & Desjardins, O. A second-order ghost-fluid method for the primary atomization of liquid fuel in air-blast type injectors *Proceedings of the Summer Program,* 2008, 143

[7] Liu, T.; Khoo, B. & Wang, C. The ghost fluid method for compressible gas-water simulation *Journal of Computational Physics, Elsevier,* 2005*, 204*, 193-221

[8] Peng, D.; Merriman, B.; Osher, S.; Zhao, H. & Kang, M. A PDE-Based Fast Local Level Set Method* 1*Journal of Computational Physics, Elsevier,* 1999*, 155*, 410-438

[9] Enright, D.; Fedkiw, R.; Ferziger, J. & Mitchell, I. A hybrid particle level set method for improved interface capturing *Journal of Computational Physics, Elsevier,* 2002*, 183*, 83-116

[10] Zhu, J. & Sethian, J. Projection methods coupled to level set interface techniques* *Journal of computational physics, Elsevier,* 1992*, 102*, 128-138