Moving Interface Problems: Methods & Applications
Tutorial Lecture II

Grétar Tryggvason
Worcester Polytechnic Institute

Moving Interface Problems and Applications in Fluid Dynamics
Singapore National University, 2007
Lecture 2:

Motivation

The One Fluid Formulation

Solving the Navier-Stokes Equations

Methods for the advection of a marker function
  Volume of Fluid (VOF)
  Level Sets
  Others methods
Front Tracking


Review

The conservation equations are solved on a regular fixed grid and the front is tracked by connected marker points.
The structure of the front
Data structure for the surface elements. The elements carry essentially all information about the structure of the front.

The points only "know" their locations.
The right data structure makes it easier to work with the interface. In 2D it is a matter of convenience, in 3D it makes the difference between an algorithm that works and one that does not!

- add and delete front objects,
- change the topology,
- handle multiple interfaces
Moving Interface Problems—Front Tracking Numerical Method

Working in barycentric coordinates simplifies the interpolations needed for the elements

\[ u + v + w = 1 \]

Quadratic interpolation

\[
p(u,v,w) = \frac{1}{2} (1 - u)(-u p_5 + (1 - v)p_3 + (1 - w)p_2) \\
+ \frac{1}{2} (1 - v)((1 - u)p_3 - v p_6 + (1 - w)p_1) \\
+ \frac{1}{2} (1 - w)((1 - u)p_2 + (1 - v)p_1 - w p_4)
\]
In two-dimensions adding or deleting a point is a relatively simple operation. We generally split an element to add a point and collapse an element to delete a point.
As the interface stretches and deforms, some parts are depleted of points while other parts become crowded by points. To maintain a nearly uniform resolution of the interface it is necessary to use **dynamic regridding**.

Regridding can be achieved by
1. Adding elements
2. Deleting elements

In 3D it is also often beneficial to
3. Reshape elements
Dynamic regridding of a buoyant bubble resolved on a 16 by 16 by 16 grid
Transferring information between the fixed grid and the front
Moving Interface Problems—Front Tracking
Numerical Method
The velocities are interpolated from the grid:

$$ \phi_l = \sum \phi_{ijk} w_{ijk} $$

The front values are distributed onto the grid by

$$ \phi_{ijk} = \sum \phi_l w_{ijk} \frac{\Delta s_l}{h^3} $$

On the front: per length
On the grid: per volume

The weights $w_{ijk}$ can be selected in several different ways
Moving Interface Problems—Front Tracking

Interpolating from grid

\[ w_{ijk}(x_p) = d(x_p - ih) \cdot d(y_p - jh) \cdot d(z_p - kh) \]

Area weighting

\[ d(r) = \begin{cases} 
  (r - ih)/h & 0 < r < h \\
  h - (r - ih))/h & -h < r < 0 \\
  0 & |r| \geq h 
\end{cases} \]

Bilinear interpolation

\[ \phi_p = A_1\phi_{i,j} + A_2\phi_{i+1,j} + A_3\phi_{i+1,j+1} + A_4\phi_{i,j+1} \]
Moving Interface Problems—Front Tracking
Interpolating from grid

Peskin’s weighting

\[ w_{ijk}(x_p) = d(x_p - ih) d(y_p - jh) d(z_p - kh) \]

\[ d(r) = \begin{cases} (1/4h)(1 + \cos(\pi r/2h)) & |r| < 2h \\ 0 & |r| \geq 2h \end{cases} \]

or

\[ d(r) = \begin{cases} d_1(r), & |r| \leq 1, \\ 1/2 - d_1(2-|r|), & 1 < |r| < 2, \\ 0, & |r| \geq 2, \end{cases} \]

where

\[ d_1(r) = \frac{3 - 2 |r| + \sqrt{1 + 4 |r| - 4r^2}}{8} \]
Several other interpolation functions are available.

For many problems, particularly for modest surface tension and material ratios, the exact form of the smoothing only has minor impact on its quality. For stiff problems, smoother is better.

Using the appropriate data structure greatly simplifies the programming and in 3D can determine the difference between success and failure.

In an actual code, it is important to always loop over the front and determine the fixed grid points from the front location, not the other way around.
Constructing the marker function
Since the fluid interface is explicitly tracked, the advection equation for the density and other material properties is not solved directly. Instead, the fields are constructed from the new location of the interface.

The simplest method is to simply set the properties directly from the location of the front. This causes a problem for interfaces that are close together.

Distribute the density gradients on a grid and integrate the gradients.
Construction of the density from the gradient

\[ \chi_{i,j} = \chi_{i-1,j} + \Delta x \left( \frac{\partial \chi}{\partial x} \right)_{i+1/2,j} \]

\[ \chi_{i,j} = \frac{1}{4} \left( \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j-1} + \chi_{i,j+1} + \right. \]

\[ \Delta x \left( \frac{\partial \chi}{\partial x} \right)_{i+1/2,j} - \Delta x \left( \frac{\partial \chi}{\partial x} \right)_{i-1/2,j} + \Delta y \left( \frac{\partial \chi}{\partial y} \right)_{i,j+1/2} - \Delta y \left( \frac{\partial \chi}{\partial y} \right)_{i,j-1/2} \]
To reconstruct the marker function from the interface it is generally sufficient to limit the iteration to a narrow band around the interface. The grid points in the band are identified by the non-zero value of the gradient and are addressed by setting up a linked list.
Moving Interface Problems—Front Tracking
Updating the Marker Function

Density gradient in x direction distributed by Peskin function.

Density gradient in y direction distributed by Peskin function.

Density field constructed by solving Poisson equation.

Density field constructed by solving Poisson equation and using area weighting.

Density field constructed from direct integration (average in x and y directions).

Density field constructed by solving Poisson equation and using Peskin function (64x64).
The amount of marker in each cell can also be constructed from the location of the interface. For a general interface location it can be difficult to do so locally, but if the overall topology of the interface is known (closed or periodic), then setting the value of the marker in all cells below the interface results in the correct value. Determining the crossing of front elements with grid lines is easy in 2D but complex in 3D!
Solution of the pressure equation
The solution of the pressure equation is the most expensive part of the simulations.

The solution of the pressure equation can be difficult when the density ratio is high.

Multigrid methods, in particular, can diverge but other iterative methods seem (BICGSTAB, for example) seem to do well.

SOR will always converge (if the density is well behaved) but the cost is very high.

For many problems, the influence of using smaller density ratios is small.
Moving Interface Problems—Front Tracking
Pressure Solution
Computing Surface Tension
For the solution of the Navier-Stokes equations, we need the net force on each front element:

\[ \kappa n = \frac{\partial s}{\partial s} \]

\[ \delta F = \int_{\Delta S} \sigma \kappa n ds \]

\[ = \int_{\Delta S} \sigma \frac{\partial s}{\partial s} ds \]

\[ = \sigma (s_2 - s_1) \]

\[ \delta F = \sigma \int_{\delta A} \kappa n dA \]

\[ = \sigma \int_{\delta A} (n \times \nabla) \times n dA \]

\[ = \sigma \int_S s \times n ds \]
The net force can be computed at either the element centers or the points.

For point based scheme:

\[ n_e = \frac{\Delta x_{12} \times \Delta x_{13}}{|\Delta x_{12} \times \Delta x_{13}|} \]

\[ \delta f_1 = \frac{1}{2} n \times \Delta x_{23} \]
Computing Surface Tension in 2D—Accuracy test

Location of 40 points

Radius of Curvature

40 points  80 points
Parasitic Currents

The regular grid induces a small anisotropy. These currents are typically small in immersed boundary methods.
Surface tension method of Shin and Juric (2005)

\[ f_{i,j} = (\sigma \kappa)_{i,j} \nabla I_{i,j} \]

To find the curvature we dot the surface tension with the force found earlier

\[ \nabla I_{i,j} \cdot f_{i,j} = (\sigma \kappa)_{i,j} \nabla I_{i,j} \cdot \nabla I_{i,j} \]

Or, solving for the curvature:

\[ (\sigma \kappa)_{i,j} = \frac{\nabla I_{i,j} \cdot f_{i,j}}{\nabla I_{i,j} \cdot \nabla I_{i,j}} \]

Surface tension method of Shin and Juric (2005)

\[ Ca = \frac{\mu U_{\text{max}}}{\sigma} \quad La = \frac{\rho D}{\mu^2} \]
It should be possible to extend the Shin and Juric method to any weights such that the curvature at grid points around the front we have

$$\left( \sigma \kappa \right)_{i,j} = \frac{\sum_l w_{i,j} \left( \sigma \kappa \Delta s \right)^l}{\sum_l w_{i,j} \Delta s^l}$$

The surface tension is then found as before

$$f_{i,j} = \left( \sigma \kappa \right)_{i,j} \nabla I_{i,j}$$
Accuracy tests
Moving Interface Problems—Front Tracking

Validations

- Comparison with analytical results: linear inviscid theories for oscillating drops, Kelvin-Helmholtz instability, and surface waves; Sangani’s results for regular arrays of drops in Stokes flow solutions for parallel flows

- Comparison with other computations such as the steady state shapes of drops and bubbles computed by Leal et al

- Grid refinement studies and comparison between different implementation of the method

- Comparison with experimental results (Law et al, for example)
Velocity field given:
\[ u(x, y) = \cos x \times \sin y \]
\[ v(x, y) = -\sin x \times \cos y \]
\(\pi \times \pi\) domain, resolved by 32 \(\times\) 32 grid points. Velocity interpolated by area weighting.
Second order predictor-corrector time integration.

\[ dt=0.025 \]
Moving Interface Problems—Front Tracking Validations

High Viscosity

Low Viscosity

Exact
64 points
32 Points
16 points
Growth of KH Instability

For perturbations of the form:

\[ A = A_0 e^{s t + i k x} \]

The growth rate is given by

\[ s = -i k \frac{\rho_1 U_1 + \rho_2 U_2}{\rho_1 + \rho_2} \pm \sqrt{\frac{k^2 \rho_1 \rho_2 (U_1 - U_2)^2}{(\rho_1 + \rho_2)^2} - \frac{T k^3}{\rho_1 + \rho_2}} \]

Defining

\[ We = \frac{\rho_2 \Delta U^2}{Tk} \]
\[ \Delta U = U_2 - U_1 \]
\[ r = \rho_1 / \rho_2 \]

it can be shown that the perturbation is unstable if

\[ We < 1 + r \]

and for given physical parameters, the wavelength determined by

\[ We = \frac{3}{2} (1 + r) \]

is the fastest growing one
Oscillations of an axisymmetric drop. Comparison with theoretical solutions for oscillation frequency and viscous decay.
Moving Interface Problems—Front Tracking Validations

Grid refinement

8x24  16x48  32x96  64x192
Other considerations
Although methods based on the one fluid formulation have been used successfully for many problems, several challenges remain. These are slowly being eliminated

- High Reynolds numbers: high order advection methods and non-conservative form of the advection terms
- Continuity of the viscous stresses: interpolation using the harmonic mean
- Solution of the pressure equation/slow convergence at high density ratios: More advanced fast solver
- Parasitic currents: increased smoothing helps—or separate computations of the curvature and the normal
The method has been implemented in a fully parallel code. The fluid equations are solved by a simple domain decomposition. For the front, two strategies have been used. In one the front is distributed to the different processors using a master-slave technique. In the other, the front is processed using a separate processor.
Use patches of finer grids in regions of high shear and around the front. The patches are moved as needed.

Changing the interface topology for 3D flow
Merge the corner points of close elements

Pair up the corner points of close elements and move both points to the average location of the point pair

Set up a temporary linked where one point of a point pair (the one with the higher index) points to the other point. The pointer for all other points is zero.

Loop over elements and re-link those corner points that are dual points.

Remove double elements

Using the linked list to elements that should be merged, identify elements with the same corner points

Delete those elements

Remove points without an element

Loop over elements and set a flag for those points that the elements are connected to

Loop over the points and delete points whose flag has not been set.
Two examples of a simulation of topology changes on a relatively coarse grid. Pinching of a thin filament above and the merging of a thin film to the right.
The accurate modeling of the rupture of thin films and the pinching of thin threads can be critical for many multiphase flow simulations. Explicit tracking of an interface allows the incorporation of subgrid models for the draining of the films and non-continuum effects, but the difficulties of incorporating topology changes have generally been considered one of the major drawbacks of front tracking methods. A recently developed method overcomes this limitation. Simulations of a head-on collision are shown below and an off-axis collision is shown to the right.

Simulations by S. Mortazavi and G. Tryggvason
The front tracking method described here has the right combination of simplicity and accuracy to allow simulation of fairly complex multiphase flow problems.

The control over topology changes provides the user with the ability to either include or exclude such changes.

Generally, explicit tracking provides more accurate results for the same resolution than marker function methods, but at the cost of slightly more complexity.

The presence of separate computational elements to track the front allows for many extensions and improvements