

# Report fulfilling the capstone requirement

Dynamics of drop coalescence with downward velocity at liquid interfaces

Author: \_\_\_\_\_

Faculty Advisor: \_\_\_\_\_

Reading Committee: \_\_\_\_\_

Graduate Coordinator: \_\_\_\_\_

Date: \_\_\_\_\_

# Dynamics of drop coalescence with downward velocity at liquid interfaces

David Hambley, Graduate Student in Applied Mathematics, School of Natural Sciences, University of California Merced

Research Advisor: François Blanchette, Assistant Professor, School of Natural Sciences, University of California Merced

1. Set Up. This research is an extension of work done by François Blanchette and Terry P. Bigioni, who looked at the conditions for partial coalescence of drops into an underlying reservoir of the same fluid<sup>1,2</sup>. Partial coalescence occurs when the merging process is incomplete due to the “pinching off” of a portion of the original drop, leaving behind a smaller “daughter” droplet. This can be pictured as using an eyedropper or pipette to place a drop of water onto the surface of a pool of water. In the process of the water drop coalescing into the pool, it may appear to jump up, and pinch off as a smaller droplet.

In the research by Blanchette & Bigioni, the model begins with a stationary drop at rest on the reservoir surface. In the computer simulations done by Blanchette, the simulations begin with the suspended drop and the underlying reservoir joined to form a single interface. The effort here is to study the influence on the coalescence outcome of introducing a non-zero impact velocity to the drop. The adaptation of the physical example with the water drop would be to release the drop above the pool surface, rather than placing it directly onto the surface. Our research was conducted by modifying the simulation computer code to begin with two separate interfaces, the drop and the reservoir, separated by a height measured in drop radii. In the modified simulation, the drop is then released and attains a downward velocity due to simulated gravity. Merging criteria based on the proximity of the drop to the reservoir was introduced to initiate coalescence.

For the drop beginning at rest on the reservoir surface, the relevant physical parameters are drop size,  $R$ , viscosity,  $\mu$ , and density,  $\rho$ , of the drop and reservoir, surface tension between the fluids,  $\sigma$ , and the force of gravity,  $g$ . Less relevant are the viscosity and density of the lighter surrounding fluid. The following are the important non-dimensional parameters. The Bond number,  $Bo = g\rho R^2/\sigma$ , is a non-dimensionalized weight comparing gravitational and surface forces. The Ohnesorge number,  $Oh = \mu/\sqrt{\rho\sigma R}$ , is a non-dimensionalized viscosity that is a ratio of viscous and surface forces, and a critical Ohnesorge number is a value above which pinch-off does not occur. We refer to the drop and reservoir fluid as the inner fluid, and the surrounding fluid as the outer fluid. We used the square root of the Weber number as a non-dimensionalized vertical impact velocity,  $u' = \sqrt{We} = u\sqrt{\rho R/\sigma}$ , where  $u$  is the component of velocity in the direction of gravity, i.e., downward. Blanchette & Bigioni found a negative correlation between the Bond number and the critical Ohnesorge. Our research looks at the possible correlation between the impact velocity and the critical Ohnesorge number, for a fixed Bond number. Our parameter for varying the Ohnesorge number is  $\mu$ , keeping the ratio of the inner and outer fluid viscosities constant. To vary  $u'$ , we changed the initial height of the drop above the reservoir,  $h$ , in units of drop radii.

2. Equations. The equations to be solved in this pursuit are the same as those solved in the underlying research of partial coalescence. This involves solving the Navier-Stokes equations (2.1), including surface tension as a force localized on the interfaces, to track the time-evolution of the drop and fluid surface shapes, as well as the pressure and velocity fields in the system.

The time derivative of the interface position must match the velocity field there (2.2). We assume incompressibility for both the studied fluid and the surrounding fluid (2.3), and isothermal conditions<sup>2,3</sup>.

$$\rho \vec{v}_t + \rho(\vec{v} \cdot \nabla) \vec{v} = -\vec{\nabla} P + \nabla \cdot (\mu \nabla \vec{v}) + \sigma \kappa' \delta_S \hat{n} + CB_0 \hat{g} \quad (2.1)$$

$$S_t = \vec{v} \quad (2.2)$$

$$\vec{\nabla} \cdot \vec{v} = 0. \quad (2.3)$$

In these equations,  $\vec{v}$  is the velocity field;  $P$  is the pressure field;  $\kappa' = \kappa R$  is the non-dimensionalized interface curvature, where  $\kappa$  is the physical interface curvature;  $\delta_S$  is a Dirac delta function that effectively limits the term to be non-zero only at the interface surface;  $\hat{n}$  is a local unit vector normal to the interface surface, with direction from the inner fluid to the outer fluid;  $C$  is a function of the volume of inner fluid, such that  $C = 1$  in the inner fluid and  $C = 0$  in the outer fluid;  $\hat{g}$  is a unit vector in the direction of gravity;  $S$  denotes the position of the interface; and the  $t$  subscript denotes a time derivative.

For initial conditions we have,

$$\vec{v} = 0 \quad (2.4)$$

$$P_i = c_i \quad (2.5)$$

$$P_o = 0, \quad (2.6)$$

so the velocity field is set to zero (2.4), and the pressure field is uniform, with a specific value for the inner fluid (2.5), and zero for the outer fluid (2.6).

At the solid boundaries,

$$\vec{v} = 0 \quad (2.7)$$

$$\frac{\partial P}{\partial n} = 0, \quad (2.8)$$

so we have a Dirichlet condition for velocity and a Neumann condition for pressure. This is actually not very relevant in the simulation, because the domain is sufficiently large for the boundary's effect on the drop and its interaction with the reservoir surface to be small.

Due to axial symmetry, the simulation domain (see Figure 1) is rectangular, pictured as a portion of the right half of a plane through the center of the drop. The fields and interfaces are tracked using cylindrical coordinates, allowing the equations above to be solved in cylindrical coordinates with no  $\theta$ , i.e., axial, dependence.

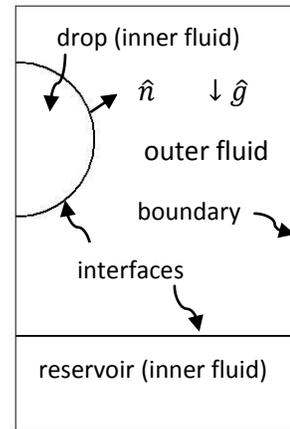


Figure 1

3. Overview of Code. The code is the SURFER fluid interface simulation program Copyright © 2001 Stephane Zaleski and others, adapted by Francois Blanchette in 2004, and further adapted for our research by David Hambley under advisement of Francois Blanchette in 2007.

Program flow begins with the initialization of global variables and parameters; it then cycles through timesteps, tracking the position of the interfaces on a staggered grid through drop descent, merging and possible pinch off; and terminates when pinch off occurs, when pinch off clearly fails to occur, or when the maximum number of timesteps is reached. In each time step, first the interface, which is discretized as markers, is advanced using the prior timestep velocity field,  $u^n$ , by Euler's method. The interpolating arcs between markers are then determined using cubic splines, and the markers for each interface are adjusted so that they are spaced with equal arclengths between them. Volume fractions, stored in the  $C$  array, are determined for those cells that the interface now passes through, with values assigned between 0 and 1. Next the pressure and velocity fields are updated. Briefly,  $u^n$  is used to generate an approximate new velocity field,  $u^* = u^n + h[\mu\nabla^2\vec{v} + \sigma\kappa'\delta_s\hat{n} + CBo\hat{g}]$  where  $h$  is the time step, and the expression  $[\mu\nabla^2\vec{v} + \sigma\kappa'\delta_s\hat{n} + CBo\hat{g}]$  is an approximation of the acceleration field; then  $u^*$  is used in an iterative process with a varying grid size, called multigrid, to determine the pressure field,  $P^{n+1}$  such that  $\nabla^2 P^{n+1} = \nabla \cdot u^*$ . Once  $P^{n+1}$  has been determined, the new divergence free velocity field,  $u^{n+1}$ , is found.

One area of concern is that the density and viscosity of the outer fluid in the simulation is significantly greater than that of air. For reasons of numerical convergence, it is faster to use a relatively small ratio of density and viscosity between the inner fluid and outer fluid. For partial coalescence from rest, Blanchette found that this has no detectable consequence. For the falling drop, it is possible that the results would be significantly different with heavy air as the outer fluid. The air likely would not exhibit the same downward force on the reservoir during the descent of the drop, leading to less deformation of the reservoir interface and an earlier merge time.

Several parameters are fixed prior to each run. These include: drop radius, typically 0.0125 cm, corresponding to a dimensionless radius of 1; inner fluid viscosity in the range 0.001 to 0.015 g/cm · s, corresponding to the dimensionless Bond number of 0.0050; inner fluid density of 0.8 g/cm<sup>3</sup>; outer fluid viscosity and density, held at 1/10<sup>th</sup> the values for the inner fluid; gravity of 981.0 cm/s<sup>2</sup>, corresponding to the dimensionless Ohnesorge number in the range 0.002 to 0.030; height of drop above reservoir in the range 0.1 to 3.0 drop radii; and surface tension of 22.0 g/s<sup>2</sup>.

There are two forms of output: a text file recording various statistics of the simulation, and a data file. The text file includes the grid resolution, the inner fluid viscosity and density, the height of the drop above the reservoir, the Ohnesorge number, the Bond number, the time

merging occurs and the maximum vertical velocity prior to merging. If pinch-off occurs, the text file also includes the time of pinch-off. The data from the text file was recorded in a spreadsheet to keep statistics on when pinch-off did or did not occur for various parameter values. This data was used to determine the correlation between maximum vertical velocity prior to merging and the critical Ohnesorge number. The data file includes periodic snapshots of the velocity, volume fraction (i.e., concentration) and pressure fields. This data was viewed graphically in Matlab in various ways, primarily to see whether anomalous behavior was observed. The data file was also used in the event of a restart of the simulation.

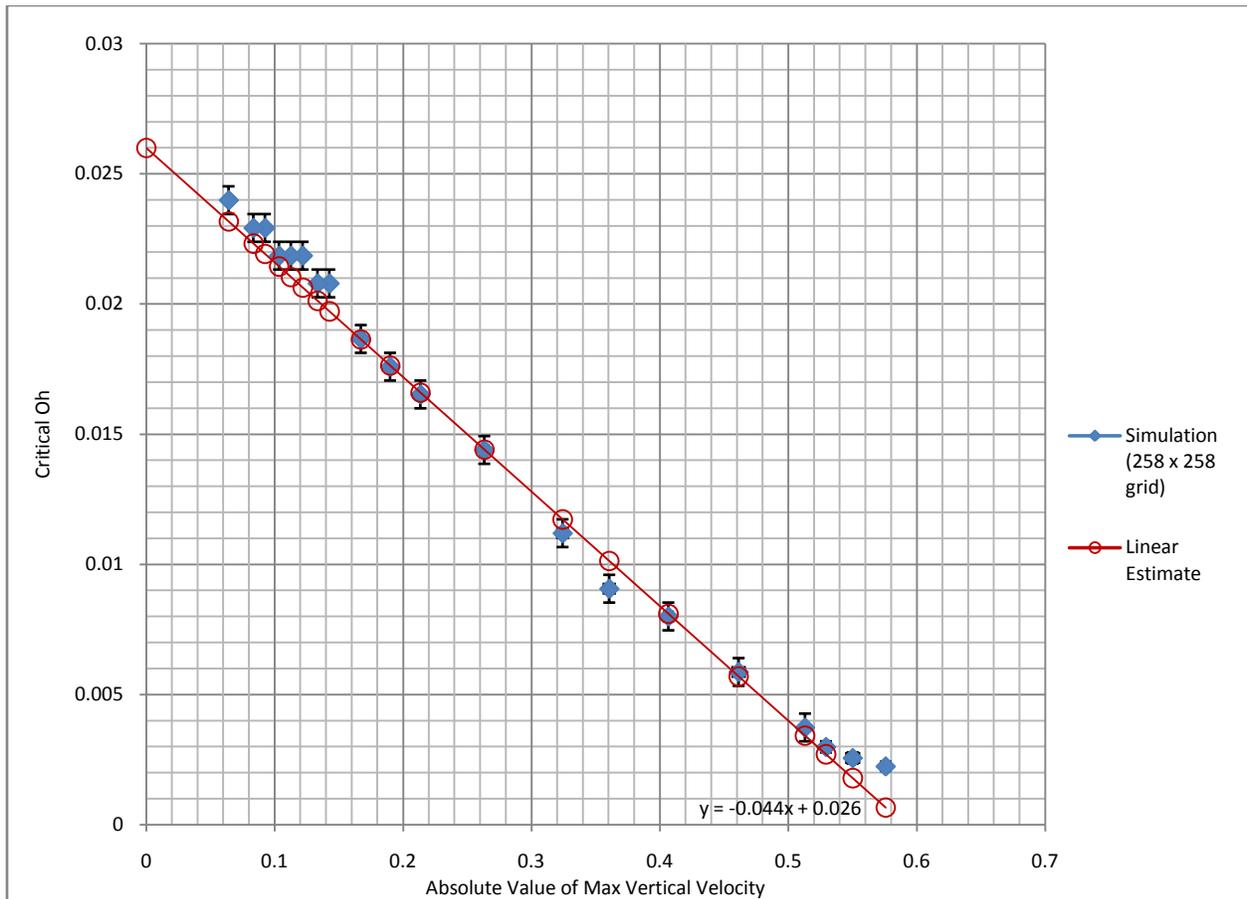
4. Changes to Code and Results. Several changes were made to the code for this research. The primary change was to create two separate interfaces at initialization, the drop and the reservoir. Parameters were added to allow for control of the initial height of the drop above the reservoir, and to track the average vertical velocity of the drop and the downward displacement of the reservoir as the drop descends. Tests were added at each timestep both for merge initiation and for success or failure to pinch-off, which leads to early termination of the simulation. Changes were also made to the statistics output, including the vertical drop velocity prior to merging.

Merging is initiated in the code based on proximity of the drop to the interface. This occurs when the lowest point on the drop interface is within a set number of grid point spacings, generally two, of the reservoir interface. At this point the drop and reservoir interfaces merge to create a single interface. Pinch-off success or failure is determined by monitoring the width of the neck connecting the drop to the reservoir. The neck initially becomes thicker as the drop continues to descend. But with the upward force of the capillary waves pulling on the drop's summit, the neck then becomes thinner. If the neck thickness drops below a set number of grid point spacings, generally two, then pinch-off is declared. If the neck begins to thicken once again, pinch-off fails to occur.

Due to longer program duration, a few changes were added for efficiency, such as the early termination on failed merge mentioned above. A method of neglecting the minimal changes that would occur a short distance from the central axis of the drop during the initial descent of the drop was implemented, to reduce irrelevant calculations. Also, a program restart, using saved data for initialization, was modified to work with the falling drop.

Ideally, the simulation would occur in an unbounded environment. This is approximated by placing the vertical boundary three radii distance from central vertical axis. The effect of rebounding waves at the drop interface is negligible for the merging to pinch-off time frame. For the efficiency method introduced for longer drop periods, mentioned above, calculations in the fluids is ignored outside 1.5 radii distance from the central axis, until the drop is relatively close to the reservoir, but well before merging could occur. This distance is a multiple of the minimal distance for merging, generally by a factor of 10.

We attempted to establish what correlation exists between maximum drop velocity prior to merging and the Ohnesorge number for the system, to establish what the critical Ohnesorge numbers for pinch-off are at different velocities. In general, we found a roughly linear negative correlation between the velocity and the critical Ohnesorge number (see figure 2, below). We ran simulations at three different mesh sizes, to verify whether the critical Ohnesorge numbers changed at higher resolutions. We found that in each case the critical Ohnesorge numbers were higher at higher grid resolutions for a given velocity, but the roughly linear negative correlation between the velocity and the critical Ohnesorge number held steady. The behavior at more extreme cases, as either the velocity or the critical Ohnesorge number approached zero, was more difficult to capture with confidence.



**Figure 2. Simulations with Bond number  $Bo = 0.05$ , showing dependence of  $Oh^*$  on the maximum vertical velocity of drop prior to merging,  $u$  with the critical Ohnesorge number,  $Oh^*$  (expressed as a range), along with a linear approximation of  $Oh^* = -0.044u' + 0.026$ , derived from the data. Simulations were performed with a grid resolution of 258x258 for the partial half plane, as pictured in Figure 1.**

At the 258x258 grid resolution for the partial half plane (as pictured in Figure 1), the linear estimate of the critical Ohnesorge number corresponding to zero velocity is in agreement with the value of  $0.026 \pm 0.001$  found by Blanchette and Bigoni for merging initiated with the drop in a resting position on the reservoir<sup>1</sup>.

5. Conclusions. There is a strong negative correlation between the maximum downward velocity of the drop prior to merging and the critical Ohnesorge number for pinch-off. The correlation is nearly linear, corresponding in a 258x258 grid to the linear equation  $Oh^* = -0.044u' + 0.026$ .

The results for different grid resolutions follow roughly the same negative correlation slope, but with higher critical Ohnesorge numbers for higher resolutions for a given maximum drop velocity. For a 514x514 grid, the linear approximation was  $Oh^* = -0.041u' + 0.030$ . Simulations on a 1026x1026 grid were difficult due to extremely long run times and the results were inconclusive, but it appeared that the critical Ohnesorge number was higher for a given velocity than was found with the lower resolution simulations. Further research would be needed to determine whether the results would converge as the resolution continued to increase. However, the strong linear negative correlation between maximum vertical velocity prior to merging and the critical Ohnesorge number should be considered reliable, in my opinion. Another avenue for future research would be to use a VOF (volume of fluid) algorithm instead of the MAC (marker and cell) algorithm, to compare the results. The VOF algorithm has the advantage of better controlling dissipative effects.

6. Appendix – Technical Guide to Code. Listed below are the primary functions in the Manypinch.c code and the purposes they serve. No functions were created or deleted for the code changes implemented in this research.
- main** – This is the central control function, standard in c language programs. Its principal purpose is the gross control of program flow, done by calling other primary functions. Some parameter initialization occurs here. At the end of the simulation, this function outputs data to the text file relating to pinch-off success or failure and the maximum velocity prior to merge.
  - initialize** – This function coordinates the initialization process, calling several other functions to help complete this task. The initialization that occurs is either for a new simulation or for the restart of a previously started simulation, based on the value of a flag. If the simulation is a restart, this function calls **iniget** to load data from the prior simulation.
  - iniget** – This function is called during the initialization process when the simulation is a restart of a prior simulation. The data file related to the prior simulation is accessed, and the function loads the velocity, pressure and concentration fields. Restart is only configured to restart simulations prior to merging, so the loaded data is assumed to have two interfaces, i.e., the drop and reservoir and still separate. The interfaces are reconstructed based on separate data files. This is the only function that is particular to the restart process.
  - initmarker** – This function is called from the **initialize** function, and it initializes the marker positions for the interfaces. One of the distinguishing features of the SURFER software code is the ability to accurately track interfaces between inner and outer fluids. Each interface exists as a series of marker positions, connected by arcs, superimposed on the grid.

- e. **creatdisk** – This function is also called from the **initialize** function, and sets the initial conditions. It sets the velocity field,  $\vec{v}$ , to zero at all grid points (equation 2.4); it sets the concentration to one for all grid points interior to the drop and reservoir (inner fluid); it sets the concentration to zero for all grid points in the outer fluid; it sets the pressure to  $2\sigma/R$  (equation 2.5) for all grid points interior to the drop (inner fluid); and it sets the pressure to zero for all grid points in the outer fluid (equation 2.6) and reservoir (inner fluid). Technically, the pressure in the reservoir should not be initialized as zero, but the pressure field is corrected in the first time step anyway, so the initialization is simplified for efficiency.
- f. **makebcf** – This function enforces the boundary conditions on the velocity field.
- g. **makebccf** – This function enforces the boundary conditions on the concentration field.
- h. **makebcpf** – This function enforces the boundary conditions on the pressure field.
- i. **makebcsf** – This function enforces the boundary conditions on the stresses field.
- j. **makebcresf** - This function enforces the boundary conditions on the residue field.
- k. **timestep** – This function governs the simulation process through each timestep, primarily calling other functions. In more detail, it calls the **findninf** function to compute the maximum velocity over the domain; it tests the CFL condition; it calls the **markers** function to advance the marker positions for the current timestep; it calls functions **initial\_arc\_length**, **splines**, and **arc\_length**, which collectively adjust the arcs between the markers; calls function **intersec** to compute intersections between interfaces and the grid; calls function **volumefraction** to compute volume fraction of inner fluids in cells intersecting the interface; calls function **outputmarkers** to output marker positions and spline coefficients to file; calls function **makebccf** to enforce concentration boundary conditions; calls the function **pressurecorrection** to correct the pressure field along the interfaces; calls the **momentum** function to update the velocity and pressure fields; calls function **energy** to compute an energy budget for the flow; calls function **pinching** to determine whether merging or pinch-off has occurred; and calls function **redistribute** to relocate interface markers at equal distances.
- l. **pinching** – This function determines whether merging or pinch-off occur, and governs the changes required to reduce or increase the number of interfaces.
- m. **momentum** – This function either calculates or coordinates the calculation of the velocity and pressure fields for each time step. In more detail, it computes viscous stresses; calls various small functions to enforce boundary conditions; calls another function to update an average density field; directly computes the viscous term, surface tension and correction to the pressure gradient for the linear portion of equation 2.1; computes the divergence present; calls another function, **mglin**, to coordinate computation of the pressure term; and, finally, the updated velocity field is calculated.
- n. **mglin** – This function coordinates the calculation of the pressure term at each time step, through computations and calls to the **relax** function.
- o. **pressurecorrection** – This function corrects the pressure field along the interfaces at each time step. It is called directly from the **timestep** function prior to the call to the **momentum** function.

- p. **relax** – This function controls the multigrid process, during which time the pressure field is updated for each time step. Multiple calls are made to this function from **mglin** each timestep.
- q. **printhist** – This function outputs to the text file the parameters as fixed after initialization and before the first time step. Additional data is output to the text file from the **main** function at the end of the simulation.
- r. **output\_velo** – This function writes to the data file. It outputs the velocity, pressure and concentration fields. It is called before the first time step; periodically for a set number of timesteps, e.g., after 1000 timesteps; at merging; and prior to most program terminations, including pinch-off success or failure.

## References

1. Blanchette, F. & Bigioni, T. P. 2006 Partial coalescence of drops at liquid interfaces. *Nat. Phys.* **2** (4), 254–257.
2. Blanchette, F. & Bigioni, T. P. 2009 Dynamics of drop coalescence at fluid interfaces. *J. Fluid Mech.* **620**, 333–352.
3. Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S. & Zanetti, G. 1994 Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comp. Phys.* **113**, 134–147.