Sandia National Laboratories

Operated for the U.S. Department of Energy by
Sandia Corporation

Albuquerque, New Mexico 87185- 0834

date: August 10, 2005

to: Distribution

from: T.A. Baer, MS 0834, Org. 9114; Randy Schunk, MS 0834, 9114;

subject: Modeling wetting contact and dewetting Phenomena with Goma level-set capability (GT-029.1)

keywords:

Level Set, Interface Tracking, Wetting, dewetting, contact angles

input records:

Level Set Interface Tracking, Level Set Length Scale, VELO_SLIP_LS, FILL_CA, WETTING_SPEED_LINEAR, WETTING_SPEED_BLAKE, LS_ADC

Introduction

Two of the most prevalent and fundamental phenomena in capillary hydrodynamic applications is fluid/fluid contact and fluid/solid contact. More specifically, the physics underpinning liquid/solid contact and wetting, dewetting, and liquid/liquid coalescence require sub-grid scale phenomena to be modeled, which poses many challenges for level-set and ALE algorithms. This memo provides a tutorial for level-set approaches to such events. Specifically, we address the best set of boundary conditions to apply towards liquid/solid contact (displacing gas), gas/solid contact (dewetting), and wetting and spreading (viz. contact angle application and surface tension forces).

This tutorial assumes that the reader is experienced with level-set technology in Goma and experienced in modeling wetting and spreading phenomena. The tutorials GT-001.4 and GT-020.3 should be consulted if needed before attempting to understand the features and algorithms described herein.

Theory: Impact, thin films, wetting and spreading phenomena with Level-Set Embedded Interface Tracking

As a quick review, the topic addressed in this memo regards the potential interaction of phases (three phases to be exact, viz. solid, liquid, and gas) and how to accommodate these interactions with level-set-based approaches. For purposes of this tutorial, we model the location/separation interface between these phases with analytical geometry and a smooth level-set function $F(x,y)$ which is related to the interface location by the fact that the interface is the curve (or surface for three dimensional problems) in space that solves:

$$F(x, y) = 0 \quad \text{(EQ 1)}$$

The level set function, *F*, will evolve over time according to a kinematic equation, known as the FILL equation, but it is always required that its zero level contour will be the location of the interface. This is the nature of the level set abstraction of the interface: the explicit location of the one-dimensional interface is replaced by finding the zero contour of a two-dimensional function. This is done as needed by solution of Eq. (1).

The level set function, *F*, is a smooth function in the sense that its gradient in a finite region around its zero contour is continuous. This is an important feature because, among other things, the normal vectors to the interface and its curvature are thus well defined and readily determined.

In fact, these geometric quantities are critical to modeling impact, wetting, dewetting and spreading events. Specifically, the contact angle that characterizes geometrically how liquid/gas/solid phases meet at a three phase line is often taken as a thermodynamic property of the three phase system (e.g. Young's equation). Hence, this angle will clearly be an important parameter as it basically embodies the level of "affinity" a liquid phase has for a solid (through surface energy arguments like surface tension). When hydrodynamic forces are brought to bear, the contact angle in such a system does not necessarily take on the static value. For purposes of this tutorial, we will assume that a moving contact angle will take on a different value than the static contact angle, and that the relationship between the two hinges on the velocity of the contact line (viz. the so-called Blake-De Coninck view point. cf. T. D. Blake and **J.** De Coninck, "The influence of solid-liquid interactions on dynamic wetting," *Advances in Colloid and Interface Science* 96, 21(2002)).

The force at and near a three-phase contact line and also in thin film regions involve much more than that which the Navier-Stokes system can accommodate. Specifically, when films (e.g. a liquid film bounded by a solid and a gas, or a gas film bounded by a liquid and a solid) are thinner than about 100 nm, a significant chemical potential difference is generated by molecular forces. This difference has come to be known as "disjoining pressure". Of the several components that make up this pressure, that governed by London-Van der Waals forces are the most dominant (see Deryagin, B. V., Proc. Second Intern. Congr. Surface Activity, London, 2, 153 (1957)). The functional form for this force (pressure) is as follows:

$$\Pi \equiv \frac{A}{h^3}$$

where the constant *A* is related to the Hammaker constant, and *h* is the film thickness. One immediate issue is noteworthy, with respect to a level-set free surface algorithm, and that is that this pressure is to be applied to the gas-liquid phase interface, and its strength depends on the location and properties of a nearby solid/liquid or solid/gas interface. Hence, this condition is non-local, and its implementation is difficult. That

said, the functional form suggests that it is a "run-away" singular pressure, and if the sign is such that there is a propensity to "dewet" (viz. a conjoining force), then the event will be more apt to happen the thinner the film and if there are perturbations or gradient in the film thickness. These concepts are important to the LS_ADC boundary condition we implemented in order to model this phenomenon.

If a three phase contact line exists, conjoining/disjoining pressure is still active near that line, but the forces (not completely unrelated to this pressure) are often dominated by surface tension. Of course the nature of the balance depends on many factors, e.g. , whether the contact line is moving under the influence of macroscopic flow, whether the liquid phase is wetting or dewetting relative to the solid (viz. Young's equation), etc. One very important factor from a modeling and simulation standpoint is the length scale over which the user is attempting to resolve this force, and the length scale over which wetting events are involved in the problem. This scale is measured best by the relative importance of capillary forces to other hydrodynamic forces, like inertia (Weber number) or viscosity (Capillary number). In any case, the mesh rarely picks up all of the important regimes of wetting, from molecular to continuum, and so wetting models are required.

Another phenomena that must be accounted for at contact line-level is fluid slip. Continuum mechanics does not allow for contact line motion at any contact angle other than 180 degrees due to a stress singularity that arises due to the demand that mass be conserved. Mass conservation at the dynamic contact line is realized by adhering to the free surface kinematic boundary condition, but the only way to do that is for the fluid velocity to be zero at the line, or for the contact angle to be 180 degrees. This is known as the kinematic paradox. Moreover, relaxing the no-slip condition at the contact line is necessary to avoid infinite stresses there. In practice the coarseness of the mesh builds in some relief, but it is necessary to add more slip for proper contact line motion. This is clearly a subgrid event, as some molecular events equivalent to macroscopic slip are obviously occurring at and near the contact line.

In summary, fluid/solid interactions can encompass many events, and we would like to be able to capture them all with Goma's level set technology. Towards this end, the following boundary conditions have been developed:

FILL_CA - Used to specify a force at the contact line that is related to the difference between the apparent contact angle and a specified static contact angle. The force is scaled by the surface tension value.

WETTING_SPEED_LINEAR - Used to specify a contact angle and wetting-line speed relationship (cf. Baer et al., "A Finite Element Method for Free-Surface Flows of Incompressible Fluids in Three Dimensions, Part II: Dynamic wetting lines", *IJNMF*, 2000.)

WETTING_SPEED_BLAKE - Used to specify a contact-angle relationship with contact line velocity according to the functional form proposed by Blake and De Coninck (reference above).

VELO_SLIP_LS - Used to specify a slip velocity in gas and liquid around a dynamic contact line. Also used to specify pre-impact slip along a solid substrate during impending impact of liquid on the solid (displacing gas).

LS_ADC - Used to specify a contact/dewetting event (probabilistic) based on the dynamics of the process and the "thinness" of the film. This is a "finger-of-God" boundary condition that steps outside of the continuum mechanics solver and manipulates the level set field directly to initiate contact/dewetting.

Boundary Condition Usage Guide

BC = VELO_SLIP_LS SS # {slip_width} {β_0} {vx} {vy} {vz} {β_1}

This boundary condition was originally developed to allow for fluid slip near a dynamic contact line, a necessary condition for dynamic wetting line motion when the contact angle is not 180 degrees (viz. rolling motion condition). The slippage mechanism was deployed through the use of Navier's slip condition, which basically goes as

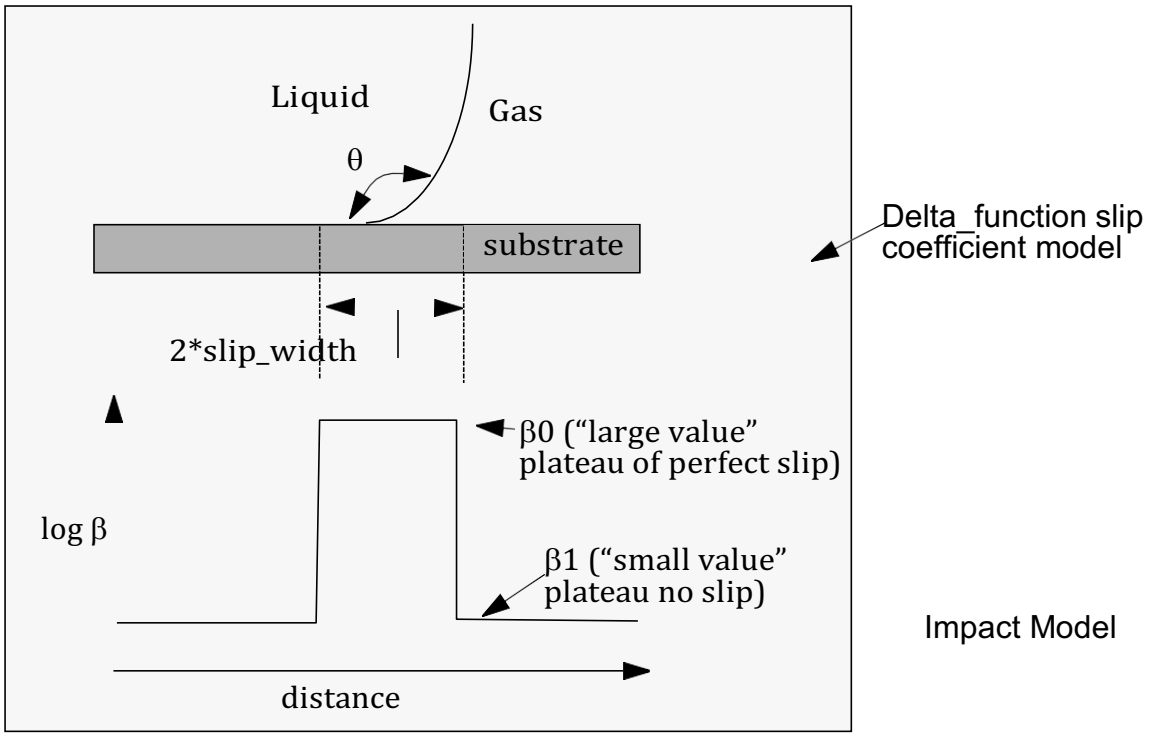
$$\underset{\sim}{n} \cdot \underset{\sim}{T} = \frac{\underline{v} - \underline{v}_s}{\beta}$$

Here β is the slip coefficient, which is taken to be variable depending on its proximity to the contact line (through the "slip_width" parameter). Note that the smaller the β, the more no-slip is enforced. The left hand side of this condition is the fluid traction on the substrate. $\underline{v}_S$ is the velocity of the substrate, specified component-wise with {vx} {vy} {vz}. This base functionality of applying the Navier slip condition still exists in this condition, but in addition it was furbished to allow for complete slip on the boundary if a gas film is being displaced by liquid. In this latter case, complete slip is a mechanism (subgrid event) that allows for the otherwise infinite stress to be relieved so that the liquid can make contact with the solid. The perfect slip condition at the substrate/gas surface is activated by just setting the slip coefficient to the large value, as this condition does anyway in the vicinity of a contact line. The "gas phase" is determined by determining which phase is the lighter one based on the density specification. The figure below details more on how this condition works for wetting/dewetting and for incipient liquid/solid impact.

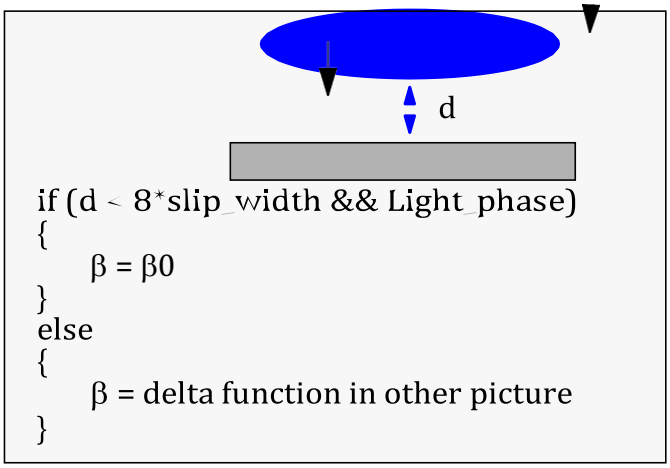
BC = FILL_CA SS # {contact_angle}

This condition is used to specify, or to be more accurate suggest, a contact angle subtended by a gas-liquid interface (or a level-set interface) and a solid/fluid interface. The contact angle condition is actually set in a "weak" sense as a force in a direction oriented with the tangent to the gas/liquid interface. In fact, it is applied in an analogous fashion as the weak-form CAP_ENDFORCE condition is often used in the ALE case. The functional form is roughly

VELO_SLIP_LS as applied to wetting and impact events



Impact Model



$$n_b \cdot \boldsymbol{T} = \sigma\delta_\alpha(F)(t_b \cdot \boldsymbol{n})(n_b \sin\theta_s + t_b \cos\theta_s)$$

where $\underline{n_b}$ and $\underline{t_b}$ and the normal and the tangent to the boundary and $\underline{n}$ is the normal to the free surface as determined from the gradient of the level set function and as evaluated at the contact line, σ is the surface tension, and δ(α) is the level-set delta function. Note that the contact angle is defined to be the angle in the liquid phase subtended by the solid surface and the liquid/gas surface, as measured on the liquid side. That is, an angle of 10 degrees is a highly "wetting" angle, and those greater than 90 degrees are "dewetting", with respect to the heavier phase.

Because this condition is applied in a weak, integrated sense, it will not always result in the contact angle specified except near equilibrium (no-flow) and when other long range body forces

are unimportant. Moreover, since the surface tension scales the force that is applied at the wetting line, this boundary condition tends to lose effectiveness when the capillary number is larger. We actually favor the use of WETTING_SPEED_LINEAR and WETTING_SPEED_BLAKE for dynamic situations over that of FILL_CA.

BC = WETTING_SPEED_LINEAR SS # {angle} {C_T} {wetting_length} {wetting_depth} {vx} {vy} {vz}

The concept behind this condition is that the speed of wetting/dewetting and the contact angle (dynamic) are found experimentally and even theoretically (see paper by Blake and De Coninck) to be related. More specifically, the velocity is related to the difference between the dynamic and equilibrium static contact angles, the latter being a thermodynamic property. It turns out that for transient problems, this relationship has proved to be superior than the "hard set" contact angle condition for ALE class problems (cf. VELO_THETA_TPL vs. CA conditions).

The WETTING_SPEED_LINEAR condition results in a weakly integrated force to be applied that is proportional to the following velocity/contact angle relationship:

$$v_{wetting} = \frac{\cos(\theta) - \cos(\Theta_s)}{C_T}$$

Note that the driving force is proportional to the difference, with receding angles leading to retraction of the contact line, and advancing angle leading to advancement of the contact line. The larger the coefficient C_T, the less-sensitive this relationship becomes.

The **`wetting_length`** parameter is used to characterize the distance over which the tangential force (which is proportional to this velocity) is applied. That is, this is the length_scale that is used to evaluate all level-set functions, like the delta and heaviside functions. If it is set to zero, then the level set length scale is used. If the length scale is set to zero, then the condition reverts to the sharp interface version.

The **`wetting_depth`** parameter is used as an additional "slip coefficient" and is applied to modulate the contact angle force. It is basically multiplicative with C_T at this stage. The smaller this parameter the greater the force applied at the contact line for a given contact angle difference and the converse is true.

BC = WETTING_SPEED_BLAKE SS # {static_contact_angle} {v_0} {g} {wetting_length} {wetting_depth} {vx} {vy} {vz}

The same concept of a contact-angle dependent wetting speed is deployed by this boundary condition (i.e., similar to WETTING_SPEED_LINEAR), but here we

deploy the same functional form suggested by Blake and DeConinck. This condition results in a weakly integrated force to be applied to the momentum equations that is proportional to the following velocity/contact angle relationship:

$$v_{wetting} = v_0 \sin h[\, g(\cos(\theta_s) - \cos(\Theta))]$$

Note that the driving force is proportional to a function of the difference between the cosine of the static and dynamic contact angles, with receding angles leading to retraction of the contact line, and advancing angles leading to advancement of the contact line. The smaller the coefficient v_0, the less sensitive this relationship becomes.

The **`wetting_length`** parameter is used to characterize the distance over which the tangential force (which is proportional to this velocity) is applied. That is, this is the length_scale that is used to evaluate all level-set functions, like the delta and heaviside functions. If it is set to zero, then the level set length scale is used. Ifthe length scale is set to zero, then the condition reverts to the sharp interface version.

The **`wetting_depth`** parameter is used as an additional "slip coefficient" and is applied to modulate the contact angle force. The smaller this parameter the greater the force applied at the contact line for a given contact angle difference and the converse is true.

BC = LS_ADC {angle} {capture_distance} {capture_rate}

This boundary condition can be used to simulate (and force in some sense) attachment, dewetting, or coalescence (ADC) events for capillary free surface problems using the LS method. Contact of a liquid and a solid surface can be facilitated with VELO_SLIP_LS, as discussed above, but the concept underpinning that implementation is often not "strong" enough to force a wetting event, i.e., simply allowing the "to-be-displaced" fluid at a solid surface to slip along it is not enough to force it out and allow contact with the other fluid. If the "to-be-displaced" fluid is a gas, then the event would be wetting contact, and if it is a liquid, then the event would be a dewetting phenomenon. VELO_SLIP_LS is only set up for the former.

This LS_ADC condition throws "caution to the wind" in some sense and allows the user to just force what they think needs to happen, with some control. Don't get us wrong, this condition does have reasonable physical underpinnings, but it allows the user to force the net effect of those underpinnings. The basic idea is that whenever contactor dewetting happens it is because of "weird molecular processes" that for a brief period of time in a localized area circumvent the normal continuum mechanics equations. Attempting to trick continuum physics into producing these sorts of events is perhaps not the best approach. VELO_SLIP_LS is such an attempt, and it doesn't always work.

Instead, introducing contact/dewetting events is something that should be done outside of the continuum mechanics time step iteration. The interfacial curve is

examined after every successful time step for finite elements in which the potential for contact/dewetting is significant. Things like "nearness" to the boundary and being "parallel" to the boundary would feed into determining this potential. Some algorithm is queried to decide if contact or dewetting should be initiated. If such a result is indicated, then the values of level set function on that element are directly manipulated to bring the interface through the boundary at that point. We might try to adhere to such continuum niceties like conservation of mass during this process, but the real process is just brute force changing level set nodal values to MAKE the interface contact the boundary.

We have added a probabilistic component to the question of whether to initiate a contact/dewet event. This is merely a first cut and there may be better ways to resolve that question. The operation and functional form of this condition are as follows:

a) if the free surface intersects the element containing the sideset on which this condition is set and

b) if the angle between the free surface normal and the solid surface normal is less than that which is specified as the first float (in degrees)

c) and a probability produced by the standard C rand() function is less than

$N_c h^2 \Delta t e^{1-\left(\frac{d}{\alpha}\right)^2}$ for d > α

and

$N_c h^2 \Delta t$ for d <= α

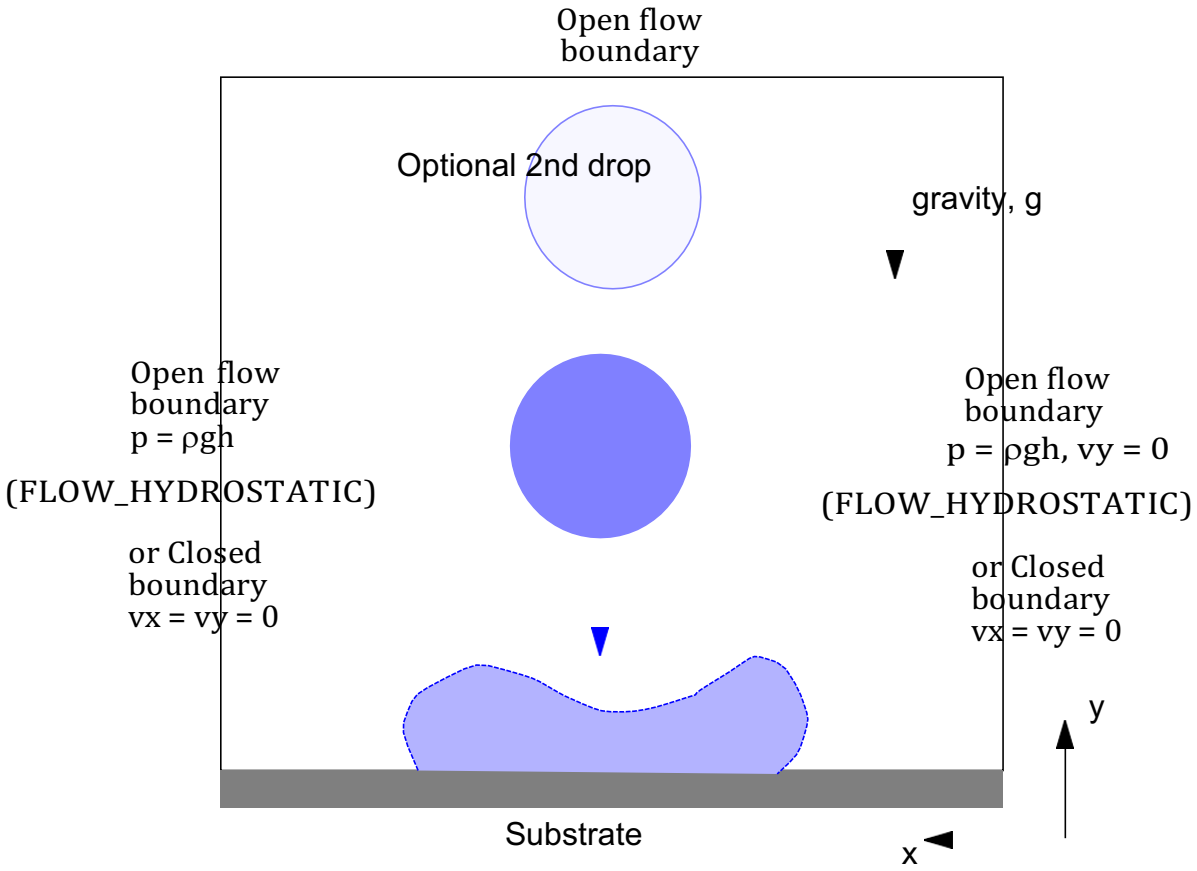
where N_c is the capture rate ($1/L^2$-t), h the element size, Δt the time step size, d the distance to the free surface from the solid surface, and α the capture distance. The capture distance is easily seen to be the boundary of maximum probability. The capture rate is harder to interpret, but should be set to roughly the inverse of the characteristic time of the process times the element size (for 2D) or the element size squared for 3D. Note also that preceding equation is applicable to two-dimensional surface elements in three dimensional problems. For two-dimensional problems, replace the h^2 term with h recognizing the out-of-plane dimension is usually taken to be unity.

Test Problem 1: Cylindrical (2D) drop contact

The directory which contains the template files for this problem is entitled **`drop_contact_tst`**. In that directory you will find a series of input files (**`push.input.*`**) and a series of corresponding results files (out.exoII.*) which can be used to examine the effects of various changes to the parameters of the conditions above. We also experimented with diffuse vs. sharp interface. The necessary changes to invoke

subelement, shape-interface methods are shown be Unix "**diffing**" **push.input** and **push_subelem.input**.

Lets review some of the features of this problem. The basic problem is shown in the following figure: Note that we are solving a two-dimensional problem, viz. the drops are



two dimensional (cylindrical) drops. For an axisymmetric version of this problem see GT-028.0. All units are specified in cgs. The computational box is basically 2.5 cm X 2.5 cm, and the drops are specified to be 0.4 cm radius (note the level-set initialization cards in push.input). Most of the cases were run with a single drop impacting the substrate. The drop was released from about 1.4 cm above the substrate and fall under the action of gravity. The sidewalls of the computational box were given both open flow and no-flow type BCs. The problem worked much better with the latter, as the open flow conditions often lead to multiple gas-flow solutions. The top of the computational box was left open, viz. no boundary conditions were applied. The time-integration and level set settings were as follows:

```
Time Integration Specifications
 ---
Time integration                  = transient
delta_t                           = 2.0e-5
Maximum time step                 = 4.0e+1
Maximum number of time steps      = 800
Maximum time                      = 2.2e+3
```

```
Minimum time step                = 1.0e-7
Maximum time step                = 0.01
Time step parameter              = 0.0
Time step error                  = 10.1 1   1   0   0   0 0
Printing Frequency               = 4
Fill Subcycle                    = 1
Fill Weight Function             = Explicit

Level Set Interface Tracking     = yes
Level Set Initialization Method  = Surfaces 1 SURF = CIRCLE 0. -0.4 0.4
Level Set Renormalization Method = Huygens
Level Set Length Scale           = 0.1
Level Set Renormalization Tolerance = 0.10000
Level Set Subelement Integration = off
Level Set Subgrid Integration Depth = 2
Level Set Contact Extension = yes
```

Note the **Explicit** fill weight function setting, which is more-or-less recommended for all problems at this point. We have found that this class of problem runs much more robustly with this option, even though you have to adhere to other Courant limits (see GT020.3). The maximum time step is set at 30, but the predictor-corrector error control (Time step error card) keeps it from growing too much. The level-set initialization method uses a circle geometry primitive, which is not surprising for a drop, and the renormalization method is set to Huygens. It is generally advisable to use Huygens_Constrained, but for whatever reason we didn't here. The level let length scale is set to 0.1, which is roughly an element dimension (good practice for diffuse level set surfaces). Quite frankly we never experimented with the effect of the level-set tolerance, the contact extension feature, or the subgrid integration depth.

For this problem we deploy the most popular and robust choice of finite element representation. Specifically we use a modified Taylor-Hood element that is biquadratic in velocity and bilinear in pressure, with additional feature in the pressure basis function to help pick up the pressure discontinuity at the capillary free surface (denoted here by Q1_XV). We use Q2 basis functions for the level set field. If you were running in 3D, you would likely use Q1/Q1_XV velocity-pressure element and pressure stabilization.

```
----
Problem Description
--
Number of Materials = 1
MAT =   fluid    1
        Coordinate System       = CARTESIAN
        Element Mapping         = isoparametric
        Mesh Motion             = ARBITRARY
        Number of bulk species  = 0
        Number of EQ            = 4
        EQ = momentum1   Q2 U1 Q2   1 1 1 1 1 0
        EQ = momentum2   Q2 U2 Q2   1 1 1 1 1 0
        EQ = continuity  Q1_XV P Q1_XV   1         0
        EQ = level_set   Q2 F  Q2   1 1     1
```

The key boundary conditions that we will discuss pertain to the impact and wetting/ spreading events on side set 2 of this geometry:

```
$impinge surface (second test series)
BC = VELO_SLIP_LS SS 2 0.05 10000. 0.0 0.0 0.0 1.e-6
BC = WETTING_SPEED_LINEAR SS 2 45. {C_T=1000.} 0.05 {wetting_depth=0.03}
0. 0. 0
$BC = FILL_CA SS 2 30.0
BC = VELO_NORMAL SS 2 0.
```

VELO_SLIP_LS as mentioned above serves two purposes. In the case shown here we have set a length-scale of 0.05 (about 1/2 of an element width), and so based on the discussion above the velocity will be allowed to slip on that boundary when the leading edge of the drop is about 8*0.05 = 0.4 cm from the substrate. This is a bit excessive and can be reduced, but nonetheless that is what is used. Once contacted, with these settings, the condition will apply a slip coefficient of 10000. around the contact line over a distance of about 1 element width. If you look at a typically solution in blot, you will notice that slippage is small but builds to a significant level as the drop closes to within an element or two of the substrate, due to the increased pressure due to the squeeze flow.

The WETTING_SPEED_LINEAR card in this case sets the static contact angle to 45 degrees, and the C_T parameter to 1000. A key parameter here is the wetting_depth (which multiplies C_T) and the wetting length scale. More on these parameters will be discussed below.

Also noteworthy is the VELO_NORMAL condition, which enforces impenetrability of the substrate. However, when a level-set problem is being run, this condition tests whether the level-set interface is within one-element width of the substrate, and if so, this condition is relaxed. This is used also as an expedient to force contact by gas leakage. We found that this leakage effect alone was not enough in many situations, hence the need for VELO_SLIP_LS. If you look at any of the solutions and visualize the velocity vectors (by magnifying them) near the substrate, you will notice this brief gas leakage just before contact.

To instruct you further on the effects of these boundary condition parameters, lets run a few tests.

If you run **`push.input`** (viz. **`goma -a -i push.input`**), you will see the result in **out.exoII.0**. Note that the drop impacts the substrate and then undergoes some oscillations until it comes to rest. Running this problem requires about 300 min on a modern linux workstation, and so you might want to run it over night. Otherwise just have a look at the "cooking show" result, viz. **`out.exoII.0`** has queued up in it the end result that you can pull out of the other oven for your viewers... Couple things are noteworthy about this run. First, it uses a diffuse interface model. We will compare with a sharp interface subelement integration below. Second, the static contact angle is 90 degrees as set on the WETTING_SPEED_LINEAR card, and you will notice that as the drop comes to rest, the angle tends to 90 degrees. However, during the transient the angle is anything

but constant as it responds to the functional relationship put forth for this model. Overall this behavior is expected and seems realistic. You might plot the velocity vectors and even the velocity components along the substrate, from nodes 1 to 4. You will see then the slip velocity just before impact (from **VELO_SLIP_LS**), around contact line after impact (from **VELO_SLIP_LS**), and the brief leakage velocity (from **VELO_NORMAL**). Finally, note that at equilibrium the drop shape is NOT a cylindrical arc. The Bond number, viz. $\rho g R^2/\sigma$, is around 3 and so the drop sags under the action of gravity at equilibrium.

Next run **push_restart.input**. Note that this setup restarts the calculation from **restart.exoII**, which is a time plane corresponding to the drop just before impact. This run employs a contact angle of 45 degrees, and a different renormalization tolerance and a larger level-set length scale. We have not saved this result for you, but you will notice that the drop comes to rest with indeed a smaller contact angle. The other two parameter changes don't have a noticeable effect.

If you visualize **out.exoII.1** and **out.exoII.3**, you will notice the effect of turning on subelement integration and increasing the C_T parameter in WETTING_SPEED_LINEAR. In both cases the drop comes to rest with a 45 degree contact angle as specified, but in the latter case which uses a lower sensitivity for the velocity dependence on contact angle, the drop does not overshoot as much and comes to rest quicker. In some sense, this corresponds to a surface with a surface energy that is such that the affinity for the liquid is higher. Subelement integration does have an appreciable effect either, and it is hard to tell in these two cases whether it is positive. Note that the mesh on these runs was VERY COARSE, and so that is why you see considerable noise at as the drop approaches equilibrium.

If you visualize **out.exoII.4**, you will see first hand what the impact model built into VELO_SLIP_LS does for you. In this run we set the slip coefficient be 1.e-6 for both the high and low plateau (see figure above), i.e., the coefficient was taken as constant and allows for very little slip:

```
> BC = VELO_SLIP_LS SS 2 0.001 1.e-6 0.0 0.0 0.0 1.e-6
```

Note that this solution shows that the drop does not contact the substrate until near equilibrium, as the air along the substrate cannot escape. This basically illustrates the benefit of the contact model.

Finally, in **push.input.9** we change the wetting parameters as follows:

```
BC = WETTING_SPEED_LINEAR SS 2 145. {C_T=10.} 0.05 {wetting_depth=0.00003}
0. 0. 0
```

Note here we have an extremely "dewetting" or hydrophobic contact angle and a very high senstivity of the wetting line speed to the contact angle driving force (not the relatively small C_T parameter and wetting depth parameter compared to other examples). Now these are extreme cases, but running this shows you that you can

make the substrate effectively hydrophobic. Visualizing the result in **out.exoII.9** reveals that the drop takes on an advancing contact angle even after initial recoil, and tends to "ball" up. Note the kinks in the free surface at later time steps, indicating lack of resolution. The drop then kind of rolls on the substrate in an unstable way. Clearly this result lakes refinement and is full of artifacts. It was performed merely to demonstrate hydrophobic behavior, albeit in an unrealistic way.

Test Problem 2: Double drop contact

This problem builds on the previous one and simply adds a second drop that is trailing the first during free fall. The idea is to test Goma's algorithms to handle multiple impacts (drop-on-drop). The problem can be found in the same directory, viz. **drop_contact_tst**, and the input file is **push_two.input**. Basically the only change over the single drop case is that the Level-Set Initialization card has two entries:

```
Level Set Initialization Method  = Surfaces 2
   SURF = CIRCLE 0. -0.9 0.3
   SURF = CIRCLE 0. 0.90 0.3
```

We are going to point out some first-hand experience here that we were able to correct by adjusting the level-set renorm tolerance, length scale, and renormalization method. We were having trouble getting drops to merge until we fiddled with these enough to achieve success.

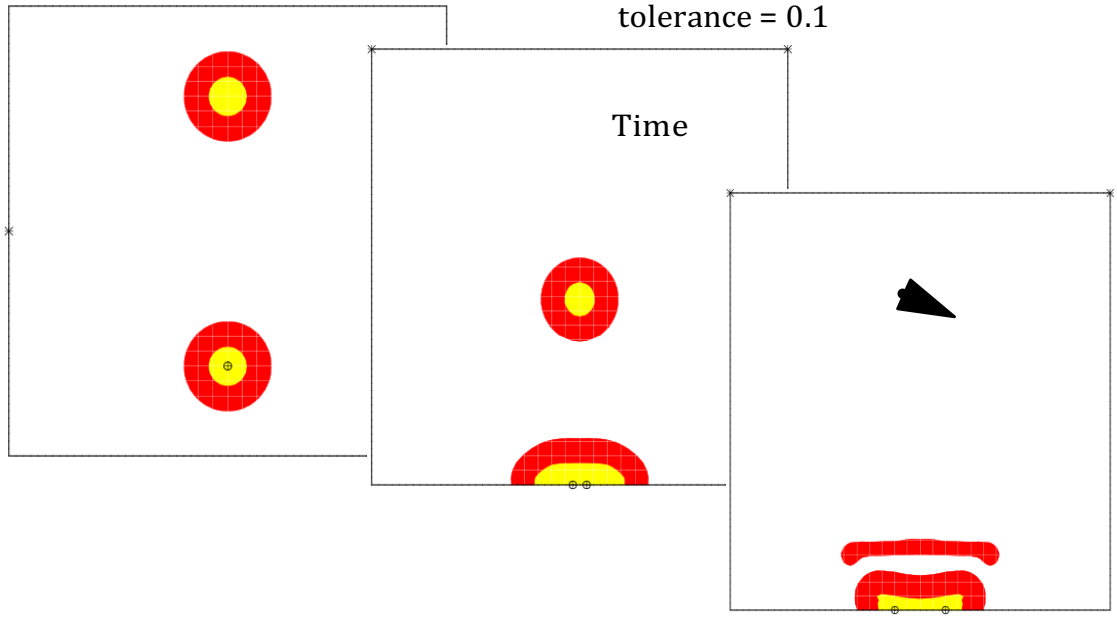
In all cases a drop is released near the bottom substrate and the second near the top of the box. The radius has been reduced to 0.3 cm. However, some earlier tests were done with larger drops and closer together. Queued up in **out_twodrop_1.exoII** is one such solution attempt. In this case we use a diffuse interface model. Note that the first drop impacts the substrate, as expected, but in this case the second drop never reaches the first due to the thin air layer that persists between the two. The calculation evolves in seemingly unnatural on counter-intuitive way, with the second drop developing all sorts of lobes and structure. PRRRRS. Upon further review, we found that HUYGENS CONSTRAINED? Raise the Rnorm tolerance?

Queued up in **out_twodrop_2.exoII** is the same case but with subelement integration. Note that we seem to be able to take the calculation further, but still the two drops never coalesce during the time-range of the run. Although they may eventually do so, we expected impact and drop joining earlier in the calculation. Several reasons for this lack of impact are possible. First, we could be dealing with too-coarse of a grid. That in combination with lack of relative momentum between the drops keeps the impact pressure too small to force all gas away from in between. Clearly, if this is the case, we need a subgrid model to force this coalescence. Keep in mind that all of the expedients added to VELO_NORMAL and VELO_SLIP_LS do not apply here. *(N.B. after writing this paragraph I discovered that if I raise the Level Set Renormalization Tolerance, this case works fine and the drops merge. More below.)*

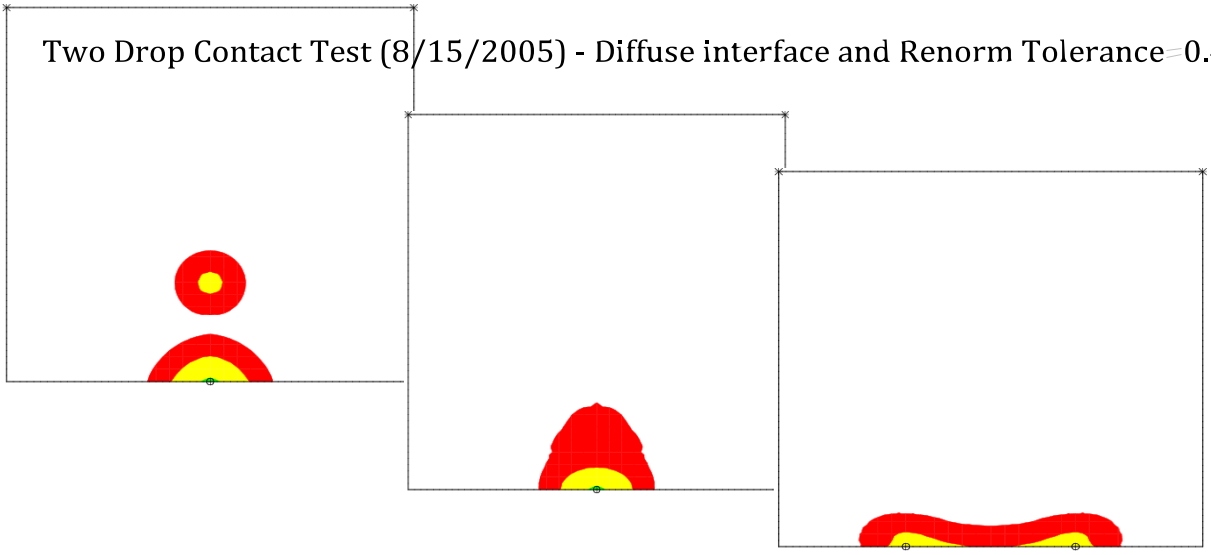
We know in some regimes of viscosity and density that this resistance to joining is possible, but feel in this case they should merge. We did find that if you run the diffuse

interface capability with a larger Renormalization tolerance (we happen to bump it up to 0.4 from 0.1) that the two drops do merge fine. In `push_two_3.input` is just such a case. Notice how the upper drop looses considerable mass before merging with the lower one, but this was due to a large Level Set length scale (0.3). We were able to correct this by lowering the level-set length scale to 0.15. The result is in `push_two_4.input` and `out_twodrop4.exoII`. Clearly we need to find some sort of compromise here, as we have been striving to sharpen the interface. When you do this you force yourself into subgrid models much more so than you would in the old diffuse cases.

Two Drop Contact Test (8/15/2005) - subelement integration with renorm tolerance = 0.1



Two Drop Contact Test (8/15/2005) - Diffuse interface and Renorm Tolerance = 0.4

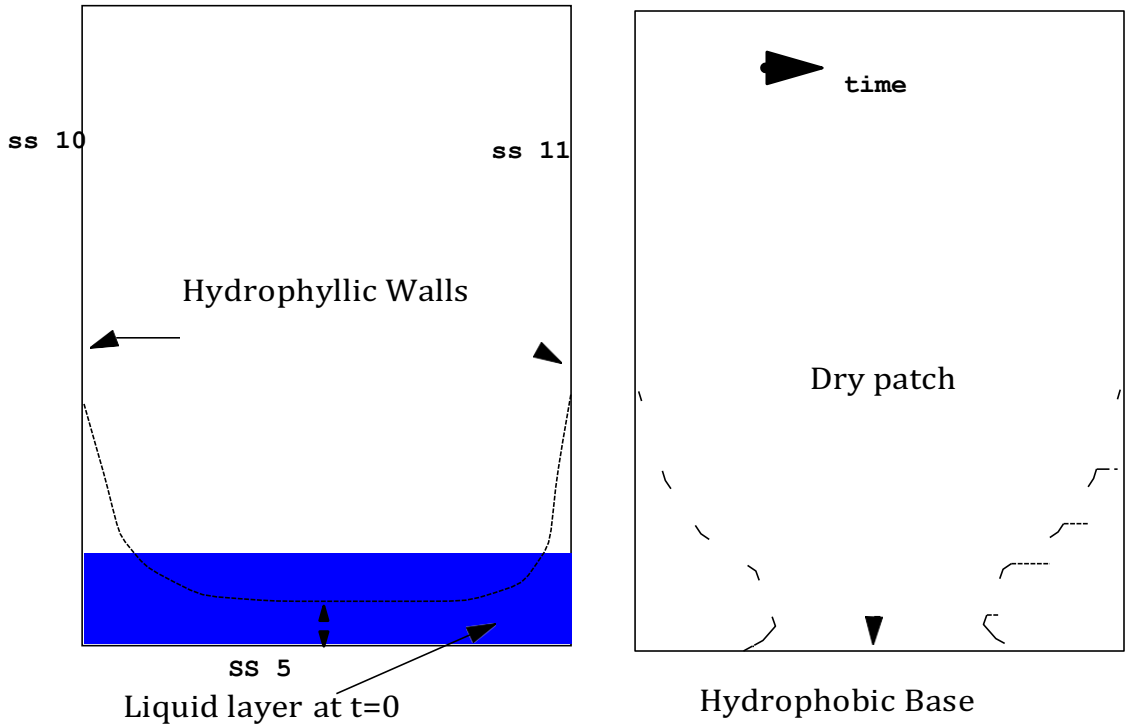


The results for a succesful run are shown in the bottom of the figure. After all this experimenting we found that the drops merge fine if you renormalize less, by either lowering the frequency or upping the tolerance. I think the frequent renorms alter the squeezing flow field in such a way as to prevent merging.

Test problem 3: Capillary tube dewetting events.

To illustrate a dewetting event, we solve a much simpler problem in which a thin liquid layer in the bottom of a container is wetting with respect to the side walls, but dewetting with respect to the bottom. The test problem can be found in the directory **dewetting_tst**. The problem is illustrated in the following figure:

Dewetting Test Problem Setup.



The boundary conditions for this problem are as follows:

```
BC = LS_ATTACH SS 5 0.
BC = VELO_SLIP_LS SS 5 0.2e-4 10000. 0. 0. 0. 1.e-6
BC = WETTING_SPEED_LINEAR SS 5 110. {C_T=10000.} 0. {wetting_depth=0.0001}
0. 0. 0.
BC = LS_ADC SS 5 {angle=2} {capture_distance=0.2e-3} {capture_rate=80e+7}

BC = LS_CAPILLARY LS 0

BC = VELO_SLIP_LS SS 10 0.1e-3 10000. 0.0 0.0 0.0 1.e-6
```

```
$BC = FILL_CA SS 10 10.0
BC = WETTING_SPEED_LINEAR SS 10 20. {C_T=10000.} 0. {wetting_depth=0.0001}
0. 0. 0.
BC = VELO_NORMAL SS 10 0.0

BC = VELO_SLIP_LS SS 11 0.1e-3 10000. 0.0 0.0 0.0 1.e-6
$$BC = FILL_CA SS 11 10.0
BC = WETTING_SPEED_LINEAR SS 11 20. {C_T=10000.} 0. {wetting_depth=0.0001}
0. 0. 0.
BC = VELO_NORMAL SS 11 0.0
```

The sidewalls denoted by Sidesets 10 and 11 have the same boundary conditions. Note that we apply a highly wetting contact angle. The bottom has also the same set of conditions, but we use a highly dewetting contact angle of 110 degrees. At time t=0, we the liquid will clearly start climbing the walls, thereby thinning the film near side set 5. This is where the boundary condition LS_ADC comes in. As the film thins and eventually recedes into the first layer of elements, this boundary condition begins computing probabilities based on the specified capture_rate and capture_distance. Basically, the higher the capture rate, the more likely a dewetting event will be initiated. With the values specified here, the nucleation of a dry patch happens rather quickly, followed by a second nucleation event.

We encourage you to experiment with these parameters and try a finer mesh. Things seem to be working properly, and are controllable, but the only way that quantitative results can be expected is for measurements to be made on real system aimed at these two parameters.

NOTE: on LS_ADC, you might have to coarsen your mesh. Yup, you heard it correctly! If the interface doesn't ever recede into the element next to the solid, dewetting surface, no dewetting will occur. In that case your mesh is too fine for your own good!