



date: August 6, 1999

Original May 15, 1998; Revised December 18, 1998, PRS; Revised May 29, 1999, PRS; Revised August 6, 1999, PRS; Revised Sept 25, 2015 PRS

to: Distribution

from: P. R. Schunk, 9111, MS0826

subject: THE NEW TOTAL-ARBITRARY-LAGRANGIAN-EULERIAN (TALE) CAPABILITY and its applicability to coating with/on deformable media. (GT-005.5)

keywords: TALE, deformable roll coating, remeshing, remapping, contact angles, CUBIT, mapvar, volume constraints, augmenting conditions, automated continuation, slot coating, brazing, boundary condition duplication

input records: Convected Lagrangian Velocity, KIN_DISPLACEMENT, NO_SLIP_RS, VELO_TANGENT_SOLID, VELO_SLIP_SOLID, SOLID_FLUID_RS, FORCE_RS, CAP_REPULSE, DX_RS, DY_RS, GD_TABLE, GD_LINEAR, RIX, Mesh Motion=TOTAL_ALE, HOOKEAN_PSTRAIN, INCOMP_PSTRAIN, VELO_NORMAL, VELO_TANGENT, Initial Guess, Post Processing Fluxes, Number of Augmenting Conditions, AC=VC, END OF AC, Newton Relaxation, Continuation=first, EQ=mom_solid, EQ=mesh, BCdup.txt

PREFACE

This tutorial assumes the user has gone through the beginner's training tutorial on GOMA (GT-001.4) and SEAMS and the subsequent tutorials on slot and roll coating, published for the October 1997 Semiannual meeting of the CRMPC. If you would like copies of these tutorials, contact Duane Labreche (dalabre@sandia.gov), or search the secure web site. You also need to have CUBIT to complete most of these exercises. Review the latest CUBIT remeshing tutorial (Document GT-006.3)

Even if you are comfortable with modeling roll coating and slot coating flows with GOMA, adding the deformable rolls or substrates adds significant complexity and nonlinearity. This memo is designed to help you pursue such complexities with GOMA by introducing the new TALE algorithm and guiding you through the developers experiences in applying it.

Summary of revisions since original documents GT-005.0 and GT-005.1: Complete theoretical report complete (GSR-02.0). TALE theory removed from this document. New deformation gradient tensor in **Convected Lagrangian Velocity** and **NO_SLIP** cards. New continuation strategy for release of upstream meniscus on film split. Plethora of new interfacial slip and kinematic conditions. New "referenced" **KIN_DISPLACEMENT_BC**. Remesh and remapping in Lagrangian and ALE conjugate problems (cf. GT-006.3). Reformulated no-slip affects in transient problems.

NOTE: In 2013 some significant changes were made to the input format of KIN_DISPLACEMENT_BC card. Those changes are summarized in the appendix of this report. Please modify these conditions where needed.

EXAMPLE PROBLEMS COVERED IN THIS MEMO:

- Flooded Deformable Roll Model on page 3 (example problem in def_roll_nip_TALE)
- Deformable Forward Roll Coating with Film Split on page 17 (example problem in **film_split_remesh**)
- Deformable Substrate with a Moving Contact Line on page 28 (**contact_line**)
- Transient contact line problems--capillary wetting in braze flow on page 33 (**capillary_wetting**)

NOTE: THIS INFORMATION IS CONSIDERED PROPRIETARY TO THE CRMPC AND SANDIA NATIONAL LABORATORIES, AND SHOULD BE TREATED UNDER THE TERMS AND CONDITIONS OF THE CRADA.

Introduction

A Total Arbitrary Lagrangian Eulerian (TALE) technique has been developed specifically for the finite element-based solution of quasi static solid/dynamic fluid mechanics problems, for which substantial interaction by stresses in the solid and fluid materials exists (see report GSR-02.0). One critical element of the development is a generalized kinematic boundary condition applicable at external and internal interfaces, unifying the concept of ALE techniques in fluid mechanics with the less common applications in solid mechanics. In CRMPC document GSR-02.0 the user can find a complete equation description of a generalized fluid-structural interaction system, theoretical foundation for the modeling of moving three-phase liquid/solid/gas contact lines on a deformable solid, and the integration of the material displacement fields with mesh displacement fields, together with the proper transformations to account for changing/varying solid-stress reference state. This tutorial seeks to instruct the users of the TALE implementation how to apply it effectively to realistic coating and other manufacturing problems.

The users are strongly encouraged to read and understand the TALE technical report (Schunk, 2000 - SAND Report SAND2000-0807) before attempting the problems in this tutorial.

Boundary conditions and capability discussed in this tutorial (please see the following documents for more details: GOMA Advanced Capability Users Manual (SAND2000-2465); CUBIT remeshing and remapping tutorial (GT-006.3), TALE technical report (SAND2000-0807):

- Volume constraint capability to aid in initial solutions for roll coating on deformable rolls (see the deformable roll film split problem).
- Augmenting conditions to deliberately manipulate mesh internal to a solid to minimize mesh distortion in TALE (see the Capillary wetting problem in **capillary_wetting**)
- Remeshing and remapping LAGRANGIAN and TALE problems (first two examples).
- Slip models for rigid, Lagrangian, or TALE solids in steady or transient situations. (third example)
- NO_SLIP_RS

Distribution

$$v_m|_{fluid} = v_m|_{solid} \quad (EQ 1)$$

- VELO_TANGENT_SOLID

$$\tilde{t} \cdot \tilde{v}_m|_{fluid} = \tilde{t} \cdot v_{sfs} \cdot \tilde{F}_m + \tilde{t} \cdot \frac{d\tilde{x}_m}{dt} \quad (EQ 2)$$

- VELO_SLIP_SOLID

$$\left(\tilde{t} \cdot \tilde{v}_m|_{fluid} - \tilde{t} \cdot v_{sfs} \cdot \tilde{F}_m - \tilde{t} \cdot \frac{d\tilde{x}_m}{dt} \right) = \beta \tilde{n} \cdot \tilde{t} \cdot \tilde{T}|_{fluid} \quad (EQ 3)$$

- SOLID_FLUID_RS

$$\tilde{n} \cdot \tilde{T} = \tilde{n} \cdot \tilde{\sigma} \quad (EQ 4)$$

- DX_RS, DY_RS

$$\left(\tilde{d} - \tilde{d}^0 \right) = 0 \quad (EQ 5)$$

- KINEMATIC DISPLACEMENT

$$\tilde{n} \cdot \left(\tilde{d} - \tilde{d}^0 \right) - \tilde{n} \cdot \left(\tilde{d} - \tilde{d}^0 \right) = 0 \quad (EQ 6)$$

Flooded Deformable Roll Model

As an instructive example of the new TALE algorithm, we will begin with the flooded deformable roll model. In a previous training memo on roll coating (GT-003.1) we treated the rolls as advected Lagrangian materials. Now we will treat them as generalized solid materials whose grid motion and material motion are not necessarily related, except on the surface of the materials. You will see that this allows us to go to larger compression of the rolls without mesh distortion. The real advantage, however, is realized in the film split and rolling bank problems below.

Basically here we are immersing the rolls in liquid and predicting the interaction of the liquid and rollers, including the roller deformation. This is the simplest of all roll coating models. The practice problem directory which corresponds to this problem is available from Duane Labreche (dalabre@sandia.gov) or Randy Schunk (prschun@sandia.gov). It is called `def_roll_nip_TALE`.

In that directory you have at a minimum the following files:

roll.exoII roll.fas contin.dat liquid.mat lower_roll.mat liquid.mat roll.gen out.exoII roll_input

Distribution

The two material files are `lower_roll.mat` and `liquid.mat`. We will do a simulation with one effectively rigid roll, the upper one, and one soft roll, the lower one. We will then move the rolls together so we have effectively what is termed a “compression”, or a negative gap.

A diagram of the basic problem is shown in the following figure, with key node sets and side sets indicated.

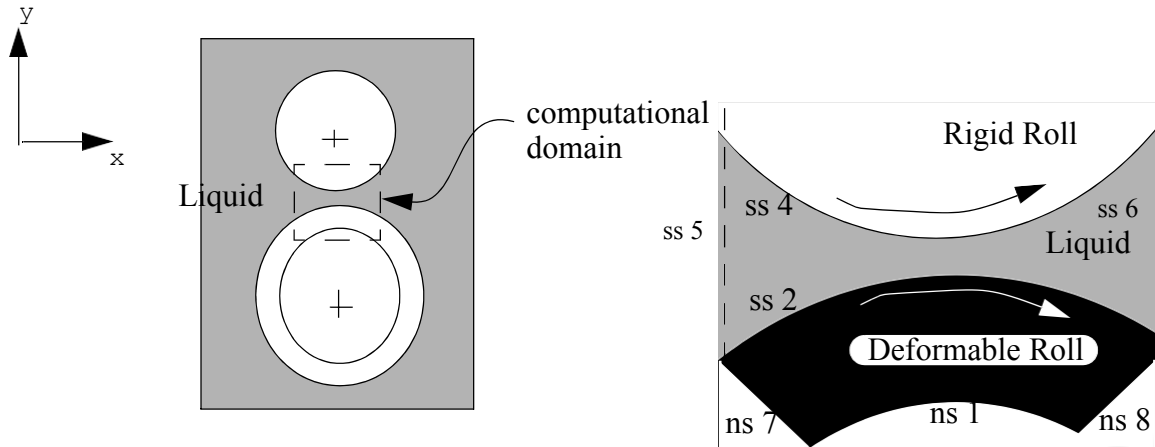


Figure 1. : Flooded Deformable roll setup.

To get an initial solution here, edit first both the `lower_roll.mat` and make sure that the `lame_mu` and `lame_lambda` cards are as follows:

```

---Mechanical Properties and Constitutive Equations
$$Solid Constitutive Equation = INCOMP_PSTRAIN
Solid Constitutive Equation = HOOKEAN_PSTRAIN
Convective Lagrangian Velocity = ROTATIONAL {rollsp_b} 0. 0. 0.
$$$$ shore A of 50 is 2.7e7 Pa for Young's modulus
Lame MU = CONSTANT 2.7e6
#Lame LAMBDA = CONSTANT 0.
Lame LAMBDA = CONSTANT 1e6

```

```

$$ Note this is set to 1 even though we've no species to
$$ that incompressibility will be satisfied when solving the
$$ continuity equation.
Stress Free Solvent Vol Frac= CONSTANT0.0

```

```

Pseudo-Solid Constitutive Equation = NONLINEAR
Pseudo-Solid Convective Lagrangian Velocity = NONE
$$Pseudo-Solid Lamé MU = CONTACT_LINE 600 1. 1.e3 .2
Pseudo-Solid Lamé MU = CONSTANT 1.
Pseudo-Solid Lamé LAMBDA = CONSTANT 1.

```

-5-August 6, 1999

Distribution

Several aspects of this first section of the roller material file are noteworthy. First you see we now have a properties of the real solid and properties of the pseudo-solid. If the pseudo-solid cards are absent, then the code will stop as the material is specified as a **TOTAL_ALE** material in the `roll_input` file, viz.

```
MAT = lower_roll 1

Coordinate System = CARTESIAN
Element Mapping = isoparametric
Mesh Motion = TOTAL_ALE
Number of bulk species = 0
```

Notice here that the real-solid constitutive equation is **HOOKEAN_PSTRAIN**, which is an compressible plane strain model and thus requires shear modulus and volume modulus, manifested here through Lamé coefficients (cf. GOMA 2.0 User's Manual, SAND97-2404). We will contrast the behavior here with a true incompressible rubber below, using the **INCOMP_PSTRAIN** model. Also notice that the pseudo-solid constitutive equation is set to **NONLINEAR**, for nonlinear elasticity. We are free to choose this in any way, so as to maintain the integrity of the mesh, as it will not interfere with the stresses in the real-solid.

Also have a look at the `liquid.mat` file. Notice it has no Pseudo-Solid cards, and that in `roll_input` the liquid material is specified as **ARBITRARY**:

```
MAT = liquid 1

Coordinate System = CARTESIAN
Element Mapping = isoparametric
Mesh Motion = ARBITRARY
Number of bulk species = 0
```

So you can see the convention adopted here. The solid material property cards remain as they always were in the material file for materials with mesh motion of type **LAGRANGIAN** or **ARBITRARY**. If the mesh motion is of type **TOTAL_ALE**, you need the additional **Pseudo-Solid** cards as shown above with the usual **Solid Constitutive** equation cards at the top being used for the real solid description.

IMPORTANT ASIDE: Note that the pseudo-solid properties for the meshes in both materials are set to the same values. This is done to avoid intense mesh shearing at the interface, which results with disparate mechanical mesh properties.

Before we run these problems, we need to understand several other portions of `roll_input` and `roll.fas`. First, in `roll_input` the problem description section is as follows:

```
Number of Materials = 2

MAT = lower_roll 1

Coordinate System = CARTESIAN
Element Mapping = isoparametric
```

-6-August 6, 1999

Distribution

```

Mesh Motion = TOTAL_ALE
Number of bulk species = 0

Number of EQ = 4
EQ = mesh1Q2D1Q2  0.  0.  1.  1.      0.  0.
EQ = mesh2Q2D2Q2  0.  0.  1.  1.      0.  0.
EQ = mom_solid1Q2D1_RSQ2  0.  0.  1.  1.      0.  0.
EQ = mom_solid2Q2D2_RSQ2  0.  0.  1.  1.      0.  0.
EQ = continuity Q1      P      Q1  1.              1.

                                div ms  adv  bnd  dif      src porous

MAT = liquid      2

```

```

Coordinate System = CARTESIAN
Element Mapping    = isoparametric
Mesh Motion = ARBITRARY
Number of bulk species = 0

```

```

Number of EQ = 5
EQ = momentum1  Q2U1Q2  0.  1.  1.  1.0  0.0  0.
EQ = momentum2  Q2U2Q2  0.  1.  1.  1.0  0.0  0.
EQ = continuityQ1P Q1  1.              0.
EQ = mesh1Q2D1Q2  0.  0.      1.  1.  0.  0.
EQ = mesh2Q2D2Q2  0.  0.      1.  1.

```

Notice that in the **lower_rol1** material we have a mesh motion of type **TOTAL_ALE**, and so in addition to the usual **mesh1** and **mesh2** pseudo-solid equations there are **mom_solid1** and **mom_solid2** equations which activate the real-solid momentum balance.

IMPORTANT ASIDE: In addition to the solid momentum equations we must also solve the continuity equation if **INCOMP_PSTRAIN** constitutive model is used. We will activate this **continuity** card when we switch to that model, but for now we will leave the number of equations at 4. Notice on the continuity equation we have an advective term multiplier of unity and a source term multiplier of unity. Both terms must be active to properly apply the incompressibility constraint to a **LAGRANGIAN** or **TOTAL_ALE** material. The advective term turns out to be the $det(F)$ term for a **TOTAL_ALE** material and the source term is unity. Be careful though that the **Stress Free Volume Fraction Solvent** card is set to zero, as that card can actually change the source to greater or less than unity in shrinking and swelling problems.

The liquid material, Block 2, on the other hand, has a mesh motion scheme set at “**ARBITRARY**” and only has a pseudo-solid description with **mesh1** and **mesh2** together with the usual Navier-Stokes equations.

In the **lower_rol1.mat** file you will see the card:

```
Convective Lagrangian Velocity = ROTATIONAL {-rollsp_b} {x1} {y1} 0
```

-7-August 6, 1999

Distribution

This card specifies that there is a advective term put on the *real*-solid momentum equations that will account for the solid body rotation of the roll, appropriately corrected for the deformation it undergoes. i.e. the term takes the user-specified velocity field and projects it into the deformation gradient tensor term to get the proper accounting of the kinematics and stresses. The model here on this card is “**ROTATIONAL**”, and the input required is the rotation rate (in Radians/sec, i.e., the **rollsp_b** APREPRO variable defined in the **roll.fas** file), and the center of rotation, which is either point 1 or 11, depending on which material file.

The geometry and mesh are defined in the file **roll.fas**. Check out the geometry in **fastq** to see, i.e., **fastq -a roll.fas**. You will notice the geometry in that file is ill-defined, as some of the circular segments are rotated completely around. That is because this file has been altered to run in CUBIT (cf. CUBIT remeshing tutorial GT-006.3). The circular arcs are defined in the opposite sense when this file is read into cubit. By switching the arc end points back on each “CIRC” line card, you can then generate a mesh with **fastq**. We use CUBIT here so we can perform deformed geometry remeshing.

Go ahead and generate the original mesh with the command

```
cubit initial_mesh.jou
```

This **CUBIT** journal file actually directs cubit to read in the **roll.fas** file and generate a mesh. That mesh is put into **roll.exoII**.

Now, back in the **roll_input** file, the boundary conditions on the roll surfaces are specified as follows:

```
---
Boundary Condition Specifications
---
Number of BC = -1

$Inflow and outflow plane geometry
BC = PLANE SS 5 1. 0. 0. {-x5}
BC = PLANE SS 6 1. 0. 0. {x5}

$$Upper roll geometry about the polar point [x11,y11]
BC = GD_PARAB SS 4 R_MESH_NORMAL 0 MESH_POSITION2 0 {x11*x11 + y11*y11 -
R_O*R_O} {-2.*y11} 1
BC = GD_PARAB SS 4 R_MESH_NORMAL 0 MESH_POSITION1 0 {0.} {-2.*x11} 1

$$Upper roll no slip velocity
BC = VELO_NORMAL SS 4 0.
BC = VELO_TANGENT SS 4 0 {-rollsp_t} 0. 0. 0.

$
$ Roll inlets and outlets and cover/roll interface
$ Comment: really need a kinematic-displacement bc
$          for SS 7 8 and 1 that is applied to the real-solid.
$          rather than the pseudo solid, as is done now.
$          This condition needs to be rotated though.
$
```

-8-August 6, 1999

Distribution

```

$$ {y1_new = y1 + delta_gap_rs}
$$BC = GD_PARAB SS 1 R_MESH_NORMAL 0 MESH_POSITION2 0 {x1*x1 + y1_new*y1_new -
R_I*R_I} {-2.*y1_new} 1
$$BC = GD_PARAB SS 1 R_MESH_NORMAL 0 MESH_POSITION1 0 {0.} {-2.*x1} 1
BC = KIN_DISPLACEMENT SS 1 1

BC = DX_RS      NS      1      0.0
BC = DY_RS      NS      1      {delta_gap_rs} 1.0
BC = DX         NS      7      0.0 1.0
BC = DY         NS      7      {delta_gap_new} 1.0
BC = DX_RS      NS      7      0.0 1.0
BC = DY_RS      NS      7      {delta_gap_rs} 1.0
BC = DX         NS      8      0.0 1.0
BC = DY         NS      8      {delta_gap_new} 1.0
BC = DX_RS      NS      8      0.0 1.0
BC = DY_RS      NS      8      {delta_gap_rs} 1.0

BC = NO_SLIP_RS      SS 2 1 2
BC = SOLID_FLUID_RS  SS 2 1 2
BC = KIN_DISPLACEMENT SS 2 1
#####
END OF BC
#####

```

A few **APREPRO** variables need to be explained here. The top part of `roll.fas` defines some key quantities:

```

$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = -0.07}
$ displacement setting after anneal {delta_gap_rs = 1.*(Gap_orig - Gap_new)}
$ displacement setting after anneal {delta_gap_new = 1.*(Gap - Gap_new)}
$ roll cover thickness (1" = 0.0254) {d = 2.54}

```

Notice the variables `delta_gap_rs` and `delta_gap_new` are defined appropriately so upon remeshing and remapping, the values of the displacement boundary conditions (for both real solid, `DX_RS`, `DY_RS`, and pseudo-solid `DX` and `DY`) are preserved. Remember that these are specified displacements, not absolute quantities. This brings up another issue for remeshing. If you remesh and remap, you typically like to set the mesh displacements back to zero, i.e., anneal the mesh stresses. If you do that, you need to reset these boundary conditions. This is why we have 3 gap variables, viz. `Gap_orig`, `Gap`, and `Gap_new`. How these variables are used during remeshing steps is taken up when we remesh below. `Gap_orig` is used to actually generate the original mesh.

The noteworthy BCs are several, including the `SOLID_FLUID_RS`, `KIN_DISPLACEMENT` and `NO_SLIP_RS` on side set 2, the lower_roll/liquid interface. There the stresses are balanced between the liquid and solid, resulting in the roll deformation. However, in this case the `_RS` extension implies that these conditions are applied to the real-solid and not the pseudo-solid, but otherwise have the same meaning as in the **LAGRANGIAN** case (cf. earlier roll coating tutorial GT-003.0 memo to CRMPC). The pseudo-solid mesh motion and the real-solid displacement are connected with the `KIN_DISPLACEMENT` condition. This condition applies the equation: $n \cdot (d - d^0) - n \cdot (d - d^0) = 0$ to the

pseudo-solid differential equation in a normal-tangential rotated fashion. The integer data following these cards are respectively `side_set_no.`, `solid_material_no.`, `liquid_material_no.` Notice that the lower roll is block/material number 1 and the liquid material/block number 2. The `NO_SLIP_RS` card simply sets the liquid velocity to the *real*-solid velocity along that surface. The displacement fields actually must be related on all boundaries, and so you notice also that we set the x- and y-*real*-solid displacements to the same value that as the x- and y-*pseudo*-solid displacements on node sets 1, 7, and 8. This is done with the `DX_RS` and `DY_RS` cards.

IMPORTANT ASIDE: It has been noticed that the combination of conditions `VELO_NORMAL` and `VELO_TANGENT_SOLID` (instead of `NO_SLIP`) are better for mass conservation at fluid solid interfaces. The film-split example below employs these instead.

IMPORTANT ASIDE: Before we practice with this model, it is noteworthy that model much more sensitive to parameter changes when the roll is taken as incompressible. Using a `HOOKEAN_PSTRAIN` model with a high shear modulus makes the roll about 90% incompressible, which is inaccurate, but allows for much larger deformations and ease of application. In this case you want to set the number of equations in `roll_input` file in the `lower_roll` material back to 4, to exclude the continuity equation and the pressure variable. If you are doing exploratory work and not worried about the details, we highly recommend this approach. You can converge from much poorer initial guesses, at much higher speeds, and much larger compressions. We have many ways to improve on this for the incompressible case, including annealing the mesh, etc., but we have not yet undertaken this.

The `INCOMP_PSTRAIN` model may be more accurate for rubber-based materials, but behaves quite differently at large compressions. In fact, in this example, it bottoms-out (i.e., rolls come into contact) much sooner, as if the roll surface lacks flexural rigidity. We think improving the models to Mooney-Rivlin, etc., would help matters.

Now make sure that your initial guess card in "`roll_input`" is set to "`zero`", viz.

```
Initial Guess          = zero
```

and run GOMA,

```
goma -a -i roll_input
```

Notice on this initial solution we have the following settings for `Gap_orig`, `Gap`, and `Gap_new`:

From file `roll.fas`

```
$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$ {Gap_orig = 0.1}
$ gap (leading edge to substrate)   {Gap = 0.1} {Gap_new = 0.1}
$ displacement setting after anneal {delta_gap_rs = 1.*(Gap_orig - Gap_new)}
$ displacement setting after anneal {delta_gap_new = 1.*(Gap - Gap_new)}
$ roll cover thickness (1"= 0.0254) {d = 2.54}
```

Distribution

```

$
$ Roll Radius          {R_I = 5.*2.54}
$ Outer Roll Radius   {R_O = R_I + d}

$ The bottom roll speed is set in the lower_roll.mat file through
$ the Convected Lagrangian Velocity card. The UNITS on that card with
$ the rotational model require RADIANS/s. So for 5.06 m/s linear speed
$ the input is 2.*Pi*(rev/s)*R_O = 5.06 => 5.06/R_O = 36.4
$ For some realistic speeds, we want 3.06 for rollsp_t and 20.4 for rollsp_b
$
$ bottom roll speed {rollsp_b = 230/R_O}
$ top roll speed {rollsp_t = 230}

$ Roll extent {theta1 = 63} {theta2 = 117} {theta_intermediate = 87}

```

From file `lower_roll.mat`

```

Lame MU      = CONSTANT    2.7e6
#Lame LAMBDA = CONSTANT 0.
Lame LAMBDA = CONSTANT    1e6

```

Noteworthy here is that the `Gap_new` variable is set to the same value as `Gap` and `Gap_orig`, which is 0.1 cm for starters. Note also that the `rollsp_t` and `rollsp_b` are set to different numbers, the `rollsp_b` (the `_b` for “bottom”) variable being divided by the outer radius of the roll. The reason for this is that the deformable roll speed is set in the “Convective Lagrangian Velocity” card of the `lower_roll.mat` file, and that card expects a rad/s rotation rate, and not a linear speed, as the `VELO_TANGENT` card used for the upper roll. Everything is in CGS units.

Also note that the extent of the rolls is defined by the theta variables. You can adjust those and regenerate the mesh to determine their effect. These are similar to the original roll-coating tutorial (GT-003.0), so look there for definitions.

Save the solution with “`cp soln.dat contin.dat`” and then edit `roll_input` and change the Initial guess card to “`read`”. You will notice that with these properties this converges with no relaxation.

Now we will press the rolls together during operation. Obviously you cannot do this from the start as the mesh can not be generated on a negative gap. This is done differently for the case in which the rolls are being modeled as a deformable moving solid than when we are modeling them just as a rigid surface, see below. Note that on node sets 1, 7 and 8 there are displacement boundary conditions as follows:

```

BC = KIN_DISPLACEMENT SS 1 1

BC = DX_RS      NS      1      0.0
BC = DY_RS      NS      1      {delta_gap_rs} 1.0
BC = DX        NS      7      0.0 1.0
BC = DY        NS      7      {delta_gap_new} 1.0
BC = DX_RS     NS      7      0.0 1.0
BC = DY_RS     NS      7      {delta_gap_rs} 1.0
BC = DX        NS      8      0.0 1.0
BC = DY        NS      8      {delta_gap_new} 1.0
BC = DX_RS     NS      8      0.0 1.0

```

Distribution

```
BC =   DY_RS           NS      8           {delta_gap_rs} 1.0
```

The Y displacement of the lower roll will be affected by moving the rigid core/rubber interfaces (Node set 1), and the artificial ends of the roll (node sets 7 and 8). The second floating point number on these cards (after the displacement) is the Newton relaxation parameter. All Dirichlet conditions in GOMA can handle this factor. If present, GOMA includes these conditions in the matrix and iterates on them “softly”. If not present, they are “hard” set, in which case you can only take minuscule steps to avoid mesh distortion. In **BLOT** or in the little figure above you can see which surfaces these are. So try the following displacement in the **roll.fas** file (N.B. this file is included in the “**roll_input**” file and there is no need to regenerate the mesh):

```
$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = 0.1}
```

to

```
$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = 0.07}
```

Now repeat the following steps for **Gap_new** = 0.04 then 0.02 and then 0.00, with zero-order continuation. Don’t forget that you are continuing by “**cp soln.dat contin.dat**”. You could use automatic continuation here with the multiparameter Hunting capability. In any case, you should now have a solution with this case:

```
$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = 0.00}
```

BLOT the solution in **out.exoII**, and look at the coating nip. You will notice that although the mesh is hanging together, it is distorting (splaying outwards). Now you should remesh and remap to correct this problem.

 IMPORTANT ASIDE: It is better to remesh and remap before the mesh gets too distorted, otherwise your remap will be wiggly and you will lose your solution. Because we have no quantitative mesh distortion measures, you will have to eye-ball it for now.

Remeshing and remapping is detailed in Document GT-006.3. Please consult that tutorial for the overall procedure and the general rules. Here you will simply do the following:

```
cp out.exoII tmp.e
cubit remesh.jou
cp rem.gen tmp.g
mapvar tmp < mapvar.in
blot tmp.int
```


Distribution

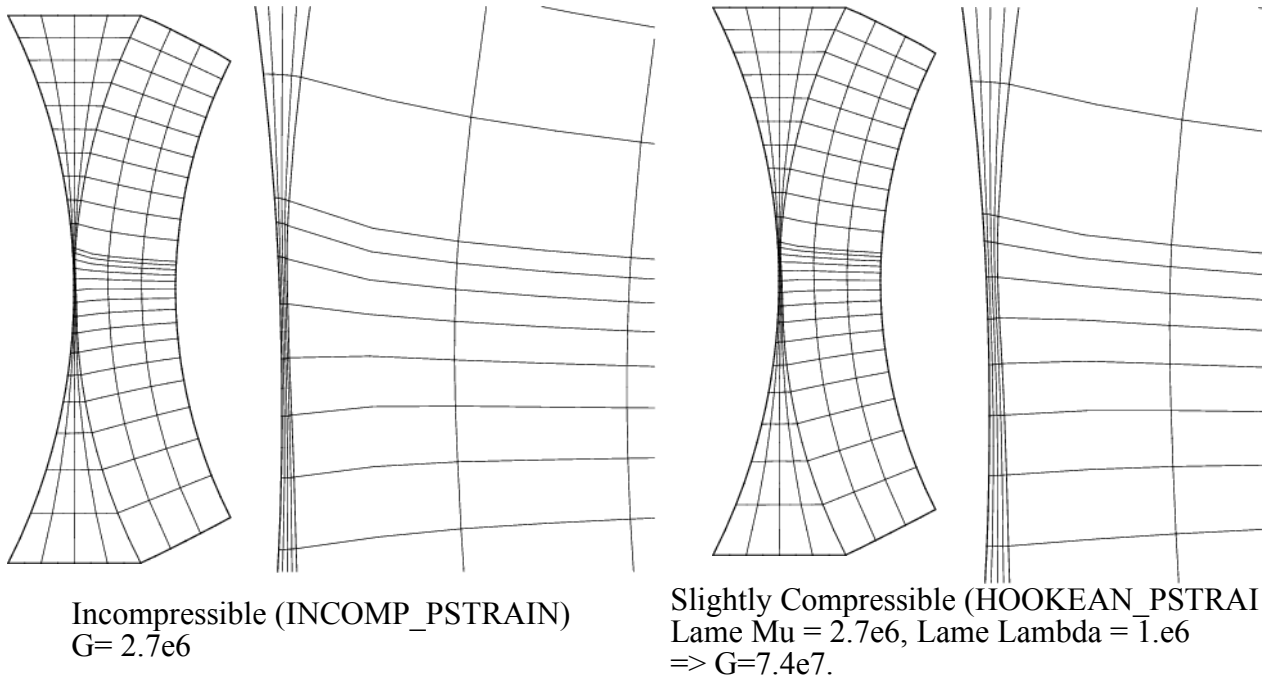


Figure 4. Sample results: comparison between compressible and incompressible roll. Same conditions as in figures above.

3. gap = 0.07 get with -r 0.1 -n 6 and then full newton

```
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = 0.06}
```

4. gap = 0.06 full newton

```
$ gap (leading edge to substrate) {Gap = 0.1} {Gap_new = 0.05}
```

5. gap = 0.05 full newton

6. Mesh splaying, so remesh.

```
711 cp out.exoII tmp.e
712 cubit remesh.jou
713 cp rem.gen tmp.g
714 cat > mapvar.in
715 mapvar tmp < mapvar.in
716 blot tmp.int
```

The last step is to make sure your $\mathbf{D}\mathbf{Y}_{RS}$ field is smooth.

7. Converge again with the following initial guess:

```
$$FEM file = roll.exoII
FEM file = tmp.int
Output EXODUS II file = out.exoII
...
$$Initial Guess =read
Initial Guess = read_exoII_file tmp.int
$$Initial Guess = read_exoII_file restart.exoII
```

Distribution

8. Next step

```
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.05} {Gap_new = 0.03}
Initial Guess =read
$$Initial Guess = read_exoII_file tmp.int
```

Converged full newton.

9. Next step

```
$                               {Gap_orig = 0.1}
$ gap (leading edge to substrate) {Gap = 0.05} {Gap_new = 0.01}
Initial Guess =read
$$Initial Guess = read_exoII_file tmp.int
```

10. Converged full newton. When to **Gap_new** = 0. then to -0.01 then to -0.02.

11. Remeshed. Same procedure as above. Reset as follows.

```
$ gap (leading edge to substrate) {Gap = -0.02} {Gap_new = -0.02}
$ displacement setting after anneal {delta_gap_rs = 1.*(Gap_orig - Gap_new)}
...
Initial Guess = read_exoII_file tmp.int
$$Initial Guess = read_exoII_file restart.exoII
```

10 steps -r 0.1.

cp soln.dat contin.dat

```
Initial Guess = read
```

Full newton.

NOTE mesh is really bad and roll/roll contact has occurred. Abort.

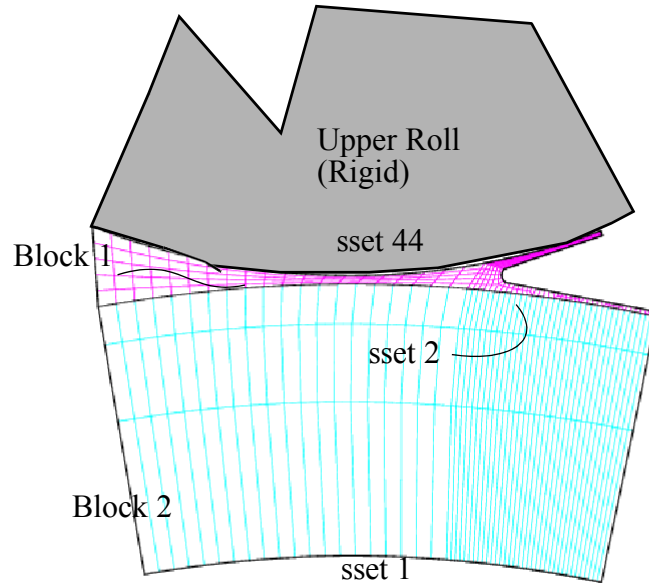
Deformable Forward Roll Coating with Film Split

The geometry and the defining **fastq** entities for this problem are for the most part the same as those used in the CRMPC roll coating tutorial memo GT-003.0 previously released. What is added here are some of the developers' experiences in getting an initial guess for realistic coating conditions, a new continuation scheme that helps you get to high coating speeds and large compressions (together with a newly designed mesh), and the additional features needed for extension of this model from two rigid rolls to one deformable and one rigid roll. Before going through this exercise, please review the other tutorial for the techniques of getting an initial guess to the film split location and the rolling bank. (e.g. GT-003.1 and GT-005.3). The directory which contains the problem discussed here is called **film_split_remesh**.

The geometry and relevant side sets/node sets in this problem is illustrated here

There are no additional TALE issues over and above those discussed for the flooded roll case. You'll notice here the lower roll is material 2 and the liquid material is material 1, for good reason. A new

Distribution



Conditions: Lower Roll Speed 18 cm/s; Upper Roll Speed 18 cm/s; Gap = 0.035 cm

set of conditions will be used for the interface between the roll and the liquid which involves VELO_NORMAL/VELO_TANGENTIAL_SOLID. We will discuss this below.

We will attempt to get a solution with the following conditions:

```

TITLE
$ GEOMETRY AND OPERATING CONDITIONS (CGS)
$ Original Gap setting      {Gap_orig = 0.035}
$ gap (leading edge to substrate)  {Gap = 0.035} {Gap_new = +0.035}
$ displacement setting after anneal  {delta_gap = 0.*(Gap_orig - 0.03)}
$ displacement setting after anneal  {delta_gap_new = 0.*(Gap - Gap_new)}
$$$$cover thickness should be 0.55 cm
$ roll cover thickness (1"=0.0254)  {d = 1.1}
$ bottom roll speed      {rollsp_b = 18.3/(1.1 +4.95)}
$ top roll speed        {rollsp_t = 18.3}
$ initial film thickness for v2/v1=1  {h_t = .02}

$$$$Bottom roll has rubber cover (should be 4.4 , so it is 4.95 total)
$ Bottom Roll Radius Inner      {R_I_b = 4.95} {R_I_b_new = 4.95}
}
$ Bottom Outer Roll Radius (radius + cover) {R_O_b = R_I_b + d} {R_O_b_new =
R_I_b_new+d}
$ Bottom Outer Roll Radius + film thickness  {R_O_b_h = R_O_b + h_t} {R_O_b_h
_new = R_O_b_new + h_t}
$ Bottom Outer Roll Radius + 2.5*film thickness {R_O_b_2h = R_O_b + 2.0*h_t}

$$$$Top roll is rigid
$ Top Roll Radius Inner      {R_I_t = 3.} {R_I_t_new = 3.}
$ Top Roll Radius + film thickness  {R_I_t_h = R_I_t + h_t}
$ Top Outer Roll Radius + 2.5*film thickness {R_I_t_2h = R_I_t + 2.0*h_t}

```

Distribution

```
$ Roll extents {theta1_t = 70} {theta2_t = 111}
$ Roll extents {theta1_b = 78} {theta2_b = 100}
```

Notice here that the conditions are such that the line speed is 18.3 cm/s (*N.B. many of the previous roll coating templates have been in MKS, so be aware*). Also notice that the gap is 0.035 cm and the hard roll radius is 3 cm.

Several things to note about the mesh being used here. It is generated from CUBIT, in much the same way that the example above was generated. If you do not have CUBIT, some small alterations will have to be made to the `roll.fas` file regarding circular arcs (see previous section for discussion). Also included are the CUBIT journal files `initial_mesh.jou` and a `remesh.jou` that are to be used in the exact same way as the flooded case above, viz., initial mesh generated by:

```
cubit initial_mesh.jou
```

which puts the mesh file into `roll.exoII`. And a deformed geometry remesh generated by:

```
cubit remesh.jou,
```

which generates a `rem.gen` file to be used for remapping with `mapvar`.

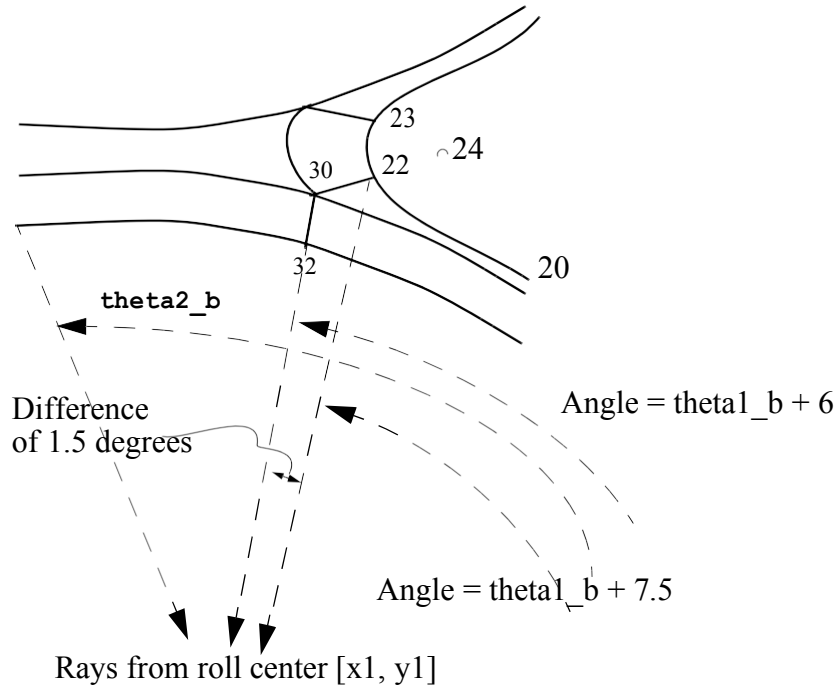
The mesh for the most part is made up of structured regions. If one chooses to pave the liquid regions, this typically translates to a much higher density of mesh. That density has to be carried through the solid roll and becomes inefficient there. HOWEVER, it is expedient to use an unstructured grid with as few of region boundaries across remeshing as possible. Unfortunately CUBIT retains those deformed region boundaries which can lead to a bad aspect ratios and angles.

As in the rigid roll case, the key to this problem is locating the film split and rolling bank. The following sections of `roll.fas` help you do that:

```
$$$$$$$The offset angles here determine the location of the film split$$$$$
$$$$$$$Keep the phase angles the same for points 30, 32. Same for$$$$$
$$$$$$$mesh region that resolves the film split.          $$$$$$
point 20 {x20=R_O_b_h*cosd(theta1_b)} {y20=R_O_b_h*sind(theta1_b)}
point 22 {x22=R_O_b_h*cosd(theta1_b+6)} {y22=R_O_b_2h*sind(theta1_b+6)}

point 30 {x30=R_O_b*cosd(theta1_b+7.5)} {y30=R_O_b*sind(theta1_b+7.5)}
point 32 {x32=R_I_b*cosd(theta1_b+7.5)} {y32=R_I_b*sind(theta1_b+7.5)}
```

These points describe the film split region shown in the figure. The key is that the points defining this region begin on the lower roll and associated with the angles and phase offsets specified on the lower roll. No longer as in earlier tutorials are we assuming that the upper roll has the same radius as the lower one. There are offset angles for the film split defined using `theta1_t` and `theta1_b`. The angle `theta1_b` is the right most extent of the roll, as measured from the right pointing horizontal axis. You can see that the film split is assumed to be 6 degrees in from that outflow, and that the mesh around the film split is at 7.5 degrees offset. If you widen the gap between these angles, the four sided region centered on the film split will simply elongate. Experimenting with these angles and simply re-running CUBIT will show you how it works. The same goes for the upper roll, with points 23 etc.



Point 24 defines the center for the film split curvature. Depending on how far your initial guess mesh is into the nip, you may have to make this point closer in the x direction to points 23 and 22, or farther. You can see it is defined as a weighted distance between points 20 and 22.

Our approach to the full solution can be summarized as follows:

1. Find a paper, thesis, or experimental data which reflects a roll coating situation under realistic speeds and conditions AND that you know had a stable, realizable solution. You have got to get into the “ballpark” on this problem or you will never make it in a reasonable time. The key is to find, under some set of conditions, what the film split location will nominally be. Actually, with the new volume constraint capability, we can get a solution for almost any guess, by adjusting the upstream pressure BC to keep it stable, but you may be so far off on that pressure that continuing to the solution is too time consuming.
2. Adjust the θ_{1_b} and θ_{1_t} and the offsets for the film split, generate some meshes, and solve for the steady state solution with a fixed free surface (i.e., using **VELO_NORMAL** on the surface rather than **KINEMATIC**). Also, fix the inflow velocity components and not the pressure on the left hand side. In the **roll_input** file you can find the **FLOW_PRESSURE** boundary condition. Comment that out and uncomment the u and v boundary conditions on the same node set. This step helps force a smooth velocity field on the initial mesh.

Make sure you save the solution “**cp soln.dat contin.dat**” and set your initial guess to **read**.

Distribution

3. Turn on the volume constraint cards:

```

Number of augmenting conditions = -1
AC = VC {mat_id = 1} {volid = 1} {bcid = 24} {dfid=0} {compid = 0} {const =2.029045e-01}
END OF AC

```

Switch over to a **FLOW_PRESSURE** condition on the inflow (left plane). Comment out the velocity. Don't worry about the pressure level as the **vc** will take care of this.

Note on this card that the **mat_id** is 1, which is the liquid region, and the **bcid** is 24, which is the number of the **FLOW_PRESSURE** boundary condition. This number can be found by setting the **SS** spec. on that card to **SC**, and running **GOMA** with the **-bc_list** option (see Advanced Capability Manual by Gates et. al. GSM-1.0).

Release the free surface by commenting out the **VELO_NORMAL** card and uncommenting the **KINEMATIC**, **CAPILLARY**, **SURFTANG** cards.

```

$$BC = VELO_NORMAL SS 100 0.
BC = KINEMATIC SS 100 0.
BC = CAPILLARY SS 100 40. 0 0 0
BC = SURFTANG NS 200 {sind(theta1_b)} {-cosd(theta1_b)} 0. 40.
BC = SURFTANG NS 210 {sind(theta1_t)} {cosd(theta1_t)} 0. -40.

```

Run ONE Newton iteration with zero relaxation to get the liquid region mesh volume, viz.

```
goma -a -i roll_input -r 0.0 -n 1
```

This will actually not try to solve the problem, but will evaluate the mesh volume of the liquid for you. The output will be something like:

```

      Residual      Correction

ToD   itn  L_oo  L_1   L_2   L_oo  L_1   L_2  lis asm/slv (sec)
-----
14:48:03 [0] 2.6e-02 1.6e-01 5.6e-02 3.9e+04 3.3e+06 3.3e+05 1 2.0e+01/8.8e+00
          AC 1.3e-02 1.3e-02 1.3e-02 5.9e+04 5.9e+04 5.9e+04 2.5e+01/7.6e+00
L_oo cause: dof=6687 u1_0 n=1494 dof=6088 P_0 n=1373

```

```

-----
Augmenting Conditions: 1
Number of extra unknowns: 1

```

```
MT[ 1] VC[ 1]=2.162914e-01 Param=3.725000e+05
```

Put the evaluated volume on the last constant of the **AC** card, viz. **{const=2.162914e-01}**

4. Now go for a free surface solution by simply issuing:

```
goma -a -i roll_input
```

You may need to relax a bit.

5. Examine the pressure at the output of the free surface solution. On a flooded nip, with a film split, that pressure should not be very large. For instance, in the case here we are considering that the left half of the domain is submerged in a bath. Calculating ρgh gives a pressure of about 3000 dyn/cm² there, so the pressure you get with the volume constraint, i.e., the one required to hold

Distribution

the film split in the nip at your presumed location, should be close. Recall that there is a building of the pressure due to the converging part of the roll, so I was comfortable with about 10000 dyn/cm² here, as that is what full roll calculations revealed. If your pressure is way off the mark, experiment with initial guess film split locations by adjusting the angle offsets above around the film split, the presumed curvature (by adjusting the weightings on point 24), viz.

```
$$ Define center for arc of circle on film split
point 24 {x24 = (16*x22+x20)/17} {y24 = (y23+y22)/2}
```

\$ here is you raise the 16 to 26 and the 17 to 27, you get a much higher curvature. Also, you can adjust the slopes down to the final film thickness by adjusting the following card in roll.fas:

```
$ Outer Roll Radius + 2.5*film thickness {R_O_2h = R_O + 1.9*h_t}
```

\$ so that the 1.9 is higher or smaller.

6. Some other noteworthy cause-and-effects that can be gleaned from papers, common sense, or by trial-and-error:
 - Increasing inertia (i.e., by raising liquid density, for instance) pushes film split out a
 - Decreasing gap sucks-in split but pushes out rolling bank.

Here is the experience I had with the conditions in this example.

First set the Lamé coefficients in the `lower_roll.mat` file to high values, to make the lower roll effectively rigid. We will soften it later. I find the 1.e10 does the job. Also, notice in the `roll_input.fixed` file that the inflow flow rate is specified with the following cards:

```
$$ Inflow boundary (specified pressure)
$$ Target is rho*g*h = 1. * 980 * 3 cm ~ 3000
$$BC = FLOW_PRESSURE SS 5 3.725e+05
BC = U NS 5 {h_t*rollsp_t^2/(y15-y5)}
BC = V NS 5 0.
```

which we find necessary to get a smooth initial guess. We will change out these cards for a specified pressure after we get the fixed grid solution.

Generate the initial mesh:

```
cubit initial_mesh.jou
```

Get the initial fixed grid solution:

```
goma -a -i roll_input_fixed -r 1.0
cp soln.dat contin.dat
```

Figure out what your initial volume of liquid is:

```
goma -a -i roll_input_VC -r 0.0 -n 1
```

Read off the output

```
-----
Augmenting Conditions: 1
Number of extra unknowns: 1
```

```
MT[ 1] VC[ 1]=2.027589e-01 Param=3.725000e+05
```

-23-August 6, 1999

Distribution

And change the {const = ???} part of the VC augmenting condition in `roll_input_vc` with the volume of VC[1]=2.027589e-01.

Run GOMA to get a free surface solution. Note that “diff” on `roll_input.fixed` and `roll_input_vc` shows that we have already released the free surface. So after updating the volume, run

```
goma -a -i roll_input_VC -n 30
cp soln.dat contin.dat
goma -a -i roll_input_VC -r 1.0
```

Some relaxation is automatically applied for you, on the Newton Relaxation card, with specified tolerances. Notice that the pressure needed to keep the film split in the nip is 3.561999e+03, as indicated by the output:

```
18:02:10 [3] 1.7e-08 8.8e-08 2.5e-08 6.2e-06 2.4e-04 2.6e-05 1 1.0e+01/5.4e+00
AC 3.4e-15 3.4e-15 3.4e-15 1.4e-05 1.4e-05 1.4e-05 1.3e+01/5.5e+00
L_oo cause: dof=1170 u2_0 n=258 dof=1065 P_0 n=242
```

```
-----
Augmenting Conditions: 1
Number of extra unknowns: 1
```

```
MT[ 1] VC[ 1]=2.027589e-01 Param=3.561999e+03
```

First we will soften the rolls before we compress them. Lower the Lamé Mu and Lamé Lambda coefficients by 3 orders of magnitude so they are

```
Solid Constitutive Equation = INCOMP_PSTRAIN
#Solid Constitutive Equation = HOOKEAN_PSTRAIN
Convective Lagrangian Velocity = ROTATIONAL {rollsp_b} 0. 0. 0.
###Shore A is 50 = 310 psi = 2.7e7 dyne/cm^2
Lame MU = CONSTANT 4700000.
#Lame LAMBDA = CONSTANT 0.
Lame LAMBDA = CONSTANT 2000000.
```

Note that with the definitions of elastic constants (defined in Eq. 5.2 in the GOMA 2.0 users manual, SAND97-2404) that these Lamé coefficients correspond to a Young’s modulus of 1.1e+7 dyn/cm² and a Poisson’s ratio of 0.15. What is important here is that the Young’s modulus corresponds to roughly a 50 Shore A roll. For higher Poisson’s ratios, we had an increasingly harder time getting to extreme compressions.

Go ahead and obtain a solution with a liquid bead of the same volume with these conditions:

```
goma -a -i roll_input_VC -n 15
```

This command gives you some automatic relaxation. You can follow these 15 iterations with some full Newton steps:

```
cp soln.dat contin.dat
goma -a -i roll_input_VC -r 1.0
```


Distribution

These changes are in `roll_input_auto_cont`. Diffing that file with `roll_input_vc` will show these changes.

```
goma -a -i roll_input_auto_cont -r 1.0
```

If you blot this file, you will see that with the continuation proceeding to lower and lower inlet pressures, the film split recedes into the nip. When you have gone as far as you can go, read off the pressure and set the `FLOW_PRESSURE` card to that value. The last successful pressure step is around $3.168882e+05$ dyn/cm². In the file `roll_input_vc1` we have set up the appropriate conditions, by turning off the first order

Now copy the continuation file:

```
cp soln.dat contin.dat
goma -a -i roll_input_VC1
```

This step verifies you have a solution. Turn on the Volume constraint in that file and run one iteration of `roll_input_vc1` to see what the new volume of the liquid material is:

```
goma -a -i roll_input_VC1 -n 1 -r 0.0
```

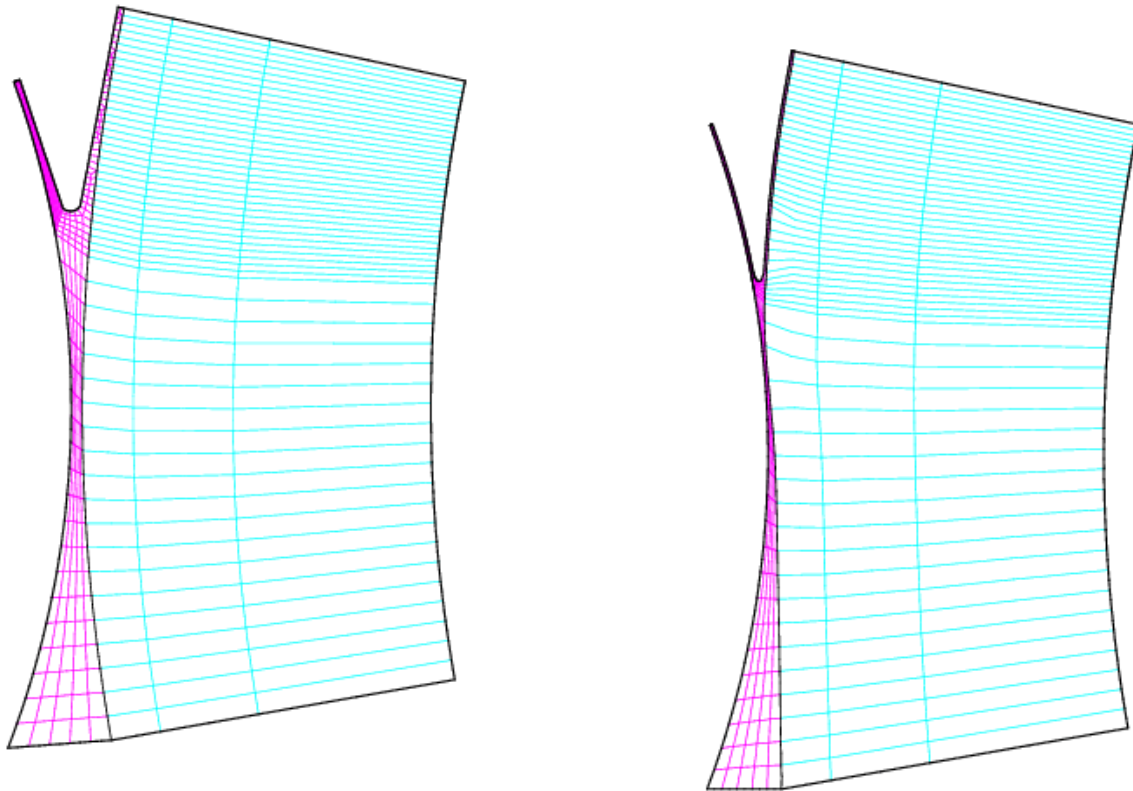
Pick off the volume ($\sim 1.728490e-01$), set it on the `vc` card in `roll_input_vc1`, and rerun:

```
goma -a -i roll_input_VC1 -r 1.0
```

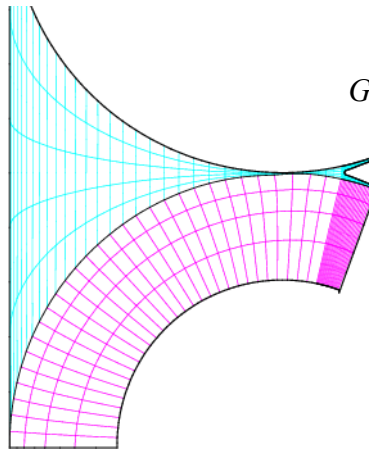
The results of these sequences can be seen in the figure below. To go to larger compressions, you can continue this procedure. You may also want to remesh. To remesh this problem we have put a `remesh.jou` `CUBIT` file in the directory. You can experiment with it. Once remeshed, on the remap phase make sure you map the solid material variables first with the `MAPVAR` “`map`” command. An example is shown here.

```
cubit remesh.jou
cp out.exoII tmp.e
cp rem.gen tmp.g
mapvar tmp
CMD> map 2 to 2
CMD> map 1 to 1
CMD> def 1
CMD> time 0
CMD> exit
```

This leaves you with a `exoII` file `tmp.int` which contains the new mesh and the solution appended to it. To continue from this mesh you must change the `FEM` file card in `roll_input_vc1` to “`tmp.int`” and the `Initial Guess` card to “`read_exoII_file tmp.int`”. In this case you are able to converge with the new mesh and new mapped solution without relaxation. NOTE: this remeshing step actually can be improved by going to a completely unstructured mesh. You will notice that even the new mesh is distorted around the film split because the bulk macro-regions defined in the original mesh were distorted. I had much better luck starting from scratch with a new domain decomposition, with fewer zones.



Gap = +0.035 cm



Gap = -0.05 cm

Full roll simulation to get the pressure at the inlet for the truncated domain.

IMPORTANT ASIDE: Note that the roll moduli are set as follows:

###Shore A is 50 = 310 psi = 2.7e7 dyne/cm²
Lame MU = CONSTANT 4700000

Distribution

```
#Lame LAMBDA = CONSTANT 0.
Lame LAMBDA = CONSTANT 2000000.
Stress Free Solvent Vol Frac= CONSTANT0.0
```

The net result is a modulus of nearly $1.e7$, corresponding to 50 Shore A. Again, we are using a slightly compressible roll. We can make it incompressible, but we cannot get to near the compressions for the same reasons as cited in the previous example, due to a curious bump that forms due to lack of refinement along the roll. The Mooney Rivlin equation is a generalization that is known to fit rubber deformation better than the NeoHookean solid we are using.

SECOND IMPORTANT ASIDE:

The conditions we employ at the roll-liquid interface are:

```
$$$BC = NO_SLIP_RS      SS 2 2 1
BC = VELO_TANGENT_SOLID SS 2 2 1
BC = VELO_NORMAL      SS 2 0 .
BC = SOLID_FLUID_RS   SS 2 2 1
BC = KIN_DISPLACEMENT SS 2 2
```

Notice how we are not use the `NO_SLIP_RS` condition on velocity. Instead we are using the set of conditions `VELO_TANGENT_SOLID` and `VELO_NORMAL`. Some explanation is warranted.

As shown in the introduction of this document, the corresponding mathematical conditions to these cards are:

- `NO_SLIP_RS`

$$v_m|_{fluid} = v_m|_{solid}$$

- `VELO_TANGENT_SOLID`

$$\tilde{t} \cdot \tilde{v}_m|_{fluid} = \tilde{t} \cdot \tilde{v}_{sfs} \cdot \tilde{F}_m + \tilde{t} \cdot \frac{d\tilde{x}_m}{dt}$$

- `VELO_NORMAL`

$$\tilde{n} \cdot (\tilde{v}_m - \tilde{v}_s) = 0$$

Note that the latter two are just the normal and tangential components of the `NO_SLIP` condition. We find this combination of conditions better as it keeps liquid from penetrating the roll material much better, due to the explicit forcing of impenetrability with the `VELO_NORMAL` condition. The tangential component also shows that the velocity of the solid actually has two contributions: one from the user-prescribed motion (first term on right of `VELO_TANGENT_SOLID` condition), and a second from deformation and motion of the solid material dictated by the balance of forces (second term on right).

Distribution

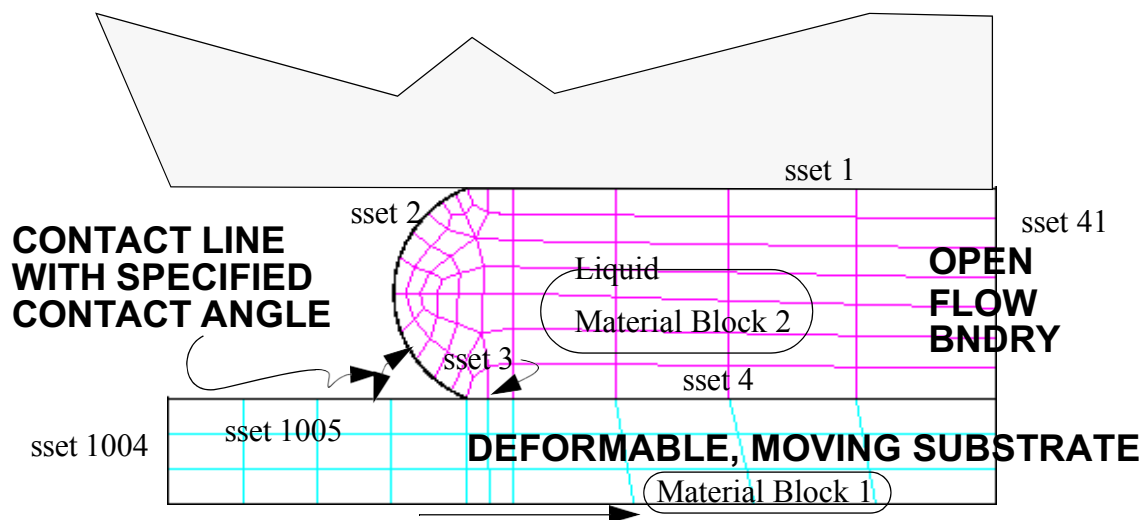
The conclusion is that this set of conditions should be used in place of the NO_SLIP_RS condition whenever possible. *The caveat to this is that the side set between the two materials must be attached to the liquid phase only, for the VELO_NORMAL condition to be applied, and to both phases for the VELO_TANGENT_SOLID to be applied. This requires either two side sets OR an alternative is to make the fluid material the first material block in the mesh, as GOMA automatically associates VELO_NORMAL for double sided side sets to that block.*

Third important ASIDE:

Make sure pseudo-solid Lamé coefficients are the same for both materials. Setting these mesh parameters to be largely different from material region to material region results in severe mesh shearing on the interface.

Deformable Substrate with a Moving Contact Line

The purpose of this example is to show the user how to deal with dynamic contact lines with a potential slip region on deformable surfaces in STEADY STATE. The test problem we will examine is available in the templates under **TALE.tst** and is called “**contact_line**”. It basically is a heal formation zone (a.k.a. the upstream meniscus of a bead coating flow) on a moving deformable substrate. The following figure illustrates the key elements:



Other key side/node sets:

- sset 3 is used for the slip region on the substrate.
- sset 4 is the no-slip part of the substrate
- sset 1005 is the unwetted portion of the substrate
- nset 100 is the dynamic contact line
- nset 200 is the static contact line

Distribution

Some of the key node sets and side sets are shown in this illustration. The first thing to notice is that the grid must now extend out beyond the dynamic contact line for the unwetted part of the substrate. Moreover, the stresses in the substrate there are important as they will influence the fluid-solid stress balance on the wetted portion. Side set 3 is a short side set that is used to apply a perfect slip condition. The choice of its length and the value of the slip coefficient chosen here was not based on any scientific reason. In fact, setting the velocity components to zero at the contact line and excluding the slip region would yield nearly the same result. The setup of the material files and the input file are similar to the roll coating cases. In this case we are allowing the substrate to be compressible. To this end, the substrate problem description portion of the input deck looks like

```
Number of Materials = 2
```

```
MAT = substrate 1
```

```
Coordinate System = CARTESIAN
```

```
Element Mapping = isoparametric
```

```
Mesh Motion = TOTAL_ALE
```

```
Number of bulk species = 0
```

```
Number of EQ          = 4
```

```
EQ = mesh1  Q2  D1  Q2  0. 0. 1. 1.      0. 0.
EQ = mesh2  Q2  D2  Q2  0. 0. 1. 1.      0. 0.
EQ = mom_solid1 Q2  D1_RS  Q2  0. 0. 1.      1.  0. 0.
EQ = mom_solid2 Q2  D2_RS  Q2  0. 0. 1.      1.  0. 0.
```

Notice again the `mom_solid` equations for the real-solid description. The mesh motion, or in this case the material type, is `TOTAL_ALE`. Correspondingly, for this material the top portion of the `substrate.mat` file looks like:

```
---Physical Properties
```

```
Density          = CONSTANT  1000.
```

```
---Mechanical Properties and Constitutive Equations
```

```
Solid Constitutive Equation  = INCOMP_PSTRAIN
```

```
#Solid Constitutive Equation = HOOKEAN_PSTRAIN
```

```
Convective Lagrangian Velocity = CONSTANT 0.133 0. 0.
```

```
Lame MU          = CONSTANT  1.4e9
```

```
#Lame LAMBDA     = CONSTANT  0.
```

```
Lame LAMBDA     = CONSTANT  0.
```

```
Stress Free Solvent Vol Frac = CONSTANT  0.0
```

```
Pseudo-Solid Constitutive Equation  = NONLINEAR
```

```
Pseudo-Solid Convective Lagrangian Velocity = NONE
```

```
Pseudo-Solid Lamé MU          = CONSTANT  100000.
```

```
Pseudo-Solid Lamé LAMBDA     = CONSTANT  100000.
```

Notice that again we have a section which specifies the pseudo-solid properties following that which specifies the real solid properties. We are also starting with a high shear modulus.

The boundary condition section of the `input` file is also important:

Distribution

Boundary Condition Specifications

```

---
Number of BC          = -1
BC                    = PLANE SS 1 0. 1. 0 -0.0005
BC                    = PLANE SS 1001 0. 1. 0. 0.00025
BC                    = PLANE SS 41 1. 0. 0. 0.000125
BC = U   NS   1   0.0
BC = V   NS   1   0.0
BC = U   NS  200  0.0
BC = V   NS  200  0.0

```

```

BC = DX NS 1004 0.
BC = DY NS 1004 0.
BC = DX_RS NS 1004 0.
BC = DY_RS NS 1004 0.
BC = DX NS 1000 0.

```

```

$
$ NOTE: KIN_DISPLACEMENT MUST
$ PRECEDE KINEMATIC in the
$ BC order here in order for
$ proper behavior on this
$ particular case. It is the
$ order rule that is enforced
$ here because of all of the others
$ failing. In the fixed rigid roll
$ case, the PLANE command would overtake
$ kinematic regardless of the order
$ BECAUSE IT IS A STRONG_COLLOC
$

```

```

$$BC = NO_SLIP_RS SS 4 1 2
$$BC = VELO_SLIP_SOLID SS 4 1 2 0.001 248 10000.
$$BC = VELO_SLIP_SOLID SS 4 1 2 0.00001
BC = VELO_TANGENT_SOLID SS 4 1 2
BC = VELO_NORMAL SS 4 0.

```

```

BC = SOLID_FLUID_RS SS 4 1 2
BC = KIN_DISPLACEMENT SS 4 1

```

```

$$BC = VELO_NORMAL SS 2 0.
BC = KINEMATIC SS 2 0.
BC = CAPILLARY SS 2 0.065 -4000.0 0.
BC = CA NS 100 1.000 0.0 -1.0 0.0
BC = U NS 100 0.
BC = V NS 100 0.
BC = CA NS 200 0.600 0.0 1.0 0.0

```

```

$$
$$ This condition is needed on the unwetted substrate
$$ to keep the pseudo-solid displacements conformed to the
$$ real substrate solid, otherwise strange bumps occur
$$
BC = KIN_DISPLACEMENT SS 1005 1

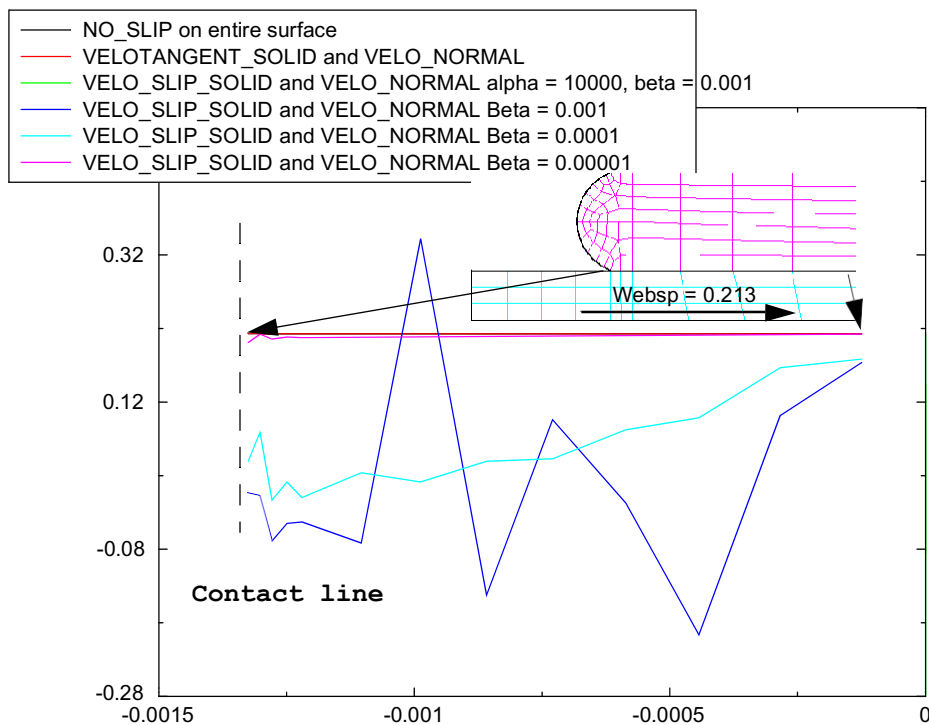
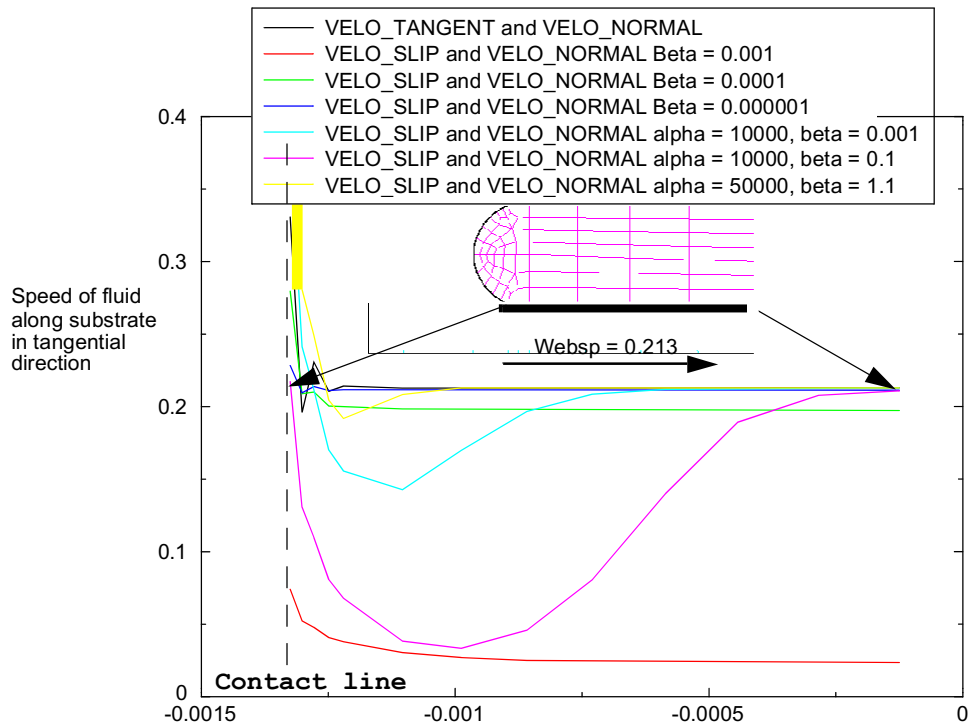
```

```

END OF BC

```

The **DX_RS** and **DY_RS** cards are used on surfaces and side sets where the real solid is not allowed to slip. Note also that we specify **DX** and **DY** of the pseudo-solid to be zero at the same location (node set



Substrate liquid velocity profiles with different slip models and slip coefficients. Comparison between rigid solid and deformable solid cases.

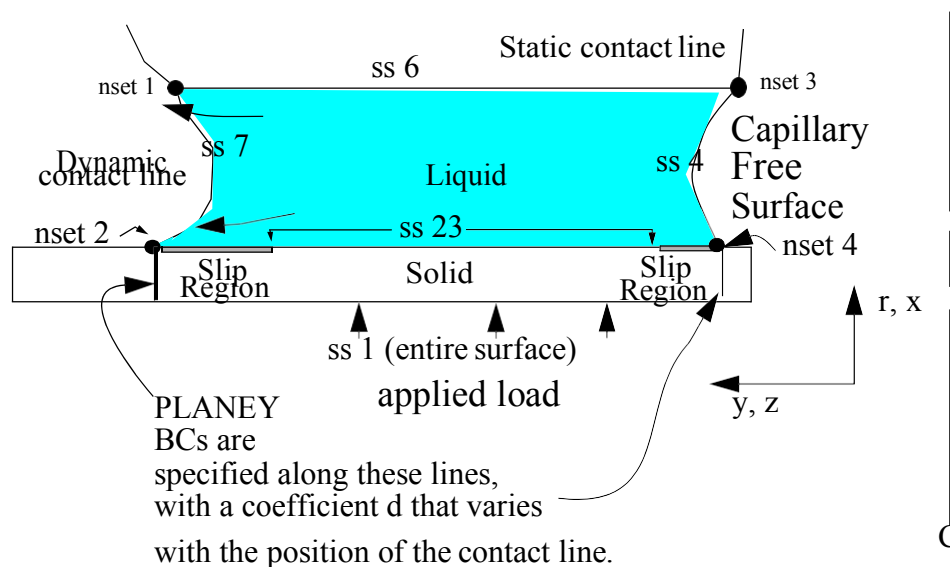
Distribution

- 5 Run `goma`. This problem converges without relaxation.
- 6 Again save the solution (`cp soln.dat contin.dat`) and change the backpressure to -6000. Rerun `goma` and notice how the meniscus moves along the deformable substrate. Contouring stresses in the for the real solid and pseudo solid are instructive.
- 7 Now experiment with the slip conditions, and their coefficients. At the bottom of the input file you will see that GOMA is being instructed to output velocity profile data along the substrate. The results of plotting those data are shown in the figure.

Transient contact line problems--capillary wetting in braze flow

The problem discussed in this section is intended to give the user some guidance on moving contact lines in transient situations. In the previous example, we discussed steady state dynamic contact lines, and associated slip conditions. In this example we give some rules-of-thumb for moving contact line problems. The problem discussed in this section is in the directory “`capillary_wetting`” under the `TALE.tst`.

Consider the following axisymmetric, annular capillary wetting flow, which is flow regime encountered often in brazing and soldering problems.



Basically, we impulsively start the wetting process from a quiescent state, with 90 degree contact angles. At that instant, the contact angles are all set to 70 degrees. Along the top we are modeling the surface as a rigid stationary solid, and hence there is no need to model the stresses in the solid. That surface is described with the `GD_TABLE` card as a set of discrete points. It actually contains a flat land and a flange on the left hand side. A load is applied to the bottom solid, which is treated with `TALE`. To allow for large mesh deformations which will accompany significant capillary wetting, we deliberately move the mesh with `PLANEY` conditions internal to the solid, along two lines

Distribution

perpendicular to the substrate, with an augmenting conditions (see figure). We are free to manipulate the mesh like this without affecting the solid stresses. The benefit this provides is reduced mesh distortion, as the mesh in the solid to the left and right of the contact lines compresses uniformly with these moving planes. Without these augmenting conditions, the mesh is sheared in the solid to the point at which excessive distortion leads to failure.

The relevant files in this test directory are:

4node_input, **cu.mat**, and **alumina_ceramic.mat**.

The mesh comes from PATRAN, and so we will not be discussing its generation here. First, in the input file for this example, **4node_input**, we see that this is a transient problem with the following material problem description section:

```

----
Problem Description
---

Number of Materials = 2

MAT = cu 1

Coordinate System    = CYLINDRICAL
Element Mapping      = isoparametric
Mesh Motion          = ARBITRARY
Number of bulk species = 0
  Number of EQ       = 5
  EQ = mesh1    Q1    D1    Q1 1. 0. 1. 1. 0. 0.
  EQ = mesh2    Q1    D2    Q1 1. 0. 1. 1. 0. 0.
  EQ = momentum1 Q1    U1    Q1 1. 0. 1. 1. 0. 0.
  EQ = momentum2 Q1    U2    Q1 1. 0. 1. 1. 0. 0.
  EQ = continuity Q1    P    Q1 1. 0.

```

```

MAT = alumina_ceramic_TALE 2

```

```

Coordinate System    = CYLINDRICAL
Element Mapping      = isoparametric
Mesh Motion          = TOTAL_ALE
Number of bulk species = 0
  Number of EQ       = 4
  EQ = mesh1    Q1    D1    Q1 1. 0. 1. 1. 0. 0.
  EQ = mesh2    Q1    D2    Q1 1. 0. 1. 1. 0. 0.
  EQ = mom_solid1 Q1    D1_RS Q1 1. 0. 1. 1. 0. 0.
  EQ = mom_solid2 Q1    D2_RS Q1 1. 0. 1. 1. 0. 0.

```

```

Post Processing Specifications

```

Notice that the alumina ceramic material is a **TOTAL_ALE** mesh motion, with the usual specifications of mesh equations and momentum equations. This problem is axisymmetric, hence the **CYLINDRICAL** specification on the **Coordinate System** card. The boundary conditions deserve most of the attention here, and the rest of this tutorial will discuss them. We will examine this in sections, pointing out the relevant features along the way.

```

Number of BC          = -1

```

Distribution

```
BC = FORCE_RS    SS 1 19.e3 0. 0.
BC = SOLID_FLUID_RS SS 16 2 1
```

```
BC = NO_SLIP_RS SS 23 2 1
```

```
BC = VELO_NORMAL SS 6 0.
```

This first group of cards sets the applied load on the substrate with **FORCE_RS**, at 19,000 dyn/cm² in the +x direction (actually z direction). The second card **SOLID_FLUID** is the usual balance of fluid and solid tractions at the interface. It is applied to side set 16 which spans the entire wetted solid, and includes both materials. Note that the solid material number is 2, and the liquid material number is 1. The third card, on SS 23, sets the fluid velocity equal to the solid velocity. Side set 23 does not span the entire wetted region, as shown in the figure. We apply slip to the top rigid solid surface (sset 6) with a simple **VELO_NORMAL** and **VELO_SLIP** (see below) BCs.

The next batch of BCs are

```
BC = PLANEY SS 14 0 1 0 0.0
BC = PLANEY SS 15 0 1 0 0.0
BC = PLANEY SS 24 0 1 0 0.0
BC = PLANEY SS 25 0 1 0 0.0
```

```
BC = VELO_SLIP SS 6 0.001 0. 0. 0.
```

```
BC = VELO_NORMAL    SS 11 0.
BC = VELO_NORMAL    SS 12 0.
```

The **VELO_NORMAL** BCs on SS 11 and 12 are for perfect slip in those regions, as no specification on tangential velocity is made there. The **VELO_SLIP** condition is applied to SS 6, as pointed out above. The four **PLANEY** BCs are used to move the mesh within the solid to accommodate large motions, as pointed out above. They are NOT applied independent of the solution, but are actually tied to the solution through the augmenting condition section of the input deck, viz.

```
---
```

```
Augmenting Conditions Specifications
Number of augmenting conditions = -1
AC = BC 4 3
AC = BC 5 3
AC = BC 6 3
AC = BC 7 3
END OF AC
```

These are four constraints that are solved together with the physics at all time steps. They basically say that boundary condition numbers 4,5,6, and 7, the count starting from 0, are to be altered to satisfy conditions which are specified in the **user_ac.c** subroutine. In that subroutine is an excerpt like:

```
i = index_solution(1, MESH_DISPLACEMENT2, 0, ndof);
AC[0] = -BC_Types[4].BC_Data_Float[3]-(Coor[1][1] + x[i] - dist1);
j = index_solution(131, MESH_DISPLACEMENT2, 0, ndof);
AC[1] = -BC_Types[5].BC_Data_Float[3]-(Coor[1][131] + x[j] + dist2);
k = index_solution(1, MESH_DISPLACEMENT2, 0, ndof);
AC[2] = -BC_Types[6].BC_Data_Float[3]-(Coor[1][1] + x[k] + dist3);
```

Distribution

```
l = index_solution(131, MESH_DISPLACEMENT2, 0, ndof);
AC[3] = -BC_Types[7].BC_Data_Float[3] - (Coor[1][131] + x[1] - dist4);
```

These are the corresponding augmenting conditions. Node numbers 1 and 131 correspond to the two contact lines on the TALE solid, determined from the “**number**” toggle in **BLOT**’s **DETOUR** program. The conditions basically determine the “d” intercept coefficient of the **PLANEY** BCs, so that they move to remain the same distance from the moving contact lines. Those distances are specified in the subroutine. The setup of augmenting conditions is described in further detail in GSR-01.0 and the building of GOMA with user-defined routines in GM2-03.3 and GDM-1.0.

IMPORTANT ASIDE: Notice also that the **PLANEY** conditions are used instead of **PLANE** conditions. The reason for this is that GOMA cannot do internal mesh residual rotations on geometry commands, because of either (1) the side set does not contain both sides of the interface, and (2) the block id on either side of the side set is the same. GOMA’s logic has not been furnished to handle this case yet (10 August 1999).

Continuing on with the BCs in the input file:

```
$ BC = PLANE    SS 6 1. 0. 0. 0.
BC = GD_LINEAR SS 6 R_MESH_NORMAL 0 MESH_POSITION1 0 0. -1.
BC = GD_TABLE  SS 6 R_MESH_NORMAL 0 MESH_POSITION2 0 1.0 LINEAR
$   Y    X
0.15 0.0
0.2316 0.0
0.4560 0.0
0.4568 0.0001
0.4575 0.0004
0.4581 0.0009
0.4699 0.0100
0.4704 0.0106
0.4707 0.0113
0.4708 0.0121
0.4709 0.0400
END TABLE
```

This is a nice way to specify the geometry of a complex shape. The **GD_TABLE/GD_LINEAR** combination here is used to specify the geometry of side set 6. The next few cards deal with the free surfaces and contact lines, and are very tricky.

```
BC = CAP_REPULSE SS 4 1285. 0. 0.001 0. 1. 0. {-18}
$$$$$NEED TO MOVE THIS BACK DOWN IF YOU HAVE RELEASED THIS MENISCUS
$$ BC = KIN_DISPLACEMENT SS 9 2
BC = KINEMATIC SS 4 0.
BC = CAPILLARY SS 7 1285. 0. 0.
BC = KINEMATIC SS 7 0.
```

These are your basic specifications for capillary free surfaces. The **CAP_REPULSE** on SS 4 is a special BC that allows for a capillary surface that is repulsed by a force which grows large near a plane, defined by the last four coefficients. The corresponding **KINEMATIC** condition on that side set is the

Distribution

typical distinguishing condition need for these sorts of free surfaces. The same **CAPILLARY/KINEMATIC** pair is applied to SS 7. The next three conditions:

```
BC = KIN_DISPLACEMENT SS 16 2
BC = KIN_DISPLACEMENT SS 3 2
BC = KIN_DISPLACEMENT SS 9 2
```

act as distinguishing conditions on the solid substrate surface. These demand that the mesh not penetrate the real solid surface. SS 16 is the wetted substrate; SSs 3 and 9 are the unwetted portions. It is important to note that at least one of these conditions must be applied at the contact line nodes, together with one **KINEMATIC** condition from the liquid surface. BOTH are required and GOMA is sensitive to the ordering of the equations. More discussion of this will be taken up below.

The next part of the deck specifies the appropriate BCs at the contact lines. You can use CA conditions as are shown here but commented out:

```
$ BC = CA      NS 2 {3.14159 - 2.1} 0.0 0.00.0
$ BC = CA      NS 4 {3.14159 - 2.1} 0.0 0.00.0
$ BC = CA      NS 2 {3.14159 - 1.9} 1.0 0.00.0
$ BC = CA      NS 4 {3.14159 - 1.9} 1.0 0.00.0
```

We strongly urge you not to use these on real solid surfaces in TRANSIENT problems. The problem is that they are a “hard set”, and need to be applied to the mesh motion equations. In doing so, they displace the **KINEMATIC** equations above, thereby taking away the proper kinematic description necessary to allow the line to move. i.e., the kinematic condition equations the mesh motion to the material motion, and if displaced, the liquid or solid will leak mass at that line. This is a moot point in STEADY problems because the velocity is set to zero at these lines and all necessary mass constraints are satisfied. Node sets 2 and 4 correspond to the solid surface contact lines. At them we specify a wetting angle through a force (see theory manual of GOMA or TALE for explanation), applied with the **SURFTANG** cards:

```
$$ wetting
  BC = SURFTANG NS 2 -0.6428 0.766 0 1285.
$$ nonwetting
$$ BC = SURFTANG NS 2 -0.6428 -0.766 0 1285.
$ BC = FIX NS 1 MESH_DISPLACEMENT1 0
$ BC = FIX NS 1 MESH_DISPLACEMENT2 0
$ BC = FIX NS 2 MESH_DISPLACEMENT2 0
$ BC = U NS 1 0.
$ BC = V NS 1 0.
$ BC = U NS 2 0.
$ BC = V NS 2 0.
$$wetting
$ BC = SURFTANG NS 4 -0.6428 -0.766 0 1285.
$$nonwetting
$$ BC = SURFTANG NS 4 -0.6428 0.766 0 1285.
$$neutral
  BC = SURFTANG NS 4 -1. 0. 0 1285.
```

Notice how several options are commented out here. We have wetting angle specified on NS 2 and a neutral angle (90 deg) specified on nset 4. The other options show how one changes the angle of the wetting, and how to arrest a line during a restart (viz. with the **FIX** commands). In this braze flow problem, the wetting lines eventually run into a chemically treated portion of the surface at which

Distribution

they arrest. Because of the force applied to the substrate, however, the menisci continue to bow out and the substrate squeeze in. When simulating this, we can restart the problem with these **FIX** commands on the mesh displacement. CARE must be taken here to retain the **KIN_DISPLACEMENT** bc at that node, and to do that you need to move them ahead of the **KINEMATIC** BCs there, as shown above. Otherwise, the mesh will penetrate the solid there.

The remaining boundary conditions are used for the contact lines on the rigid top surface, and the end points of the solid domain. These are fairly standard, but not too restrictive. Note how they allow for the solid to squeeze inward:

```

$$ wetting
  BC = SURFTANG NS 1 0.6428 -0.766 0 1285.
$$ nonwetting
$$ BC = SURFTANG NS 1 0.6428 0.766 0 1285.

$$wetting
$ BC = SURFTANG NS 3 0.6428 0.766 0 1285.
$$nonwetting
$$ BC = SURFTANG NS 3 0.6428 -0.766 0 1285.
$$neutral
  BC = SURFTANG NS 3 1. 0. 0 1285.
  BC = DY_RS    NS 6 0.
  BC = DY_RS    NS 7 0.
  BC = DY       NS 7 0.
  BC = KIN_DISPLACEMENT SS 1 2

END OF BC

```

Some sample results are shown below at different times. Note the mesh movement in the solid.

IMPORTANT ASIDE: It is critical in these problems to look at the **BCdup.txt** file on the output of a GOMA run. Locate the nodes in there that correspond to the contact lines, and reason out that what is being applied there is correct. For this problem, the way it is set up, it looks like this:

```

At node 1:
  R_MOMENTUM1 gets SURFTANG    from NS 1, VELO_NORMAL    from SS 6, VELO
  _SLIP      from SS 6, CAPILLARY    from SS 7,
  R_MOMENTUM2 gets SURFTANG    from NS 1, VELO_SLIP      from SS 6, CAPI
  LLARY      from SS 7,
  R_MESH1    gets GD_LINEAR     from SS 6, GD_TABLE     from SS 6,
  R_MESH2    gets KINEMATIC     from SS 7,

```

```

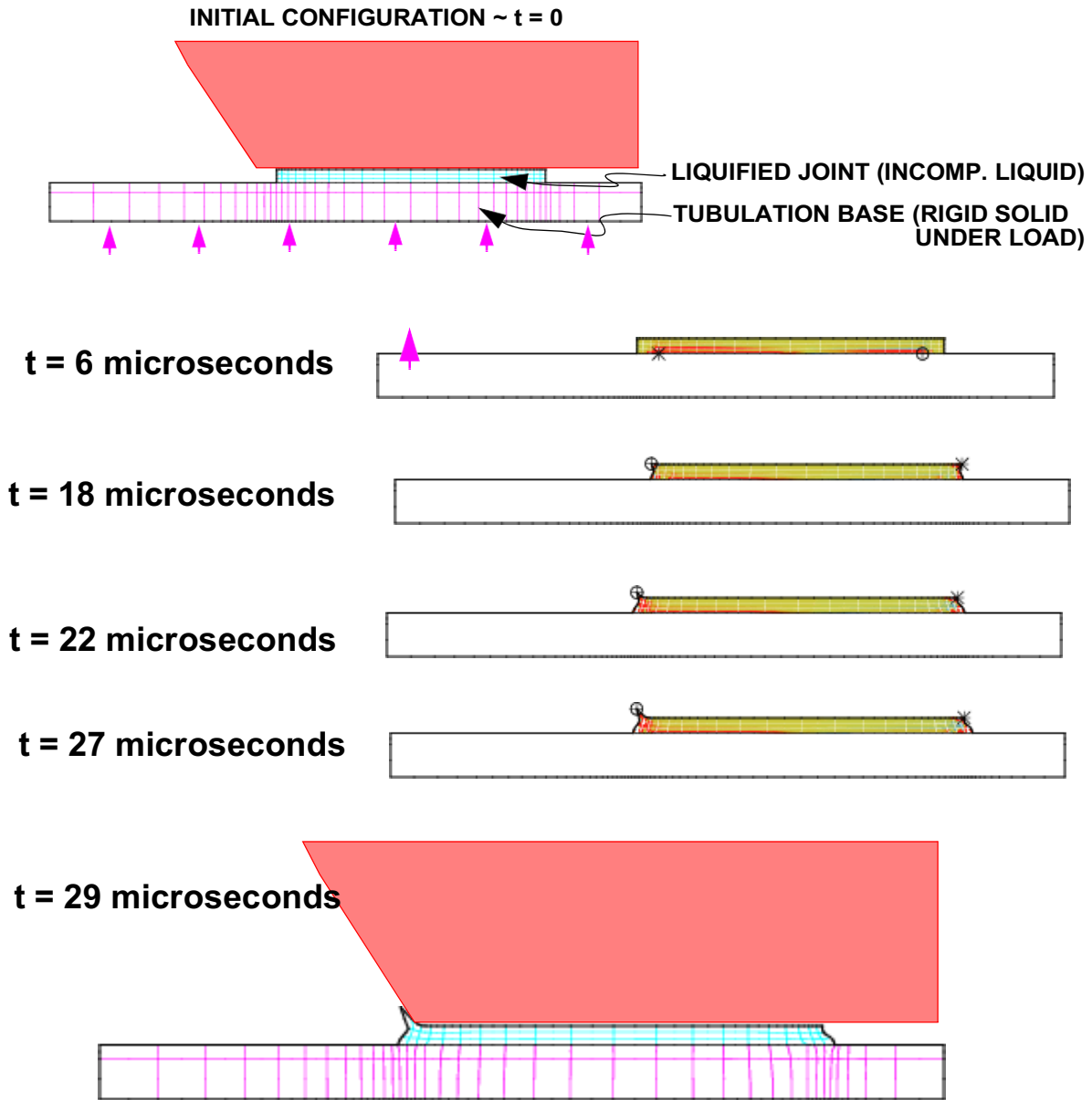
At node 2:
  R_MOMENTUM1 gets SURFTANG    from NS 2, VELO_NORMAL    from SS 11, CAP
  ILLARY     from SS 7,
  R_MOMENTUM2 gets SURFTANG    from NS 2, CAPILLARY     from SS 7,
  R_MESH1    gets KINEMATIC     from SS 7,
  R_MESH2    gets KIN_DISPLACEMENT from SS 16,
  R_SOLID1   gets SOLID_FLUID_RS from SS 16,
  R_SOLID2   gets SOLID_FLUID_RS from SS 16,

```

```

At node 132:

```



R_MOMENTUM1 gets SURFTANG from NS 4, VELO_NORMAL from SS 12, CAP
 _REPULSE from SS 4,
 R_MOMENTUM2 gets SURFTANG from NS 4, CAP_REPULSE from SS 4,
 R_MESH1 gets KINEMATIC from SS 4,
 R_MESH2 gets KIN_DISPLACEMENT from SS 16,

Distribution

APPENDIX A: Modifications to KIN_DISPLACEMENT BC

I am attempting to check-in improvements we have made into the Goma svn repository after a rather long hiatus. While most of the code modifications will not affect Goma runs, there are some exceptions. Foremost among these is the case of TALE calculations.

It was discovered that the reference coordinates for the real solid displacements was not correct as evidenced by the fact that the reference locations (computed by subtracting the real solid displacements from the current deformed coordinates) deviated substantially from the original undeformed geometry. This discrepancy is most obvious in the case of curved geometry, i.e. rolls, and may be of little consequence for planar TALE applications.

To remedy this, the current 3M version of Goma adjusts the real solid displacements so that the reference location is where it should be. This requires knowledge of the original geometry. Some geometry information is available through the parameters in the Convected Lagrangian Velocity model. But the missing geometry parameters need to be specified with the KIN_DISPLACEMENT boundary condition. For rolls, the Convective Lagrangian Velocity model will be ROTATIONAL or ROTATIONAL_3D and the only missing geometry parameter is the roll radius. Hence, the KIN_DISPLACEMENT boundary condition will now look as follows;

```
BC = KIN_DISPLACEMENT SS 2 1 {R_O}
```

For planar TALE geometries, the Convective Lagrangian Velocity model will be CONSTANT or may be absent and the four plane parameters specifying the undeformed solid-liquid interface will need to be listed on the KIN_DISPLACEMENT boundary condition.

```
BC = KIN_DISPLACEMENT SS 2 1 {plane_A} {plane_B} {plane_C} {plane_D}
```