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

Albuquerque, New Mexico 87185-0826

date: 1 September, 2002
(original 1-29-99; revised 8/17/1999; revised 5/1/2002; revised 9/1/2002)

to: Distribution

from: P. R. Schunk, 9114, MS0834

subject: *GOMA*'S CAPABILITIES FOR ***PARTIALLY SATURATED FLOW*** IN POROUS MEDIA (GT-009.3)

keywords: porous media, deformable porous media, impregnation, saturation models, poroelasticity, swelling, saturation hysteresis, Media Type, Permeability, Porosity, Saturation, Rel Liq Permeability, Rel Gas Permeability.

input records: Media Type, All porous-media related material-property cards, All porous media-related boundary conditions, POROUS_PRESSURE, DARCY_CONTINUOUS

Introduction

This tutorial will be most effective if the user has gone through the beginner's training tutorial on GOMA and SEAMS (GT-001.4) and the Porous Penetration tutorial (GT-008.2). Copies of those tutorials can be obtained through Duane Labreche (dalabre@sandia.gov).

GOMA has extensive capability for modeling transport phenomena in porous media, including several formulations of the porous transport equations for saturated and unsaturated flow, a variety of porous-medium-related boundary and interfacial conditions, and even poroelastic stress constitutive equations using the effective stress principle. Reports and papers by Gartling, et al. (1996), Schunk, et al. (1998) and Cairncross, et al. (1996) are important to the discussion here. The first gives a comprehensive overview and detailed study of the computational issues surrounding coupled fluid and porous flow problems. That report details the comparison between a Darcy formulation and a Brinkman formulation for flow in porous materials, and the implications of each formulation to the interfacial boundary conditions between porous and continuous phases. Those boundary conditions are central to this tutorial. The second report describes most of the boundary conditions and the governing equations available in GOMA 4.0 for modeling transport in porous media. However, because that report is a generalized user's manual for GOMA, the information is spread out and not detailed. It is recommended that the user consult the new user's manual that is to be released in conjunction with GOMA 4.0 (contact Duane Labreche, dalabre@sandia.gov or Randy Schunk prschun@sandia.gov). The third report by Cairncross et al. (1996) describes in detail the equations, boundary conditions, constitutive relations, and even the necessary GOMA input for a partially-saturated poroelastic sol-gel dip coating process. That report is extremely comprehensive with regard to that problem, but has little or no information regarding other situations often encountered in manufacturing and technology. Moreover, this report follows up with updates to the equations. At the minimum, please read the report by Cairncross, et al. (1996) as it covers the necessary basics of the problems addressed in this memo. In addition, it is highly recommended that before attempting unsaturated porous media problems that you consult some of the classical references on the subject. Examples include books by Baer (1972) and Scheidegger (1960). *Please note that GOMA 4.0 and* *subsequent releases contain several new boundary conditions and material property models discussed here. It is recommended that you install that revision before proceeding.*

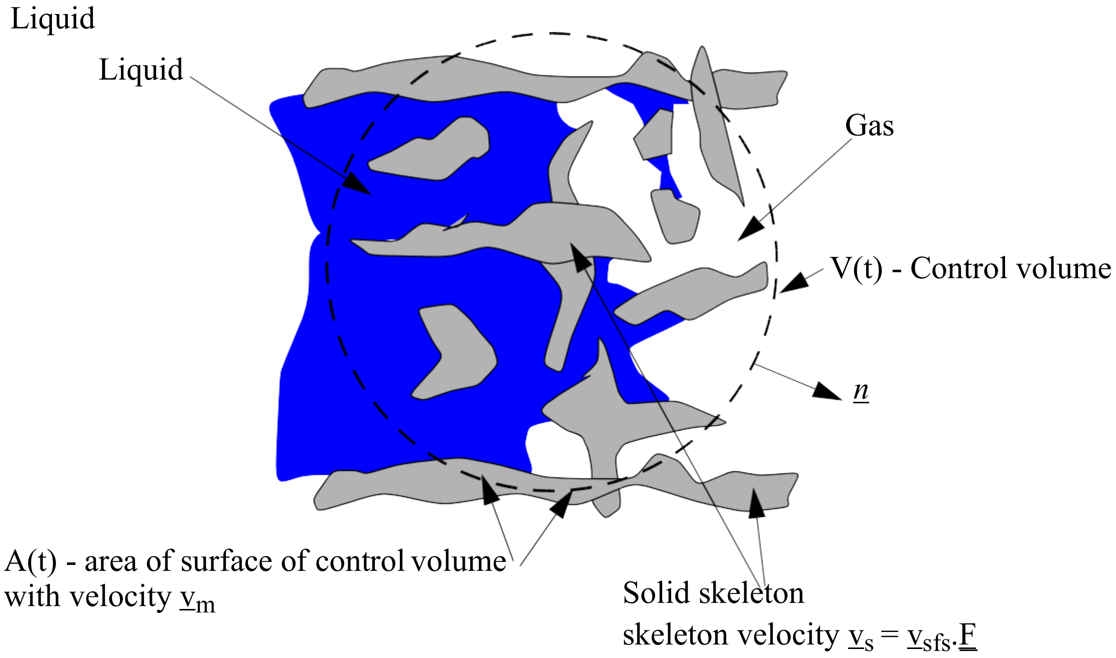
It is the author's extensive experience that these problems are challenging numerically, due to rapidly advancing and sharp saturation fronts, and conceptually, due to the necessary but awkward formulations used. The formulation in GOMA is no exception. The concepts of how liquid phase pressures, gas-phase pressures, vapor-pressure, capillary pressure, and concentrations all interrelate to determine saturation at any point in the medium are crucial to success in modeling. The first section in this report is designed to acquaint the user with these concepts.

The second part of this report addresses the need for tutorial problems involving fluid and species transport in porous media for manufacturing processes. Two important applications with respect to GOMA involve porous penetration in continuous liquid film coating and metals joining processes; in these and other cases, a porous medium is imbibing or expelling fluid from an adjacent layer (depending on the external penetration pressure or internal elastic network stress). In a previous memo (GT-008.2), we focused on the saturated media case and covered in detail how situations of saturated porous media with bounding continuous fluid phases are modeled with GOMA. In this memo, we extend the same concepts to the partially-saturated case. We address six one-dimensional sample problems which provide the understanding the user will need to attack more complex multidimensional problems.

MATERIAL COVERED IN THIS DOCUMENT:

To run the problems discussed herein, you need the following directories: `saturated_squish, 1D_hysteresis, sat_wick, porous_plug, simple_sat_flow_tsts, porous_film_dry, stretch_bend` `and` `unsat_squish.` *These directories of problems are all underneath a master directory named* `porous_tests`*. These can be by contacting the GOMA team.*

Current Capability for Modeling Coupled Fluid Flow and Partially Saturated Porous Flow in Deformable Media Problems with GOMA



For problems involving partially-saturated flow in porous media a general conservation equation is written for each species in the system. The peculiarity of a Darcy approach is that all balances are statements of species conservation, and the forces and dynamics enter in through the constitutive equations. Hereafter we refer to the "solvent" as the primary component in each of three phases, viz. solid, liquid, and gas, and in fact limit our discussion to these three phases for the time being. We further restrict our formulation to a single base liquid phase component (e.g. water) and a single gas-phase "solvent" component (e.g. air) which is insoluble in the liquid and solid, and a single solid phase component, hereafter referred to as the "solid solvent". The solid is also taken as insoluble in the gas and liquid phases.

We begin first with the saturated case, for which we've only two components: liquid and solid. Because there is no gas, and hence no vapor phase of the liquid solvent, the liquid inventory locally is simply $\phi\rho_l$, where ϕ is the porosity and ρ_l is the pure liquid density.

With the control volume pictured above, we can write a Liquid Solvent balance as:

IN - OUT = Accumulation

or, mathematically as

$$-\int_{A(t)} (n \cdot F_l)dA = \frac{d}{dt}\int_{V(t)} \phi\rho_l dV \qquad \text{(EQ 1)}$$

Here F_l is the mass flux of liquid (M/L^2/t) *relative* to the control volume. Note that the control volume can be moving relative to a chosen inertial frame, as indicated by the velocity V(t). Also note that the porous skeleton can be moving at a different velocity $\underline{v}_s = \underline{v}_{sfs}.\underline{\underline{F}}$, where $\underline{v}_{sfs}$ is the stress free state velocity and $\underline{\underline{F}}$ is the deformation gradient tensor. The skeleton can be moving due to surface forces or body forces, or in a user-prescribed way, e.g., as would be the case in a translating porous web.

We will apply the Transport Theorem to this previous equation to get:

$$-\int_{A(t)} (n \cdot F_l)dA = \int_{V(t)} \frac{\partial}{\partial t}\phi\rho_l dV + \oint_{A(t)} (n \cdot v_m)\phi\rho_l dA \qquad \text{(EQ 2)}$$

v_m is the velocity of the control volume. Note that this control volume velocity can be thought of as a finite element mesh velocity, viz. V(t). Invoking a Lagrangian description with respect to the solid skeleton, this velocity will also be equivalent to the material skeleton velocity, an assumption that we will invoke later.

Apply the Green-Gauss-Ostrogoski (GGO) theorem, viz. the divergence theorem, to the area integrals to get:

$$-\int_{V(t)} (\nabla \cdot F_l)dV = \int_{V(t)} \frac{\partial}{\partial t}\phi\rho_l dV + \int_{V(t)} \nabla \cdot (v_m\phi\rho_l)dV \qquad \text{(EQ 3)}$$

Divide by the volume, invoke the theorem of the mean, and passing to V->0 following the motion v_m:

$$\frac{\partial}{\partial t}(\phi\rho_l) + \nabla \cdot (v_m \phi\rho_l) = -(\nabla \cdot F_l) \qquad \text{(EQ 4)}$$

The flux of liquid relative to the control volume V(t) can be written as:

$$\underset{\sim}{F}_l = \rho_l \underset{\sim}{v}_l = (\underset{\sim}{v}_s)\phi\rho_l - \rho_l\frac{\kappa}{\mu}\nabla(p_l + \rho_l g) \qquad \text{(EQ 5)}$$

The first term on the right is the flux of liquid relative to the control volume due to the superimposed motion of the stress-free-state of the solid skeleton, an option in GOMA that is used only when the frame of reference is changed relative to the skeleton due to some conjugate phase (e.g. a continuous phase or a superimposed force). The second term is the relative motion due to liquid potential gradient, as prescribed by Darcy’s law, written here for a saturated medium.

In any algorithm that involves a moving mesh, we must make provisions for the evaluation of the time derivatives. The “control volume” we speak of here is taken as the finite element mesh, which moves in general at a velocity $\underline{v}_m$. The mesh in GOMA is taken to move with the skeleton material for the poroelastic modeling, and so it is “Lagrangian” only with respect to that material. The time derivative in a fixed frame above is really evaluated in GOMA in a moving reference frame as the mesh is moving (viz. the node points are moving). Hence, we need to transform the stationary time derivative into something GOMA can evaluate, using again the transport theorem:

$$\frac{\partial}{\partial t}(\phi\rho_l) = \frac{d}{dt}(\phi\rho_l) - v_m \cdot \nabla\phi\rho_l \qquad \text{(EQ 6)}$$

Here the first derivative in time is from a stationary view point and the time derivative on the right hand side is from a viewpoint of motion $\underline{v}_m$. Note that neither of these are following the material motion. Combining the last three equations gives our final liquid solvent balance as it occurs in GOMA, for a saturated porous flow:

$$\frac{d}{dt}(\phi\rho_l) + \rho_l\phi(\nabla \cdot \underset{\sim}{v}_m) = \nabla \cdot \left(-\underset{\sim}{v}_s\phi\rho_l + \rho_l\frac{\kappa}{\mu}\nabla(p_l + \rho_l g)\right) \qquad \text{(EQ 7)}$$

In a fully saturated system there are two components, viz. liquid and solid. We need to solve together with this equation either an overall mass balance or another component balance. For a rigid porous medium the inventory of solid stays constant and hence the quantity that describes its concentration is the porosity. In a deformable medium, the concentration of solid (viz. $1-\phi$) is related to the kinematics of the solid deformation, which is embodied in the deformation gradient tensor:

$$\underset{\sim}{F} \equiv \frac{d\underset{\sim}{x}}{d\underset{\sim}{X}} \qquad \text{(EQ 8)}$$

where $\underline{x}$ denotes the coordinates of the deformed configuration of a material point, and $\underline{X}$ denotes the coordinates of the same material point in the reference state. Because we treat the control volume, or mesh, as moving with the solid skeleton in a Lagrangian fashion and because we assume an affine deformation of the skeleton in the control volume, viz. the volume change of the skeleton is proportional to the volume change of the control volume, then the component balance can be written as

$$det(\underset{\sim}{F}) = (1-\phi_0)/(1-\phi) \qquad \text{(EQ 9)}$$

The fact that det($\underline{F}$) is independent of time is a statement that the total material composite is incompressible, with the interstitial liquid and the solid struts being incompressible. We can derive this equation in another way starting with a similar integral flux balance for the solid, with respect to the moving control volume:

$$-\int_{A(t)} (n \cdot \underset{\sim}{F}_s)dA = \int_{V(t)} \frac{\partial}{\partial t}(1-\phi)\rho_s dV + \oint_{A(t)} (n \cdot \underset{\sim}{v}_m)(1-\phi)\rho_s dA \qquad \text{(EQ 10)}$$

Here ρ_s is the density of the solid in the network, and F_s is the flux of solid relative to the moving control volume. As stated above, the control volume, or mesh, is assumed to move with the solid network, viz. in a Lagrangian fashion, and so we $\underline{F}_s$ is zero, unless we account for the stress free state motion that is superimposed:

$$\underset{\sim}{F}_s = (\underset{\sim}{v}_s)(1-\phi)\rho_s \qquad \text{(EQ 11)}$$

Distilling these two equations down to a differential equation, using the theorem of the mean and the Lagrangian assertion described above, the right-most integral is identically zero (no net mass flux of solid out of the control volume), and we have:

$$\frac{\partial}{\partial t}((1-\phi)\rho_s) + \nabla \cdot (v_s(1-\phi)\rho_s) = 0 \qquad \text{(EQ 12)}$$

If $\underline{v}_s = 0$, then this equation implies

$$\rho_s^0(1-\phi^0) = \rho_s(1-\phi) \qquad \text{(EQ 13)}$$

where the superscript “0” implies a reference state, like the undeformed material coordinate state. Furthermore, it can be shown that $\frac{\rho_s}{\rho_s^0} = det\underset{\sim}{F}$, leading to the solid balance above. In the examples below it is made clear how this solid mass balance is activated in *GOMA*.

Equations 7and 9 comprise two component mass balances that need to be solved for flow in a fully saturated, deformable porous medium. Because of the relative motion of the solid and liquid, we also need to solve for the stress and displacements of the solid. For this we employ the equation of quasistatic equilibrium:

$$\nabla \cdot (\rho \underset{\sim}{v}_s \underset{\sim}{v}_s) = \nabla \cdot \underset{\sim}{\sigma} \quad *** \qquad \text{(EQ 14)}$$

where $\underset{\sim}{\sigma}$ is the stress tensor of a nonlinear elastic material, augmented with an extra stress due to the effective stress principle of the liquid-phase pressure: $\sigma = -p_{liq}\underset{\sim}{I} + \lambda\underset{\sim}{\varepsilon} + 2\mu\underset{\sim}{E}$.

Here, λ and μ are Lame’ coefficients (or elastic constants), $\underset{\sim}{\varepsilon}$ and $\underset{\sim}{E}$ are the volumetric expansion tensor and Cauchy Strain tensor, respectively (see GOMA 4.0 manual for detailed discussion). Notice the flow pressure in the porous skeleton contributes only an isotropic pressure stress to the overall effective stress.

In partially saturated porous skeleton networks we can employ the equations above in the case of constant gas-phase pressure, so long as we change the local inventory measure of solvent. The change requires the use of a saturation quantity, as one would expect. The local inventory of liquid solvent is best written as

$$C_{ls} = \phi[S\rho_l + (1 - S)\rho_{gv}] \quad . \qquad \text{(EQ 15)}$$

Note that this quantity is independent of phase, and the saturation S, the porosity ϕ, and the gas-phase partial density of solvent vapor ρ_{gv} determine the composite concentration of the liquid-phase component. The porosity in a rigid porous medium is a specified constant; in a deformable medium it is given by Eqn 9 above. The saturation is determined from the capillary pressure curve, the most important and abstract function in partially saturated flow which embodies an empirical correlation of saturation with the liquid and gas phase pressures, viz. the prime dependent variables. The solvent vapor density is determined from these same pressures together with an equation of state that depends on local thermodynamic vapor pressure. These concepts are discussed below.

When the pressure of the base-gas, or gas-solvent, is not constant, we must also keep track of its inventory,

$$C_{gs} = \phi[(1 - S)\rho_{ga}] \qquad \text{(EQ 16)}$$

ρ_{ga} is the partial density of gas-"solvent", viz. air. Here the gas-solvent (usually air) is taken as insoluble in the liquid and gas phases. If the pressure in the gas is constant, so is the gas concentration of air for low-solvent fraction vapor.

Note that the local liquid solvent inventory in general includes solid, liquid, and gas, but the liquid solvent components are insoluble in the solid, and the solid is insoluble in the liquid and gas. Now, following the same derivation above but with this new liquid-solvent inventory measure, we can derive the following mass component balance for each component *i*:

$$\frac{dC_i}{dt} = -C_i \nabla \bullet v_m - \nabla \bullet [\mathbf{v}_s C_i + \mathbf{v}_g \rho_{gi} + \mathbf{v}_l \rho_{li} + \mathbf{J}_{gi} + \mathbf{J}_{li}] \qquad \text{(EQ 17)}$$

Here, C_i is the total concentration of species *i* (e.g. either air or water in both phases, in the simplest case) from both phases (per unit volume of media). ν_g and ν_l are the velocities of gas and liquid relative to the solid skeleton and are normally calculated from Darcy's Law. ρ_{gi} and ρ_{li} are the concentrations of species *i* in the gas and liquid phases, and should be in local equilibrium. J_{gi} and J_{li} are the diffusion fluxes of species *i* in the gas and liquid phases. Currently *GOMA* is set up for *unsaturated* (**`UNSATURATED`** media type) or *two-phase* (**`TWO_PHASE`** media type) transport in porous media. Furthermore, in “unsaturated” flow, the liquid phase is still taken as pure solvent (provisions for multicomponent base liquid phases have now been put in as of 6/13/2001, but have yet to be hooked into the multicomponent species transport equations), and the gas phase is assumed to be at a constant pressure; the dependent variable is p_l, for which the liquid-solvent balance (Eq 17) accounts. In two-phase flow, the liquid phase is still assumed (as of 6/13/2001) pure solvent, and the gas phase contains both air and solvent vapor; the dependent variables are p_l and p_g. In a deformable case a third variable that denotes the solid-phase concentration, viz. the porosity, is added to the variable list, as discussed above. In *GOMA* the whole last term in Eq. (17) is considered the diffusion term, even though it includes convection in the gas and liquid phases.

Darcy’s law describes the flux of the interstitial fluid through the porous medium relative to motion of the solid skeleton. The Darcy velocities, viz. ν_g and ν_l, are the local volume fluxes per unit area of the medium (i.e. the Darcy velocity is an averaged, macroscopic variable while the micro-scale fluid velocity in pores can be considerably higher). In multi-phase unsaturated porous media, the extended Darcy Law weights the mass flux in each phase with a relative permeability to account for the reduced flow due to partial saturation of the porous medium (see Baer, 1972 or Martinez, 1995):

$$\boldsymbol{F}_l = \rho_{lw}\boldsymbol{v}_l = -\frac{\rho_l k k_l}{\mu_l}(\nabla p_l - \rho_l \boldsymbol{g}) \qquad \text{(EQ 18)}$$

$$\boldsymbol{F}_g = \rho_g\boldsymbol{v}_g = -\frac{\rho_g k k_g}{\mu_g}(\nabla p_g - \rho_g \boldsymbol{g}) \qquad \text{(EQ 19)}$$

Here ν_g and ν_l represent the Darcy flux, or Darcy velocity, in the liquid and gas phases respectively, k is the permeability of the porous medium, k_l and k_g are the relative permeabilities for the gas and liquid phases respectively, μ_l and μ_g are the viscosities of the liquid and gas phases respectively, p_l and p_g are the pressures in the liquid and gas phases respectively, and $\boldsymbol{g}$ is the gravitational force vector. ρ_g is the density of the gas phase and is equal to the sum of the partial densities of air and solvent vapor, $\rho_g = \rho_{gv} + \rho_{ga}$.

This information is summarized in Figure 1, followed by a detailed explanation of each equation.You may recognize Equations 18 and 19 in a simpler form, viz. Darcy’s law for the volumetric average flux $\boldsymbol{v}$ in an isotropic fully saturated medium:

$$-\nabla p = \frac{\mu}{k}\boldsymbol{v} \qquad \text{(EQ 20)}$$

Equations 18 and 19 represent a generalization of this simple statement to partially saturated cases and have been used extensively in numerical analysis. In general, the relative permeabilities are a function of the saturation, and the permeability is a function of the porosity (Cairncross et al., 1996 and Martinez, 1995 give some examples of various models for permeabilities).

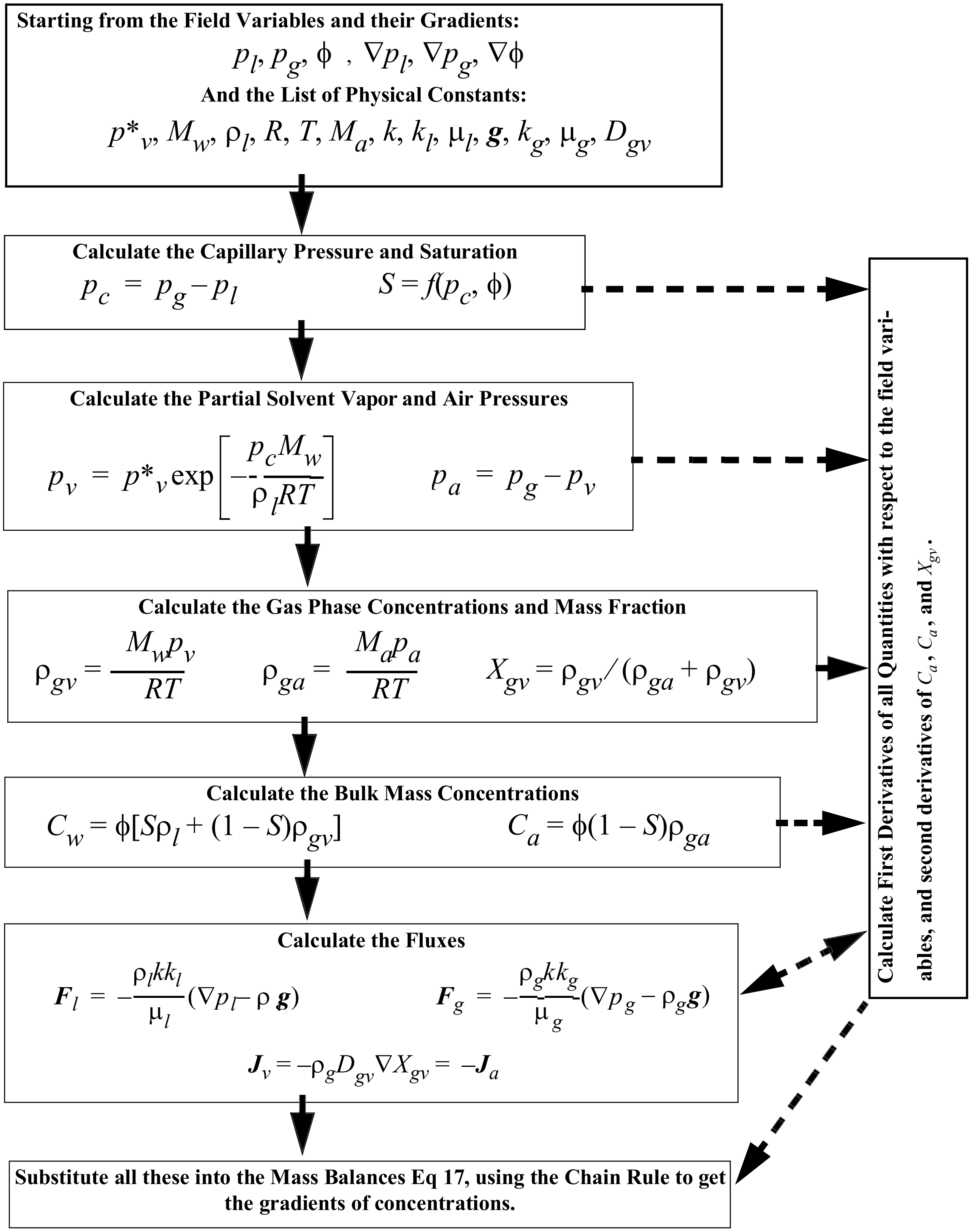


Figure 1. Strategy for calculating physical quantities from field variables in an unsaturated porous medium

In the gas phase, binary diffusion causes diffusive flux of air and solvent vapor relative to the bulk gas flow. This flux is described by Fick's Law:

$$\boldsymbol{J}_v = -\rho_g D_{gv} \nabla X_{gv} = -\boldsymbol{J}_a \qquad \text{(EQ 21)}$$

Here D_{gv} is the effective binary mutual diffusion coefficient for solvent vapor through the pores in the gas phase, and X_{gv} is the mass fraction of solvent vapor in the gas phase ($X_{gv} = \rho_{gv}/(\rho_{ga} + \rho_{gv})$). The effective binary mutual diffusion coefficient is related to the volume fraction of the gas phase, the tortuosity (τ), and the mutual diffusion coefficient in bulk vapor (D_v^0):

$$D_{gv} = D_v^0 \frac{\phi(1-S)}{\tau} \qquad \text{(EQ 22)}$$

τ is the tortuosity of the porous medium, which represents the increased path-length required for solvent to diffuse through the pore-space. Currently, we assume the mutual diffusion coefficient in bulk vapor, D_v^0, is a constant, but this assumption could easily be relaxed in future calculations.

Now we have relations for the fluxes and mass balances in terms of the phase pressures and mass concentrations. To be able to calculate the relationship between the mass concentrations of solvent and air in both phases, we need to invoke an equation of state for the gas, the ideal gas equation, and a statement of local equilibrium between the two phases, the Kelvin equation.

In small pores, the equilibrium vapor pressure above a curved meniscus is lower than the vapor pressure above a flat meniscus (p^*_v). Neglecting the effect of adsorbed layers on pore walls, this is described by the Kelvin equation:

$$p_v = p^*_v \exp\left[-\frac{p_c M_w}{\rho_l R T}\right] \qquad \text{(EQ 23)}$$

$p_c \equiv p_g - p_l$ is the capillary pressure, M_w is the molecular weight of solvent, R is the gas constant, and T is the temperature. This effect of vapor pressure lowering becomes large in pore-sizes in the range of nanometers.

In an ideal gas, the mass concentration is directly proportional to pressure (at constant temperature). Thus the mass concentrations of solvent vapor and air ($p_c \equiv p_g - p_l$) can be calculated:

$$\rho_{gv} = \frac{M_w p_v}{RT} \qquad \text{(EQ 24)}$$

$$\rho_{ga} = \frac{M_a p_a}{RT} \tag{EQ 25}$$

The saturation, or volume-fraction of the pore-space that is occupied by liquid, is an important parameter in the mass balance equations and is crucial to the calculation of the water inventory (vapor and liquid) at any given part of the domain. *It should be noted that the saturation can often be correlated to the capillary pressure, and the functional form must embody many static and dynamic phenomena in order to capture the proper drainage and filling behavior.* Many formulations (e.g. Martinez 1995) choose to track saturation rather than liquid pressure. The current formulation in *GOMA* tracks the liquid phase pressure and determines the saturation as an auxiliary quantity. Whatever the case it is important to understand this relationship in running unsaturated flow problems. It will help you conceptualize what is going on.

As previously mentioned, experimental observations have shown that for a given porous medium the saturation is closely tied to the capillary pressure of fluid in the medium. There are many empirical and theoretical models of this relationship (cf. Cairncross et al. 1996), but for now, we assume that there is a direct functional relationship between saturation, capillary pressure, and porosity:

$$S = f(p_c, \phi) \tag{EQ 26}$$

We have found recently that to model systems involving drainage, imbibition, and deformation this function must be augmented to account for the saturation dependence on rate of filling or drainage. Effectively, this behavior leads to a hysteretic response with respect to rate-of-change of liquid inventory. In all cases, when the liquid wets the solid pore-walls, these relations show that at low capillary pressure, the saturation tends to one (or pore-space saturated with liquid), and at high capillary pressures, the saturation tends to zero (or pore-space saturated with gas). We often describe the saturation function, and the physics underlying it, as follows: *It is a function that relates solvent inventory to dynamics variables, like phase pressures. Its functional form and behavior with respect to capillary pressure can be described as hysteretic for most porous materials which have amorphous structure and some kind of characteristic pore-size distribution. As the medium becomes more unsaturated, viz. S->0, the smallest of pores become partially filled and hence expose the regions of highest capillary pressure: hence the curve shape. However, to drain the smaller pores requires rupture under higher tensions, and hence higher capillary pressures; whereas filling the pores happens with capillary action/imbibition under much lower pressures. Hence the hysteretic behavior.*

IMPORTANT CONCEPT: The sign convention for capillary pressure is critical to the saturation concept. Recall that we define capillary pressure as the difference between gas-phase and liquid-phase pressures, viz. $p_c \equiv p_g - p_l$ *. Clearly, negative capillary pressures imply complete saturation, and in fact in this regime the concept of saturation is meaningless. So* $p_c > 0$ *implies* $0 < S < 1$*. For POROUS_UNSATURED media types, in which the gas-phase pressure is constant, the liquid phase pressure will be less than the specified gas-phase or ambient pressure when the medium is unsaturated, viz. the liquid is in tension. The rate at which the medium drains, or the rate at which it is forced to drain by solid network expansion (in poroelastic problems) is critical to the saturation* *function above. We are currently looking at ways to bring this dynamic dependence to bear. In all cases a family of curves depending on all these conditions must be pursued, as a sort of hysteretic saturation function behavior.*

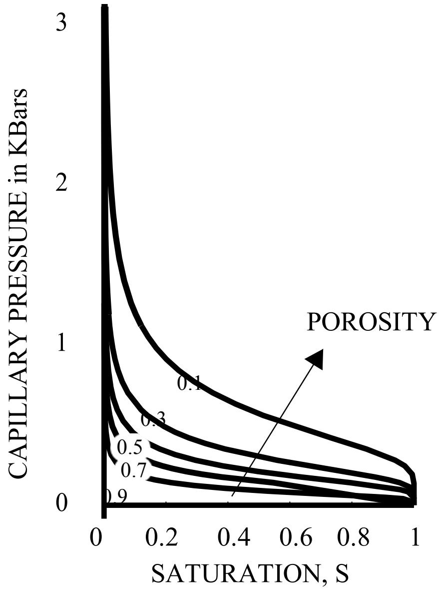
The range of capillary pressure over which the saturation changes from zero to one also depends upon the pore-structure, the surface tension and contact angle of the liquid, and whether liquid is entering or leaving the pore-space; the range of this capillary transition is narrow in media with large pores or narrow pore-size distributions. In deformable porous media, as the medium compresses and the pore-size shrinks, the capillary pressures at which the saturation falls (i.e. pores empty) increases. For purposes of this memo just consider the pattern of saturation versus capillary pressure curve shown in Figure 2 which is generated for a medium characterized by a Gaussian distribution of pore sizes.

We have recently installed a saturation function model that accommodates the often hysteretic behavior of the medium. On the Saturation card in the material input file you will find the **TANH_HYST** model. This model was built on the much simpler TANH model which allows one to fit a single saturation function curve to the hyperbolic tangent function. While similar to the **TANH** model, the **TANH_HYST** model invokes a detailed algorithm to compute scanning curves between the a main wetting curve defined as

$$S = a_w + b_w \tanh\left(c_w - \frac{d_w}{P_c}\right) \qquad \text{(EQ 27)}$$

and a main drying curve defined as

Figure 2. Capillary pressure versus saturation curves for a weighted exponential pore-size distribution as the porosity varies due to deformation in medium without adsorption (pure capillarity)



$$S = a_d + b_d \tanh\left(c_d - \frac{d_d}{P_c} \right) \quad . \qquad \text{(EQ 28)}$$

Here the constants a_w, b_w, a_d, and b_d are computed by scaling the main curve with respect to the capillary pressure and saturation at the point which the conditions switch from wetting to drying or drying to wetting. These constants are defined as follows:

$$b_d = \left[\frac{S_{switch}}{\left(-1 + \tanh\left(c_w - \frac{d_w}{p_c^{switch}} \right) \right)} \right] \qquad \text{(EQ 29)}$$

$$a_d = S_{min} - b_d \qquad \text{(EQ 30)}$$

$$b_w = \frac{S_{switch} - S_{max}}{2} \qquad \text{(EQ 31)}$$

$$a_w = \frac{S_{switch} + S_{max}}{2} \qquad \text{(EQ 32)}$$

The remaining constants are set on the input record. In the above expressions, S_{switch} and p_c^{switch} is the saturation and capillary pressure at the moment that a computational point switches from a wetting to draining regime, or vice versa (viz. they are determined dynamically and not entered as input). S_{min} is the irreducible water saturation from the input record and S_{max} is the maximum water saturation from the input record. The switching criteria is based on the sign of the function (cf. equation 17 above):

$$\frac{dC_{ls}}{dt} + C_{ls} \nabla \bullet \underset{\sim}{v}_m = f \quad . \qquad \text{(EQ 33)}$$

If $f>0$, then the calculation proceeds on the current wetting curve, as the liquid accumulation rate is increasing at that point. Likewise, if $f<0$ the calculation proceeds on the current drying/draining curve. Switching and scaling are invoked if the function f locally switches sign and its absolute value exceeds 1.e-05. ***The implementation and usage of this model are still under development.*** Details of the algorithm can be found in Kool and Parker (1987).

As previously mentioned, Equations 18 through 26 are summarized in the flow chart of Figure 1. How each quantity is computed, the order in which it is computed, and the general flow of the partially saturated porous media calculation is hopefully clarified at this point. One of the many important concepts to understand is “for what quantity is *GOMA* solving”, as this bears on the initial and boundary conditions of the problem. For partially-saturated porous media problems the initial conditions are crucial to the numerics. Poor initial conditions often lead to numerical failure (cf. trouble shooting tips in the Appendix). The example problems below will cover this aspect of the problem.

The physical quantities for which *GOMA* actually solves are the liquid phase pressure p_l for **`UNSATURATED`** or **`POROUS_PART_SAT`** media types, porosity ϕ for all porous media types with variable porosity models, and the liquid-phase and gas-phase pressures p_l, p_g for **`TWO-PHASE`** media types. Here we focus on the simplest case in this tutorial: **`UNSATURATED`** or **`POROUS_PART_SAT`** media types. In this case, we assume that the gas-phase pressure is constant, and solvent transport through the medium can only occur by capillary migration of the saturation front and by vapor diffusion. Interestingly, in this case we are solving for the liquid-phase pressure p_l at all time steps and all Newton iteration steps. This is somewhat abstract for a partially-saturated problem as all initial and boundary conditions must involve this pressure, and this pressure is only associated with the liquid phase. The saturation functions (Equation 26 above) dictate that in low-saturation regions the liquid-phase pressure is high due to the large number of capillary menisci present. As saturation tends to unity, that pressure drops precipitously as those menisci are much less prevalent. With p_l and all physical properties of the system, the flow chart in Figure 1 clearly shows how all other relevant quantities, like solvent concentration in the liquid and the gas phases, can be calculated. The initial conditions for the liquid-phase pressure for several cases will be covered in the examples below. Here we will cover the boundary conditions.

In order to apply the Darcy-potential form to a porous penetration problem, several important boundary conditions are needed. Here, for the sake of completeness, we present the most commonly used boundary conditions currently supported in *GOMA* that pertain to the Darcy-potential formulation (see GOMA 4.0 manual for remainder of supported boundary conditions):

•Boundary Conditions for Darcy Potential Formulation (Equation 18)

GOMA BC Name and Description	Mathematical Form and Differential Equation to which it is applied	Formulation
DARCY_CONTINUOUS Applied to a side set. Continuity of velocity at an interface between a continuous fluid and a porous medium.	$\rho n \cdot (\underset{\sim}{v} - \underset{\sim}{v}_s) = \boldsymbol{F}_l = \rho_{lw} v_l = -\frac{\rho_l k k_l}{\mu_l} \nabla p_l$ Applied to normal component of continuous fluid momentum equation at the interface. Note that this assumes fully saturated medium at the interface. (cf. GT-008.2 memo for saturated cases.)	Darcy potential Form. Saturated
POROUS_PRESSURE Applied to a node set. Set the continuity of hydrodynamic pressure in the continuous fluid to the *liquid* Darcy pressure in the porous medium, at the interface.	$p_l = p_c$ Applied to the porous liquid solvent balance equation from the porous medium at the interface.	Darcy potential Form Saturated

•Boundary Conditions for Darcy Potential Formulation (Equation 18)

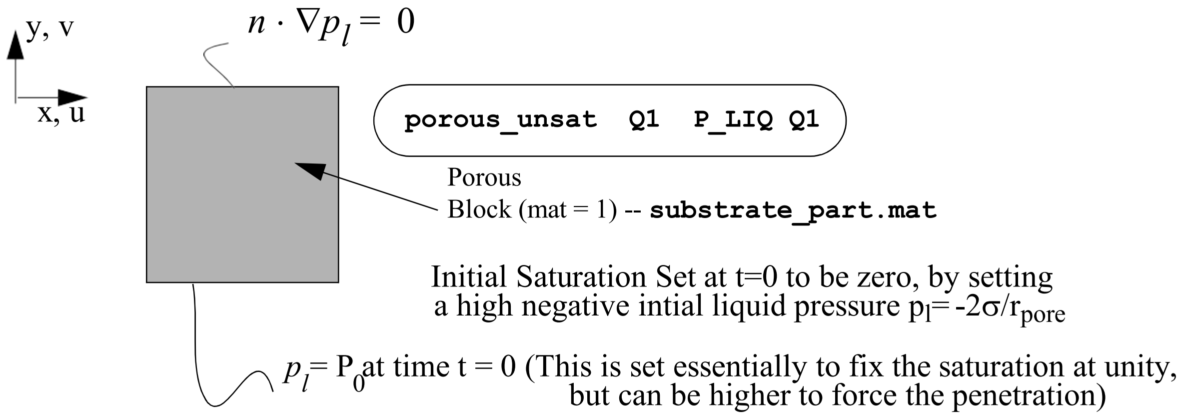
GOMA BC Name and Description	Mathematical Form and Differential Equation to which it is applied	Formula-tion
VELO_TANGENT_SOLID Applied to a side set. Set tangential fluid velocity to the tangent velocity of an advecting/ deforming porous medium	$t \bullet v = t \bullet v_s$ Equivalent to tangential component of the NO_SLIP condition (see Schunk et al. 1998) but allows for penetration into translating bodies. Applied to tangential component of fluid momentum equation from the continuous phase.	Any form. Saturated or unsaturated.
POROUS_FLUX Applied to a side-set Set flux of solvent at surface of porous media equal to mass transfer coefficient times driving force	$n \cdot (v_g \rho_{g_i} + v_l \rho_{l_i} + J_{g_i} + J_{l_i}) = h\phi(\rho_{g_i} - \rho^0_{g_i}) + Hh_2(p_l - p^\infty_l)$	Darcy-Potential form. Unsaturated

Together with these conditions, we also have the option of setting the dependent variables to a desired value at a boundary. These are known as Dirichlet boundary conditions. Available are *POROUS_LIQ_PRES*, for setting p_l on a boundary, *POROUS_GAS_PRES*, for setting p_g on a boundary. There are also several others described in the GOMA 4.0 manual.

This completes a description of the current *GOMA* formulation for modeling of partially-saturated flow in isotropic, rigid porous media. Please consult the appendix of this document for a listing of the current limitations of the Darcy-Potential approach in *GOMA*, together with a troubleshooting guide for modeling problems of this type. Also consult the GOMA 4.0 manual for a better description and complete list of the material properties and parameters required to run these problems. With these equations and boundary conditions, we now address the tutorial problems. All are one-dimensional and all pertain to imbibition, drainage and drying of coatings on porous, deformable substrates. The concepts however can be easily extended to multidimensional cases.

Capillary Imbibition - single phase

This problem should help the user to understand some of the issues of following a saturation front. The directory is called **`1D_hysteresis`**. It is a one-dimensional problem and uses lumped parameter models for the bounding continuous phases, viz. it is single phase:



We will first follow just capillary imbibition. Other tutorial problems in this memo will be extended to allow for drying from above and below, or pressurized input. Lets consider the important portions of the *GOMA* input deck **`input_imbibe`** for this simpler case:

```
{include("Defs")}
$
 FEM File Specifications
 ---
...
 ---
General Specifications
 ---
...
Initial Guess                     = zero
###initialize liquid pressure to really large => low init. satura###
 ---
Time Integration Specifications
 ---
Time integration                  = transient
delta_t                           = {tmin=1.e-6}
Maximum number of time steps      = 1000
Maximum time                      = 4.0
Minimum time step                 = {tmin/10.}
$$%Maximum time step                 = 0.001
Maximum time step                 = 10.0e-3
Time step parameter               = 0.
Time step error                   = -0.01  0  1  0  1  1  0
Printing Frequency                = 1

...
```

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

$$Substrate-film interface
####Set to insure 100% sat. at boundary
BC      = POROUS_LIQ_PRESSURE NS 6 {-pcmin}

END OF BC
----
Problem Description
---

Number of Materials = 1

 MAT =   substrate_part

  Coordinate System = CYLINDRICAL
  Element Mapping   = isoparametric
  Mesh Motion = LAGRANGIAN
   Number of bulk species = 0

 Number of EQ  =  {auto_count}
 EQ = porous_unsat Q1 P_LIQ    Q1      1.  1.    1.   1.   1.
 END OF EQ
                                   div  ms  adv   bnd  dif  src   porous
##########
END OF MAT
##########
```

The **`aprepro`** variables can be found in the files **`Defs`** in the directory entitled **`1D_hysteresis`**. This example makes extensive use of **`aprepro`**.

This excerpt is from the file **`input_imbibe`** in the **`1D_hysteresis`** directory. First of all notice that the mesh motion type is **`LAGRANGIAN`**, even though we are considering only the rigid case and not solving for the mesh motion. We are also only solving for the p_l variable, which nominally is the liquid pressure. The corresponding material file is **`substrate_part.mat`** (see below). Notice we use **`Q1`** elements in the mesh are graded and packed in for good refinement; however, we have found that after startup, 20 or 30 elements across suffices for the settings in this problem. These problems require finer meshes the more narrow the pore-size distribution, or the steeper the saturation curve, which are cases in which the saturation front gets steep. The problem is transient, of course, and in this case we use a variable time step starting at 1.e-6. Actually, it pays to experiment with the time stepping scheme. We find variable and fixed time stepping works, but small time steps are necessary sometimes so the front movement is captured. Also notice that the time-step error parameter is set to -0.01. We find that in a coupled of problems with continuous phase and a porous phase that this has to be set even higher. These problems are poorly conditioned and if the error-criterion is too strict the time step will grind to a halt.

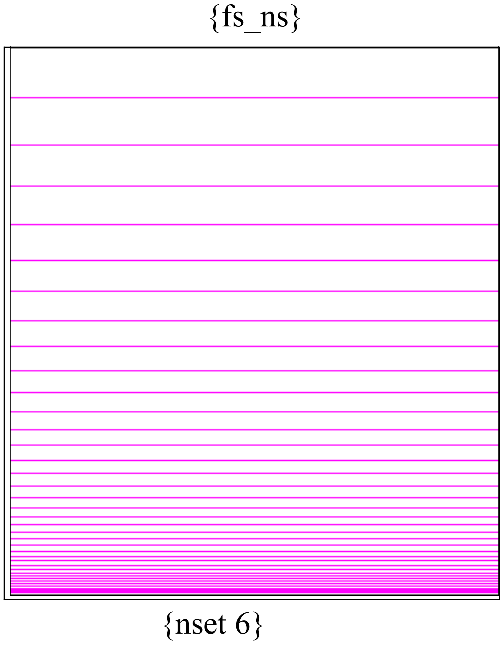


Figure 3. Mesh for `porous_dry` tutorial problem.

The first case we will consider is the one in which the bottom side of the porous medium (the node set number denoted by `{nset 6}` above) is placed in a bath of liquid. Presumably, the bottom side of the medium will be instantly saturated at the start of the analysis. The capillary pressure-saturation model (Equation 26 and Figure 2) dictates that at high saturation we must have low capillary pressure and hence low liquid pressure (cf. Figure 1, note that $p_c = p_l$ if the gas pressure is taken as constant and atmospheric, i.e., 0). Because we are solving for the liquid phase pressure this means that setting the pressure at the top to atmospheric, or low, implies we are setting the saturation there to unity. This is both an initial condition and a boundary condition. Also, initially, the medium is dry. To initialize the medium as dry requires a high capillary pressure, or high liquid phase pressure, hence the initialize card in the `substrate_part.mat` deck which sets the initial liquid pressure to `{-10000}` (-`pcmax`, as set in the `Defs` file).

This pressure seems arbitrary but there is a simple way to see if your initialization is right. First recall that there are many physical property models and parameters that are set in the `substrate_part.mat` file, and we have not discussed those yet. The saturation vs. capillary pressure curve parameters are defined there, as is the porosity which also affects the curve. Please consult Cairncross et al. (1996) for a description on how those curves are generated. Now, if your run a few iterations with the "`Write Intermediate Solution`" card turned to "`yes`" and the update parameter set to zero, viz.

```
goma -a -i input.imbibe -r 0.0
```

then you can `blot` the `out.exoII` file and plot the saturation curve across the porous material. You can see then if your initial pressures were appropriate. Use "`splot`" in `blot` to look at the x-y profiles of your pressure and saturation vs. time. That is, within blot simply do:

```
blot out.exoII
>splot
>nodes path 1 31
>sat
>p
```

Here the node numbers 1 and 31 were picked off a mesh plot using the "`number`" toggle in blot. ***You might check your version of the mesh as these node numbers may not apply.*** The nodes are the extremes (bottom and top) of the centroid nodes in the middle of the film. You have to zoom in to read them in most cases.

Perhaps the most important input in this problem comes through the material file, `substrate_part.mat`. The relevant portions of that are shown here:

```
{include("Defs")}
...
---Microstructure Properties (Porous section 1)
Media Type               = POROUS_UNSATURATED
$$%Porosity                 = DEFORM {poros}
Porosity                 = CONSTANT    {poros}
Permeability             = CONSTANT     {permeab_0comp}
/*Interesting here, but compressiblity is defined as
 *d(ln(rho)/d(P). My overall pressure is much smaller...
 */
$$%Liquid phase compressibility = CONSTANT {1.0e-5*beta_liq}
$$%Liquid phase reference pressure = CONSTANT {-pcmin}
/*For compressible must have power law lame's */
Capillary Network Stress = PARTIALLY_WETTING
Rel Gas Permeability       = CONSTANT  {1./airvisc}
Rel Liq Permeability       = VAN_GENUCHTEN  0. 0. 0.7 {lqvisc}

Saturation               = TANH 0.1163 0.0 4.5 30000.0
$Saturation               = TANH_HYST 0.1163 0.0 4.5 30000.0 0.1163 0.0 4.5
92000.0

--Misc Porous numerical props (Porous Section 2)
Porous Weight Function          = SUPG 1.0
Porous Mass Lumping             = TRUE
Porous Time Integration         = TAYLOR_GALERKIN 0
Porous Advective Scaling         = CONSTANT 0 1.

---Porous Species Properties (Porous Section 3)
/* Darcy fickian only for saturated media */
Porous Diffusion Constitutive Equation = DARCY_FICKIAN
Porous Gas Diffusivity              = CONSTANT  0  {gpdiff}

Porous Latent Heat Vaporization = CONSTANT  0   0.
Porous Latent Heat Fusion       = CONSTANT  0   0.
```

```
Porous Vapor Pressure           = FLAT      0  {vappres} {sldens} {mwsl}
{rgas}
{temp} {mwair} {ambpres}
Porous Liquid Volume Expansion  = CONSTANT  0   1.
                                            ^
***Species Number (all zero for primary liquid solvent)****|

--Porous Gas-inert properties
/*Note. if you change p_ambient here you need to make sure the initialize
card and the boundary conditions are compatible with the saturation curve.
*/

Porous Gas Constants           = IDEAL_GAS   {mwair} {rgas} {temp}
{p_ambient=0}

----Source Terms--
Navier-Stokes Source            = CONSTANT      {-980.0*sldens}  0.  0.
Solid Body Source               = CONSTANT      0.  0.  0.
Mass Source                     = CONSTANT      0.
Heat Source                     = CONSTANT      0.
Species Source                  = CONSTANT  0   0.
Species Source                  = CONSTANT  1   0.
Species Source                  = CONSTANT  2   0.
Current Source                  = CONSTANT      0.
Initialize = POR_LIQ_PRES 0 {-pcmax}

sass2521-atm:/home/prschun/fem/gomadir/m_matl/porous_tests/1D_hysteresis
245 % more Defs
$  BASE UNITS are CGS

$ Conversion for Length change new scale = cm * conv
$ So to convert from cm to mm -> conv = 10.0
$ Length conversion             {L =1.0}
$ Mass conversion               {K =1.0}
$ Time conversion               {S =1.0}
$ moles conversion              {M =K}

$Substrate thickness            {h_s = 0.5*L}  h_s = 0.5*L
$Substrate radius               {r_s = 0.1*L} r_s = 0.2*L

$Number elements across substrate       { ne_s = 1}
$Number elements across substrate width { ne_ws = 40} ne_ws = 50

$ Permeability of Substrate  0%         {permeab_0comp = 2.0e-7*L^2}
$ Permeability of Substrate 25%         {permeab = 1.43e-7*L^2}
$ Measured pore size                    {poresize = 0.0112*L}
$ Surface Tension of Water              {surftens = 72.0*K/S^2}
$ Liquid Viscosity                      {lqvisc = 0.01*K/L/S}
/*Note. if you change p_ambient here you need to make sure the initialize
card and the boundary conditions are compatible with the saturation curve.
*/

Porous Gas Constants           = IDEAL_GAS   {mwair} {rgas} {temp}
{p_ambient=0}
```

```
----Source Terms--
Navier-Stokes Source           = CONSTANT      {-980.0*sldens}  0.  0.
Solid Body Source              = CONSTANT      0.  0.  0.
Mass Source                    = CONSTANT      0.
Heat Source                    = CONSTANT      0.
Species Source                 = CONSTANT  0   0.
Species Source                 = CONSTANT  1   0.
Species Source                 = CONSTANT  2   0.
Current Source                 = CONSTANT      0.
Initialize = POR_LIQ_PRES 0 {-pcmax}
```

First and of foremost importance is the specification of the media type; in this case we have it set to **`POROUS_UNSATURATED`** or **`POROUS_PART_SAT`**. Another choice available for the media type card is **`POROUS_TWO_PHASE`**, which is used when pressure gradients in the gas phase are of concern. That choice requires an additional transport equation to account for the phase pressure. This memo addresses the case in which transport through the gas phase is strictly in the form of Fickian diffusion, with the gas pressure held constant. The properties of the medium like the porosity and permeability are all held constant in this case, although extensive capability is available to handle nonuniform pore-sizes, with pore-size distribution models like **`PSD_VOL`**, **`PSD_SEXP`** and **`PSD_WEXP`**. These models use probability size distributions of exponential or Gaussian form to calculate the property or parameter of concern. **`Permeability`** and **`Saturation`** are two inputs which allow such models. In this example, we keep most properties constant and uniform. Obviously this does not apply to the **`Saturation`** card, or the **`Rel Liq Permeability`** card. (The units of this problem are specified in the accompanying **`Defs file`**.) Often times when debugging your input deck you should vary these parameters to see if the calculations proceed more smoothly. It is beneficial to consult the supplied references for more detailed descriptions of the microstructure properties, as such discussions here are beyond the scope of this tutorial but help in troubleshooting your problem.

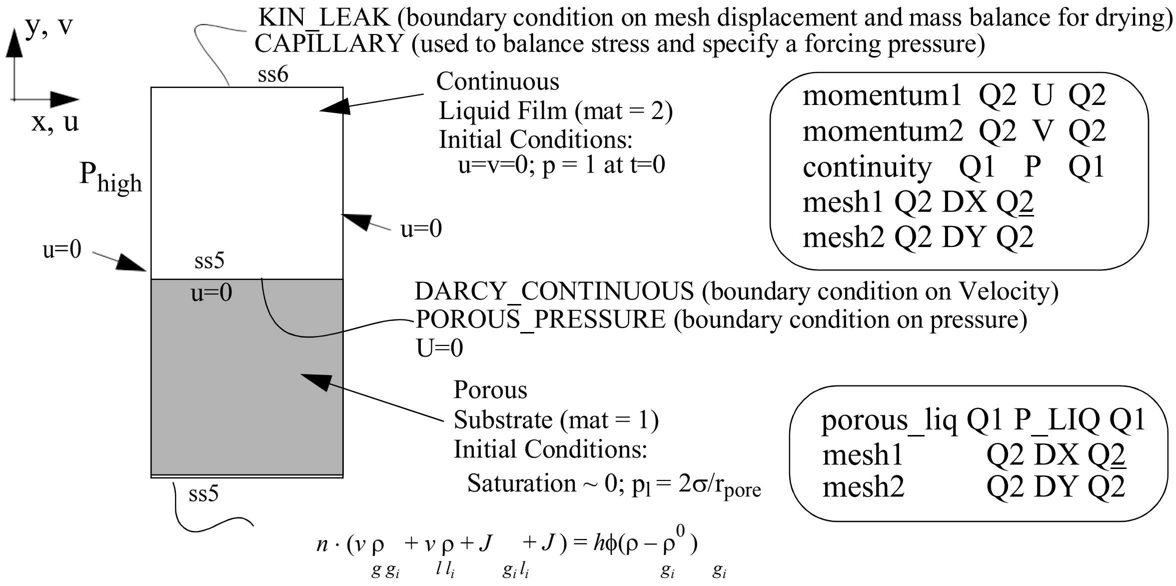
The next section of the material file deals with the specification of the numerical treatment of the porous flow equations. **It is always recommended that you use Porous Mass Lumping (viz. set to true or yes), with Q1 elements in partially saturated flow problems.** The **`SUPG`** porous weight function doesn't seem to help too much. The **`Porous Time Integration`** type and **`Advective Scaling`** are no longer use.

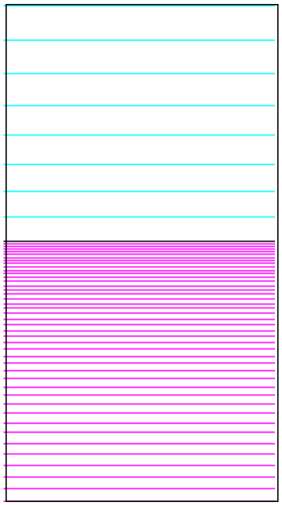
The transport constitutive equations for solvent and gas in the porous medium are specified next. For the **`POROUS_PART_SAT`** media type, only properties associated with the liquid in its vapor form are needed. Notice that in this example we are using a DARCY_FICKIAN diffusion model, meaning we account for no solvent vapor diffusion in the pore space, i.e., only capillary imbibition is present. In this isothermal case, the vapor-pressure card is the last remaining important card. Finally, notice that the initial liquid phase pressure is specified at the bottom of the mat file. This pressure is set to a large, negative value so as to lead to an initially small saturation.

An interesting exercise to help verify expected results is to increase the inlet pressure from -pcmin to +pcmin (i.e., making the inlet pressure positive, the capillary pressure negative and hence the medium fully saturated). This should increase the penetration rate. We find that indeed it does, from about 1.1s to 0.7s reduction in penetration time.

Capillary Imbibition, two phases (film on substrate) with drying and forced impregnation

This problem illustrates in a simple one-dimensional geometry one of the most complicated cases addressed in this memo. The geometry is similar to the first problem addressed (see page 17); however, in this case we a liquid film on top that is modeled as a continuous phase. The idea here is that we can pressurize the continuous phase for fast penetration into the porous substrate, dry the continuous phase, dry from the porous substrate, etc., much as in a continuous liquid film drying process on a porous substrate. This directory name for this problem is **`porous_film_dry`**. For convenience we show again here the geometry, appropriate boundary conditions, and the mesh:





The input files and material files for this example can be found in a directory called **`porous_film_dry`**. Noteworthy portions of the GOMA input deck are shown first.

```
{include(Defs)}

FEM File Specifications
 ---
FEM file                           = ml.exoII
Output EXODUS II file              = out.exoII
GUESS file                         = contin.dat
SOLN file                          = soln.dat
Write intermediate results         = no
 ---
General Specifications
 ---
Number of processors               = 1
Output Level                       = 0
Debug                              = 0
Initial Guess                      = zero
$$Initial Guess                      = read_exoII_file restart.exoII
Initialize = TEMPERATURE   0 {T_init}
Anneal Mesh on Output              = no
 ---
Time Integration Specifications
 ---
Time integration                   = transient
delta_t                            = 1.0e-6
Maximum number of time steps       = 2000
Maximum time                       = 100.0e+0
Minimum time step                  = 1.e-10
Maximum time step                  = 1.e+1
Time step parameter                = 0.
Time step error                    = -1.e-07  0  0  1  1  0  0
Printing Frequency                 = 1
 ---
Solver Specifications
 ---
Solution Algorithm                 = umf
Number of Newton Iterations        = 20
Newton correction factor           = 1.0
Normalized Residual Tolerance      = 1e-8
Residual Ratio Tolerance           = 1e-2
Pressure Stabilization             = no
Pressure Stabilization Parameter = 1.0
```

Proper initialization of this problem is crucial to successful time integration. The top part of the input deck (shown above as an excerpt from the file **`partsat_input_dry`**) shows no initialization of the variables globally (i.e., in both materials) with the exception of temperature. Even though the energy equation is carried along in this calculation, all temperature-dependencies in the properties have been turned off so the equation is decoupled from the rest of the problem. As we will see below, the initialization of the porous saturation or alternatively liquid-phase pressure occurs in the material file **`substrate_part_dry.mat`**. The time stepping parameters show a small (1.e-6) but variable initial time step. Notice stringent time step error tolerance, of -1.e-07. Also notice we are using a 1st-order scheme as the **`Time Step Parameter`** set to 0.0 (cf. Schunk et al. 1998). With mass lumping treatment it is advisable to always use a first-order backward Euler time integration scheme (see trouble shooting tips below).

The boundary condition section for this problem is as follows:

```
 ---
Boundary Condition Specifications
 ---
Number of BC                = -1
$BC = PLANE SS 1 0.00 1. 0. -0.0
BC = DX NS 1 0.
BC = DY NS 1 0.
BC = PLANE SS 2 1.00 0.0 0. {-0.05*h_s}
BC = PLANE SS 4 1.0 0. 0. -0.0
BC = U NS 2 0.
BC = U NS 4 0.


BC = KIN_LEAK SS 6 0. 0. /*Top drying surface of continuous film */
BC = CAPILLARY SS 6 0. {pcmin} 0. 0.

$$Substrate-film interface
BC     = POROUS_PRESSURE SS 5 1 2
BC     = U NS 5 0.
BC     = DARCY_CONTINUOUS SS 5 1 2

$$ Bottom of porous substrate
BC = POROUS_FLUX SS 1 0 0.3 0.0

$For saturated case...
$BC = POROUS_LIQ_PRESSURE NS 1 {pcmax}
END OF BC
```

These boundary conditions are similar to the saturated version of this problem discussed in GT-008.2. We see we can dry from the top free surface as specified by the **`KIN_LEAK`** card, even though now the constants are set to zero for this demonstration. One can easily add some drying here with the plethora of capability we have installed over the past five years. Again as in the earlier problems we are assuming a pure solvent. Advancing this model to include multicomponent solvents will be undertaken at a later date. We use the **`CAPILLARY`** card to specify an external gas pressure. This card will enforce the pressure in the liquid phase to be constant at **`{pcmin}`**, as defined in the **`Defs`** file, right down to the film/substrate surface. The interfacial conditions balance the hydrodynamic and the pore liquid pressure with **`POROUS_PRESSURE`**, and balance the mass flux across the interface with **`DARCY_CONTINUOUS`**; the latter condition assumes that the pore space is saturated at the interface. This places some severe limitations on the pressure in the liquid film which is relevant to the discussion here. Because saturation is calculated based on the level of pressure, appropriate **`Saturation`** model parameters must be chosen so that high liquid pressures still leave the pore pressure in a region of total saturation, according to Figure 2. Scaling the pressure properly should be adequate for near term applications, but this does raise some questions as to how we will handle forced penetration with an externally applied pressure, for partially saturated problems. At the bottom of the domain, or the bottom of the substrate, we are using the **`POROUS_FLUX`** condition to control the solvent vapor and liquid loss from the underside. The high mass transfer coefficient specified here will be varied below to illustrate the effect.

Finally, in the input deck we specify the following problem description:

```
----
Problem Description
---

Number of Materials = 2

 MAT =   substrate_part    1

  Coordinate System = CARTESIAN
  Element Mapping   = isoparametric
  Mesh Motion = LAGRANGIAN
  Number of bulk species = 1

    Number of EQ = 4
    EQ = mesh1          Q2 D1 Q2      0.  0.  1.  1.  0.
    EQ = mesh2          Q2 D2 Q2      0.  0.  1.  1.  0.
    EQ = porous_liq    Q1 P_LIQ Q1   1.   1.   1.   1.   1.
    EQ = energy         Q2 T  Q2      1.  1.  1.  1.  1.

 MAT =   layer1    2

  Coordinate System = CARTESIAN
  Element Mapping   = isoparametric
  Mesh Motion = ARBITRARY
  Number of bulk species = 1

   Number of EQ = 6
    EQ = mesh1          Q2 D1 Q2      0.  0.  1.  1.  0.
    EQ = mesh2          Q2 D2 Q2      0.  0.  1.  1.  0.
    EQ = momentum1      Q2 U1 Q2      0.  0.  1.  1.  1.  0.
    EQ = momentum2      Q2 U2 Q2      0.  0.  1.  1.  1.  0.
    EQ = continuity     Q1 P  Q1  1.              0.
    EQ = energy         Q2 T  Q2      1.  1.  1.  1.  1.
```

In the liquid material (mat number 2, with material parameters in `layer1.mat`) we have six differential equations, including the Navier-Stokes equations, the energy equation and the mesh motion equations. The role of these equations should be clear from earlier tutorials and discussion. In the substrate material (mat number 1, with material parameters in `substrate_part.mat`) we have four equations, including again the mesh motion equations. The `porous_liq`equation as usual will play the role of Eq. 17, i.e., it is responsible for tracking the liquid phase pressure and the gas phase vapor concentration, through the relationships diagrammed in Figure 1. Note that the material mesh motion type is `LAGRANGIAN`, but as we will see the elastic constants are set high enough that no significant deformation of the pore structure will occur. Note that we again choose interpolations and weightings that are Q2 for every equation except for the porous-liquid and continuity equations. Only the continuous phase pressure in material 2 is interpolated with Q1 functions, and the weighting function is Q1 for the continuity equation in order to make the equations mathematically well-posed.

Excerpts from `substrate_part_dry.mat` that are relevant here are:

```
---Microstructure Properties (Porous section 1)
Media Type              = POROUS_UNSATURATED
```

```
Porosity                 = CONSTANT {poros}
Permeability             = CONSTANT      6.8e-06
/*For compressible must have power law lame's */
Capillary Network Stress = PARTIALLY_WETTING
Rel Gas Permeability      = CONSTANT  {1./airvisc}
Rel Liq Permeability      = VAN_GENUCHTEN  0. 0. 5.0 {lqvisc}
$Saturation              = TABLE 2 CAP_PRES LINEAR
FILE=pg_test_table_real.prn
Saturation                = VAN_GENUCHTEN  0.05 0. 8.403 82000

--Misc Porous numerical props (Porous Section 2)
Porous Weight Function          = SUPG 1.0
Porous Mass Lumping             = TRUE

---Porous Species Properties (Porous Section 3)
/* Darcy fickian only for saturated media */
Porous Diffusion Constitutive Equation = DARCY_FICKIAN
Porous Gas Diffusivity               = CONSTANT  0  0.0026

Porous Latent Heat Vaporization = CONSTANT  0   0.
Porous Latent Heat Fusion       = CONSTANT  0   0.
Porous Vapor Pressure           = FLAT      0  {vappres} {sldens} {mwsl}
{rgas}
{temp} {mwair} {ambpres}
Porous Liquid Volume Expansion  = CONSTANT  0   1.
                                            ^
***Species Number (all zero for primary liquid solvent)****|

--Porous Gas-inert properties
/* Note, to change p_amb you might have to change the initial conditions
 * and boundary conditions so as to get the same solution, viz. one that
 * has a S=1 on the left
 */
Porous Gas Constants           = IDEAL_GAS   {mwair} {rgas} {temp} {0.}

----Source Terms--
Navier-Stokes Source            = CONSTANT      0.  0.  0.
Solid Body Source               = CONSTANT      0.  0.  0.
Mass Source                     = CONSTANT      0.
Heat Source                     = CONSTANT      0.
Species Source                  = CONSTANT  0   0.
Species Source                  = CONSTANT  1   0.
Current Source                  = CONSTANT      0.
$Initialize = POR_LIQ_PRES 0 {pliq_for_low_saturation}
Initialize = POR_LIQ_PRES 0 {pcmax}
```

First of all note that we are using constant overall permeabilities and a variable liquid phase relative permeabilities. We also use the `VAN_GENUCHTEN` model for the relative liquid permeability.

The `Saturation` card deserves some discussion. The model specified for the saturation vs. capillary-pressure function is `VAN_GENUCHTEN`, which is an empirical fit to experimental data. The parameters on the `Saturation` card are in effect used to set the properties of the pore space.

We are using a diffusivity model called **`DARCY_FICKIAN`**, which indicates that in the gas-phase, transport can occur by diffusion of solvent according to Equation 21. Accordingly, we have to set a gas-phase diffusion coefficient with the **`Diffusivity`** card, and the vapor pressure model becomes important. Shown in this material file is the **`FLAT`** model, viz.

$$\rho_{gv} = \frac{M_w p_v}{RT} S \qquad \text{(EQ 34)}$$

where S is the local saturation. This model is ad-hoc but nonetheless leads to some interesting results. It basically says that as saturation increases, the gas-liquid menisci, and correspondingly the interfacial area available for evaporation, become more concentrated and hence the gas-phase vapor concentration increases.

We now present some results of running **`partsat_input_dry`**.

Then start GOMA with the full simulation, viz.

```
goma -a -i partsat_input_dry.
```

In Figure 4 we show the film surface position versus time for three cases. The first case includes only substrate imbibition, the second case includes both imbibition and slow drying from beneath (mass transfer coefficient of 0.03), and the third case imbibition and fast drying (mass transfer coefficient on POROUS_FLUX card of 30). As expected the surface recedes much faster in the latter case. Interestingly in both cases the recession is initially very rapid due to incipient imbibition. At early stages the flow in the porous substrate is encountering little or no resistance due to a short impregnation length. Recall that in this example we are not allowing any drying from the continuous film layer, or the top of the domain, although that would be easy with the KIN_LEAK card.

In Figure 5 we show the effect of low and high vapor pressure on the penetration front dynamics. You will notice that the low-vapor pressure case leads to faster saturation front penetration and actually allows imbibition to eventually saturate the substrate. In the high vapor-pressure case a steady penetration rate is reached at which the imbibition rate equals the drying rate, viz. the imbibition rate decreases with increasing penetration (an expected result due to the greater the pressure drop required to feed the front) until the drying rate (really the diffusion rate plus the mass transfer off the bottom) equals the penetration rate.

Finally, we use this example to benchmark mass conservation in the coupled continuous/porous impregnation problem. In Figure 6 we show a sequence of snap shots during the imbibition case, without drying from the bottom or top. To verify proper mass conservation, the inventory of the liquid in the bottom substrate should equal the loss of liquid from the continuous film. We have performed this balance and found that the liquid inventory in the porous substrate was 98% of that lost from the film. We measured the inventory in the porous substrate with an numerical volume integral of the quantity $\rho S \phi$; the crude integral in a spread sheet could be one source of error.

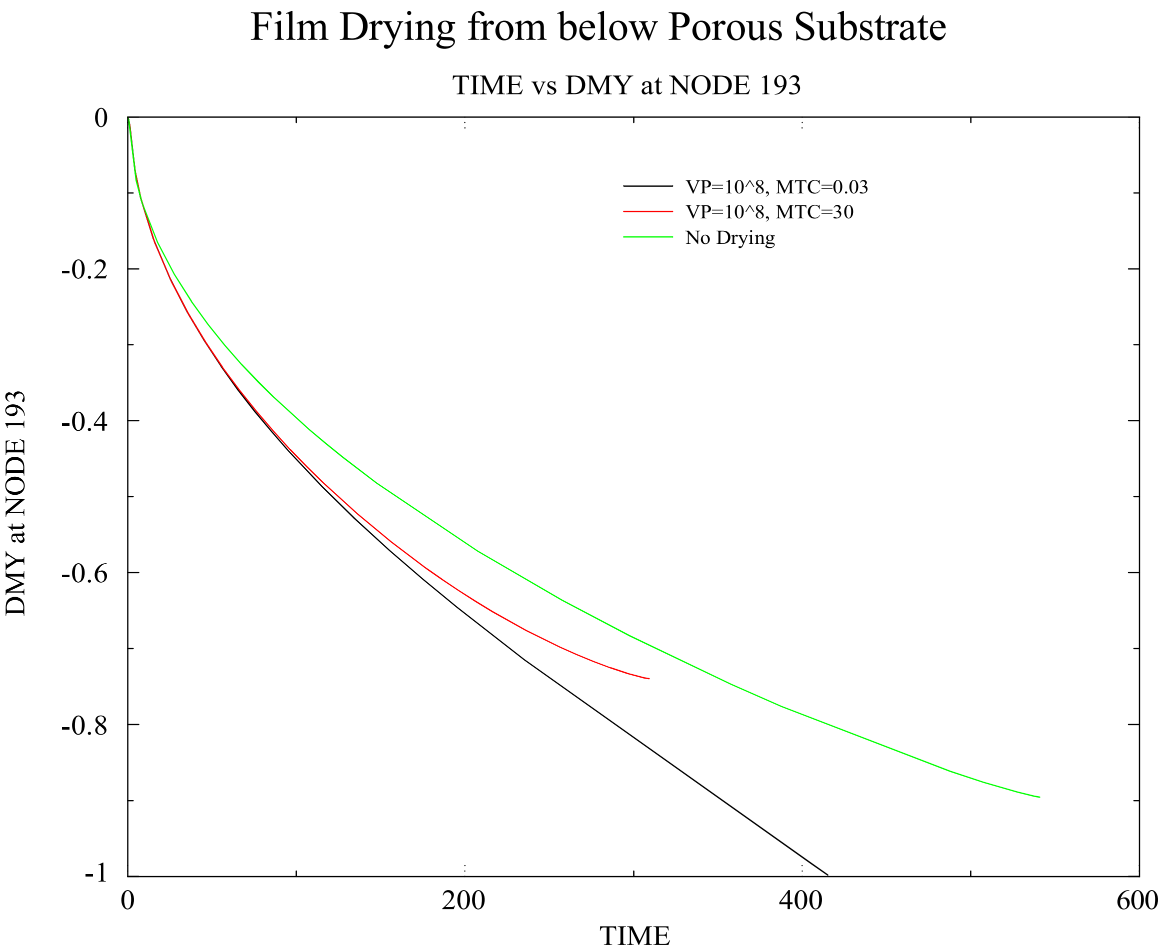


Figure 4. Free surface position versus time for the case with no drying with imbibition (green) and drying and imbibition (black and red). Vapor pressure model is the FLAT type and the flat-interface vapor pressure is 1.e8; the mass transfer coefficient on the underside is 0.03 in one case and 30. in the other.

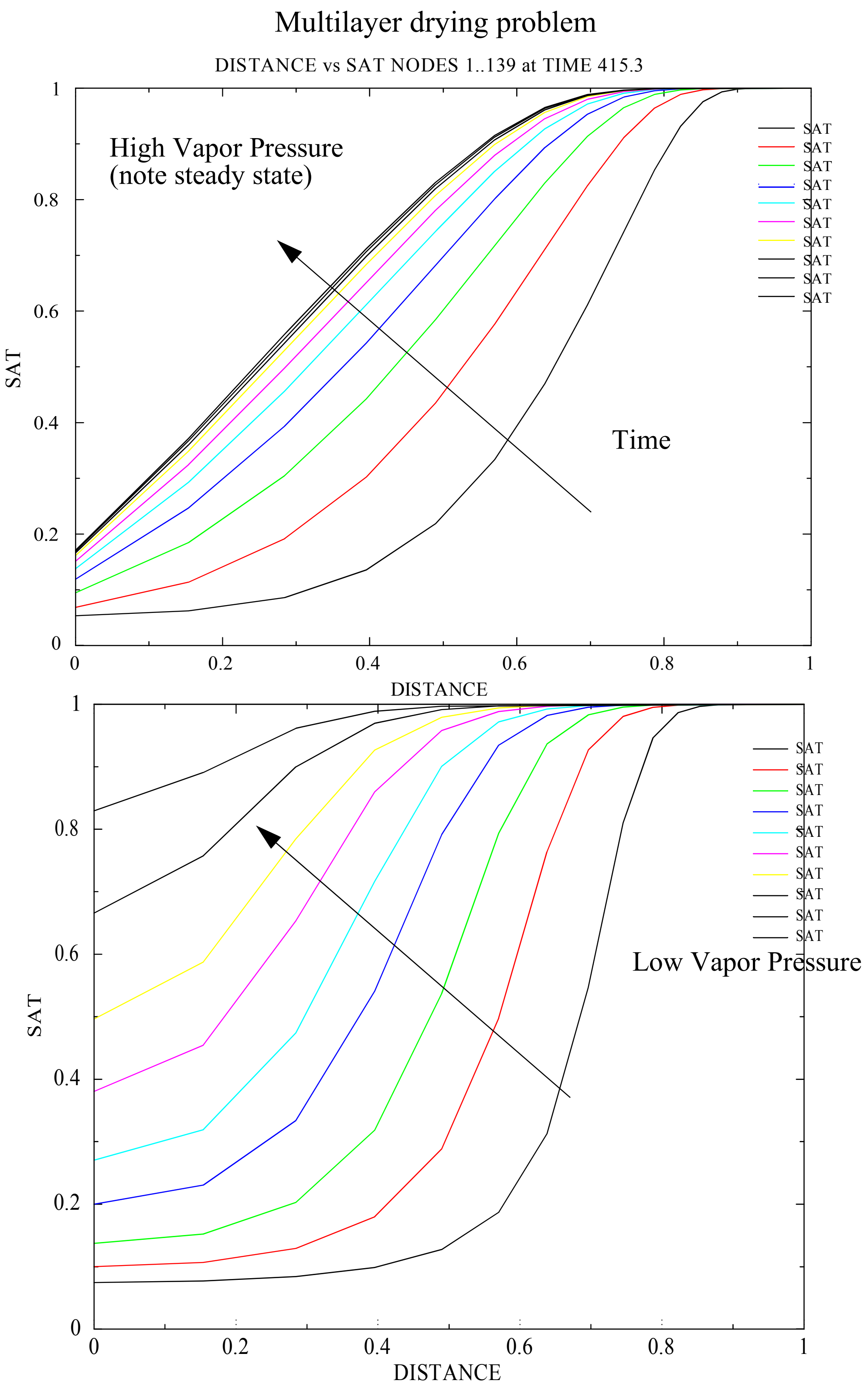


Figure 5. Saturation in porous substrate measured from interface versus distance. High vapor pressure case (top) and low vapor pressure case (bottom)

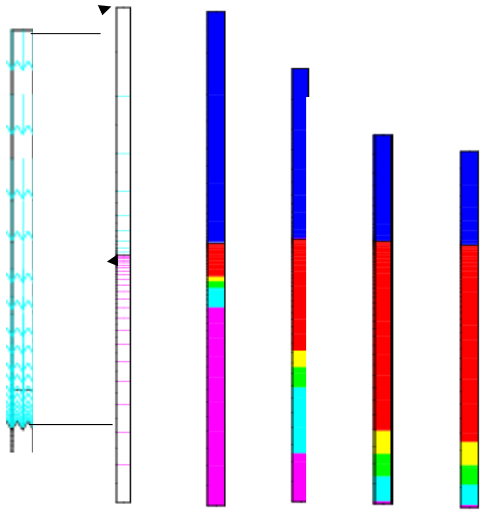
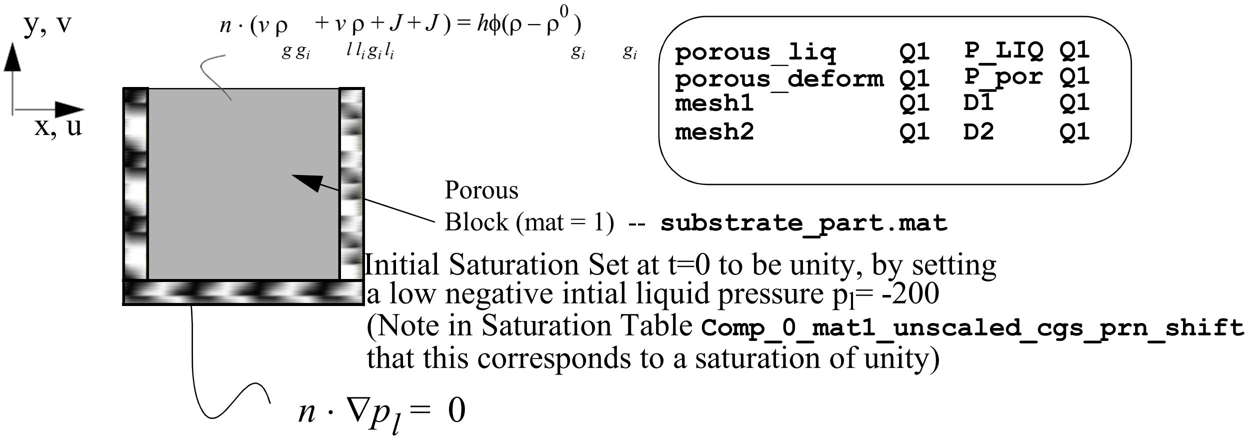


Figure 6. Verification of mass conservation. Measure inventory of liquid uptake in porous substrate equals the mass-loss from the continous film.

Drying of a porous deformable material (capillary shrinkage and springback)

This problem tests poroelasticity, capillary shrinkage and partially saturated flow. The geometry, boundary conditions and other relevant features are pictured here:



Porous
Block (mat = 1) -- **`substrate_part.mat`**

Initial Saturation Set at t=0 to be unity, by setting a low negative intial liquid pressure p_l= -200 (Note in Saturation Table **`Comp_0_mat1_unscaled_cgs_prn_shift`** that this corresponds to a saturation of unity)

The problem input files can be found in the directory **`sat_wick`**. We basically have a porous deformable material, initially saturated with liquid, undergoing drying and capillary shrinkage. With the POROUS_FLUX boundary condition on the top of the domain, we specify a mass-transfer rate of vapor. Diffusion of vapor out of the domain there will create a disequilibrium with the liquid in the porous and drive more liquid into vapor, thereby lowering the saturation. However, as the capillary pressure rises (which happens when the saturation drops slightly), the network stress activated by the "Capillary Network Stress" card deforms the matrix (if it is soft enough which in this example it is intentionally made that way). As the matrix compresses under this capillary stress, the porosity drops and hence the saturation is then held high. This shrinkage continues until the network stress exceeds the capillary pressure, in which case the menisci then recede into the film. Stress can release in the unsaturating regions and the network pops back out. Some sample results are shown in Figure 7. It is quite instructive to try different cases. If the Lame coefficients of the network are raised to real high numbers (viz. 1.e10), then the network never compresses and the drying front proceeds through the film.

The distinguishing features of this test problem are as follows. First, in the input file input.unsat_dry we have the following initialize cards:

```
#Initial Guess          = read_exoII
Initial Guess          = zero
Initialize             = POR_POROSITY 0 {poros_mat1}
Initialize = POR_LIQ_PRES 0 {-200}
```

Note that not only are we initializing the pressure, but also the porosity. We are not solving the solid-component mass balance (as described above in Eq. 9) and the degree of freedom for that balance is the porosity. Hence, the porosity is no longer taken from the material property **`Porosity`** card, but from the initialization. The liquid phase pressure is initialized to -200, which is commensurate with

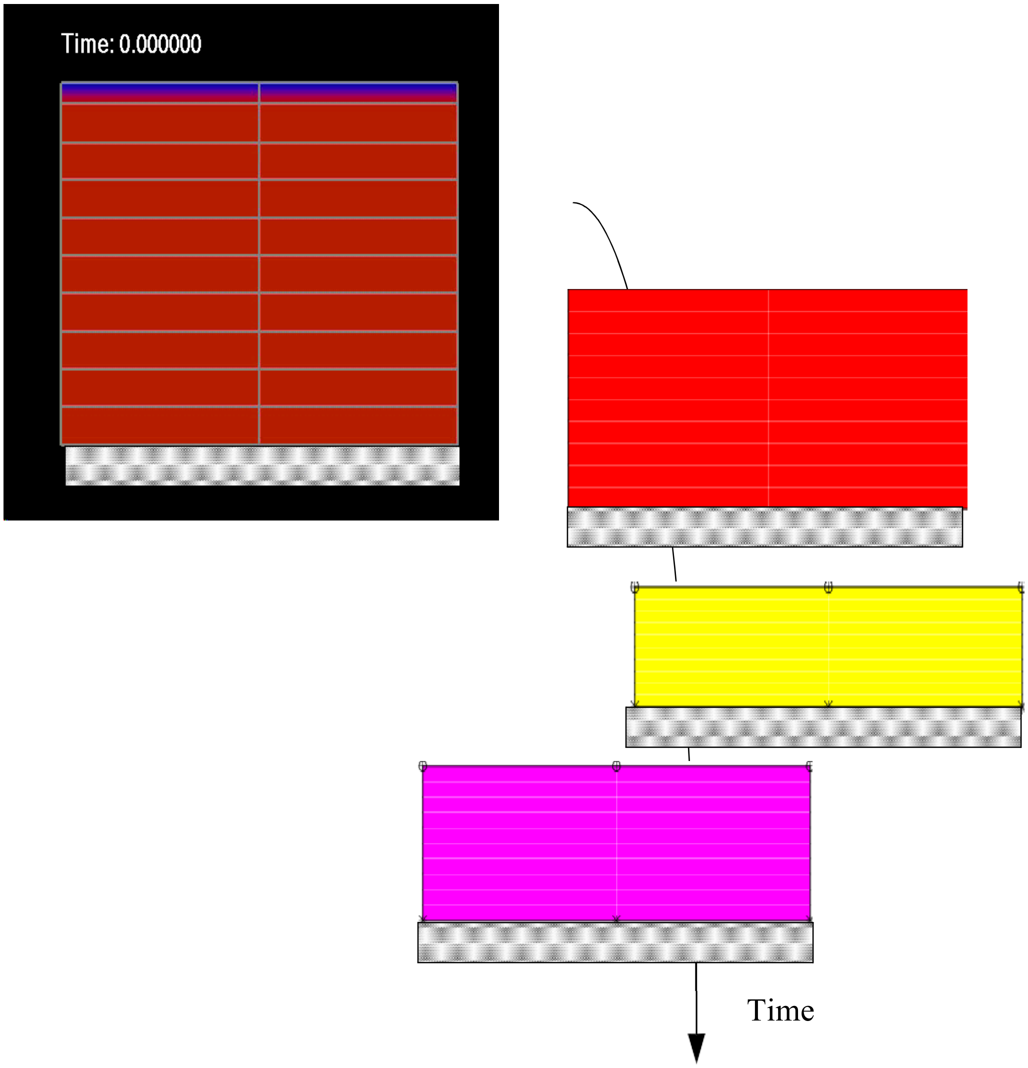


Figure 7. Capillary shrinkage induced by directional drying. Time sequence from top to bottom. Notice how a maximum shrinkage is reached as the pore space empties, followed by a small springback.

a saturation of unity from the saturation vs. vapor pressure curve. In this case that is a table function (see porous_unsat.mat file) and the table file is called `Comp_0_mat1_unscaled_cgs.prn_shift`

We need Lagrangian mesh motion equations for the network stress and displacement vector and the two component mass balances. The corresponding equations and boundary conditions are:

```
Boundary Condition Specifications
 ---
Number of BC = -1
$BC =   POROUS_LIQ_PRESSURE     NS      1     {ambpres = -700}
BC =   DX      NS     1       0.
BC =   DY      NS     1       0.
BC =   DX      NS     2           0.
```

```
BC =    DX      NS     4           0.
BC =    DX      NS     5           0.
$BC = POROUS_LIQ_PRESSURE NS 3 {-2.8e+2}
BC = POROUS_FLUX SS 30 0 1.0 0.0 0.0 0.0

$BC =    POROUS_LIQ_PRESSURE     NS      3 {0.9 * ambpres}
$BC = POROUS_KIN SS 30 0 0.
#########
END OF BC
#########

----
Problem Description
---

Number of Materials = 1

 MAT = porous_unsat 1

  Coordinate System = CARTESIAN
  Element Mapping = isoparametric
  Mesh Motion = LAGRANGIAN
  Number of bulk species = 1

 Number of EQ = 4
 EQ = porous_deform Q1 P_por Q1          1. 1. 1. 1.  0.
 EQ = porous_liq  Q1     P_LIQ      Q1    1.  1.    1.   1.   0.
 EQ = mesh1      Q1      D1      Q1       0.  0.    1.   1.   1.  0.
 EQ = mesh2      Q1      D2      Q1       0.  0.    1.   1.   1.  0.
                             div ms adv bnd dif src porous
```

Note the Lagrangian Mesh Motion as well. The boundary conditions are simple: slippery walls and drying from the top only.

Read the README file in this directory for further hints on the important features of this problem. Parameters that you can change to get different effects are the network Lame coefficients, viz.

```
Lame MU    = CONSTANT 1.e5

Lame LAMBDA = CONSTANT 1.e5
```

Note that when the liquid phase pressure (or capillary pressure) rises to this magnitude, significant deformation will occur. Of course it will as the network saturation goes down. Also you will find that the porous vapor pressure and other material properties can lead to interesting effects.

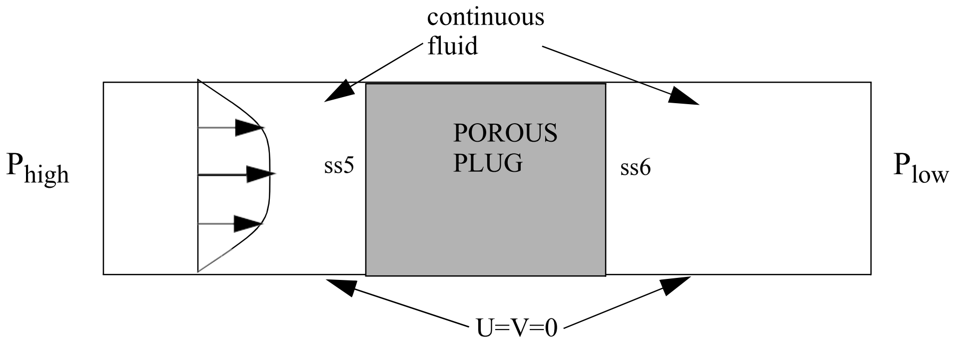
```
/*For compressible must have power law lame's */
Capillary Network Stress = PARTIALLY_WETTING
Rel Gas Permeability     = CONSTANT  {1./gas_visc}
Rel Liq Permeability     = VAN_GENUCHTEN 0. 0. 5.0 {lqvisc}
Saturation           = TABLE 2 CAP_PRES LINEAR  FILE=Comp_0_mat1_unscaled_cgs.prn_shift
#Saturation            = VAN_GENUCHTEN 0.05 0. 8.403 82000

--Misc Porous numerical props (Porous Section 2)
$Porous Weight Function         = GALERKIN
$Porous Time Integration        = TAYLOR_GALERKIN 0
Porous Advective Scaling        = CONSTANT 0 1.
```

```
Porous Mass Lumping          = yes

---Porous Species Properties (Porous Section 3)
/* Darcy fickian only for saturated media */
Porous Diffusion Constitutive Equation =DARCY_FICKIAN
Porous Gas Diffusivity            = CONSTANT 0  0.0026

Porous Latent Heat Vaporization = CONSTANT 0 0.
Porous Latent Heat Fusion       = CONSTANT 0  0.
Porous Vapor Pressure          = FLAT      0 {vappres} {sldens} {30.} {rgas} {temp} {30.} {ambpres}
Porous Liquid Volume Expansion = CONSTANT 0  1.
```

The Saturation card is also noteworthy. You should have a look at the table files for the saturation versus capillary pressure curve to get a feel for what capillary pressures will result in lower saturations. This is key for the behavior and definitely one place where hysteresis will play a role. I have shifted the curve to the right or left to get different levels of shrinkage and springback. Note there are other table files named similarly in the directory.

Poroelastic examples: Darcy-Plug

This problem tests several coupled porous flow options and the accompanying boundary conditions between porous and continuous fluid regions (files are in subdirectory `porous_plug`). Note that this problem is also addressed in the tutorial for unsaturated flow (viz. GT 008.2), but with rigid media. We complicate matters here by extending the problem to porous, deformable media. In the previous memo we also addressed the solution to the problem with three different formulations (two of which were highly restrictive and only applied to rigid media). Here we focus on the general formulation for poroelastic media. The problem is diagrammed below



The problem is quite simple: a pressure drop is applied across a porous plug through a continuous fluid phase. The expected solution is one of steady state flow that transitions from Poiseuille to plug flow and back to Poiseuille flow. To run the rigid case simple type:

```
goma -a -i darcy_input
```

You will notice upon visualizing the results that we have flow from left to right (or top to bottom, in the actual exoII case), with the entire pressure drop occurring across the plug. As the previous memo indicates, the complex features of this problem pretty much regard the continuous phase/porous phase boundary conditions. `DARCY_CONTINUOUS` and `POROUS_PRESSURE` are the key boundary conditions. The results also indicate that we have specified perfect slip along the walls in the continuous phase, which means that we have no pressure drop there. It turns out that this is not an appropriate condition for the deformable case. Finally, note that the magnitude of the velocity in the y-direction in the continuous phase is 2.e-5 (viz. the variable `VY` in the `out_darcy.exoII` file) as is the y-component of the darcy velocity in the porous phase (viz. the variable `darcy_vel_l_1` in the `out_darcy.exoII` file). This verifies proper implementation of the boundary conditions. Note that the Darcy velocity is really the total volume average velocity, as the real fluid velocity in the pore space would be quite a bit higher due to the confined area.

A far more challenging problem is one in which the plug is deformable. Boundary conditions become more complex, and we in fact must solve a multidimensional problem, i.e., there is no 1D solution even with slippery walls. The input deck file for this case, in the same directory, is called `darcy_deform_input`. It is helpful to discuss the boundary conditions and other multiple material aspects of this file:

```
        Number of BC = -1
        $$
```

```
BC = DX NS 1  0.
BC = DY NS 1  0.
BC = DX NS 2  0.
BC = DY NS 2  0.
BC = DX NS 3  0.
BC = DY NS 3  0.
BC = DX NS 4  0.
BC = DY NS 4  0.

BC = SOLID_FLUID SS 5 2 1
BC = SOLID_FLUID SS 6 2 3
$BC = NO_SLIP SS 5 2 1
$BC = NO_SLIP SS 6 2 3
$ Major errors in using this.  Need to
$ check num jac etc.
BC = VELO_TANGENT_SOLID SS 5 2 1
BC = VELO_TANGENT_SOLID SS 6 2 3

BC  = U NS 2 0
BC  = V NS 2 0.

BC    = U NS 4 0
BC    = V NS 4 0.


$GD_CONST SS 3 R_PRESSURE 0 VELOCITY1 0 1.
$GD_TIME  SS 3 R_PRESSURE 0 SINUSOIDAL 0 1. 1.
$BC      = FLOW_PRESSURE SS 3 10.11
BC  = V NS 3 -0.01
BC  = U NS 3 0.

BC = POROUS_PRESSURE SS 5 2 1
BC = POROUS_PRESSURE SS 6 2 3
BC = DARCY_CONTINUOUS SS 5 2 1
BC = DARCY_CONTINUOUS SS 6 2 3
END OF BC

###########
----
Problem Description
---

Number of Materials = 3

MAT = 1   1
   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 Q11.                          0.
   EQ = mesh1Q2D1Q2      0.     0.        1.   1.      0.  0.
```

```
    EQ = mesh2Q2D2Q20.    0.        1.   1.      0.  0.

                                     div  ms  adv  bnd  dif    src   porou
MAT = substrate1  2
    Coordinate System = CARTESIAN
    Element Mapping                  = isoparametric
    Mesh Motion = LAGRANGIAN
    Number of bulk species = 0
    Number of EQ = 4
    EQ = porous_sat Q2 P_liq Q2       1.   1.   1.   1.   1.
    EQ = porous_deform Q2 P_por Q2       1.   1.   1.   1.   1.
    EQ = mesh1Q2  D1  Q2        0.   0.   1.   1.   0.
    EQ = mesh2Q2  D2  Q2  0.   0.   1.   1.   0.

MAT = 2   3
    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 Q11.                          0.
    EQ = mesh1Q2D1Q2      0.    0.        1.   1.      0.  0.
    EQ = mesh2Q2D2Q20.    0.        1.   1.      0.  0.
```

First, regarding the differential equations to be solved, note that in the two continuous phases, viz. `1.mat` and `3.mat`, we are solving the complete Navier-Stokes system of equations together with the moving mesh equations. In the porous phase we are solving the `porous_sat` equation for the liquid phase pressure, the `porous_deform` equation for the solid-network porosity, and the Lagrangian mesh motion equations for the network stress. Notice also that the mesh motion type is `LAGRANGIAN` for the porous material and `ARBITRARY` for the continuous phases. The corresponding material data files (*.mat) are specified accordingly for a fully saturated flow.

The boundary conditions on this geometry are perhaps the most tricky part of the analysis. We begin with the no-slip condition on the side walls of the continuous phase, nodesets 2 and 4 (specification of U and V velocity components). *It should be noted that this problem does not converge well, or not at all, if you try to make these slippery walls.* Some pathology in the formulation arises and the pressure field goes awry. It seems to have something to do with adjoining a truly 1D solution and a 2D solution, for this incompressible situation.

Note also that the mesh is pinned around the entire outer edge. The mesh motion is Lagrangian only in the porous substrate, the ends of which are clamped at the side walls. Because the mesh must also move in the continuous liquid regions to accommodate the interfacial motion of the plug, we must solve for the mesh equations in all phases. Pinning the mesh at the edges everywhere was an expedient, as more general conditions in the `ARBITRARY` phases may be more flexible. The remainder of the boundary conditions deal with the fluid-structure interfaces, side sets 5 and 6. We begin with the Navier-Stokes equations. There we need make some statement about the rate of penetration being proportional to the pressure gradient in the porous phase. This condition is handled with the `DARCY_CONTINOUS` boundary condition, which serves as a kinematic condition for both the interfaces

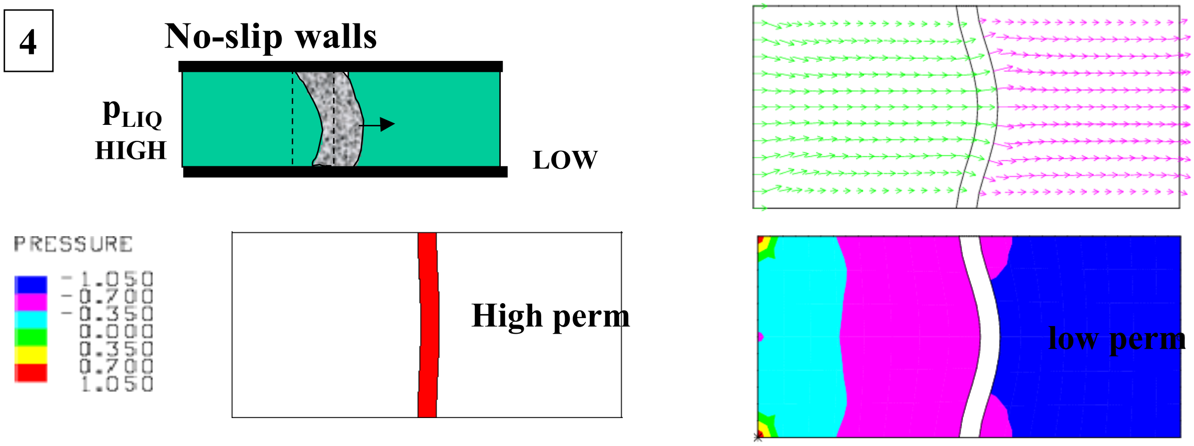


Figure 8. Flow across a porous plug. Problem description (top left), velocity profile in continous fluid phase (top right), plug deformation for high permeability (bottom left), and plug deformation for low permeability (bottom right). This is a three material GOMA test problem.

(side set 5 and 6 system) by satisfying the balance of mass to and from the porous material. Together with this condition on penetration/expulsion rate, we need a condition for the pressure in the continuous liquid phase in the porous material. Notice that we use **POROUS_PRESSURE** for this reason, which equates the two and sets the pressure level in the pore liquid. We also have a solid-stress differential equation which needs a boundary condition. It is tempting to think that the porous pressure equation is that boundary condition between the fluid and the structure, but we need more for the additional displacement degrees of freedom. For that we use the stress balance between the fluid and the drained network solid, which is invoked by the card **SOLID_FLUID**. Again we need to apply this condition on both sides of the interface.

It is interesting to have a look at the solution without the fluid-structural stress balance. The membrane expands slightly due to the elevated pressure in the liquid on the upstream, and interestingly shows a small deformation to the windward high pressure side.

The final boundary condition worthy of discussion is the **VELO_TANGENT_SOLID**, which is used to specify no slip along the membrane surface, viz. zero tangential velocity. The problem does not converge without these conditions.

Sample results are shown in Figure 8. Notice that the solution for a high permeability shows smaller deformation, because as expected the pressure drop isn't as great. Also notice the singular pressure fields in the inlet region due to a boundary layer-development along the wall. All other intuitive tests on this problem check out, but this in no way verifies the quantitatively correct solution.

Poroelastic examples: Saturated Sponge Squish/Pull

This example is used to demonstrate the effects of applied external mechanical loads on a saturated porous material. The two simple test cases involve a saturated network, or sponge, squished against a screen and stretched in an infinite bath of pore-liquid. Both cases are designed to remain completely saturated throughout the test. Note that before running these problems we suggest running the three simple tests in the directory `simple_sat_flow_tsts`. In that directory look at the `README` file for a description of the problems. These problems will give you a feel for adding deformability to an otherwise rigid saturated porous flow problem.

The first problem we will discuss is located in `saturated_squish`. In that directory the base input file `q1.input`. The problem description is illustrated in Figure 9. Basically we start with a saturated porous medium fully contained on the sides (viz. the problem is 1D) and squish it with a prescribed moving boundary against a screen on the bottom. The "screen" boundary condition implies that the liquid sees the atmosphere and can leave, but the solid network cannot leave the domain as it is squished up against the screen. The proper boundary conditions and equations are as follows.

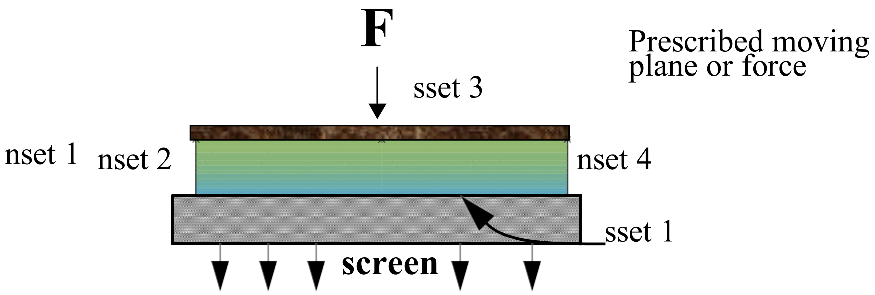


Figure 9. Saturated sponge squish against a screen.

```
Number of BC = -1
BC = MOVING_PLANE SS 3 0. 1. 0. -0.3 0.1 0.0 0.0
BC                  = DX NS 1 0. 1.0
BC                  = DY NS 1 0. 1.0
BC                  = DX NS 2 0. 1.0
BC                  = DX NS 4 0. 1.0

$$ Backpressure on underside of web (see page 109 of 2.0 Revised edition man)
BC      = POROUS_LIQ_PRESSURE NS 1 0.0

END OF BC
##########
```

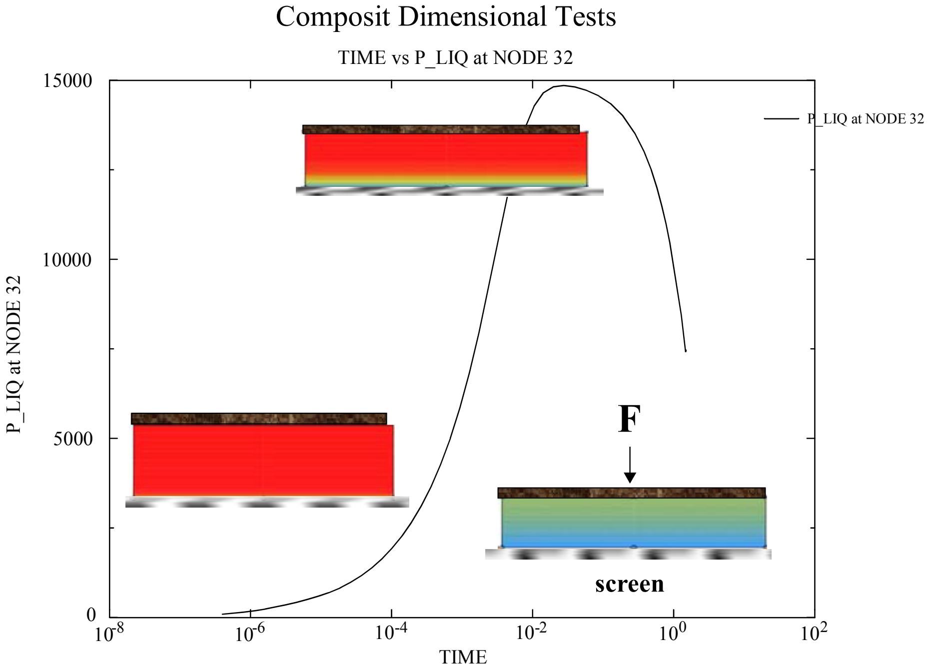


Figure 10. Liquid phase pressure at moving plate versus time. Notice rapid rise (log scale in time) to set up a pressure gradient capable of driving a flow rate that matches the prescribed displacement, followed by gradual fall in pressure as the sponge is squeezed.

```
Problem Description
Number of Materials = 1
MAT = substrate2 1
    Coordinate System = CARTESIAN
    Element Mapping = isoparametric
    Mesh Motion = LAGRANGIAN
    Number of bulk species = 0

    Number of EQ = 4
      EQ = porous_sat Q1 P_liq Q1        1. 1. 1. 1.  0.
      EQ = porous_deform Q1 P_por Q1 1. 1. 1. 1. 0.
      EQ = mesh1Q1  D1  Q1       1.  0.  1.  1.  1.
      EQ = mesh2Q1  D2  Q1       1.  0.  1.  1.  1.
```

As in the previous case we must solve for the liquid phase pressure, the solid concentration (porosity), and the solid displacement (mesh motion equations). This is a Lagrangian solid calculation. The boundary conditions indicate that we are containing the side walls and the top with respect to the pore liquid (viz. the natural boundary condition for the Darcy liquid flux is the zero flux on a fixed or moving interface, and so you specify nothing). ON the bottom we set the pore-liquid pressure to zero, indicating an atmospheric datum pressure, above which the pore pressure must exceed for the liquid to leave the medium. Note that on the top of the domain (sset 3) we set the surface to move at a velocity of 0.1, in the negative-y direction).

We expect that the pressure field at the moving surface will spike up, a gradient to set up, and the liquid to be squeezed out. This is indeed what happens, as shown in Figure 10. If we integrate the outflow Darcy velocity field over time and compare it with a calculation of the lost mass based on the porosity change and the overall volume change, we find that the mass-conservation is good to within 4%.

Underneath the directory **`saturated_squish`** is a directory named **`FORCE`**. The input and material files are basically the same, but the same squish problem is run a bit differently. The boundary conditions are as follows:

```
Number of BC = -1
$$$$delta_t of force ramp {delta_t=1.e-5}
BC   = FORCE_USER SS  3 {delta_t} 0. -150. 0.
$BC  = FORCE SS  3  0. -150 0.
$
BC                                = DX NS 1 0. 1.0
BC                                = DY NS 1 0. 1.0

BC                                = DX NS 2 0. 1.0
BC                                = DX NS 4 0. 1.0

$$ Backpressure on underside of web (see page 109 of 2.0 Revised edition
man)
BC      = POROUS_LIQ_PRESSURE NS 1 0.0
END OF BC
```

In place of the **`MOVING_PLANE`** condition we used in the first case, we now use an applied force and then follow the displacement versus time. This problem is much harder to run than the applied moving plane case, and in fact requires smaller time steps and denser meshes. Applying an instantaneous, y-directed force of -150 to side set 3 in fact leads to poor convergence. We found it advantageous to apply a ramped force over the first 1.e-05 time units from zero to -150. Notice that this is accomplished with the use of the user-defined force boundary condition option **`FORCE_USER`**. A snippet of code written in GOMA that allows for this ramp in the routine **`force_user_surf`** is

```
  if (time <= p[0])
     {
        func[0] = p[1]*time/p[0]; /* + fv->snormal[0]*fv->p_liq;*/
        func[1] = p[2]*time/p[0]; /* + fv->snormal[1]*fv->p_liq;*/
        func[2] = p[3]*time/p[0]; /* + fv->snormal[2]*fv->p_liq;*/
     }
   else
     {
         func[0] = p[1]; /* + fv->snormal[0]*fv->p_liq; */
         func[1] = p[2]; /* + fv->snormal[1]*fv->p_liq; */
          func[2] = p[3]; /* + fv->snormal[2]*fv->p_liq; */
     }
/* Add your function and sensitivities here */
```

Notice how the four parameters on the FORCE_USER card are used in this simple code. We also found it necessary to grade the mesh heavily to the bottom screen in anticipation that steep pressure gradients would develop there. Figure 11 shows a sample result. We plot the displacement field in the solid versus depth of the porous layer. Notice that at early times the deformation occurs in a narrow boundary layer at the bottom near the screen, where the pressure is the lowest. The layers then

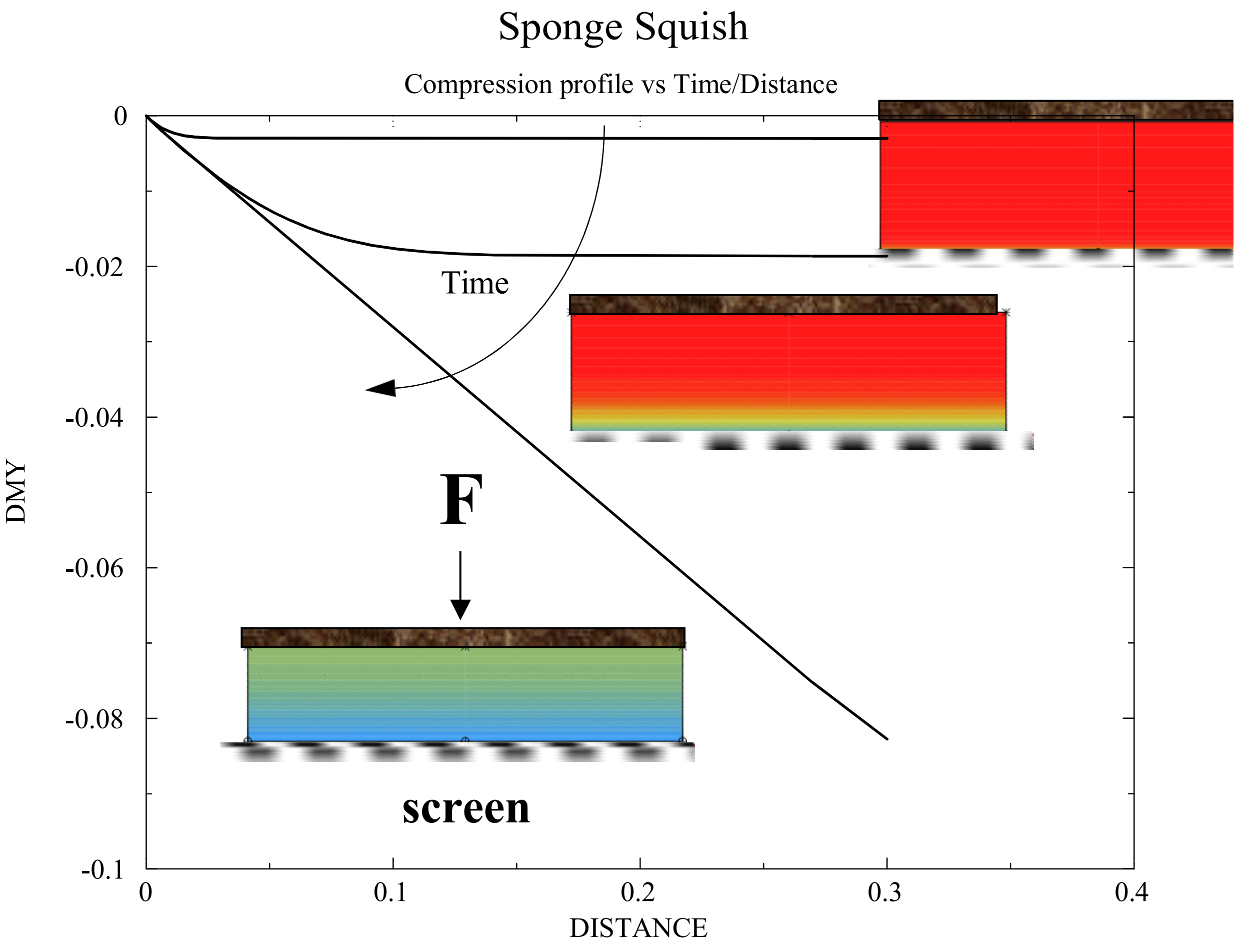
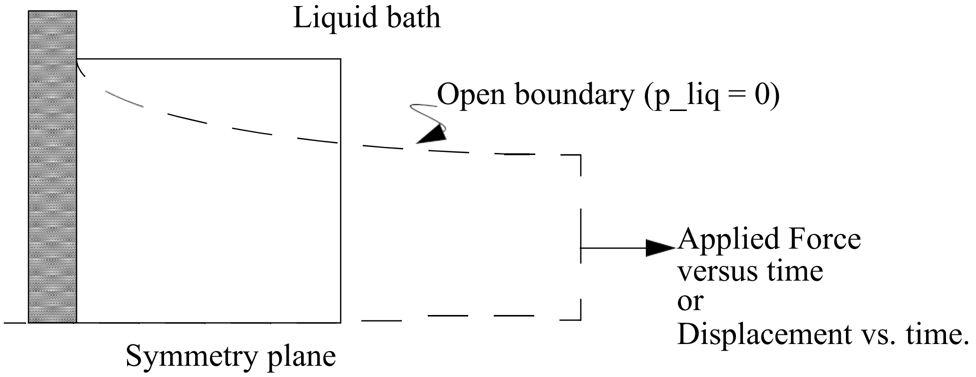


Figure 11. Vertical displacement profile for the saturated solid network at three different times. A force is applied on the top plate and the liquid is squeezed out to the atmosphere at the bottom. Notice the displacement gradient in the lower layers at early times, indicating that the bottom layers of the network collapse first.

collapse like a “deck of cards” as the boundary layer grows to meet the moving plate. The color contours are of pressure, showing that at early times the pressure gradient is greatest near the screen.

Our final example in this section regards a sponge pull test. The problem is diagrammed below. Interestingly, this problem is not too well posed if the solid-network by itself is compressible; viz. when stretched volume is created, and to fill that volume liquid must be allowed to come into the domain. If no liquid is allowed to enter or leave the domain then the total pore-volume must stay constant upon stretching. We run several of these cases to see if the force response of a stretching, saturated porous medium follows our intuition.

This problem is located in the directory `stretch_bend`(under the directory `porous_tests`). The input file relevant to this discussion is `input_deform_sat`. The equations solved are exactly the



same as those in the previous two problems in this section. The boundary conditions, however, are slightly different:

```
Number of BC                    =   -1
BC = DX NS 6 0.0
BC = DY NS 6 0.0
BC = DY NS 4 0.
$BC = FORCE_USER SS 13 -1000.
$BC = FORCE SS 3 {0.0000000179117} 0. 0.
$BC = FORCE_USER SS 3  0.0001 100000000.51 0. 0.
BC = MOVING_PLANE SS 3 1. 0. 0. -26.7e-3 -1.0 0.0 0.0
$BC =    FIX     NS      4       MESH_DISPLACEMENT1   0
$BC =    FIX     NS      4       MESH_DISPLACEMENT2   0

BC    = POROUS_LIQ_PRESSURE NS 2 0.

END OF BC
```

We again use the moving plane boundary condition to deform the sample. Also notice that we specify the liquid phase pressure to be zero on the exposed side boundary (nset 2). This implies that as the liquid in the pore-space goes into tension during stretching (viz. a negative pressure) liquid will be brought into from the boundary to relieve the disequilibrium. We fully pin the sample on the left hand boundary (nset 6). Figure 12 shows some sample results. Shown is the integrated force on the moving plane boundary versus time during the pull. In this case we pulled for 0.01 time units (see `Maximum Time` card in input deck) and then held the sample in the stressed state for it to equilibrate. With a dry sample, viz. no dissipation from the pore liquid) the pull force rises in an expected nonlinear fashion according to a neoHookean elastic solid to a constant state (during which the sample is held in the deformed position). Notice that with pore-liquid involved the stress overshoots the steady state due to the hydraulic resistance that arises from the Darcy-flow rearrangement. Moreover, this maximum overshoot depends on the rate-of-pull, indicating a viscoelastic solid response.

It should be noted that in the same directory you will see a file named `input_deform_sat_cont`, which is virtually the same as the `input_deform_sat` file, except that it is used to restart the solution after the maximum pull length is reached. At that point the boundary conditions are changed so that the displacement on the end is fixed and liquid is sucked into the pore space due to the differential stress until equilibrium is reached. It is interesting to observe the effects of permeability and pull rate

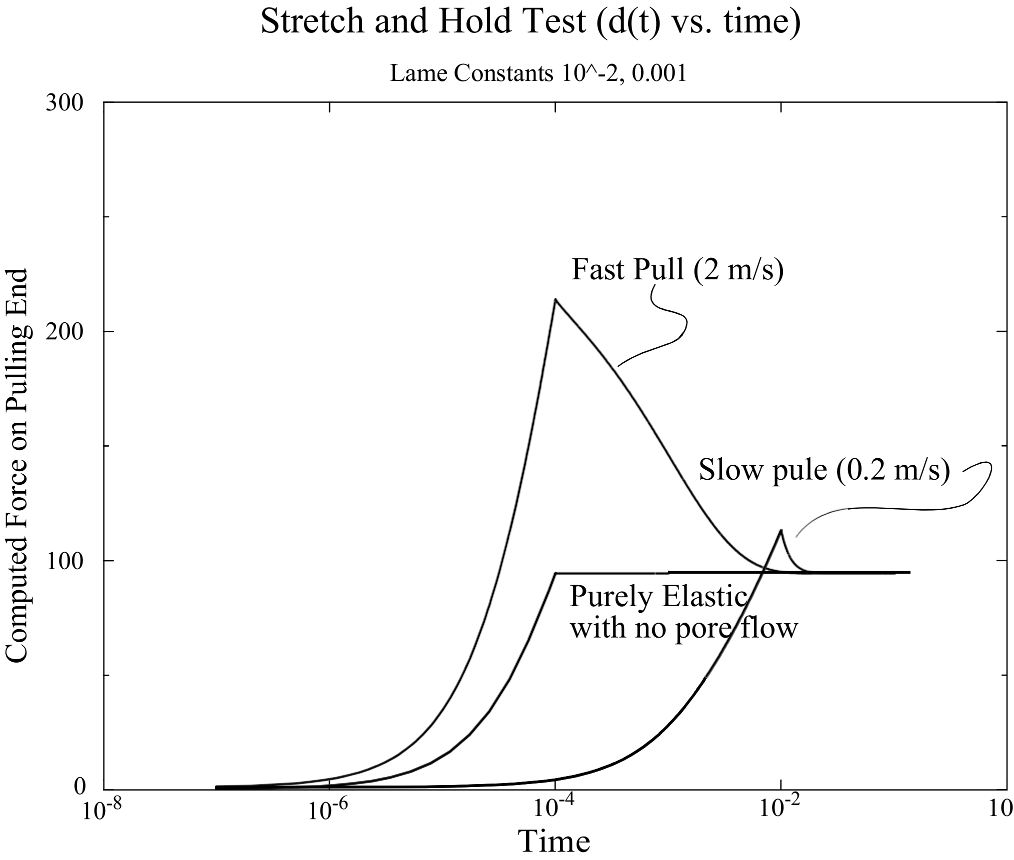


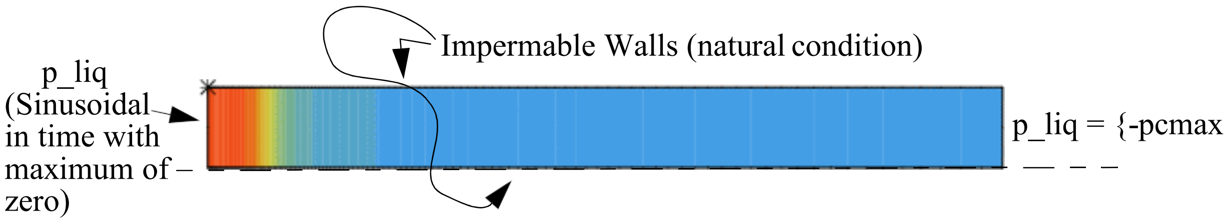
Figure 12. Total force on pulling end versus time. In all cases pulling was initiated at constant speed and held to 0.01 time units, at which time the pulling was stopped and the sample held in place. Notice the final stress state is the same in all cases, as expected, but the viscoelastic effects are evident when there is pore liquid.

on the shape of the sample. You will find the faster you pull and the less permeable the material, the more “necking”, or network compression you will see before the recovery. All this seems intuitive and qualitatively correct.

Partially Saturated Flow with Hysteretic Saturation Function (Rigid Materials)

Nearly all porous materials for which the porous-flow capability in *Goma* was designed are characterized by highly nonuniform pore sizes. Actually most permeable microstructures, including simple ones made up of close-packed spheres, are characterized by channels that are tortuous with numerous expansions or contractions; this is especially the case for disordered media. It is well known that the saturation function, viz. the often empirically-determined saturation vs. capillary pressure relationship central to our modeling capability, exhibits hysteresis. That is, multiple saturations are possible at a single capillary pressure. The appropriate saturation depends on how the state-of-saturation came to be, viz. from imbibition or from drainage. Upon draining a porous material either by drying, applied pressure gradient, or applied mechanical compressive load, the capillary pressure that must be overcome to drain the smallest of pores is much larger than that required to fill the pores. The reasons underpinning this behavior are several, the most important of which involves the fact that liquid can often support considerable tension before cavitating. This means under a pressure gradient the menisci in the smallest pores may be able to migrate to achieve even larger radii of curvature, hence larger capillary pressure. Previous research has proved that this hysteretic phenomena, while involving the characteristic contact angle of liquid with the solid network, is mainly driven by a pore-size distribution effect.

Goma has recently been extended with a saturation constitutive equation that includes hysteresis effects. The underlying formulation and related theory is presented above. The test problem we use for rigid materials is located in a directory called **`1D_hysteresis`**. In that directory you will find an input file, **`input_hyst`**, and a material file **`substrate_part.mat`**. The problem description is shown in the following figure:



MAT = substrate_part 1
Coordinate System = CYLINDRICAL
Mesh Motion = LAGRANGIAN
Number of EQ = 1
EQ = porous_unsat Q1 P_LIQ Q1 1. 1. 1. 1. 1.
ms adv bnd dif src por

The geometry is cylindrical but the saturation front moves left-to-right on average in the axial direction only (implying in this 1D problem that we could run this with a CARTESIAN coordinate system). We are solving for partially saturated flow in this 1D column of porous material. The left boundary is subjected to a sinusoidal liquid-phase pressure fluctuation. The gas pressure is taken as zero throughout and the initial saturation is also set to 0.14 by initializing the liquid phase pressure accordingly (see saturation function parameters below). The liquid-phase pressure varies sinusoidally between zero (max) and -14000 via the generalized Dirichlet boundary condition set (cf. **`input_hyst`** and the variable definitions file **`Defs`**):

```
BC = GD_LINEAR SS 6 R_POR_LIQ_PRES 0 POR_LIQ_PRES 0 {-(pcmax-pcmin)/2.0} 0.0
BC = GD_TIME SS 6 R_POR_LIQ_PRES 0 SINUSOIDAL 0 {3.14159/2.0} {2.0*3.14159/1.0}
BC = GD_LINEAR SS 6 R_POR_LIQ_PRES 0 POR_LIQ_PRES 0 {-pcmin-(pcmax-pcmin)/2.0} -1.0
```

The net effect, regardless of the saturation function, is pulse-pumping of a saturated plug from left to right (actually, it would be good to set the far-field boundary condition on the right to a liquid pressure corresponding to the minimum residual saturation state with the card {**`BC = POROUS_LIQ_PRESSURE NS 3 {-pcmax}`**}. The **`aprepro`** variable **`pcmax`** is set in the Defs file and defines the minimum level of saturation in the problem. Note that not applying any boundary condition to the right-hand boundary, equivalent to an impenetrable boundary, leads to some noisy fluctuations at early time.

The critical part of this problem is the saturation function. In the **`substrate_part.mat`** file you will find the following set of cards:

```
$Saturation          = TANH 0.1163 0.0 2.5 10000.0
   {draining = 0.0}
$Hyst 1:
$Saturation          = TANH_HYST 1. 0.0 2.5 10000.0 0.1163 0.0 1.2 90000.0 {draining}
$Hyst2
Saturation          = TANH_HYST 1. 0.0 2.5 10000.0 0.1163 0.0 2.5 60000.0 {draining}
$hyst 3: Du
$Saturation           = TANH_HYST 1. 0.0 2.0 10000.0 0.1163 0.0 2. 10000.0 {draining}
```

The test cases shown here involve each of these saturation function specifications. The two we will discuss are the *TANH* model (first, commented out card) and the *TANH_HYST* model for saturation labeled **`Hyst2`** (the uncommented card). Each of these functions and the actual experimental data to which they were fit (viz. the data that gives rise to the various constants on the card) are shown in the following figure.

The definitions of the constants on the *TANH_HYST* model are described in detail in the Goma 4.0 manual. The *TANH* saturation model results in all saturation/capillary pressure evaluations to occur on the "main wetting curve" as shown in Figure 13, the parameters for which being specified on the card. These parameters were obtained from a crude curve fit to the data also illustrated in the figure. The *TANH_HYST* model invokes a scanning curve concept illustrated in the Figure 13 and explained at the outset of this report in the theoretical section. In that case the saturation/capillary pressure relationship switches from a wetting curve and a draining curve, both of which are scaled versions of the main wetting and draining curves as shown. This approach is explained in detail by Kool and Parker (1987). Figure 14 shows some sample results using both modes saturation function. Shown is the saturation versus time at a specific point in the domain (specifically, node 26 which is located about 1/5 of the way into the domain from the left). Note that the saturation fluctuates as expected, and that the progression of the saturation at that point to a fully saturated state is never reached. Actually, results show that liquid drains out of the left side of the domain when the pressure is negative, thereby impeding the progress of the saturation front to the right. Note that with the hysteresis model the fluctuation in saturation level at the same node is much smaller, and in fact the

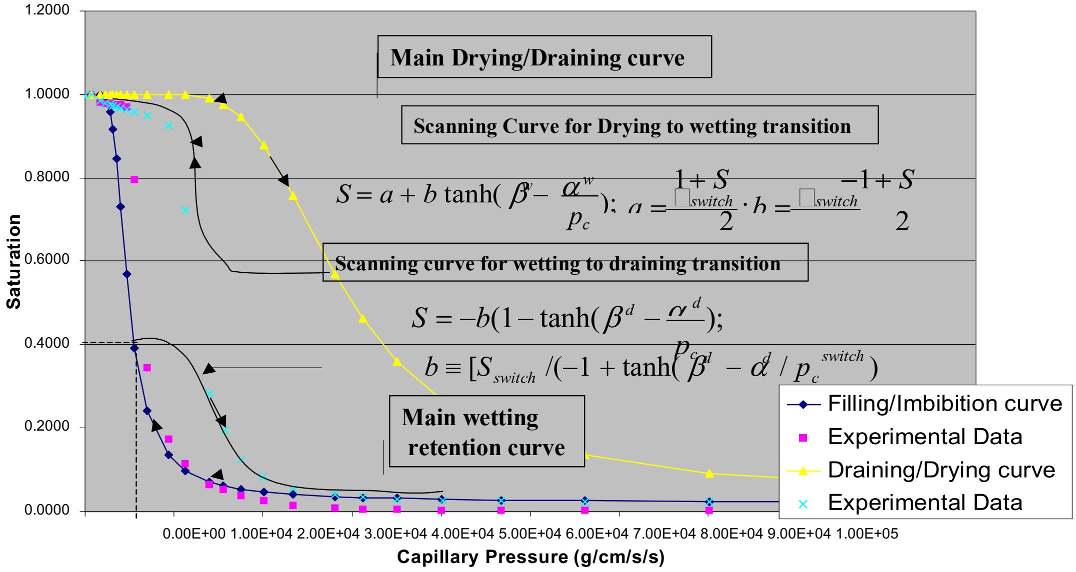


Figure 13. Hysteretic saturation curves.

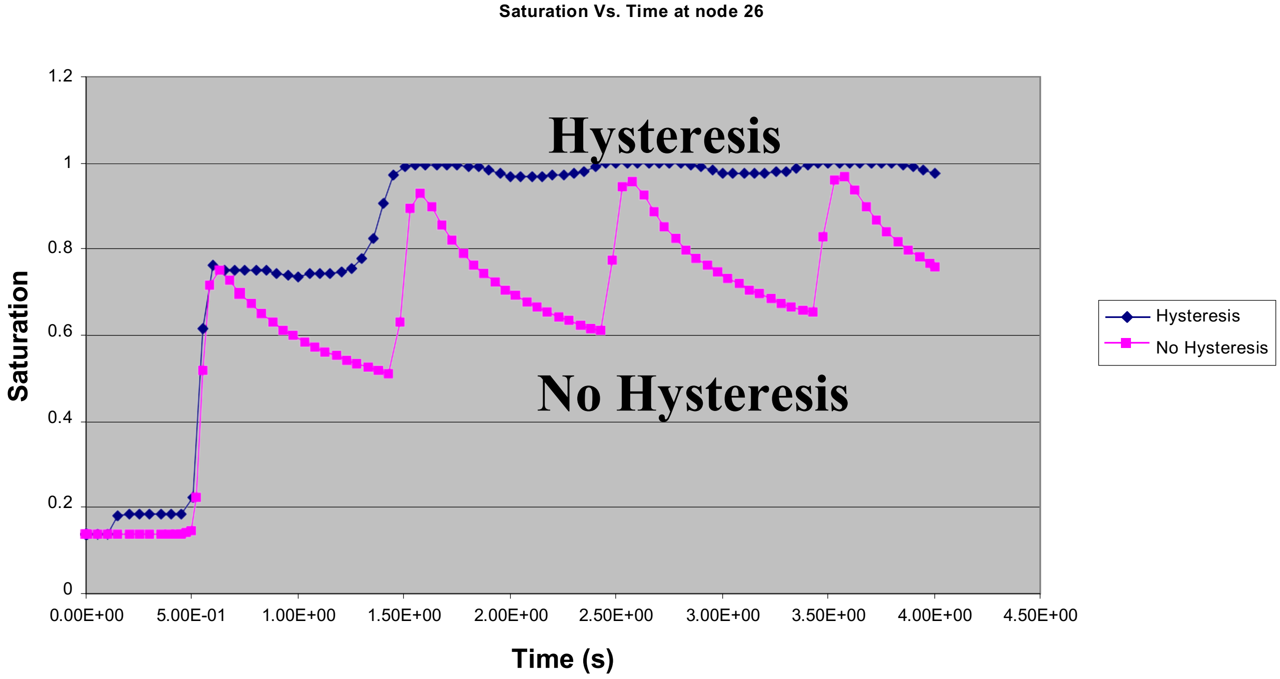


Figure 14. : Saturation vs. time at node 26 (approx. 1/5 the distance across from left to right of the 1D sample). Properties for the hysteretic case correspond to the curves in Figure 13.

medium effectively is saturated faster with a larger net flow from left to right. This “pumping” effect is a result of the hysteretic saturation function. We show in Figure 15 the trajectory taken by the saturation level, overlayed on the saturation versus capillary pressure curves, viz. the trajectory created by plotting saturation vs. capillary pressure at a single point (node 26 in this problem) throughout time. Note that the saturation level at node 26 begins on the main wetting curve and

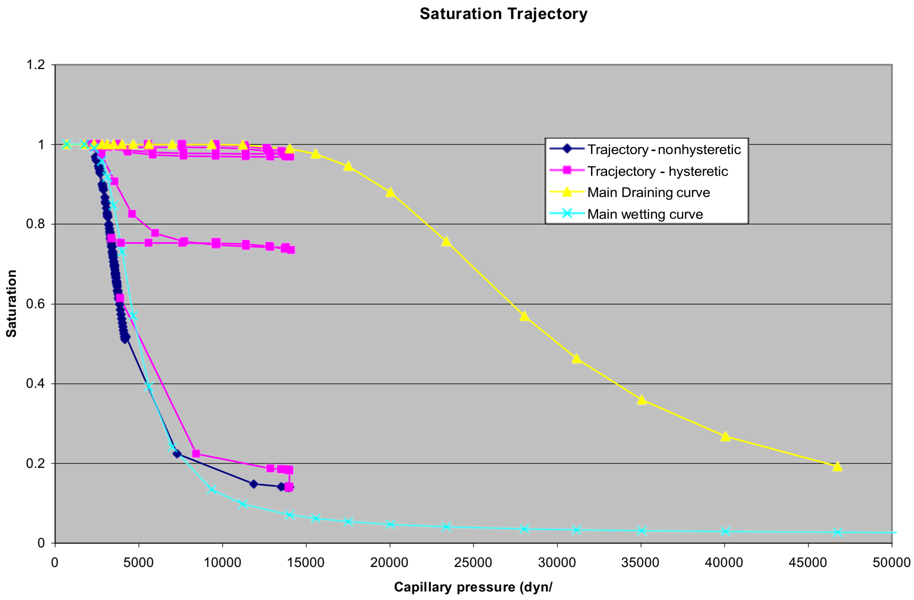
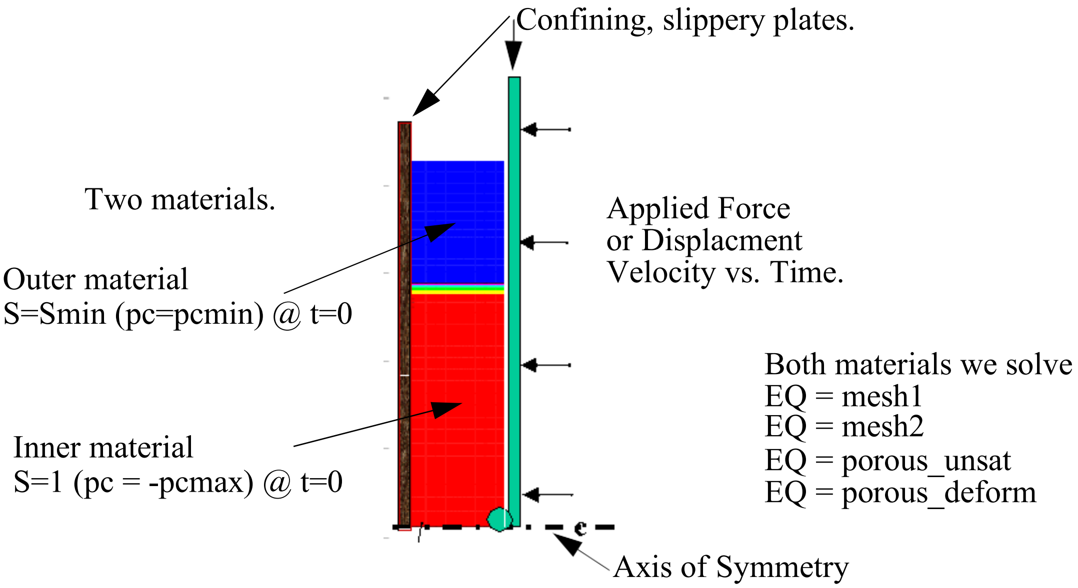


Figure 15. Saturation trajectory during pumping process, for both the hysteretic and nonhysteretic case.

progresses up to about 0.8 before it switches to a scanning draining curve. That curve is basically a scaled version of the main draining curve and is followed until the next switch point is reached. You can see that eventually the saturation fluctuates along the upper plateau of the main draining curve and the main wetting curve a latter times, indicative of the pumping effect.

Poroelastic Examples: Unsaturated Squish, Pull

At the time of this revision, the unsaturated deformable capability with hysteretic saturation functions was not fully functional. The example problem here uses a non-hysteretic function for illustration purposes. The problem is diagramed in the following figure.



The directory that contains this sample problem is called **`unsat_squish`**. In that directory you will see a series of input files. Start with the simplest one: **`input_squeeze`**. In that file you will notice that we are solving the component mass balance for a single component liquid and its vapor in a porous material. That equation is the usual **`porous_unsat`** differential equation. In addition to that equation we require a component mass balance on the solid, which is covered with the **`porous_deform`** equation specification. The momentum balance for the deformable medium is covered with the Lagrangian **`mesh1`** and **`mesh2`** equations. The input deck problem description section looks as follows:

```
----
Problem Description
---

Number of Materials = 2
MAT = substrate_1 1
####### Both regions use same material file
###MAT =   substrate_2   1

  Coordinate System = CYLINDRICAL
  Element Mapping = isoparametric
  Mesh Motion = LAGRANGIAN
  Number of bulk species = 0

       Number of EQ              = 4
        EQ = porous_unsat Q1 P_LIQ Q1 1. 1. 1. 1. 1.
        EQ = porous_deform Q1 P_POR Q1 1. 1. 1. 1. 1.
        EQ = mesh1       Q1 D1 Q1    0.   0.   1.   1.   1.
        EQ = mesh2       Q1 D2 Q1    0.   0.   1.   1.   1.
                              ms adv bnd dif src por
```

```
MAT = substrate_2 2

 Coordinate System = CYLINDRICAL
 Element Mapping = isoparametric
 Mesh Motion = LAGRANGIAN
 Number of bulk species = 0

      Number of EQ               = 4
       EQ = porous_unsat Q1 P_LIQ Q1 1. 1. 1. 1. 1.
       EQ = porous_deform Q1 P_POR Q1 1. 1. 1. 1. 1.
       EQ = mesh1       Q1 D1 Q1     0.  0.  1.  1.  1.
       EQ = mesh2       Q1 D2 Q1     0.  0.  1.  1.  1.
                           ms adv bnd dif src por
```

Notice that the geometry is axisymmetric (viz. we are squishing a disk that has a local saturated spot in the center) and that in both materials we are solving the same equation. The material files are substrate_1.mat and substrate_2.mat. Note that these files are basically the same as can be seen by the values of the imported **`Defs_squeeze`** file. The only difference lies in the initialization of the liquid phase pressure. Note that the difference there is

```
> #### Set such that the inital saturation level is 5%
> Initialize = POR_LIQ_PRES 0 -0.2300e+5
107,108c105,109
< Initialize = POR_LIQ_PRES 0 -7.00e+2
```

These cards appear at the bottom of the respective files. The high-negative initial porous liquid pressure corresponds to a low-saturation state (about 0.14), and the low-negative initial pressure corresponds to a saturation of about unity. These pressures were chosen from the saturation function curve, which is also set in the files as

```
#### while mat2 corresponds to AY (Flow from AY to DT)
Saturation          = TABLE 2 CAP_PRES LINEAR  FILE=Comp_0_mat1_unscaled_cgs.prn
```

The curve is read in from discrete data points found in the file **`Comp_0_mat1_unscaled_cgs.prn`**. The net effect of this initialization is a material with a saturated center region. Note that the only reason we split this problem into two materials is to accomplish this initialization. In the future we will deploy the *Goma* geometry function as a means of initializing phase pressure.

The boundary conditions on this problem are fairly straight forward. The section in input_squeeze looks as follows:

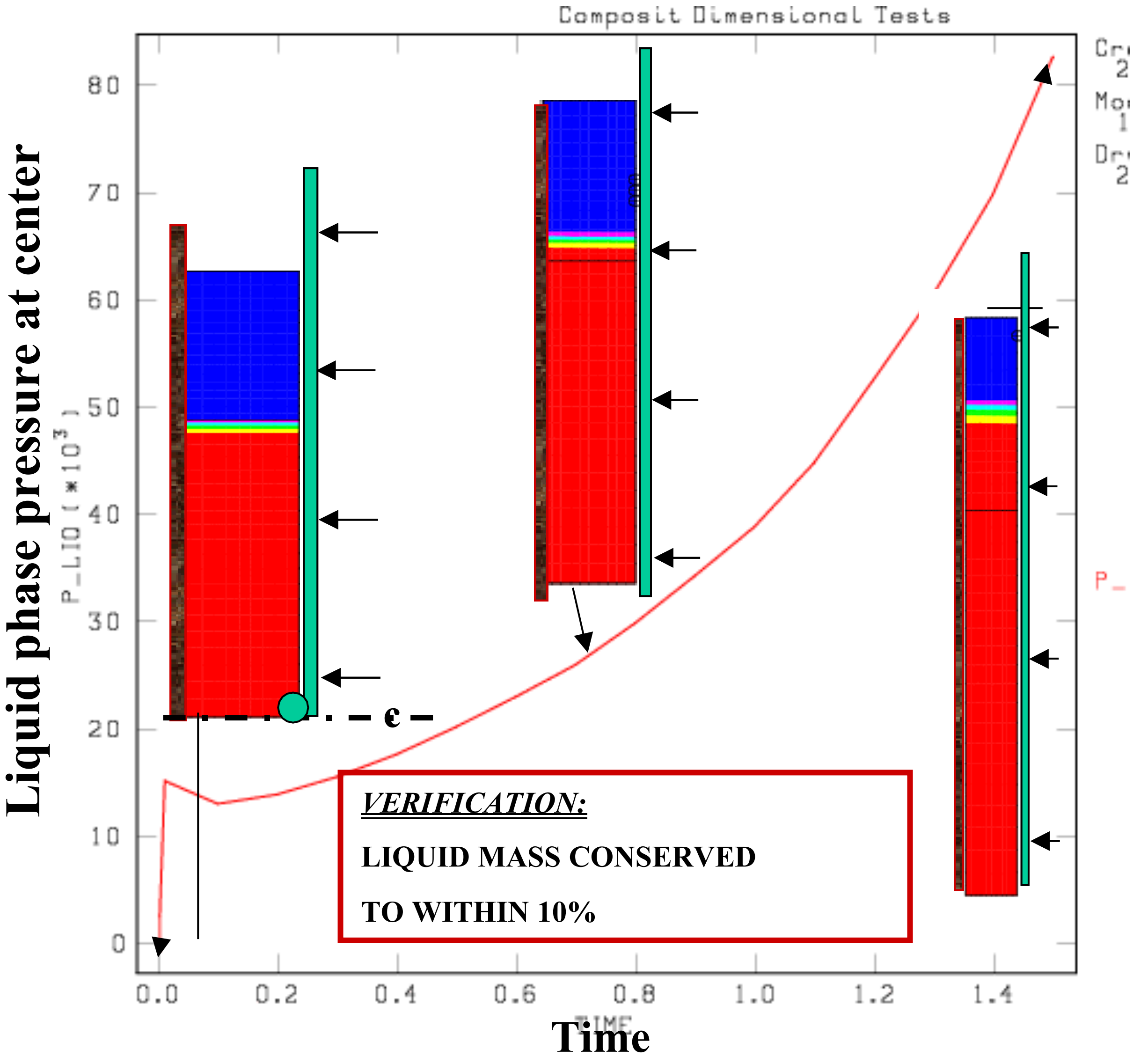
```
   ---
Boundary Condition Specifications
 ---
Number of BC                    =   -1
BC = DX NS 1 0.
$BC = DY NS 1 0.
BC = DX NS 5 0.
$BC = DY NS 5 0.
$Axis of symmetry
BC = PLANE SS 4 0. 1. 0. 0.
$Applied displacement
BC = MOVING_PLANE SS 3 1. 0. 0. -0.3 0.1 0.0 0.0
```

```
$BC = FORCE SS 7 -1. 0. 0.
$BC = FORCE SS 3 -1. 0. 0.

END OF BC
```

The important features to mention here regard the application of x-displacements (or axial displacements) across the bottom disk of zero, without specifying y- or radial-displacements, resulting in a slippery boundary. If y-displacements where applied then the bulging solution would be obtained that is not uniform and is more limiting in extent of displacement. The user can try this if he/she wishes. `PLANE` conditions are applied to the axis and the top moving plate. The `PLANE` command also results in a slippery surface as it results in no tangential constraints. The moving plane in this example results in the top plate moving at a velocity of 0.1.You can also apply a force condition as shown on side set 3, instead of a PLANE condition, but it must be applied with a linear ramp up. The input deck `input_squeeze_hyst` contains an example of this.

The following figures show some sample results. As expected the result of squeeze the disk of partially saturated (centrally saturated) porous material is to force the liquid to the outside by the pressure generated from the squeezing motion. You can see that this is the result. The first figure shows that the liquid phase pressure rises with time at the center, and the liquid saturation front is forced outward. We verified that the liquid mass was conserved to within 10% in this calculation.

It is also noteworthy that we have had trouble releasing the force and letting the porous structure spring back, with hydraulic resistance resulting from liquid motion through the pores. At the time of this memo revision, we were working on this problem.



VERIFICATION:

LIQUID MASS CONSERVED

TO WITHIN 10%

Appendix: Limitations to Darcy-Potential Approach and Troubleshooting Tips

1. No velocity components are computed in the porous phase. They must be backed out from a user-defined post processing variable (i.e., DARCY_VEL_0, DARCY_VEL_1, VGAS0, VGAS1, etc).

2. When solving the **`POR_POROSITY`** equation for poroelastic problems ALWAYS initialize the porosity with the **`Initialize = POR_POROSITY`** card. Several models rely on an initial porosity and must have some initial value.

3. Jacobian errors only remain in mass-matrix term for lumped mass. Factor of 2.5. PRS (12/21/01)

4. Unchecked form of boundary conditions. Need to check whether porosity multipliers are needed a'la Scherer et al. for the porous pressure at the interface between a porous medium and a continuous phase. Also need to add the normal viscous stress piece from the fluid in POROUS_PRESSURE. ----PRS 7/01: use scale factor on FLUID_SOLID bc for FSI problems to scale the interaction with the solid-only area....Still don't think a porosity multiplier is needed for the POROUS_PRESSURE condition.

5. From the authors experience, the following points regarding this class of problems should be reviewed during troubleshooting:

   -DARCY_CONTINOUS boundary condition: make sure the liquid density in the continuous phase and LAGRANGIAN phase mat files are the same!!!

   -Linear elements, viz. Q1 elements, are better for saturation front startup at an external boundary if the difference between the boundary specified liquid-phase pressure and the medium-initialized liquid phase pressure are drastically different. Quadratic elements in this case can lead to zero or low Saturation values at all computational Gaussian integration points and the front may never penetrate.

   -Specifying a pressure gradient (using the **`POROUS_FLUX`** and other related boundary conditions, more below) at the boundary of a porous medium should be done with care for coupled continuous/porous flow problem. Pressure profiles in the porous phase and in the continuous phase are obviously the most important aspects of the problem and care must be taken in the boundary conditions. For saturated cases, it is often best to just specify the absolute pressure at both extremes if they are known for the real problem. For unsaturated cases, a further pathology arises from the capillary pressure-saturation relationship (Equation 26). One must be careful with the relationship of the base pressure datum of each region, so that high liquid pressures in the continuous phase do not push the liquid pressure in the bounding porous phase above the threshold for full saturation, artificially, through the **`POROUS_PRESSURE`** boundary condition.

   -At low saturations **`POROUS_FLUX`** is innocuous using the **`KELVIN`** vapor pressure model because the gas phase concentrations are so small. Use the **`FLAT`** vapor-pressure model to bring these up. This is a pathology of the saturation vs capillary pressure curve with which all pressure levels must be consistent.

-"`NaNs`" and "`Arithmetic Exception`" errors are usually caused by excessive capillary pressures, which by virtue of Kelvin's equation (Equation 23) can lead to machine underflow. The largest exponent allowed on a Sun Ultra2 in the "`exp`" function is somewhere around 15000. Actually, this is not necessarily a sign of a bad formulation. It can be caused by wiggles in your solution due to too coarse of a mesh, too large of a step or bad scaling of the problem. You are dealing with large numbers due to small pore sizes (i.e., high capillary pressures, viz. > 3 kbars) and so your problem scaling is important.

-Use caution in specifying a pressure datum in bounding continuous phases when your porous phase is partially saturated. Recall that the saturation curve is set by the liquid phase or capillary pressure. If you set a high pressure in the continuous phase, then you are setting the saturation in the porous phase by virtue of the `POROUS_PRESSURE` boundary condition at the interface. If that pressure is too high, you might be setting the saturation to less than one at the interface, which is not desirable. The liquid-phase pressure is not necessarily the same in both phases. Ultimately it shouldn't matter, but you need to be consistent with the external gas pressure from the `Porous Gas Constants` card and the initial pressures and other BCs. (PRS, 07/01)

-`UNSATURATED` and `TWO_PHASE` media-type specifications now, unlike in prior versions, use P_LIQ as the dependent variable. Pc is strictly an auxiliary variable computed as p_gas-p_liq. Pc is also always >0 to mean anything. Pc<0 => medium is saturated and all is clipped.

-Big mismatches in velocities of a bounding porous phase and a continuous phase could be a result in different densities. DARCY_CONTINUOUS uses liquid densities now, and gets contributions from the continuous side and the porous side. Make sure they match.

-Partially-saturated flow problems need high mesh densities. Please experiment with finer and coarser meshes to get the feel of how the solution responds and to optimize your mesh. It is now more expedient to use linear elements due to the fact that you are propagating shocks, and quadratic elements cannot handle a discontinuity without undershooting to negative saturations, etc.

-Time stepping is all important. There are three relevant parameters: time-step scheme, initial time step size, and time-step error factor. The rules of thumb that can be established are as follows:

1) If you are using *Porous Mass Lumping*, you must set the *Time Step Parameter* to 0.0, or your performance will suffer. In fact, it is always a good idea in steep penetration front problems to use backward Euler techniques.
2) With mass lumping and first order time integration, you must control your step size with the tolerance setting. Too big of time step early on can propagate to large errors at later times when time stepping. You may need to experiment with the error tolerance on the *Time step error* card. Constantly scrutinize your results for correctness and suspect an error growth here.
3) You must have a significant capacitance term on the first time step. If your capacitance term is small, then the problem is elliptic and will try to satisfy all boundary conditions, and this can mess up your penetration front. You can use *Porous Liquid Compressibility* property to help this for steep front startup.

4) Are you getting stagnant calculations with time-step decreases but not change in iteration history? Problem is that you have lost your capacitance term. Compressibility of the liquid is sometimes a remedy, but also a more accurate predictor. Mass lumping can help too and accomplishes the same thing. Sometimes your initial time step can be too small for a good start. Try increasing it ...

-Permeability and relative permeabilities are real important to these problems. When in doubt, experiment with the permeabilities to just get a solution. These are very important with respect to startup, as the example problems in this memo point out.

-For the time being you cannot solve for species transport other than air and water in the porous phase. ... other restrictions

-"Aborts" inexplicably in the middle of the run can be from several sources, but most likely result from a division by zero. The rel-liq-permeability VAN_GENUCTEN model has a division by zero in cases where SAT is very close to but not exceeding 1. Although provisions have been installed to prevent that, this is a possibility. Also, the frontal solver "front" does not exit gracefully from a def-grad and will result in an abort. Switching over to a umf fixes the problem.

-Another startup issue: Steep discontinuities at boundaries and internally for initial conditions are bad, obviously. If your time step is such that the front cannot penetrate beyond one element in one time step, then with linear elements the capacitance term is ineffective (small) upon reduced time steps. Somehow you have got to get the front beyond one or two elements before things work properly. I find that ramping up the initial boundary conditions helps. Sometimes a large first time step to kick it is good too.

-On startup of a pressurized column of liquid penetrating into a porous substrate, I noticed that at zero-based p_liq, there was no problem elevating the applied pressure on the penetration, but at Atm-based p_liq we couldn't start the problem without severe compressibility. However, compressibility affects the solution, and in fact allows you to push all of your column of liquid into a compressed layer in the substrate, with no Sat from propagation. So beware of poorly defined compressibility of liquid. Also, refinement in the porous layer helped the startup. But the most significant thing for the problem I was solving, don't be surprised if just a little perturbation on externally applied pressure greatly affects the penetration rate. In fact, in one problem simply changing from p_ext of 1.01325e+6 to 1.11325+6 increases the penetration rate 2-fold initially. The steeper curves are harder to handle.

-Note: Restarts for problems that employ a hysteretic saturation function, viz. **`TANH_HYST`** model on the **`Saturation`** card, can ONLY be accomplished using the **`read_exoII_file`** option on the **`Initial Guess`** card. Note that after each run with the hysteresis model a series of element variables are output to the exoII database of the following sort:

```
Element:  SAT_CURVE_TYPE0                SAT_CURVE_TYPE1
           SAT_CURVE_TYPE2                SAT_CURVE_TYPE3
           SAT_SWITCH0                    SAT_SWITCH1
           SAT_SWITCH2                    SAT_SWITCH3
           PC_SWITCH0                     PC_SWITCH1
          PC_SWITCH2                      PC_SWITCH3
```

You can see these with a `list names` command in `blot`. You can contour these element variables if you wish to visualize how the scanning curve hysteresis behavior is proceeding point wise. The **`SAT_CURVE_TYPE`*** variables indicate locally whether the curve type is draining or wetting, the **`SAT_SWITCH`*** variables indicate locally the saturation level at the instant the saturation curve switched between draining and wetting (or vice versa), and the **`PC_SWITCH`** variables indicate the capillary pressure level at the same instant. The variables are stored for each Gaussian integration point, hence the 4 versions of each type (this was the output from a 2D simulation). In 3D there would be 8 versions of each of these variable types, corresponding to 8 Gaussian integration points per element. All this assumes that we are using linear elements. Upon restart with the read_exoII_file option these variables are loaded back into the element-level storage structure.

References

Baer, J. 1972. Dynamics of Fluids in Porous Media, Dover Publications, New York.

Cairncross, R. A., Schunk, P. R. Chen, K. S., Prakash, S., Samuel, J., Hurd, A. J., and Brinker. C. J. 1996. “Drying in deformable partially saturated porous media: Sol-Gel coatings”, Sandia Technical Report, SAND96-2149.

Gartling, D. K., Hickox, C. E., and Givler, R. C. 1996. “Simulations of coupled viscous and porous flow problems”, *Comp. Fluid Dynamics*, **7** 23-48.

Martinez, M. J. 1995. “Formulation and numerical analysis of nonisothermal multiphase flow in porous media”, Sandia Technical Report, SAND94-0379.

Scheidegger, A. E. 1960. The Physics of Flow Through Porous Media, 3rd Ed., University of Toronto Press.

Scherer, G. W. 1992. Recent progress in drying of gels”, *J. of Non-Crystalline Solids*, **147**, 363.

Schunk, P. R., et al.. “GOMA 4.0--A full-Newton finite element program for free and moving boundary problems with coupled fluid/solid momentum, energy, mass, and chemical species transport: User’s Guide”, Sandia Technical Report, SAND2002-2404.

Copy to:

MS 0834	9114	Dayfile
MS 0834	9114	J. E. Johannes
MS 0834	9114	P. R. Schunk
MS 0834	9114	H. K. Moffat
MS 0834	9114	R. A. Roach
MS 0834	9114	D. A. Labreche