Richards Theory Manual

Andy Wilkins
CSIRO

December 12, 2013

Contents

1 Introduction

The Richards' equation[1] describes slow fluid flow through a porous medium. This document describes the theoretical and numerical foundations of the MOOSE implementation.

- Chapter 2 describes the governing equation and associated nonlinear functions.
- Chapter 3 describes the sources and sinks that I have implemented.
- Chapter 4 describes upwinding using the SUPG method.
- Chapter 5 describes the multi-phase Richards' equations that I have implemented. For notational simplicity, the rest of this document focusses on the single-phase case, but this chapter shows the multi-phase version is a straightforward generalisation.
- Chapter 6 briefly discusses tolerancs and convergence criteria.
- Chapter 7 discusses technical issues with the finite-element discretisation.

There are two other accompanying documents: (1) A description of the unit tests and benchmark verifications; (2) Examples of input syntax that users can utilise when building models.

[1] Contrary to the urban legend, "Richards" is unrelated to Richard Martineau: see [1].

2 Governing equations for the single-phase situation

The Richards' equation [1] describes the movement of fluid through the connected pore space of a porous medium. For a single phase, the equation is

$$\phi \frac{\partial}{\partial t}(\rho S) = \nabla_i \left(\frac{\rho \kappa_{ij} \kappa_{\rm rel}}{\mu} (\nabla_j P - \rho g_j) \right) + F \ , \tag{2.1}$$

where the independent variable is P, the fluid pressure. The multi-phase version is very similar and is described in Chapter 5. On the right-hand side, the index summation convection has been used, so i and j are both summed from 1 to 3 in three dimensions (or 1 to 2 in two dimensions). The following notation has been used.

- ϕ is the porosity of the medium (dimensionless). It is a material property (independent of the fluid pressure). The porous medium is assumed to be made up of solid material with a porespace through which the fluid can flow, and the porosity is $V_{\rm porespace}/V_{\rm medium}$. The porosity is assumed time-independent, but may be spatially varying. A typical value for rock is 0.1. To be strict, ϕ is the *connected* porosity, as the medium may have disconnected pores which are irrelevant for the fluid flow, and will be ignored in the following.
- t is time.
- $\mathbf{x}$ is space, and ∇_i is the gradient operator.
- ρ is the fluid density (measured in mass.length^{-3}, usually kg.m^{-3}). ρ is typically a function of pressure, which is discussed in more detail below.
- S is the fluid saturation (dimensionless). It is bounded, $0 \leq S \leq 1$, and is the fraction of pore volume that is filled with the fluid. Hence $\phi\rho S$ appearing on the LHS of Richards' equation is the volume-density of fluid at a point. The term "fully saturated" means $S = 1$, so that the entire pore volume is filled with the fluid. When $S < 1$, the medium is called "partially saturated". The porevolume of a partially saturated medium is filled partially with the fluid, and partially with another fluid which I shall call air in the following (the other fluid need not actually be air, but is convenient to call it so). In the single-phase version, this air is free to move throughout the porevolume — it may even be spontaneously created in a region where there was previously no air — and the assumption of Richards' equation is that it has no affect on the dynamics of the fluid. In the multi-phase version (Chapter 5), the air is given a dynamics of its own. S is a function of fluid pressure and is discussed in more detail below.

- κ_{ij} is the permeability tensor of the medium (measured in length^2, usually m^2). It is a material property (independent of the fluid pressure). It may be spatially varying. It is usually diagonal, since the axes $\mathbf{x}$ are chosen in its principal directions. In groundwater scenarios it is often only transversely isotropic, so that $K_{xx} = K_{yy} \neq K_{zz}$. Typical values for rocks are $K_{xx} = K_{yy} = 10^{-14}\,\mathrm{m}^2$ and $K_{zz} = 10^{-15}\,\mathrm{m}^2$, although rocks with permeability up to six orders of magnitude greater or less than these values are common.

- κ_{rel} is the relative permeability (dimensionless). It is bounded $0 \leq \kappa_{\mathrm{rel}} \leq 1$ as is a function of fluid saturation. It is discussed in more detail below.

- μ is the fluid's dynamic viscosity (measured force.length^{-2}.time, usually Pa.s). While in reality it may be a function of pressure, I have implemented it as constant. A typical value for liquid water is 10^{-3} Pa.s.

- P is the fluid porepressure (measured force.length^{-2}, usually Pa). It is the independent variable.

- $\mathbf{g}$ is the acceleration of gravity (measured in length.time^{-2}, usually m.s^{-2}) as a vector pointing "downwards". It is constant. For instance $\mathbf{g} = (0,\ 0,\ -9.8)\,\mathrm{m.s}^{-2}$.

- F is a source term (measured in mass.length^{-3}.time^{-1}, for instance kg.m^{-3}.s^{-1}).

2.1 Residual saturations, effective saturation, and immobile saturation

Although physically the fluid saturation, S, can never exceed its bounds, $0 \leq S \leq 1$, in practice it is sometimes found that

$$S_{\mathrm{res}} \leq S \leq 1 - S_{\mathrm{air}}\ . \tag{2.2}$$

Here S_{res} is termed the "residual saturation", and S_{air} is the "residual air saturation".

The residual saturation is attained in experiments by applying an "infinite negative fluid pressure" to the porous material to suck out as much fluid as possible. What may happen is that the fluid becomes discontinuous so that small blobs or very thin films of fluid exist in the material, but because they are discontinuous they do not feel the "infinite negative pressure". This gives rise to the nonzero S_{res}. However, further fluid may be extracted through other means such as heating or mechanical stimulation.

The residual air saturation may be nonzero because as water is pumped into an unsaturated medium, air might be trapped so the saturation might not be able to attain $S = 1$.

For $S < S_{\mathrm{res}}$ Richards' equation (2.1) is no longer valid as pressure can no longer be transmitted through the fluid. It is convenient to define the effective saturation

$$S_{\mathrm{eff}} = \frac{S - S_{\mathrm{res}}}{1 - S_{\mathrm{air}} - S_{\mathrm{res}}}\ , \tag{2.3}$$

which has bounds $0 \leq S_{\mathrm{eff}} \leq 1$ in the domain of applicability of Richards' equation.

It is a moot point as to whether $S_{\rm res} = 0 = S_{\rm air}$, because experiments are hard to perform. What is of more practical use is the "immobile saturation", $S_{\rm imm}$, below which the relative permeability is zero, and that is discussed further in Section 2.4. (The physical reasons for the relative permeability going to zero below $S_{\rm imm}$ are probably those given above: the continuous film of fluid breaking down in the presence of large negative porepressures.)

2.2 Density

The MOOSE implementation allows users to specify arbitrary relationships between density and porepressure. One common case for liquids is that the fluid bulk modulus is constant, which means that

$$\rho = \rho_0 e^{P/K} \ . \tag{2.4}$$

In this expression

- ρ_0 is a constant reference density (measured in mass.length^{-3}). A typical value for water is 10^3 kg.m^{-3}.
- P is the fluid porepressure.
- K is the fluid bulk modulus. A typical value for water is 2 GPa.

2.3 Capillary suction

Because of the different wetability of the fluid and air, there will be capillary effects in the medium. In an unsaturated medium, this allows definition of the capillary pressure

$$P_{\rm c} = P_{\rm air} - P \ , \tag{2.5}$$

in terms of the air pressure and the fluid porepressure. For the single-phase case it is convenient to define

$$P_{\rm air} = 0 \ , \tag{2.6}$$

so that the porepressure is referenced to this constant air pressure. The situation is more complicated in the multi-phase case as discussed in Chapter 5.

For an unsaturated wetting fluid, $P_{\rm c} > 0$, and Laplace showed that $P_{\rm c} \propto 1/r$, where r is the radius of the capillary. This $P_{\rm c}$ is unbounded as $r \to 0$, but clearly there must be some cutoff where the fluid undergoes some physical change (like boiling), as porepressures can never actually go below a vacuum. Alternately, it may be that for such small scales ($r \to 0$), the concept of pressure becomes meaningless. This is ignored in many parts of the literature. Indeed, as we shall see below, the relative permeability tends strongly to zero at $P_{\rm c} \to \infty$, so it is difficult to envisage a real-life scenario where porepressures are less than zero (van Genuchten writes the region $P_{\rm c} \to \infty$ is "fairly unimportant for most practical field problems").

For porous materials, the capillary pressure is usually thought of as a function of saturation. Physically I suppose this is because of the distribution of pore-throat sizes: many pore throats

can support a relatively small P_c, while only the small pore throats can contribute towards a large P_c. Hence, for large P_c, the fluid is only occupying a small fraction of pore space.

Various relationships between P_c and $S_{\rm eff}$ have been proposed and used, and the MOOSE implemtnation allows users to specify any functional relationship. The most common is due to van Genuchten [2]:

$$S_{\rm eff} = \left(1 + (\alpha P_c)^{\frac{1}{1-m}}\right)^{-m} \quad \text{for } 0 < m < 1 \, . \tag{2.7}$$

This has a corresponding relative permeability function, detailed in Sec 2.4. Sometimes the van Genuchten functions are defined in terms of a parameter $n = 1/(1-m) > 1$.

In van Genuchten's paper, he finds good fits with experimental data for various soils and rock when the parameter m ranges between about 0.5 and 0.9 (meaning $2 < n < 10$, roughly), and α is beween $4 \times 10^{-5}\,{\rm Pa}^{-1}$ and $2 \times 10^{-4}\,{\rm Pa}^{-1}$. Figure 2.1 shows the shape of the van Genuchten suction, P_c, as a function of $S_{\rm eff}$.

Numerically there are three important features of Eqn (2.7).

1. $S_{\rm eff}$ is a monotonically decreasing function of P_c, which is necessary for a unique solution.

2. $P_c \to \infty$ as $S_{\rm eff} \to 0$. As mentioned above, this is not justifiable physically, but numerically it is extremely advantageous over $P_c \to P_c^0 < \infty$, as this latter version often causes algorithms to "get stuck" around $S_{\rm eff} = 0$. As also mentioned above, because of the low relative permeability around $S_{\rm eff}$, physically realistic problems rarely explore the $S_{\rm eff} \sim 0$ region.

3. $S_{\rm eff} \to 1$ and ${\rm d}S_{\rm eff}/{\rm d}P_c \to 0^+$ as $P_c \to 0$, for all m. This ensures that there is continuity in the porepressure, P, and the derivative ${\rm d}S/{\rm d}P$ around full saturation (remember that by definition $S_{\rm eff} = 1$ for $P_c < 0$). Also ${\rm d}^2S_{\rm eff}/{\rm d}P_c^2 \to 0$ as $P_c \to 0^+$ if $m > 0.5$. This type of term enters the Jacobian[1] in an implicit time-stepping scheme, so it is advantageous that it be continuous around full saturation.

I encourage users to set $m > 0.5$.

2.4 Relative permeability

The relative permeability obeys $0 \le \kappa_{\rm rel} \le 1$. It encodes the fact that in the unsaturated region, continuous fluid films only exist in the small pores, and only moves through these small pores. In contrast, for a fully-saturated region the fluid has access to the entire porespace, so by definition

$$\kappa_{\rm rel}(S_{\rm eff}) = 1 \quad \text{for } P > 0 \, . \tag{2.8}$$

Moreoever, $\kappa_{\rm rel}$ must be a monotonically-increasing function of $S_{\rm eff}$.

Define the "immobile saturation" $S_{\rm imm}$. Then all relative permeabilities are functions of

$$\tilde{S} = \frac{S_{\rm eff} - S_{\rm imm}}{1 - S_{\rm imm}} \, . \tag{2.9}$$

[1] It appears explicitly from the time derivative in Eqn (2.1) if the derivative is written in terms of $\dot{P}$. In fact, that formulation does not give good mass-balance characteristics so it is not used in the MOOSE implementation, but nevertheless it is good to have ${\rm d}^2S_{\rm eff}/{\rm d}P_c^2$ continuous.

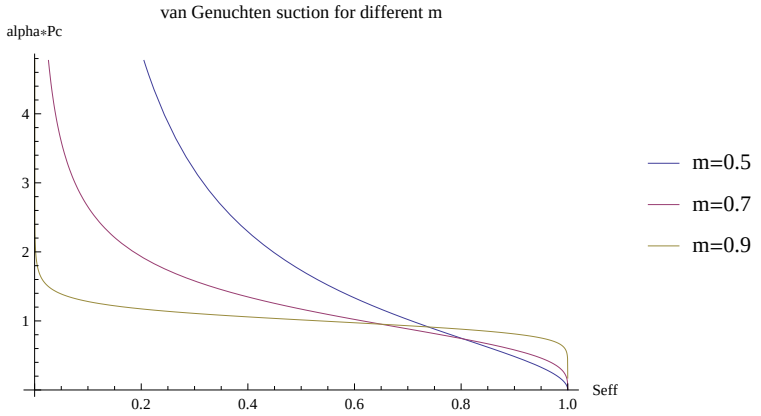

Figure 2.1: αP_c as a function of $S_{\rm eff}$ as given by van Genuchten's expression Eqn (2.7). Three values of m are shown: 0.5, 0.7 and 0.9 (the $m = 0.5$ case has largest P_c around $S_{\rm eff} \sim 0$).

The immobile saturation obeys $0 \leq S_{\rm imm} < 1$, so that $\tilde{S} \leq 1$. The concept of immobile saturation is introduced so that

$$\kappa_{\rm rel} = 0 \quad \text{for} \quad S_{\rm eff} \leq S_{\rm imm} \ , \tag{2.10}$$

(that is, for $\tilde{S} \leq 0$). This holds for all relative permeability functions below. The immobile saturation ensures that in pure fluid flow (ignoring heating or mechanical effects, for instance), $S_{\rm eff} \geq S_{\rm imm}$, and this places a lower bound on physically-obtainable porepressures.

The MOOSE implementation of Richards' equations allows users to specify arbitrary relationships between $\kappa_{\rm rel}$ and $S_{\rm eff}$. Some relationships that have already been coded and tested are given below.

Corresponding to the van Genuchten capillary curve of Eqn (2.7), van Genuchten [2] showed there was a relative permeability function

$$\kappa_{\rm rel}(\tilde{S}) = \sqrt{\tilde{S}}\left(1 - \left(1 - \tilde{S}^{1/m}\right)^m\right)^2 \quad \text{for} \ \ 0 < m < 1 \ . \tag{2.11}$$

Three different plots are shown in Figure 2.2. Evidently, as m inceases the relative permeability also increases.

As $\tilde{S} \to 0$, $k_{\rm rel} \to \tilde{S}^{\frac{1}{2}+\frac{2}{m}} < \tilde{S}^{5/2}$, where the final inequality results from $0 < m < 1$.

As suggested in Figure 2.2, $\mathrm{d}k_{\rm rel}/dP_c \to 0$ as $P_c \to 0^+$ if $m > 0.5$ when using the van Genuchten relative permeability and capillary pressure relationships. This is advantageous numerically when using an implicit time-stepping solution technique, since then both $k_{\rm rel}$ and its pressure-derivative are continuous around full saturation (because of Eqn (2.8)). Once again, I encourage users to set $m > 0.5$.

However, setting $m < 0.5$ can sometimes be useful because: (1) the derivative $\mathrm{d}S_{\rm eff}/\mathrm{d}P$ is smaller (see Fig 2.1) which helps convergence; (2) published literature or experiment specify the value of m. In this case, it is extremely advantageous to modify the relative permeability curve close to $S_{\rm eff} = 1$ to ensure continuity of its pressure derivative in that region. This is standard practice in other codes too. A modified relative permeability function has been coded

into the MOOSE implementation that allows users to specify a cutoff in $\tilde{S}$, `vg_1_cutoff`, above which the relative permeability is approximated by a cubic:

$$\kappa_{\rm rel}(\tilde{S}) = \begin{cases} \text{van Genuchten} & \text{for } \tilde{S} < \texttt{vg\_1\_cutoff} \\ \text{cubic} & \text{otherwise} \end{cases} \tag{2.12}$$

The cubic is chosen so that the result is C^2 continuous, monotonically increasing[2], and is unity at $\tilde{S} = 1$. Figure 2.3 shows two examples.

Another class of relative-permeability curves has also been coded into MOOSE, which have the form

$$\kappa_{\rm rel}(\tilde{S}) = (n+1)\tilde{S}^n - n\tilde{S}^{n+1} \quad \text{for } n \geq 2\ . \tag{2.13}$$

Here n is unrelated to van Genuchten's parameter. These functions have the nice property that $d\kappa_{\rm rel}/dS_{\rm eff} \to 0$ as $S_{\rm eff} \to 1$. Moreover, for small $\tilde{S}$, $\kappa_{\rm rel} \sim \tilde{S}^n$, so that users can easily choose n large enough to ensure that $\kappa_{\rm rel}$ is arbitrarily small for small $\tilde{S}$. However, caution should be applied when choosing large n, as the increasing nonlinearity adversely affects convergence (I suggest choosing $n \leq 5$). Figure 2.4 shows 3 example curves.

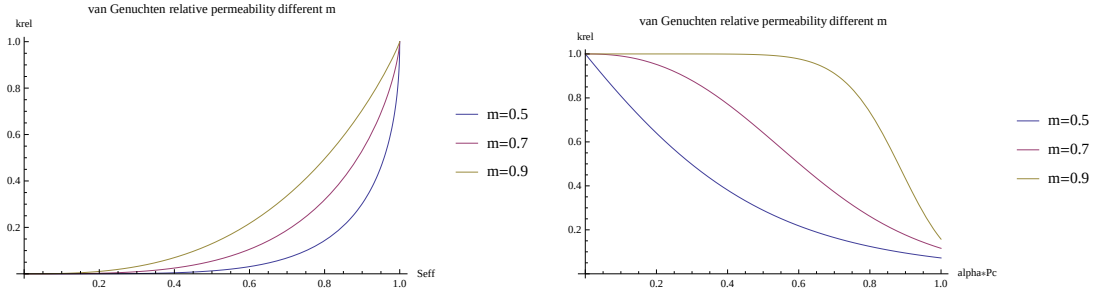

Figure 2.2: Left: $k_{\rm rel}$ as a function of $\tilde{S}$ as given by van Genuchten's expression Eqn (2.11). Right: $k_{\rm rel}$ as a function of P_c utilising Eqn (2.7). In each graph, three values of m are shown: 0.5 (bottom line), 0.7 and 0.9 (topmost line).

[2]This is true, since the van Genuchten $\kappa_{\rm rel}$ and its first and second derivatives are positive at all values of the cutoff (assuming the cutoff is greater than zero), and the first derivative of the cubic is positive at $\tilde{S} = 1$. Because cubics have at most 2 turning points, monotonicity is therefore guaranteed.

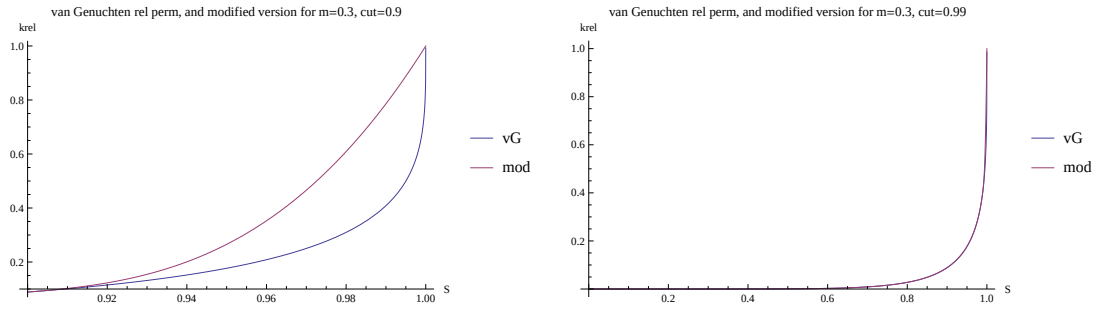

Figure 2.3: Comparison of the modified van Genuchten relative permeability given by Eqn (2.12) with the original. Both cases show $m = 0.3$, and the original van-Genuchten expression is the lower (blue) curve with an infinite slope at $\tilde{S} = 1$. Left: with `vg_1_cut`$= 0.9$, plotting $\tilde{S} \geq 0.9$ to show the difference between the two formulations. Right: with `vg_1_cut`$= 0.99$ the result is almost indistinguishable from the original expression.

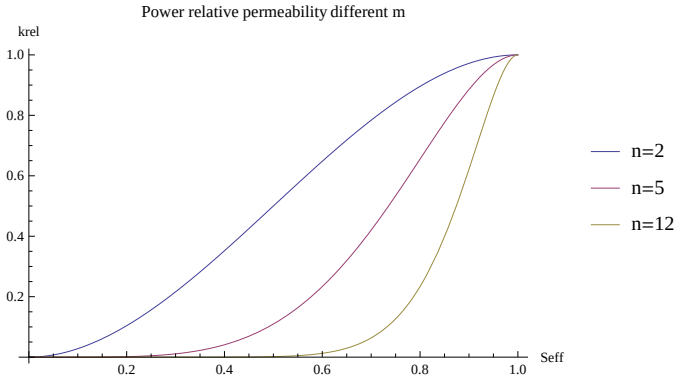

Figure 2.4: k_{rel} as a function of $\tilde{S}$ as given by expression Eqn (2.13). Three values of n are shown: 2 (topmost line), 5 and 12 (bottom line)

3 Sources and Sinks

F in Eqn (2.1) denotes a source term (if positive) or a sink (if negative). There are a number of sources and sinks that are of particular interest to me, which have been coded into MOOSE. This chapter details these. F is a volumetric source (measured in mass.length^{-3}.time^{-1} in 3D), but many of the sources and sinks described below are defined on a $(d-1)$-dimensional submanifold of the model (the 2D boundary of a 3D model, for instance). In these cases, think of F as a surface-source, measured in mass.length^{-2}.time^{-1}. I hope this abuse of notation leads to no confusion.

3.1 Fluxes from boundaries

Fluxes from boundaries can be imposed by fixing the porepressure at the boundary, or by specifying a conductance for the fluid through the boundary. The former is a standard feature in MOOSE. The later is implemented in MOOSE by making F (the surface source) a piecewise-linear function of the porepressure:

$$F = F(P) \ . \tag{3.1}$$

Examples of different linear functions are given in Section 3.2.

3.2 Rainfall recharge

Rainfall recharge through a surface of a model is implemented by making the surface source, F, constant. The standard units of F is kg.m^{-2}.s^{-1}, so a rainfall recharge of 1 cm/year corresponds to

$$F = 3.17 \times 10^{-7}\,\text{kg.m}^{-2}\text{.s}^{-1} \ , \tag{3.2}$$

if the density of the fluid is 1000 kg.m^{-3}.

Usually, however, a seepage condition is imposed at the same time as the rainfall recharge, in order that the porepressure at a model's surface doesn't rise appreciably above atmospheric pressure. (Recall Eqn (2.6) that references porepressure to zero air pressure, so that P_{atm} must also be referenced similarly: usually $P_{\text{atm}} = P_{\text{air}} = 0$.) Then

$$F = \begin{cases} 3.17 \times 10^{-7} & \text{for} \quad P \le P_{\text{atm}} \\ 3.17 \times 10^{-7} - C(P - P_{\text{atm}}) & \text{for} \quad P > P_{\text{atm}} \end{cases} \tag{3.3}$$

In models, the conductance, C, must be chosen carefully, otherwise the system will become mathematically stiff, and PETSc will find it difficult to invert the stiffness matrix. A suitable choice is

$$C = \frac{\kappa_{zz}\rho}{L\mu} \ , \tag{3.4}$$

where κ_{zz} is the vertical component of the permeability tensor at the model's surface, ρ is the fluid density, μ is its viscosity, and L is a distance variable. In my experience $L = 1\,\text{m}$ is a good choice.

As described in the accompanying Example document, in the MOOSE input file, Eqn (3.3) is implemented by specifying the pressure tuple $(P_{\text{atm}}, P_{\text{big}})$, and the flux tuple $(-3.17E-7, -3.17E-7+C(P_{\text{big}}-P_{\text{atm}}))$, for porepressure P_{big} that is a little bigger than the modeller expects to see in the model.

3.3 Evapotranspiration

Evapotranspiration from a surface of the model is modelled using a half-Gaussian sink for the surface-source F:

$$F = \begin{cases} -E_{\text{max}} \exp\left(-\frac{1}{2}\left(\frac{P-P_{\text{o}}}{\sigma}\right)^2\right) & \text{for} \quad P < P_{\text{o}} \\ -E_{\text{max}} & \text{for} \quad P \geq P_{\text{o}} \end{cases} \tag{3.5}$$

Here E_{max} is the maximum value of evapotranspiration, and in standard units is measured in $\text{kg.m}^{-2}.\text{s}^{-1}$. Eqn (3.5) also contains P_{o}, which is the centre of the Gaussian (usually $P_{\text{o}} = P_{\text{air}} = 0$). The final parameter is σ, which parameterises the plant root depth. When $P = P_{\text{o}} - \sigma$, the evapotranspiration will be reduced to 37% of its maximum value. Some example curves are shown in Figure 3.1

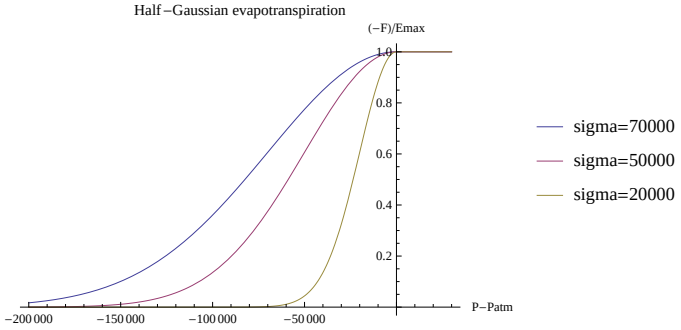

Figure 3.1: Evapotranspiration $-F/E_{\text{max}}$ as a function of porepressure $P - P_{\text{o}}$ for different σ parameters. Top (blue) shows $\sigma = 7 \times 10^4$, middle (red) $\sigma = 5 \times 10^4$, and bottom (greend) $\sigma = 2 \times 10^4$.

A typical recorded value for maximum evapotranspiration would be 4 mm/day, which corresponds to $E_{\text{max}} = 4.63 \times 10^{-5}\,\text{kg.m}^{-2}.\text{s}^{-1}$, assuming the fluid density is $1000\,\text{kg.m}^{-3}$. Usually P_{o} is chosen to be atmospheric pressure $P_{\text{o}} = P_{\text{atm}}$, and usually $P_{\text{atm}} = P_{\text{air}} = 0$. Typical tree roots reach to about 5 m, so a standard value for σ is $\sigma = 5 \times 10^4$ Pa, assuming the density of fluid is $1000\,\text{kg.m}^{-3}$ and gravity is $10\,\text{m.s}^{-2}$.

If the half-Gaussian sink is unsuitable, a piecewise-linear approximation to evapotranspiration could be used instead, similar to that described in Section 3.2

3.4 Streams

Streams are implemented in a similar way to the fluxes from boundaries, except that the fluxes are now only applied at certain discrete points, rather than over an entire boundary, and F is a volume source:

$$F = \sum_i f(P_i)\delta(x - x_i) \ . \tag{3.6}$$

Here δ is the Dirac delta function, and x_i are the spatial points at which the stream lies (with corresponding porepressure P_i). The $\sum_i \delta(x - x_i)$ approximates a polyline source using a number of point sources. F is a volume source, and in standard units it is measured in kg.m^{-3}.s^{-1}, so the function f is measured in kg.s^{-1}.

It is advantageous to choose the set of x_i so that it is sufficiently dense so it well represents the polyline source that describes a river. However, it is pointless to make the density much finer than the underlying finite element mesh (or the fully-refined mesh, if mesh adaptivity is used). The set of points x_i need not correspond with the nodal positions of the mesh.

The function $f(P)$ is a piecewise-linear function, similar to that described in Section 3.2. The following give some examples. Each example contains a riverbed conductance, C, measured in kg.Pa^{-1}.s^{-1} which is

$$C = \frac{\kappa_{zz}\rho}{L\mu} L_{\rm seg} W_{\rm seg} \ . \tag{3.7}$$

As in Eqn (3.4), κ_{zz} is the vertical component of the permeability tensor at the model's surface, ρ is the fluid density, μ is its viscosity, and L is a distance variable. In my experience $L = 1\,$m is a good choice (but in real models this is actually a calibration parameter to match measured baseflow). The other parameters are $L_{\rm seg}$ and $W_{\rm seg}$ which are, respectively, the length and width of the segment of river that the point x_i is representing. The current implementation in MOOSE has f being the same for all points along a river, so the $L_{\rm seg}$ and $W_{\rm seg}$ parameters must be the same for all x_i along a river (otherwise, the river should be broken into a number of individual reaches).

- A perennial stream, where fluid can seep from the porespace to the stream, and viceversa, is modelled with

  $$F(P) = C(P - P_{\rm atm}) \tag{3.8}$$

  where C is the riverbed conductance. As described in the Examples document, this is implemented in the MOOSE input file using the pressure tuple $(-P_{\rm big}, P_{\rm big})$ and the flux tuple $(C(-P_{\rm big} - P_{\rm atm}), C(P_{\rm big} - P_{\rm atm}))$.

- An ephemeral stream, where fluid can only seep from the porespace to the stream, but not viceversa, is modelled with

  $$F(P) = \begin{cases} 0 & \text{for} \quad P \le P_{\rm atm} \\ C(P - P_{\rm atm}) & \text{for} \quad P > P_{\rm atm} \end{cases} \tag{3.9}$$

  where C is the riverbed conductance. As described in the Examples document, this is implemented in the MOOSE input file using the pressure tuple $(P_{\rm atm}, P_{\rm big})$, and the flux tuple $(0, C(P_{\rm big} - P_{\rm atm}))$.

- A rate-limited ephemeral stream, where fluid can only seep from the porespace to the stream, but with an upperbound on this rate, is modelled with

$$F(P) = \begin{cases} 0 & \text{for } P \le P_{\rm atm} \\ C(P - P_{\rm atm}) & \text{for } P_{\rm atm} < P < P_{\rm cutoff} \\ C(P_{\rm cutoff} - P_{\rm atm}) & \text{for } P \ge P_{\rm cutoff} \end{cases} \tag{3.10}$$

  where C is the riverbed conductancel. As described in the example document, this is implemented in the MOOSE input file using the pressure tuple $(P_{\rm atm}, P_{\rm cutoff})$, and the flux tuple $(0, C(P_{\rm cutoff} - P_{\rm atm}))$.

3.5 Wellbores

Wellbores are implemented in a similar way to Eqn (3.6). Here

$$F = \sum_i f(P_i, x_i)\delta(x - x_i) \ , \tag{3.11}$$

but f is a special function (measured in kg.s^{-1} in standard units) defined in terms of the pressure at a point in the wellbore

$$P_{\rm wellbore}(x_i) = P_{\rm bot} + \gamma \cdot (x_i - x_i^{\rm bot}) \tag{3.12}$$

The form of f is

$$f(P_i, x_i) = \begin{cases} W_i \frac{\kappa_{\rm rel}\rho}{\mu}(P_{\rm wellbore} - P) & \text{for } P < P_{\rm wellbore} \\ W_p \frac{\kappa_{\rm rel}\rho}{\mu}(P_{\rm wellbore} - P) & \text{for } P \ge P_{\rm wellbore} \end{cases} \tag{3.13}$$

There are a number of parameters in these expressions:

- $P_{\rm bot}$ is an input parameter. It is the pressure at the bottom of the wellbore.
- $x_i^{\rm bot}$ is the position of the bottom of the wellbore.
- γ is a weight vector pointing downwards (product of fluid density and gravity). This means that $P_{\rm wellbore}(x_i)$ will be the pressure at point x_i in the wellbore, due to gravitational head. If these gravitational effects are undesirable, the user may simply specify $\gamma = (0,0,0)$.
- $\kappa_{\rm rel}$, ρ, and μ are the fluid relative permeability, density and viscosity at the point x_i.
- W_i and W_p are the injection and production well constants, respectively. They are measured in length^3.

  If $W_i = 0$ then the wellbore is a production well, which is, as far as MOOSE is concerned, a sink that produces fluid only if $P > P_{\rm wellbore}$. If $W_p = 0$ then the wellbore is an injection well, which is a source that injects fluid only if $P < P_{\rm wellbore}$.

3.6 Excavations

Excavations are not a flux-type boundary condition. Instead, they are defined by imposing Dirichlet boundary conditions,

$$P = P_{\text{excav}} \ , \tag{3.14}$$

on a time-dependent $(d-1)$-dimensional subspace of the domain. This subspace is the excavation boundary. For instance, it is typically the surface of a rectangular prism, that grows as the excavation proceeds. The parameter P_{excav} is the excavation pressure.

As described in the accompanying Examples document, the time-dependant excavation boundary is defined easily and succinctly through a time-dependent level-set function, that is positive on parts of the domain where excavation has occurred, and negative on the other region.

4 Upwinding

It is well-known that numerical implementations of Eqn (2.1) must use upwinding so that the propagation of fronts be accurately simulated (eg [3, 4, 5]). This chapter details the Streamline-Upwind-Petrov-Galerkin (SUPG) upwinding I have implemented, which is based on [6, 8].

First consider the fairly general case of one scalar variable whose evolution within a region Ω is governed by the transport continuity equation with a source f

$$\frac{\partial M}{\partial t} + \nabla \cdot u = f \ . \tag{4.1}$$

In this equation:

- M is a monotonically increasing function of the scalar variable. The monotonicity is important — without it there probably won't be a unique solution. The "increasing" requirement is just so that signs can be fixed. Physically M could be thought of as the mass density of the scalar variable.
- u is a vector field. It is a function of scalar variable and its spatial derivatives. Special cases are: the advection equation when u is linear in the variable; the diffusion equation for u linear in the gradient of the variable. Observe that the vector field u points in the direction of information propagation.

In addition, on the boundary $\Gamma \subseteq \partial\Omega$,

$$u_n = h \quad \text{on } \Gamma \ , \tag{4.2}$$

where u_n is the outward normal component of u to Γ.

Before considering any upwinding arguments, I would like to present the Petrov-Galerkin method used. The Petrov-Galerkin finite element formulation uses test functions of the form

$$\tilde{\psi} = \psi + p \ . \tag{4.3}$$

The functions ψ are the usual Galerkin functions (linear Lagrange, for instance), while the p functions are smooth on the element interiors, but are discontinuous between elements. In the SUPG formulation, the weak form of Eqn (4.1) is

$$\int_\Omega \psi(\dot{M} - f) - \int_\Omega u \cdot \nabla\psi + \sum_e \int_{\Omega_e} p \left(\frac{\partial M}{\partial t} + \nabla \cdot u - f \right) = -\int_\Gamma wh \ . \tag{4.4}$$

Here e are the elements, which have element interiors Ω_e. Note that the petrov function p is only integrated within elements, and it is continuous there: it is not integrated over the element boundaries. Integrating by parts yields

$$\sum_e \int_{\Omega_e} \tilde{\psi} \left(\frac{\partial M}{\partial t} + \nabla \cdot u - f \right) = \int_\Gamma w(u_n - h) + \int_{\Gamma_{\text{int}}} w[u_n] \ . \tag{4.5}$$

Thus

- Eqn (4.1) holds within the element interiors
- the boundary condition on Γ is satisfied
- the final term is the integral over the internal element boundaries (except those that coincide with Γ), and $[u_n]$ is the jump of u_n as one passes from one element its neighbour. The last term then ensures continuity: $[u_n] = 0$.

So far, no mention of upwinding has been made.

The upwinding comes about through the choice of p. The canonical example for $\tilde{\psi}$ is depicted in Figure 4.1. Evidently $\tilde{\psi}$ is discontinuous at the element boundary. Notice that $\tilde{\psi}$ is larger upstream of the node A than it is downstream of the node A. This is how the SUPG method implements upstream weighting.

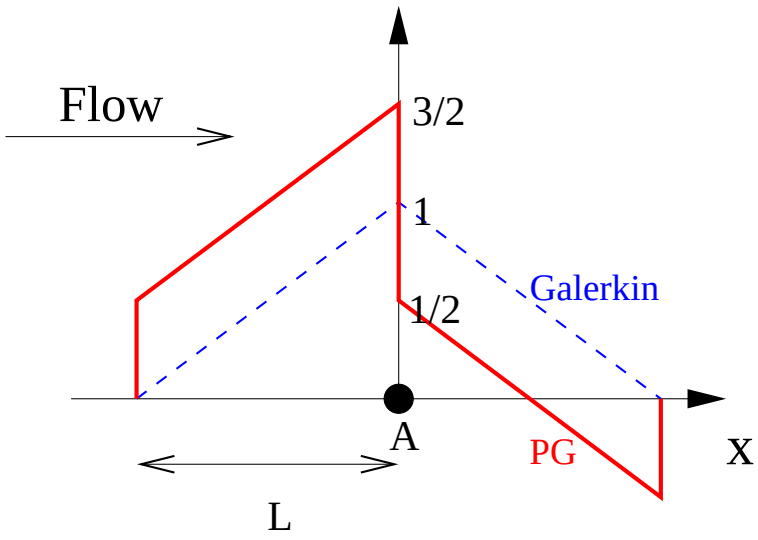

Figure 4.1: Test functions for node A in 1D. The blue dashed line shows the usual linear Lagrange (Galerkin) test function. The red line shows a typical Petrov-Galerkin test function built upon this for the given flow direction. This particular example is the one given by Eqn (4.11) and Eqn (4.17). The more complicated case of Eqn (4.18) is just a linear combination of the two test functions: as aL increases relative to k it moves closer to the red line.

4.1 SUPG and the advection equation in one dimension

The advection equation in arbitrary dimensions is

$$\frac{\partial u}{\partial t} + a \cdot \nabla u = 0 \,, \tag{4.6}$$

where a is a constant.

Consider the 1D version and denote the coordinate by x. Eqn (4.6) describes right-moving advection if $a > 0$, and left-moving advection if $a < 0$. In a numerical discretisation we therefore want to "upwind" as follows:

$$a\frac{\partial u}{\partial x} = \begin{cases} a\frac{u(x)-u(x-\Delta x)}{\Delta x} & \text{for } a > 0 \\ a\frac{u(x+\Delta x)-u(x)}{\Delta x} & \text{for } a < 0 \end{cases} . \tag{4.7}$$

Then $\dot{u}(x)$ depends on what is happening "upstream" of the point x.

In [6] it is explained comprehensively that to implement this upwinding, an extra diffusive term can be added to Eqn (4.6) so that it reads

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - \tilde{k}\frac{\partial^2 u}{\partial x^2} = 0 , \tag{4.8}$$

for some $\tilde{k}$. This is easy to see in this example. If we use standard central-differences for the $\partial u/\partial x$ term, and the standard second-order approximation to the second-derivative term, and $\tilde{k} = |a|\Delta x/2$, we obtain

$$a\frac{\partial u}{\partial x} - \tilde{k}\frac{\partial^2 u}{\partial x^2} = a\frac{u(x+\Delta x)-u(x-\Delta x)}{2\Delta x} - |a|\frac{u(x+\Delta x)-2u(x)+u(x-\Delta x)}{2\Delta x} , \tag{4.9}$$

which is exactly Eqn (4.7).

The extra diffusive term was originally called "artificial diffusion". However, the more modern thinking is that the central difference approximation is actually artificially *under diffuse*, and the addition of $-\tilde{k}\partial^2 u/\partial x^2$ corrects this.

The weak form of the spatial derivative terms is

$$\int \psi \left(a\frac{\partial u}{\partial x} - \frac{|a|\Delta x}{2}\frac{\partial^2 u}{\partial x^2} \right) = \int \left(\psi + \frac{\Delta x}{2}\frac{|a|}{a}\frac{\partial \psi}{\partial x} \right) a\frac{\partial u}{\partial x} + \dots , \tag{4.10}$$

where ψ is a test function, and the "$\dots$" are the boundary term from integrating by parts on the second-derivative term. This manipulation is in fact very important. The additional diffusive term has been manipulated into the test function. Emphasising this, we can define

$$\tilde{\psi} = \psi + \frac{\Delta x}{2}\frac{|a|}{a}\frac{\partial \psi}{\partial x} . \tag{4.11}$$

Then, the weak form of the spatial derivative is identical to the non-upwinded weak form, except that the test function ψ has been replaced by $\tilde{\psi}$. Notice the Petrov part involves $\partial\psi/\partial x$ which is not necessarily continuous. This function $\tilde{\psi}$ is shown explicitly in Figure 4.1 (with $\Delta x = L$).

In the very earliest papers on the subject, this was the end of the story. However, it was soon realised that in order to remove difficulties with the time-derivative term, it too should be integrated against $\tilde{\psi}$. The weak form of the SUPG-stabilised advection equation is therefore

$$\int \tilde{\psi} \left(\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} \right) = 0 . \tag{4.12}$$

A special case of Eqn (4.5) has thus been derived.

4.2 Streamlining: The advection equation in higher dimensions

It is pretty easy to dream up higher-dimensonal generalisations of the 1D $\bar{\psi}$ of Eqn (4.11) to use in Eqn (4.5), however, in higher dimensions "streamlining" is needed. This is mostly easily seen by recalling Eqn (4.8): SUPG is adding an extra diffusion term to Eqn (4.6), which will then read

$$\frac{\partial u}{\partial t} + a \cdot \nabla u - \tilde{k}_{ij} \nabla_i \nabla_j u = 0 \; . \tag{4.13}$$

Note that the velocity a is a vector (it is constant).

If $a_{\mathrm{x}} = 0$, for instance, then $\tilde{k}_{xj}$ should be zero, otherwise so-called cross-wind diffusion will occur where the solution diffuses in the "x" direction only because the extra diffusion term has been added, and not for any physical reason. Therefore, the standard is to take

$$\tilde{k}_{ij} = \tilde{k} \frac{a_i a_j}{|a|^2} \; . \tag{4.14}$$

This is called "streamlining", and the problem has simplified to finding a suitable $\tilde{k}$.

Forming the weak version of Eqn (4.13) and integrating the second-derivative term by parts yields the SUPG version:

$$\int \tilde{\psi} \left(\frac{\partial u}{\partial t} + a \cdot \nabla u \right) = 0 \; . \tag{4.15}$$

The SUPG test function is

$$\tilde{\psi} = \psi + \tilde{k} \frac{a \cdot \nabla \psi}{|a|^2} = \psi + \tau a \cdot \nabla \psi \; , \tag{4.16}$$

in terms of the basis function ψ. The parameter τ has been introduced: $\tau = \tilde{k}/|a|^2$, which is more commonly used in the literature than $\tilde{k}$. The time-derivative is also integrated against $\tilde{\psi}$, as discussed above. Eqn (4.16) is the streamline-upwind-Petrov-Galerkin test function.

In a finite-element discretisation using rectangular elements, the parameter $\tilde{k}$ could be taken as, for instance,

$$\tilde{k} = \sum_{i=1}^{d} L_i |a_i| / 2 \quad \text{or} \quad \tau = \sum_{i=1}^{d} L_i / (2|a_i|) \; , \tag{4.17}$$

where d is the number of dimensions, and L_i is the element length in the i^{th} direction. The 1D version of this is shown in Figure 4.1.

Ohter choices of $\tilde{k}$ have been studied in the literature, and one popular version is

$$\tilde{k} = \tau |a|^2 = \sum_{m=1}^{d} \frac{L^m a^m}{2} \left(\coth \alpha_m - \frac{1}{\alpha_m} \right) \; , \tag{4.18}$$

in which

$$\alpha_m = \frac{a^m L^m}{2k} \quad \text{(no sum on } m\text{)} \; , \tag{4.19}$$

where k is a physically-relevant diffusion parameter. For rectangular or rectangular-prism elements elements, L^m is the length of the element in the m direction, and a^m is the component of a in the m direction, while for other element shapes, L^m is defined using more elaborate expressions [6].

The associated $\tilde{\psi}$ is a linear combination of the Galerkin and the Petrov-Galerkin functions shown in Figure 4.1. The function $\coth\alpha - 1/\alpha$ is sketched in Figure 4.2, where from which it is clear that $\alpha(\coth\alpha - 1/\alpha) \sim |\alpha|$ for large α. So, as α_m grows with respect to k, the PG function because closer to that of Eqn (4.17).

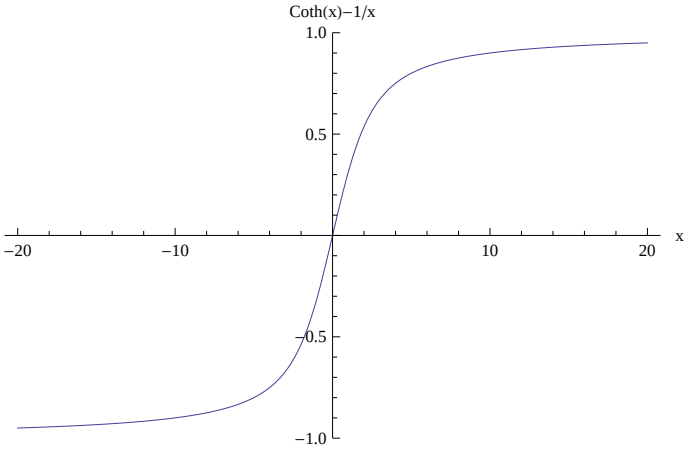

Figure 4.2: The function $\coth x - x^{-1}$

4.3 SUPG and the continuity equation

Now we are armed with some examples, it is clear that the PG test function in Eqn (4.5) can have the form

$$\tilde{\psi} = \psi + \tilde{k}\frac{u\cdot\nabla\psi}{|u|^2} = \psi + \tau u\cdot\nabla\psi\ . \tag{4.20}$$

This is streamlined, as u points in the direction of information propagation.

It is worth mentioning that other forms for $\tilde{\psi}$ have been explored and shown to be better than Eqn (4.20) in various situations. For instance in [7], an extra term is added to $\tilde{\psi}$ that is in the direction of ∇u. This better models discontinuities in u if ∇u is not parallel to a. Probably this is irrelevant in the Richards' situation, since, effectively ∇u *is* a, as discussed below. In [8] the case for more than one field in multiple dimensions is studied.

4.4 SUPG upwinding of Richards' equation

Richards' equation (2.1) can be written in the form of the continuity equation with the substitution

$$u_i = -\frac{\rho\kappa_{\rm rel}}{\mu}\kappa_{ij}(\nabla_j P - \rho g_j) = \frac{\rho\kappa_{\rm rel}}{\mu}v_i\ . \tag{4.21}$$

In the second equality, the vector $v_i = -\kappa_{ij}(\nabla_j P - \rho g_j)$ has been introduced. This points in the direction of information propagation.

In this case the SUPG test function Eqn (4.20) takes the form

$$\tilde{\psi} = \psi + \tau v_i \nabla_i \psi \ . \tag{4.22}$$

The parameter τ needs to scale like $1/|v|$, and in 1D needs to reduce to something like $L/(2|v|)$. Some alternatives are written in Eqns (4.17) and (4.18). I do not know the "best choice" — the one that yields the least error in the problem studied — at this time, as I think this can only be found using experimentation. According to [6], most choices give quite similar results. Therefore, I choose to use the expressions given in the Appendix of [7].

Consider a single element. Denote its isoparametric coordinates by $\xi^m = \xi^m(x)$ ($m = 1, \ldots$, and x are the spatial coordinates). Then define the quantities

$$b^m = v_i \nabla_i \xi^m \ . \tag{4.23}$$

Physically, these correspond to the projections of v along the isoparametric direction m, except that since $\sum_i |\nabla_i \xi^m|^2 \neq 1$ (in general), there is some information regarding the element size in the projection.

The b^m quantities scale inversely with element size. For instance, if $\xi^1 = 2x/L$, for an element which sits between $-L/2 \leq x \leq L/2$ (corresponding to $-1 \leq \xi^1 \leq 1$), then $b^1 = 2v_x/L$. More generally,

$$|b| = (\sum_m |b^m|^2)^{1/2} \ , \tag{4.24}$$

is approximately $2|v|/L$, where L is a measure of the length of the element in the v direction.[1]

Define $|v| = (\sum_i |v_i|^2)^{1/2}$. An element mesh parameter is defined by

$$\tfrac{1}{2} h = |v|/|b| \ , \tag{4.25}$$

which is a measure of the length of the element in the v direction, but note that it depends on x.

Next define a Peclet number similar to [7]:

$$\alpha = h|v|/\text{diffusivity} = \frac{h|v|}{2P_{\text{SUPG}} \text{Tr}\kappa} \ , \tag{4.26}$$

where P_{SUPG} is a user-defined pressure, and $\text{Tr}\kappa = \sum_{i=1}^d \kappa_{ii}$. Then $\tilde{k}$ is defined through

$$\tau = |b|^{-1} \left(\coth \alpha - \frac{1}{\alpha} \right) \ . \tag{4.27}$$

In practice, for problems that involve tracking fronts, SUPG is needed. Then it is advantageous to set P_{SUPG} less than the expected range in pressures in the unsaturated zone. For problems where fronts are less important, SUPG is not usually necessary, so the parameter should be set larger, otherwise it will cause problems in convergence when simulations start to reach steady state as the upwinding direction changes every timestep.

[1] In $|b|$, [7] use the p-norm, and go on to explore the ∞-norm and the 2-norm. Both give similar results, so I just use $p = 2$.

5 Multi-phase Richards' equations

The MOOSE implentation of Richards' equation is not only valid for the single-phase case as described elsewhere in this document, but for the multi-phase case too. Denote the phase number by γ. The Richards' equations for multi-phase flow are

$$\phi \frac{\partial}{\partial t}(\rho^\gamma S^\gamma) = \nabla_i \left(\frac{\rho^\gamma \kappa_{ij} \kappa^\gamma_{\rm rel}}{\mu^\gamma} (\nabla_j P^\gamma - \rho^\gamma g_j) \right) + F^\gamma \, , \tag{5.1}$$

where the independent variables are porepressures of the phases, P^γ. The notation is the same as in Eqn (2.1).

The MOOSE implementation allows users to specify arbitrary functions for the phase densities and relative permeabilities, but assumes the following.

- The density for phase γ is only a function of its pressure: $\rho^\gamma = \rho^\gamma(P^\gamma)$.
- The relative permeability for phase γ is only a function of its effective saturation: $\kappa^\gamma_{\rm rel} = \kappa^\gamma_{\rm rel}(S^\gamma_{\rm eff})$.

The immobile and relative saturations may be set independently for each phase.

The effective saturations are more complicated, as they are functions of all the pressure variables:

$$S^\gamma_{\rm eff} = S^\gamma_{\rm eff}(P_1, \, P_2, \ldots) \, . \tag{5.2}$$

It is therefore the saturations that couple the separate phases together. The MOOSE implementation allows users to specify arbitrary functions for the effective saturations, but these must obey the constraint

$$\sum_\gamma S^\gamma = 1 \, . \tag{5.3}$$

For two phases — water and gas — a common practise is to define

$$P_c = P_{\rm gas} - P_{\rm water} \, , \tag{5.4}$$

and to use the van Genuchten expression for the water effective saturation:

$$S^{\rm water}_{\rm eff} = \left(1 + (\alpha P_c)^{\frac{1}{1-m}} \right)^{-m} \quad \text{for } \; 0 < m < 1 \, . \tag{5.5}$$

Then the gas effective saturation is just $S^{\rm gas}_{\rm eff} = 1 - S^{\rm water}_{\rm eff}$ (assuming residual saturations are zero). This common case has been coded and tested in the MOOSE implementation.

6 Tolerances and Convergence

It is sometimes difficult to set appropriate tolerances on the nonlinear solver when running real models. You should never expect the nonlinear residual to go to exactly zero, because there will always be problems associated with precision loss. The absolute tolerance on PETSc's nonlinear solver (`-snes_atol`) should be set at the maximum of the numbers calculated below, otherwise tolerance will never be achieved (unless it is through testing of the relative sizes of the residuals).

6.1 Minimum residual from spatial-derivative terms

Consider the part of the residual

$$R = \int \psi \nabla_i \left(\frac{\rho \kappa_{ij} \kappa_{\rm rel}}{\mu} (\nabla_j P - \rho_0 g_j) \right) \ . \tag{6.1}$$

When discretised over the mesh this looks like

$$R_{\rm element} \sim L^d \frac{1}{L} \frac{\rho |\kappa|}{\mu} \frac{P_1 - P_0}{L} \ , \tag{6.2}$$

where L is the element size, d is the number of dimensions, and P_1 and P_0 are the values of P at neighbouring quadrature points. The relative permeability has been dropped because $\kappa_{\rm rel} \leq 1$ and I want to place an upper bound on the residual.

The key point is $P_1 - P_0$ is subject to precision loss. If, for example $P_1 = 1\,$MPa, then $|P_1 - P_0| \geq 10^{-15+6} = 10^{-9}\,$Pa, assuming that there are 15 digits of precision, and barring the possibility that the computer has luckily converged upon $P_1 = P_0$ exactly. This means that

$$R_{\rm element} \sim\!> 10^{-P} L^{d-2} \frac{\rho |\kappa|}{\mu} |P| \ , \tag{6.3}$$

where P is the number of digits of precision in the computer code.

Example Suppose the user's model is expected to produce pressures of a maximum of 10 MPa, the density of their fluid is approximately $1000\,{\rm kg.m}^{-3}$, the permeability tensor has components of order $10^{-12}\,{\rm m}^2$, the fluid viscosity is $10^{-3}\,$Pa.s, and their elements are of size $L = 100\,$m. Then for a 3D model with $P = 15$ (double precision) the residual can never be expected to fall below

$$R \sim 10^{-15} 10^2 \frac{10^3 10^{-12}}{10^{-3}} 10^7 = 10^{-12} \ . \tag{6.4}$$

Clearly it is best to over-estimate the parameters in order to give a maximum value for R, otherwise a model may apparently never converge. However, some judicious reasoning might be necessary if the parameters in Eqn (6.3) vary substantially throughout the mesh.

6.2 Minimum residual from spatial-derivative terms with SUPG

Consider the part of the residual

$$R = \int \tilde{\psi} \nabla_i \left(\frac{\rho \kappa_{ij} \kappa_{\rm rel}}{\mu} (\nabla_j P - \rho_0 g_j) \right) \ . \tag{6.5}$$

The standard part of this has been considered in Section 6.1. The SUPG part of $\tilde{\psi}$ is $\tau v_i \nabla \psi$, and $\tau |v| \sim L$, which gives

$$\begin{aligned} R_{\rm element} &\sim L^d L \left(\frac{\psi_1 - \psi_0}{L} \right) \left(\frac{\rho|\kappa|(0) - \rho|\kappa|(1)}{\mu L} \right) \left(\frac{P_1 - P_0}{L} \right) \\ &\sim 10^{-2P} L^{d-2} \frac{\rho|\kappa|}{\mu} |P| \ . \end{aligned} \tag{6.6}$$

Evidently this is smaller than Eqn (6.3), so that SUPG does not need to considered for the spatial-derivative terms.

6.3 Minimum residual from temporal derivative terms

Consider the part of the residual

$$R = \int \psi \phi \frac{\partial}{\partial t} (\rho S) \ . \tag{6.7}$$

When discretised over the mesh this looks like

$$R_{\rm element} \sim L^d \frac{\rho S_{{\rm d}t} - \rho S_0}{{\rm d}t} \ . \tag{6.8}$$

The key point is that the time derivative of ρS is subject to precision loss. For instance, if $\rho S = 10^3\,{\rm kg.m^{-3}}$, then $(\rho S_{{\rm d}t} - \rho S_0) \geq 10^{-15+3} = 10^{-12}\,{\rm kg.m^{-3}}$. This means that

$$R_{\rm element} \sim> 10^{-P} L^d |\rho S| / {\rm d}t \ . \tag{6.9}$$

Notice that this depends on dt. Hence when nonlinear-solving models with very small timesteps, the residual may not reduce substantially from its initial value!

Example Suppose the user's model has $S \sim 1$, $\rho \sim 1000\,{\rm kg.m^{-3}}$, elements of size $L \sim 100\,{\rm m}$. Then for a 3D model with $P = 15$ (double precision), the residual can never be expected to fall below

$$R \sim 10^{-15} 10^6 \frac{10^3}{{\rm d}t} = 10^{-7} / {\rm d}t \ . \tag{6.10}$$

6.4 Minimum residual from temporal derivative terms with SUPG

Consider the part of the residual

$$R = \int \tilde{\psi} \phi \frac{\partial}{\partial t} (\rho S) \ . \tag{6.11}$$

The standard part of this has been considered in Section 6.3. The SUPG part of $\tilde{\psi}$ is $\tau v_i \nabla_i \psi$ which is approximately 1. Hence the addition of SUPG does not affect any considerations of minimum residual.

7 Discretisation

7.1 General comments

All nonlinear functions (ρ, S and $\kappa_{\rm rel}$) are evaluated at each quadrature point as functions of the porepressure at that quadrature point. For the flux term (the right-hand side of Eqn (2.1)), this is standard, but in most other finite-element solvers of Richards' equation, the time-derivative term is lumped to the nodes. That is, the part of the residual at a node that comes from $\partial(\phi\rho S)/\partial t$ just depends on the porepressure at that node. It has been shown in many papers that this lumping is advantageous for mass consideration and reduces spurious oscillations of the pressure around sharp fronts [9]. This is explained in Section 7.4. In the current implementation of Richards' equation in MOOSE, this residual depends on the porepressure at all the quadrature points surrounding the node, and I'm not sure how to implement lumping in MOOSE.

7.2 Examples of Richards upwinding

Consider the 1D situation shown in Figure 7.1. This shows 2 elements, each of length L, and the corresponding shape (or test) functions. Define the potential function

$$\phi = P + \rho|g|z \ , \tag{7.1}$$

and the mobility function

$$\lambda = \frac{\rho\kappa_{\rm rel}}{\mu} \ . \tag{7.2}$$

The isoparametric coordinate for the first element is $\xi = \frac{z-L/2}{L/2}$, which yields $b = 2v/L$ (Eqn (4.23). When $P_{\rm SUPG}$ is set small, the upwinding parameter

$$\tau = \left|\frac{L}{2v}\right| \ . \tag{7.3}$$

Here the velocity is just $v = -\kappa\partial_z\phi$.

Consider the flux part of Eqn (4.5):

$$I = \int \tilde{\psi}\partial_z(-\lambda\partial_z\phi) \ , \tag{7.4}$$

integrated against the zeroth test function $\tilde{\psi} = \tilde{\psi}_0$ (this is called $S0$ in Figure 7.1). Then, for Linear Lagrange elements,

$$\begin{aligned} I &= \int \partial_z\psi_0\lambda\kappa\partial_z\phi - \int \frac{L}{2}\frac{v}{|v|}\partial_z\psi_0\partial_z\lambda\kappa\partial_z\phi \\ &= -\kappa\frac{\phi_1-\phi_0}{L}\left(\frac{\lambda_a+\lambda_b}{2} - \frac{L}{2}\frac{v}{|v|}\frac{\lambda'_a+\lambda'_b}{2}\frac{\phi_1-\phi_0}{L}\right) \ . \end{aligned} \tag{7.5}$$

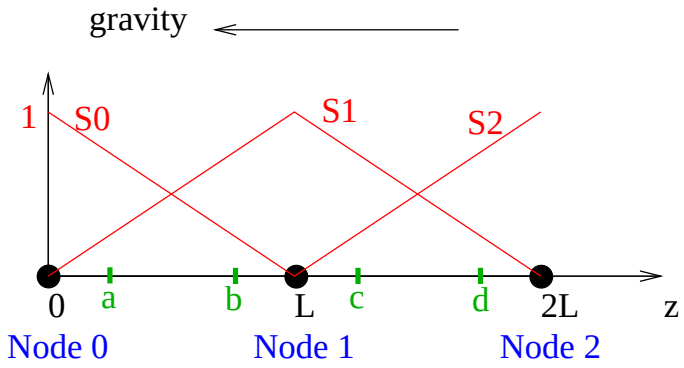

Figure 7.1: Two elements of length L. Linear Lagrange shape/test functions for each node are shown in red. Gravity acts in the direction $-z$. Gauss points are shown in green.

Here I've used $\partial_z \psi_0 = -1/L$, and $\partial_z \phi = (\phi_1 - \phi_0)/L$, where ϕ_0 is the value of ϕ at node 0. I have also evaluated the integrands at the Gauss points a and b as labelled in Figure 7.1. Define

$$\Delta = \frac{\phi_1 - \phi_0}{L} \,, \tag{7.6}$$

and define l to be the distance of Gauss point a from $z = 0$ (this is $L(1 - 1/\sqrt{3})/2$ for a Linear Lagrange element). By performing a Taylor expansion of the nonlinear function $\lambda = \lambda(\phi)$ it is not too difficult to show that

$$\begin{aligned} \tfrac{1}{2}(\lambda_a + \lambda_b) &= \tfrac{1}{2}(\lambda_0 + \lambda_0) + \tfrac{1}{4}\left((L-l)^2 + l^2 - L^2\right)\Delta^2 \lambda_0'' + \tfrac{1}{12}\left((L-l)^3 + l^3 - L^3\right)\Delta^3 \lambda_0''' \quad (7.7) \\ \tfrac{1}{2}(\lambda_a' + \lambda_b')\Delta &= \frac{\lambda_0 + \lambda_1}{L} + \tfrac{1}{12}\left(2L^2 + 3(L-l)^2 + 3l^2\right)\Delta^3 \lambda_0''' \quad (7.8) \end{aligned}$$

The important result is that to second order

$$I = \begin{cases} -\kappa \frac{\phi_1 - \phi_0}{L} \lambda_1 + O(\lambda_0'') & \text{for} \quad v < 0 \\ -\kappa \frac{\phi_1 - \phi_0}{L} \lambda_0 + O(\lambda_0'') & \text{for} \quad v > 0 \end{cases} \tag{7.9}$$

Hence, to second order, upwinding is achieved. Therefore, for linear functions λ, SUPG upwinding is identical to a fully upwinding method (in 1D with Linear Lagrange elements). For nonlinear functions, this is not true.

Now consider the situation where Node 1 is at immobile saturation. The potential will attempt to transfer fluid from Node 1 to Node 0, but fluid transfer shouldn't occur in reality since the mobility $\lambda_1 = 0$. Without SUPG, however, some transfer will take place (assuming Node 0 is not also at immobile saturation), since $\lambda_a > 0$ and $\lambda_b > 0$ in Eqn (7.5). This means that the saturation at Node 1 will reduce below immobile saturation. With SUPG this effect is reduced due to the fluid transfer depending only on $\lambda_1 = 0$ to second order (Eqn (7.9)).

7.3 The time derivative and mass conseration

The time derivative is discretised as

$$\psi\phi\frac{\rho S - \rho_{\rm old}S_{\rm old}}{{\rm d}t} \ , \tag{7.10}$$

instead of the perhaps more common $\psi\phi(\rho S)'\dot{P}$. Eqn (7.10) conserves fluid mass more effectively.

7.4 Lumping the time derivative

Consider the situation in Figure 7.1, and suppose that Node 2 has high potential, and that nodes 0 and 1 are at immobile saturation. Then fluid will flow from node 2 to node 1 (and then to node 0 in the next time step). For simplicity, imagine that $\phi\rho S$ is a linear function of the potential ϕ. Then, up to constants, the discretised Richard's equation reads

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} (\dot{\rho S})_0 \\ (\dot{\rho S})_1 \\ (\dot{\rho S})_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \tag{7.11}$$

The matrix on the LHS comes from performing the numerical integration of $(\dot{\rho S})$ over the two elements. Note that it is not diagonal because the integration over an element depends on the potential at both of its two nodes. The RHS encodes that no fluid is flowing between nodes 0 and 1, but fluid is flowing from node 2 to node 1.

The important point is the solution of these sets of equations is

$$(\dot{\rho S})_0 < 0 \ . \tag{7.12}$$

This means the finite element solution of Richard's equation will be oscillatory around fronts. This is why lumping is necessary [9]. With lumping the matrix in the above equation becomes diagonal, and the solution is $(\dot{\rho S})_0 = 0$.

Bibliography

[1] LA Richards “Capillary conduction of liquids through porous mediums” Physics 1 (1931) pp 318–333

[2] MT van Genuchten “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils” Soil Sci Soc Am J 44 (1980) 892–898.

[3] PS Huyakorn and GF Pinder “A new finite element technique for the solution of two-phase flow through porous media” Advances in Water Resources 1 (1978) 285–298

[4] V Dalen “Simplified finite-element models for reservoir flow problems” SPEJ (Oct 1979) 333–343

[5] R Helmig and R Huber “Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media” Advances in Water Resources 21 (1998) 697–711

[6] AN Brooks and TJR Hughes “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations” Computer Methods in Aplied Mechanics and Engineering 32 (1982) 199–259.

[7] TJR Hughes, M Mallet and A Mizukami “A new finite element formulation for computational fluid dynamics: II. Beyond SUPG” Computer Methods in Applied Mechanics and Engineering 54 (1986) 341–355

[8] TJR Hughes and M Mallet “A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems” Computer Methods in Applied Mechanics and Engineering 58 (1986) 305–328.

[9] MA Celia, ET Bouloutas and RL Zabra “A general mass-conservative numerical solution for the unsaturated flow equation” Water Resources Research 26 (1990) 1483–1496.