Development of Variably Saturated Flow Model (VSFM) in the **A**CME **L**and **M**odel using the approach of solving multi-physics problems in PETSc

Gautam Bisht

August 31, 2015

Contents

List of Figures

List of Tables

1 Overview of Approach

A recent workshop organized by the Office of Advance Scientific Computing Research at the US Department of Energy summarized challenges and opportunities in solving multiphysics problems across a wide range of applications, including climate modeling, subsurface science, and surface-subsurface hydrology (Keyes et al., 2013). There are a large number of third party softwares that can provide robust numerical solution to partial differential equations (eg. MOOSE, PETSc, SUNDIALS, Trilinos).

PETSc's `DMComposite` approach provides a modular framework to solve coupled multi-physics problems. PETSc-SNES example 28 demonstrates the use of `DMComposite` to solve the following system of PDE coupled to an algebraic equation in 1D:

$$-(ku_x)_x = 1 \text{ on (0,1), subject to u(0) = 0, u(1) = 1} \tag{1a}$$

$$e^{k-1} + k = \frac{1}{\frac{1}{(1+u)} + \frac{1}{(1+u_x^2)}} \tag{1b}$$

The finite difference discretization of the above equations lead to the following set of residual equations,

$$F_u = -\frac{1}{\Delta x}\left[k_i\left(\frac{u_{i+1}-u_i}{\Delta x}\right) - k_{i-1}\left(\frac{u_i - u_{i-1}}{\Delta x}\right)\right] - 1 \tag{2a}$$

$$F_k = e^{k_i - 1} + k_i - \frac{1}{\frac{1}{(1+0.5(u_i+u_{i+1}))} + \frac{1}{(1+(u_{i+1}-u_i)^2/(\Delta x)^2)}} \tag{2b}$$

The above mentioned discretized set of equations can be solved by Newton-Raphson method as

$$\begin{bmatrix} \frac{\partial F_u}{\partial u} & \frac{\partial F_u}{\partial k} \\ \frac{\partial F_k}{\partial u} & \frac{\partial F_k}{\partial k} \end{bmatrix} \begin{bmatrix} \Delta u \\ \\ \Delta k \end{bmatrix} = - \begin{bmatrix} F_u \\ \\ F_k \end{bmatrix}. \tag{3}$$

A pseudo Fortran code for computing residual and Jacobian corresponding to the above mentioned equations are as follows:

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
subroutine FormFunction_All()
 ! Assemble residual vector: F = [fu fk]T;
 ! determine number of DMs
 call DMCompositeGetNumberDM(dm, nDM, ierr)
 ! Get sub-vectors for X and F
 call DMCompositeGetAccessArray(dm, X, nDM, PETSC_NULL_INTEGER, X_subvecs, ierr)
 call DMCompositeGetAccessArray(dm, F, nDM, PETSC_NULL_INTEGER, F_subvecs, ierr)
 ! Call respective subroutines to compute residual evaluation
 call FormFunctionLocal_U(X_subvecs(1), F_subvecs(1), ...)! fu
 call FormFunctionLocal_K(X_subvecs(2), F_subvecs(2), ...)! fk
end subroutine  FormFunction_All
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
subroutine FormJacobian_All()
 ! Assemble jacobian matrix: J = [dfu/du dfu/dk; dfk/du dfk/dk]
 ! determine number of DMs
 call DMCompositeGetNumberDM(dm, nDM, ierr)
 ! Get sub-vectors for X and F
 call DMCompositeGetAccessArray(dm, X, nDM, PETSC_NULL_INTEGER, X_subvecs, ierr)
 call DMCompositeGetAccessArray(dm, F, nDM, PETSC_NULL_INTEGER, F_subvecs, ierr)
 ! Get sub-matrices
 allocate(B_submats(nDM,nDM))
 call DMCompositeGetLocalISs(dm, is, ierr)
 do row = 1,nDM
  do col = 1,nDM
   call MatGetLocalSubMatrix(B, is(row), is(col), B_submats(row,col), ierr)
  enddo
 enddo
 ! Assemble diagonal matrices
 call FormJacobianLocal_U(B_submats(1,1),...) ! dfu/du
 call FormJacobianLocal_K(B_submats(2,2),...) ! dfk/dk
 ! Assemble off-diagonal matrices
 call FormJacobianLocal_UK(B_submats(1,2),...)! dfu/dk
 call FormJacobianLocal_KU(B_submats(2,1),...)! dfk/du
end subroutine  FormJacobian_All
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

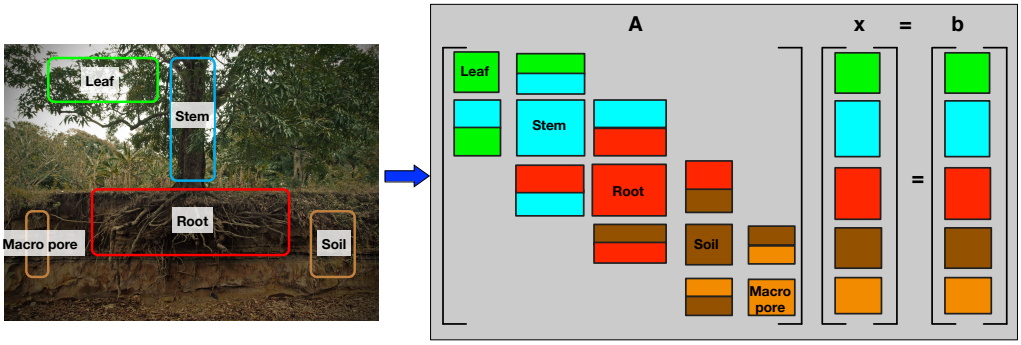

Figure 1: Schematic representation of a soil-root-plant hydraulics model.

For the **A**CME **L**and **M**odel (ALM), the PETSc-based multi-physics framework would be use to solve soil–root–plant hydraulic model, as shown in Figure 1.

2 Richards Equation

The Richards equation implemented in the ALM is similar to `RICHARDS` Mode in PFLOTRAN. For water equation of state, PFLOTRAN uses IFC-67 steam tables, while ALM uses Tanaka et al. (2001). Additionally, PFLOTRAN uses PETSc's DMDA, while ALM uses PETSc's DMComposite.

The following subsections are obtain from PFLOTRAN Developer Guide with permission from PFLOTRAN developers[1].

2.1 Governing Equations

The governing mass conservation equation for Richards equation is given by

$$\frac{\partial}{\partial t}\left(\varphi s \rho\right) + \boldsymbol{\nabla} \cdot \left(\rho \boldsymbol{q}\right) = Q_w, \tag{4}$$

and

$$\boldsymbol{q} = -\frac{k k_r(s)}{\mu} \boldsymbol{\nabla} \left(P - W_w \rho g z\right). \tag{5}$$

Here, φ denotes porosity [-], s saturation [$\mathrm{m^3 m^{-3}}$], ρ water density [kmol $\mathrm{m^{-3}}$], $\boldsymbol{q}$ Darcy velocity [m $\mathrm{s^{-1}}$], k intrinsic permeability [$\mathrm{m^2}$], k_r relative permeability [-], μ viscosity [Pa s], P pressure [Pa], W_w formula weight of water [kg $\mathrm{kmol^{-1}}$], g gravity [m $\mathrm{s^{-2}}$], z the vertical component of the position vector [m], and Q_w is source of water [kmol $\mathrm{m^{-3}}$ $\mathrm{s^{-1}}$]. Density and viscosity of water are non-linear function of pressure.

2.1.1 Capillary Pressure Relations

Capillary pressure is related to saturation by various phenomenological relations, one of which is the van Genuchten (1980) relation

$$s_e = \left[1 + \left(\frac{p_c}{p_c^0}\right)^n\right]^{-m}, \tag{6}$$

where p_c represents the capillary pressure [Pa], and the effective saturation s_e is defined by

$$s_e = \left[\frac{s - s_r}{s_0 - s_r}\right], \tag{7}$$

[1] Peter Lichtner (OFM), Glenn Hammond (SNL), Satish Karra (LANL)

where s_r denotes the residual saturation, and s_0 denotes the maximum saturation. The inverse relation is given by

$$p_c \;=\; p_c^0 \left(s_e^{-1/m} - 1\right)^{1/n}. \tag{8}$$

The quantities m, n and p_c^0 are empirical constants determined by fitting to experimental data.

2.1.2 Relative Permeability

Relative permeability function is based on the Mualem and the quantity n is related to m by the expression

$$m \;=\; 1 - \frac{1}{n}, \qquad n \;=\; \frac{1}{1-m} \tag{9}$$

For the Mualem relative permeability function based on the van Genuchten saturation function is given by the expression

$$k_r \;=\; \sqrt{s_e}\left\{1 - \left[1 - (s_e)^{1/m}\right]^m\right\}^2. \tag{10}$$

2.2 Finite Volume Discretization

The number of degrees of freedom is equal to the number of control volumes N with one degree of freedom, fluid pressure P, per control volume. The following applies to both structured and unstructured grids assuming a two-point flux approximation. For accuracy this requires in the case of an unstructured grid that the line connecting neighboring control volumes be perpendicular to their common interface.

2.2.1 Residual Function

The residual function for the Richards equation at the $k+1$st time level is given by

$$R_n \;=\; \left((\varphi s\rho)_n^{k+1} - (\varphi s\rho)_n^k\right)\frac{V_n}{\Delta t} + \sum_{n'\neq n} F_{nn'}^{k+1} A_{nn'} - Q_{wn}^{k+1} V_n, \tag{11}$$

for the nth control volume with volume V_n and interfacial area $A_{nn'}$, where the sum over n' is over all control volumes connecting with the nth control

volume. The finite volume form of the flux $F_{nn'}$ is given by

$$F_{nn'}^{k+1} \;=\; \rho_{nn'}^{k+1}\,(q)_{nn'}^{k+1}. \tag{12}$$

The Darcy velocity $q_{nn'}$ is evaluated as (the superscript $k+1$ is omitted in the following)

$$q_{nn'} \;=\; -\Big(\frac{kk_r}{\mu}\Big)_{nn'}\left[\frac{P_{n'}-P_n-W_w\rho_{nn'}g(z_n-z_{n'})}{d_{n'}+d_n}\right], \tag{13}$$

where the subscript nn' implies the quantity is evaluated at the interface between n and n'. The density $\rho_{nn'}$ is set equal to the inverse distance mean (*not* arithmetic mean)

$$\rho_{nn'} \;=\; \omega_{n'}\rho_n + (1-\omega_{n'})\rho_{n'}, \tag{14}$$

where

$$\omega_n \;=\; \frac{d_n}{d_{n'}+d_n} \;=\; 1-\omega_{n'}. \tag{15}$$

The quantity in brackets is evaluated using the harmonic mean for permeability and upwinding for mobility $\lambda = k_r/\mu$

$$\Big(\frac{kk_r}{\mu}\Big)_{nn'} \;=\; \frac{k_nk_{n'}(d_{n'}+d_n)}{d_nk_{n'}+d_{n'}k_n}\lambda_{nn'}, \tag{16}$$

where

$$\lambda_{nn'} \;=\; \begin{cases} \lambda_n, & q_{nn'}>0, \\ \lambda_{n'}, & q_{nn'}<0, \end{cases} \tag{17}$$

where $q_{nn'} > 0$ for flow from n to n', and $q_{nn'} < 0$ for flow from n' to n. Combining these relations it follows that

$$q_{nn'} \;=\; -\frac{k_nk_{n'}}{d_nk_{n'}+d_{n'}k_n}\lambda_{nn'}\Big[P_{n'}-P_n-W_w\rho_{nn'}gz_{nn'}\Big]. \tag{18}$$

2.2.2 Jacobian

The Jacobian $J_{nn'}$ is given by the derivatives of the residual function with respect to pressure as

$$J_{nn'} \;=\; \frac{\partial R_n}{\partial P_{n'}}. \tag{19}$$

From the expression for the residual function it follows that

$$\frac{\partial R_n}{\partial P_n} = \frac{V_n}{\Delta t}\frac{\partial}{\partial P_n}\left(\varphi_n s_n \rho_n\right) + \sum_{n' \neq n} \frac{\partial F_{nn'}}{\partial P_n} A_{nn'} - \frac{\partial Q_{wn}}{\partial P_n} V_n, \tag{20}$$

and for $n' \neq n$

$$\frac{\partial R_n}{\partial P_{n'}} = \sum_{n' \neq n} \frac{\partial F_{nn'}}{\partial P_{n'}} A_{nn'} - \frac{\partial Q_{wn}}{\partial P_{n'}} V_n. \tag{21}$$

For the accumulation term one has

$$\frac{\partial}{\partial P_n}\left(\varphi_n s_n \rho_n\right) = s_n \rho_n \frac{\partial \varphi_n}{\partial P_n} + \varphi_n \rho_n \frac{\partial s_n}{\partial P_n} + \varphi_n s_n \frac{\partial \rho_n}{\partial P_n}. \tag{22}$$

The derivative of the flux terms is found to be

$$\frac{\partial F_{nn'}}{\partial P_n} = \frac{\partial \rho_{nn'}}{\partial P_n} q_{nn'} + \rho_{nn'} \frac{\partial q_{nn'}}{\partial P_n}, \tag{23}$$

and

$$\frac{\partial F_{nn'}}{\partial P_{n'}} = \frac{\partial \rho_{nn'}}{\partial P_{n'}} q_{nn'} + \rho_{nn'} \frac{\partial q_{nn'}}{\partial P_{n'}}, \tag{24}$$

with

$$\frac{\partial q_{nn'}}{\partial P_n} = \frac{k_n k_{n'}}{d_n k_{n'} + d_{n'} k_n} \lambda_{nn'} \left\{ 1 + W_w g z_{nn'} \frac{\partial \rho_{nn'}}{\partial P_n} \right\} + \frac{\partial \ln \lambda_{nn'}}{\partial P_n} q_{nn'}, \tag{25}$$

and

$$\frac{\partial q_{nn'}}{\partial P_{n'}} = \frac{k_n k_{n'}}{d_n k_{n'} + d_{n'} k_n} \lambda_{nn'} \left\{ -1 + W_w g z_{nn'} \frac{\partial \rho_{nn'}}{\partial P_{n'}} \right\} + \frac{\partial \ln \lambda_{nn'}}{\partial P_{n'}} q_{nn'}. \tag{26}$$

References

M Tanaka, G Girard, R Davis, A Peuto, and N Bignell. Recommended table for the density of water between 0 c and 40 c based on recent experimental reports. *Metrologia*, 38(4):301, 2001. URL http://stacks.iop.org/0026-1394/38/i=4/a=3.