APPENDIX B

CODE_BRIGHT: FURTHER INFORMATION

Table of contents

B.1. Coupled THM formulation in CODE_BRIGHT ................................................................. 2
  B.1.1. Balance equations ........................................................................................................ 2
    B.1.1.1. Mass balance of solid ......................................................................................... 2
    B.1.1.2. Mass balance of water ...................................................................................... 2
    B.1.1.3. Mass balance of air ......................................................................................... 3
    B.1.1.4. Momentum balance for the medium ................................................................. 4
    B.1.1.5. Internal energy balance for the medium ............................................................ 4
  B.1.2. Numerical treatment of different terms ...................................................................... 4
    B.1.2.1. Treatment of storage terms .............................................................................. 4
    B.1.2.2. Treatment of advective terms .......................................................................... 6
    B.1.2.3. Treatment of non-advective terms (diffusive/dispersive) .................................... 6
    B.1.2.4. Treatment of volumetric strain terms .............................................................. 7
    B.1.2.5. Treatment of mechanical equilibrium equations .............................................. 8
  B.2. Solving systems of equations in CODE_BRIGHT ......................................................... 9

Table of figures

Figure 1: Picture depicting the concept of node (i, j, k), element (e_m, e_n) and cell (painted in broken lines) in a finite element mesh ................................................................. 5
B.1. Coupled THM formulation in CODE_BRIGHT

B.1.1. Balance equations

B.1.1.1. Mass balance of solid

First, mass balance of solid present in the medium can be written as:

\[ \frac{\partial}{\partial t}(\rho_s (1 - \phi)) + \nabla \cdot (j_s) = 0 \]  

(1)

where \( \rho_s \) is the mass of solid per unit of volume of solid and \( j_s \) is the flux of solid. The latter can be written as the velocity of the solid multiplied by the volumetric fraction occupied by the solid phase and the density (i.e. \( j_s = \rho_s (1 - \phi)(d\mathbf{u}/dt) \)). Porosity variation can then be obtained as follows:

\[ \frac{D_s \phi}{dt} = \frac{1}{\rho_s}(1 - \theta) \frac{D_s \rho}{dt} + (1 - \phi) \nabla \cdot \frac{d\mathbf{u}}{dt} \]  

(2)

where the material derivative with respect to the solid (note the subindex \( s \)) has been used. The material derivative of a property \( \psi \) is defined as:

\[ \frac{D_{s} \psi}{dt} = \frac{\partial \psi}{\partial t} + \frac{d\mathbf{u}}{dt} \cdot \nabla \psi \]  

(3)

Equation (2) expresses the variation of porosity caused by volumetric deformation and solid density variation.

B.1.1.2. Mass balance of water

Water is a component (or species) present in liquid and gas phases, so both of them have to be taken into account. The total mass balance of water can be written as:

\[ \frac{\partial}{\partial t}(\omega_1^w \rho_1 S_l \phi + \omega_g^w \rho_g S_g \phi) + \nabla \cdot (j_l^w + j_g^w) = f^w \]  

(4)

where \( S_l \) and \( S_g \) are the liquid and gas phase degree of saturation, respectively; \( \omega_1^w \) and \( \omega_g^w \) are the mass fractions of water in each phase; \( \rho_1 \) and \( \rho_g \) are the phase densities; \( j_l^w \) and \( j_g^w \) are the mass fluxes of water in each phase; and \( f^w \) is an external supply of mass of water. An internal production term is not included because we are performing the total mass balance inside the medium.
As mentioned in chapter 3, the mass flux of a species $i$ ($i=w$ for water) is a combination of:

- A non-advective flux (diffusion + dispersion) written as $i^d_i$ and $i^d_g$
- The advective Darcy flux, written as $q_i$ and $q_g$
- Another advective term caused by solid motion, which is proportional to the solid velocity $du/dt$

Thus, the total flux of the species $i$ can be rewritten as:

$$j^i + j^i_g = i^i + i^i_g + \omega^i \rho_i q_i + \omega^i_g \rho^i q^i_g + (\omega^i_1 \rho^i S_i + \omega^i_g \rho^i S^g)\phi \frac{du}{dt}$$

For water component, and after using the material derivative in (4), we obtain:

$$\phi \frac{D_s}{Dt} (\omega^w \rho^w S^w + \omega^w S^w) + (\omega^w_1 \rho^w S_i + \omega^w_2 \rho^w S^g) \frac{D_s}{Dt} \phi + (\omega^w \rho^i S_i + \omega^w \rho^i S^g)\phi \nabla \cdot \frac{du}{dt} + \nabla \cdot (i^w + i^w_g + \omega^w \rho^w q^w) = f^w$$

If we introduce the mass balance of solid in (6), and after some algebra, we obtain:

$$\phi \frac{D_s}{Dt} (\omega^w \rho^w S^w + \omega^w S^w) + (\omega^w \rho^w S^w + \omega^w \rho^w S^w) \frac{1}{\rho_s} \frac{D_s}{Dt} \rho_s + (\omega^w \rho^i S_i + \omega^w \rho^i S^g)\phi \nabla \cdot \frac{du}{dt} + \nabla \cdot (i^w + i^w_g + \omega^w \rho^w q^w) = f^w$$

### B.1.1.3. Mass balance of air

Air is the main component of the gas phase but may also be present as air dissolved in the liquid phase. As in (6), it can be written as:

$$\phi \frac{D_s}{Dt} (\omega^a \rho^a S^a + \omega^a S^a) + (\omega^a \rho^a S_a + \omega^a \rho^a S^g) \frac{D_s}{Dt} \phi + (\omega^a \rho^a S_i + \omega^a \rho^a S^g)\phi \nabla \cdot \frac{du}{dt} + \nabla \cdot (i^a + i^a_g + \omega^a \rho^a q^a) = f^a$$

The mass balance equations and the deformation of the porous medium are coupled by the volumetric deformation term ($d\varepsilon_v/dt = \nabla \cdot (du/dt)$). This requires the coupled solution of the mechanical equations.
Appendix B: CODE_BRIGHT further information

B.1.1.4. Momentum balance for the medium

If the inertial terms are neglected, the momentum balance reduces to the equilibrium of stresses:

$$\nabla \cdot \sigma + \mathbf{b} = 0$$  \hspace{1cm} (9)

B.1.1.5. Internal energy balance for the medium

The balance of energy is introduced in the case of non-isothermal conditions. Taking into account the internal energy of each phase ($E_s, E_t, E_g$), the equation for internal energy balance for the porous medium can be written as:

$$\frac{\partial}{\partial t} \left[ E_s \rho_s (1 - \phi) + E_t \rho_t \phi + E_g \rho_g S_g \phi \right] + \nabla \cdot \left( \mathbf{i_c} + \mathbf{j_{E_s}} + \mathbf{j_{E_t}} + \mathbf{j_{E_g}} \right) = f^Q$$  \hspace{1cm} (10)

where $\mathbf{i_c}$ is the energy flux due to conduction through the porous medium; the other fluxes ($\mathbf{j_{E_s}}, \mathbf{j_{E_t}}, \mathbf{j_{E_g}}$) are advective fluxes of energy caused by mass motions and $f^Q$ is an internal/external energy supply. An equation formally similar to that of the mass balance of species is obtained after using the material derivative.

B.1.2. Numerical treatment of different terms

B.1.2.1. Treatment of storage terms

The storage or accumulation terms are computed in a mass conservative approach, which discretizes directly the accumulation terms in contrast with the capacitive approach in which the chain rule is used to transform time derivatives in terms of the unknowns (Allen and Murphy, 1986; Celia et al., 1990; Milly, 1984). In general, more accurate solution is obtained when using the mass conservative approach. Mass conservation in time is achieved if the time derivatives are directly approximated by a finite difference in time, while the finite element method for space discretization already conserves mass (Milly, 1984).

To illustrate how CODE_BRIGHT treats storage terms, let’s take a the term representing the variation of water in gas phase in equation 5. In this case, the material derivative with respect to the solid can be approximated as an eulerian derivative because of the small strain assumption:

$$\phi \frac{D_s (\theta^w S_p)}{Dt} \approx \phi \frac{\partial (\theta^w S_p)}{\partial t}$$  \hspace{1cm} (11.a)

After the application of the weighted residual method to the governing equations (and for node $i$ surrounded by elements $e_1, \ldots, e_m$) this expression is transformed into:

$$\int_{\Omega} \xi_i \phi \frac{\partial (\theta^w S_p)}{\partial t} \, dv = \int_{e_1} \xi_i \phi \frac{\partial (\theta^w S_p)}{\partial t} \, dv + \ldots + \int_{e_m} \xi_i \phi \frac{\partial (\theta^w S_p)}{\partial t} \, dv$$  \hspace{1cm} (11.b)
Now, it is convenient to define what an element-wise and a cell-wise variable is (Voss, 1984). As its name indicates, an element-wise variable is space-constant over every element but varies from one element to another and, obviously, can change its value with time. Having said this, we assume that porosity is an element-wise variable, and we will write $\phi^k_{em}$ to express porosity in element $e_m$ at time $t^k$. As for a cell-wise variable, it is a variable which is constant over the cell centered in the node (see figure). The latter are the most convenient to simplify transformations since they would allow the program to compute the aforementioned time derivatives having only information of the values for node $i$.

Since part of these time derivatives are also material dependent (for example, the degree of saturation which depends on porosity or retention parameters), they have to be computed from nodal unknowns but also with material properties of every element in contact with the node. On the other hand, if part of the time derivative is not material dependent (such as density that is a function of pressure and temperature) the corresponding variables are only computed in the node, leading to a kind of modified cell-wise variables.

![Figure 1](image.png)

**Figure 1**: Picture depicting the concept of node $(i, j, k)$, element $(e_m, e_n)$ and cell (painted in broken lines) in a finite element mesh.

With all this, and using a simple finite difference scheme for the time discretization, the following expression for one random integral in 11.b is obtained:

$$
\int_{e_m} \xi_i \phi \frac{\partial (\theta_g \theta_{sg})}{\partial t} \, dv \approx \phi_{em}^k \left( \frac{\theta_{g \theta_{sg}}^{k+1}_{lem} - \theta_{g \theta_{sg}}^k_{lem}}{t^{k+1}-t^k} \right) \int_{e_m} \xi_i \, dv \quad (11. c)
$$

This approximation allows to make the space integration independently of the physical variables, which means that computation of geometrical coefficients is necessary only once for a given finite element mesh. As for the shape function $\xi_i$, its integral over an element is equal to $V_{e_m} / n_{em}$ (i.e. the volume of the element over the number of nodes in that element) and can be computed either numerically or analytically without loss of generality.

All in all, this formulation gives rise to a concentrated scheme. This means that the storage term in node $i$ is only function of unknowns in that node and not others, which is clearly advantageous from a computational point of view (Huyakorn and Pinder, 1983).
B.1.2.2. Treatment of advective terms

Again, the weighted residual method is applied (in fact it is applied to each balance equation). As mentioned, application of the Green theorem is a common procedure when dealing with this type of divergence terms. This allows one to reduce the order of the derivatives and the divergence of flows is transformed into two terms, one of them with the gradient of the shape function. For node \(i\), the advective term considering water in gas phase is treated as follows (note that generalized Darcy’s law has been introduced):

\[
- \int_v \nabla^t \xi_i \theta_i^w v_k \ dV = \left( \int_v \nabla^t \xi_i \theta_i^w \frac{k_{rg}}{\mu_g} \nabla \xi_j \ dV \right) (p_g^j) - \left( \int_v \nabla^t \xi_i \theta_i^w \frac{k_{rg}}{\mu_g} \rho_g \ g \ dV \right) \tag{12.a}
\]

where \(j\) indicates summation over element nodes, such as in \(P_g\) since it is a node-wise variable (Voss, 1984), which mean that it is defined by its nodal values and interpolated on the elements using the shape functions. For node \(i\) volume \(v\) is composed by elements \(e_1, \ldots, e_m\), so the advective terms in (12.a) represent the lateral mass fluxes to cell associated to node \(i\) from contiguous cells. Considering the pressure term first, the contribution of element \(e_m\) tot the total lateral flux towards node \(i\) is approximated with time as:

\[
\left( \int_{e_m} \nabla^t \xi_i \theta_i^w \frac{k_{rg}}{\mu_g} \nabla \xi_j \ dV \right) (p_g^j) \approx \left( \theta_i^w \frac{k_{rg}}{\mu_g} \right)_{e_m}^{k+\theta} \left( \int_{e_m} \nabla^t \xi_i \theta_i^w (k)_{e_m} \nabla \xi_j \ dV \right) (p_g^j)^{k+\theta} \tag{12.b}
\]

where the intrinsic permeability remains in the integral because it is a tensorial quantity. However, if its product with the shape function gradients is slip, its coefficients can be taken out of the integral. Pressure, intrinsic permeability and the remaining coefficients are evaluated, respectively, at intermediate points \((t^{k+\theta}), (t^k)\) and \((t^{k+\varepsilon})\). Since porosity is assumed to vary slowly, intrinsic permeability is evaluated at time \(t^k\), which means that it is handled explicitly. Finally, the integrals of products of shape function gradients are also considered geometrical coefficients, and so they have to be computed for each element but only once for a given mesh (Huyakorn et al., 1986).

The gravity term in (12.a) is approximated in a similar way. Due to the treatment given in (26.b), it is convenient to evaluate density element-wise, so as to balance pressure gradients with gravity forces at element level correctly.

B.1.2.3. Treatment of non-advective terms (diffusive/dispersive)

At first, its structure is similar to that of the advective term. In the balance equation for node \(i\), and after using Fick’s law to compute the diffusive flux, we obtain, typically, the following diffusive term for water in gas phase:

\[
- \int_v \nabla^t \xi_i I_g \ dV = \left( \int_v \nabla^t \xi_i \phi \tau P_g S_g D^w_I \nabla \xi_j \ dV \right) (\omega^w_i) \tag{13.a}
\]
where $\omega_g^w$ is considered a node-wise variable (and, again, $j$ indicates summation over the nodes), $\tau$ is a tortuosity parameter, $D_g^w$ is the molecular diffusion (function of temperature and gas pressure) and $I$ is the identity matrix.

The contribution of element $e_m$ to the total lateral diffusive flux towards node $i$ is approximated with time as:

$$
\left( \int_{e_m} \nabla^T \xi_i \phi \rho_g S_g D_g^w \nabla \psi \psi dV \right) (\omega_g^w)_{ij} \approx (\phi \tau)^{k+\varepsilon} \left( \sum_{e_m} \int_{e_m} \nabla^T \xi_i \phi \rho_g S_g D_g^w \nabla \psi \psi dV \right) (\omega_g^w)_{ij} \quad (13. b)
$$

where similarities with $(26.b)$ are obvious. The treatment of these diffusive terms takes advantage of the fact that the iterative scheme used is the Newton-Raphson method. Mass fractions are directly interpolated and then gradients are computed.

As for the dispersive term, it is treated in a similar way. Dispersivities are element-wise dependent variables and, in principle, the liquid and gas fluxes used to compute the dispersion tensor are also computed element-wise.

**B.1.2.4. Treatment of volumetric strain terms**

If the equation that expresses the variation of porosity is introduced in the other balance equations, porosity only appears as parameter or coefficient and terms of volumetric strain remain in the balance equations. Once space discretization is performed, for node $i$ these terms have the following structure:

$$
\int \xi_i(\alpha) \nabla \cdot \frac{du}{dt} dV = \int \xi_i(\alpha) m^t B \frac{du}{dt} dV \quad (14. a)
$$

where $\alpha$ is defined from the equations, $m^t = (1,1,0,0,0)$ is an auxiliary vector and $du/dt$ is the vector of solid velocities, which is in fact transformed form a continuous vectorial function to a nodal-discrete vectorial function despite keeping the same symbol (i.e. $u_x = \xi_j u_{xj} \ldots$ where $j$ indicates summation). $B$ is the matrix used in the finite element approach for the mechanical problem and is composed of gradients of shape functions ($Zienkiewick and Taylor, 1989$).

Now the approximation of the integral is performed like in $(13.b)$ and $(13.b)$, so the contribution of element $e_m$ to node $i$ is:

$$
\int_{e_m} \xi_i(\alpha) m^t B \frac{du}{dt} dV = (\alpha)^{k+\varepsilon} \int_{e_m} \xi_i m^t B_j \frac{du^{k+\varepsilon}}{t^{k+\varepsilon} - t^k} \quad (14. b)
$$

where $j$ indicates summation over the nodes in element $e_m$, $u$ is the vector of nodal displacements and $B_j$ is the $j$-submatrix of $B$. 


B.1.2.5. Treatment of mechanical equilibrium equations

Once again, the weighted residual method followed by application of the Green’s theorems to the stress equilibrium equation (or momentum balance) leads to:

\[ r(\sigma^{k+1}) = \int_V B^i \sigma^{k+1} \, dv - f^{k+1} = 0 \]  \hspace{1cm} (15.a)

where \( r(\sigma^{k+1}) \) represents the residual corresponding to the mechanical problem, \( \sigma^{k+1} \) is the stress tensor and \( f^{k+1} \) represents both force and boundary traction terms. The aforementioned matrix \( B \) is defined in such a way that stress is a vector and not a tensor.

Now the constitutive model is introduced, hence stresses are related to strains, fluid pressures and temperature. If elasticity is used (and only in this case), the total strain rate is decomposed in the following way:

\[
\frac{de}{dt} = \frac{de^e}{dt} = (D^e)^{-1} \frac{d\sigma^e}{dt} + m \alpha \frac{d(P_g - P_l)}{dt} + m \beta \frac{dT}{dt} \]  \hspace{1cm} (15.b)

where \( D^e \) is the elasticity matrix, \( \sigma^e \) is the net stress tensor (\( \sigma^e = \sigma + m \, P_g \), compression negative) and \( \alpha, \beta \) are coefficients for volumetric elastic dilation. Every point of the medium must satisfy this equation. Since \( e = B \, u \), strains can be written in terms of displacement.

After space and time discretization, we obtain:

\[
h^{k+1} = (\sigma^{k+1} - \sigma^k) - D^e B_i (u^{k+1} - u^k) + m \xi_j (P_{g}^{k+1} - P_{l}^k) + D^e m \xi \alpha (P_{g} - P_{l})^{k+1} - (P_{g} - P_{l})^k \] 

\[
D^e m \xi \beta (T^{k+1} - T^k) \]  \hspace{1cm} (15.c)

where \( h \) is the residual of stresses at every point. Stress can be simply substituted in expression (29.a) if it can be obtained in an explicit way from (15.c). When working with non-linear models, however, a substitution of the differential or incremental forms is necessary.

While according to the numerical approximations proposed for the flow problems we would have a tendency to use element-wise matrices, the mechanical problem shows some peculiarities related to the type of element used which makes it necessary to be computed at the integration points. In other words, stress is not element-wise and a different approximation from what is proposed for flow problems is made. More specifically, the selective integration (B-bar) method is used, and it consists in using a modified form of matrix \( B \) which implies that the volumetric and the deviatoric part of deformation are integrated with different order of numerical integration (Hughes, 1980). Nevertheless, and in spite of the differences, element-wise variables or parameters such as porosity or saturation are maintained.
B.2. Solving systems of equations in CODE_BRIGHT

As mentioned in Chapter 3, CODE_BRIGHT approaches THM problems by solving four equations, each of them associated to an unknown. However, it is possible to solve only some of these equations if the problem is not completely coupled. In any case, equations are solved in a monolithic way, and problems usually require variable time stepping and also an interactive method for the solution, because the problem is generally non-linear.

Since solving complex problems implies that a great amount of numerical data has to be stored, and because of the fact that the resulting systems of equations are usually non-symmetric, matrix storage mode is an important issue when talking about numerical equation solving. Originally, a solver using LU decomposition was implemented along with a band storage mode, which has turned out not to be efficient, especially when more than one degree of freedom is considered. Consequently, more efficient storage of the matrix coefficients had to be considered.

Nowadays, two alternatives to the band storage mode are available:

- **Skyline mode**, which seems adequate for LU-solvers because in such case filling of the matrix is important. It is important to remember that, when matrixes are factorized in such a way, more non-zero coefficients than in the original matrix may appear, but the skyline structure is maintained.

- **Sparse storage mode**, which means that only the non-zero values are stored. This mode is 100% efficient and very convenient when solving systems of equations using an iterative algorithm. In CODE_BRIGHT, the CSR format (Compressed Sparse Row format) is used as it is briefly described in the following lines:

Let \( A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 \\ a_6 & a_7 & a_8 \\ a_9 & a_{10} \end{bmatrix} \) be a \( N \times N \) matrix with \( NZ \) non-zero coefficients.

Then, the non-zero coefficients of \( A \) are stored as follows:

\[
\begin{align*}
AN : & \quad (a_1 \ a_2 \ a_3 / a_4 \ a_5 / a_6 \ a_7 \ a_8 / a_9 \ a_{10}) \\
JA : & \quad (1 \ 3 \ 4 / 2 \ 3 / 1 \ 2 \ 3 / 1 \ 4) \\
IA : & \quad (1 \ 4 \ 6 \ 9)
\end{align*}
\]

where \( AN \) (dimension \( NZ \)) is a real array that contains the non-zero values of the matrix, \( JA \) (dimension \( NZ \)) is an integer array that contains the column indices of the coefficients and \( IA \) (dimension \( N \)) contains the position in \( AN \) where every row begins. This method is also adapted so as to deal with several degrees of freedom per node without changing the size of the vectors \( JA \) and \( IA \).
Finally, several alternatives for solving linear systems of equations (generally, non-symmetric) in an iterative way exist. The conjugate gradients squared method (CGS) proposed by Sonneveld, 1989 and Van der Vorst, 1990 as a modification of the conjugate gradients for non-symmetric matrices is, perhaps, the simplest algorithm that can be used. This type of iterative methods require some extra space and a pre-conditioner, being diagonal scaling (i.e. using the inverse of the diagonal of the matrix) the simplest one. In particular, a block diagonal scaling consists in the inversion of the NDFxNDF (i.e. number of degrees of freedom) block corresponding to each node (block in the matrix diagonal). A quite simple structure of the solver is obtained when using CRS storage mode plus CGS iterative method plus block-diagonal scaling, since only products between vectors or matrices and vector, as well as the inversion of NSFxNDF matrices have to be performed.