Geothermal Modeling in Complex Geological Systems with the ComPASS Code

Simon Lopez¹, Roland Masson², Feng Xing¹,², Laurence Beaudë¹,², Nabil Birgle², Farid Smai¹, Konstantin Brenner², Michel Kern³, Gabriel Courrioux¹, Severine Caritg¹ and Yannis Labeau¹,⁴

1. BRGM, Georesources Division, 3 av. Claude-Guillemin - BP 36009, 45060 Orléans Cedex 02 – France
2. Lab. Jean Alexandre Dieudonné, Université Nice Sophia Antipolis, 06800 Nice Cedex 02 – France
3. INRIA, SERENA, 2 rue Simone Iff, 75589 Paris – France
4. Université des Antilles, Bioosphères, BP 7029, 97275 Schoelcher – France

s.lopez@brgm.fr

Keywords: reservoir modeling, hydrothermal simulations, multiphase flow, unstructured meshes, fracture network

ABSTRACT

Deep geothermal systems often lie in complex geological settings, with multi-scale geological structures that exert a dominant control on convective processes and the transfer of geothermal heat. Methods based on the implicit description of geometrical objects offer an efficient framework to quickly build structural models of such contexts with the occurrence of discontinuities like faults and fractures. Yet, when it comes to discretizing such models the implicit nature of surfaces make volume meshing a non-trivial task and the results are unstructured polyhedral meshes. Over the last few years, much progress has been made towards the consistent and robust discretization of diffusion processes in porous media. These research efforts resulted in several numerical schemes designed with a sound mathematical basis and able to deal with subsurface spatial heterogeneities (permeability variations, anisotropies… ) and general polyhedral meshes. We introduce hereafter the ComPASS platform, an open source initiative that aims at building a geothermal simulation platform relying on one such scheme and recent numerical techniques. The current code is able to handle compositional multiphase Darcy flows, relying on a Coats type formulation, coupled to the conductive and convective transfers of energy. Simulations can be run on unstructured meshes including complex networks of fractures with intersecting fractures. Flow inside the fractures is modelled with a so called hybrid-dimensional model, using a 2D model in the fractures that can have variable apertures and permeability and is coupled with 3D transfers in the matrix. The physics is discretized using a fully implicit time integration combined with the Vertex Approximate Gradient (VAG) finite volume scheme. The fully coupled system is assembled and solved in parallel using the PETSc library and can be run on large computing clusters. An efficient preconditioner is implemented to solve the linear systems at each time step and each Newton type iteration of the simulation. A high level interface to describe the simulations is provided by the Python language, whereas the core routine are written in Fortran and C++. This paper reviews the theoretical foundations and presents a few examples of applications along with the current development perspectives.

1. INTRODUCTION

1.1 Towards a better integration of geological and hydrothermal modeling

Deep geothermal energy allows clean, non-intermittent heat and/or power production with very limited environmental impact. It can have a substantial contribution to the decarbonization of our economy (ANCRE, 2015). However, as for all subsurface natural resources, it is faced with the geological risk. This makes deep geothermal operations risk to high-risk projects with substantial initial investments related to drilling costs. Even if insurance policies have been set up for long and recently adapted to new targets, an observation of the history of French geothermal development shows that a single exploration failure may deter operators from a region with assumed good potential for several decades.

It has been observed that one of the most efficient way to mitigate this geological risk is the collaborative integration of multidisciplinary data and interpretations into a geomodel of the subsurface, or possibly several ones if alternative hypotheses are to be considered (e.g. Calcagnò, 2015). Considering that, the main success indicators of a geothermal project are resource temperature and exploitation flowrate, the first goal of such geothermal conceptual models is often the prediction of the spatial distribution of temperature. Yet, as transient convective processes, which are ubiquitous in high temperature magmatic settings, also control the temperature distribution and the natural state of sedimentary basins and basement type geothermal plays, the aforementioned conceptual models should be dynamic by nature and integrate the quantitative modeling of subsurface mass and energy transfers.

1.2 Geological static modelling

Various types of multi-scale geological structures exert a dominant control on subsurface convective transfers. Among these, fault zones are ubiquitous in tectonically active areas such as high temperature magmatic settings where they act as corridors with a preferential location of flow, spatial discontinuities and sharp contrasts in petrophysical properties. Their role has also been acknowledged as a major control in the thermal structure of sedimentary basins where they can act as connections between aquifer levels (Magri et al., 2010; Person et al., 2012, Simms and Garven, 2004). Moreover, most of the time, feed zones of geothermal wells are found at the intersection of the well path and fault zones or fractures (Grant and Bixel, 2011). The heterogeneous distribution of aperture and permeability along the fracture surfaces leads an overall reduction of the fracture transmissivity and flow channeling (de Dreuzy et al., 2012), that, in the case of geothermal exploitation, will lead to non-uniform temperature decrease of the reservoir.
Methods based on the implicit description of geometrical objects offer an efficient framework to quickly build complex structural models with the occurrence of faults and fractures. Such frameworks are now implemented in commercial off-the-shelf geomodeling software (e.g. GeoModeller, Gocad-SKUA, Leapfrog…), yet the modeling of complex fault systems remains a challenging task (Caumon et al., 2015). The approach chosen in the so-called “potential field method” (Calcagno et al., 2008; Lajaunie et al., 1997) is especially intuitive and powerful as fault slip is not prescribed but automatically deduced from available data. Moreover, the facts that they can be entirely parameterized and do not require manual interaction, like so-called explicit methods (Collon and Caumon, 2017), make that implicit modeling methods are particularly well suited for sensitivity studies and the quantitative analysis of uncertainties associated with 3D geological models (de la Varga et al., 2015; Wellmann et al., 2012, 2014).

Yet, when it comes to produce conformable meshes of such complex geological models in order to run dynamic simulations, the implicit nature of surfaces make volume meshing a non-trivial task. Corner-point grids are widely used in the industry but rely on the sampling of the geological models and generate important approximations of the geometries. State of the art meshing algorithms essentially rely on tetrahedral meshes produced with various algorithms. As an example, the Computational Geometry Algorithms Library (CGAL) offers a powerful 3D mesh generation package dedicated to implicit frameworks. Courrioux et al. (2012) used these algorithms in the GeoModeller software to produce conformal tetrahedral meshes that exactly match any geological object and its 2D geometrical boundaries (geological interfaces, faults…) or 1D sharp features (surfaces intersection, well paths…).

1.3 Dynamic modelling of hydrothermal flows

The full integration of static and dynamic modeling has long been recognized as an ideal long-term goal and involves a considerable amount of interrelated complex operations. Yet, an unsatisfactory one-way linear workflow is still often observed in day-to-day practice. A practical explanation has long been computational limitations with reservoir engineers forced to develop considerable expertise in workarounds and tip and tricks to preserve physical meanings while achieving affordable simulation times. The unfortunate side-effect is often an over-simplification of geological models even though they represent the only modeling level at which the intrinsically complex behavior of the reservoir can be represented.

Indeed, a vast majority of current operational simulations are still performed on cartesian grids. Yet, simulating complex structures is intractable with such grids which do not allow the inclusion of the accurate position of surface discontinuities such as faults or geological interfaces and may lead to so-called grid orientation effects (Eymard et al., 2013). Beyond their inherent simplicity, the use of orthogonal grids is linked to the fact that codes have long been using the two points finite volume scheme that shows very interesting numerical properties (Eymard et al., 2014) but require orthogonality constraints. This orthogonality is natural for grids. It is also an intrinsic property of Voronoï tessellations which may be used to perform simulations but need specific adaptations to preserve geological bodies geometries (Freeman et al., 2014).

However, if one wishes to achieve a two-way interaction between static and dynamic models, the geological model must be kept in a consistent and meaningful state so that one can come back to it, modify it and test the effect of its modifications on dynamic processes.

In the following we focus on the description of the ComPASS simulation platform which is currently developed with the goal to be able to efficiently perform flow simulations on large complex domains such as those obtained from the aforementioned discretization of geological models, without simplification of the underlying geological model. Its main specifications are:

- to perform multiphase multicomponent thermal flow simulation on generic 3D unstructured meshes, possibly containing immersed intersecting fractures, without geometric discretization effects such as “grid orientation effects”,
- to show good convergence behavior when solving the highly nonlinear physics of multiphase hydrothermal circulations,
- to achieve good scalability properties to take advantage of the multi-core and parallel architectures of current computers and benefit from the ever increasing availability of supercomputers,
- to accurately deal with the abrupt variations of petrophysical properties and distributions generated from geostatistical techniques, including aperture/permeability distributions along fault/fracture surfaces,
- include the possibility to specify a wide range of boundary conditions with complementary conditions with a special focus on the modeling of shallow processes (shallow processes in the vadose zone),
- accurately describe transport phenomena (heat and tracer) in an eulerian framework.

Besides the oil & gas industry standards, there has been several recent initiatives for the development of high temperature geothermal reservoir simulators (Burnell et al., 2015; Cacace and Jacquey, 2017; Driesner et al., 2015) and code comparison studies (White and Phillips, 2015 that focused on E.G.S.). This paper is essentially a summary of recent and ongoing work on the ComPASS platform. It will highlight briefly its main components and characteristics. The reader interested in further details is referred to the bibliography, namely the works by Xing et al. (Xing et al., 2017a) for the description of the theoretical and numerical aspects of the modeling of compositional multiphase flows in fractured media, Beaude et al. for the integration of complex well architectures (Beaude et al., 2017a) and the specification of complex boundary conditions (Beaude et al., 2017b, 2018). Additional information is also available on http://www.anr-CHARMS.ORG.

2. THE MODEL AND ITS DISCRETIZATION

2.1 The continuous multiphase multicomponent model

Xing et al. (Xing et al., 2017a) detailed the formulation of the compositional model currently implemented in the ComPASS code, which is based on a Coats’ type formulation (Coats, 1989) and which extends the work by Eymard et al. (2012) to non-isothermal flows. It accounts for an arbitrary nonzero number of components in each phase allowing to model immiscible, partially miscible or fully miscible flows.
\( \mathcal{P} \) denotes the set of phases and \( \mathcal{C} \) denotes the set of components. Each phase \( \alpha \in \mathcal{P} \) is described by its non empty subset of components \( \mathcal{C}^\alpha \subset \mathcal{C} \) in the sense that it contains the components \( i \in \mathcal{C}^\alpha \). It is assumed that, for any component \( i \in \mathcal{C} \), there is at least one phase that can contain \( i \), i.e. the set of phases containing the component \( i \) denoted as \( \mathcal{P}_i = \{ \alpha \in \mathcal{P} | i \in \mathcal{C}^\alpha \} \) is non-empty. The number of phases and number of components are hard-coded for each thermodynamical module along with the previous relations between phases and components which is implemented as the following boolean table:

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( i_1 )</th>
<th>( i_2 )</th>
<th>( i_3 )</th>
<th>( i_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

\( \mathcal{C}_m = \{ i_1, i_3, i_4 \} \)

\( \mathcal{P}_i = \{ \alpha_2, \alpha_3 \} \)

Table 1: Specification of the relations between phases and components

<table>
<thead>
<tr>
<th>Subcritical pure water</th>
<th>Black-oil model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{H}_2\text{O} )</td>
<td>( \text{liquid} )</td>
</tr>
<tr>
<td>( \text{gas} )</td>
<td>✔</td>
</tr>
</tbody>
</table>

Table 2: Example on two simple cases: subcritical pure water (on the left) and so-called black-oil model (on the right)

As the model takes into account phase change reactions which are assumed to be at equilibrium, it results that phases can appear or disappear. The unknown representing the set of present phases is noted \( \mathcal{Q} \subset \mathcal{P} \) with \( \mathcal{Q} \neq \emptyset \). For a given set of present phases \( \mathcal{Q} \), it may occur that a component \( i \in \mathcal{C} \) does not belong to the subset \( \bigcup_{\alpha \in \mathcal{Q}} \mathcal{C}^\alpha \subset \mathcal{C} \). Hence, we define \( \overline{\mathcal{C}}_i \) the subset of absent components, in the sense that there is no phase in \( \mathcal{Q} \) that could contain \( i \) (what implies that \( i \) can not be present). For example, taking the example of table 1, if \( \mathcal{Q} = \{ \alpha_2, \alpha_3 \} \) we will have \( \overline{\mathcal{C}}_i = \overline{\{ \alpha_2, \alpha_3 \}} = \{ i_1 \} \).

Then, thermodynamical properties of each phase \( \alpha \in \mathcal{P} \) depend on the phase pressures \( P^\alpha \), the temperature \( T \), and the phase molar fractions \( \mathcal{C}^\alpha = (\mathcal{C}^\alpha_i)_{i=1}^{\mathcal{Q}} \). These properties are specified by the user as hard coded functions along with their derivatives.

The formulation relies on the so-called natural variables and uses the following set of unknowns:

\[
X = \left\{ P^\alpha, T, (S^\alpha)_{\alpha \in \mathcal{Q}}, (\mathcal{C}^\alpha_i)_{i=1}^{\mathcal{Q}}, (n_i)_{i=1}^{\mathcal{Q}} \right\}
\]

where \( S = (S^\alpha)_{\alpha \in \mathcal{P}} \) is the vector of the phase volume fractions (saturations) and \( n_i \) is the number of moles of the component \( i \in \mathcal{C} \) per unit pore volume. The latest is defined as an independent unknown when \( i \in \overline{\mathcal{C}}_i \) and otherwise given by the formula

\[
n_i(X) = \sum_{\alpha \in \mathcal{Q} \setminus \mathcal{P}_i} \mathcal{C}^\alpha_i \mathcal{C}_i^\alpha \quad \text{with } \mathcal{C}^\alpha_i \text{ being the phase molar density.}
\]

Then, one classically writes the molar conservation for each component \( i \in \mathcal{C} \) and the energy conservation that writes in the form of the following system of P.D.E.:

\[
\begin{align*}
\phi \mathcal{H} n_i + \nabla \cdot \mathbf{q}_i & = 0 \quad i \in \mathcal{C} \\
\phi \mathcal{H} E + (1 - \phi) \mathcal{H} E_e + \nabla \cdot \mathbf{q}_e & = 0
\end{align*}
\]

(2)
Lopez, Masson, Xing et al.

where \( t \) denotes time, \( \phi \) is rock porosity which is assumed to be constant, neglecting mechanical stress, \( E \) and \( E_r \) are respectively the fluid and rock internal energy per unit rock volume. \( \mathbf{q} \) denotes the molar flux of component \( i \). It is obtained using a generalized Darcy velocities for each phase \( \alpha \in \mathcal{Q} \cap \mathcal{P}_i \) that involves the phase relative permeabilities, dynamic viscosities and the rock intrinsic permeability tensor. \( \mathbf{q} \) is the energy flux obtained as the sum of an advective component related to the enthalpies of the phases advected by the aforementioned phase Darcy velocities and a diffusive component given by the Fourier law and involving the rock bulk thermal conductivity.

The previous system of conservation equations is finally coupled to the following local closure laws:

\[
\begin{align*}
\sum_{\alpha \in \mathcal{Q}} S^\alpha &= 1 \\
\sum_{\alpha \in \mathcal{C}_i} C^\alpha &= 1 & \alpha \in \mathcal{Q}
\end{align*}
\]

and the thermodynamical equilibrium for each component \( i \) present in at least two phases among the set of present phases \( \mathcal{Q} \). This equilibrium is currently written:

\[
f_i^\alpha (P, T, C^\alpha) C_i^\alpha = f_i^\beta (P, T, C^\beta) C_i^\beta & \quad \alpha \neq \beta (\alpha, \beta) \in (\mathcal{Q} \cap \mathcal{P})^2 & i \in \mathcal{C} \tag{4}
\]

where \( f_i^\alpha \) is the fugacity of component \( i \) in phase \( \alpha \). The system is finally completed with an additional equation for the discrete unknown \( \mathcal{Q} \) which is typically obtained by a flash calculation:

\[
\mathcal{Q} = \mathcal{Q}_{\text{flash}} (X) \tag{5}
\]

or by simpler criteria depending on the specific thermodynamical system.

2.2 Hybrid model including a discrete fracture network

Two classes of models, dual continuum and discrete fracture models, are typically employed and possibly coupled to simulate flow and transport in fractured porous media. Dual continuum models assume that the fracture network is well connected and can be homogenized as a continuum coupled to the matrix continuum using transfer functions (e.g. the MINC approach in TOUGH2 (Pruess, 1992)). On the other hand, discrete fracture models (DFM), represent explicitly the fractures as codimension one surfaces immersed in the surrounding matrix domain.

ComPASS uses the DFM approach with a lower dimension physical model along the fracture which is derived from the equi-dimensional model of the previous paragraph by integration and averaging along the - possibly variable - width of each fracture. The resulting so-called hybrid-dimensional model couples the 3D model in the matrix with a 2D model in the fracture network taking into account the jump of the normal fluxes as well as additional transmission conditions at the matrix fracture interfaces. These transmission conditions depend on the mathematical nature of the equi-dimensional model and on additional physical assumptions depending on the fracture behavior (drain vs. barrier). The interested reader is referred to the paper by Xing et al. (Xing et al., 2017a) for further details.

2.3 Spatial and temporal discretizations

Over the last few years, much progress has been made in the consistent and robust discretization of diffusion processes in porous media. These research efforts resulted in several numerical schemes designed with a sound mathematical framework and able to deal with subsurface spatial heterogeneities (permeability variations, anisotropies…) and general polyhedral meshes. ComPASS currently implements the Vertex Approximate Gradient (VAG) finite volume scheme (Eymard et al., 2012) which belongs to a broader family of numerical scheme called Gradient Schemes (Droniou et al., 2016).

The VAG discretization of hybrid-dimensional two-phase Darcy flows was introduced by Brenner et al. (2016) and generalized to multiphase multicomponent flow by Xing et al. (Xing et al., 2017a). It considers generalised polyhedral meshes of the simulation domain, which are assumed conforming. The cells are star-shaped polyhedrons, and faces are not necessarily planar in the sense that they can be defined as the union of triangles joining the edges of the face to a so-called face centre.

The VAG scheme has different degrees of freedom associated with: cell, cell nodes and fracture face (cf. figure 1). It is a control volume scheme in the sense that it results, for each non Dirichlet degree of freedom, in a molar or energy balance equation. The two main ingredients are therefore the conservative fluxes and the control volumes around degrees of freedom. Figure 1 shows one cell \( K \) with one fracture face \( \sigma \) in bold. The matrix fluxes (in blue on figure 1) connect the cell \( K \) to the degrees of freedom located at the boundary of \( K \): namely the cell nodes (e.g. \( F_{K,s} \) and \( F_{K,s'} \)) and the fracture faces if any (e.g. \( F_{s,k,d} \)). The fracture fluxes (in red on figure 1) connect each fracture face \( \sigma \) to its nodes (e.g. \( F_{\sigma,s} \)). The expression of the matrix fluxes is linear and local to each cell as the expression of
fracture fluxes is linear and local to each fracture face. Fluxes ensure the cell-cell, cell-fracture face and fracture face-fracture face connections.

Figure 1: Degrees of freedom and fluxes of the VAG discretization of the hybrid model (cf. text for explanation).

Then, the construction of the control volumes at each degree of freedom is based on partitions of the cells and of the fracture faces: cell (resp. fracture face) volumes are splitted between the cell (resp. fracture face) center and its boundary nodes. In the usual case of cellwise constant rocktypes in the matrix and facewise constant rocktypes in the fracture network, the implementation of the scheme does not require to build explicitly the geometry of these partitions and it is sufficient to define the volume fractions.

As shown in Brenner et al. (2015), the flexibility in the choice of the control volumes is a crucial asset, compared with usual Control Volume Finite Element Methods (CVFE) approaches and allows to significantly improve the accuracy of the scheme when the permeability field is highly heterogeneous. Figure 2, also shows that, as opposed to usual CVFE approaches, this flexibility allows to define the control volumes in the fractures with no contribution from the matrix in order to avoid the artificial enlargement of the flow path in the fractures thus limiting numerical diffusion.

Figure 2: On the left: two cells (K and L) splitted by one fracture face (in red). On the right: example of control volumes associated with the two cells centers (light grey) and with the fracture face center (dark grey) and with (matrix and fracture) nodes (in white). The width of the fracture is enlarged.

To avoid too small control volumes at the nodes located at the fracture intersection, all the fracture faces containing such a node share their volume with it. It results that the control volumes at the fracture intersection nodes are not smaller than at any other fracture degrees of freedom.

The time discretization is based on a fully implicit Euler scheme to avoid severe restrictions on the time steps due to the small volumes and high velocities in the fractures. A phase based upwind scheme is used for the approximation of the mobilities in the Darcy fluxes.

2.4 Non-linear solver
Spatial and temporal discretization of the system of conservation equations (2), including boundary conditions, results in a discrete non linear system $R(X) = 0$ for each degree of freedom with $X = (X_v)_{v \in \mathcal{D}}$ denoting the set of all unknowns (1) for all degrees of freedom $\mathcal{D}$ (cells, cell nodes and fracture faces). The problem is closed by adjoining a discrete non linear version $L_v(X_v) = 0$ of the closure laws (3) and thermodynamic equilibrium (4).

Then the non-linear system to be solved at each time step writes in vector form $R(X) = 0$ with:
Lopez, Masson, Xing et al.

\[
\mathcal{R}(\mathbf{X}) = \begin{cases} 
R_s(\mathbf{X}) & \text{for each node } s \\
L_s(\mathbf{X}) & \\
R_{\sigma}(\mathbf{X}) & \text{for each fracture face center } \sigma \\
L_{\sigma}(\mathbf{X}) & \\
R_K(\mathbf{X}) & \text{for each cell center } K \\
L_K(\mathbf{X}) & 
\end{cases}
\]  

(6)

The non-linear system \( \mathcal{R}(\mathbf{X}) = 0 \) coupled to the flash fixed point equations (5) expressed locally at each degree of freedom \( Q_e = Q_{\text{flash}}(X_J) \) is solved by an active set Newton-Raphson algorithm (e.g. Coats, 1989) which is detailed by Xing et al. (Xing et al., 2017a). For each Newton-Raphson iteration the jacobian matrix is computed:

\[
J(\mathbf{X}) = \frac{\partial \mathcal{R}}{\partial \mathbf{X}}\bigg|_{\mathbf{Q}=Q_{\text{flash}}(\mathbf{X})}
\]  

(7)

to solve a linear system of the form:

\[
J(\mathbf{X})\Delta \mathbf{X} = -\mathcal{R}(\mathbf{X})
\]  

(8)

An important point is that the part of the jacobian \( J(\mathbf{X}) \) of the full system (6) corresponding to the sub-systems \( L_s(\mathbf{X}) = 0 \) appearing in each Newton-Raphson iteration are local to each degree of freedom. Consequently, they can be eliminated straightforwardly by splitting the unknowns between primary and secondary unknowns. Additionally, an interesting property of the VAG scheme is that the part of the jacobian \( J(\mathbf{X}) \) corresponding to the sub-systems \( R_K(\mathbf{X}) = 0 \) are local to each cell and only involves degrees of freedom located on the cells (cell nodes, cell center and possibly fracture face center). It results that, for each Newton-Raphson iteration, the cell unknowns can be eliminated too and expressed as linear combinations of the nodes and fracture face unknowns through a Schur complement system. These two successive elimination steps considerably reduce the size of the linear system to be solved for each Newton-Raphson iteration. It makes that the ComPASS implementation of the VAG scheme is particularly efficient on meshes composed mainly of tetrahedrons.

Then, the resulting ill-conditioned linear system is solved using an iterative solver (typically GMRES) combined with a preconditioner adapted to the elliptic or parabolic nature of the pressure unknown and to the coupling with the remaining hyperbolic or parabolic unknowns. One of the most efficient preconditioners for such systems is the so-called CPR-AMG preconditioner introduced by Lacroix et al. (2001) and Scheichl et al. (2003). It combines multiplicatively an algebraic multigrid preconditioner (AMG) for a pressure block of the linear system with a more local preconditioner for the full system. Currently, the pressure equations defining the pressure block are obtained as the sum of the molar conservation equations in each control volume. Then an Incomplete LU factorization of order 0 is used for the full system.

2.5 Wells integration

A first version of reservoir well interaction has been implemented into the ComPASS platform and validated for monophasic cases (Beaude et al., 2017a). Even at a few kilometers scale – typical reservoir scale, the mesh can hardly resolve the well boundary with a radius of the order of 10 cm. Then, the well is modeled as a Dirac source term along the well trajectory. Most well models in reservoir simulations are defined by a set of connected perforations, each perforation belonging to a cell of the mesh (e.g. source block in TOUGH2). In order to take advantage of unstructured meshes and of the nodal feature of the VAG scheme, it is more convenient in our case to discretize each well as a subsidence of edge of the mesh. This alternative approach provides an efficient way to represent slanted and multi-branch wells. Consequently, wells can easily be taken into account into the mesh generation step by integrating the well paths as additional mesh constraints (e.g. via the sharp edge features in the CGAL 3D Mesh Generation Package).

The fluxes connecting the well with the 3D matrix and the 2D fracture network at each node of the well are computed using Peaceman’s approach (Chen and Zhang, 2009; Peaceman, 1978, 1983). It is based on a Two Point Flux Approximation (TPFA) with a transmissibility taking into account the unresolved singularity of the pressure (or temperature) solution in the neighborhood of the well. The non-isothermal flow model inside the well is defined using a single implicit unknown for each well corresponding to a reference pressure often coined as the bottom hole pressure. The pressures along the well are then deduced from this bottom hole pressure with the crude assumption that the pressure is hydrostatic inside the well. The temperatures along the well are computed assuming thermal equilibrium and a stationary flow inside the well.
which are discretized by mass flow rate of the mesh, distribution of p with a total mass flow rate of 10^5.10^-5. The mesh is a 3D tetrahedral mesh conforming to the fracture network and to the wells (cf. figure 3). It was generated using the implicit framework from the 3D mesh generation package from the Computational Geometry Algorithms Library (CGAL). There is one injection well (red line in figure 3) and one multi-branch production well (green line in figure 3). This mesh contains a little less than 5.10^6 cells, 3.10^6 fracture faces and 8.10^7 nodes. The radius of both wells is set to 10 cm and the fractures are assumed to represent high permeability zones with a constant thickness of 1 m. The permeabilities are isotropic and set to 10^{-14} m^2 in the matrix domain and to 10^{-11} m^2 in the fracture network. The porosities are respectively set to 0.1 in the matrix domain and 0.4 in the fractures.

Approximate analytical Peaceman type formulas were used for this test case and were considered to provide a good order of magnitude of the Peaceman indices. The domain is initially set to a uniform temperature of 140°C. At the injection well, a 60°C cold water is injected with a total mass flow rate of 100 t/h and a limit maximum bottom hole pressure of 30 MPa. At the production well, hot water is produced
Lopez, Masson, Xing et al.

with the opposite total mass flow rate and a minimum bottom hole pressure of 1 MPa. Figure 4 shows the temperature in the matrix domain and in the fractures after nearly 30 years of continuous exploitation.

![Figure 4: Temperature in K, in the matrix domain (left) and in the fractures (right) after nearly 30 years of continuous exploitation.](image)

3.2 Two-phase subcritical convection

This test case is detailed by Xing et al. (Xing et al., 2017a). The simulation domain is a cube with 3 km edges and rectangle shaped fault zones. A conforming tetrahedral mesh was generated with Tetgen (Si, 2015) which is shown in figure 5.

![Figure 5: Conforming tetrahedral mesh of the simulation domain (matrix domain on the left, fractures on the right)](image)

The fault zones thicknesses are still fixed to 1 m. Matrix (respectively fracture network) permeability is set to the uniform and isotropic value of $10^{-14}$ m$^2$ (respectively $10^{-12}$ m$^2$). Matrix (respectively fracture network) porosity is set to 0.25 (respectively 0.35).

At the intersection of the bottom boundary of the domain with the fault network, the temperature is fixed to 450 C and a total mass flow rate of 100 kg/s is prescribed. The matrix bottom boundary is set to 200°C with no flow conditions. The top boundary, is set to atmospheric conditions (1 bar, 20°C) with Dirichlet type conditions so that the fluid is liquid. A zero flux condition for both mass and temperature is imposed at the lateral boundaries of the domain. The simulation domain is initially in liquid phase with a hydrostatic pressure defined by the pressure boundary condition at the top boundary and a linear decrease of the temperature from matrix bottom conditions to surface conditions. Figure 6 shows the temperature in the faults (left) and the gas saturation in the faults and in the matrix domain (right) after 200 years of simulation.
3.3 Regional scale simulation

Figure 7: Preliminary results from the modeling of the geothermal potential of the Baie Lamentin area (Martinique). A geological model (upper left) was made then meshed (lower left). Right figures show the yet uncalibrated results of temperature in the matrix (upper right) and the fracture network (lower right).

Figure 7 shows preliminary results from Yannis Labeau Ph.D thesis. A geological model of the Baie Lamentin area (Martinique) that includes several fault zones and volcanic formations was made with the GeoModeller software. A ten million cells conforming mesh was generated using the CGAL libraries (Courrioux et al., 2012). Classical geothermal area conditions were applied with a deep heat source simulated as a temperature Dirichlet conditions and a fracture network which is 3 orders of magnitude more permeable than the surrounding matrix formations. The dynamic model is currently being calibrated to observation data. Each run takes approximately 2 hours on 120 processors on the BRGM computing cluster (5 computing nodes with two AMD Abu Dhabi à 2.6 GHz, 12 cores each and 64 Gb of memory).
Lopez, Masson, Xing et al.

4. PERSPECTIVES

4.1 API choices and code development

On the software level we target a modular evolutionary and open framework with a two levels API for the ComPASS platform. The first level targets operational users that want to perform reservoir simulation whether the other one targets numerical engineer and researchers. The version 3 of the ComPASS code was mainly written in Fortran and C++ for the links to the METIS partitioning library and VTK visualization library. Current development version will be released as version 4. It involves a large amount of code refactoring with a shift towards modern C++ and a Python API.

Python is a high level programming language which is widespread amongst the engineering and scientific communities (e.g. Sullivan et al., 2013). It has a lot of efficient libraries designed for scientific work (e.g. the NumPy package) and gives the user the ability to write clean, concise and efficient scripts in a natural and understandable way, considerably reducing development cycles. The use of Python does not preclude the use of static form input formats that can directly be managed through the Python interface and data serialization standard/language along with dedicated libraries (e.g. YAML). This option is currently left to the users.

As of today, output of the simulation are performed as compressed binary numpay arrays that can be readily converted to Paraview parallel file formats using provided scripts.

Development is currently performed through INRIA’s instance of the Gitlab collaborative platform, under a GPLv3 license. Access to the code can be granted freely upon request at anr-charms@brgm.fr. Several tests are currently defined and gathered for continuous integration.

4.2 Performance and scalability

Version 3 of the ComPASS code benefited from a performance assessment in the context of a joint workshop between the EOCOE and POP centres of excellence for High Performance Computing. The code performed well and potential improvements could be identified.

The current multiphase multicomponent version of the code shows good scalability results (Xing et al., 2017a). However, the scalability of the simulation depends highly on the scalability and the robustness of the preconditioner and linear solver used at each Newton-Raphson iteration. It is well known that AMG type preconditioners require a sufficient number of unknowns per MPI process, say 10,000 as a classical order of magnitude, to achieve a linear strong scaling. For example, on test cases with 2,10^6 unknowns for the pressure block, ComPASS runs show a scalability that is still not far from linear on up to 64 processes and then degrades more rapidly for 128 processes. A detailed instrumentation of the code clearly shows that this degradation is mainly due to the communication overhead. This is consistent with PETSc FAQ according to which “there must be enough work for each process to outweigh the communication time […] [with] an absolute minimum of about 10,000 unknowns per process, better [being] 20,000 or more.”

4.3 Physics

Thermodynamic modules are hard coded in Fortran/C++ so that they can be optimized at compile time. Nevertheless, they can be dynamically loaded at execution, what makes their use quite flexible. We currently implemented in ComPASS, three simple set of thermodynamic properties: subcritical water with relatively simple thermodynamic functions, diphasic module with two components (water and air), and a black-oil module (cf. table 2), mainly used for test purposes and benchmarking with oil and gas software. Developments being currently considered are supercritical water/brine (Croucher and O’Sullivan, 2008; Driesner and Heinrich, 2007; Weis et al., 2014) and the inclusion of non condensable gas (Battistelli et al., 2009).

The general multiphase multicomponent framework used in ComPASS makes it relatively straightforward for the user to code new thermodynamic modules provided that he implements the numerical version of the thermodynamic functions he wants to use along with their derivatives. Current improvements of the Python API aim at providing the possibility to explicitly specify equation of states and thermodynamic properties on the Python side to achieve a full configuration of test cases in a high level programing language, without performance loss.

4.5 Boundary conditions

One of the main target of the current developments (cf. also Laurence Beaude et al., 2018) is to overcome several limitations in terms of boundary conditions. Classical choices for boundary conditions are often limited to the following types: fixed value/Dirichlet type for all primary variables or fixed fluxes/Neumann type for all conserved quantities. Mixed-type transient boundary conditions are rarely supported which impedes the convenient modeling of natural processes such as recharge or seepage or water table fluctuations. Workarounds may exist (e.g. Hurwitz and Kipp, 2003) but are relatively tedious to implement and are not formulated in a generic way. Transient complex upper or lower boundary conditions could be helpful in taking into account some crucial processes ranging from more realistic heat source modelling to interactions between the vadose zone and fresh water recharge that may hide geothermal resources.

5. CONCLUSIONS

We introduced the open source ComPASS platform which is able to handle compositional multiphase Darcy flows, relying on a Coats type formulation, coupled to the conductive and convective transfers of energy. Simulations can be run on unstructured meshes including complex networks of fractures with intersecting fractures such as those obtained from the discretization of complex geological models. Flow inside the fractures is modelled with a hybrid-dimensional model, using a 2D model in the fractures that can have variable apertures and permeabilities and is coupled with 3D transfers in the matrix. Many of the developments currently performed aim at achieving a
flexible API through a high-level programing language without loss of performances so that geological and dynamic models can be more tightly and easily integrated, leading to both better conceptual models of geothermal systems and operational reservoir models.

ACKNOWLEDGEMENTS

The work of Feng XING who is the main contributor to the version 3 of the ComPASS code was supported by a joint project between INRIA and BRGM Carnot institutes (ANR, INRIA, BRGM). Current developments around the ComPASS code are performed in the framework of the consortium of the CHARMS project between BRGM, Nice Sophia-Antipolis University, Maison de la Simulation, STORENGY and Sorbonne Université (UPMC). The CHARMS project is supported by ANR grant ANR-16-CE06-0009. Laurence BEAUDE’s Ph.D grant is jointly supported by BRGM and Région PACA. The Ph.D grant of Yannis Labeau is supported by ADEME Martinique.

REFERENCES


Lopez, Masson, Xing et al.


