Adaptive Mesh Refinement and Time Stepping Strategies
for Incorporating Discrete Fracture Networks Into a High Performance Computing Framework for Geothermal Reservoir Simulation

Katherine Smith1, Mitch Plummer1, Jacob Bradford2, and Robert Podgorne1

1Energy Resources Recovery and Sustainability Department, Idaho National Laboratory
2Energy & Geoscience Institute, University of Utah

Keywords
Reservoir simulations, discrete fracture networks, adaptive mesh refinement, adaptive time stepping, high performance computing, thermo-hydro-mechanical simulator, thermal stimulation

ABSTRACT

In order to better understand how geothermal reservoirs with spatially varying permeability and complex fracture networks behave in response to temperature and pressure stimulations due to subsurface exploration, we have developed a fully-coupled, fully-implicit, single and two-phase thermo-hydro-mechanical (THM) geothermal reservoir simulator, FALCON (Fracturing And Liquid CONvection) that is capable of modeling complex fracture geometry. We briefly review the preconditioned JFNK method that enables FALCON to solve systems of highly non-linear and tightly coupled geothermal reservoir physics in an efficient manner. Details of the recent addition of two discrete fracture networks (DFN) readers, a FracMan®[4] file reader and a 2D image reader, and how they facilitate FALCON to model complex and field-data-supported fracture networks is provided. In order to accurately capture these DFN’s and the solution without incurring a greater computational cost, adaptive mesh refinement (AMR) schemes are employed. Additionally, to reduce overall solve time for these large-scale simulations, adaptive time stepping schemes are applied. The DFN reader, AMR and adaptive time stepping are applied to an example THM simulation, where combinations of three different meshing schemes (static, conservative AMR, and aggressive AMR) and three different time stepping schemes (static, 10% step increase, and 20% step increase) are compared. A significant performance gain is seen in total computation time and calculated degrees of freedom (DOFs) when using both the AMR and adaptive time stepping.

2 Introduction

Our ability to expand geothermal energy production requires a greater capability to describe and predict complex subsurface interactions in highly heterogeneous and complex fracture networks. Development of high-fidelity simulations that fully encompass the complex dynamics of tightly coupled fluid-flow, heat-transfer, and rock-deformation within a fractured network enables us to predict the impacts and potential successes of future subsurface explorations. However, sound predictions, at scales relevant to large geothermal reservoirs, require solving problems with a large number of unknowns. This is particularly challenging when systems of governing partial differential algebraic equations (PDAEs) are highly nonlinear and tightly coupled due to the complex interactions among processes. Additionally, modeling fractures with high resolution can be resource intensive.

In this paper, we present a fully-coupled, fully-implicit, multiphysics simulator for solution of reservoir-scale, single phase or two-phase thermo-hydro-mechanical (THM) problems (FALCON (Fracturing and Liquid CONvection)), and outline capabilities that have recently been added that enable the modeling of large fracture networks. A more comprehensive overview of FALCON’s capabilities can be found in previous reports [1]. FALCON is developed in the Multiphysics Object Oriented Simulation Environment (MOOSE) [2] computing framework developed at INL. MOOSE’s use of a state-of-the-art nonlinear solver, the preconditioned Jacobian-Free-Newton-Krylov (JFNK) method [3], enables efficient simultaneous solution of the coupled nonlinear PDAEs, without explicitly computing and storing the Jacobian. Solving these large, tightly coupled THM problems with this globally implicit approach (GIA) makes for a more accurate solution then the widely used operator splitting approach [2, 5] and avoids the tedious and potentially error-prone practice of coupling multiple codes. The preconditioned JFNK solution approach will be briefly explained in Section 3. Sections 4.1 details the recent addition of two discrete fracture network (DFN) readers, a 2D image reader and a FracMan®[4] file reader, within FALCON that allows for the inclusion of statistically field realistic subsurface fracture networks within THM simulation domains. However, uniformly meshing these large domains at a level sufficient to capture these fractures can come at a high computational cost. MOOSE and FALCON are built upon a large finite element library, libMesh [6], which enables the use of adaptive mesh refinement/coarsening schemes.
and parallelization to reduce CPU time with very little reduction in solution accuracy. The specific mesh adaptivity schemes used in solving THM problems with FALCON are explained in Section 4.2. When solving these large reservoir-scale problems on time scales relevant to predicting thermal depletion and long term performance, employing small static time steps in order to capture acute events can lead to long solve times. The adaptive time stepping capabilities within FALCON allow for adjustable and automated time step size in accordance with the current state of the solution. Section 4.3 briefly discusses the use of a simple percent increase time stepping algorithm which is frequently used within FALCON to speed up solve time. Section 5 compares results for nine simulation runs of the same THM problem, combinations of static/conservative AMR/aggressive AMR and static/conservative/aggressive time step adaptivity. Solve-times, degrees of freedom (DOFs) solved for, and percent error are compared to assess the validity of this adaptive meshing and time stepping strategies.

3 Preconditioned Jacobian-Free Newton-Krylov Method

Newton’s method for solving coupled nonlinear PDEs typically begins with a discrete form of the governing PDEs and casts it into a general residual function

$$ F(u) = M(u)\dot{u} + K(u)u - R(u) = 0, \quad (1) $$

where $u$ is the solution vector, $M$ is the mass accumulation matrix, $K$ is the stiffness matrix (often with element values as functions of $u$) and $R$ is the source term vector. $F(u): \mathbb{R}^l \rightarrow \mathbb{R}^l$ is the system nonlinear residual vector, where $N$ is the number of unknowns. The traditional Newton iterative method typically requires the full Jacobian matrix

$$ J(u) = \frac{\partial F(u)}{\partial u} \quad (2) $$

to update the solution vector by solving the linearized system

$$ J(u^{(k)})\delta u^{(k)} = -F(u^{(k)}) \quad (3) $$

followed by an update of the solution state $u^{(k+1)} = u^{(k)} + \delta u^{(k)}$. In this process, forming each element of $J$ can be difficult, time consuming and error-prone. Storing full $J$ matrix also requires large amount of memory.

To avoid such hurdles, by taking the advantage of the fact that a Krylov solver does not require a full $J$ in solving Eq. 3, the JFNK solution approach is adopted in our solution approach. Staring with an initial guess of $(\delta u)_0$, initial linear residual is formed according to

$$ r_0 = -F - J \cdot (\delta u)_0 \quad (4) $$

then the approximate solution of Eq. (3) at the $p$th Krylov iteration is constructed from a linear combination of the Krylov vectors $\{r_0, Jr_0, (J)^2r_0, \ldots, (J)^{p-1}r_0\}$ constructed from the previous $p-1$ Krylov iterations,

$$ (\delta u)_p = (\delta u)_0 + \sum_{j=0}^{p-1} \alpha_j (J)^jr_0 \quad (5) $$

where the scalar coefficient $\alpha_j$ is part of the Krylov iteration. Eq. 5 shows that the Krylov method for solving Eq. 3 only requires the product of the Jacobian matrix $J$ and Krylov vector $v$, not the Jacobian itself. Specifically, to evaluate this matrix-vector product, [Equation 5-2], a finite difference approach, Eq. 6, can be used, where $e$ is a very small perturbation.

$$ J(u^{(k)})v \approx \frac{F(u^{(k)} + e v) - F(u^{(k)})}{e} \quad (6) $$

The use of preconditioning in solving the linear system of Eq. 3 is to efficiently cluster eigenvalues of the iteration matrix, which in turn will reduce the number of Krylov iterations required for convergence. When applying the right preconditioning to Eq. 3, one solves

$$ \left( J(u^{(k)})P^{-1} \right) \left( P \delta u^{(k)} \right) = - F(u^{(k)}) \quad (7) $$

where $P$ is a linear operator and symbolically represents the preconditioning process. $P$ is chosen in the manner such that it is a suitable approximation to the Jacobian matrix. The right preconditioned version of the JFNK matrix-vector product approximation of Eq. 6 becomes

$$ \left( J(u^{(k)})P^{-1} \right) v \approx \frac{F(u^{(k)} + e P^{-1}v) - F(u^{(k)})}{e} \quad (8) $$

Thus the preconditioned JFNK process is implemented in two steps: 1) Solve $Py = v$ for $y$; 2) Perform matrix-vector product approximation using Eq. 8; and repeat steps 1 and 2 until the Krylov iteration converges. In a physics-based preconditioning approach, the preconditioner $P$ is not obtained by algebraically manipulating the Jacobian matrix; rather, approximations are made to the original differential system to form an approximate Jacobian only for preconditioning purpose. We illustrate a general implementation of the physics-based preconditioner below.

The pressure-temperature formulation for flow and heat transfer is shown in Eq. 9 and 10:

$$ \frac{\partial(np)}{\partial t} - \nabla \cdot \left( \frac{k_p}{\mu_w} (\nabla p - \rho g z) \right) - Q = 0 \quad (9) $$

$$ \frac{\partial n p c_w + (1-n) c_e T}{\partial t} - \nabla \cdot (K_m \nabla T) + \rho c_q q \nabla T - Q = 0 \quad (10) $$

Where $n$ is porosity, $\rho$ and $\rho_e$ are density of water and rock, $k_p$ is permeability, $\mu$ is the viscosity of water, $g$ is the gravity, $z$ is elevation, $Q$ is the source term, $c_w$ and $c_e$ are the heat capacities of water and rock, $T$ is temperature, $K_m$ is thermal conductivity, and $g$ is the Darcy flux term.

Assuming linear elasticity for stress-strain relationship and adopting finite small strain formulation, we get the governing equation for displacement:

$$ \frac{\partial^2 u}{\partial t^2} - B^T DBu - \rho g - \alpha \nabla p - \beta K \nabla T = 0 \quad (11) $$

Where $u$ is the displacement vector, $\alpha$ is the biot effective stress coefficient, $\beta$ is the thermal expansion coefficient and
Where $E$ is the Young’s Modulus and $\nu$ is the Poisson’s ratio for the rock. Detailed notations can be found in our related publications [2, 5].

With the set of governing equations of Eq. 9, 10, and 11, we could form weak formulations of them using Galerkin finite element method. Then one can cast the weak form into a residual function and take the derivative with respect to the primary variables of pressure, temperature, and displacement to obtain approximate Jacobian, which could be used as the preconditioning matrix $P$. It can be further simplified to have only the diagonal blocks (D) as

$$
P \approx \begin{bmatrix}
J_{11} & 0 & \cdots & 0 \\
0 & J_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & J_{NN}
\end{bmatrix}
$$

In our implementation of preconditioned JFNK, the full preconditioning matrix is never actually formed. We have routinely used only the diagonal blocks of Jacobian matrix as the preconditioning matrix and achieved satisfying computational performance.

## 4 Modeling Large Scale Fracture Networks With Adaptive Mesh Refinement

In this section, we discuss the recent implementation of a FracMan® DFN reader and utilization of a mesh adaptivity scheme to model the heat and fluid flow within a large and complex fracture network. This feature improves the simulator’s ability to model realistic field-scale problems, thereby allowing the user to solve and make predictions using a domain that is spatially relevant to their specific application.

![](image1)

**Figure 1.** 2D fracture distribution image (left) and FALCON’s material property assignment map (right).

![](image2)

**Figure 2.** Temperature field (top), permeability (middle), and x-displacement (bottom) for earlier (left panels) and later (right panels) times in thermal stimulation simulation.

### 4.1 FracMan® Discrete Fracture Network

Rarely do isolated single planar fractures exist within a subsurface of geothermal exploration interest. In order to gain a more complete picture of the interactions between spatially approximate and frequently intersecting fractures, FALCON solves for systems of multiple fractures that intersect and affect one another. Within FALCON, DFN’s can be included in the input file in the format of a 2D fracture distribution image or a 3D FracMan®.fab output file. These can be used as the map from which FALCON assigns material properties to the domain.

Once a 2D bitmap fracture distribution image is converted into a list of integer values (0 and 1 for the case in Figure 1 below), FALCON cycles through the list and assigns the values to each of the elements within the mesh. It stores these values within an elemental auxiliary variable, to be called upon as a reference map for the material property assignment of each element. This allows for multiple material properties to be assigned throughout a single mesh block. Sequential elements assigned higher permeability values then become flow pathways for the injected fluid, thereby allowing any complicated or irregular fracture pattern to be modeled. With this method, 2D images of scanned well bores could easily be turned into material property maps to be used in simulations. Alternatively, a systematic study of fracture orientation or intensity within a domain can be carried out using a series of synthetic images.

For modeling large sets of 3D DFN’s, FALCON creates its material property map using input from a FracMan® file. FracMan® software generates large-scale DFN’s based on particular
subsurface characteristics, such as fracture intensity, orientation, and shape/length. The DFN models from FracMan® can be entered into FALCON’s input file as a map for material property assignment, similarly to the 2D image, allowing for the use of a fully customizable, 3D, statistically field realistic fracture network. Once a particular fracture network is described using FracMan®, a full, reservoir scale HTM simulation can be run within FALCON. The potential induced stress-strain effects of a thermal stimulation, the transport times and expected heat recovery between injection and production wells, or the possible far-field impacts of subsurface exploration on groundwater can be modeled using this capability within FALCON.

Figure 3. 3D FracMan® fracture distributions in FALCON with 50 (a), 30 (b), and 10 (c) fractures.

4.2 Adaptive Meshing

MOOSE and FALCON are built upon the libMesh [6] finite element library, which includes parallel adaptive mesh refinement and coarsening schemes. This capability allows for significant savings in computational costs while still capturing areas of interest, like fractures, at high resolution. For each of these schemes, a method of element marking is used to dictate which elements will be refined, coarsened, or remain the same. If an element is marked for refinement, the initial “parent” element is split up into four equally sized “child” elements, thereby increasing the local resolution. For element coarsening to occur, all four “child” elements need to be marked for coarsening, at which point they are combined to once again form their “parent” element.

For efficient modeling of DFN’s within FALCON, a series of three different mesh adaptivity schemes are used. Due to the material property assignment method employed for defining fractures within FALCON, an initial uniform refinement is applied to the whole domain while the material property mapping is taking place in the first time step. This allows for greater detail and a sharper definition of the fractures to be defined. Steps are then taken to reduce the overall number of DOFs with mesh coarsening. This is achieved using a value threshold marker, where elements in areas of no fractures are marked for coarsening and those on the fractures are marked as “do nothing” or to remain at their current refined level. Those elements between these regions take on the size necessary to form a gradient between the two extremes. The number of steps defined by the user dictates how many levels of coarsening an element will go through, up until it reaches the original mesh element size. Lastly, in order to capture changes in temperature, pressure, or displacement, a gradient jump error indicator and error-value-threshold scheme is used. As a temperature front propagates though the domain, the temperature gradients that develop between two adjacent elements increases. As the gradient jump gets sharper across the element boundaries, the potential for error in the solution is greater and a refinement of those elements can reduce this error. The gradient jump error indicator used in FALCON estimates the total error for each element. The error-value-threshold scheme then marks elements for refinement or coarsening based upon a provided maximum/minimum percent error threshold. Those elements that have error values larger than the maximum acceptable error are refined and those with error values lower than the minimum acceptable error are coarsened. The adjustment of these parameters allows for a more or less aggressive adaptive meshing approach.

Figure 4 shows mesh evolution over the course of a simulation. It can be seen that, initially, resources are concentrated in and around the fracture pattern in order to capture their geometry and are sparse in the far-field rock domain. As the simulation progresses, the mesh adapts to follow a temperature front that propagates through the rock, providing higher resolution of the temperature field. This method allows for greater detail to be captured in areas of interest, like rapid fluid and heat flow through the individual fractures, while saving significant computational costs normally expended on areas of little activity.

Figure 4. Evolution of adaptive meshing from initial (left) to final (right) time step.

4.3 Adaptive Time Stepping

Within FALCON and MOOSE, several adaptive time stepping schemes are available in order to increase the time step magnitude over the course of solving the problem. The use of adaptive stepping can significantly decrease the amount of time a simulation takes to complete. If relatively slow physics are taking place or the solution is at near steady-state, large time steps can be taken to speed up the solve time, yet if an acute/rapid event occurs, the time step can be scaled back in order to better capture the event. For this paper, we demonstrate a simple percent increase time stepping algorithm. An adjustable percent parameter is set to dictate how much increase to apply to the time step size at each successive step. This increase will be applied to each step unless the wall time, the actual waiting around time, increases significantly in order to complete the solve, at which point the magnitude of the increase is scaled back.
5 Example THM Simulation With Static and Adaptive Mesh Refinement

In this section, we present an example coupled THM problem within a 2D fractured domain. In Section 5.2, we apply three meshing schemes, static, conservative AMR and aggressive AMR to the problem. In section 5.3, we apply three time stepping schemes, static, a 10% increase, and a 20% increase to the time step.

5.1 Set-up of THM Stimulation With an Irregular Fracture Network

In this example, a low temperature fluid is allowed to flow through a series of intersecting, orthogonal fractures contained within a rock domain initially at high temperature. In this case, a predefined fracture distribution was provided using the 2D DFN image reader.

The two-dimensional domain is 480m by 160m. The initial domain temperature is 140°C and a Dirichlet boundary condition of 60°C is applied to the left side of the domain for the injected cold water. A small pressure gradient is applied across the domain to create fluid flow through the fractures. Permeability is assigned to be 1.0E-17 m² to the surrounding rock matrix and 12.4E-11 m² to the fractures. The prescribed fracture permeability corresponds, based on the cubic law for fracture transmissivity, to a fracture aperture of 0.017 mm. A schematic of the conceptual model is shown in Figure 5.

5.2 Adaptive Meshing

The problem presented in 5.1 was run with three different meshing schemes, (1) static mesh, (2) conservative AMR, and (3) aggressive AMR. The static mesh had a grid resolution of 1m², 76800 elements, 4 non-linear implicit variables, and 15 explicit variables, totaling ~1.46e6 DoFs at run time. As was discussed in the previous section, the error-estimating mesh adaption method employed in this simulation has adjustable parameters that dictate the mesh refinement/coarsening based upon the element’s error exceeding a threshold value. In the conservative AMR case, those elements with errors exceeding 1% were marked for refinement and coarsening, respectively. In Figure 6, the three cases are compared at an earlier and later simulation time.

5.3 Adaptive Time Stepping

The problem presented in 5.1 was also run with three different time stepping schemes, (1) static/fixed time steps, (2) 10% increased step size, and (3) 20% increased step size. All three simulations used an initial time step size of 1000s; the static time steps remained at this size, while the adaptive stepping increased with each step. An exact 10 or 20% increase in time stepping was not seen in either of the adaptive time cases, meaning an increase in wall time was seen which resulted in a recalculation of the time step increase.

6 Results and Conclusions

Combinations of the three AMR cases and the three time stepping cases were simulated for the example problem described in Section 5.1. A matrix of results for % DoFs and time saved are presented in Table 1 below.

<table>
<thead>
<tr>
<th>Static Mesh</th>
<th>Aggr. Mesh</th>
<th>Conserv. Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.0%</td>
<td>19.3%</td>
</tr>
<tr>
<td>DoFs</td>
<td>0.0%</td>
<td>40.3%</td>
</tr>
<tr>
<td>10% Step Inc.</td>
<td>Time</td>
<td>79.9%</td>
</tr>
<tr>
<td>DoFs</td>
<td>0.0%</td>
<td>40.5%</td>
</tr>
<tr>
<td>20% Step Inc.</td>
<td>Time</td>
<td>87.4%</td>
</tr>
<tr>
<td>DoFs</td>
<td>0.0%</td>
<td>40.4%</td>
</tr>
</tbody>
</table>

In the aggressive and conservative AMR cases, an average of 40% and 43% reduction in DoFs calculated was seen as compared to the static mesh simulations. Figure 7 shows the run-time DoFs for each of the three meshing schemes in combination with a static time step. Initially, both AMR schemes have equal DoFs to that of the static mesh, but quickly decline within the first couple of steps. As the temperature front propagates through the fracture...
distribution and diffuses into the surrounding rock, the number of DoFs increase in order to effectively capture the solution.

For the 10 and 20% increased time step simulations, solve times were decreased 80-94%. As one would expect, the conservative AMR - 20% time step adaptivity combination saw the largest decrease in solve time. However, the computational savings in time and DOFs due to using these AMR/time stepping schemes did incur some margin of error. Figure 8 shows the errors for each combination of mesh adaption and time step adaption in comparison to the static mesh/static time step case. As would be expected, the maximum error, 6.2e-3%, was seen with the conservative AMR - 20% time step adaptivity combination. It is interesting to note that the percentage of time step adaptivity had a higher influence on solution accuracy than mesh adaptivity, as can be seen by the three groupings of error lines.

Significant performance gains for large-scale simulations can be seen when employing AMR and times step adaptivity in combination with the FracMan® or 2D DFN readers within FALCON. For a given 2D DFN, DoFs and computations times were reduced by up to 43% and 94% respectively when using aggressive and conservative AMR schemes and two different % increase time step adaptions as compared to a static mesh – static time step simulation. A quantitative comparison of the predicted temperature field at 30m into the domain showed that a maximum error of only 6.2e-3% was seen due to the use of these adaptivity schemes. Further research is underway to determine the ideal combination of AMR and time step adaptation parameter values to obtain a maximum performance gain and minimum in solution error for various fracture distributions.

7 References