**Geothermal Design Tool (GeoDT) (GTDT)**

# Overview:

Advanced modelling capabilities are now available to evaluate hydraulic fracture stimulation, fracture network flow, subsurface fluid-rock heat and mass transfer, multi-well dynamics, geomechanical feedbacks, geochemical feedbacks, and long-term power or hydrocarbon generation. While all these components should be considered together in the optimized design of wells and well fields, no models are yet known to us that couple all these processes efficiently and intuitively enough to aid timely design and decision making in active projects. Our aim here is not to develop a new model to add to the existing collection of high-fidelity and high-performance codes. Instead, out goal is to develop a tool that can be used to aid decision making in the early stages of a new field development when uncertainties are very high and data is sparse to non-existent. To achieve this goal, we focus on controllable parameters (e.g., well placement and injection rate) and site parameters that can be effectively characterized by common and low-cost methods.

# Workflow:

The workflow of this tool begins with model parameterization and ends with an estimate of flow and enthalpy inputs and outputs. The general workflow is depicted in Fig. 1.

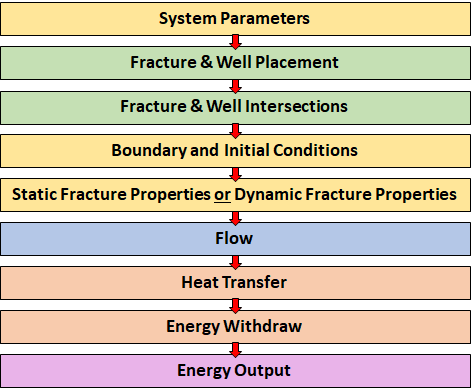


Fig. 1. General model workflow to calculate flow and heat transfer through multi-borehole leaky fracture networks.

As arranged in the modeling code, the sequence is broken down into a set of key functions that can be sequenced as desired for the problem to be solved (Fig. 2). For example, natural fracture generation can include three uses of the same function in order to build three families of fractures in a conjugate joint set. Also, the script can be set in a modular nature so as to keep the same fracture network when stochastic variability in the fractures is not desired as one investigates an optimal well spacing.

In Fig. 2, the line items marked with a star currently lack some desired functionality as follows:

1. The hydraulic stimulation module currently assumes a fracture radius, fracture orientation, and stimulation location. It would be better if the stimulation model considered hydraulic fracture interaction with natural fractures as well as cumulative leak-off. To do this quickly, we can use a shortcut that estimates stimulation based on fluid volume balance and injection rate, circumventing the need to model tip stresses and XFEM-type high-fidelity propagation. This shortcut would need to consider stress state and slip criteria for the natural fractures as well as dilation generating storativity.
2. Fracture property assignments are currently in the wrong place for this code. Originally, we thought that transient stress-dependent aperture would be good to include in this model. However, this including causes a significant numerical instability which will cause implicit solvers to fail or give impossible outcomes. Now knowing this issue, the code would be more efficient by statically defining fracture properties when they are created and/or stimulated.
3. The flow solver currently only considers fracture network flow. It does not consider transient effects such chemical infilling. Stress dependent effects were included, but found not to work. To make the model more applicable to more cases, we need a matrix-to-fracture source term. Luckily, the physics are very similar to how the heat transfer is modeled so we may be able to use that approach. However, a better method would be to couple a pressure dependent source term into each fracture that is computed by an updated flow solver.
4. The general structure of the code needs to be improved. The current version underutilizes classes to define the fracture and well properties. As it is, there is a strong reliance on correct use of index numbers in large arrays that store fracture properties. This is efficient for math, but poor for debugging and poor for clean modular applications of the code.

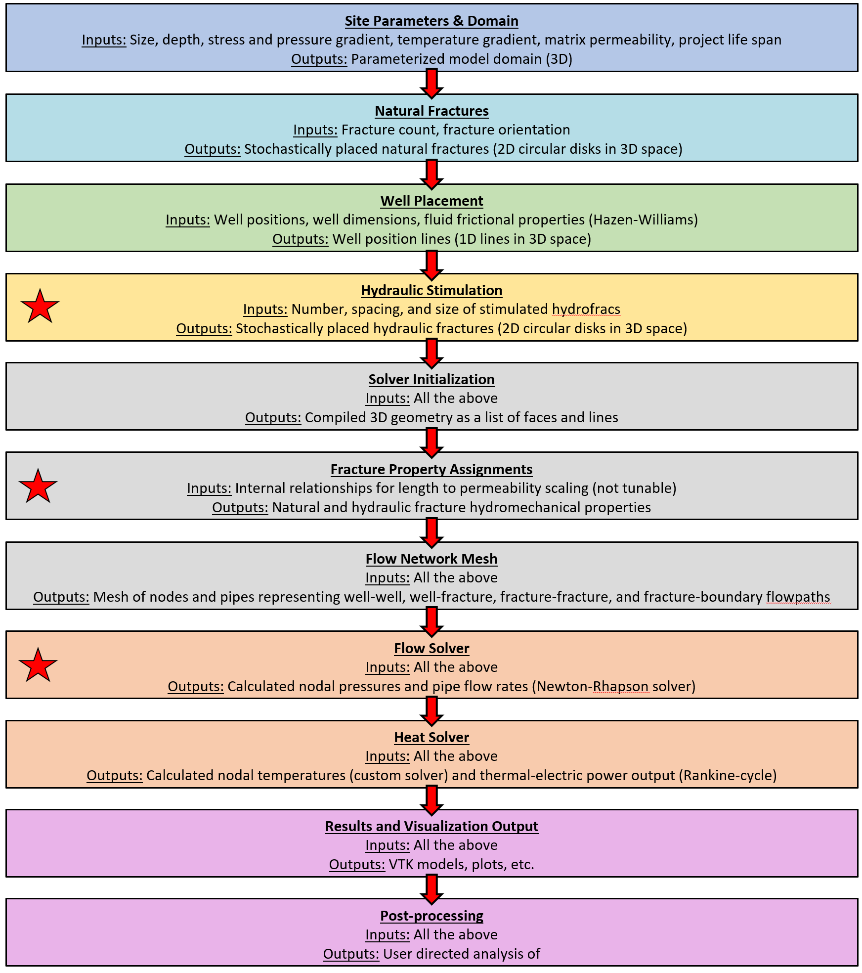


Fig. 2. More detailed breakdown of functional workflow in the code. Details on what each function is and how it works is not provided here. Red stars indicate high-priority focus areas for development.

# Numerical Methods:

## Fracture and Well Placement

Wells are placed by the user based on what is planned for a given site. The specified well location is open-hole (no-casing) by default, but cased wells can also be modeled. Fractures are 2D circles in 3D space. The center position, length, strike, and dip of these fractures can be specified by and desired distribution. For example, fractures originating from perforated zones along the well can be simulated and natural fractures can be simulated. The size of the model domain must also be specified. Fractures intersect the domain’s boundary are able to leak off to or be produced from the far-field.

## Fracture and Well Intersections

To speed-up analysis and improve ease of visualizing and understanding flow networks, we model a 3D fracture and well system using a network of pipes and nodes. Pipes represent flow channels through fractures or wells. Nodes represent well-fracture, fracture-fracture, well-boundary, and fracture-boundary intersections. Every fracture that connects to the stimulated fracture network includes one node at its center point. This fracture-center node is used to calculate fracture pressure for slip criteria. Also, using a common node prevents unrealistic crossing flow paths with unequal crossing pressures from being modeled. The intersection between two planes is set at the midpoint of the intersected chord that is common to both fractures (Fig. 3).

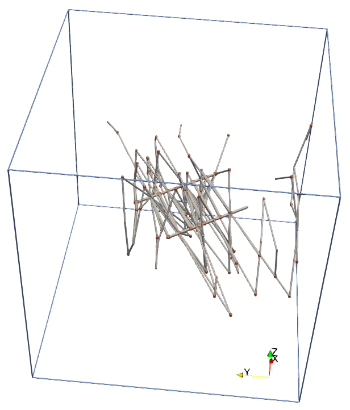
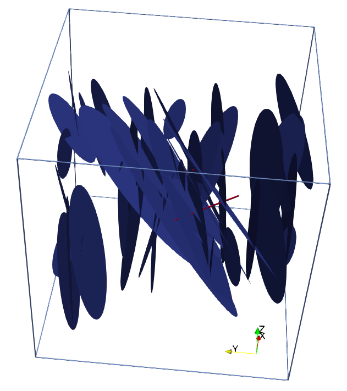
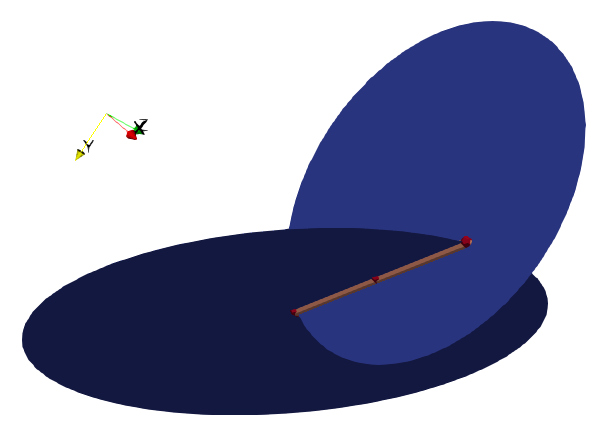


Fig. 3. (Left) Intersection line of two planes. The midpoint of this line becomes a node in the pipe flow network. The length of the intersection line is used as the characteristic width of the fracture flow path. (Center) 3D fracture network and two wells. (Right) Pipe and node flow network for this same fracture network with fractures intercepting the back and right faces of the cubical domain boundaries.

Intersections are sequentially identified starting from the wells, such that they can be thought of as a first, second, third, etc. set in a series. The intersection algorithm will ignore isolated fractures and will not repeat opposite direction pipes between the same nodes. This may sound like an obvious description, but it is worth mentioning here because finding an algorithm that does this successfully is not straightforward. The sequential identification of pipes and nodes is also key to the flow mesh generation where pipes are required to have an assumed direction for positive vs negative flow.

To represent near-well flow constrictions and leaky boundaries, these intersections must have special handling (Fig. 4). A well intersection with a fracture has a small line of contact that that will constrict flow to and from the well. This zone is often referred to as the “near well zone” where tortuosity and this flow constriction to cause large pressure losses during flow. In this model, the width of this ‘choke’ is set at the circumference of the well (π\*Dw) and the length is set at triple the diameter (3\*Dw). Beyond the choke, the width of the flow path is the diameter of the fracture. This simplification will not be equal to high-fidelity models but it will simulate the important choke effect near to the fracture intersection with a well. The next special case is a fracture intersection with the domain boundary. Here a far field connectivity flow path is added instead of the link from the intersection midpoint to the center of the boundary’s face. The length of this flow line is set at 100x the size of the model domain, based on the assumption that this flow path is constricted by roughness and other flow friction effects. We feel this is more appropriate than a short or free-flowing path that would act as a free-draining leak which could give rise to higher flows than are likely to be encountered in practice. Here the aperture of the choke and fracture paths is taken from the aperture value of the fracture that is intersected.



Fig. 4. Pipe and fracture intersections converted to a pipe and node mesh. The pipe and fracture intersection adds a ‘choke’ segment in the flow path. The fracture (blue) intersection with the boundary (brown) is a special case that extends the flow path to the far-field (100x domain size).

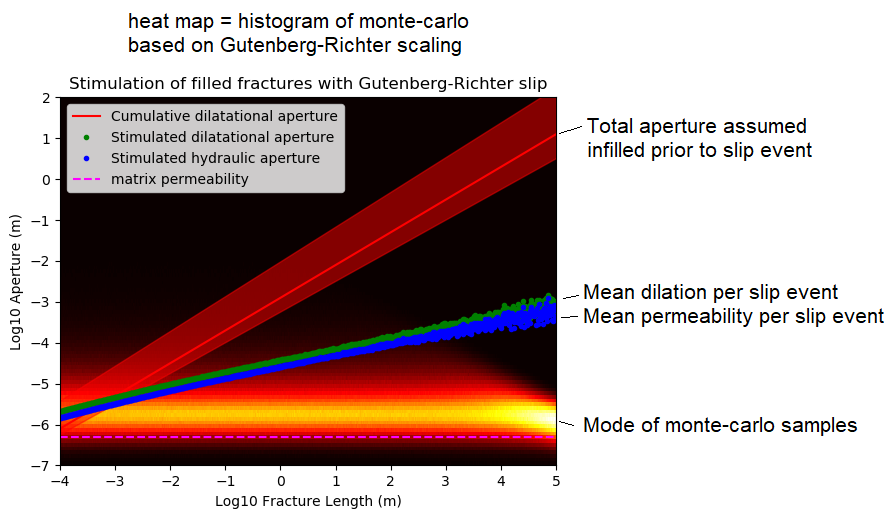
## Boundary and Initial Conditions

The boundary nodes are set at a constant pressure condition. This permits far field fluid flow to simulate leaky fracture networks. The initial pressure and temperature conditions are set based on provided site parameters. Injection and production wells can be specified with either a constant pressure or constant flow rate boundary condition. Typically, injection well flow rate is assigned and a production well pressure is assigned. This choice provides good model stability and aids the characterization and understanding of the complete well and fracture system.

## Pipe and Fracture Properties

Fluid flow through fractured rock is complicated. A solution is needed that provides useful model predictions for decision making despite this complexity. In other words, model speed and simplicity is more important than accuracy when the data required for higher fidelity models is not available. To start, water is the typical fluid of interest for subsurface flow and transport. Therefore, this will be the working fluid for our model. For pipe flow, the Hazen-Williams equations enables quick estimation of water pipe fictional losses. For fracture flow, frictional losses can be estimated using the effective cubic law. This differs from the ideal parallel plate law by including aperture reduction factor to account for non-ideal geometry. It is rare to have information about actual fracture permeability, aperture, length, or location in the field, but these terms are crucial to connectivity and transport. To address this, we can use stochastic fracture placement and aperture-length scaling relationships that estimate the range of potential fracture properties that could occur in rock. In addition, we seek a numerical model that is solvable. As it turns out, a solvable fracture flow network that uses nodal pressures as the state variable to estimate flow rates is subject to numerical limits at large and small aperture limits. For systems in the 1 to 10000 m scale, these limits for effective hydraulic aperture are typically in the range of 0.05 mm to 5.0 mm. Fractures with apertures smaller than this range will have near-zero flow at large pressure gradients. Fractures with larger apertures will have such excessively high permeability that machine-precision becomes an issue when estimating flow from pressure differences. Owing to this limitation, the model currently assigns all fractures an effective hydraulic aperture of 0.0002 m (0.2 mm). This value achieves detectable pressure differences across fracture segments and treats tight fractures as negligible features for flow at the macro-scale. Fracture density is also assigned a value less than what one would observe in core logs to account for unstimulated or impermeable fractures that are observable but which do not contribute significantly to flow.

### Fracture length-aperture scaling relationship and stress dependency (Frash et al., 2020):



## Flow Solver

In this model, we solve for flow through the pipe and node network directly. To do this, we use the Newton-Raphson iterative solver (Jeppson, 1974) with some modifications to make the solution more universally applicable to fractures and pipes. Frictional losses in pipes are modeled using the Hazen-Williams equations for turbulent flow. Frictional losses in fractures are modeled using cubic law based on the effective hydraulic aperture () instead of the mechanical aperture (). This is an important distinction because mechanically large fracture apertures can have negligible hydraulic apertures due roughness effects. For this solution, node pressure is the state variable being solved.

### Hazen-Williams Flow Equations:

(H.1)

(H.2)

(H.3)

Where, is pipe or fracture flow-path length, is pipe inner diameter, is the Hazen-Williams friction coefficient (e.g., 130 for welded steel pipe or 80 for corroded steel pipe), is volumetric flow rate of liquid water, is frictional pressure head loss, is lengthwise pressure gradient, is fluid density, is fluid dynamic viscosity, is effective hydraulic aperture, is fracture width (not length, not aperture), and is the gravitational constant.

### Newton-Raphson Iterative Solver:

(J.1)

(J.2)

(J.3)

(J.4)

(J.5)

(J.6)

(J.7)

(J.8) (J.9)

Where, and are node numbers corresponding to pipe ends, is an iteration counter, is an initialization applicable to the first iteration only, is the initial formation pore pressure head, is nodal pressure head, is a boundary condition flow rate that is positive for flow out of the node, and is an overshoot damping factor that takes the value of 0.7 to achieve faster convergence. The iterative loop repeats J.2 through J.8 until the absolute maximum of the error term () is less than a goal value. The implementation of this code was validated against Jeppson’s Chapter 6 Examples #1 and #2 (1974). This solution produced the same result as the examples, but with less iterations and reduced user effort. The key solver differences were J.2 verses J.1 and the randomizer in J.6. We also validated the fracture flow equation implementation into this solver using a simple structured geometry (Fig. 5) and found the solver to produce a correct result.

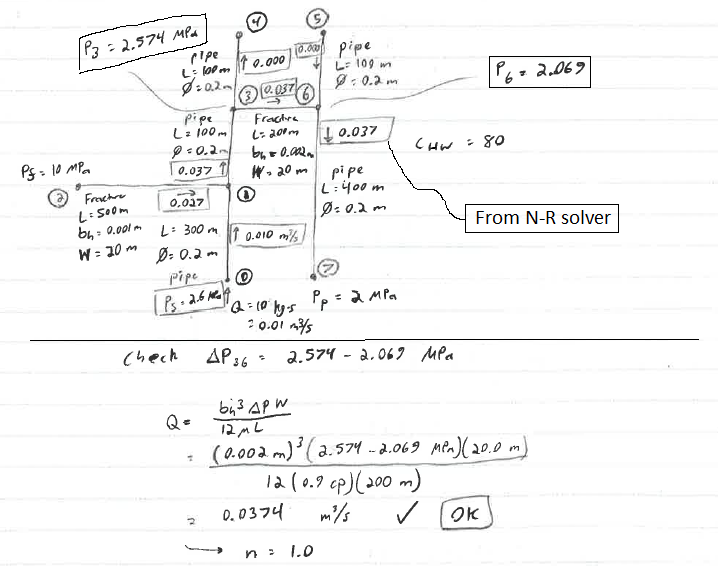


Fig. 5. Notebook test case to validate flow solver with fractures. This case includes two wells and connection to a constant pore pressure boundary.

## Fracture Stimulation Solver

Flow through a multi-well discrete fracture network involves many subtle complexities that arise from coupled hydro-mechanical processes. Dependent upon the injection rate, stress states, and network geometry, this flow can exist in four different general categories.

1. Leakoff / Diffusion Dominated: Scenarios with flow rates that are fully accommodated by the transmissivity of the existing fracture network without inducing fracture slip. This state is stable with decreasing flow rates being proportional to decreasing pressure gradients.
2. Shear Stimulation: Scenarios where the injection rate and pressure is high enough to cause natural fracture pressures to reach a critical value for slip. This state is unstable in the short term but can become stable as stimulation events cause permanent increased fracture transmissivity that may eventually become a leakoff dominated system.
3. Hydraulic Jacking and Fracking: Scenarios where the injection rate and pressure is so high that new tensile fractures can form. Open fractures have extremely high transmissivity compared to closed fractures. This state is highly unstable because high pressure is required to maintain the high transmissivity and the high transmissivity causes frictional losses to be very low. As long as the tensile fractures remain open, flow rate can take nearly any value while the pressure of injection will remain relatively constant, typically at less than 2MPa net fracture pressure.
4. Dynamic Stimulation: Scenarios with very high-rate loading (e.g., shock) introduce mechanics that cannot be solved using conventional static-equilibrium methods. This will induce fracture patterns and rock damage via dynamic loading-unloading criteria, often producing rubble in the near field but minimal permanent displacement further from the working point. WE do not address this scenario in our model, but it is nonetheless important to acknowledge.

We seek a fast method to evaluate 3D fracture stimulation that can achieve reasonable solutions and also accommodate the non-unique flow-pressure relationship that occurs during hydraulic jacking and fracking. To achieve this, we employ a volume-based approach based on conservation of mass (Fig. S1). This approach starts with calculating the injection rate and system pressures at a constant injection pressure just below the hydraulic fracture threshold. If the calculated injection rate exceeds the target injection rate, the existing fracture system will be leakoff dominated. Otherwise, intersected slip-critical fractures must stimulate or the injection pressure will increase. Hydraulic fractures are created randomly or at perforation locations if no shear fractures exist that are prone to stimulation at the current injection pressure. The new fracture growth and pressure sensitive dilation will cause the fracture volumes to change, a process that can be characterized as fracture storativity. This filling process takes time and volume, detracting from the scheduled total injection volume. If this change in volume, combined with the concurrent steady leakoff is sufficient to deplete the remaining injection volume, no more stimulation occurs. Otherwise, the process repeats until the scheduled injection volume is depleted or overcome by the leakage and production rates.

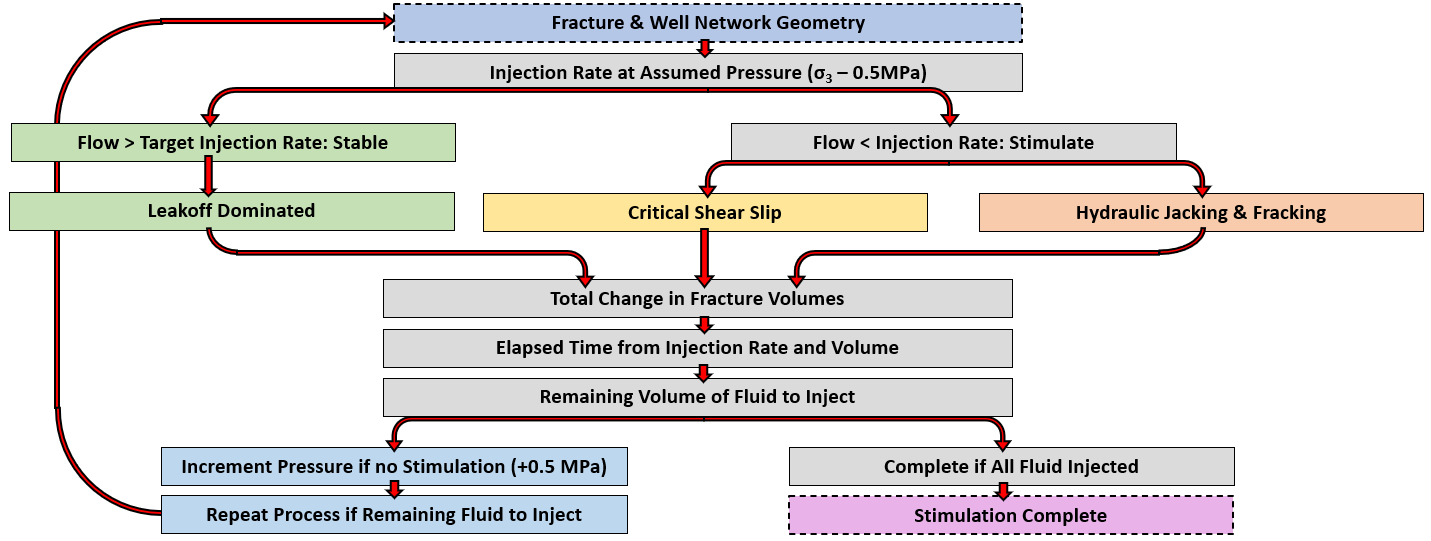
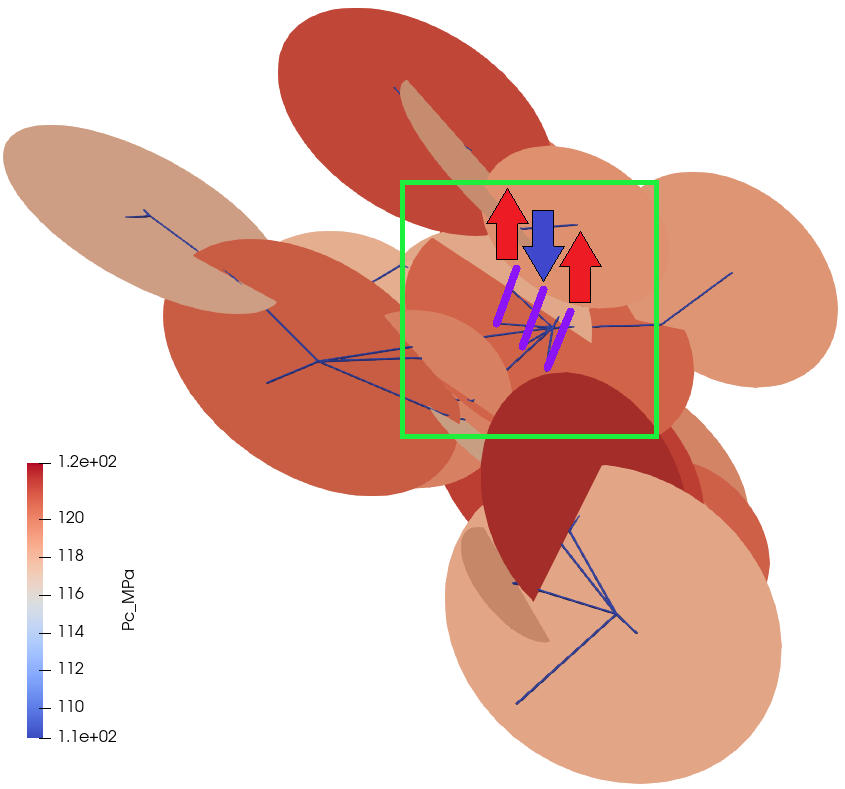
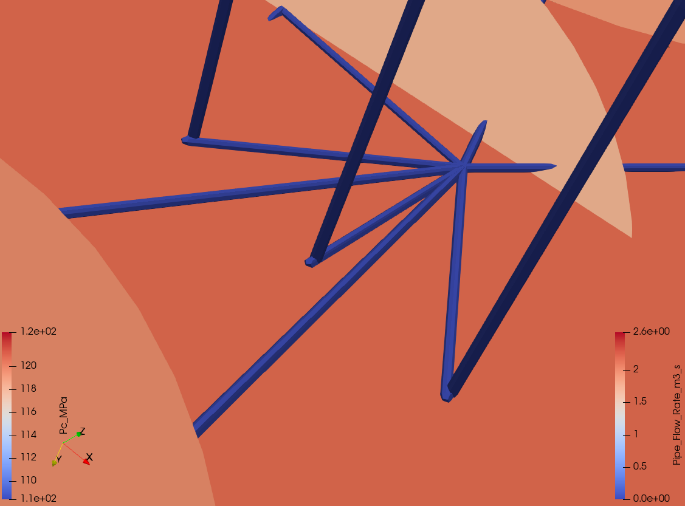


Fig. S1. Fracture stimulation workflow that accommodates leakoff dominated flow, shear stimulation, and hydraulic fracturing. The solution is iterated until the targeted injection volume is depleted by the combined effect of leakoff, production, and fracture volume changes.

To account for changing flow paths during fracture growth, the hydraulic fractures and shear fractures extended in a stepwise manner (Fig. S2). The increment of shear growth is estimated by our scaling relationships between fracture length, shear displacement, and associated seismic slip magnitude. Hydraulic fracture growth is propagated by increasing the fracture size by a factor in each iteration loop. We find that a factor of 0.15 size increase in each step allows for consideration of fracture growth effects without making the simulation process too slow or unrealistic.

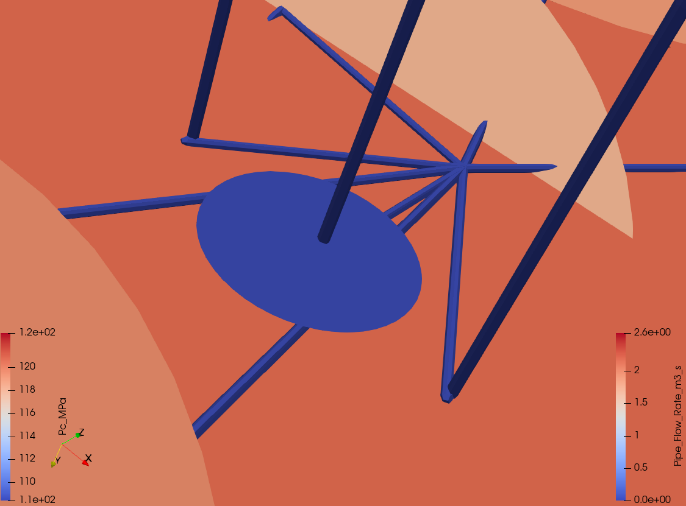
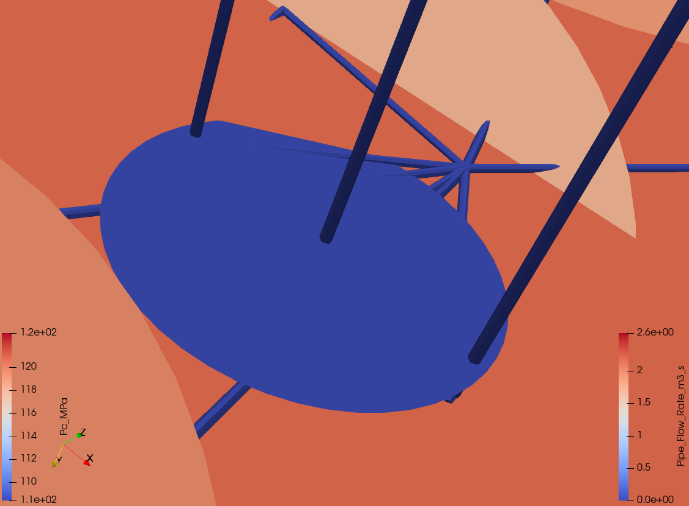
 

Fig. S.2. Hydraulic fracture propagation when natural fractures are too tight, strong, and unfavorably oriented for hydraulic shearing. The hydraulic fracture initiates near a natural fracture from the center injection well (blue arrow). When the hydraulic fracture intersects the two production wells, the sudden increase in leakoff causes the propagation to halt because the production wells drain the fracture faster than the injection rate can resupply fluid. Fracture colors are shaded by the critical pore pressure required for stimulation (blue for low pressure, red for high pressure). These critical pressures are a function of the stress state and fracture orientation.

Our solution approach is able to simulate the instability caused by hydraulic jacking, where fracture permeabilities can suddenly transition from very low to extremely high over an injection pressure change as small as 0.5 MPa (Fig. S3). Here the in-situ pore pressure is 55 MPa. Before hydraulic fracture stimulation, an injection pressure of 108.3 MPa produces a flow rate of 0.0012 m3/s via closed natural fracture leakoff. After hydraulic fracture stimulation, an injection pressure of 108.8 MPa achieves a flow rate anywhere between 0.001 m3/s and 2.2 m3/s. The extreme pressure sensitivity of hydraulic fracture permeability combined with it very high permeability necessitates special handling of this flow condition. If system is hydropropped, we will use the calculated pressure that is required for hydraulic fracturing and then set a suitable long term injection rate, based on the flow solution at this high pressure. Furthermore, hydraulically jacked fractures have a potential to continuously stimulate if they do not intercept sufficiently high-capacity flow-sinks (e.g., wells and faults). This model is able to simulate this process via its volume based approach combined with the calculated pressures from the flow solver.

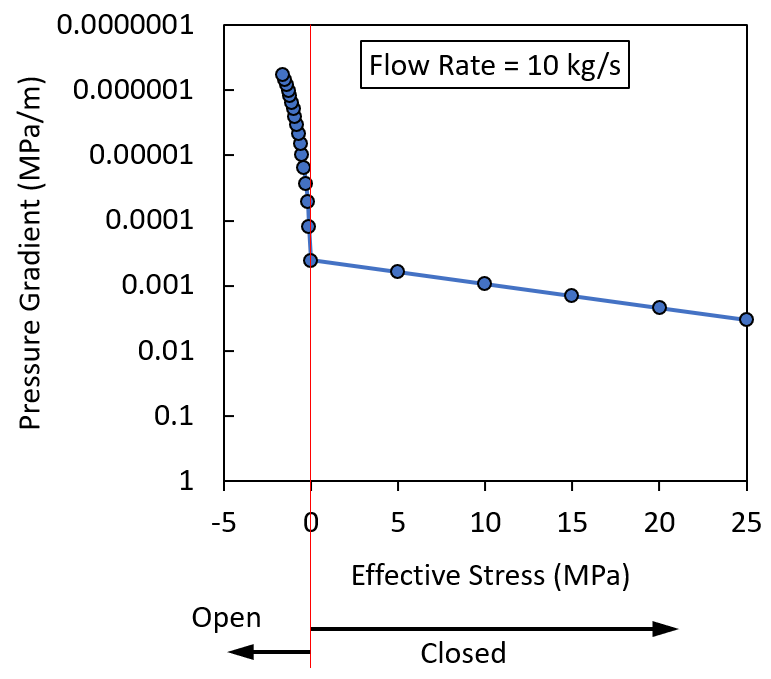
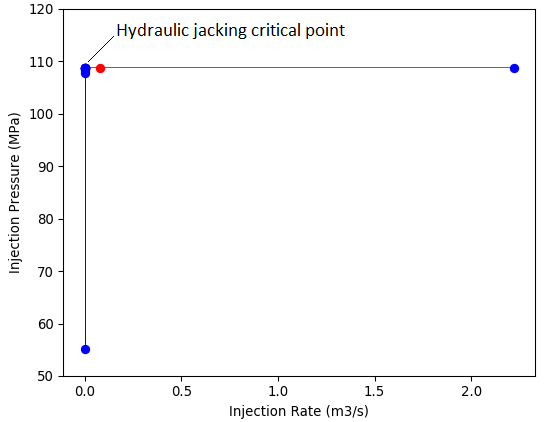
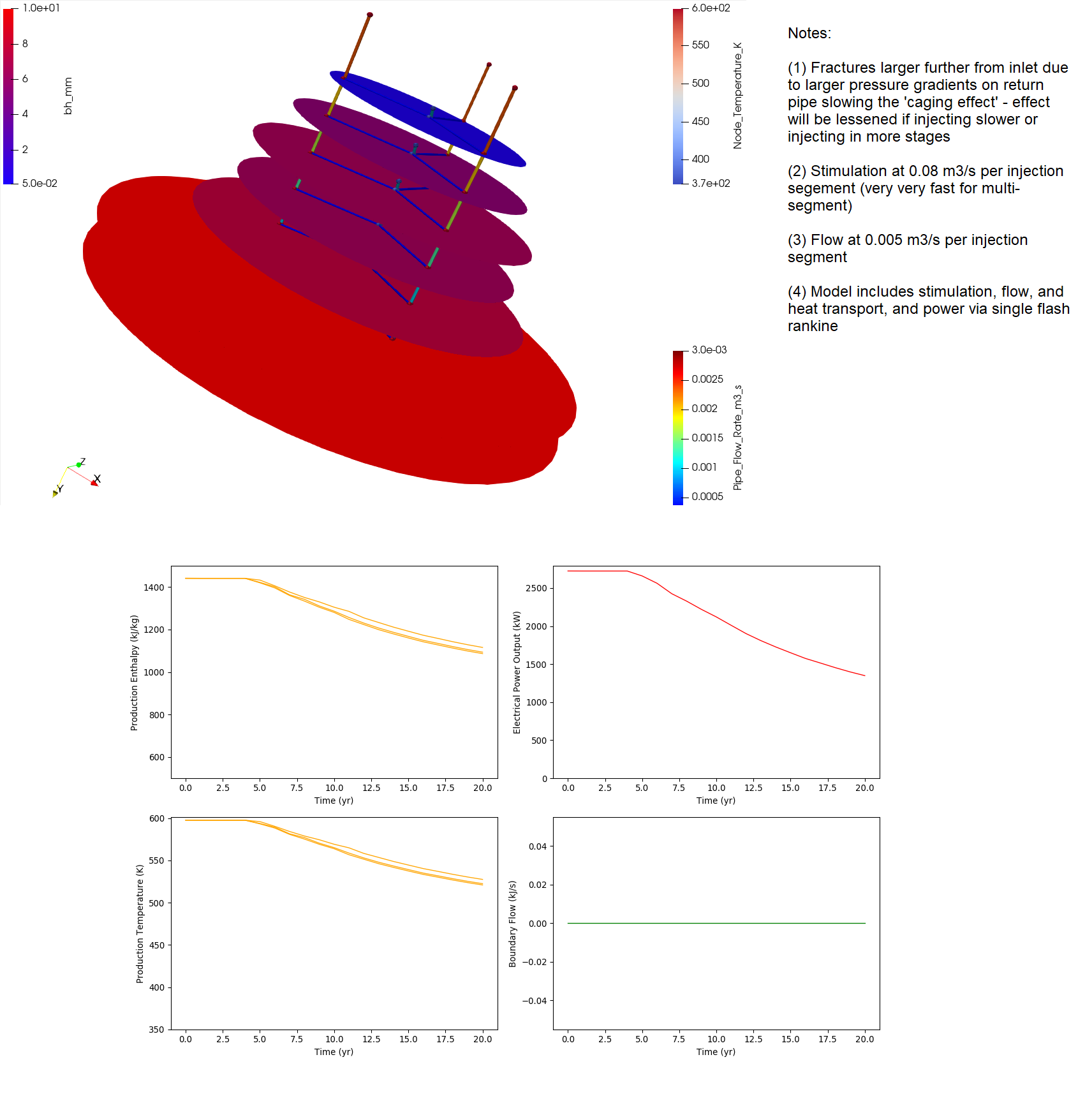


Fig. S3. (Left) DFN modeled instability of pressure and flow rate at the onset of tensile hydraulic jacking and fracking. (Right) Theoretical relationship between pressure gradient (i.e., inverse permeability) and effective closure stress fracture opening on a fixed 0.72 m width and 0.68 m length fracture showing the discontinuity at the transition from tension to compression. For closed fractures, the pressure and flow rate relationship is uniquely determinable. With open fractures, a minimum injection pressure can be estimated but the flow rate is not uniquely estimable due to its dependence on full system geometry. However, we can estimate the minimum injection rate required to sustain the open fracture.



## Heat Transfer Solver

Typically, heat transfer for complex geometries will use a 2D or 3D meshed method such as finite differences. However, we want a fast desktop-compatible solution that can support <1 mm aperture fractures, >1 km scale reservoirs, and predict results for >20 yr of operation. In addition, we want a solution requiring only one processor that completes a start-to-finish (Fig. 1) field-scale model in less than 5 min. While achieving this, we also want to avoid the challenges of mesh refinement and large element counts that arise when modelling fractures in a continuum. Thanks to the high uncertainty of fracture dominated flow through hot rock, we can reasonably accept higher than conventional errors for our solver. Based on this, we now propose a solution that achieves our fast model goal. We hope that future benchmarking against a high-fidelity model will show that this fast solver is close enough to an ‘accurate’ solution that it is useful for decision making.

Our heat transfer solver uses an explicit time marching scheme with an embedded iterative nodal temperature solver for each time step (Fig. 6). This provides a transient solution for fluid-rock heat transfer that is capable of predicting thermal breakthrough, short circuiting, and long term produced fluid enthalpy. These predictions address the foremost concerns for geothermal site development.

Node temperature initialization assigns the initial conditions for nodal temperatures as appropriate for their depth at the site. This step also assigns the far-field reservoir temperature for these nodes as required for transient heat transfer analysis.

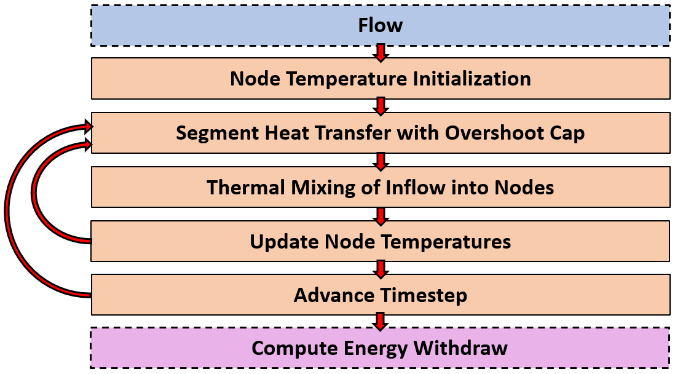


Fig. 6. Workflow of the heat solver in our model. This uses inputs from the flow solution.

Heat transfer at a given timestep is solved for each segment (Fig. 7) and node in the flow network. To attain a transient solution without a finite-element approach, we use that we call a ‘pseudo-steady/pseudo-transient 1D heat transfer’ approach (Fig. 8). For this we combine:

1. An initial guess for a ‘thermal radius’ ( or ) where the rock temperature is not yet significantly changed from the initial state at the start of the timestep.
2. A 1D steady-state heat transfer approximation for a round pipe or planar fracture based on this thermal radius and the mean of the nodal temperatures that define the segment.
3. A forecast of the total energy exchange to/from the rock adjacent to this segment over the explicit timestep of the model.
4. An updated thermal radius based on the new temperature gradient and the total energy exchange from the rock.

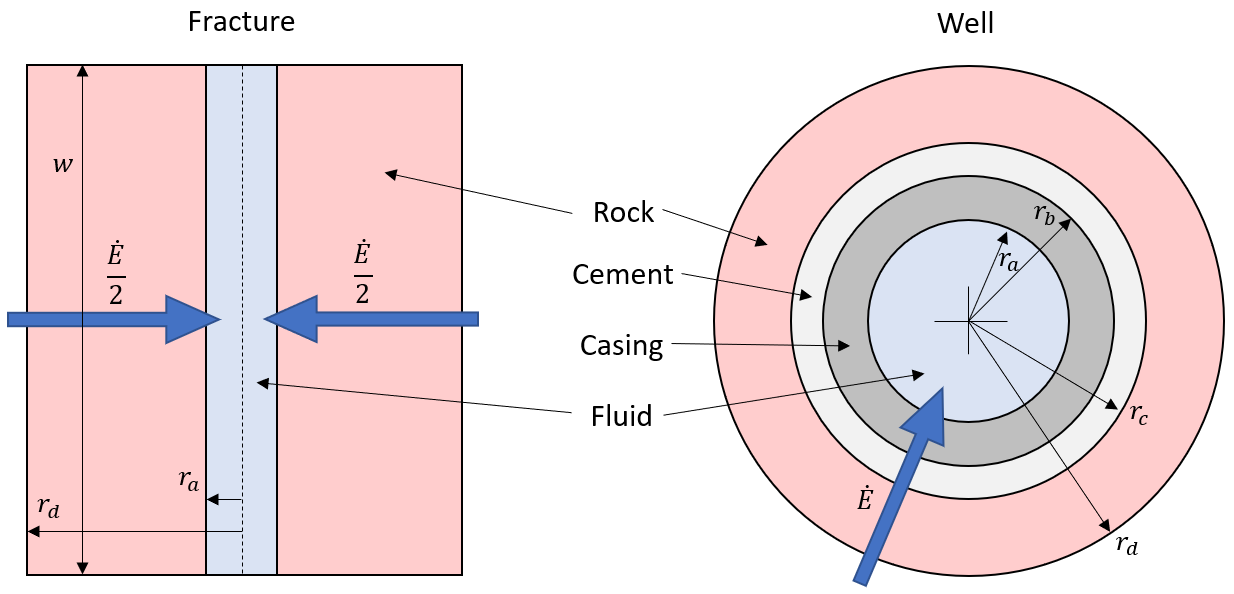


Fig. 7. Heat transfer geometry for a fracture and a cased well.

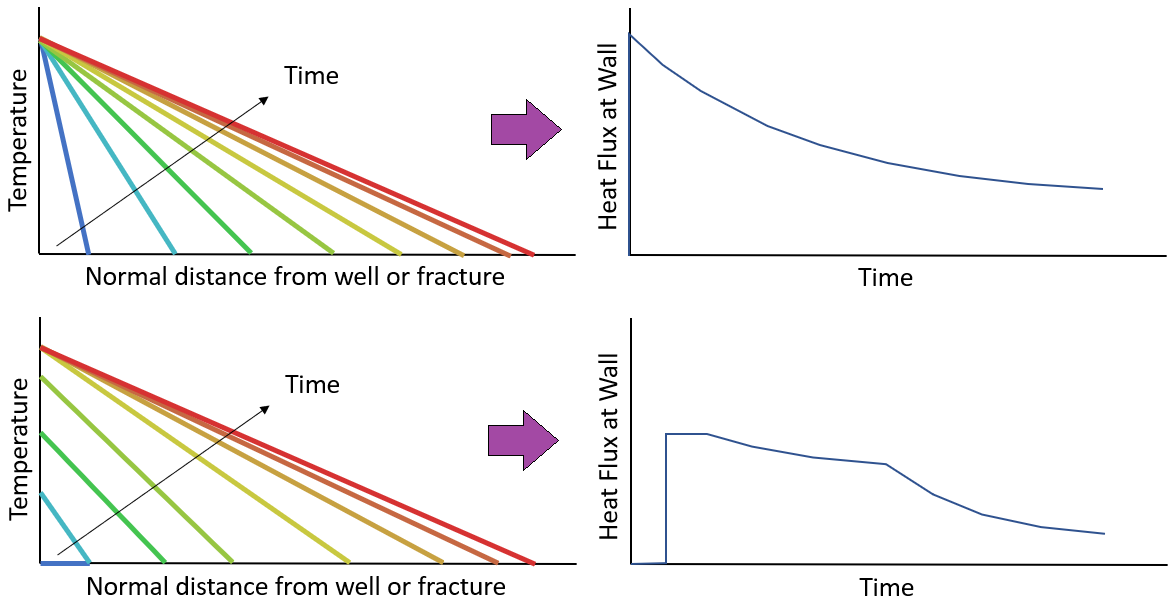


Fig. 8. Scenarios we aim to solve with pseudo-steady/pseudo-transient heat flow. (Top) A sudden increase or decrease in rock temperature at a fracture face that results in rapid heat transfer at first, but this rate of transfer diminishes over time. (Bottom) A slow increase or decrease in rock temperature at a fracture face to an equilibrated value so that heat transfer is initially negligible, then accelerates, and finally decelerates.

Evaluation of heat transfer in the complete network of wells and fractures necessitates mixing flows and dividing flows at intersection nodes (Fig. 9). To do this, we use conservation of energy and conservation of mass to mix the inflows into a node and then obtain a mixed enthalpy that is used for the outflow. The mixed enthalpy determines the temperature at the node. This approach enables modelling the effect of sensible heating for low-quality steam. Relations for the temperature, enthalpy, pressure and steam quality for water in this method are obtained from IAPWS97 (International Association for the Properties of Water and Steam, 1997).

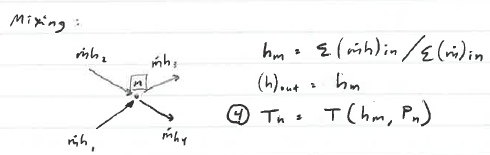


Fig. 9. Nodal thermal fluid enthalpy mixing.

To iteratively solve for nodal temperatures, the mixed nodal temperatures replace the initial temperature guesses and the model is repeated until the temperature changes between iterations become less than 0.5 °C. After this convergence, a timestep is taken.

### Thermal Radius and Energy Storage (reference Fig. 7):

For 1D steady-state heat transfer through a series of materials,

(E.1)

(E.2)

Where, is heat transfer rate, is the fluid convection coefficient is layer thickness, is thermal conductivity, is cross-sectional area that heat is flowing across, and is the temperature gradient.

In the case of radial heat flow through a cylinder, steady-state heat transfer through one layer is,

(E.3)

Where, is outer radius and is inner radius.

The temperature distribution for the planar (i.e., fracture) and radial (i.e., well) cases will be,

(E.4)

Convection coefficients are not trivial to estimate and can vary significantly. For forced water flow, commonly used reference values are in the nominal range of 1000 to 15000 W/m2K for water (TLV Corp, 2020; Engineers Edge, 2020). To show this, the thermal conductivity term for convection is stated as,

(E.5)

From these fundamental equations, steady-state heat transfer through a cemented and cased well or into a planar rock fracture (Fig. 7) can be stated as,

(E.6)

Next, energy withdraw from the rock can be stated as,

(E.7)

Or alternatively as,

(E.8)

Here we must solve the integral for total energy withdraw per unit of length along the axis of flow. The fractures and wells will have negligible temperature gradients along the length, so axial heat transfer will be neglected. It can also be shown that the cement, casing, and fluid convection terms are negligible for heat transfer in the long term because the heat transfer is most severely limited by the thermal conductivity of the rock, so we can ignore those terms. This gives us the following,

(E.9)

(E.10)

(E.11)

For our solution we will need to solve for both the total energy transfer () with a given ‘thermal radius’ () and for the ‘thermal radius’ for a given total energy transfer. Since the above relationship for a well would be cumbersome to solve for thermal radius, we instead can fit an effectively equivalent function using linear regression (Fig. 10) to quantify the constants ( and ) in,

(E.12)

(E.13)

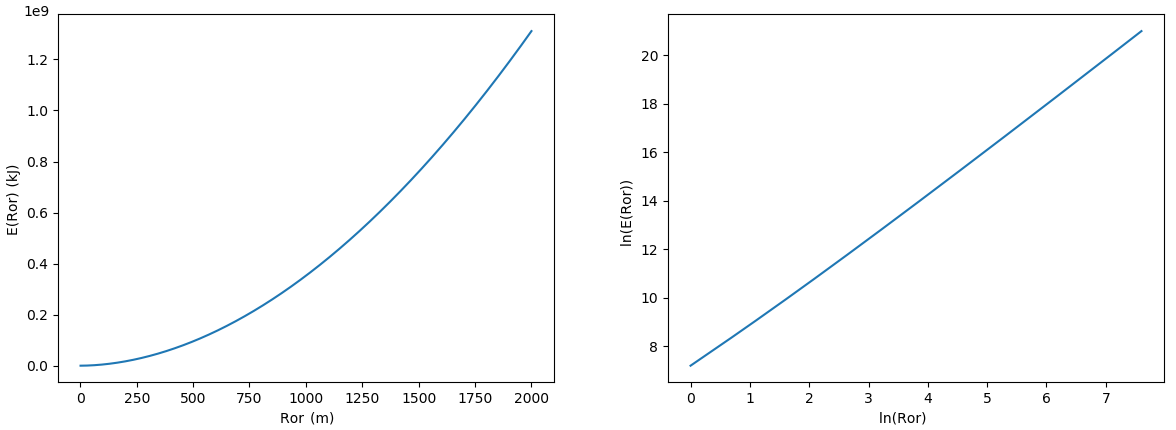


Fig. 10. Relationship between energy withdraw () and thermal radius () for a 152 mm inner diameter borehole (), a volumetric heat capacity () of 2063 kJ/m3K, and a unitary temperature gradient.

From the above relationships, we can now estimate the thermal radius or total energy withdraw for the rock at any time after flow begins through a fractured system. Towards the goal of a fast model, this approach bypasses the discretized 2D or 3D mesh and advanced solvers that would more commonly be used to estimate heat transfer to the fluid in the fractures or well. Even with this simplification, our model can predict transient heat transfer effects by tracking the total energy withdraw from the rock, which is fundamentally a conservation of energy solution. In other words, our model does not require a constant wall (i.e., internal fluid) temperature to function so thermal breakthrough can be predicted.

### Explicit Heat Transfer Solver:

As the thermal radius approaches zero, the thermal conduction will approach infinity. To mitigate this instability, the model starts by assuming thermal withdraw from a small thickness zone around the wells and fractures. For each segment, this is calculated by,

(S.1)

Where, is a user selected initial energy withdraw that yields a stable model output as determined by trial and error. Our nominal value for is 500 kJ/m2 when fracture lengths are nominally around 500 m and timesteps are 1 yr.

Next, this energy withdraw is used to obtain an initial thermal radius for heat conduction as (Eqns. 10 & 13),

(S.2)

Which then yields the initial rate of thermal conduction for the timestep (Eq. E.6),

(S.3)

Next in the timestep, nodal temperatures are computed using the result from the flow model, fluid mixing at nodes, and fluid enthalpy data as dependent on temperature, pressure, and steam quality. This yields information on the heat that can be absorbed by the fluid, which can be less than the amount of heat that the rock can conduct to the fluid at a given temperature gradient. Thus, the energy extracted in a given timestep becomes,

(S.4)

Where, is the energy transferred to or from the fluid in the timestep. The absolute value of this term is used to prevent numerical instability when heat transfer switches from heating to cooling between timesteps, or vice-versa. This new value allows us to update the thermal radius and heat transfer rates for the next timestep by repeating Eqns. (S.2 & S.3). If no heat was transferred on a given segment (e.g.,), the values from the previous iteration will be used again.

### Iterative Nodal Temperature Solver:

As mentioned earlier, the nodal temperatures are initialized at the in-situ equilibrium rock temperature. Overriding this, the inlet, outlet, and/or boundary temperatures can be specified. The enthalpy and other fluid properties at these nodes can then be calculated from steam tables (e.g., IAPWS, 1997). At the end of the timestep, these temperatures are updated by considering heat transfer to and from the fluid and mixing of flow streams into nodes. This requires an iterative process because the initial guess of temperatures will not typically be correct. The first part of this iteration calculates heat flow and a second part calculates mixing, to update node temperatures. With the new node temperatures, the next iteration is run until temperature changes between iterations reduced to an acceptably small value, this being the convergence criteria

Our explicit solver uses large timesteps () to rapidly solve for network heat transfer. This can be on the order of 1 yr or perhaps even larger. To support these large steps, we consider the rock’s heat deliverability and the fluid’s heat capacity across each flow segment. In a timestep, this requires solving the following set of equations,

(T.1)

(T.2)

(T.3)

Where, is rock conduction along a segment where both the upstream and downstream nodes are out of equilibrium with the adjacent rock temperature. is rock conduction where only the upstream node is out of equilibrium. is fluid enthalpy change if the downstream node is at equilibrium. is fluid enthalpy as determined from the pressure, temperature, and steam quality (IAPWS, 1997). Also, is specific volume at a reference enthalpy that we typically take from the injection well’s fluid inlet state, is the upstream node of the segment, is the downstream node of the segment, and is the equilibrated state at the downstream segment. Which of , , or occurs depends on the process that limits heat transfer such that the rock can deliver the heat and the fluid can adsorb this heat. This is determined by,

(T.4)

(T.5)

Now that we know the limiting heat transfer factor for the segment, we can calculate the enthalpy at the downstream node as,

(T.6)

We can then use the mixed inflowing enthalpy for each node to obtain an updated nodal enthalpy by,

(T.7)

(T.8)

Where, is mixed nodal inflow enthalpy and only inflow streams are included in the calculation. By reference to the nodal pressure, this completes the state variables to get nodal temperature, . We acknowledge that this approach does not support compressible phase change inside the pipe and node flow network being modeled. However, geothermal waters are predominantly in a saturated liquid state due to the high pressure in the subsurface. This model is expected to function properly if the water remains in a saturated liquid state inside the region being modeled.

## Power Output

A key output from our model is a prediction of the produced fluid enthalpy and flow rate from a leaky and fractured geothermal reservoir. This provides the required input for thermal-electric power output estimation. For this, there are many available technologies to convert the high-enthalpy produced water into electrical energy, with common examples including the Flash Rankine Cycle and the Binary or Organic Rankine Cycle. For a first cut at this prediction, we will focus on the Single Flash Rankine Steam Cycle (Fig. 11). This provides a low estimate of electrical output potential from a reservoir of a given design compared to what is possible. However, using this cycle helps to reconcile enthalpy and flow rate into an easily understood term of net-energy production.

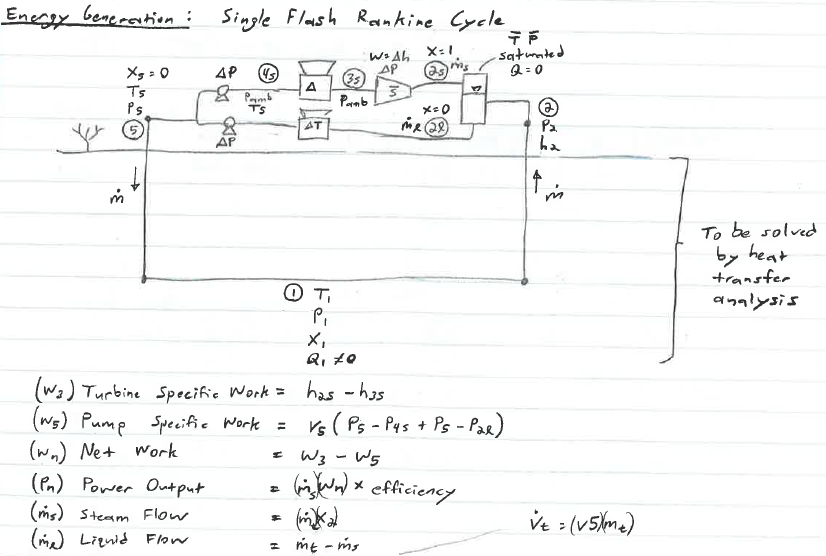


Fig. 11. Single flash Rankine cycle used to estimate geothermal to electrical energy production potential.

# Example Model Results

This model is ultimately a tool to estimate Enhanced Geothermal System performance as a function of key decisions in site development when only a minimum of information is known about the site. To steer development, we anticipated that these key decisions would include:

1. Selecting an optimal placement for all injection and production wells (e.g., well spacing).
2. Selecting optimal production and injection rates for long term reservoir performance.

To successfully aid this decision making, the model must we able to predict the primary factors that will lead to failure of an EGS site. These factors include:

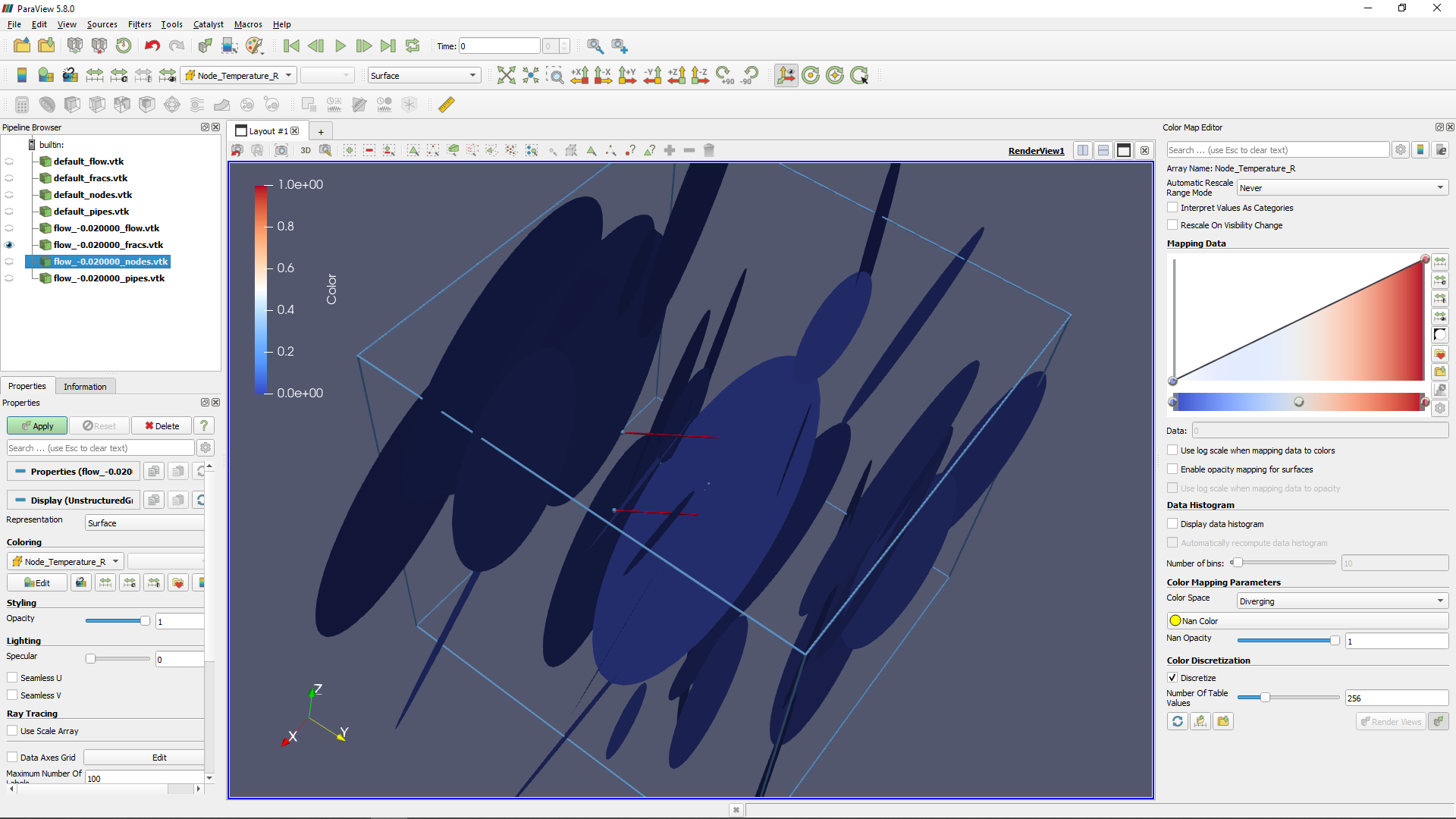
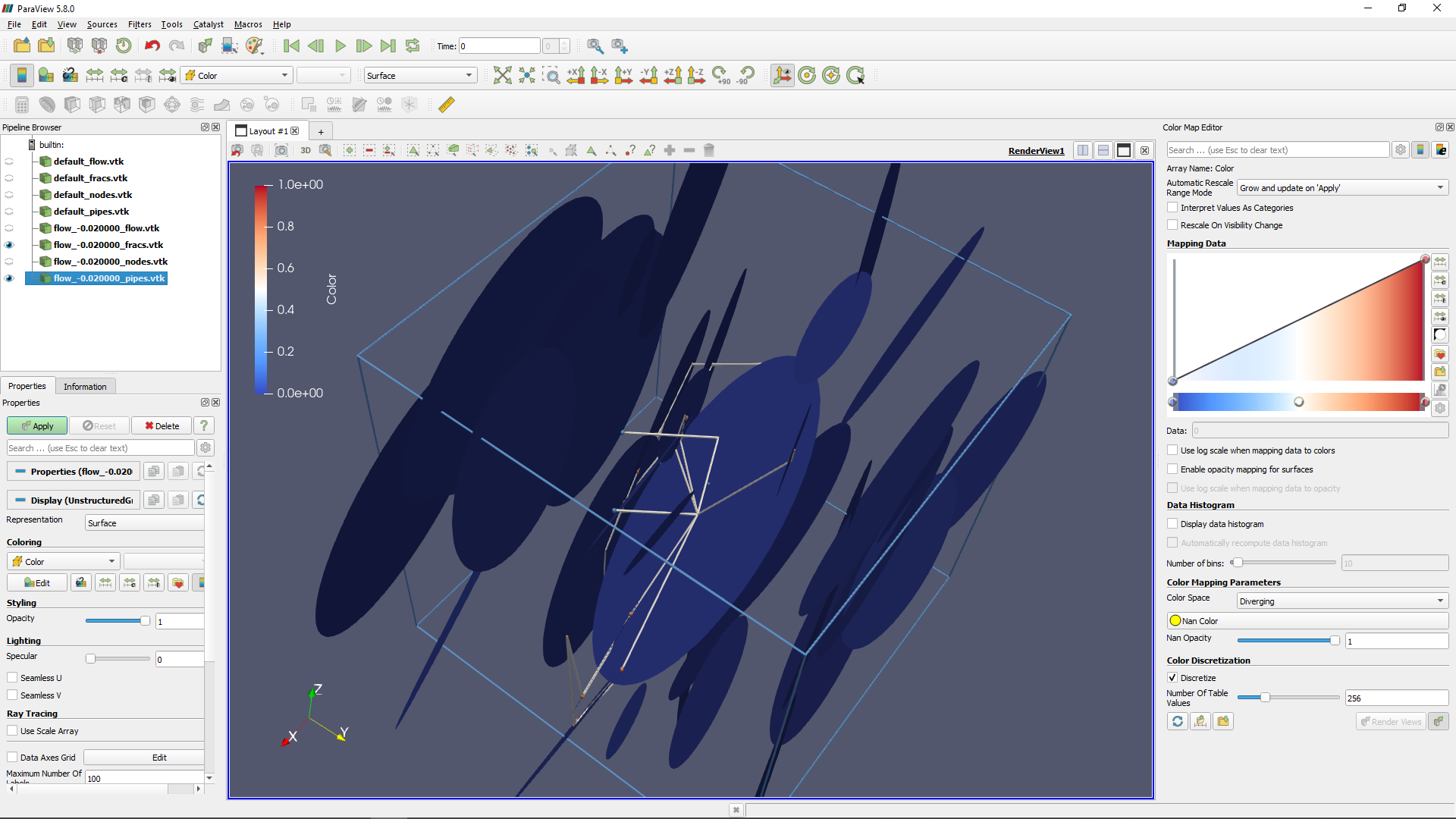
1. Short-circuiting where only one or two fractures dominate reservoir flow, resulting in early thermal breakthrough and reduced project life.
2. Poor well connectivity, resulting in high leak-off of the injected fluid, reduced reservoir efficiency, or lower than required output for economical needs.
3. Excessive injection pressures, resulting in reduced net power output or becoming victim to the technical feasibility barriers of pump and casing pressure limits. This high-pressure situation can also result with increased risk of induced seismicity.
4. Low produced fluid enthalpy or flow rate, resulting in low potential for energy generation.

To demonstrate our model’s application to these needs, we investigate the performance of an injection and production well pair in an EGS reservoir having the properties summarized in Table 1. It is important to note that these properties represent the required inputs for our model. Key decision variables that were explored included injection rate and well spacing. Analysis of the sensitivity of the results to changes in the inputs is a site-specific subject that this model is intended to be used for.

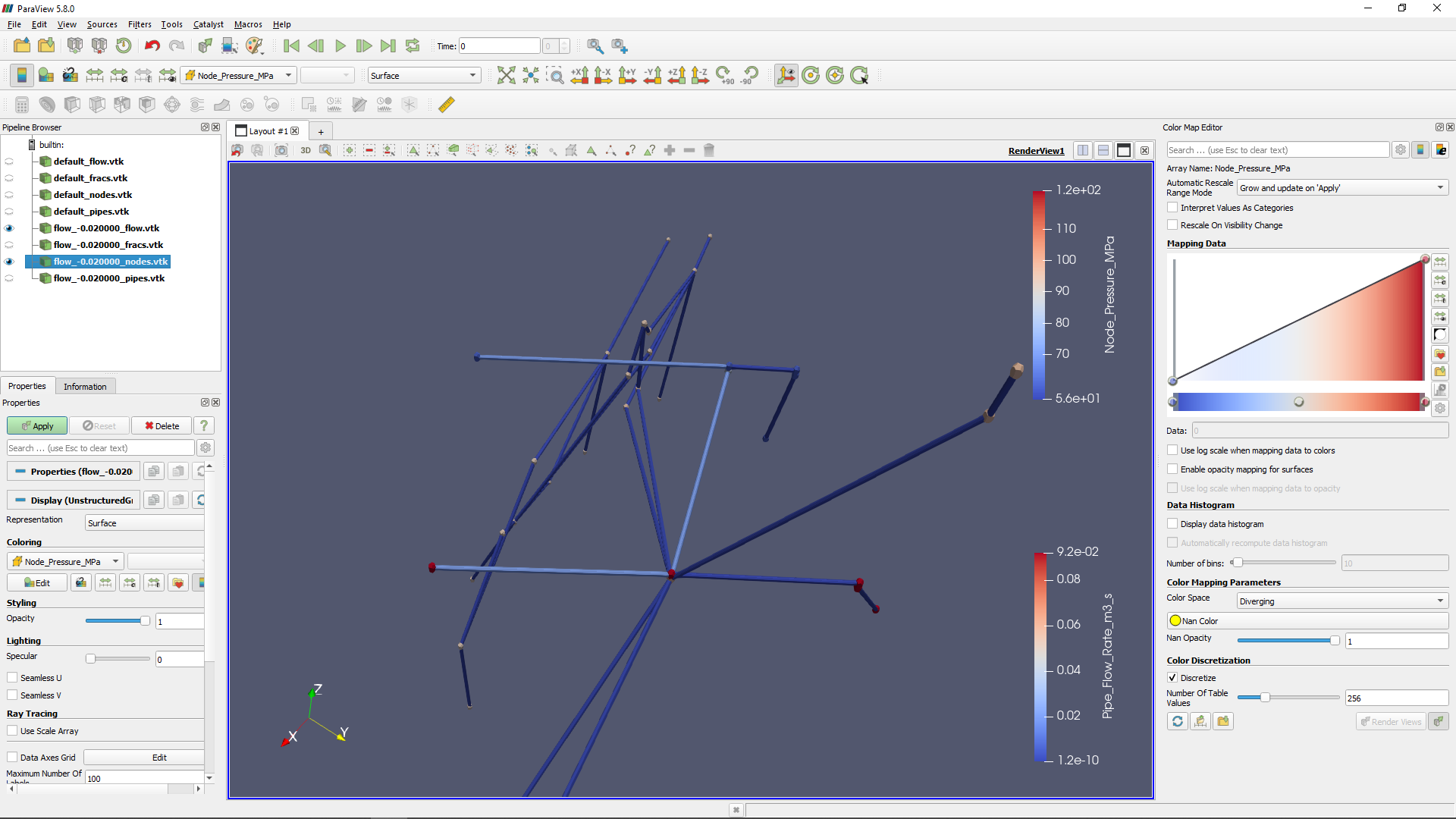
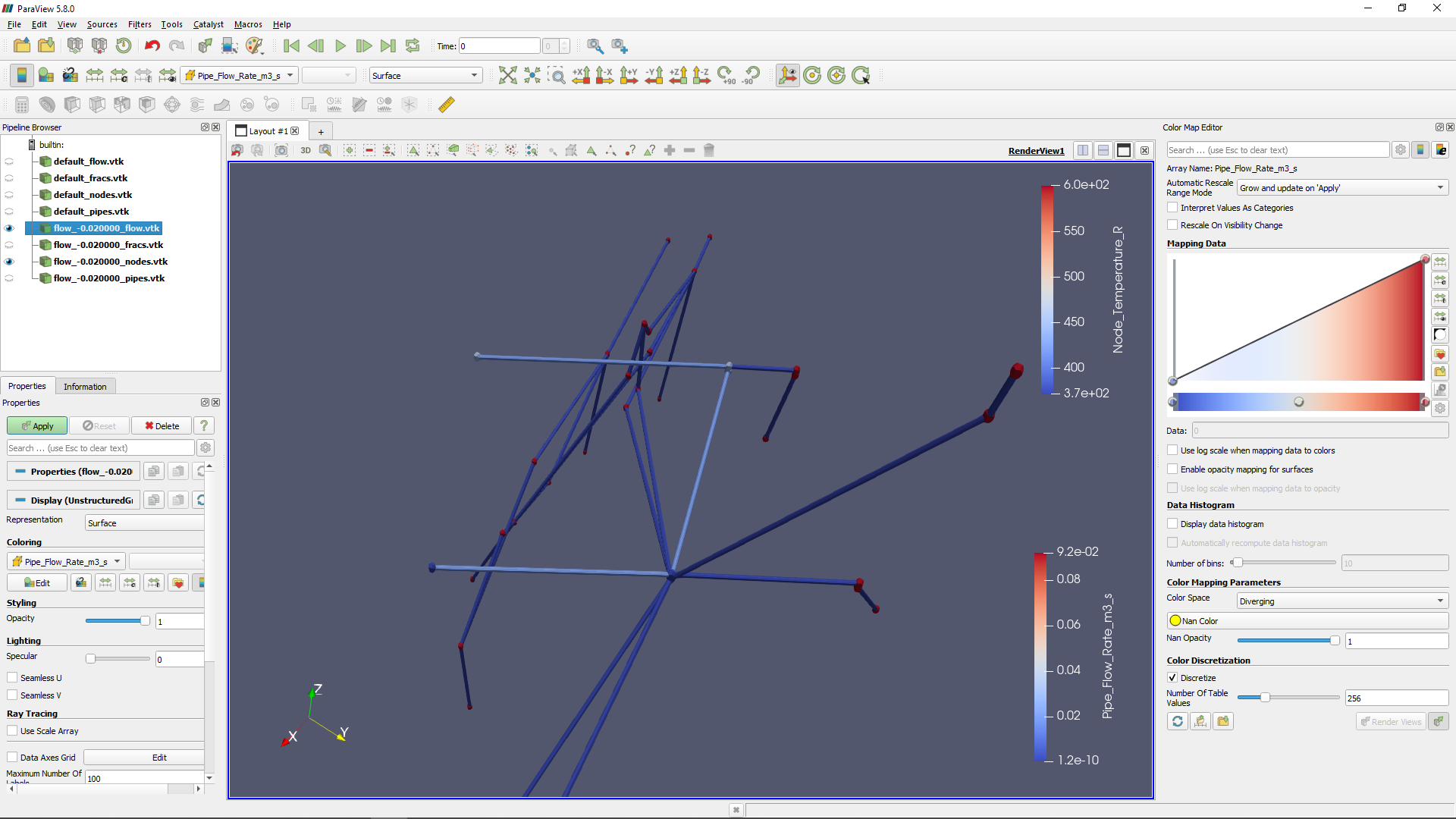
Table 1. Example model input parameters with optional model inputs indicated by shaded boxes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Unit** | **Value (mean)** | **Uncertainty (±)** |
| Domain size (i.e., cubic side length) | m | 1600 | - |
| Nominal reservoir depth | m | 6000 | - |
| Geothermal gradient | K/km | 50 | - |
| Rock density | kg/m3 | 2700 | - |
| Rock thermal conductivity | W/mK | 2.5 | - |
| Rock volumetric specific heat capacity | kJ/m3K | 2063 | - |
| Ambient surface temperature | C | 25.0 | - |
| Ambient surface pressure | MPa | 0.101 | - |
| Cement thermal conductivity | W/mK | 2.0 | - |
| Cement volumetric specific heat capacity | kJ/m3K | 2000 | - |
| Electrical generator efficiency | % | 85 | - |
| Project lifespan | yr | 20 | - |
| Thermal analysis timestep | yr | 1.0 | - |
| Casing inner radius | m | 0.0762 | - |
| Casing outer radius | m | 0.0889 | - |
| Borehole radius | m | 0.1016 | - |
| Borehole thermal convection coefficient | W/m2K | 3000 | - |
| Hazen-Williams friction coefficient | - | 80.0 | - |
| Water density for flow analysis | kg/m3 | 980.0 | - |
| Water dynamic viscosity | cP | 0.9 | - |
| Reservoir pore pressure | MPa | 57.7 | - |
| Reservoir temperature | C | 325.0 | - |
| Well spacing | m | 360 | - |
| Well length | m | 600 | - |
| Well azimuth | deg | 324.0 | - |
| Well dip | deg | 0.0 | - |
| Natural fracture count | fractures | 20 | - |
| Natural fracture diameter | m | 550.0 | 350.0 |
| Natural fracture strike | deg | 79.0 | 16.0 |
| Natural fracture dip | deg | 90.0 | 15.0 |
| Hydraulic fracture count | fractures | 1 | - |
| Hydraulic fracture diameter | m | 1266.0 | 214.0 |
| Hydraulic fracture strike | deg | 65.0 | 8.0 |
| Hydraulic fracture dip | deg | 57.5 | 7.5 |
| Fracture aperture (all) | m | 0.0002 | 0.0001 |
| Injection rate | m3/s | 0.002 | - |
| Injection temperature | C | 95 | - |
| Production wellhead pressure | MPa | 1.0 | - |

The fracture and borehole system generated for our example is shown in Fig. 12. In this scenario, the model predicts that the injection and production rates will be nearly equal, but there is some injected fluid leak-off to the boundary via the natural fractures. Most of the flow travels through the hydraulic fracture from the injection well to the production well.

(a) Fractures and wells (b) Pipe and node network from fracutres

(c) Node pressure and pipe flow rates (d) Node temperatures and pipe flow rates

Fig. 12. Example model prediction of flow, pressure, and fluid temperatures for a pair of wells in a naturally and hydraulically fractured network.

In addition, the model predicts thermal breakthrough and net electrical power output (Fig. 13) as a function of injection rates. As would be conceptually anticipated, the solution predicts earlier thermal breakthrough with increasing injection rates. In this example, thermal breakthrough in the first 20 yrs begins with a flow of only 0.005 m3/s (5 kg/s). However, the optimal flow rate for power generation is higher at around 0.020 m3/s (20 kg/s), despite thermal breakthrough occurring sooner. With more stochastic realizations, the model can be used to identify an optimized borehole layout that produces the highest likelihood of a high-output long-life geothermal system.

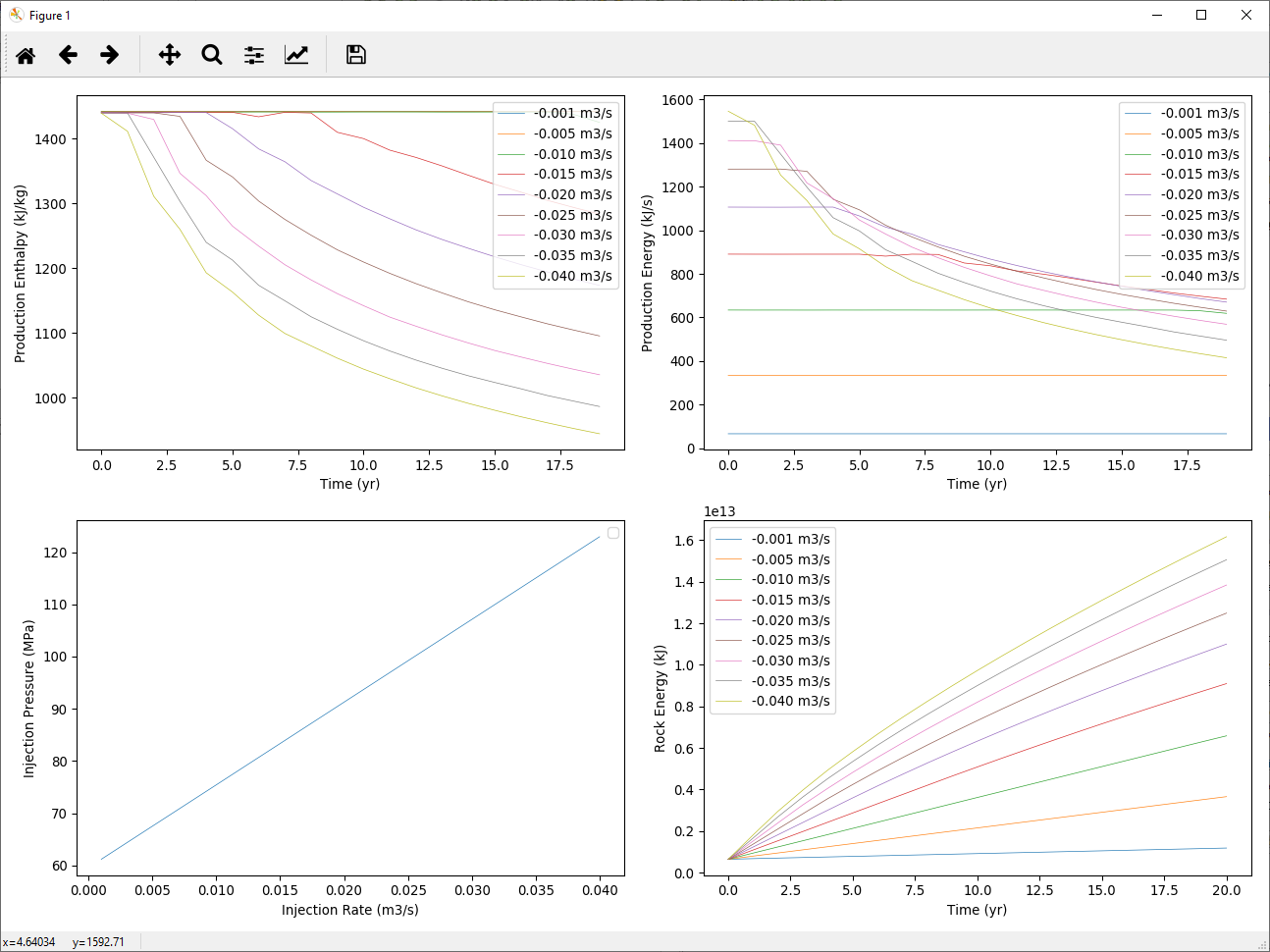


Fig. 13. Predicted produced fluid properties and injection pressure as a function of injection rate in a stochastic fracture network realization. Most importantly, the production energy (electrical output) clearly shows an optimal injection flow rate for maximized power output.

To address borehole placement decision making, we use our model to investigate the variable of well spacing for its control on net electrical power output (Fig. 14). Here, the model predicts an optimal well spacing of 400 m for a single stochastic realization of natural fractures. This observation agrees with other predictions from high-fidelity models for the Utah-FORGE site (Asai, 2018). However, unlike the prior work, our model adds in the complication of fluid loss to the environment and changes the context for design evaluation from rate-enthalpy output to electrical power generation. This change helps to better understand the factors that will lead to success of failure of an EGS project.

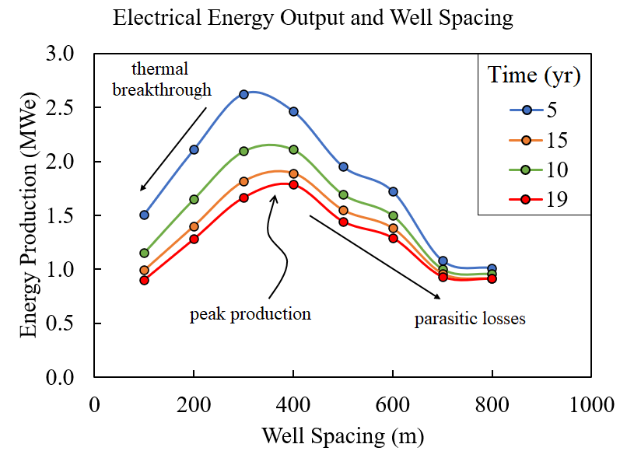


Fig. 14. Net electrical energy production from a Single Flash Rankine Cycle with different spacing between parallel wells. This result predicts that close spacing will cause thermal breakthrough to arrive too early. On the other hand, large well spacing will increase fluid loss, lower reservoir efficiency, and increase pumping costs. Results show the optimal injection rate from the range of 0.001 to 0.065 m3/s.

# Conclusions

Here we present the workings of a fast numerical model that is intended to be useful as a tool for decision making for EGS site development. This tool is not meant to compete with or replace the available high-fidelity numerical models that will undoubtedly be more accurate. Instead, it is meant to help its users more quickly identify the parameters for key design decisions such as well placement when uncertainties are highest and site information is poor. Often, this situation arises early in a project when no deep wells have yet been drilled so the characterization of the site will not yet be known to the high degree that is demanded for more accurate models. Our model also provides a single-script framework that includes model parameterization, mesh generation, a flow solver, a heat transfer solver, and an estimation of net electrical power output. All these steps are completed in our target of less than 5 minutes for basic models that include 40 fractures, two wells, a leaky boundary, and 20 years of production, with a $700 desktop computer.

# Acknowledgements

E3W1, AD33, DG16.

# References

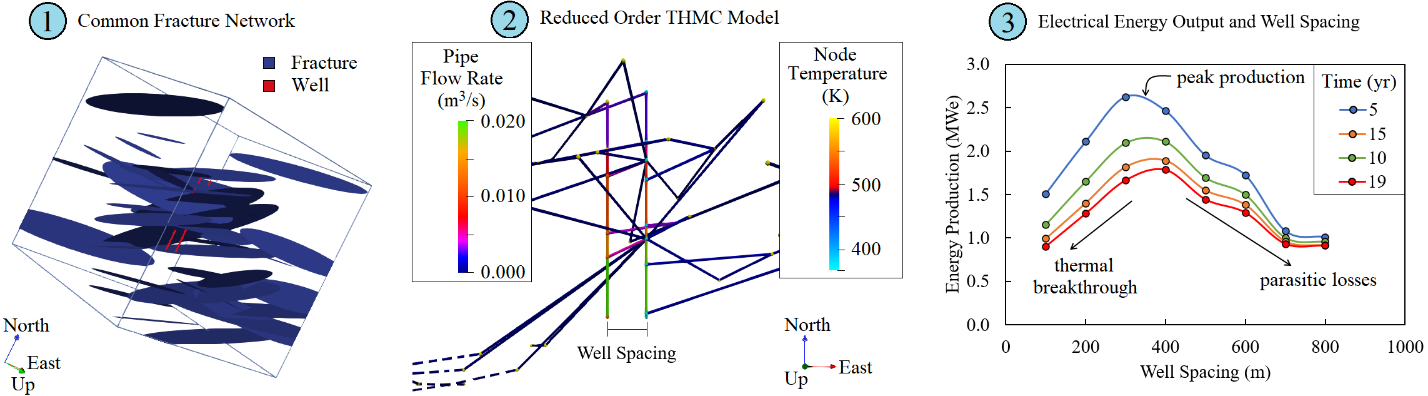
Jeppson, R.W. (1974) Steady Flow Analysis of Pipe Networks: An Instructional Manual. *Reports*. Paper 300.

Engineers Edge. (2020) Convective heat transfer coefficients table chart. *Engineers Edge*. Accessed on Dec 15, 2020. https://www.engineersedge.com/heat\_transfer/convective\_heat\_transfer\_  
coefficients\_\_13378.htm.

Overall heat transfer coefficient. (2020) Accessed on Dec 15, 2020. *TLV Corp.* https://www.tlv.com/  
global/US/steam-theory/overall-heat-transfer-coefficient.html.

Asai, P., Panja, P., McLennan, J., Moore, J. (2018). Performance evaluation of enhanced geothermal system (EGS): surrogate models, sensitivity study and ranking key parameters. Renewable Energy 122: 184-195.

**MORE STUFF TO LOOK AT**



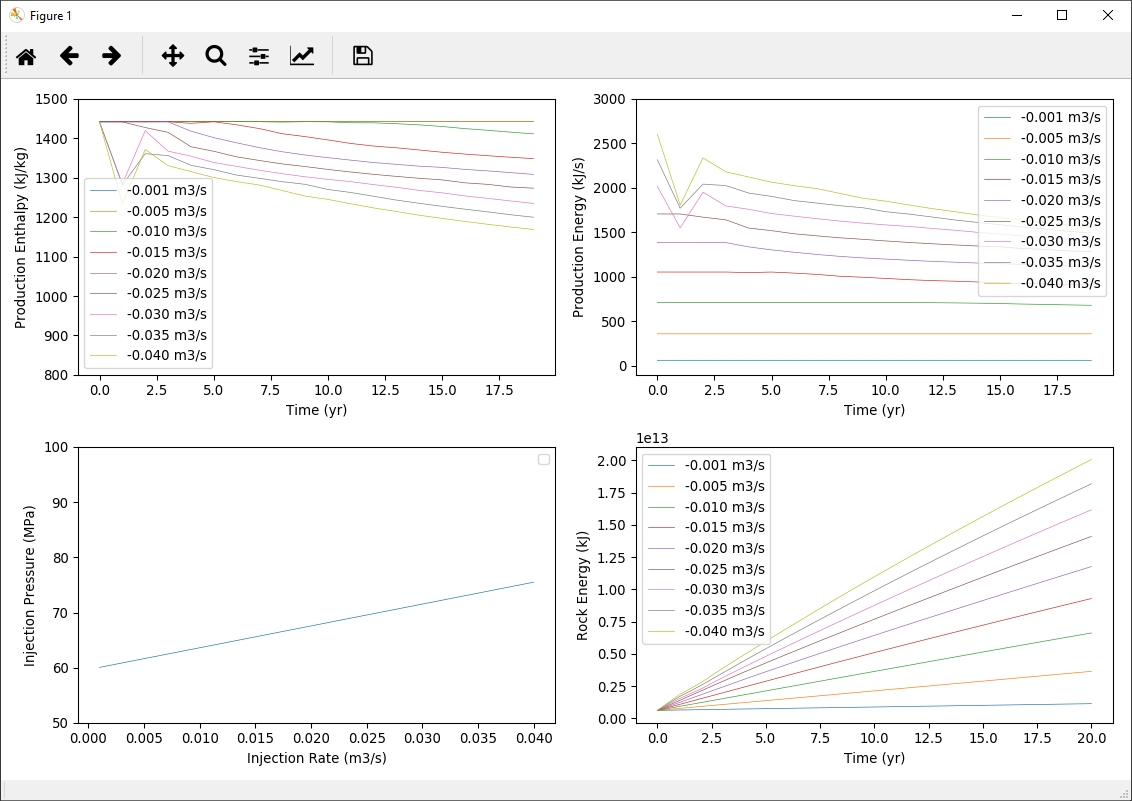


Fig. 6. Parallel 600 m long North oriented wells with injection at 0.001 to 0.040 m3/s (~1.0 to ~40.0 kg/s). Well spacing is 100 m.

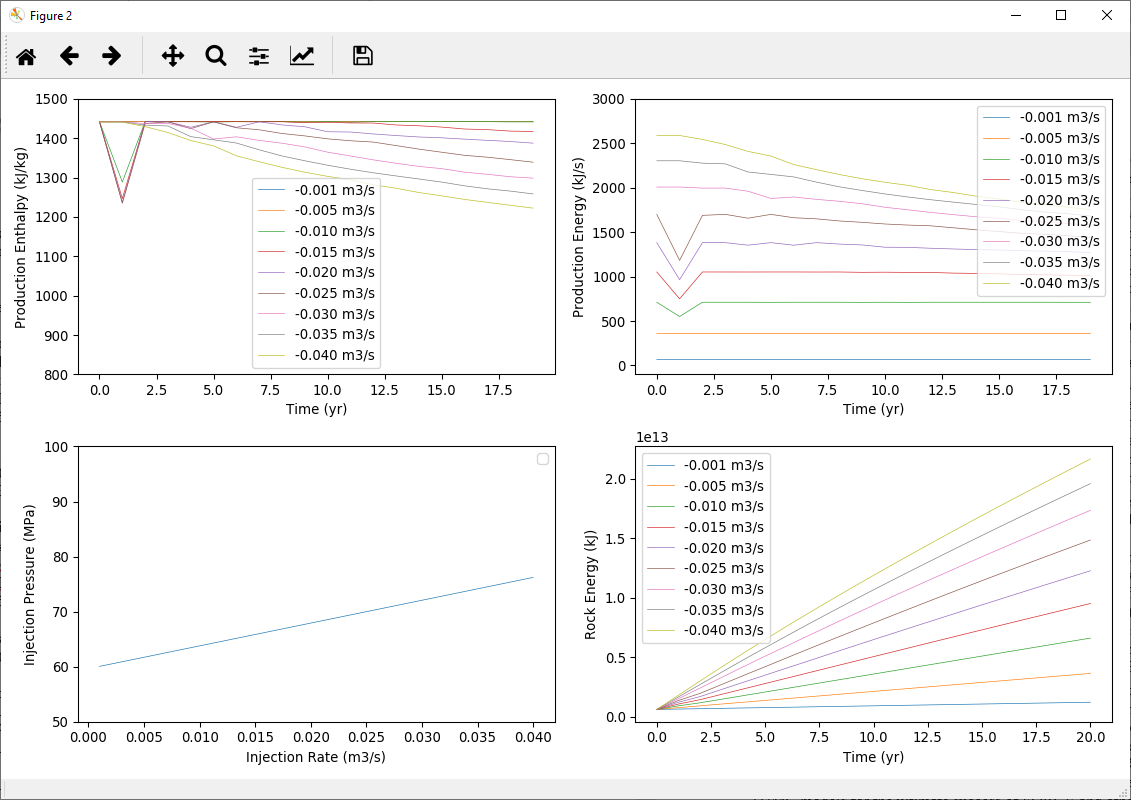


Fig. 7. Well spacing of 200 m.

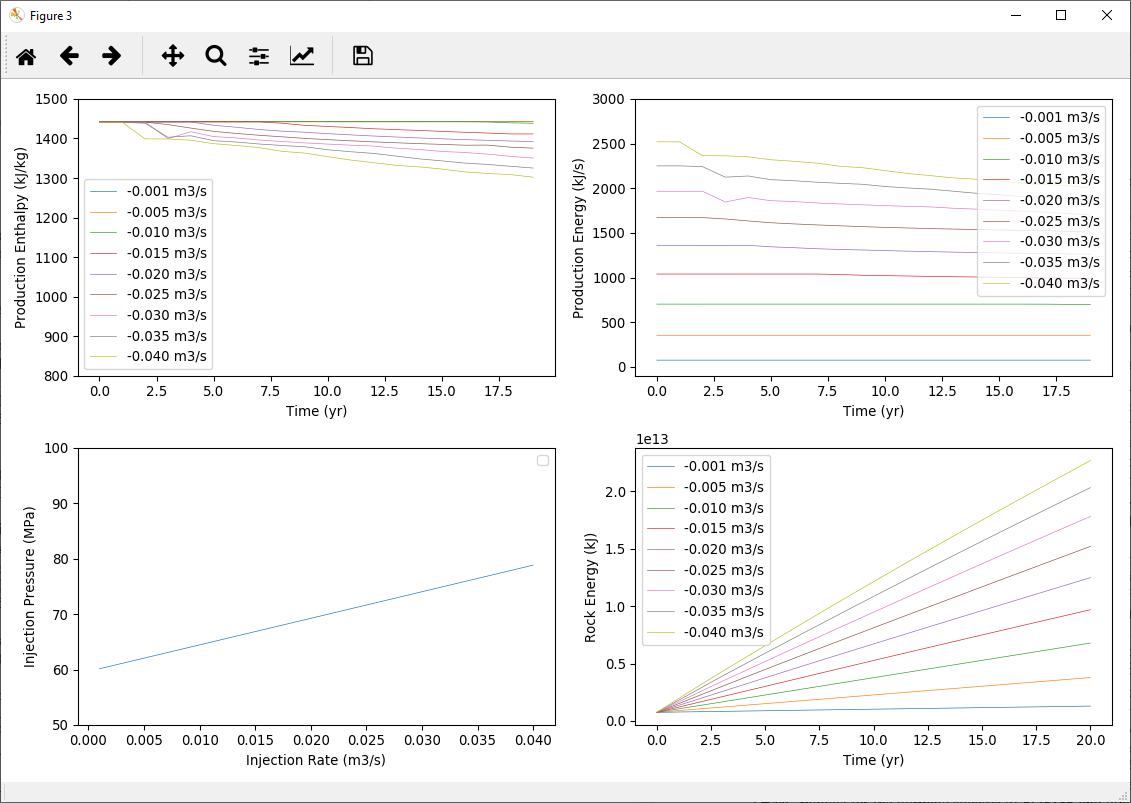


Fig. 8. Well spacing of 300m.

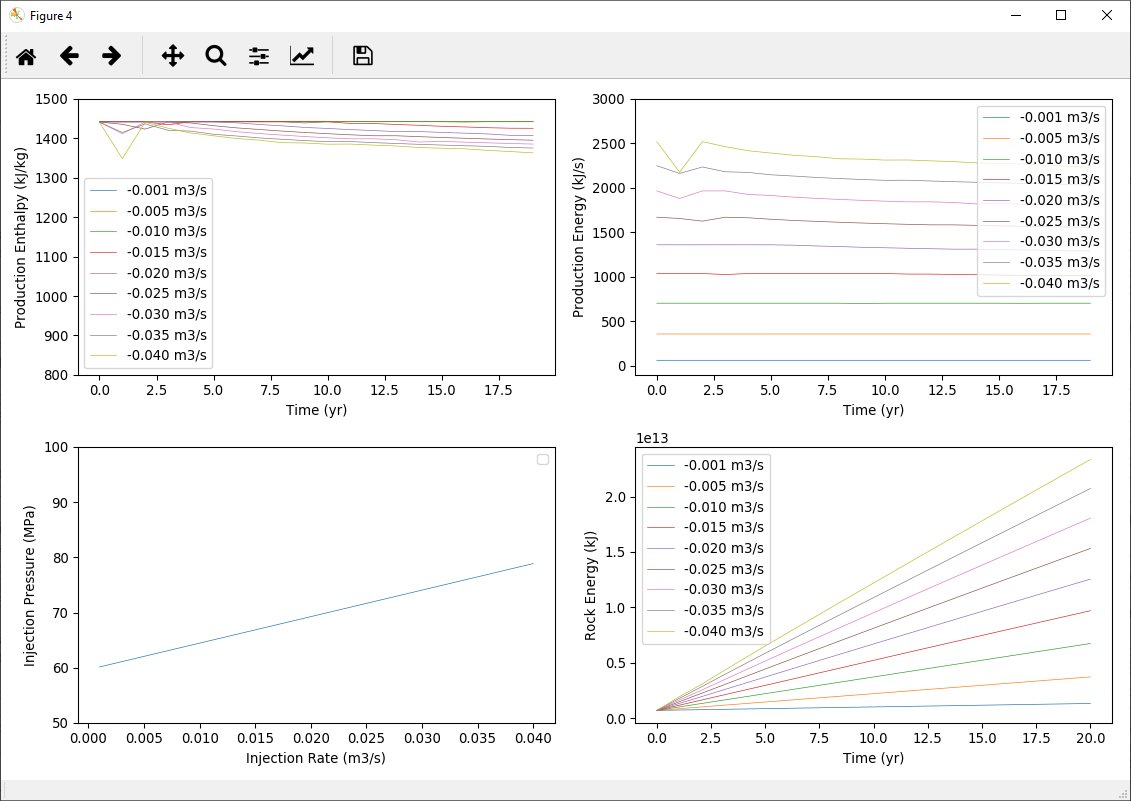


Fig. 9. Well spacing of 400 m.

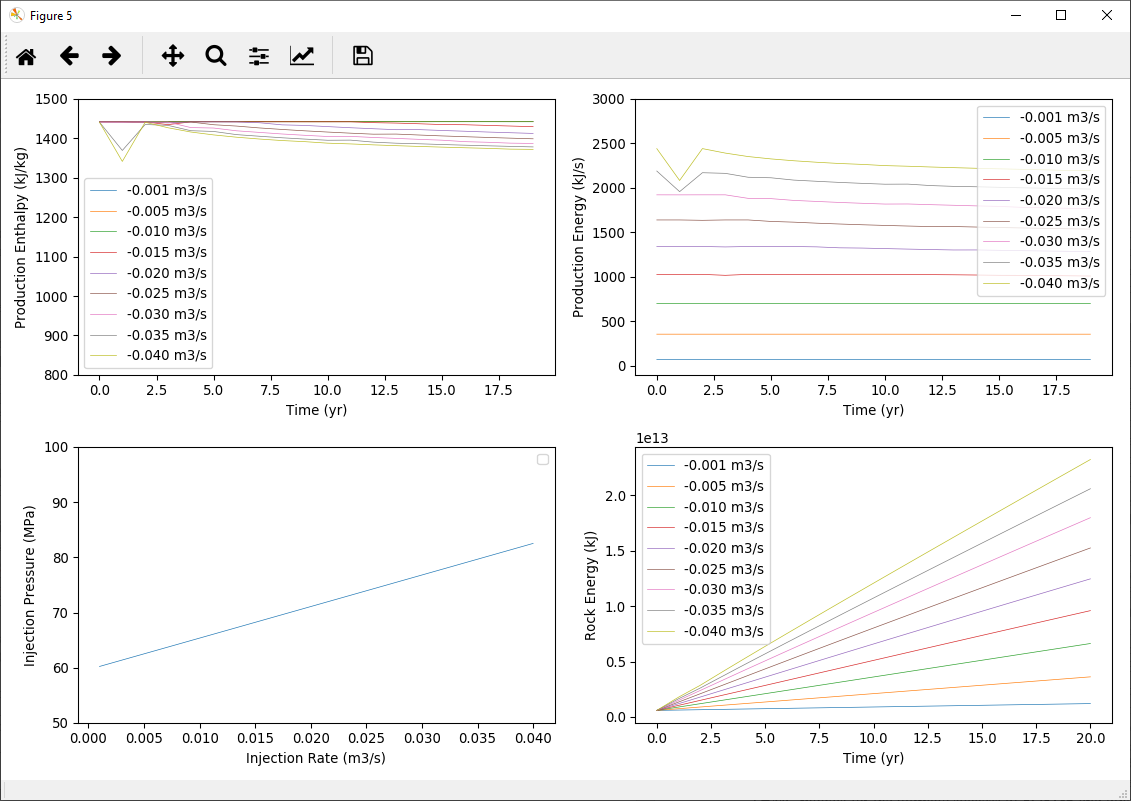


Fig. 10. Well spacing of 500m.

Test Run 2: Parallel North oriented wells with spacing from 100 to 800 m and flow rate from 0.001 to 0.065 m3/s. Fracture orientations from the MSEEL site.

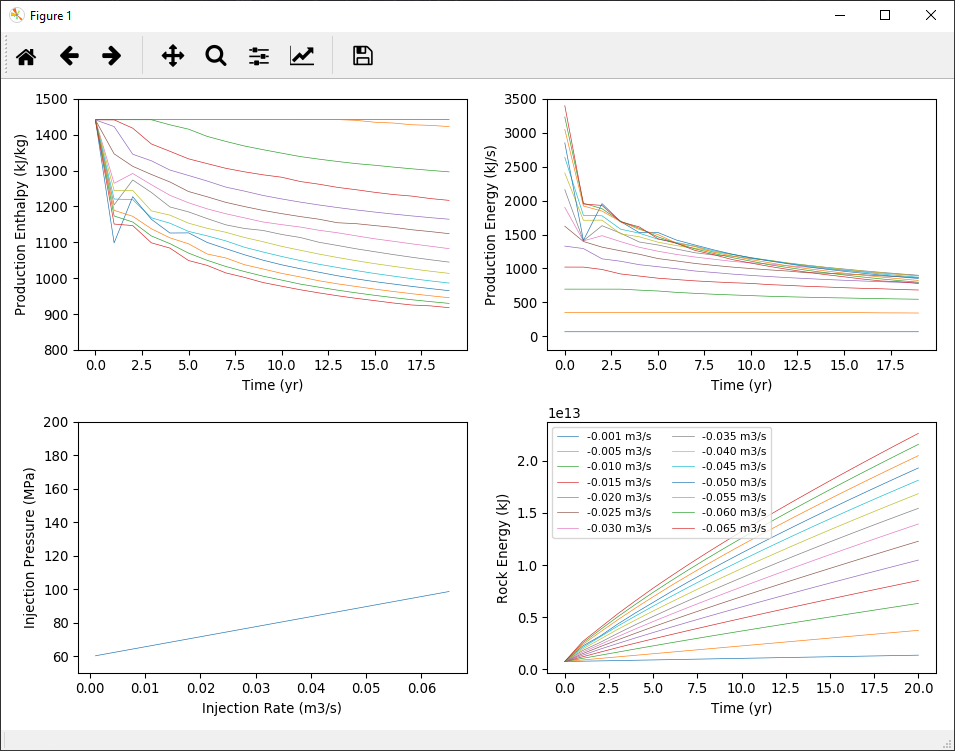


Fig. 11. Wells with 100 m spacing

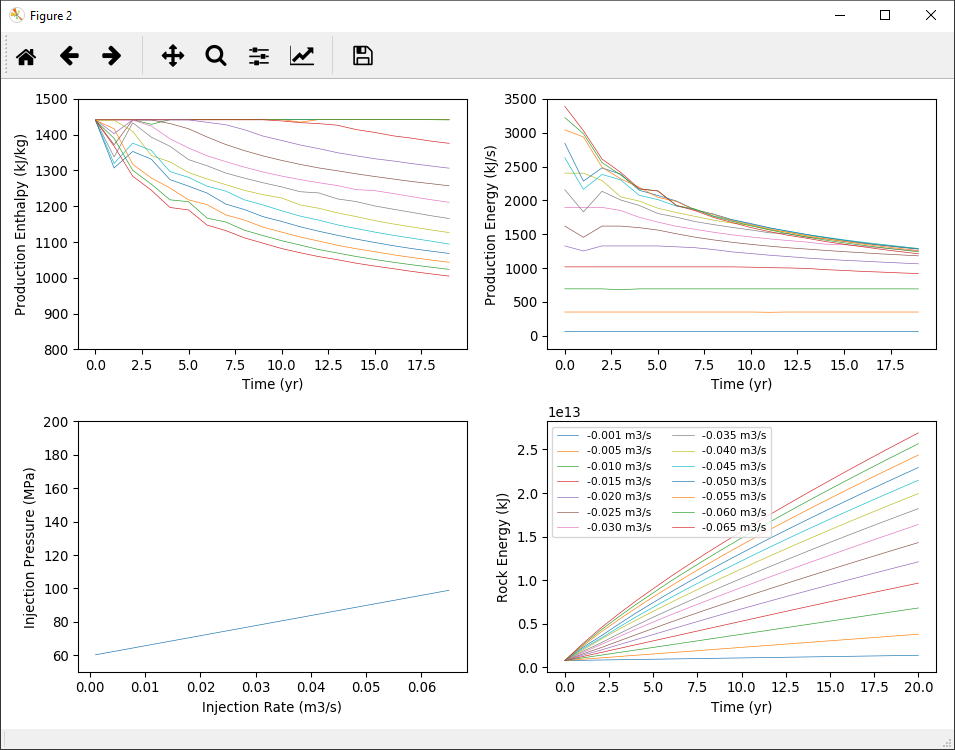


Fig. 12. Wells with 200 m spacing.



Fig. 13. Wells with 300 m spacing.

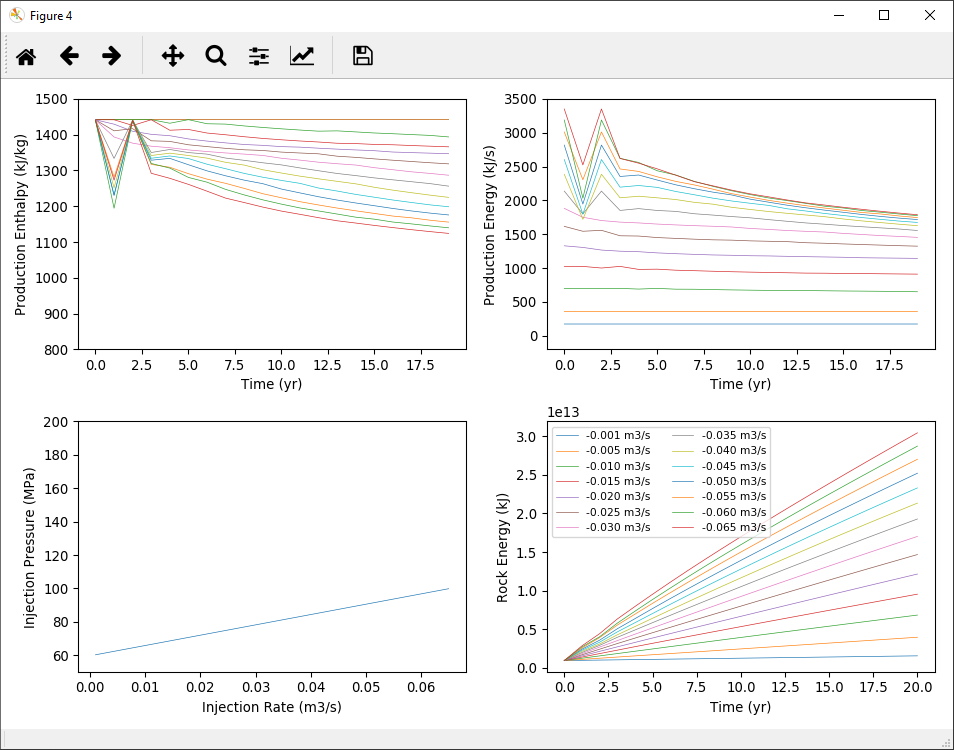


Fig. 14. Wells with 400 m spacing

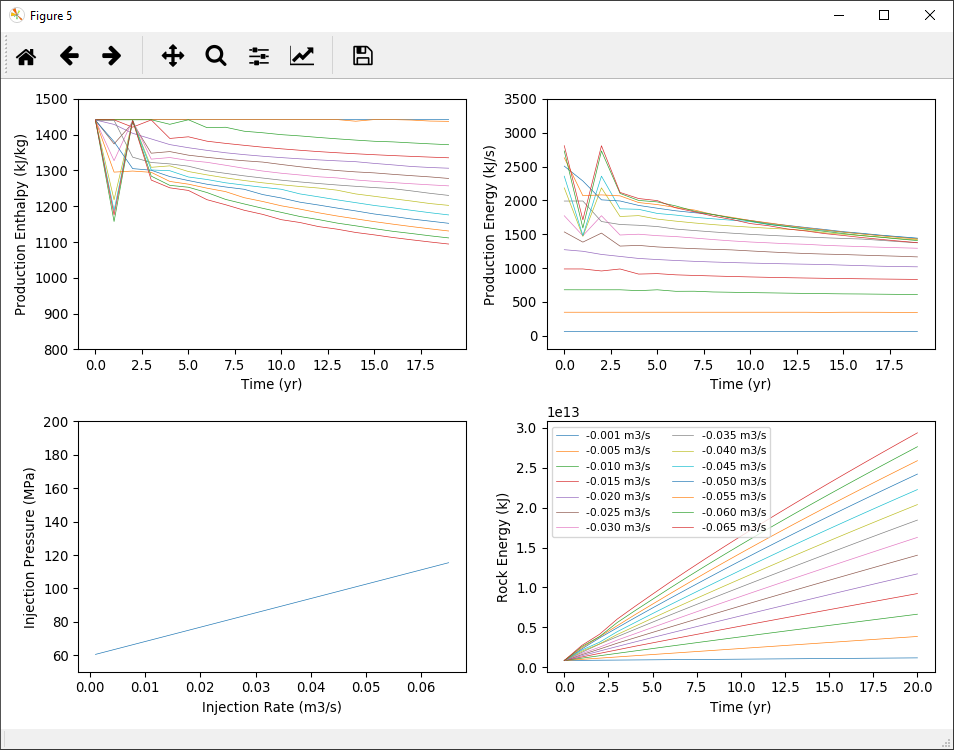


Fig. 15. Wells with 500 m spacing.

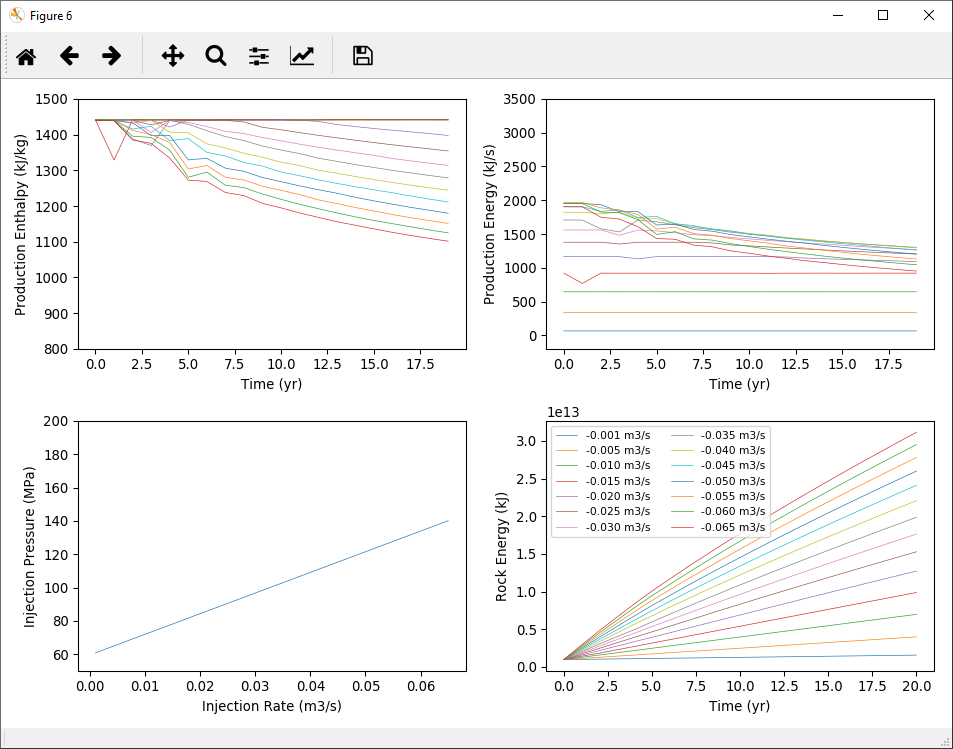


Fig. 16. Wells with 600 m spacing.

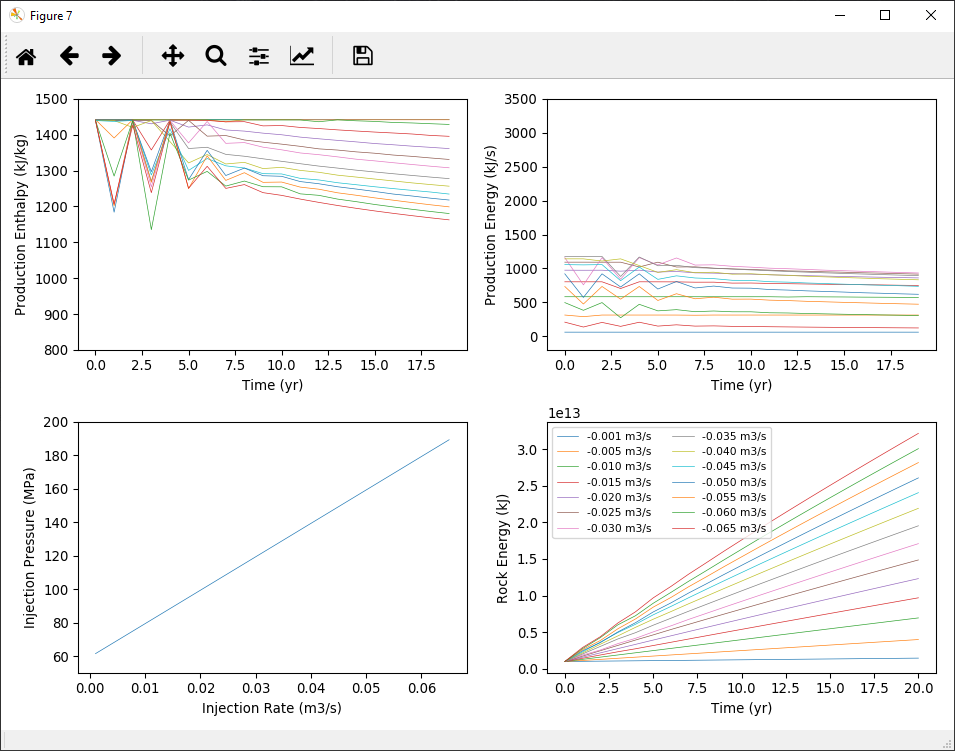


Fig. 17. Wells with 700 m spacing.

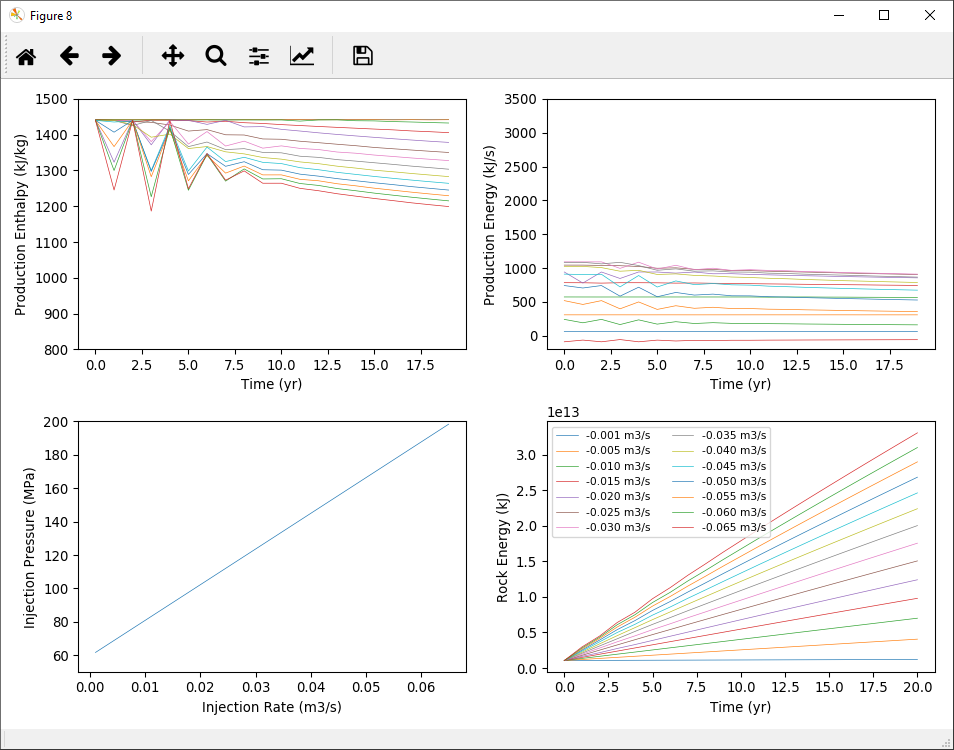


Fig. 18. Wells with 800 m spacing.