PROCEEDINGS, 51st Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, February 9-11, 2026
SGP-TR-230

A Full Three-Dimensional GPU-Accelerated Model for Deep Borehole Heat Exchangers (DBHEs) Enabling Simulation of Well Arrays

Collin Wittenstein[†1,3], Emmanuel Lujan[‡1], Andrew Inglis[2], Robert Metcalfe[1], Alan Edelman[§1] and Hendrik Ranocha[¶3]

[1]Julia Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[2]MIT Proto Ventures, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[3]Institute of Mathematics, Johannes Gutenberg University Mainz, 55099 Mainz, Germany

Corresponding Author: Collin Wittenstein, collin.wittenstein@gmail.com

Keywords: Deep borehole heat exchangers, coaxial, well arrays, geothermal energy, GPU computing, three-dimensional simulation, numerical modeling, operator splitting, Julia

ABSTRACT

Deep borehole heat exchangers (DBHEs) present significant computational challenges due to their multi-scale geometry and long operational timescales. We present a GPU-accelerated three-dimensional model that makes well array simulations computationally tractable through an operator splitting strategy tailored to the problem's physics. The method separates vertical diffusion (stabilized explicit Runge–Kutta–Chebyshev), horizontal diffusion (alternating direction implicit), and advection (semi-Lagrangian), achieving near-unconditional stability with high efficiency. We validate against three published models using different numerical approaches, showing excellent to good agreement. The vendor-agnostic Julia implementation enables full three-dimensional simulation of multi-well arrays on a single GPU, opening new possibilities for systematic design optimization and long-term performance assessment of geothermal well systems. The implementation is released as the open-source Julia package GeothermalWells.jl.

1 INTRODUCTION

Deep borehole heat exchangers (DBHEs) offer a compelling approach to geothermal energy extraction by circulating a working fluid through a closed-loop pipe-in-pipe system, eliminating the need for hydraulic fracturing and avoiding direct contact with formation fluids [12]. This closed-loop design significantly reduces geological risk and eliminates induced seismicity concerns from fracking [10].

While single-well DBHE systems have been studied extensively, the thermal output of individual wells is often insufficient for large-scale applications. Arrays of networked wells present a natural path toward scalability [5,21]: multiple boreholes can aggregate thermal power, provide redundancy against individual well failures, and potentially achieve near-continuous capacity factors through coordinated operation. However, designing and optimizing such arrays requires understanding the thermal interactions between wells over operational timescales of years to decades. Limited research exists on DBHE arrays [12], with existing studies noting that sufficient borehole spacing is needed to avoid thermal interference [4] and that computational costs have restricted systematic investigation of array configurations [12]. The present work addresses this gap by developing a computational framework that makes full three-dimensional simulation of well arrays tractable.

1.1 The Computational Challenge

Accurately modeling the thermal performance of deep coaxial borehole heat exchangers presents significant computational challenges. The problem is inherently multi-scale: borehole diameters are on the order of centimeters, the lateral extent of the thermal influence zone can reach a hundred meters, and boreholes extend to depths of several kilometers. Furthermore, realistic performance assessment requires simulating operational timescales spanning years to decades, during which the thermal state of the surrounding rock evolves continuously [12]. For well arrays, where multiple boreholes interact thermally over these extended periods, the computational demands increase accordingly.

1.2 Existing Modeling Approaches

Existing numerical approaches for modeling deep coaxial borehole heat exchangers can be categorized into three main strategies, as reviewed by Kolo et al. [12].

The first employs dual-continuum methods that couple a one-dimensional discretization of the borehole with a three-dimensional representation of the surrounding formation. This approach is particularly attractive due to its computational efficiency, but it requires

[†]ORCID: 0009-0006-8591-278X
[‡]ORCID: 0000-0002-5945-4766
[§]ORCID: 0000-0001-7676-3133
[¶]ORCID: 0000-0002-3456-2277

careful treatment of interface coupling conditions and cannot fully resolve the flow and heat transfer within the DBHE structure. Examples include models by Cai et al. [5] and Brown et al. [3].

The second strategy reduces the problem to two dimensions by employing cylindrical axisymmetric discretizations in which the borehole is coupled to the surrounding rock. While computationally efficient, this method cannot easily accommodate well arrays due to its inherent symmetry assumptions. An example is the model by Li et al. [14].

The third approach employs full component discretization, encompassing both the DBHE and the surrounding rock formation in a unified discretization that is either two- or three-dimensional. An example of a single three-dimensional well is the work by Hu et al. [10]. This approach, using three dimensions, has previously been considered prohibitively expensive for long-term simulations of DBHE arrays [12].

1.3 Our Contribution

In this paper, we demonstrate that full three-dimensional discretization can be made computationally tractable, even for multi-year well array simulations, through careful numerical design and GPU acceleration. We present a numerical model that solves the three-dimensional advection-diffusion equation on a unified computational domain. The key innovation lies in an operator splitting strategy specifically tailored to the physics and geometry of the problem, which separates the problem into subproblems, each treated with an appropriate numerical method: vertical diffusion is treated with stabilized explicit Runge–Kutta methods, horizontal diffusion with the alternating direction implicit (ADI) scheme, and advection with the semi-Lagrangian method. This combination achieves great numerical stability while maintaining computational efficiency. The implementation leverages GPU acceleration through vendor-agnostic kernels written in Julia [2], enabling portable high-performance execution across different hardware platforms. We validate our model against existing published results, demonstrating good to excellent agreement.

This new approach enables systematic design space exploration, optimization studies, and assessment of long-term thermal interference effects in multi-well configurations. The simulation framework is available as the open-source Julia package GeothermalWells.jl [23], and all code to reproduce the results in this paper is available in an accompanying repository [22]. Multi-GPU parallelization is currently under development and will enable faster simulations of even larger well arrays.

1.4 Paper Organization

The remainder of this paper is organized as follows. We begin by presenting the physical model in Section 2, including the governing equation, geometry representation, boundary conditions, and modeling assumptions. Section 3 then describes our numerical approach: spatial discretization, operator splitting strategy, and time integration schemes. In Section 4, we establish the model's accuracy through convergence studies and validation against published results. Section 5 demonstrates the method's capabilities through simulations of well arrays. We conclude with a discussion of potential future work in Section 6 and closing remarks in Section 7.

2 MODEL DESCRIPTION

We present a numerical model for simulating heat transfer in deep coaxial borehole heat exchangers. The model solves the advection-diffusion equation on a three-dimensional domain encompassing both the borehole and surrounding rock formation.

2.1 Governing Equation

The temperature distribution T(x,y,z,t) evolves according to the advection-diffusion equation with spatially varying thermal properties:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \rho c\, v_z \frac{\partial T}{\partial z},$$

where $\rho c = \rho c(x, y, z)$ denotes the volumetric heat capacity, $k = k(x, y, z)$ the thermal conductivity, and $v_z = v_z(x, y, z)$ the vertical fluid velocity. We adopt a coordinate system where z increases downward from the surface. The equation is adapted from [10], omitting viscous dissipation and pressure work terms.

Diffusion occurs everywhere, with different volumetric heat capacities and thermal conductivities depending on the material (rock, grout, pipe wall, working fluid). The advection term acts only within the borehole pipes. In the coaxial configuration, fluid descends through the outer annulus with velocity v_{inlet} and ascends through the inner pipe with velocity v_{outlet}. Both velocities are assumed to be constant in each section. Rather than solving the full fluid dynamics equations, we model only the temperature field, with the advection term transporting heat at prescribed flow velocities.

2.2 Geometry and Material Representation

The computational domain includes both the borehole structure and the surrounding rock formation. We represent different materials through spatially varying thermal properties on a Cartesian grid. Each grid point is assigned material-specific values of k and ρc based on its location: rock, grout, outer pipe wall, working fluid in the outer annulus, inner pipe wall, or working fluid in the inner pipe.

This material-map approach offers flexibility in modeling complex geometries, such as multiple boreholes or depth-dependent rock properties (e.g., layered formations), by simply modifying the property assignments. However, it does not explicitly account for solid-fluid interfaces.

2.3 Boundary Conditions

At the outer boundaries of the computational domain, we impose homogeneous Neumann (no-flux) conditions for the diffusion, choosing the domain large enough that temperature perturbations from the borehole do not interact significantly with the boundaries within the simulation time.

At the borehole inlet (top of the outer annulus), the temperature is prescribed according to a Dirichlet boundary condition. This may take the form of a constant temperature, a time-varying schedule, or a model that couples the outlet temperature to the inlet through a heat exchanger.

At the borehole outlet (top of the inner pipe), a free outflow condition is applied: the advected temperature simply exits the domain.

At the borehole bottom, where the fluid transitions from the outer annulus to the inner pipe, we assume perfect mixing: we take the average temperature in the x-y plane at the bottom of the annulus and advect this averaged value to the bottom of the inner pipe. We do not explicitly model the U-turn pipe connection at the bottom, nor does the simulation track actual fluid flow.

2.4 Model Limitations

As with every mathematical model of a physical system, the present model involves several simplifications. For clarity, we summarize the key modeling assumptions and limitations introduced in the preceding subsection:

- **Simplified fluid dynamics.** Temperature is advected at prescribed constant velocities rather than solving fluid dynamics equations.
- **Pure heat diffusion in rock.** The model only accounts for heat diffusion in the surrounding rock formation, with no advective heat transport.
- **No explicit fluid-solid interface treatment.** The material-map approach assigns thermal properties based on grid point location but does not enforce any interface conditions.
- **Constant thermal properties.** The model assumes temperature-independent properties, such as thermal conductivity, heat capacity, or density.
- **Perfect mixing assumption at borehole bottom.** The U-turn pipe section is not included in the computational domain. Instead, the fluid transition from the outer annulus to the inner pipe is modeled by averaging temperatures in the x-y plane at the bottom.

3 NUMERICAL METHODS

3.1 Spatial Discretization

The domain is discretized using a non-uniform Cartesian grid. The grid spacing is finest inside the borehole to accurately resolve its geometry and progressively coarsens in the surrounding rock further away from the boreholes. Typical resolutions use $\Delta x, \Delta y \approx 2.5$ mm within and closely around the borehole. In the vertical direction, a uniform spacing of $\Delta z \approx 10$ m–100 m is typically chosen, as vertical gradients in the rock are much weaker than horizontal gradients near the borehole. Typical grid dimensions for a converged solution of a single borehole are approximately $N_x, N_y \sim 150$ and $N_z \sim 50$.

This grid design results in fine horizontal grid spacing at the x- and y-coordinates of the borehole(s) throughout the entire lateral extent, including regions of pure rock, as seen in Figure 1, showcasing the grid in the x-y plane for a 3×3 well array. While this introduces some unnecessary resolution in the rock away from the borehole, maintaining the structured Cartesian topology is essential for computational efficiency and the operator splitting ansatz. The uniform grid-line structure enables the use of highly efficient line-based implicit solvers, particularly the ADI scheme described in Section 3.4.

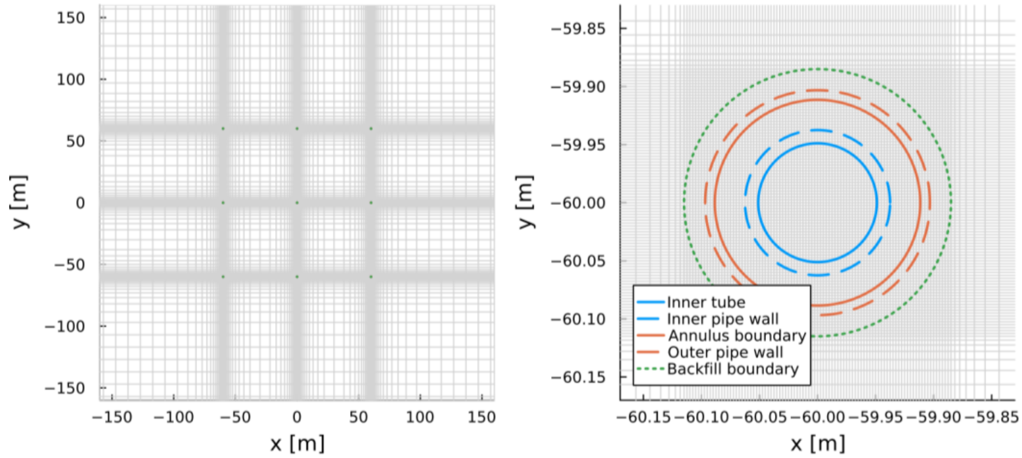

Figure 1: *Left*: Illustration of the computational grid in the x-y plane for a 3 × 3 well array. *Right*: Zoomed-in view of the bottom-leftmost borehole, showing the fine grid spacing at the borehole.

3.2 Operator Splitting

The advection-diffusion equation is solved using operator splitting [19]. We split the right-hand side into three components:

1. Vertical diffusion: $\mathcal{D}_z\phi = \frac{1}{\rho c}\frac{\partial}{\partial z}\left(k\frac{\partial \phi}{\partial z}\right)$
2. Horizontal diffusion: $\mathcal{D}_{xy}\phi = \frac{1}{\rho c}\left[\frac{\partial}{\partial x}\left(k\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial \phi}{\partial y}\right)\right]$
3. Advection: $\mathcal{A}\phi = -v_z\frac{\partial \phi}{\partial z}$

This splitting is motivated by the strongly anisotropic grid: the very fine horizontal resolution ($\Delta x, \Delta y \sim$ mm) would impose prohibitive time-step constraints if treated explicitly, while the coarser vertical resolution ($\Delta z \sim$ m) permits explicit treatment for the diffusion without imposing overly restrictive constraints.

The spatial derivatives in the diffusion term are approximated using second-order finite differences with variable coefficients on a non-uniform grid [13].

3.3 Vertical Diffusion: Stabilized Explicit Runge–Kutta

The vertical diffusion operator $\mathcal{D}_z$ is integrated using a stabilized explicit Runge–Kutta method, specifically the second-order Runge–Kutta–Chebyshev scheme ROCK2 of [1]. Unlike standard explicit methods, ROCK methods are designed for parabolic problems and feature an extended stability region along the negative real axis. This allows significantly larger time steps than those permitted by classical explicit methods.

3.4 Horizontal Diffusion: Alternating Direction Implicit Method

The horizontal diffusion operator $\mathcal{D}_{xy}$ is integrated using the Alternating Direction Implicit (ADI) method [15]. ADI achieves unconditional stability by treating the x- and y-directions implicitly in alternating half-steps as

$$\begin{aligned}\left(I - \frac{\Delta t}{2}\mathcal{D}_x\right)\phi^* &= \left(I + \frac{\Delta t}{2}\mathcal{D}_y\right)\phi^n,\\ \left(I - \frac{\Delta t}{2}\mathcal{D}_y\right)\phi^{n+1} &= \left(I + \frac{\Delta t}{2}\mathcal{D}_x\right)\phi^*,\end{aligned}$$

where $\mathcal{D}_x$ and $\mathcal{D}_y$ denote the one-dimensional diffusion operators in the x- and y-directions, respectively, which both produce a simple tridiagonal matrix structure after finite difference discretization.

Each half-step requires solving a set of independent tridiagonal systems, one along each line in the implicit direction ($N_y \times N_z$ systems for the x-implicit step and $N_x \times N_z$ for the y-implicit step), where each tridiagonal system is solved efficiently using the Thomas algorithm in $\mathcal{O}(N_{x/y})$ operations, where $N_{x/y}$ is the number of grid points along that direction [16].

The key advantage of the ADI method is that it decomposes a multi-dimensional implicit problem into a sequence of one-dimensional problems that can be solved independently, efficiently, and in parallel. This dramatically reduces both computational cost and memory requirements compared to a fully implicit method with direct solvers, which would require solving a coupled system of size $N_x \times N_y \times N_z$.

3.5 Advection: Semi-Lagrangian Method

The advection operator $\mathcal{A}$ transports the heat along the borehole pipes at the prescribed flow velocities, mimicking fluid flow. Standard explicit upwind or central difference schemes would impose a Courant–Friedrichs–Lewy (CFL) condition [8] of $\Delta t \leq \Delta z/|v|$, resulting in $\Delta t \sim$ 1s–10s, which is too restrictive for long-term simulations.

We instead employ a semi-Lagrangian method [18], which is unconditionally stable for advection. For each grid point in the pipe region, the method traces backward along the fluid trajectory over one time step to find the departure point. The updated temperature is then

$$\phi^{n+1}(z_i) = \phi^n(z_i - v_z \Delta t),$$

with v_z constant within each pipe section. Since the departure point $z_i - v_z \Delta t$ generally does not coincide with a grid point, we use linear interpolation between the two neighboring grid points:

$$\phi^n(z_i - v_z \Delta t) = (1 - \alpha)\, \phi^n(z_k) + \alpha\, \phi^n(z_{k+1}),$$

where z_k and z_{k+1} are the grid points bracketing the departure point and α is the interpolation weight.

3.6 Putting Everything Together

Within each full time step Δt, typically on the order of 50 s – 150 s, the operators are applied using the following splitting strategy. First, the vertical diffusion is integrated using the RKC method. Then, the horizontal diffusion and advection are handled together via the ADI scheme, which consists of two half-steps with alternating implicit directions. Within each ADI half-step, the advection is applied between the explicit diffusion update and the implicit solve.

1. Vertical diffusion for Δt (RKC method)
2. First half-step (ADI + advection for $\Delta t/2$):
    a. Explicit y-diffusion (forward Euler)
    b. Advection (semi-Lagrangian)
    c. Implicit x-solve (Thomas algorithm)
3. Second half-step (ADI + advection for $\Delta t/2$):
    a. Explicit x-diffusion (forward Euler)
    b. Advection (semi-Lagrangian)
    c. Implicit y-solve (Thomas algorithm)

The placement of advection after the explicit diffusion step and before the implicit solve was determined empirically. While we lack a complete theoretical justification for this ordering, it has proven robust across all tested configurations. One possible explanation is that applying advection before an explicit Euler-like update can introduce large thermal gradients that destabilize the subsequent step, whereas the implicit solve that follows can handle these gradients stably.

4 NUMERICAL EXPERIMENTS

We have implemented the numerical method described above as the Julia package GeothermalWells.jl [23]. The implementation uses KernelAbstractions.jl [7] for vendor-agnostic kernels, enabling seamless execution across CPU and GPU architectures, including NVIDIA and AMD GPUs. While CPU execution is prohibitively slow for the multi-year simulations typical of geothermal applications, it remains invaluable for local debugging, testing, and prototyping. Time integration of the vertical diffusion operator is performed using the stabilized explicit second-order Runge–Kutta–Chebyshev method ROCK2 from OrdinaryDiffEq.jl [17], while the horizontal diffusion (ADI) and advection (semi-Lagrangian) operators are implemented with custom kernels. Visualizations in this section are done using Plots.jl [6].

All Julia source code to reproduce the numerical results presented in this work is available online [22].

In the following case studies, the parameters needed to reproduce the simulations are not explicitly stated in the text. They can be found either in the cited literature or in the GitHub repository accompanying this paper [22]. Where parameters were not specified, we made reasonable assumptions based on typical values reported in the geothermal literature.

4.1 Convergence Study

As our numerical method employs a non-standard operator splitting approach that treats vertical and horizontal diffusion separately, we first study the accuracy and consistency of this scheme by performing temporal and spatial convergence studies.

To establish baseline accuracy, we restrict our convergence analysis to the pure diffusion case with constant diffusivity D, where an analytical solution is available [9]. We employ the Gaussian diffusion as our test problem

$$\phi(x, y, z, t) = (4\pi D t)^{-3/2} \exp\left(-\frac{x^2 + y^2 + z^2}{4tD}\right).$$

This solution describes heat diffusion from an initial point source in an infinite homogeneous medium. The computational domain for the test is chosen sufficiently large to avoid boundary effects during the simulation time.

We initialize the solution at $t = 1.0$ and evolve it numerically to $t = 2.0$, comparing the computed solution against the analytical expression.

4.1.1 Temporal Convergence

To assess the temporal accuracy of our method, we fix the spatial resolution and systematically refine the time step Δt. Figure 2 shows the L_2 error as a function of Δt on a log-log scale.

The results demonstrate clear second-order convergence. This confirms that our operator-splitting approach, combining the second-order RKC method and ADI, works correctly.

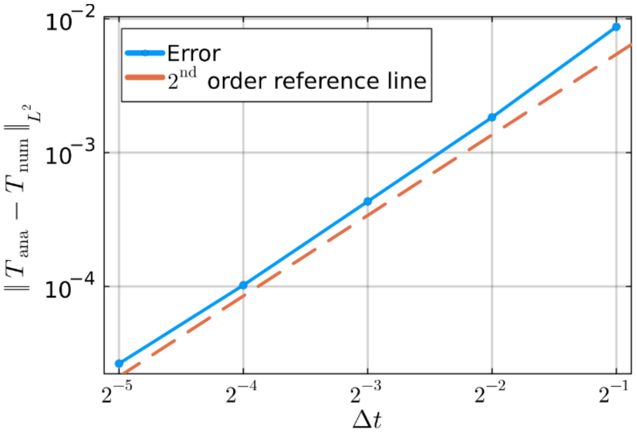

Figure 2: Temporal convergence study for the three-dimensional Gaussian diffusion test case. The L_2 error decreases as Δt^2, confirming second-order accuracy in time for our operator splitting scheme (advection not included in this test).

4.1.2 Spatial Convergence

To verify spatial accuracy, we refine the grid resolution uniformly while now fixing the time step size. Figure 3 presents the L_2 error versus the grid spacing Δx in each of the three spatial directions.

The spatial convergence study again confirms second-order accuracy, consistent with the second-order finite difference approximations employed for spatial derivatives.

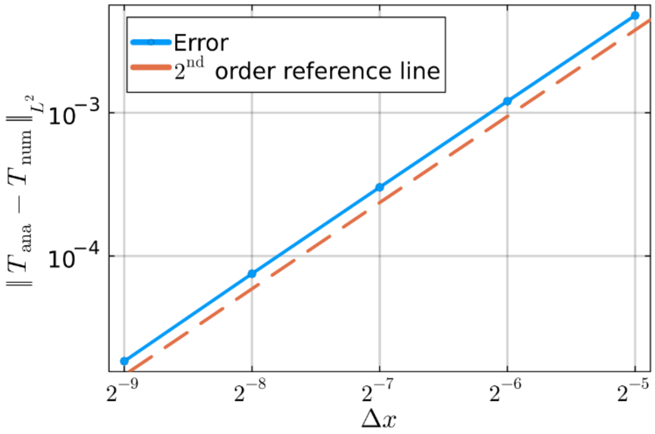

Figure 3: Spatial convergence study for the three-dimensional Gaussian diffusion test case. The L_2 error decreases as Δx^2, demonstrating second-order accuracy in space (advection not included in this test).

These convergence studies, achieving second-order accuracy in both space and time for the diffusion operator, provide confidence in our method's reliability for the more complex geometries encountered in the subsequent case studies.

4.2 Case Study 1: Comparison with Coupled 1D-2D Cylindrical Model

Li et al. [14] developed a coupled numerical model using cylindrical coordinates (r, z) for the surrounding rock and a one-dimensional representation for the borehole, solved via finite difference methods in MATLAB.

We reproduce their setup and simulate the configuration for 120 days of operation. Figure 4 compares the radial temperature profiles in the surrounding rock at depths of 500 m, 1000 m, 1500 m, and 2000 m. Our results show excellent agreement with those reported by Li et al., validating our approach for capturing heat diffusion into the surrounding rock formation.

The agreement degrades noticeably when we vary the extracted temperature difference ΔT from the heat exchanger coupling the inlet temperature to the outlet or simulate for durations other than 120 days. This sensitivity to operating conditions demonstrates that the good agreement at 120 days does not result merely from an inert thermal response.

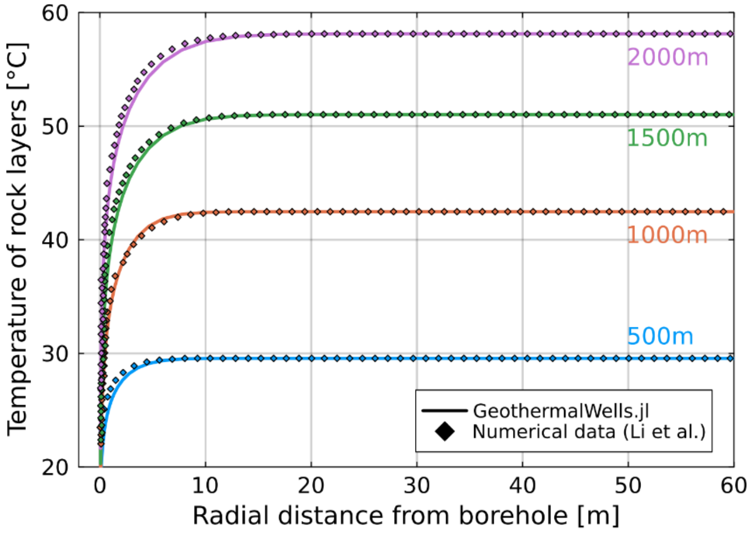

Figure 4: Comparison of radial temperature profiles in the surrounding rock after 120 days of operation at depths of 500m, 1000m, 1500m, and 2000m. Our simulation (solid lines) shows excellent agreement with the results of Li et al. [14] (diamonds).

The time taken for this 120-day simulation on a single NVIDIA H200 GPU was approximately 10 minutes.

4.3 Case Study 2: Comparison with Full 3D COMSOL Model

Hu et al. [10] employed a fully three-dimensional model implemented in COMSOL Multiphysics for simulating a single closed-loop coaxial deep borehole heat exchanger. Their model solves the coupled fluid flow and heat transfer equations using a k-epsilon turbulence model for the flow field and the Conjugate Heat Transfer module for thermal transport, with temperature-dependent thermodynamic properties for both the working fluid and reservoir rock. These temperature-dependent properties were not included in our model, as time-varying thermal conductivity and heat capacity introduced significant numerical instabilities. Additionally, our model does not resolve any fluid dynamics explicitly (see Section 2.4).

We simulated the same configuration described in their paper for 25 years of continuous operation. Figure 5 compares temperature profiles along the borehole depth in both the outer annulus and inner pipe at 1 year, 5 years, 10 years, and 25 years. Our results show good agreement with Hu et al.'s model despite the differences in approach. The good agreement at the top of the inlet is expected as a constant inlet temperature of 20°C is prescribed.

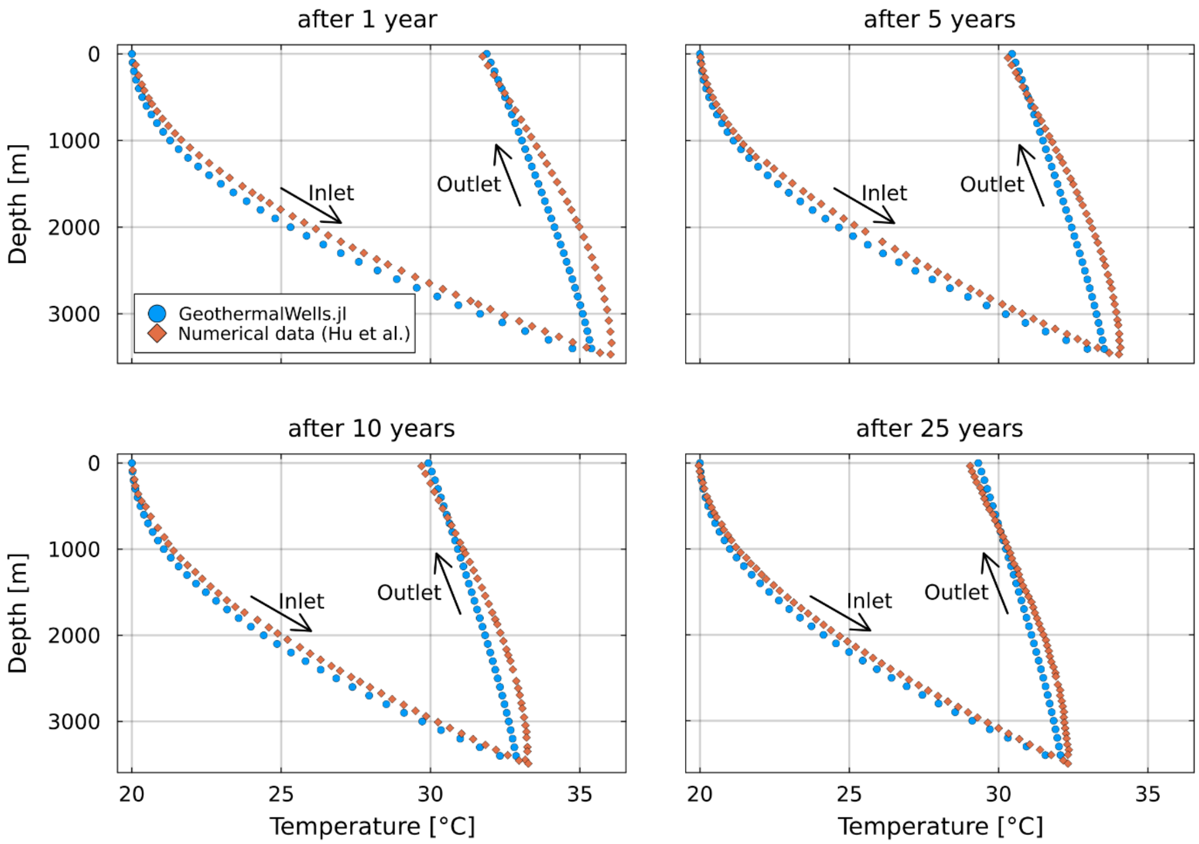

Figure 5: Temperature profiles along borehole depth in the outer annulus (inlet) and inner pipe (outlet) after 1 year, 5 years, 10 years, and 25 years of operation. Our simulation (blue dots) shows good agreement with the COMSOL results of Hu et al. [10] (orange diamonds).

The time taken for this 25-year simulation on a single NVIDIA H200 GPU was approximately 12.9 hours.

4.4 Case Study 3: Comparison with Dual-Continuum OpenGeoSys Model

Brown et al. [4] employed OpenGeoSys [11], a finite element model that uses a dual-continuum approach, treating the deep borehole heat exchanger as a one-dimensional discretized medium coupled to a three-dimensional representation of the surrounding rock.

We simulated the same configuration described in their paper for 20 years of continuous operation, examining both the temperature evolution within the borehole and the thermal impact on the surrounding rock formation.

The temperature profile of the surrounding rock, as shown in Figure 6, matches closely with our simulation. There appears to be a discrepancy only close to the borehole, where the temperature starts to decrease linearly, but this is most likely an artifact from the plotting of the data in Brown et al., where a continuous line is drawn through individual data points.

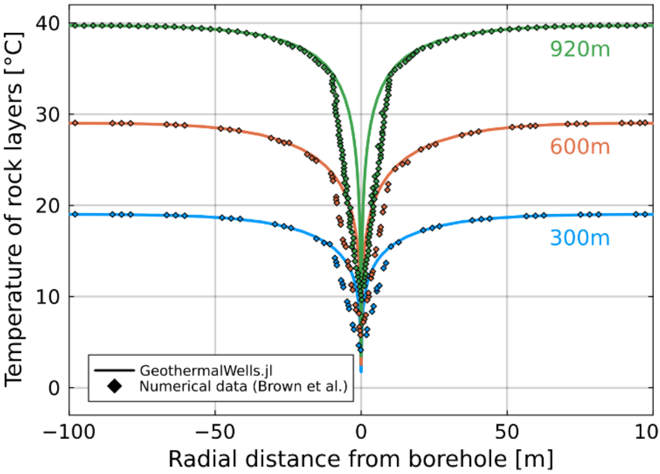

Figure 6: Radial temperature profiles in the surrounding rock at various depths after 20 years of operation. Our simulation (solid lines) matches the results of Brown et al. (diamonds) closely, with minor discrepancies near the borehole, which are most likely plotting artifacts.

Figure 7 compares the temperature profiles along the borehole depth in both the outer annulus and inner pipe after 20 years of operation. Here, our results show a visible discrepancy of about 1.5°C from the results obtained by Brown et al.

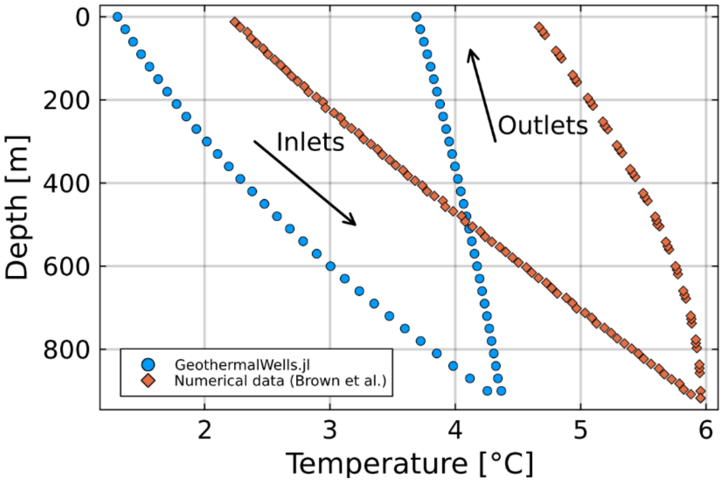

Figure 7: Temperature profiles along borehole depth in the outer annulus and inner pipe after 20 years of operation. Our simulation shows some discrepancies with the OpenGeoSys results obtained by Brown et al.

The time taken for this 20-year simulation on a single NVIDIA H200 GPU was approximately 8 hours.

5 SIMULATION OF WELL ARRAYS

The preceding validation studies established confidence in the numerical method for single-well configurations. However, the primary motivation for developing this framework is to enable simulations of well arrays, where thermal interactions between boreholes become important over long operational timescales. This section presents preliminary results demonstrating the feasibility and computational cost of such simulations. A systematic study of array performance and thermal interference effects is left for future work.

5.1 Simulation Times

We examine the computational time required to simulate one year of operation for different well array configurations. The setup follows that described in Section 4.2 varying only the number of boreholes. All simulations were performed on a single NVIDIA H200 GPU. As described in Section 3.1, the grid uses fine spacing near each borehole and progressively coarsens toward the far-field. Each borehole is resolved with identical fine grid spacing, ensuring consistent accuracy across all wells regardless of array size. For regular array configurations, boreholes share the coarse far-field regions between them, so the horizontal degrees of freedom ($N_x \times N_y$) scale sub-linearly with the number of wells (from approximately 33,000 per well for a single borehole to 23,000 per well for a 5×5 array).

However, as shown in Figure 8, this does not directly translate to computational time. For the small 3 × 1 array, the time per well is actually lower than for the single-well case, most likely attributable to the fixed overhead of GPU kernel launches being amortized over more wells. For larger arrays, however, this efficiency gain reverses: the time per well increases roughly linearly with array size, likely due to memory bandwidth saturation as the working dataset grows.

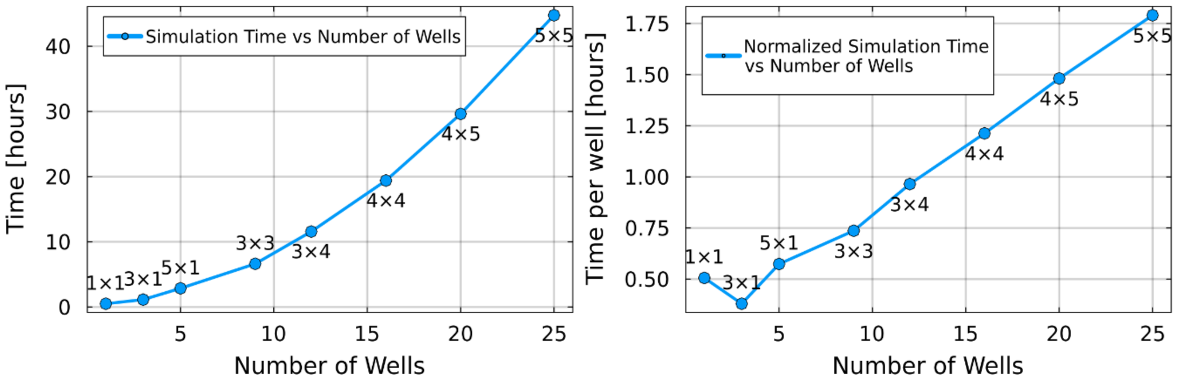

Figure 8: *Left*: The time taken to simulate one year of operation for different well array sizes on a single NVIDIA H200 GPU. The time per well increases with larger well arrays, likely due to memory bandwidth limitations. *Right*: The normalized time per well (total time divided by number of wells) against the number of wells in the array. There seems to be a linear relationship for larger arrays.

6 FUTURE WORK

Several directions for future work present themselves as natural extensions of this work:

- Investigate the role of the lack of explicit solid-fluid interface treatment and assess its impact on simulation accuracy.
- Implement multi-GPU parallelization to enable faster simulations of larger well arrays.
- Conduct systematic studies of well array performance, including thermal interference effects and optimal borehole spacing, potentially driven by agentic LLM workflows [20].
- Develop manufactured solutions for convergence tests using diffusion with variable coefficients together with advection.
- Train surrogate models based on the full three-dimensional model for fast exploration of parameter spaces in well array optimization studies.

7 CONCLUSION AND SUMMARY

We have developed a GPU-accelerated numerical method for the full three-dimensional simulation of deep coaxial borehole heat exchangers and well arrays. The approach employs an operator splitting strategy tailored to the multi-scale geometry and anisotropic grid structure of the problem, separating vertical diffusion (treated with stabilized explicit Runge–Kutta–Chebyshev methods), horizontal diffusion (handled via the alternating direction implicit scheme), and advection (solved using the semi-Lagrangian method). This combination achieves near-unconditional stability while maintaining computational efficiency.

We validated the operator splitting through convergence studies demonstrating second-order accuracy in both space and time for the pure diffusion case. Comparisons with three published models employing fundamentally different numerical approaches (coupled 1D-2D cylindrical, full 3D COMSOL, and dual-continuum finite element) show excellent to good agreement, establishing confidence in the method's reliability despite the simplified physical assumptions.

The vendor-agnostic implementation in Julia enables portable high-performance execution across different GPU architectures. Most importantly, this approach makes previously intractable simulations feasible: multi-year simulations of well arrays can now be performed on a single GPU, opening new possibilities for systematic design space exploration, optimization studies, and assessment of long-term thermal interference effects in multi-well configurations. The simulation framework, released as the open-source package GeothermalWells.jl [23] could provide a foundation for further development by the geothermal research community.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. National Science Foundation under award Nos CNS-2346520, RISE-2425761, and DMS-2325184, by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112490488, by the Department of Energy, National Nuclear Security Administration under Award Number DE-NA0003965 and by the United States Air Force Research Laboratory under Cooperative Agreement Number FA8750-19-2-1000. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof." The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government.

CW and HR were supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, project numbers 513301895 and 528753982).

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in our reproducibility repository at https://github.com/cwittens/2026_DBHEs_Arrays and archived at doi:10.5281/zenodo.18405213.

AI STATEMENT

AI assistance was used in writing this manuscript. The authors reviewed and approved all content and are fully responsible for the work presented herein.

REFERENCES

[1] A. Abdulle and A. A. Medovikov. “Second order Chebyshev methods based on orthogonal polynomials.” In: Numerische Mathematik 90.1 (2001), pp. 1–18. doi: 10.1007/s002110100292.

[2] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. “Julia: A Fresh Approach to Numerical Computing.” In: SIAM Review 59.1 (2017), pp. 65–98. doi: 10.1137/141000671.

[3] C. S. Brown, N. J. Cassidy, S. S. Egan, and D. Griffiths. “Numerical modelling of deep coaxial borehole heat exchangers in the Cheshire Basin, UK.” In: Computers & Geosciences 152 (2021), p. 104752. doi: 10.1016/j.cageo.2021.104752.

[4] C. S. Brown, I. Kolo, G. Falcone, and D. Banks. “Investigating scalability of deep borehole heat exchangers: Numerical modelling of arrays with varied modes of operation.” In: Renewable Energy 202 (2023), pp. 442–452. doi: 10.1016/j.renene.2022.11.100

[5] W. Cai, F. Wang, S. Chen, C. Chen, J. Liu, J. Deng, O. Kolditz, and H. Shao. “Analysis of heat extraction performance and long-term sustainability for multiple deep borehole heat exchanger array: A project-based study.” In: Applied Energy 289 (2021), p. 116590. doi: 10.1016/j.apenergy.2021.116590.

[6] S. Christ, D. Schwabeneder, C. Rackauckas, M. K. Borregaard, and T. Breloff. “Plots.jl a user extendable plotting API for the Julia programming language.” In: Journal of Open Research Software (2023). doi: 10.5334/jors.431.

[7] V. Churavy. KernelAbstractions.jl. doi: 10.5281/zenodo.4021259. url: https://github.com/JuliaGPU/KernelAbstractions.jl.

[8] R. Courant, K. O. Friedrichs, and H. Lewy. “Über die partiellen Differenzengleichungen der mathematischen Physik.” In: Mathematische Annalen 100.1 (1928), pp. 32–74. doi: 10.1007/BF01448839.

[9] L. C. Evans. Partial differential equations. en. Second. Vol. 19. Section 2.3.1. Providence, RI: American Mathematical Society, 2010.

[10] X. Hu, J. Banks, L. Wu, and W. V. Liu. “Numerical modeling of a coaxial borehole heat exchanger to exploit geothermal energy from abandoned petroleum wells in Hinton, Alberta.” In: Renewable Energy 148 (2020), pp. 1110–1123. doi: 10.1016/j.renene.2019.09. 141.

[11] O. Kolditz, S. Bauer, L. Bilke, N. Bottcher, J. Delfs, T. Fischer, U. Gorke, T. Kalbacher, G. Kosakowski, C. Mcdermott, C. Park, F. Radu, K. Rink, H. Shao, H. Shao, F. Sun, Y. Sun, A. Singh, J. Taron, M. Walther, W. Wang, N. Watanabe, Y. Wu, M. Xie, W. Xu, and B. Zehner. “OpenGeoSys: an open-source initiative for numerical simulation of thermos-hydro-mechanical/chemical (THM/C) processes in porous media.” English. In: Environ mental Earth Sciences 67.2 (Sept. 2012), pp. 589–599. doi: 10.1007/s12665-012-1546-x.

[12] I. Kolo, C. S. Brown, W. Nibbs, W. Cai, G. Falcone, T. Nagel, and C. Chen. “A comprehensive review of deep borehole heat exchangers (DBHEs): subsurface modelling studies and applications.” en. In: Geothermal Energy 12.1 (2024). doi: 10.1186/s40517-024-00297-3.

[13] R. J. LeVeque. Finite Difference Methods for Ordinary and Partial Differential Equations. Sections 2.15 and 2.18. Society for Industrial and Applied Mathematics, 2007. doi: 10.1137/1.9780898717839.

[14] J. Li, W. Xu, J. Li, S. Huang, Z. Li, B. Qiao, C. Yang, D. Sun, and G. Zhang. “Heat extraction model and characteristics of coaxial deep borehole heat exchanger.” In: Renewable Energy 169 (2021), pp. 738–751. doi: 10.1016/j.renene.2021.01.036.

[15] D. W. Peaceman and H. H. Rachford Jr. “The Numerical Solution of Parabolic and Elliptic Differential Equations.” In: Journal of the Society for Industrial and Applied Mathematics 3.1 (1955), pp. 28–41. doi: 10.1137/0103003.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes 3rd Edition: The Art of Scientific Computing. 3rd ed. Section 2.4. USA: Cambridge University Press, 2007.

[17] C. Rackauckas and Q. Nie. “DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia.” In: The Journal of Open Research Software 5.1 (2017). doi: 10.5334/jors.151.

[18] A. Staniforth and J. Côté. “Semi-Lagrangian integration schemes for atmospheric models — A review.” en. In: Mon. Weather Rev. 119.9 (Sept. 1991), pp. 2206–2223. doi: 10.1175/1520- 0493(1991)119<2206:SLISFA>2.0.CO;2.

[19] G. Strang. “On the Construction and Comparison of Difference Schemes.” In: SIAM Journal on Numerical Analysis 5.3 (1968), pp. 506–517. doi: 10.1137/0705041.

[20] Ouko, E., Lujan, E., Edelman, A., and Metcalfe, R.: Decision-Support and Modeling with Large Language Models for Geothermal Well Arrays, Proceedings, 50th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA (2025).

[21] Metcalfe, R., Lujan, E., Wittenstein C., Inglis A., Edelman A.: Arrays of Networked Standard Geothermal Wells, Proceedings, 51st Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA (2026).

[22] C. Wittenstein, E. Lujan, A. Inglis, R. Metcalfe, A. Edelman, and H. Ranocha. Reproducibility repository for "A Full Three-Dimensional GPU-Accelerated Model for Deep Borehole Heat Exchangers (DBHEs) Enabling Simulation of Well Arrays". 2026. doi: 10.5281/zenodo.18405213. url: https://github.com/cwittens/2026_DBHEs_Arrays.

[23] C. Wittenstein. GeothermalWells.jl: GPU-accelerated simulation of deep borehole heat exchanger (DBHE) arrays. 2026. doi: 10.5281/zenodo.18405325. url: https://github.com/cwittens/GeothermalWells.jl.