Turning a Serial Forward Code into a Parallel Inverse Code: A Case Study from Geothermal Engineering

H. Martin Bücker, M. Ali Rostami, and Ralf Seidler

***Abstract*—**Significant human effort is spent in developing numerical simulation codes that represent a forward problem in computational science and engineering. Typically, the only focus of these developments is on issues of the forward model, exclusively addressing how the output of interest is efficiently and accurately simulated for given input parameters. Though these issues may be numerous and intricate we argue that, most likely, the forward model will later also be used in a framework addressing inverse problems. Therefore, any design of a forward model should consider issues related to inverse modeling as well. We show by example of the geothermal engineering code SHEMAT that this requires to rethink design decisions that were originally taken for the forward model.

***Keywords*-**Inverse problems, optimal experimental design, automatic differentiation, software engineering, message passing, OpenMP, SHEMAT.

I. Introduction

ADVANCED mathematical models to describe and analyze complicated phenomena of interest have led to a tremendous increase of the understanding in various disciplines, including physical, engineering, financial, and life sciences. There is also a long tradition in transforming these mathematical models into computer models to be efficiently executed on state-of-the-art computer architectures. These substantial advances were only possible by bringing together profound knowledge and sophisticated techniques from applied mathematics, computer science, and the corresponding application disciplines.

Often, the design and implementation of these forward models require substantial human effort that is justified by the scientific findings resulting from using these models to better understand the given application problem at hand. It is not uncommon to use these forward models over and over again while varying certain input parameters of interest. However, these computer models are increasingly used as the starting point for further, more advanced investigations that go far beyond executing the forward model multiple times with different input parameters. The design of these forward models should therefore consider the requirements of an inverse framework as well. Unfortunately, today, most forward models are designed without taking into account that they will be used to solve inverse problems.

Here, we focus on the interplay between parallel computing and automatic differentiation when designing a forward model.

H. M. Bücker, M. A. Rostami, and R. Seidler are with the Department of Mathematics and Computer Science, Friedrich Schiller University Jena, Jena, Germany.

Manuscript received January 31, 2014; revised March 31, 2014.

To the best of our knowledge, there is no previous publication dealing with this topic. Somehow related are approaches that try to reduce the computational complexity of a forward model that is used in an optimization framework. An example in the context of a geoscientific application is the approach in [1] that introduces a simplification of forward models in magnetotellurics to enable evolutionary optimization algorithms in three space dimensions. However, the reduction of the computational complexity is not the focus of the present article. The new contribution of the present article is to demonstrate the mutual dependencies between parallel computing used in various different ways for a forward model and automatic differentiation.

The outline of this paper is as follows. We first give a short overview on a forward model developed for the solution of geothermal engineering problems in Sect. II. We then show how the forward model is used in the context of inverse problems in Sect. III. Finally, in Sect. IV we describe the enabling technologies that need to fit together for the solution of problems arising from geothermal engineering.

II. Forward Modeling

The Institute for Applied Geophysics and Geothermal Energy, E.ON Energy Research Center, at RWTH Aachen University is currently developing a geothermal simulation package. This software is called Simulator for HEat and MAss Transport (SHEMAT) [2], [3]. It solves the coupled transient equations for groundwater flow, heat transport, and the transport of reactive solutes in porous media at high temperatures in three space dimensions. Here, we consider a mathematical model for fluid flow [2] and heat transport. The equation of fluid flow for the hydraulic potential (head) h_0 at location (x, z) and time t is given by

$$\rho_f g(\alpha+\psi\beta)\frac{\partial h_0}{\partial t} - \nabla\cdot\left(\frac{\rho_f g\kappa}{\mu_f}(\nabla h_0 + \rho_r \nabla z)\right) = W. \quad (1)$$

Here, ρ_f and ρ_r denote the fluid and the rock density, α and β represent the compressibilities of rock and fluid phase, respectively, and ψ is the porosity. The hydraulic permeability tensor is denoted by κ, while the fluid dynamic viscosity is represented by μ_f. The symbol g is used for the gravitational acceleration and W corresponds to a mass source term due to externally inflowing water.

ISBN: 978-1-61804-233-0

The temperature T is governed by the conductive-advective heat transport equation

$$(\rho c)_e \frac{\partial T}{\partial t} - \nabla \cdot (\lambda_e \nabla T) - (\rho c)_f \, \mathbf{a} \cdot \nabla T = H, \qquad (2)$$

where $(\rho c)_e$ denotes the effective heat capacity of the saturated porous medium and the fluid, λ_e is the effective thermal conductivity, $(\rho c)_f$ represents the volumetric heat capacity of the fluid, and H corresponds to a possible heat source term. Equations (1) and (2) are coupled via the Darcy velocity $\mathbf{a}$ satisfying Darcy's law

$$\mathbf{a} = -\frac{\kappa}{\mu_f}(\nabla P + \rho_f \, g \nabla z), \qquad (3)$$

where the pressure P depends on the head h_0. Given suitable initial and boundary conditions, a numerical scheme for the solution of (1)–(3) computes approximations of head h_0 and temperature T.

The numerical scheme implemented in SHEMAT consists of a finite-difference approach. After the discretization of the domain, the following iteration for the solution to the head equation is presented in [2]

$$-(1-\omega)K \cdot h_{\text{old}} - R \cdot h_{\text{old}} - W = \omega K \cdot h_{\text{new}} - R \cdot h_{\text{new}}, \quad (4)$$

in which ω is the time weighting parameter controlling the implicity or explicity of the solution approach, the symbol R is given by

$$R = \frac{\rho_f g(\alpha + \psi\beta)}{\Delta t},$$

and K is the conductivity matrix. This matrix is computed via

$$K = Ah_{k-1} + Bh_{j-1} + Ch_{i-1} + Dh + Eh_{i+1} + Fh_{j+1} + Gh_{k+1},$$

where A, B, C, D, E, F and G are the coefficients achieved by discretization and $h_{i,j,k}$ refers to the head in the three-dimensional cell (i, j, k). In this formula, the the indices of the head h are simplified in an obvious fashion, for example h_{i+1} means $h_{i+1,j,k}$ and h is used to denote $h_{i,j,k}$.

Equation (4) solves the steady-state flow equation. For a solution of the more general case, another time step iteration is needed. The following algorithm is a fixed-point method for this nonlinear problem implemented in SHEMAT.

1: **for** $t = 0, \Delta t, t_{\text{end}}$ **do**
2: Initial guess for h at the current time step, $h^t = h_0^{t+\Delta t}$, usually using the head value from the previous step
3: Compute time step control Δt
4: **for** $k = 1, 2, ..., \max$ (Fixed-point iteration) **do**
5: Setup the coefficient matrix $M(h_{k-1}^{t+\Delta t})$
6: The right-hand side: $B = -(1-\omega)Mh^t - Rh^t - W$
7: Solve $Mh_k^{t+\Delta t} = B$ for $h_k^{t+\Delta t}$
8: Compute the residuals: $RE_k = |h_k^{t+\Delta t} - h_{k-1}^{t+\Delta t}|$
9: Adaptive Relaxation: $h_k^{t+\Delta t} = (1 - \delta_k)h_{k-1}^{t+\Delta t} + \delta_k h_k^{t+\Delta t}$
10: **if** $\max |RE_k| \geq \epsilon$ **then**
11: $h^t = h_k^{t+\Delta t}$ and go to the step 3
12: **end if**
13: **end for**
14: **end for**

III. Going beyond Forward Modeling

Once a sophisticated forward model is available, a typical situation is to address—in some form or the other—an inverse problem related to that forward model. A prominent example is a parameter estimation problem. That is, the input parameters to the simulation are estimated so that they represent the real world as close as possible. In particular, these problems are ubiquitous in the geosciences. In that application area, a large number of properties of the subsurface are to be estimated from only a few data points obtained from some measurements.

More formally, a forward model computes approximations of, say, the temperature T at spatial position x for some unknown input parameters p. Throughout this article, we assume that the unknown parameter p is the hydraulic permeability κ. Conceptually, this forward model, which is described by (1)–(3) together with appropriate boundary and initial conditions, is nothing but a function

$$f(x, p). \qquad (5)$$

So, a parameter estimation problem is represented by minimizing a suitably-defined distance function between some measurements $m(x)$ and the solution of the forward problem $f(x, p)$. That is, the unknown parameters p are obtained from the solution of

$$\min_p \|m(x) - f(x, p)\|, \qquad (6)$$

where $\|\cdot\|$ is some norm.

In the context of SHEMAT, various algorithms for the solution of different inverse problems are implemented. These algorithms that make use of the forward model fall into the classes of stochastic techniques and deterministic techniques.

Stochastic or Monte-Carlo-like methods offer the advantage that large systems can be sampled in a number of random configurations, and that the data obtained from multiple forward models can be used to describe the system as a whole. In the context of SHEMAT, several stochastic methods are implemented, including sequential Gaussian simulation and ensemble Kalman filtering techniques [4]. These techniques make use of the forward model only in the form of evaluations of the forward model. A disadvantage of these techniques is that they may require a large number of these evaluations. The textbook [5] covers stochastic methods from a geoscientific point of view.

Deterministic methods for the solution of inverse problems tend to involve a smaller number of evaluations of the forward model. However, in addition to the evaluation of the forward model, they also need the derivatives of the forward model. For instance, a deterministic algorithm for the solution of (6) requires the evaluation of

$$\frac{\partial f(x, p)}{\partial p}. \qquad (7)$$

See the textbooks [6], [7], [8] for more details on these methods.

Optimal Experimental Design (OED) refers to a set of techniques to minimize the uncertainty in a parameter estimation

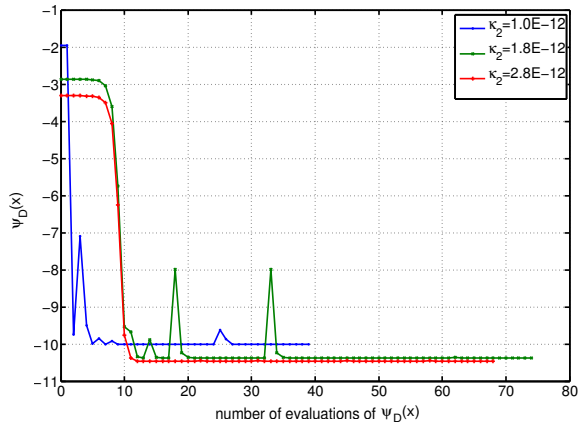

Fig. 1. Values of the objectives Ψ_D plotted versus the number of evaluations of Ψ_D carried out during the optimization process. Each curve corresponds to a different permeability value κ. See [9] for more details.

problem with respect to some experimental conditions. In the context of geothermal engineering, one can ask for questions

- Where to drill a borehole?
- When to take the measurements?

such that the uncertainty when estimating the hydraulic permeability in a certain rock layer from temperature measurements in a borehole is minimized.

More formally, OED solves an optimization problem of the form

$$\min_x \Psi(x) \tag{8}$$

where the objective $\Psi(x)$ quantifies the uncertainty of an experimental condition x. In the above example, the experimental condition is the position of a borehole. Typical objectives are

$$\Psi_A(x) = \log\bigl(\mathrm{trace}(F^{-1})\bigr)$$
$$\Psi_D(x) = -\log\bigl(\det(F)\bigr),$$

where

$$F = \left(\frac{\partial f(x,p)}{\partial p}\right)^T \left(\frac{\partial f(x,p)}{\partial p}\right) \tag{9}$$

is the so called Fisher Matrix. It is important to observe that gradient-based algorithms for the solution of (8) need the forward model not only in the form of evaluations of $f(x,p)$, and first-order derivatives like (7); they also require evaluations of mixed second-order derivatives

$$\frac{\partial^2 f(x,p)}{\partial p\,\partial x}. \tag{10}$$

As an example we consider the OED problem addressed in [9]. The problem there is to find the location to drill a borehole such that the uncertainty to estimate the hydraulic permeability from temperature measurements taken in the borehole is minimized. A geothermal reservoir in western Australia is considered in two space dimensions. Figure 1 shows one result of that particular discussion.

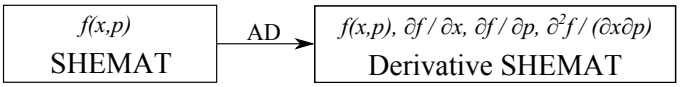

Fig. 2. The overall principle of automatic differentiation (AD) where a new program for the derivatives of the forward model is generated mechanically from the program of the forward model.

IV. Enabling Technologies

To successfully solve the above mentioned optimization problems, various building blocks are necessary. First of all, a forward model $f(x,p)$ needs to be developed. However, the development of forward problems—though complicated, challenging, and often needing considerable human effort—is a "standard procedure." Therefore, we do not address this building block in this article.

Another building block for the algorithms sketched in the previous discussion is the availability of derivatives of the forward model. More precisely, we need first-order derivatives of the form (7) and mixed second-order derivatives (10). One option is to derive these derivatives in an analytic way and implement another computer model for these derivatives by hand. Since this requires too much human effort for most real-world problems, we do not consider this to be feasible. Numerical differentiation by divided differencing is another option, but this inherently involves truncation error so that the derivatives are always inaccurate. Therefore, we rely on derivatives computed by automatic differentiation (AD) [10], [11]. Given a program to evaluate the forward model $f(x,p)$, the set of AD techniques mechanically generate another program for the evaluation of its derivatives without truncation error. The overall AD transformation process is schematically depicted in Fig. 2.

As soon as the problem sizes increases, for instance by increasing the resolution of the grid or the number of parameters to be estimated, the time to compute the overall solution process becomes prohibitively large. Often, the storage requirement exceeds the amount of available memory on a conventional computer architecture too. So, it is quite common that parallelism has to be introduced to solve these challenging problems. SHEMAT involves various different forms of parallelism. From the programming model point of view, it is parallelized with MPI [12] for distributed-memory computers as well as with OpenMP [13] for shared-memory computers.

In particular, the Monte Carlo methods are parallelized using OpenMP. The reason is that these methods rely on (mostly) independent evaluations of the forward model at different input values. This inherently-parallel structure is easily parallelized with OpenMP which is the de facto standard for shared-memory parallel programming. For the Monte Carlo methods, OpenMP is used outside of the forward model. We stress that there are parts inside the forward model that are also parallelized with OpenMP.

Shared-memory computers do not scale to a very large number of processes. Distributed-memory computers allow to increase the number of processors further. Therefore, the forward model of SHEMAT is currently redesigned to also

involve parallelization based on MPI. This is tricky since the usual hybrid parallel programming paradigm is based on using OpenMP from within an MPI process. However, in the Monte Carlo methods of the current SHEMAT version, there would be need to use a hybrid MPI/OpenMP program within a parallel OpenMP region which is currently considered to be unreasonable.

Furthermore, there is currently no AD software that is capable of reliably transforming programs using MPI and/or OpenMP. Our strategy is therefore to implement a library that hides most of these parallel programming constructs form the AD transformation process. There is also another problem arising from the various levels of parallelism currently implemented in SHEMAT. Given a serial version of the forward model, there is a way to automatically parallelize the derivatives in the AD-generated code. This parallelization strategy is orthogonal to the parallelization strategy used to parallelize multiple instances of forward solvers in the Monte Carlo methods [14]. However, both these parallelization strategies need to work together. Moreover, they also need to be compliant to the new MPI parallelization of the forward model.

V. Conclusion

If a forward model is used in the context of an inverse problem, there are many intricate issues that have to be considered. In particular, we focus on a case study from geothermal engineering where a serial code for the forward model is turned into a parallel code used in an inverse framework. In fact, some parts of the forward model are already parallelized. We demonstrate the need for a clear design of the forward model and show the various implications that make difficult the redesign of this forward code in the context of its integration into a parallel inverse framework. The major difficulties arise from the various levels of parallelism. Some of these parallel levels are orthogonal to each other while others are nested and, thus, involve dependencies. Another challenge is to cope with parallelism in combination with automatic differentiation.

Acknowledgment

The authors thank our colleagues from the E.ON Energy Research Center and the Institute for Mathematics (Continuous Optimization), both at RWTH Aachen University, for a fruitful collaboration. This work is partially supported by the German Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU) within the project MeProRisk II, contract number 0325389 (F) as well as by the German Federal Ministry of Education and Research (BMBF) within the project HYDRODAM, contract number 01DS13018.

References

[1] B. Alexander, S. Thiel, and J. Peacock, "Application of evolutionary methods to 3D geoscience modelling," in *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference*, ser. GECCO '12. New York, NY, USA: ACM, 2012, pp. 1039–1046. [Online]. Available: http://doi.acm.org/10.1145/2330163.2330308

[2] J. Bartels, M. Kühn, and C. Clauser, "Numerical simulation of reactive flow using SHEMAT," in *Numerical Simulation of Reactive Flow in Hot Aquifers*, C. Clauser, Ed. Springer Berlin Heidelberg, 2003, pp. 5–74.

[3] V. Rath, A. Wolf, and H. M. Bücker, "Joint three-dimensional inversion of coupled groundwater flow and heat transfer based on automatic differentiation: Sensitivity calculation, verification, and synthetic examples," *Geophysical Journal International*, vol. 167, no. 1, pp. 453–466, 2006.

[4] C. Vogt, G. Marquart, C. Kosack, A. Wolf, and C. Clauser, "Estimating the permeability distribution and its uncertainty at the EGS demonstration reservoir Soultz-sous-Forêts using the ensemble Kalman filter," *Water Resources Research*, vol. 48, no. 8, p. W08517, 2012. [Online]. Available: http://dx.doi.org/10.1029/2011WR011673

[5] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[7] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: John Wiley & Sons, 1987.

[8] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs: Prentice-Hall, 1983.

[9] R. Seidler, H. M. Bücker, K. Padalkina, M. Herty, J. Niederau, G. Marquart, and A. Rasch, "Redesigning the EFCOSS framework towards finding optimally located boreholes in geothermal engineering," accepted for publication at TMCE 2014, Budapest.

[10] L. B. Rall, *Automatic Differentiation: Techniques and Applications*, ser. Lecture Notes in Computer Science. Berlin: Springer, 1981, vol. 120.

[11] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed., ser. Other Titles in Applied Mathematics. Philadelphia, PA: SIAM, 2008, no. 105.

[12] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-passing Interface*, ser. Scientific and engineering computation. MIT Press, 1999, no. 1.

[13] B. Chapman, G. Jost, and R. van der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming*, ser. Scientific Computation Series. MIT Press, 2008, no. 10.

[14] A. Wolf, "Ein Softwarekonzept zur hierarchischen Parallelisierung von stochastischen und deterministischen Inversionsproblemen auf modernen ccNUMA-Plattformen unter Nutzung automatischer Programmtransformation," Dissertation, Department of Computer Science, RWTH Aachen University, 2011.