



Automatic adjoints without the algebra: flexible sensitivity analysis of nonlinear problems in the geosciences

Jacob Frasunkiewicz¹, Christian Schuler^{1,2}, Evangelos Moulas^{1,3}, and Boris J.P. Kaus^{1,3}

¹Institute of Geosciences, Johannes Gutenberg University Mainz, Mainz, Germany

²Institute of Earth Sciences, University of Lausanne, Lausanne, Switzerland

³Mainz Institute of Multiscale Modeling (M³ODEL), Johannes Gutenberg University Mainz, Mainz, Germany

Correspondence: Jacob Frasunkiewicz (jafraun@uni-mainz.de)

Abstract. Material parameter values used in geoscientific simulations are often poorly constrained due to the sparse nature of the observations. In addition, the constitutive relationships used to describe natural processes typically involve nonlinearities, which results in large changes in simulation output when material parameters or initial conditions change. It is therefore important to have a tool that can efficiently determine the quantitative effect of parameter changes on the simulation output. The adjoint gradient method addresses this by delivering the gradient of an observation, here the result of the forward simulation, with respect to all material parameters at a computational cost independent of their number. Deriving the adjoint gradient analytically can be labor-intensive and in some cases must be re-derived when the constitutive relationships change. Here, we present a framework that avoids the re-derivation by applying automatic differentiation directly to the code, which is used for the forward simulation. The residual calculation contains all discrete operations needed to assemble the Jacobian of the forward problem and the parameter sensitivity matrix. The full gradient with respect to all spatially varying parameters then follows from a single solution of the (linear) adjoint problem, regardless of the dimension of the parameters space. We additionally give a geometric interpretation of the discrete calculation of the adjoint gradients, making its computational structure explicit to guide implementation in other codes. The framework is implemented in Julia and demonstrated on four problems of increasing complexity: transient heat diffusion in a petrological experimental setup, coupled fluid flow and heat transport in a Central European geothermal reservoir, thermo-visco-elasto-plastic Stokes flow applied to a rising mantle plume, and to localizing shear bands. We deliberately emphasize flexibility and transparency over computational efficiency, providing a compact and reproducible basis for sensitivity analysis.

1 Introduction

Geoscientific computational models have grown substantially in complexity and computational scale over the past decades.

State of the art simulations often couple multiple physical processes such as mechanics, fluid flow, chemical reactions and heat transport across domains spanning several magnitudes in length and time (e.g. Gerya, 2019; Turcotte and Schubert, 2014).

Each additional process brings its own set of governing equations, material parameter distributions and constitutive equations, resulting in numerous parameters whose values are not directly observable in nature.



This expansion in model complexity is not matched by the same growth in observational constraints. Geological and geophysical observations remain fundamentally sparse as we are limited to surface-near observations. Boreholes only sample discrete points in depth, rock samples record complicated histories that must be recovered through inversion of sparse, uncertain data (e.g. Moutzouris et al., 2026)., surface observations such as GPS displacements or heat flow measurements are separated from the processes that generate them by kilometers of uncertain material (Vasco and Mali, 2020; Smith and Vasco, 2021). Methods such as seismic, geodetic and magnetotelluric surveys provide integrated signals that must be inverted to recover subsurface structure, which also carries uncertainty (Virieux and Operto, 2009; Siripunvaraporn, 2012). Even where measurements exist, they carry uncertainties from the instruments, processing and natural variability. In several cases, the material parameters required to solve the governing equations are often known to within an order of magnitude, if at all. Addressing this gap between simulations and data requires quantitative tools for determining how strongly a given observation depends on a given material parameter.

A central object in this analysis is the sensitivity kernel. The sensitivity kernel is a measure of how much a scalar quantity of interest, often the misfit between model and data, responds to a change in the model parameters (Plessix, 2006; Givoli, 2021). If there is only a single material parameter, then the sensitivity kernel is also just a scalar value. But if the material parameter varies in space, then the sensitivity kernel highlights which regions of the model most strongly control the observation. These kernels underpin a range of geoscientific applications from inverse problems (Tromp et al., 2004; Ghattas and Willcox, 2021) and model calibration (Jansen, 2011) to data assimilation (Errico, 1997) and experimental design (Alexanderian, 2021).

It is useful to distinguish between local and global sensitivities in this context. Local sensitivities are derivatives evaluated at a specific state of the model and quantify the response of small perturbations around that state (Givoli, 2021; Plessix, 2006). Changes in the material parameters may lead to significantly different simulation results and in turn the local sensitivity kernels of the prior simulation are no longer valid. Global sensitivities, on the other hand, aim to characterize how model output responds across the full range of plausible parameter values and typically requires a large number of simulations to explore the entire parameter space (Campolongo et al., 2000; Degen and Wellmann, 2025). This makes global sensitivities highly informative, but also computationally expensive for large parameter spaces, whereas local sensitivities always scale linearly to the problem size. This work focuses on local sensitivities, with the understanding that they are the building block for iterative procedures, such as Gauss-Newton, enabling practical inversions in high-dimensional parameter spaces (Ghattas and Willcox, 2021; Villa et al., 2021).

Computing local sensitivities naively via finite differences requires one forward solve per parameter, which scales poorly for high-dimensional parameter spaces (Plessix, 2006; Reuber et al., 2018). The adjoint method circumvents this scaling by reformulating the sensitivity calculation as a single additional solution of a linear system (i.e. linear solve), yielding the full gradient of the observation with respect to all parameters at a cost that is effectively independent of material parameters (Griewank and Walther, 2008). The method has a long history in the geosciences, with applications in seismic full-waveform inversion (Tromp et al., 2004; Fichtner et al., 2006), mantle convection and retrodiction (Bunge et al., 2003; Liu and Gurnis, 2008; Horbach et al., 2014; Ghelichkhan and Bunge, 2016; Colli et al., 2020), plate coupling and rheology inversion (Ratnaswamy et al., 2015; Li et al., 2017; Reuber et al., 2018, 2020), subsurface fluid flow and geothermal reservoir characterization (Jansen, 2011; Tian



et al., 2024; Scott et al., 2026), quasi-static geodetic deformation (Vasco and Mali, 2020; Smith and Vasco, 2021; Puel et al., 2024), and ice sheet dynamics (MacAyeal, 1992; Utkin et al., 2025). Recent years have seen a renewed interest in adjoint methods driven by advances in computational infrastructure and the maturation of differentiable programming frameworks, which have made it possible to apply adjoint based inversion to problems of increasingly physical and geometrical complexity (Ghelichkhan et al., 2024; Reuber, 2021; Utkin et al., 2025; Scott et al., 2026).

Two broad strategies exist for deriving the adjoint of a given problem. In the continuous, or optimize-then-discretize, approach, the adjoint equations are derived at the level of the governing equations and subsequently discretized (Plessix, 2006; Gunzburger, 2000). In the discrete, or discretize-then-optimize, approach, the adjoint is derived directly from the already discretized forward problem (Giles and Pierce, 2000). The two strategies are equivalent in the limit of vanishing discretization error, but differ in practical implementation. The continuous adjoint yields equations that are often "cleaner" to analyze and "cheaper" to solve, while the discrete adjoint is exact with respect to the discretized forward problem (Giles and Pierce, 2000). Additionally, changes in the forward problem often require complete re-derivation of the continuous adjoint equations, which makes keeping them in sync with a forward code that continues to evolve labor-intensive.

Automatic differentiation (AD) fundamentally changes this picture. By systematically applying the chain rule to the discrete operations executed in the forward code, AD tools produce exact derivatives of the implemented equations without requiring the user to derive them by hand (Gunzburger, 2000; Baydin et al., 2018; Margossian, 2019). This shifts the effort from mathematical derivation to code implementation, yields a problem-agnostic adjoint, and makes the discretize-then-optimize approach viable for complex, coupled, nonlinear multiphysics problems. Changes to the forward solver are automatically reflected in the adjoint, simplifying code maintenance.

Within AD, two evaluation modes are particularly relevant. Forward-mode AD computes Jacobian-vector products (JVPs), propagating perturbations from inputs to outputs, making it most efficient when the number of inputs is small relative to the number of outputs (Baydin et al., 2018; Griewank and Walther, 2008). Reverse-mode AD computes Vector-Jacobian products (VJPs), propagating sensitivities from the outputs back to the inputs and is efficient in the opposite regime (Baydin et al., 2018; Griewank and Walther, 2008). This is precisely regime of adjoint sensitivity analysis, where a single scalar cost function depends on many parameters (Baydin et al., 2018; Margossian, 2019). In practice the two modes play complementary roles, where forward-mode AD is well suited for assembling the Jacobian Matrix, $\mathbf{J}$, required for the solution of the nonlinear problem using Newton's method (forward Newton solve), while reverse mode AD is natural for the sensitivity calculation itself. In the following we utilize exclusively forward mode AD, because we make use of assembled Jacobians in both the forward and adjoint problems. Differentiable programming frameworks in languages like Julia and Python are now well established (Revels et al., 2016; Innes et al., 2019; Rackauckas et al., 2021; Moses and Churavy, 2020).

In this work, we present a flexible framework for solving nonlinear multiphysics problems in the geosciences and computing adjoint-based sensitivity kernels using AD. The framework is implemented in Julia and demonstrated on examples of increasing complexity, from thermal diffusion to viscoelastoplastic localisation. The same adjoint workflow, using a Newton solver for the forward problem, assembling Jacobian matrices with AD and linear adjoint solve, is applied across all cases without modification. We emphasize flexibility over efficiency, to highlight a compact workflow that can be applied over fundamen-



95 tally different problems rather than the implementing the fastest workflow possible, losing transparency of the underlying framework. In addition, we attempt to make the geometric and computational structure of the adjoint method as transparent as possible to guide implementation in other codes.

2 Solving non-linear Forward Problems

Geoscientific models are typically based on conservation laws of mass, momentum, or energy. These laws are expressed as partial differential equations (PDEs) that relate physical quantities such as temperature, pressure, velocity, or porosity. To solve
100 these equations numerically, we discretize them in space (and time), replacing derivatives with algebraic approximations. This transforms the PDE system into a system of algebraic equations defined at discrete grid points. In this way we could calculate adjoint sensitivities as it is done for systems of coupled algebraic equations. The discrete form of the PDE system $\mathbf{G}$ is a function of the state variables $\mathbf{u}$ and the material parameters $\mathbf{m}$ and has the general form

$$\mathbf{G}(\mathbf{u}, \mathbf{m}) = 0, \quad \text{with } \mathbf{u} \in \mathbb{R}^N, \mathbf{m} \in \mathbb{R}^M. \quad (1)$$

105 The goal is to find the solution of the state variable vector $\mathbf{u}$ such that all equations are satisfied simultaneously. In the simplest case the discretized equations are linear in the unknowns and have the form

$$\mathbf{R}(\mathbf{u}, \mathbf{m}) = \mathbf{J}(\mathbf{m})\mathbf{u} - \mathbf{b}(\mathbf{m}) = 0, \quad \text{with } \mathbf{J} \in \mathbb{R}^{N \times N}, \mathbf{b} \in \mathbb{R}^N \quad (2)$$

In this case $\mathbf{J}$ is a matrix of coefficients that calculates the state variables $\mathbf{u}$ from the state variables of the previous time step $\mathbf{b}$. This system can be solved using a single linear solve.

110 However, most geoscientific problems are not linear. Material properties often depend on the solution itself — for example, viscosity may depend on temperature or stress, permeability may depend on porosity, and plastic yielding introduces threshold behavior. In such cases, the coefficient matrix itself depends on the unknown solution. The system can be linearized and takes the general form

$$\mathbf{R}(\mathbf{u}, \mathbf{m}) = 0, \quad \text{with } \mathbf{R} \in \mathbb{R}^N \quad (3)$$

115 where $\mathbf{R}$ is now a nonlinear function of $\mathbf{u}$ and $\mathbf{m}$. The residual vector $\mathbf{R}$ measures imbalance of the discretized equations. For each gridpoint the residual represents the difference between the left- and right-hand sides of the governing equation. A solution is obtained when $\mathbf{R} = 0$. Because the system is nonlinear, it can no longer be solved in a single step. Instead, we apply an iterative method. In this work, we use Newton's method, but other methods such as the pseudo-transient solvers (Räss et al., 2022) can be applied as well.

120 In general, iterative solvers perturb the state variables in $\mathbf{u}$ until the equality in equation 2 is satisfied. But how do we know how to perturb $\mathbf{u}$ so that $\mathbf{R}$ goes to zero? Due to the nonlinear nature of the problem we don't know how $\mathbf{R}$ behaves globally, but we can approximate it locally. If we have a guess for $\mathbf{u}$, called $\mathbf{u}^k$, where k indicates the current iteration, we can apply a small perturbation to $\mathbf{u}$, $\delta\mathbf{u}$, such that

$$\|\mathbf{R}(\mathbf{u}^k)\| > \|\mathbf{R}(\mathbf{u}^{k+1})\|, \quad \text{with } \mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u} \quad (4)$$



125 For sufficiently small corrections we can approximate the change in the residual as the first order variation

$$\mathbf{R}(\mathbf{u}^{k+1}) \approx \mathbf{R}(\mathbf{u}^k) + \frac{\partial \mathbf{R}}{\partial \mathbf{u}^k} \delta \mathbf{u} \quad (5)$$

This is simply the multidimensional analogue of approximating a nonlinear curve, in this case $\mathbf{R}$, by its tangent line at the point $\mathbf{u}^k$ and it is the basis for Newton's method. The slope of the tangent line is given by $\frac{\partial \mathbf{R}(\mathbf{u}^k)}{\partial \mathbf{u}^k}$, giving us the direction in which to perturb $\mathbf{u}^k$. Here we define the Jacobian matrix as $\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}}$.

130 The ultimate goal is to find the $\mathbf{u}^k + \delta \mathbf{u}$ such that equation 3 is satisfied. Using the approximation

$$\mathbf{R}(\mathbf{u}^k) + \mathbf{J}^k \delta \mathbf{u} = 0, \quad (6)$$

we can assemble the linearized system that can be solved for the perturbation $\delta \mathbf{u}$ using $\mathbf{J}^k = \frac{\partial \mathbf{R}}{\partial \mathbf{u}^k}$

$$\mathbf{J}^k \delta \mathbf{u} = -\mathbf{R}(\mathbf{u}^k) \quad (7)$$

135 which in turn is used to update $\mathbf{u}$ following equation 4 and $\mathbf{R}$ is reevaluated. This process is repeated until the evaluation of $\mathbf{R}(\mathbf{u} + \delta \mathbf{u})$ is sufficiently close to zero, determined by a threshold. To prevent overshooting the true solution, we apply a line search method, scaling the update by $\alpha \in (0, 1]$ to ensure a sufficient decrease in the residual and maintain robustness for highly non-linear problems. The new update is then

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha \delta \mathbf{u}, \quad \text{with } \alpha = \arg \min \|\mathbf{R}(\mathbf{u} + \alpha \delta \mathbf{u})\| \quad (8)$$

3 Sensitivity of Results to Solution Parameters

140 3.1 Sensitivity Kernels

In many geoscientific applications, we are not only interested in solving the forward model, but also in understanding how outputs respond to changes in model input parameters. This is crucial in inverse problems, model calibration, data assimilation, or uncertainty quantification. To formalize this, we define a cost function $F(\mathbf{u}, \mathbf{m})$, which measures the misfit between model predictions $\mathbf{u}$ and observations $\mathbf{u}_{\text{obs}}$, or encodes an objective we wish to optimize. Strictly speaking, F is a functional, as it maps continuous solution fields to a scalar value. After spatial discretization, however, the fields are represented by finite-dimensional vectors, and F can be treated as a scalar function of $\mathbf{u}$ and $\mathbf{m}$. In the following, we adopt this discrete viewpoint. For example, F might represent the squared difference between observed and simulated surface velocities, temperature fields, or chemical profiles.

$$F(\mathbf{u}, \mathbf{m}) = \frac{1}{2} \sum_i (u_i - u_{\text{obs},i})^2, \quad \text{with } F \in \mathbb{R} \quad (9)$$

150 Here, we are interested in the effect of perturbations in $\mathbf{m}$ on the solution $\mathbf{u}$ itself. A simplified cost function might then be

$$F(\mathbf{u}, \mathbf{m}) = \sum_i u_i, \quad \text{with } F \in \mathbb{R} \quad (10)$$



which is the sum of the solution at selected observation points. In the continuous setting, observing a single point corresponds to a Dirac delta functional, which is not differentiable in the classical sense. Replacing the point observation with a Gaussian kernel yields a smooth approximation that is differentiable for any finite $\sigma > 0$, while recovering a point like behavior in the limit $\sigma \rightarrow 0$ ().

$$F(\mathbf{u}, \mathbf{m}) = \sum_i u_i \exp - \frac{\|x_i - x_0\|^2}{2\sigma^2} \quad (11)$$

The sensitivity of F with respect to the model parameters $\mathbf{m}$ tells us how a small change in $\mathbf{m}$ affects the cost function. We therefore define the sensitivity kernel as

$$\mathbf{K} = \frac{dF}{d\mathbf{m}}, \quad \text{with } \mathbf{K} \in \mathbb{R}^{1 \times M} \quad (12)$$

Note that the sensitivity kernel is a row vector with the number of the material parameters M , which can be large as they vary at every grid point.

Computing this naively with finite differences is extremely expensive for high-dimensional $\mathbf{m}$, as each parameter would require an additional forward solve. The adjoint method allows computation of $\mathbf{K}$ at a cost independent of the number of parameters, typically requiring only one adjoint solve. In the following section we will introduce the derivation of the adjoint system using the discretize-then-optimize method after Reuber et al. (2018) (appendix A).

3.2 Adjoint Method

We are interested in computing how a small change in the material parameters $\mathbf{m}$ affects our cost function F . This means we are looking for the total derivative of F with respect to $\mathbf{m}$, written as:

$$\mathbf{K} = \frac{dF(\mathbf{u}, \mathbf{m})}{d\mathbf{m}} = \frac{\partial F(\mathbf{u}, \mathbf{m})}{\partial \mathbf{m}} + \frac{\partial F(\mathbf{u}, \mathbf{m})}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{m}}. \quad (13)$$

This expression follows from the chain rule and has two contributions. The first term, $\frac{\partial F}{\partial \mathbf{m}}$, accounts for any direct dependency of the cost function on the parameters. The second term captures the indirect effect: changes in $\mathbf{m}$ modify the state $\mathbf{u}$, which in turn alters the value of F .

In many applications, such as data misfit problems, without regularization the cost function depends on $\mathbf{m}$ only through the solution $\mathbf{u}$. In that case, the first term vanishes and we obtain

$$\mathbf{K} = \frac{\partial F}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{m}}. \quad (14)$$

The first derivative, $\frac{\partial F}{\partial \mathbf{u}} \in \mathbb{R}^{1 \times N}$, is usually straightforward to compute. For example, if the cost function is defined as the sum of the solution at selected observation points, this term simply becomes a vector of ones and zeros that follows the gaussian kernel — a mask that selects the parts of $\mathbf{u}$ that contribute to F .

The second derivative, $\frac{d\mathbf{u}}{d\mathbf{m}} \in \mathbb{R}^{N \times M}$, is more challenging. The state $\mathbf{u}$ is not given by an explicit formula, rather it is implicitly defined through the nonlinear system $\mathbf{R}(\mathbf{u}, \mathbf{m}) = 0$. To determine how $\mathbf{u}$ changes with $\mathbf{m}$ we differentiate this



relation with respect to $\mathbf{m}$

$$\frac{d\mathbf{R}(\mathbf{u}, \mathbf{m})}{d\mathbf{m}} = \frac{\partial \mathbf{R}(\mathbf{u}, \mathbf{m})}{\partial \mathbf{m}} + \frac{\partial \mathbf{R}(\mathbf{u}, \mathbf{m})}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{m}}. \quad (15)$$

After solving the forward problem, the residual satisfies $\mathbf{R}(\mathbf{u}, \mathbf{m}) = 0$ for all the (perturbed) parameter combinations. Therefore its total derivative must vanish,

$$0 = \frac{\partial \mathbf{R}}{\partial \mathbf{m}} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{m}}. \quad (16)$$

Now we introduce the notation

$$\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}}, \quad \mathbf{B} = \frac{\partial \mathbf{R}}{\partial \mathbf{m}}, \quad (17)$$

where $\mathbf{J} \in \mathbb{R}^{N \times N}$ is the Jacobian of the forward problem and $\mathbf{B} \in \mathbb{R}^{N \times M}$ describes how the residual directly depends on the material parameters. Rearranging equation 16 yields

$$\mathbf{J} \frac{d\mathbf{u}}{d\mathbf{m}} = -\mathbf{B}. \quad (18)$$

This expression represents solving a batched linear system, where each column of $\frac{d\mathbf{u}}{d\mathbf{m}}$ is obtained by solving a linear system with the same coefficient matrix $\mathbf{J}$ but a different right hand side given by the corresponding column of $-\mathbf{B}$. To circumvent solving this batched system of linear equations, we rearrange equation 18 for $\frac{d\mathbf{u}}{d\mathbf{m}}$ and insert into equation 14 to obtain

$$\mathbf{K} = -\frac{\partial F}{\partial \mathbf{u}} \mathbf{J}^{-1} \mathbf{B}. \quad (19)$$

Equation 19 is a formal expression. While algebraically correct, it does not eliminate the need to solve the batched system of linear equations and, if interpreted literally, suggests explicitly forming $\mathbf{J}^{-1}$, which is neither computationally feasible nor necessary in practice. Here we introduce the adjoint variable $\psi \in \mathbb{R}^{1 \times N}$ which is defined as

$$\psi^T = \frac{\partial F}{\partial \mathbf{u}} \mathbf{J}^{-1}. \quad (20)$$

By taking the transpose of 20 and making use of $(\mathbf{J}^{-1})^T = (\mathbf{J}^T)^{-1}$, we attain a linear system of equations of the form

$$\mathbf{J}^T \psi = \left(\frac{\partial F}{\partial \mathbf{u}} \right)^T, \quad (21)$$

that can be solved for ψ which now involves the transpose of the forward Jacobian. Equation 21 describes the adjoint problem. Once ψ is obtained, the total gradient calculation simplifies to

$$\mathbf{K} = -\psi^T \mathbf{B}. \quad (22)$$

This expression allows computation of the entire gradient with respect to all parameters using only one adjoint solve, regardless of the dimension M . The computational cost therefore scales with the number of state variables N , but is independent of the number of parameters M .



3.3 Summary of the Adjoint Gradient Calculation

The previously described procedure can be summarized in three steps: 1. Solve the (non-)linear forward problem for $\mathbf{u}$ such that

$$\mathbf{R}(\mathbf{u}, \mathbf{m}) = 0. \quad (23)$$

2. Solve the linear adjoint problem for ψ

$$\mathbf{J}^T \psi = \left(\frac{\partial F}{\partial \mathbf{u}} \right)^T. \quad (24)$$

3. Compute the adjoint gradients

$$\mathbf{K} = -\psi^T \mathbf{B}. \quad (25)$$

3.4 Geometric interpretation of adjoint gradients

While the derivation of the adjoint equation is quite established, visualizing the adjoint derivation as geometric objects on the residual surface $R(u, m)$ helps build intuition when encountering this topic for the first time. For a fixed parameter value, m_0 , the forward problem is a one-dimensional root finding problem such that $R(\hat{u}, m_0) = 0$, where $\hat{u}$ is the solution of the forward problem. Panel (a) of Figure 1 shows this as a slice through the residual surface at fixed m_0 . Starting from the initial guess, u^1 , the residual is evaluated and the derivative, J^1 , is computed. The perturbation δu^1 is calculated as the root of the tangent line, the residual is reevaluated and the process is repeated until the residual is sufficiently small. It is important to highlight that although the system has converged when $u = \hat{u}$, the derivative J at the solution is generally not zero. Note that J lies completely in the R, u -plane and carries no information about the m -direction.

Panel (b) shows slices of the residual in the R, m -plane for different values of $\hat{u}$. The filled circles show the points at which a solution exists, i.e. where $R = 0$ for a given u . At different values of u along the Newton trajectory, the shape of R as a function of m changes. The slope of each curve at its filled circle is $B = \partial R / \partial m$, the sensitivity of the residual to the parameter. Critically, B is only meaningful at these solution points, as the implicit function theorem only applies on the constraint surface $R = 0$, so B must be evaluated where the forward problem is satisfied. The filled circles in panel (b) collectively trace the same curve that appears as the white $R = 0$ contour in panel (c).

Panel (c) shows the residual surface together with the tangent plane at the solution point $(\hat{u}, m_0, 0)$. This plane is spanned by the two tangent directions $J = \partial R / \partial u$ (yellow) and $B = \partial R / \partial m$ (red). When calculating sensitivity kernels, we are interested in how the solution itself changes when m is perturbed, as the cost function is $F = u$. The sensitivity kernel is therefore du/dm itself (see equation 18) and is the slope of the tangent of $R = 0$ contour in the u, m -plane. This is the orange dashed line, namely the intersection of the tangent plane with the u, m -plane, whose slope is $d\hat{u}/dm = -J^{-1}B$. Geometrically, the slope of the orange line is determined by the relative steepness of the surface in m -direction compared to the u -direction. A steep B relative to J means the solution shifts rapidly with changes in m , whereas a large J relative to B means the forward problem is stiff and the solution is insensitive to parameter changes.

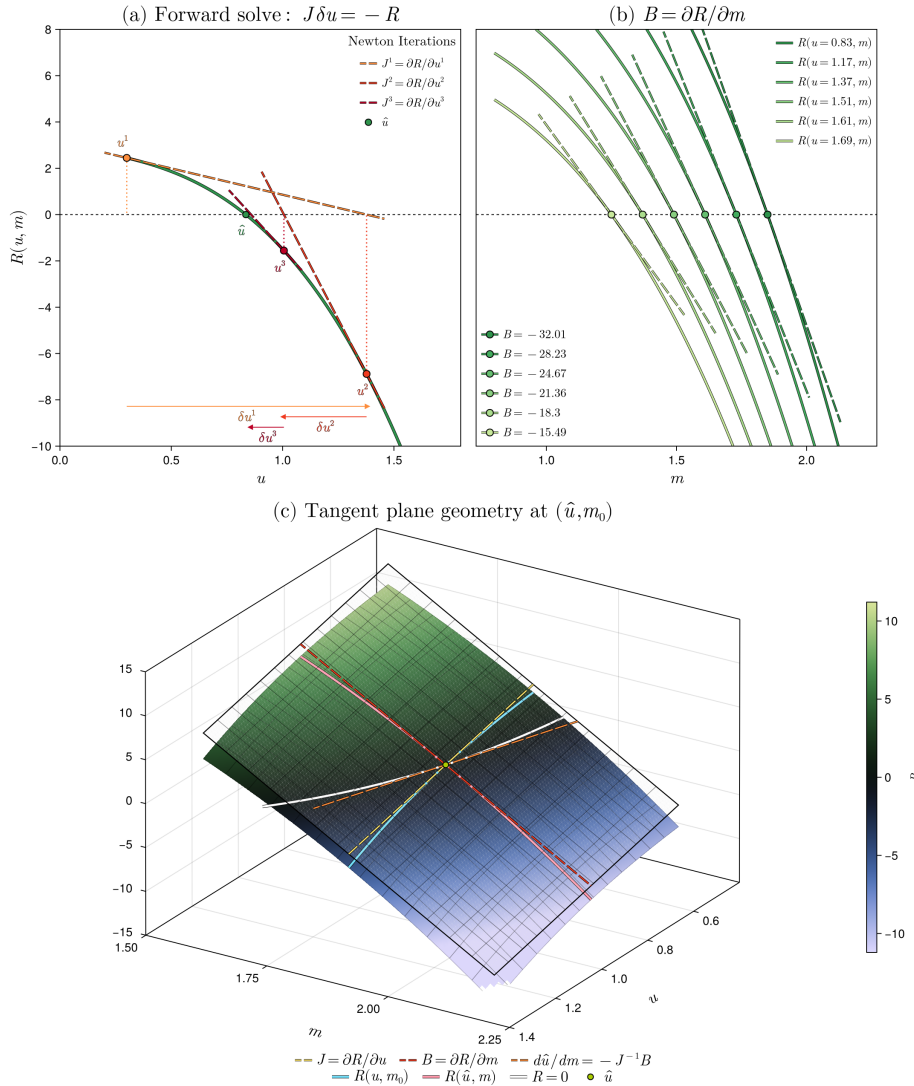


Figure 1. Geometric interpretation of the forward problem and the objects entering the sensitivity calculation. (a) Slice through the residual surface at fixed parameter value m_0 , showing the one-dimensional Newton iteration. From the initial guess u^1 , successive tangent lines with slopes $J^k = \partial R / \partial u^k$ yield perturbations δu^k that drive the residual toward zero. The converged solution $\hat{u}$ satisfies $R(\hat{u}, m_0) = 0$, at which point the derivative J is the Jacobian matrix of our forward problem. (b) Slices through the residual surface in the R, m -plane evaluated at different values of $\hat{u}$, represented as solid points. The derivatives of R with respect to m are shown as dashed tangent lines for these $\hat{u}$'s. The implicit function theorem requires B to be evaluated on the constraint manifold, which is shown as the white $R = 0$ contour in panel (c). The solid points follow this contour (see Figure 2). (c) Residual surface $R(u, m)$ together with the tangent plane at the solution point $(\hat{u}, m_0, 0)$. The plane is spanned by the two tangent directions $J = \partial R / \partial u$ (yellow) and $B = \partial R / \partial m$ (red). The orange dashed line is the intersection of the tangent plane with the u, m -plane, whose slope $d\hat{u} / dm = -J^{-1}B$ is the sensitivity kernel for the case $F = u$.

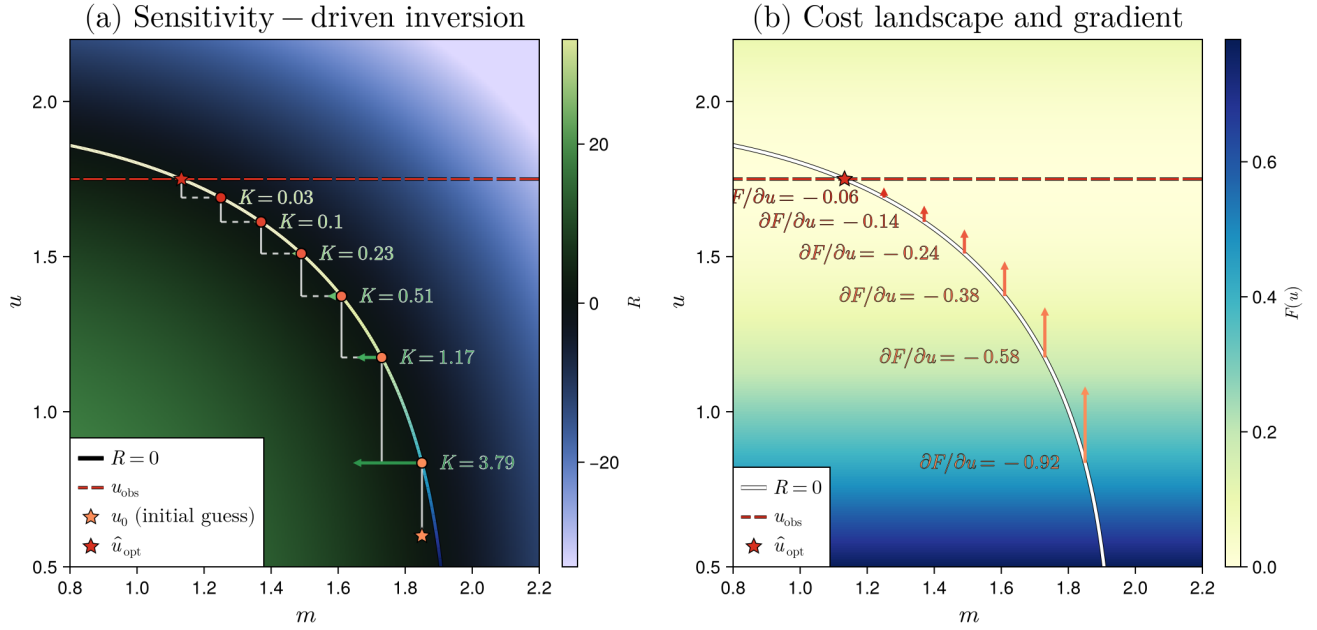


Figure 2. Geometric interpretation of the gradient-based inversion for the scalar cost function $F = \frac{1}{2}(u - u_{\text{obs}})^2$. (a) View of the residual surface from above showing the $R = 0$ contour and the horizontal observation line u_{obs} . The $R = 0$ contour is colored by the corresponding cost function value, highlighting the distance to the objective. The points on the contour where the solutions are evaluated are colored by the derivative of the cost function. (b) View of the cost function surface $F(u, m)$ from above, with the $R = 0$ manifold shown in white. Orange arrows indicate $\partial F/\partial u$ at discrete points along the contour. The inversion proceeds iteratively: starting from the converged solution, $\partial F/\partial u$ is evaluated (panel b), the projected onto the $R = 0$ manifold as the adjoint variable ψ via J^T (panel a). Combining ψ with B gives the sensitivity kernel K , shown as green arrows pointing in the R, m -space that reduces the cost function at the next forward solve. White lines trace the iteration, with dashed segments indicating the parameter perturbation δm and solid segments the subsequent forward solve. As the solution approaches u_{obs} the sensitivity kernels shrink with $\partial F/\partial u$.

The geometric picture also clarifies the role of the adjoint variable. When the cost function is not simply $F = u$ but a general scalar function of the solution, the sensitivity kernel acquires an additional factor that projects $\partial F/\partial u$ onto the solution manifold, $R = 0$. The adjoint variable, ψ , encodes precisely this projection, as it is defined as $J^T \psi = (\frac{\partial F}{\partial u})^T$. It maps the cost function gradient from the unconstrained F, m, u -space onto the constraint surface $R = 0$, accounting for the coupling between u and m imposed by the forward problem. The final sensitivity kernel, $K = -\psi^T B$, then combines these two pieces of information: ψ captures how the cost function weights perturbations in u , while B captures how the residual surface tilts when m is perturbed. In the special case where $F = u$ used in panel (c) and in the following examples, the kernel simplifies to $\frac{du}{dm} = -J^{-1}B$, recovering the slope of the orange dashed line.

Figure 2 illustrates the connection between the sensitivity kernel and gradient based inversion for the material parameter m that satisfies $u(m) = u_{\text{obs}}$. Here we use the cost function $F = \frac{1}{2}(u - u_{\text{obs}})^2$. Panel (a) shows the residual surface from above



with the $R = 0$ contour and the horizontal observation line u_{obs} . The $R = 0$ contour is also colored by the corresponding cost function value, to highlight the distance to the objective (see panel b). Panel (b) shows the cost function surface from above, with the $R = 0$ manifold in white with the derivative of the cost function with respect to u indicated as arrows at discrete points. The inversion begins after the convergence of the forward problem from the initial guess to the $R = 0$ contour. Here the cost function is evaluated with its derivative (see the first arrow in panel (b)). Then the adjoint variable is calculated using J^T , bringing $\frac{\partial F}{\partial u}$ to the $R = 0$ manifold in panel (a). This step is necessary because $\frac{\partial F}{\partial u}$ alone measures how F changes with u when u varies freely, but the forward problem constrains u to remain on the $R = 0$ manifold, and ψ corrects for this constraint. With the information of $\frac{\partial F}{\partial u}$ now in the R, M, u -space in ψ and the information of how the residual changes with respect to m in B all in the same point on the $R = 0$ manifold, the sensitivity of F with respect to m can be determined. The sensitivity kernels are depicted as green arrows, pointing in the direction in the $R - m$ -space which reduces the cost function after the next forward solve. The repetition of this process is indicated as by the white lines, where the dashed line is the parameter perturbation δm and the solid line is the forward solve. As the solution approaches the observation, the sensitivity kernels shrink with $\frac{\partial F}{\partial u}$, despite B remaining fairly large (Figure 2, panel (b)).

The geometric interpretation presented here used scalar functions where u and m are each one-dimensional. In this setting, computing $du/dm = -J^{-1}B$ directly and solving the adjoint system are algebraically identical. The computational distinction emerges in the general case where $u \in \mathbb{R}^N$ and $m \in \mathbb{R}^M$. There, computing du/dm directly requires solving equation 18 once per parameter, yielding M linear solves. This is equivalent to tilting the tangent plane at a given $\hat{u}$ along M parameter directions, and repeating this for every $\hat{u}$ in our forward problem. The adjoint method replaces this with a single linear solve. Projection of $\frac{\partial F}{\partial u}$ onto the $R = 0$ manifold as ψ on the other hand only happens N times, once for every $\hat{u}$. Afterwards, ψ is just a scaling factor for all material parameter derivatives on the $R = 0$ manifold, without any further solves.

4 Technical implementation

In the following section, the numerical implementation will be presented using the 1D linear, steady-state heat diffusion equation as an example. Despite the simplicity of the governing equation, the discrete adjoint derivation mirrors the structure of the general nonlinear case presented above.

We consider steady-state heat diffusion in one dimension. For the case where the conductivity $k(x)$ varies in space, but ρC_p does not, the equation becomes:

$$\frac{\partial}{\partial \mathbf{x}} \left(\mathbf{D}(\mathbf{x}) \frac{\partial \mathbf{T}}{\partial \mathbf{x}} \right) = 0 \quad (26)$$

where $\mathbf{T}$ is the temperature, $\mathbf{D}(\mathbf{x})$ the spatially variable diffusion coefficient and x the spatial domain. The domain is defined by $x \in [0, L]$, where $L = 1.0\text{m}$. The diffusion coefficient is assumed to be positive and bounded. In our numerical experiments, it is initialized using a spatially varying profile following

$$D(x) = \exp \frac{(x - 0.1)^2}{2 \cdot 0.1^3} + \exp \frac{(x - 0.5)^2}{2 \cdot 0.05^2} + \exp \frac{(x - 0.7)^2}{2 \cdot 0.1^2} + 0.05 \quad (27)$$



To close the problem, we impose inhomogeneous Dirichlet boundary conditions on temperature. The temperature is fixed at
280 the left boundary to a value of $\mathbf{T}(0.0) = 1.0$, while the temperature at the right boundary is fixed to $\mathbf{T}(1.0) = 0.0$.

The spatial domain is discretized using a uniform grid with spacing Δx . We use a staggered finite difference discretisation where temperature $\mathbf{T}$ is defined at grid vertices x_i , while the diffusion coefficient D is defined at the cell centers $x_{i+\frac{1}{2}}$. This staggered arrangement ensures that diffusive fluxes are naturally defined at the same locations as the diffusion coefficient.

The diffusive heat flux is given by Fourier's law

$$285 \quad \mathbf{q}_T = -\mathbf{D} \frac{\partial \mathbf{T}}{\partial x}. \quad (28)$$

The discrete temperature gradient at the cell vertices is calculated using a two-point finite difference approximation

$$q_{T,i+\frac{1}{2}} = -D_{i+\frac{1}{2}} \frac{T_{i+1} - T_i}{\Delta x}. \quad (29)$$

Following the general framework introduced in Section 2, we formulate the discrete problem in residual form $\mathbf{R}(\mathbf{T}, \mathbf{D})$. This formulation makes it possible to apply the adjoint method directly to the discretized equations. Substituting the discrete flux
290 into the conservation equation, $\frac{\partial \mathbf{q}}{\partial x} = 0$, and applying a finite difference approximation for the divergence yields the discrete residual equation

$$R_i(\mathbf{T}, \mathbf{D}) = -\frac{1}{\Delta x^2} \left(D_{i+\frac{1}{2}}(T_{i+1} - T_i) - D_{i-\frac{1}{2}}(T_i - T_{i-1}) \right). \quad (30)$$

The forward problem therefore consists of finding the temperature vector $\mathbf{T}$ such that the system is satisfied.

The Jacobian of the forward problem is defined as

$$295 \quad \mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{T}}. \quad (31)$$

Due to the local stencil of the diffusion operator, each residual R_i depends only on the neighboring temperatures T_{i-1}, T_i, T_{i+1} . The resulting Jacobian therefore has a tri-diagonal structure with the entries

$$\frac{\partial R_i}{\partial T_{i-1}} = -\frac{D_{i-\frac{1}{2}}}{\Delta x^2}, \quad \frac{\partial R_i}{\partial T_i} = \frac{D_{i+\frac{1}{2}} + D_{i-\frac{1}{2}}}{\Delta x^2}, \quad \frac{\partial R_i}{\partial T_{i+1}} = -\frac{D_{i+\frac{1}{2}}}{\Delta x^2}. \quad (32)$$

Following the derivation presented in Section 3.2, the adjoint variable ψ is obtained by solving the linear system

$$300 \quad \mathbf{J}^T \psi = \left(\frac{\partial F}{\partial \mathbf{T}} \right)^T. \quad (33)$$

The vector $\frac{\partial F}{\partial \mathbf{T}}$ specifies which parts of the temperature field contribute to the cost function. In this example we consider the sensitivity of the temperature at the center of the domain with respect to the diffusion coefficient field. The derivative of the cost function therefore corresponds to a vector containing zeros everywhere except at the central grid point.

The final ingredient required for the adjoint gradient calculation is the derivative of the residual with respect to the diffusion
305 coefficient

$$\mathbf{B} = \frac{\partial \mathbf{R}}{\partial \mathbf{D}}. \quad (34)$$



with the entries

$$\frac{\partial R_i}{\partial D_{i-\frac{1}{2}}} = \frac{T_i - T_{i-1}}{\Delta x^2}, \quad \frac{\partial R_i}{\partial D_{i+\frac{1}{2}}} = -\frac{T_{i+1} - T_i}{\Delta x^2}. \quad (35)$$

Note that the matrix is bi-diagonal as each residual R_i depends only on the neighboring diffusion coefficients.

310 Rather than assembling the Jacobian matrices analytically, we make use of automatic differentiation to compute the required derivatives directly from the residual implementation. This approach allows the same code structure to be applied to more complex nonlinear problems without requiring manual derivation of the Jacobian entries.

Two short examples of a possible implementation in the Julia programming language are provided. Listing 1 shows how the residual could be defined, how the Jacobian matrix is assembled using automatic differentiation and how the forward problem
315 is solved. For nonlinear problems, the linear solve should be replaced by Newton iterations. Listing 2 demonstrates how to assemble the matrix $\mathbf{B}$, the derivative of the cost function $\frac{\partial F}{\partial \mathbf{T}}$, solve the linear adjoint system and compute the resulting adjoint sensitivity kernels.

A few remarks regarding the implementation are worth noting. In Julia, the residual is typically written in an in-place form such that the residual vector $\mathbf{R}$ is updated for a given temperature field $\mathbf{T}$ and diffusion coefficient field $\mathbf{D}$. Here, we use the
320 package ForwardDiff.jl to evaluate derivatives and assemble the matrices $\mathbf{J}$ and $\mathbf{B}$, although other automatic differentiation tools such as Enzyme.jl or Zygote.jl could be used as well. Similarly, we employ a simple custom Newton solver for clarity, but efficient solver libraries such as NonlinearSolve.jl or NewtonKrylov.jl are readily available. The goal of this example is therefore not to demonstrate the most efficient software configuration, but rather to illustrate the structure of the adjoint workflow in a clear and reproducible manner.

325 The forward solution for the above problem along with sensitivity kernel and adjoint variable is shown on Figure 3. The agreement between the analytical and automatically differentiated solutions demonstrates that the discrete adjoint implementation correctly reproduces both the forward and adjoint operators. In panel (a), the temperature field computed using the AD-assembled Jacobian overlaps the analytical solution throughout the domain. Similarly, panel (d) shows that the adjoint variable obtained from the automatically differentiated system matches the analytical adjoint solution. The resulting sensitivity
330 kernels in panel (c) are likewise in close agreement with both the analytical sensitivities and the finite-difference approximation. Overall, the comparison confirms that the automatic differentiation approach reconstructs the Jacobian operators required for the discrete adjoint formulation and produces physically consistent sensitivity kernels. A more rigorous gradient convergence test for the nonlinear visco-elasto-plastic case, where an analytical reference is not available, is presented in Appendix A.

5 Geoscientific Examples

335 Next, we want to discuss some more realistic geoscientific examples using two dimensional (2D) codes. The basic principle of the approach stays the same, and because we leverage automatic differentiation tools, very little code changes are required when considering more complex examples. We will start with a diffusion example, but build up complexity in subsequent examples. All source code for the examples presented in this section is openly available at Frasniewicz et al. (2026).

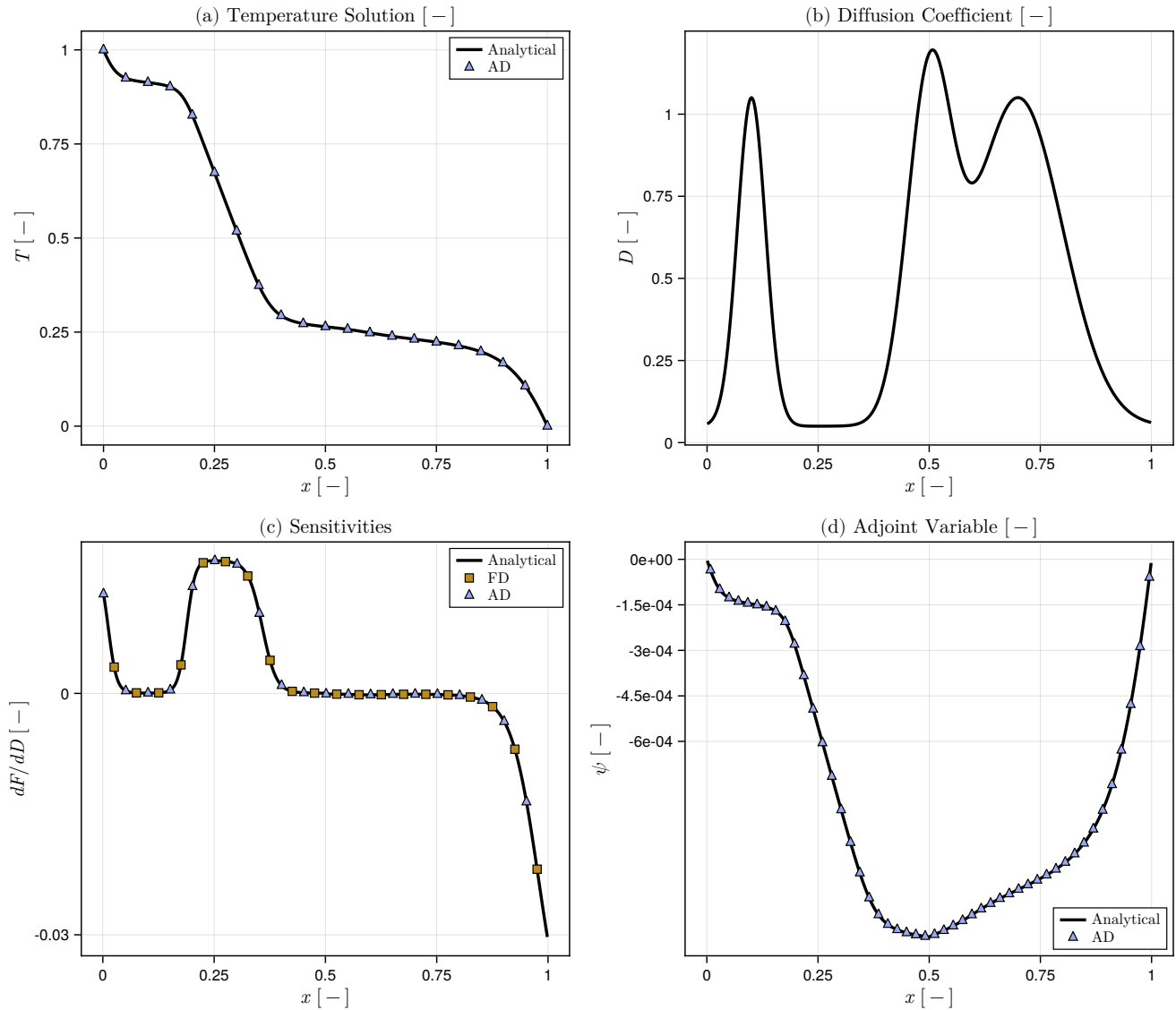


Figure 3. A) Temperature fields calculated using the analytically assembled Jacobian (black lines) and the AD assembled Jacobian (blue triangles). B) Distribution of the spatially varying diffusion coefficient C) Adjoint sensitivities calculated using the analytically assembled Jacobians (black lines) and AD assembled Jacobians (blue triangles). The solution is benchmarked with finite difference sensitivities (brown squares) calculated with a parameter perturbation of $\varepsilon = 10^{-4}$. D) Analytically and AD computed adjoint variables used to calculate the adjoint sensitivities.



Listing 1 Discretized steady-state heat diffusion equation and calculation of the Jacobian using ForwardDiff.jl

```
using ForwardDiff, SparseArrays, LinearAlgebra
# --- Setup ---
N = 100           # number of grid points
L = 1.0           # length of the domain
dx = L / (N - 1)  # spatial step size
D = ones(N)

# Residual of the steady-state diffusion equation  $-\partial(D \partial T) = 0$ 
function Residual!(R, T, D, dx)
    R[1] = T[1] - 1.0 # Dirichlet BC at left boundary
    R[end] = T[end] - 0.0 # Dirichlet BC at right boundary
    for i in 2:length(T)-1
        R[i] = -(D[i]*(T[i+1] - T[i]) - D[i-1]*(T[i] - T[i-1])) / dx^2
    end
end

res!(R, T) = Residual!(R, T, D, dx) # Wrap so ForwardDiff sees only (R, T)
T = ones(N) # Initial guess for T
R = zeros(N) # Initialize residual vector
res!(R, T) # evaluate residual at initial guess
J = sparse(ForwardDiff.jacobian(res!, similar(R), T)) # sparse Jacobian
T_ss = T .- J \ R # Solve (one Newton step; exact for linear problems)
```

Listing 2 Solution of the adjoint system and calculation of the adjoint gradients using ForwardDiff.jl

```
include(joinpath(@__DIR__, "residual_forward.jl"))
# wrap the residual to only take the parameter as argument
Residual_adjoint!(R, D) = Residual!(R, T_ss, D, dx)
# evaluate the residual at the converged solution
Residual_adjoint!(R, D)
# assemble the parameter sensitivity matrix using automatic differentiation
B = sparse(ForwardDiff.jacobian(Residual_adjoint!, R, D))
# define the derivative of the cost function with respect to the state
∇F = zeros(nx); ∇F[round(Int, nx/2)] = 1.0
# solve the linear adjoint problem
ψ = J' \ ∇F
# compute the adjoint sensitivity kernel
K = -ψ' * B
```



5.1 Petrological Capsule Sensitivity Analysis

340 5.1.1 Physical Setup

We consider a high-temperature petrological experiment in which a small sample is enclosed within a platinum capsule and surrounded by an experimental apparatus. The model solves transient heat conduction in a two-dimensional domain, capturing the interaction between the sample, capsule, and surrounding material. The geometry and material zonation are intentionally simplified as the goal is to illustrate the adjoint sensitivity method rather than to faithfully reproduce a specific experimental assembly. This setup is intended to quantify the sensitivity of the sample's temperature field to variations in the surrounding apparatus, which is critical for interpreting laboratory measurements, where it is usually assumed that such effects are negligible. The implementation is provided in `examples/petrological_capsule/2D_diffusion.jl`.

The governing equation is the transient heat conduction equation:

$$\frac{\partial T}{\partial t} = \nabla \cdot (D(\mathbf{x}) \nabla T), \quad (36)$$

350 where T is temperature and $D(\mathbf{x})$ is the thermal diffusivity field. Four distinct material zones are defined: the outer MgO pressure medium, the graphite furnace sleeve, the platinum capsule, and an olivine sample (Figure 4).

The computational domain represents a vertical cross-section of the experimental setup with horizontal and vertical extents of $Lx = Lz = 2$ cm. The domain is discretized using a uniform Cartesian grid with $Nx = Nz = 250$ cells in each direction. The innermost region represents the sample, which is enclosed by a capsule and surrounded by the experimental apparatus. The initial temperature field is set uniformly to $T_0 = 25^\circ\text{C}$, representing room temperature prior to heating, although more complex initial distributions could be used to mimic pre-heated samples. Time integration is performed using an implicit Euler scheme with adaptive time stepping within the bounds $\Delta t_{\min}$ and $\Delta t_{\max}$. Spatial derivatives are approximated with second-order central differences, and the residuals are computed on cell centers.

Boundary conditions are shown in Figure 4. All four domain boundaries are thermally insulating (zero-flux Neumann, $\partial T / \partial n = 0$, dashed red lines), except for the lower portion of the bottom boundary that coincides with the furnace footprint, where a Dirichlet condition of $T_{\text{bc}} = 800^\circ\text{C}$ is imposed (solid blue line).

The sample region is identified using a binary mask array, such that grid cells corresponding to the sample are flagged with 1 and all other cells with 0. This mask is used in both the objective function and the adjoint calculations to focus sensitivities on the sample.

365 5.1.2 Objective Function and Adjoint Sensitivities

The objective function for the petrological capsule sensitivity analysis is defined as the mean temperature within the sample region at each time step. By averaging the temperature over all grid cells corresponding to the sample, the objective function quantifies the integrated thermal state of the material of interest. This choice of objective ensures that the adjoint sensitivities directly reflect how perturbations in the surrounding capsule and apparatus influence the sample's overall heating, providing a

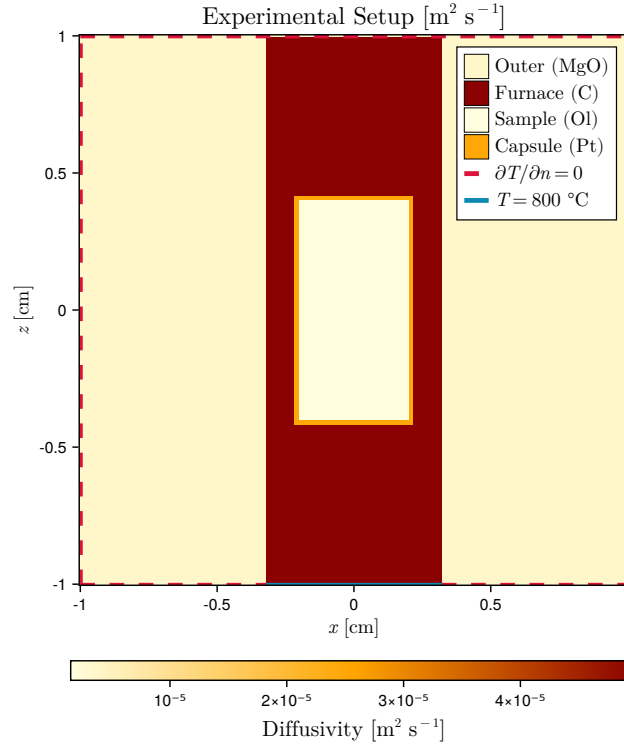


Figure 4. Experimental setup showing the four material zones and boundary condition annotations. Dashed red lines indicate zero-flux (Neumann) boundaries, the solid blue segment marks the Dirichlet contact at 800 °C.

Table 1. Geometric, numerical, and material parameters used in the petrological capsule sensitivity simulation.

Parameter	Symbol (units)	Value	Reference
<i>Thermal diffusivity at ~1000 K</i>			
Outer (MgO)	$D_{\text{outer}} \text{ (m}^2 \text{ s}^{-1}\text{)}$	4×10^{-6}	Hofmeister (1999)
Furnace (graphite)	$D_{\text{furnace}} \text{ (m}^2 \text{ s}^{-1}\text{)}$	5×10^{-5}	Powell and Touloukian (1973)
Sample (olivine)	$D_{\text{sample}} \text{ (m}^2 \text{ s}^{-1}\text{)}$	1.5×10^{-6}	Gibert et al. (2003)
Capsule (Pt)	$D_{\text{capsule}} \text{ (m}^2 \text{ s}^{-1}\text{)}$	2.5×10^{-5}	Hasegawa et al. (2023)

370 physically meaningful metric for interpreting laboratory measurements. Mathematically, the objective can be expressed as

$$F = \frac{1}{N_s} \sum_{(i,j) \in \text{sample}} T_{i,j}, \quad (37)$$



where N_s is the number of sample cells and $T_{i,j}$ denotes the temperature at the (i,j) -th grid cell in the sample. Normalizing by N_s ensures that the objective represents the mean sample temperature, independent of grid resolution, and keeps the adjoint right-hand side comparable in magnitude across all grid sizes.

375 We report the relative, or logarithmic, sensitivity

$$\frac{dF}{d \ln D} = D \frac{dF}{dD}, \quad (38)$$

obtained by multiplying the raw adjoint gradient element-wise by the local diffusivity.

The adjoint-derived gradients of the objective function with respect to the spatially varying thermal diffusivities provide a powerful tool for optimizing the experimental setup. By analyzing how changes in the surrounding apparatus and capsule
380 properties influence the total temperature in the sample, one can iteratively adjust material properties, capsule design, or furnace conditions to maximize heating efficiency and minimize internal temperature gradients. This normalization removes the dependence on the absolute magnitude of D and enables direct comparison between material zones whose diffusivities differ by more than an order of magnitude (olivine 1.5×10^{-6} vs. graphite $5 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$). The resulting quantity has units of $^\circ\text{C}$ and is interpreted as the change in mean sample temperature for a 1% relative change in diffusivity at a given location.

385 Figure 5 shows the results of the forward simulation and the corresponding adjoint sensitivities after 100 time steps. Panel (a) shows the temperature field across the experimental apparatus. The dominant thermal gradient runs vertically from the bottom boundary upward, where the highest temperatures are concentrated.

Panel (b) shows the magnitude of the heat flux, interpolated to the cell centers. The largest fluxes occur at the bottom boundary and at the lower capsule–furnace interface, where the contrast in diffusivity is the largest. Inside the sample itself,
390 the heat flux is comparatively low, indicating that the sample takes longer to equilibrate than the surrounding apparatus.

Panel (c) shows the relative adjoint sensitivity $dF/d(\ln D)$. The sensitivity field shows a pronounced dipolar structure in the furnace around the sample. Directly above and beneath the sample the sensitivities are small, because the diffusivity of the platinum capsule effectively conducts the temperature and wouldn't contribute to changes in mean temperature. The largest sensitivities are in the furnace on the side of the sample. This represents the regions of highest heat flux, which would contribute
395 the largest temperature changes in the sample. The sensitivities in the sample show the temperature redistribution inside of the sample would be affected by diffusivity changes. This suggests that in practice, the material filling the space between the capsule and the furnace wall has more influence on the sample temperature, than the capsule material itself. This finding is directly relevant to experimental design.

5.2 Flow in a Geothermal Reservoir

400 5.2.1 Physical Setup

Next, we consider a simplified two-dimensional representation of a geothermal reservoir typical of Central European sedimentary basins, inspired by the geology of the Trier–Luxembourg Embayment (Schintgen, 2015). The model captures single-phase fluid flow and heat transport within a fully saturated porous medium, driven by pressure-controlled injection and production.

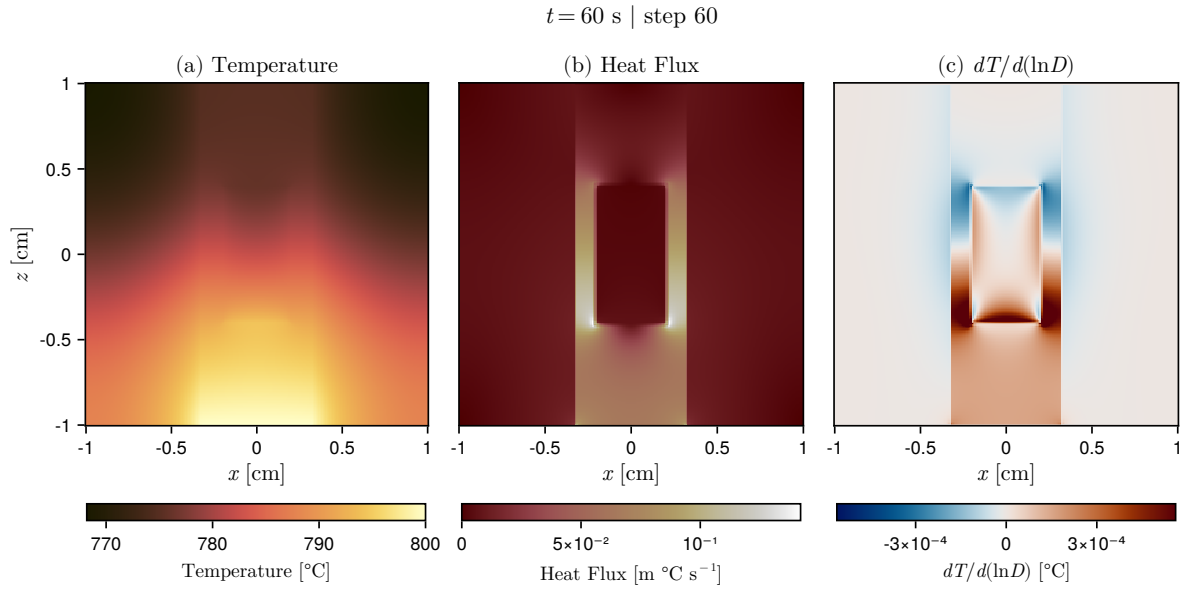


Figure 5. Forward simulation and adjoint sensitivities after 60 time steps ($t \approx 60 \text{ s}$). **(a)** Temperature field ($^\circ\text{C}$). **(b)** Heat-flux magnitude ($\text{m } ^\circ\text{C s}^{-1}$). **(c)** Relative adjoint sensitivity $dF/d(\ln D)$ ($^\circ\text{C}$): the change in mean sample temperature for a 1% increase in local thermal diffusivity.

This formulation captures the dominant processes governing heat extraction in liquid-dominated geothermal systems at reser-
 405 void scale, while remaining simple enough to facilitate interpretation and adjoint-based sensitivity analysis. The implementation
 is provided in `examples/geothermal_reservoir/porous_convection.jl`.

Fluid pressure evolution is governed by the compressible mass balance equation,

$$\varphi \beta_f \frac{\partial P}{\partial t} = -\nabla \cdot \mathbf{q}_D + \frac{Q_P}{\rho} \quad (39)$$

where P is fluid pressure in Pa, φ porosity, β_f fluid compressibility in Pa^{-1} and Q_P a prescribed volumetric source/sink term in
 410 s^{-1} . The Darcy flux $\mathbf{q}_D$ is defined as

$$\mathbf{q}_D = -\frac{k}{\mu} (\nabla P - \rho \mathbf{g}) \quad (40)$$

where k denotes intrinsic permeability in m^2 , μ the dynamic fluid viscosity in $\text{Pa} \cdot \text{s}$, and $\mathbf{g}$ the gravitational acceleration vector
 in $\text{m} \cdot \text{s}^{-2}$. The gravitational term introduces buoyancy effects that are essential for geothermal circulation driven by temperature-
 dependent density contrasts.

415 Temperature evolution is governed by a bulk energy balance

$$\bar{\rho} \bar{C}_p \frac{\partial T}{\partial t} + \nabla \cdot (\rho_f C_{p,f} \mathbf{q}_D T) - \nabla \cdot (\bar{\lambda} \nabla T) = Q_T, \quad (41)$$



where T is temperature, ρ bulk density, C_p bulk heat capacity, α is bulk thermal diffusivity and Q_T is a thermal source term associated with fluid injection. Bulk density and specific heat capacity are computed using porosity-weighted mixing laws

$$\bar{\rho} = \varphi \rho_f + (1 - \varphi) \rho_s \quad (42)$$

420

$$\bar{C}_p = \varphi C_{p,f} + (1 - \varphi) C_{p,s} \quad (43)$$

where subscripts f and s denote fluid and solid matrix properties, respectively. The bulk thermal conductivity is computed using the self-consistent effective-medium formula of Budiansky (1970), which assumes spherical pore geometry (in the closed form given by Gerya (2019))

$$\varphi \frac{\bar{\lambda}}{2\bar{\lambda} + \lambda_f} + (1 - \varphi) \frac{\bar{\lambda}}{2\bar{\lambda} + \lambda_s} = \frac{1}{3}, \quad (44)$$

where the phase conductivities are computed from the stored diffusivity parameters as

$$\lambda_p = \alpha_p \rho_p C_{p,p} \quad \text{for } p \in \{f, s\}. \quad (45)$$

The fluid is assigned standard water properties with density $\rho_f = 1000 \text{ kg m}^{-3}$, heat capacity $C_{p,f} = 4000 \text{ J kg}^{-1} \text{ K}^{-1}$, thermal conductivity $\lambda_f = 0.6 \text{ W m}^{-1} \text{ K}^{-1}$, dynamic viscosity $\mu_f = 10^{-3} \text{ Pa s}$, and compressibility $\beta_f = 10^{-8} \text{ Pa}^{-1}$. The solid matrix is assigned $\rho_s = 2650 \text{ kg m}^{-3}$, $C_{p,s} = 1000 \text{ J kg}^{-1} \text{ K}^{-1}$, and $\lambda_s = 5 \text{ W m}^{-1} \text{ K}^{-1}$, representative of a generic sedimentary rock.

The governing equations are discretized on a two-dimensional Cartesian grid using a finite-difference formulation. Pressure and temperature are stored at cell centers, while fluxes are evaluated on cell faces using face-centered interpolation of material properties. Spatial derivatives of diffusive terms are approximated using second-order central differences. Advective heat transport is handled with a first-order upwind scheme, selecting the upstream temperature value based on the sign of the Darcy velocity at each face:

$$F_{x,i+1/2} = \begin{cases} q_{Dx} T_{i,j} & \text{if } q_{Dx} > 0, \\ q_{Dx} T_{i+1,j} & \text{if } q_{Dx} < 0. \end{cases} \quad (46)$$

Time integration is performed using a fully implicit backward Euler scheme, providing unconditional stability for the strongly coupled advection–diffusion system typical of geothermal reservoir simulations. The initial time step is $\Delta t = 0.01 \text{ yr}$, with adaptive bounds $\Delta t \in [10^{-3}, 1] \text{ yr}$.

No-flow Neumann boundary conditions are applied on all domain boundaries for pressure, emulating a fully confined aquifer. Temperature is fixed at the top and bottom boundaries to reflect a regional geothermal gradient of $40^\circ \text{C km}^{-1}$. Lateral temperature boundaries impose the same depth-dependent gradient as a Neumann condition, suppressing artificial lateral heat fluxes. Initial pressure follows a hydrostatic distribution,

$$P(x, z, t = 0) = \rho_f g z, \quad (47)$$



and initial temperature is set to a linear geothermal profile with $T_{\text{surf}} = 25^\circ\text{C}$ at the surface and an increased bottom temperature of $T_{\text{bot}} = 150^\circ\text{C}$ to represent a thermal anomaly.

An injection well and a production well are represented as Gaussian volumetric source terms in the pressure equation. The injection well is located at $x \approx -0.80 L_x/2$ and $z \approx 0.25 L_z$ above the base of the domain, within the Buntsandstein sandstone
450 aquifer. The production well is located at $x \approx +0.40 L_x/2$ at the same depth, adjacent to the eastern fault. Cold water is injected at $T_{\text{inj}} = 20^\circ\text{C}$, while hot water is extracted at the local reservoir temperature. All model parameters are summarized in Table 2.

The computational domain represents a vertical cross-section of the subsurface with horizontal extent $L_x = 4\text{ km}$ and vertical extent $L_z = 2\text{ km}$, with the bottom of the domain located at $z_{\text{bot}} = -2\text{ km}$ (Figure 6a). The domain is discretized using a
455 uniform Cartesian grid with $N_x = 1000$ cells in the horizontal direction and $N_z = 500$ cells in the vertical direction.

Seven lithological units are stacked from bottom to top, reflecting the characteristic stratigraphy of the region: a crystalline basement, a deep marl/shale aquitard, the Grès de Luxembourg sandstone aquifer, a Muschelkalk limestone unit, a Keuper clay/marl aquitard, an upper Buntsandstein sandstone aquifer, and a volcanic/weathered surface cover. Assigned permeability and porosity values for each unit are listed in Table 2. The interfaces between units are gently folded using a two-harmonic
460 sinusoidal profile,

$$f(x) = A_1 \sin\left(\frac{2\pi x}{W_1}\right) + A_2 \sin\left(\frac{2\pi x}{W_2} + \varphi_2\right), \quad (48)$$

with primary amplitude $A_1 = 0.06 L_z$, primary wavelength $W_1 = 0.70 L_x$, secondary amplitude $A_2 = 0.03 L_z$, secondary wavelength $W_2 = 0.35 L_x$, and phase offset $\varphi_2 = \pi/3$. This mimics the gentle doming and cross-folding observed in the Mesozoic cover sequences of the Luxembourg Embayment.

465 Two steeply dipping normal faults cut through the full sedimentary sequence. Fault 1 is centred at $x_0 = -0.20 L_x$ and dips slightly to the right; Fault 2 is centered at $x_0 = +0.22 L_x$ and dips slightly to the left. Each fault is assigned a damage-zone half-width of 1.2% of L_x and a vertical throw of $0.06 L_z$. The fault damage zones are treated as high-permeability conduits with $k_{\text{fault}} = 10^{-12}\text{ m}^2$, approximately three orders of magnitude above the surrounding aquitards and one order of magnitude above the sandstone aquifers, consistent with hydraulically active fault zones reported in the Upper Rhine Graben system.

470 5.2.2 Sensitivity Kernel Calculation

In this example, we are not only interested in the forward temperature and pressure evolution, but also in quantifying how the outputs of interest respond to perturbations in the uncertain subsurface parameters. Two cost functions are defined, each encoding a physically distinct objective. For both, the adjoint framework described in Section 3 allows the full gradient with respect to all parameters $\mathbf{m} = (\log_{10} k, \log_{10} \mu, \phi, \alpha_f, \alpha_s, \rho_f, \rho_s, C_{p,f}, C_{p,s})$ to be computed at the cost of a single adjoint
475 solve, independent of the number of parameters.

Sensitivities are computed with respect to $\log_{10} k$ rather than k directly. Since permeabilities in the domain span six orders of magnitude (10^{-17} to 10^{-11} m^2), absolute perturbations are not physically meaningful, and working in log-space also improves the conditioning of the sensitivity problem for subsequent gradient-based inversion. The same is done for the fluid viscosity, μ .

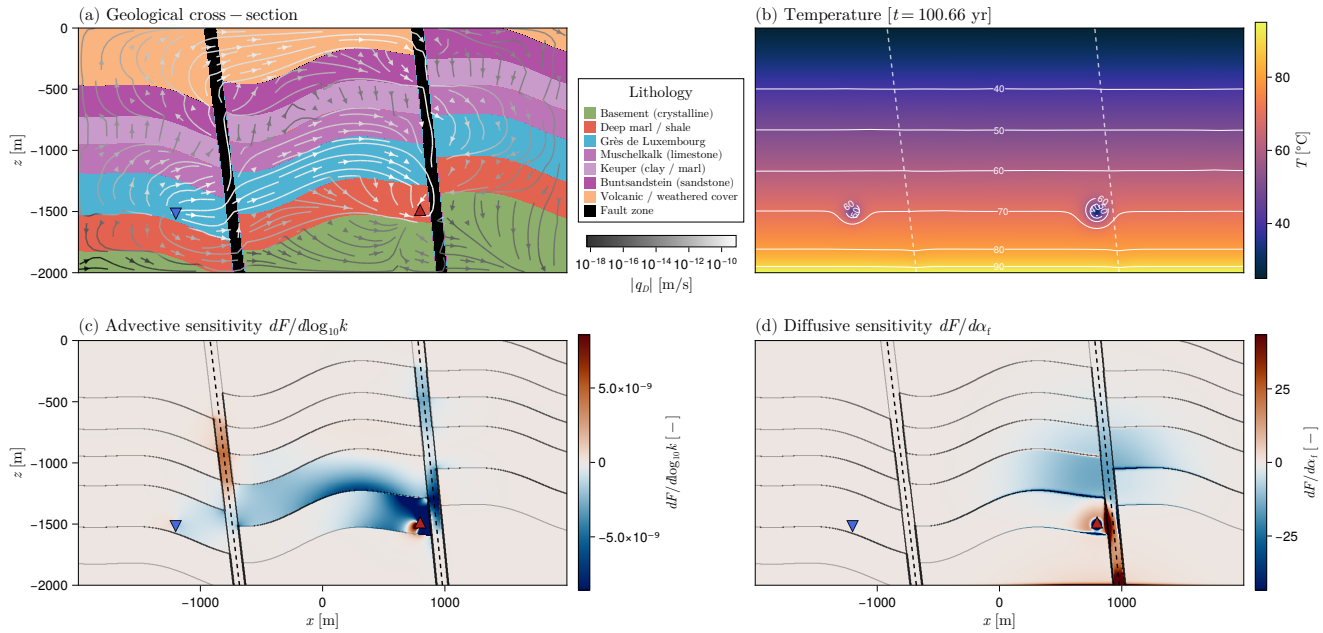


Figure 6. Results of the coupled flow and heat transport simulation and adjoint sensitivity kernels for the geothermal reservoir after $t \approx 100$ years of operation. (a) Geological cross-section showing the seven lithological units, two normal faults. Streamlines indicate regions of fast flow (white) and slow flow (black). Injection and production wells are shown as blue and red triangles respectively. (b) Temperature profile after about 100 years. Solid white lines show the contour of the temperature field, whereas dashed white lines indicate fault locations. (c) Advective sensitivity $dF/d\log_{10}k$, showing how much the extracted thermal power would change if the permeability at each location were increased by one order of magnitude. The largest positive values are concentrated in the Buntsandstein sandstone aquifer between the two wells and within the fault damage zone adjacent to the production well. Negative values in the deeper Grès de Luxembourg aquifer indicate that enhanced permeability there would divert flow away from the production well. (d) Diffusive sensitivity $dF/d\alpha_f$, showing the influence of fluid thermal diffusivity on extracted power. The sensitivity is concentrated near the production well and in the immediately surrounding rock, reflecting the local control of conductive recharge on the thermal state of the reservoir at the extraction point.

**Table 2.** Permeability and porosity assigned to each lithological unit in the geothermal reservoir model, adjusted from Schintgen (2015).

Lithological unit	Permeability k (m ²)	Porosity φ (–)
Crystalline basement	10^{-17}	0.01
Deep marl / shale	5×10^{-15}	0.06
Grès de Luxembourg	5×10^{-13}	0.20
Muschelkalk (limestone)	10^{-13}	0.12
Keuper (clay / marl)	10^{-16}	0.08
Buntsandstein (sandstone)	2×10^{-13}	0.18
Volcanic / weathered cover	10^{-11}	0.25
Fault damage zone	10^{-12}	0.25

The primary objective is the rate of thermal energy extracted at the production well. This is defined as

$$F_{\text{power}} = \sum_{i,j} G_{\text{obs}}(x_i, z_j) \rho_f C_{p,f} |q_D| T_{i,j} \Delta V \cdot \text{scale}_{\text{MW}}, \quad (49)$$

where G_{obs} is a Gaussian observation kernel centered on the production well with radius $\sigma_{\text{obs}} = 20\sigma$, normalized such that its spatial integral equals the total extraction flux $\sum |q_{D,\text{in}}|$. The scalar scale_{MW} converts from non-dimensional to physical units of megawatts. The gradient $dF_{\text{power}}/d\mathbf{m}$ therefore identifies where in the subsurface a local change in any parameter would increase or decrease the extractable power.

The second cost function mimics a temperature observation at a fixed subsurface location, such as a monitoring borehole,

$$F_{\text{temp}} = \sum_{i,j} G_{\text{obs}}(x_i, z_j) T_{i,j}, \quad (50)$$

where G_{obs} is again a Gaussian kernel centered on the production well. This cost function is relevant when monitoring data, rather than extracted power, drives model calibration. Its gradient indicates which regions of the subsurface have the largest influence on the measured temperature.

The advective sensitivity $dF/d(\log_{10} k)$ is a first-order predictor of how much the extracted power would change if the permeability at each location were increased by one order of magnitude. Regions of large positive sensitivity identify formations where additional hydraulic connectivity would yield the greatest increase in thermal output. Because this gradient is computed in a single adjoint solve, it allows rapid screening of candidate well locations without having re-run the forward model for each alternative.

The fault zones in this model act as preferential flow paths connecting the sandstone aquifers across the intervening aquitards. The sensitivity of F with respect to $\log_{10} k$ in the fault damage zones therefore directly quantifies how strongly the fault hydraulic properties control heat extraction. In a gradient-based inversion framework, time-series pressure and temperature obser-



variations at the wells could be used to iteratively update the fault permeability field using this gradient, effectively constraining one of the most uncertain geological parameters in geothermal reservoir characterization.

500 Fault geometry parameters, such as throw, dip, and damage-zone width, could in principle also be included as inversion targets by parameterizing them analytically and differentiating through the geological setup function. Using these gradients, the optimal fault configuration could be determined for the given thermal and hydraulic parameters, that one would aim to find in nature.

The temperature component of the adjoint solution, ψ_T , has a direct physical interpretation beyond its role in computing
505 gradients. The spatial distribution of ψ_T reflects where the current cost function is most sensitive to perturbations in the temperature field. In this sense, it provides a qualitative guide for observation placement, a region where $|\psi_T|$ is large is one where the existing observations already are "looking", and where an additional measurement would most directly augment the sensitivity calculation. However, this interpretation has an important limitation, as adding a new observation changes F , which in turn changes ψ_T itself. The adjoint field is therefore not a static optimality map, but a local, cost-function diagnostic. To properly
510 address the optimal observation placement requires solving an optimal experimental design problem in which the expected reduction in posterior uncertainty is maximized over candidate observation locations. (Haber et al., 2008; Alexanderian et al., 2014), which exceeds the scope of this work. The adjoint field nonetheless provides a computationally free first indicator that requires no additional forward solves.

Figure 6 shows the results of the coupled flow and heat transport simulation and the corresponding adjoint sensitivity kernels
515 after $t \approx 100$ years of operation. Panel (a) shows the geological cross-section with the seven lithological units, two normal faults, and the locations of the injection (blue triangle) and production (red triangle) wells. Flow paths are indicated by the streamlines, where the logarithm of the flow magnitude is indicated by the color. White lines show regions of fast flow, black lines show regions of slow flow. Panel (b) shows the simulated temperature field, with isotherms indicating the thermal plume developing around the injection point. The fault zones act as preferential vertical conduits, locally deflecting isotherms and
520 channeling buoyancy-driven flow between the sandstone aquifers.

Panel (c) shows the advective sensitivity $dF/d(\log_{10} k)$. The largest positive values are concentrated in the Buntsandstein sandstone aquifer between the two wells and within the fault damage zone adjacent to the production well, indicating that increasing permeability in these regions would most directly enhance extracted power. Negative sensitivity in parts of the deeper Grès de Luxembourg aquifer suggests that enhanced permeability there would divert flow away from the production
525 well, reducing output.

Panel (d) shows the diffusive sensitivity $dF/d\alpha_f$. This is concentrated near the production well and in the immediately surrounding rock, reflecting the fact that conductive recharge of the thermal reservoir between injection and production cycles is controlled primarily by the thermal properties of the formation in which the production well is screened. The spatial contrast between the advective and diffusive sensitivity fields illustrates a key result: permeability controls the large-scale flow routing
530 and thus the primary thermal sweep, while thermal diffusivity governs local conductive recharge on shorter timescales and length scales.



5.3 2D Stokes Flow

The final two examples are governed by thermo-visco-elasto-plastic Stokes flow and demonstrate adjoint-based sensitivity analysis for fully coupled, nonlinear thermo-mechanical systems. We first describe the general governing equations and rheological framework, before presenting two examples that activate different subsets of the physics.

Deformation is governed by the quasi-static momentum balance,

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} = 0, \quad \text{with} \quad \boldsymbol{\sigma} = -P\mathbf{I} + \boldsymbol{\tau}, \quad (51)$$

where $\boldsymbol{\sigma}$ is the total stress tensor, P is pressure, $\boldsymbol{\tau}$ is the deviatoric stress tensor, ρ is density, and $\mathbf{g}$ is the gravitational acceleration. Inertial terms are negligible for long term processes and set to zero.

Conservation of mass in the effective medium is given by:

$$\nabla \cdot \mathbf{v} = \dot{\theta}^{\text{el}} + \dot{\theta}^{\text{pl}} + \dot{\theta}^{\text{th}}, \quad \text{with} \quad \dot{\theta}^{\text{el}} = \beta \frac{DP}{Dt}, \quad \dot{\theta}^{\text{pl}} = \dot{\lambda} \frac{\partial Q}{\partial P}, \quad \dot{\theta}^{\text{th}} = \alpha \frac{DT}{Dt}, \quad (52)$$

where $\mathbf{v} = (V_x, V_z)^T$ is the velocity vector in two dimensions, $\dot{\theta}$ denotes volumetric strain-rate contributions, β is the compressibility, D/Dt is the material derivative, $\dot{\lambda}$ is the plastic multiplier, Q is the plastic flow potential, and α is the coefficient of thermal expansion. This formulation corresponds to a weakly compressible approximation, in which volumetric strain arises from elastic compression, thermal expansion, and plastic dilation or compaction. We employ a visco-elasto-plastic rheology, in which the deviatoric stress is related to the deviatoric strain-rate tensor $\dot{\epsilon}'$ through an additive decomposition into viscous, elastic, and plastic components,

$$\dot{\epsilon}' = \dot{\epsilon}^{\text{vis}} + \dot{\epsilon}^{\text{el}} + \dot{\epsilon}^{\text{pl}}, \quad (53)$$

with

$$\dot{\epsilon}^{\text{vis}} = \frac{\boldsymbol{\tau}}{2\eta}, \quad \dot{\epsilon}^{\text{el}} = \frac{1}{2G} \frac{D\boldsymbol{\tau}}{Dt}, \quad \dot{\epsilon}^{\text{pl}} = \dot{\lambda} \frac{\partial Q}{\partial \boldsymbol{\tau}}. \quad (54)$$

Here, η is the shear viscosity and G is the shear modulus. The total strain-rate tensor is computed from the symmetric part of the velocity gradient,

$$\dot{\epsilon} = \frac{1}{2} (\nabla \mathbf{v} + (\nabla \mathbf{v})^T), \quad (55)$$

and its deviatoric part is $\dot{\epsilon}' = \dot{\epsilon} - \frac{1}{3} (\nabla \cdot \mathbf{v}) \mathbf{I}$. The viscous and elastic terms correspond to a Maxwell viscoelastic rheology, which is integrated implicitly in time and may be expressed in terms of an effective viscosity over each time step. Plastic yielding is activated when stresses exceed a Drucker–Prager yield criterion and is regularized through a Perzyna-viscoplastic formulation with an additional tensile cap after Popov et al. (2025).

Density depends on both pressure and temperature



$$\rho = \rho_0[1 - \alpha(T - T_0) + \beta(P - P_0)] \quad (56)$$

560 where ρ_0 , T_0 and P_0 are reference density, temperature, and pressure, respectively. This formulation allows buoyancy forces to arise self-consistently from thermal heterogeneities.

Temperature evolution is governed by the conservation of energy

$$\rho C_v \frac{dT}{dt} = \nabla \cdot (\lambda \nabla T) - \frac{\alpha T}{\beta} \nabla \cdot \mathbf{v} \quad (57)$$

565 where T is the temperature, λ the thermal conductivity, and C_v the specific heat capacity at constant volume. The last term represents adiabatic heating or cooling associated with volumetric deformation, where α/β is the ratio of thermal expansivity to compressibility.

The governing equations are discretized on a uniform staggered Cartesian finite-difference grid. Horizontal and vertical velocities are defined on corresponding cell faces. Pressure, temperature, and all material properties are stored at cell centers. The diagonal components of the stress tensor, τ_{xx} and τ_{zz} , are collocated with pressure at cell centers, while the off-diagonal shear stress component τ_{xz} is defined at cell vertices. Spatial derivatives of diffusive terms are approximated using second-order central differences.

570 Each time step is solved using Newton iterations to a residual tolerance of 10^{-8} . The sparsity pattern of the Jacobian is determined once at the beginning of the simulation using `SparsityTracing.jl` and cached to disk for reuse in subsequent runs. The time step is adaptive: it is halved upon non-convergence and increased by a factor of 1.2 when convergence is achieved in fewer than 10 Newton iterations. All fields and material parameters are non-dimensionalized prior to solving using reference scales for stress, length, time, and temperature, ensuring that the residual contributions from the momentum, mass, and energy equations are of comparable magnitude.

5.3.1 Rising Mantle Plume

580 The model domain represents a vertical cross section through the lithosphere and upper mantle, with a horizontal extent of $W = 1000$ km and a vertical extent of $L = 700$ km, (simplified after Schuler (2025), Figure 7). The vertical coordinate z increases upward, with the bottom of the domain located at $z_{\text{bot}} = -660$ km. Four material phases are used to construct the setup: background mantle (phase 1), crust (phase 2), plume material (phase 3), and sticky air (phase 4). The implementation is provided in `examples/rising_mantle_plume/Rising_Sphere.jl`. Material properties for each phase are listed in Table 3.

585 Phase fractions, temperature, and stress components are advected using a WENO5 scheme (e.g., Dominguez et al., 2024). Material parameters at each cell are updated each time step via a geometric mean interpolation across phase fractions, which avoids the linear-mixing artifact for parameters that span several orders of magnitude such as viscosity.

The initial temperature field is uniform at $T = 1350^\circ\text{C}$ throughout the mantle and transitions to a geothermal gradient of $\partial T/\partial z = 30^\circ\text{C km}^{-1}$ within the crust. The sticky-air layer is held at a constant temperature of $T = 20^\circ\text{C}$. Density is computed from the pressure- and temperature-dependent equation of state (Eq. 56).

590

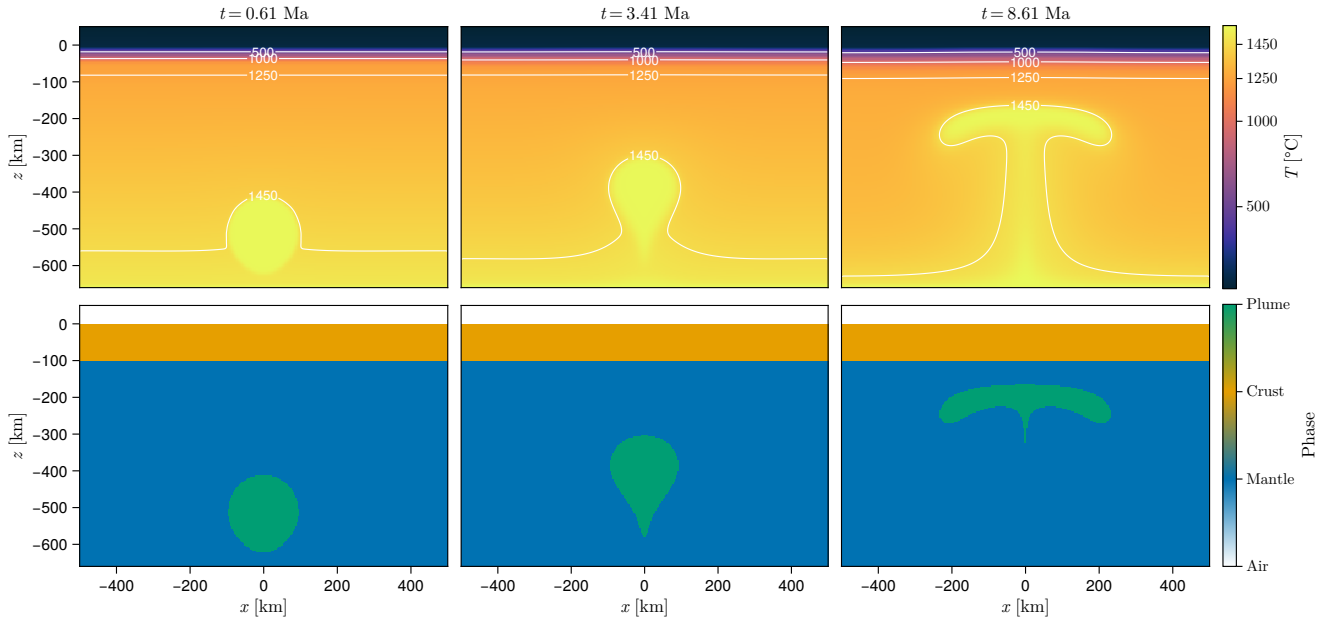


Figure 7. Temporal evolution of rising plume simulation. Top row: snapshots of temperature evolution in °C. Bottom row: snapshots of phase evolution. Phase fields are defined by phase ratios > 0.5 .

To allow the mechanical fields to equilibrate from the initial conditions, the model is first run for an equilibration period of $t_{\text{eq}} = 0.01$ Myr without any thermal perturbation. After the background stress and pressure fields have stabilized, a spherical thermal anomaly representing a mantle plume is introduced. The anomaly has a radius of $R = 100$ km and is located at $x = 0$ km and $z = 0.8 z_{\text{bot}}$. The plume region is assigned the material properties of phase 3 and an initial temperature of $T = 1550$ °C.

From this point onward, the plume rises buoyantly and interacts self-consistently with the surrounding mantle through the coupled thermo-mechanical system.

Free-slip boundary conditions are applied on all sides of the domain. Temperature is fixed at $T_{\text{top}} = 20$ °C at the surface and $T_{\text{bot}} = 1560$ °C at the bottom boundary, while zero-flux (Neumann) conditions are imposed on the lateral boundaries.

To investigate sensitivities in this system, we define two cost functions that capture different aspects of plume dynamics.

The first cost function is the sum of the vertical velocities within the plume phase. This quantity reflects the integrated buoyant ascent of the plume and allows us to assess which material parameters most strongly influence plume dynamics. The second cost function is the sum of vertical velocities evaluated at the top of the crust. This mimics geophysical observations, such as vertical surface velocities measured by GPS stations. In a realistic inversion setting, such observations could be used to constrain uncertainties in material parameters or plume geometry by comparing model predictions with measured data.

**Table 3.** Material parameters for the four phases used in the rising-plume Stokes simulation.

Parameter	Symbol (units)	Mantle	Crust	Plume	Sticky air
Viscosity	η (Pa s)	3×10^{20}	1×10^{23}	1×10^{20}	1×10^{18}
Reference density	ρ_0 (kg m ⁻³)	3000	2700	3000	1
Shear modulus	G (Pa)	5×10^{10}	5×10^{10}	5×10^{10}	5×10^{10}
Heat capacity	C_v (J kg ⁻¹ K ⁻¹)	1050	1050	1050	1050
Thermal conductivity	k (W m ⁻¹ K ⁻¹)	3	3	3	100
Thermal expansivity	α (K ⁻¹)	3×10^{-5}	3×10^{-5}	3×10^{-5}	0
Compressibility	β (Pa ⁻¹)	1×10^{-11}	1×10^{-11}	1×10^{-11}	1×10^{-11}
Reference temperature	T_0 (°C)	1300	1300	1300	20
Reference pressure	P_0 (Pa)	1	1	1	1
Cohesion	C (Pa)	3×10^{10}	3×10^7	3×10^{10}	1×10^{10}
Regularization viscosity	η_{vp} (Pa s)	1×10^{20}	1×10^{18}	1×10^{20}	1×10^{18}
Friction angle	ϕ (°)	10	10	10	10
Dilation angle	ψ (°)	0	0	0	0

605 The fundamental problem in geophysics is that measurements can only be made at or near the surface, but the actual physical process is happening at depth. The two cost functions deliberately sit on the opposite sides of this gap. If the two sensitivity fields look similar in a given region, it means that the surface observation is genuinely informative about that region. Therefore, perturbing a parameter there affects both the plume dynamics and the surface signal in the same way. If this is not the case, using the surface velocities to invert for mantle and plume viscosity would be fitting a different signal, not necessarily related to the underlying plume dynamics.

615 Rather than computing sensitivities with respect to viscosity directly, we evaluate sensitivities with respect to the viscosity exponent n , defined through $\eta = 10^n$. This parameterization has two advantages. First, perturbations in viscosity expressed in absolute units (Pa s) are not meaningful when viscosities span several orders of magnitude (typically 10^{18} – 10^{23} Pa s in our simulations). Second, expressing viscosity in logarithmic form significantly improves the conditioning of the Hessian, which is beneficial for gradient-based inversion methods.

The adjoint sensitivity kernels are computed following the workflow described in Section 4. The Jacobian sparsity pattern is determined once at the start of the simulation using SparsityTracing.jl and reused across all time steps, while the parameter Jacobian $\mathbf{B}$ is assembled via ForwardDiff.jl at each saved snapshot. The sensitivity kernel shown in Eq. (22) is then evaluated independently at each saved snapshot, providing a time series of instantaneous sensitivities.

620 For the surface velocity cost function (top row of figure 8), the sensitivity to $\log_{10}(\eta)$ shows a characteristic dipolar pattern spreading outward from the plume stem. Positive sensitivity directly above the plume indicates that increased viscosity there suppresses lateral flow divergence, effectively channeling momentum vertically and increasing the surface signal despite

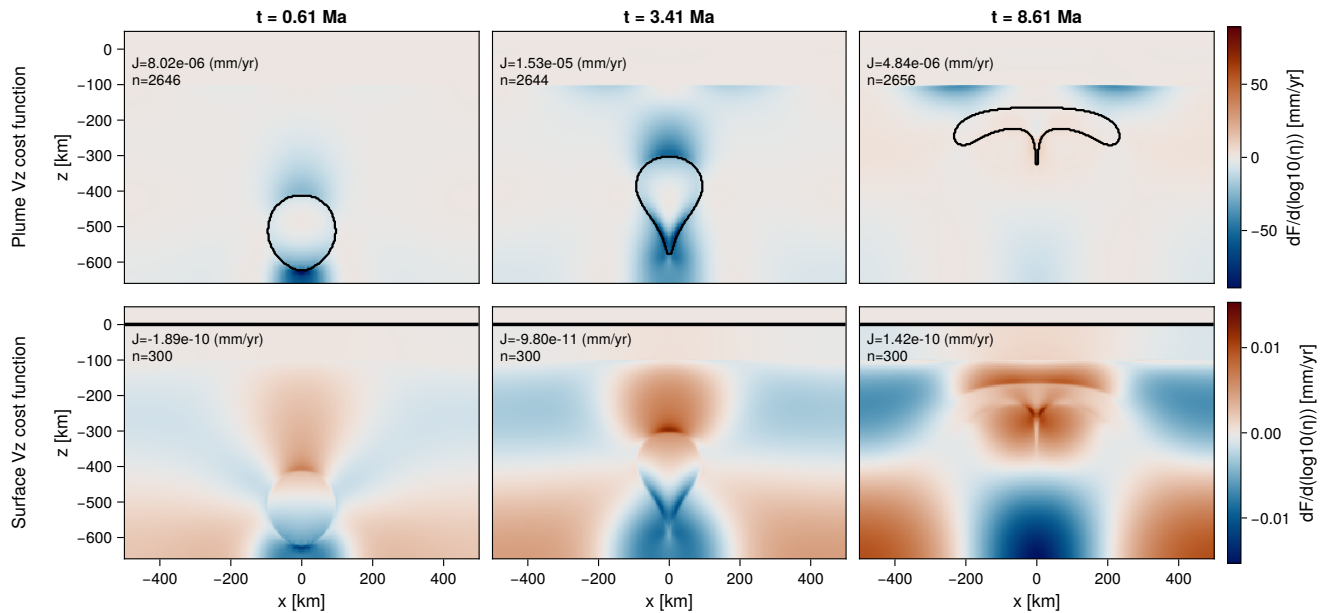


Figure 8. Temporal evolution of sensitivities w.r.t. viscosity exponent. The absolute values of the cost function for each snapshot, J , are presented with the number of observation points, n . Top row: Cost function is the normed sum of vertical surface velocities, marked by the black line. Bottom row: Cost function is the normed sum of vertical velocities where Plume phase > 0.5 , marked by black contour.

reducing overall flow velocities in that region. In contrast, negative sensitivity in the flanking mantle indicates that reduced viscosity there enhances the return circulation around the plume, indirectly increasing plume ascent velocity and therefore the surface signal. Similarly, negative sensitivity in the mantle flanking the plume increases the vertical velocities above the plume, indirectly increasing vertical surface velocities. The largest magnitude sensitivities reside beneath the plume, indicating that a reduction of viscosity would increase the surface velocities the most.

For the plume cost function (bottom row), sensitivity is strongly concentrated within and immediately around the plume body, with viscosity reductions above and below the plume producing the largest increase in plume-internal vertical velocities. As the plume approaches the base of the crust and begins to spread laterally, the sensitivity shifts upward into the mantle directly underlying the crust, where resistance to lateral spreading becomes the dominant control on plume velocity.

The sensitivity fields presented in Figure 8 are not merely diagnostic, but they are the direct ingredients for several geophysically relevant applications.

The most immediate use is rheological inversion from surface observations. In a realistic setting, the cost function would be defined as the misfit between modeled and observed surface velocities, such as GPS uplift rates above a suspected plume. The gradient $\frac{dF_{\text{surf}}}{d\log_{10}(\eta)}$ shown in the top row of figure 8 is then precisely the update direction for the viscosity, which reduces the



misfit of the cost function. It is important to note that in this case the broad spatial extent of the sensitivities is relevant. Surface observations carry information about viscosity far from the plume itself, meaning that a surface velocity dataset constrains not just the plume viscosity, but the ambient mantle as well.

640 This immediately raises the question of how much the surface signal is actually informative about the plume versus the surrounding mantle. Comparing the two sensitivity fields in Figure 8 gives a direct answer. In the regions where the two fields have a similar structure, and to a lesser degree magnitude, a surface observation is sensitive to plume properties and inverting for it would recover meaningful information about plume rheology. Where the surface sensitivity is large but the plume sensitivity is near zero, the surface signal is controlled by the intermediate mantle rather than the plume itself. Updating
645 viscosity there to fit surface observations would improve the data fit while masking the plume. The two gradients together act as a filter, identifying which parts of the surface velocity inversion are physically informative about the source and which are not.

5.3.2 Viscoelastoplastic shear localisation

This example activates the full visco-elasto-plastic rheology described in section 5.3, including Drucker-Prager plastic yielding.
650 The setup consists of a two-dimensional unit domain discretized on a staggered finite-difference grid. A circular inclusion of radius $R = 0.1\text{m}$ is centered at mid-depth and assigned a lower shear modulus, $G = 0.5\text{Pa}$, relative to the surrounding matrix, $G = 1.0\text{Pa}$. Both the inclusion and the matrix share the same reference viscosity, $\eta = 1.0\text{Pa s}$, and compressibility, $\beta = 0.01\text{Pa}^{-1}$. Plastic yielding is applied uniformly to both phases and follows a Drucker-Prager criterion with cohesion $C = 1.6\text{Pa}$, friction angle $\phi = 30^\circ$, dilation angle $\psi = 3^\circ$ and regularization viscosity $\eta_{vp} = 0.001$. A tensile cap is defined
655 with a tensile strength of -2.0Pa to recover pure mode II failure. Strain weakening of the cohesion is not applied. Background pure-shear conditions are applied ($\dot{\epsilon}_{xx} = -\dot{\epsilon}_{zz} = 1.0\text{s}^{-1}$) with free-slip on all sides. Buoyancy forces are neglected and gravity is set to zero. The initial setup and the evolution of the mean pressure within the weak inclusion are shown in Figure 9. The implementation is provided in `examples/vep_shear_bands/shear_bands.jl`. A benchmark of the adjoint gradients against finite differences for this example is presented in Appendix A.

660 The forward solution is shown in Figure 10 at three representative time steps. At $t = 0.7\text{ s}$, prior to yielding, the pressure field increases elastically in the horizontal direction around the inclusion, corresponding to compression, and decreases vertically around the inclusion corresponding to extension. The mean pressure in the inclusion remains zero and accumulated plastic strain (APS) is absent. By $t = 1.21\text{ s}$, plastic yielding has initiated and shear bands begin to propagate in the low pressure zones, which is visible in both the pressure and APS fields. The mean pressure within the inclusion has slightly reduced.
665 At $t = 1.619\text{ s}$ the shear bands are fully developed and localized, extending across the entire domain and reflecting at the boundaries. The pressure field shows the characteristic pressure reduction in the shear bands.

The cost function F is the integrated pressure of the weak inclusion and the sensitivities are calculated with respect to the spatially varying field η , G , and β . The adjoint sensitivity kernels at the same three time steps are shown in Figure 11. Prior to yielding, the sensitivity fields for all three parameters are smooth and dominated by the elastic stress concentration around the
670 inclusion, reflecting the viscoelastic response of the matrix. As the shear bands develop, the sensitivity fields show patterns that

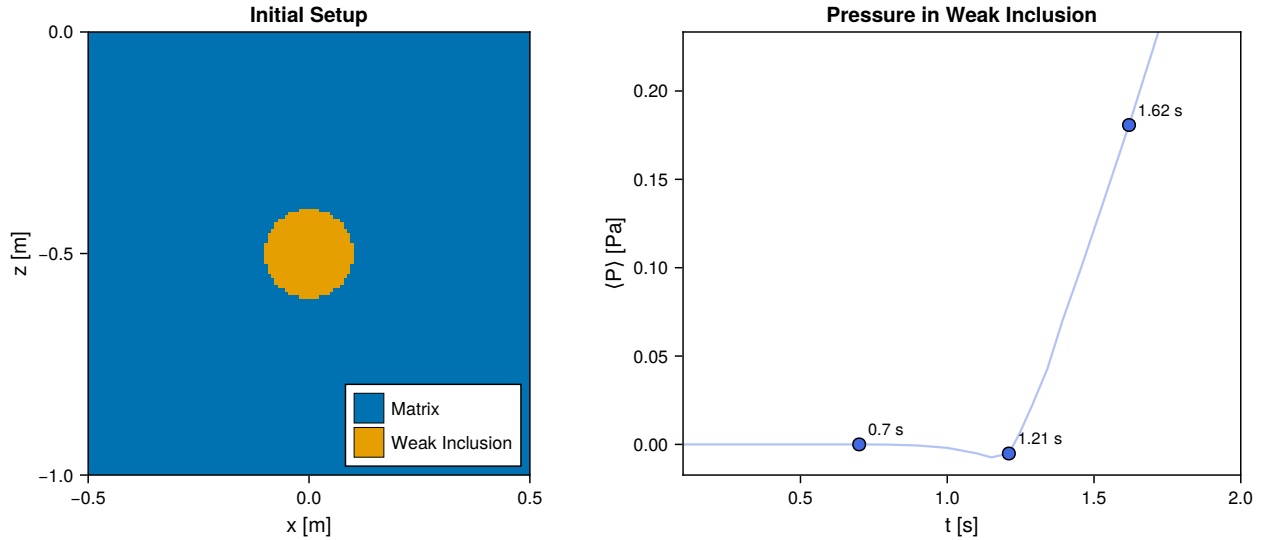


Figure 9. Initial configuration and cost function evolution for the VEP shear band example. left: Initial setup showing the weak circular inclusion (radius $R = 0.1$) embedded in the surrounding matrix. right: Mean pressure within the weak inclusion as a function of time. The three snapshots at $t = 0.7$ s, $t = 1.21$ s, and $t = 1.619$ s correspond to the pre-yielding, early localization, and fully developed shear band stages, respectively (see Figure 10).

follow the shear band trajectory. While the fields of η and G maintain their general structure and polarity, the sensitivity field of β highlights that reductions in compressibility in the shear bands increase the mean pressure of the inclusion. After localization of the shear bands, the sensitivity of β within the weak inclusion dominates, as the pressure of the inclusion increases linearly in time after shear band initiation (see right plot of Figure 9).

675 6 Discussion

The framework presented here applies a single adjoint workflow across diffusion, coupled flow and heat transport, and visco-elasto-plastic Stokes flow. A few aspects of the framework deserve closer attention, both as current limitations and as future directions to extend it to full-scale inverse problems.

We have presented a framework to calculate the adjoint gradients that is suitable in the inversion of parameters in cases
 680 where elliptic PDEs are used as constraints (e.g. steady state heat conduction). Despite the fact that this method is well suited for the elliptic problem, it can only be used on time instances of the transient steps as well. In a time-dependent inverse problem, the adjoint equation must be integrated backwards in time, accumulating gradient contributions from each timestep in reverse order (e.g., MacAyeal, 1992; Errico, 1997). This requires storing the forward state at each timestep, also known as checkpointing, which carries significant memory and computational cost. The per timestep sensitivities shown here should
 685 therefore be understood as diagnostic tools rather than time-dependent, inversion-ready gradients.

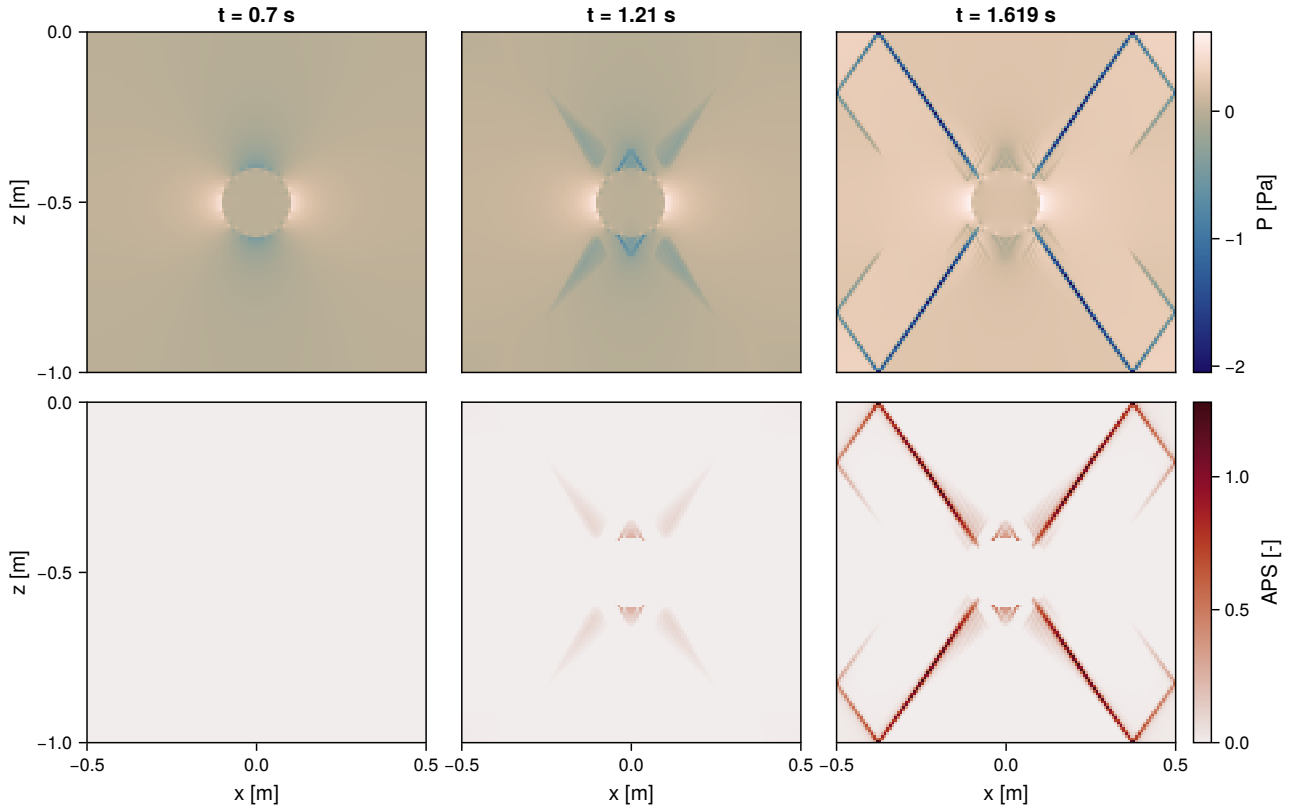


Figure 10. Forward solution of the VEP shear band example at three time steps. top row: Pressure field in Pa showing the stress concentration around the weak inclusion and its evolution as plastic yielding develops. bottom row: Accumulated plastic strain (APS), illustrating the progressive localization of deformation into conjugate shear bands.

In a continuous adjoint derivation, boundary terms arise naturally from integration by parts and impose adjoint boundary conditions that depend on the forward boundary conditions (Plessix, 2006; Gunzburger, 2000). The discrete adjoint, by contrast, automatically inherits the boundary treatment of the forward discretization, a natural consequence of the discretize-then-optimize approach and straightforward to achieve via AD. For sensitivity analysis, this is the desired behavior as the adjoint correctly reflects how perturbations to the parameters propagate through the same discrete system, including at boundaries. Differences with respect to the continuous adjoint boundary conditions may become relevant when inverting for parameters concentrated near boundaries, or when boundary conditions themselves are the target of inversion, but this is outside the scope of the present work.

The choice of the cost function has a direct and substantial influence on the sensitivity kernels. Because the adjoint variable ψ is determined by the right-hand side $\frac{\partial F}{\partial \mathbf{u}}$, different cost functions apply different loading on the adjoint solve and therefore illuminate different aspects of the parameter space. For example, a cost function measuring the integrated pressure will pro-

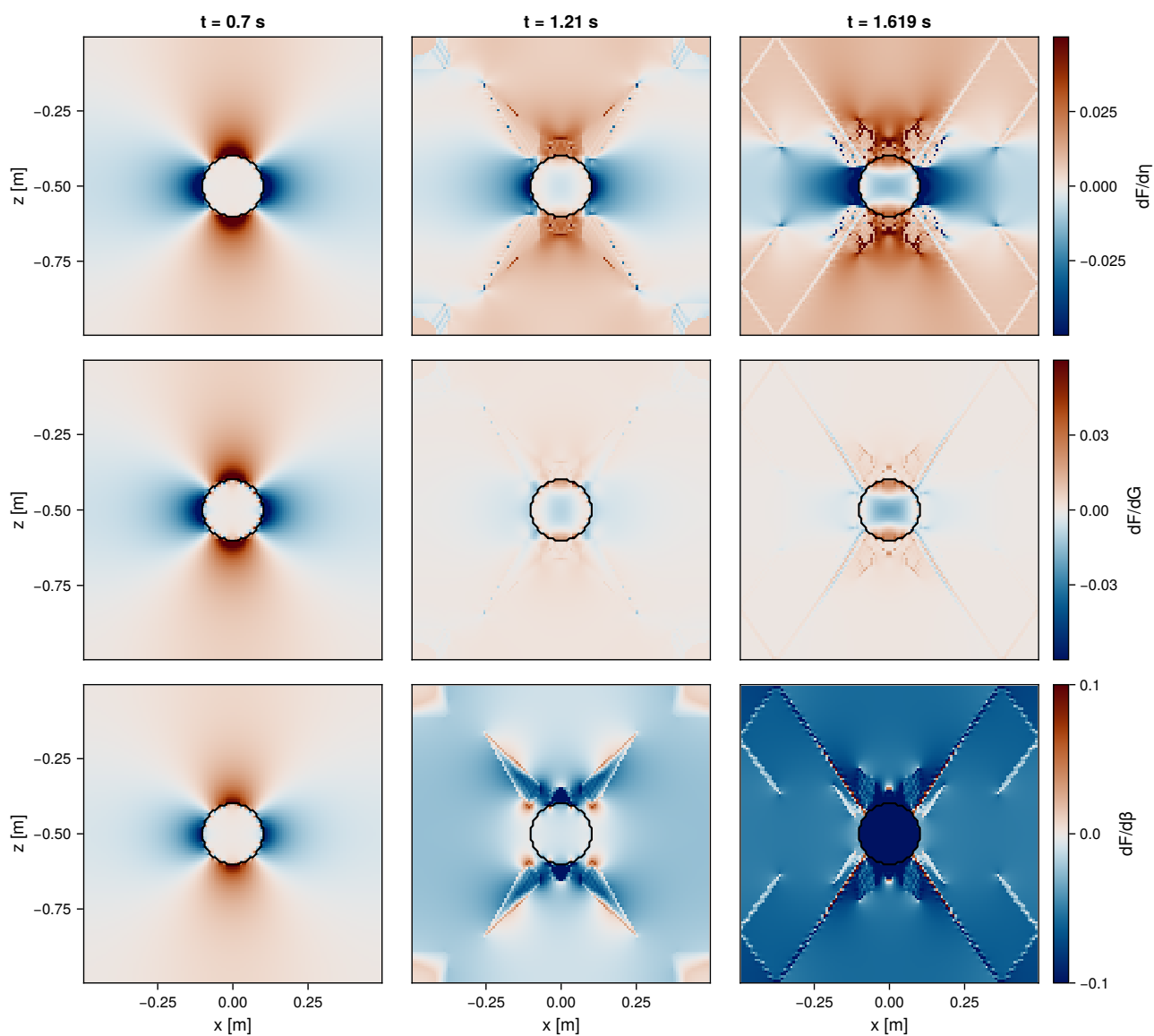


Figure 11. Adjoint sensitivity kernels $\partial F/\partial\eta$ (top row), $\partial F/\partial G$ (middle row), and $\partial F/\partial\beta$ (bottom row) at three time steps, where F is the integrated pressure within the weak inclusion. The inclusion boundary is shown as a black contour.



duce kernels concentrated around pressure generating structures, while measuring surface velocities will weight near-surface rheology. In practice, the cost function should be chosen to reflect what is actually observed and sensitivities computed under one cost function should not be interpreted as general parameter importance measures. It is therefore recommended to compare kernels computed using different cost functions.

Raw sensitivities are only directly comparable when the parameters are similar. When parameters span several orders of magnitude, differentiating with respect to the logarithm of the parameter is natural and we adopt this for permeability and viscosity. One could also use relative sensitivities, where the kernels are normalized by the value of the material parameter itself (Reuber et al., 2018). The same consideration applies on the observation side, where different data types carry different units. Due to the different scales of the observations, gradients with respect to specific observations may dominate an inversion. Therefore relative weighting of observables is a modeling choice that influences inversion outcomes as strongly as the gradient computation.

Finally, the examples here assemble and store the full Jacobian matrices J and B , which is computationally expensive and memory intensive, and rely on direct linear solvers that do not scale to the problem sizes typical of high-resolution 3D geodynamic applications. Scaling to such problems requires two changes: matrix-free evaluation of the Jacobian-Vector product (JVP) for the forward and the Vector-Jacobian product (VJP) for the adjoint (Baydin et al., 2018), and replacing direct solvers with iterative methods such as multigrid or preconditioned Krylov solvers. The latter imposes an additional requirement on solver accuracy, since a loose residual tolerance corrupts the adjoint gradient. In practice both the forward and adjoint systems must be converged to a tolerance well below the desired gradient accuracy.

7 Conclusions

We have presented a framework for computing adjoint sensitivity kernels for nonlinear multiphysics problems in the geosciences using automatic differentiation. The approach was demonstrated on four examples of increasing complexity: a capsule from a petrological experiment governed by transient heat conduction, a coupled fluid flow and heat transport model of a Central European geothermal reservoir, a thermo-visco-elasto-plastic Stokes model of a rising plume, and viscoelastoplastic shear localisation. In each case, the Jacobian J and parameter sensitivity matrix B were assembled via AD rather than derived analytically, allowing the same adjoint workflow to be applied to different problems.

The key insight is that computing the full gradient with respect to all model parameters required only one additional linear solve, regardless of the number of parameters. This property was exploited across all examples despite differences in parameter space dimension and physical complexity. The combination of Newton-based nonlinear solvers, AD-assembled Jacobians, and the discrete adjoint method provides a flexible and reproducible basis for both forward simulation and gradient computation in settings where the governing equations are strongly coupled, parameters are high-dimensional, and observations are sparse.

The adjoint-based sensitivity framework presented here offers a clear path from forward simulation toward systematic model analysis in the geosciences. The ability to automatically compute gradients with regards to the underlying model parameters, without having to derive the problem specific adjoint-equations by hand, is necessary for gaining deeper understanding on what



730 controls physical processes. As geoscientific models continue to grow in physical realism and computational scale, we expect AD-based adjoint methods to become an increasingly essential component of the modeling toolkit.

Code availability. The Julia source code for all examples presented in this manuscript, including the adjoint framework, forward solvers, and visualization scripts, is archived at <https://zenodo.org/records/21157952> Frasunkiewicz et al. (2026) and also available on GitHub at <https://github.com/jfrasunk/Adjoint-Sensitivities-in-the-Geosciences>. All simulation output shown in the figures can be reproduced by running the scripts provided in the repository following the instructions in the README.

Appendix A: Adjoint Gradient Benchmarking

The adjoint derivation for the visco-elasto-plastic Stokes problem is considerably more involved than for the preceding examples. Plastic yielding introduces a non-smooth transition in the constitutive behaviour, and dilational plasticity adds correction terms to the stress and pressure update that depend on parameters absent in the viscoelastic case. As a result, the sparsity pattern of $\mathbf{B}$ changes in cells where plasticity is active. Deriving the analytical adjoint of this system exceeds the scope of this work, so we instead verify the AD-assembled gradients directly against finite differences.

For a cost function $F(m)$ that depends on m through the forward solution, the first order finite-difference approximation along the perturbation direction δm is

$$\frac{F(m + h\delta m) - F(m)}{h} \approx \nabla F \cdot \delta m, \quad (\text{A1})$$

745 where h is a small scalar step. This requires only the forward solver, but scales poorly and is viable only for small problems.

A more stringent diagnostic is the gradient convergence test (Martin and Monnier, 2014; Ghelichkhan et al., 2024)). Rather than perturbing each parameter individually as in equation (A1), $\delta \mathbf{m}$ is a random perturbation vector with unit Euclidean norm,

$$\frac{F(\mathbf{m} + \epsilon \delta \mathbf{m}) - F(\mathbf{m})}{\epsilon} = \frac{\partial F}{\partial \mathbf{m}} \cdot \delta \mathbf{m} + \mathcal{O}(\epsilon). \quad (\text{A2})$$

750 where ϵ is a small positive value.

As $\epsilon \rightarrow 0$ the left-hand side converges to the true directional derivative. If the gradient is correct, the (relative) error

$$\mathcal{E}(\epsilon) = \left| \frac{F(\mathbf{m} + \epsilon \delta \mathbf{m}) - F(\mathbf{m})}{\epsilon} - \nabla_{\mathbf{m}} F \cdot \delta \mathbf{m} \right| \quad (\text{A3})$$

decreases at first order, i.e. with slope 1 on a log–log plot of $\mathcal{E}$ against ϵ .

We benchmark the VEP shear-band setup in three ways: a direct cell-by-cell comparison of the adjoint and finite-difference kernels, a scatter comparison over all cells and time steps, and the gradient test applied to both the adjoint and the finite-difference gradients. Because of the cost of the finite-difference reference, all comparisons are performed at a resolution of 32^2 . We use a relative step size of $h = \sqrt{\epsilon_{\text{machine}}} |m| \approx 1.5 \times 10^{-8} |m|$.

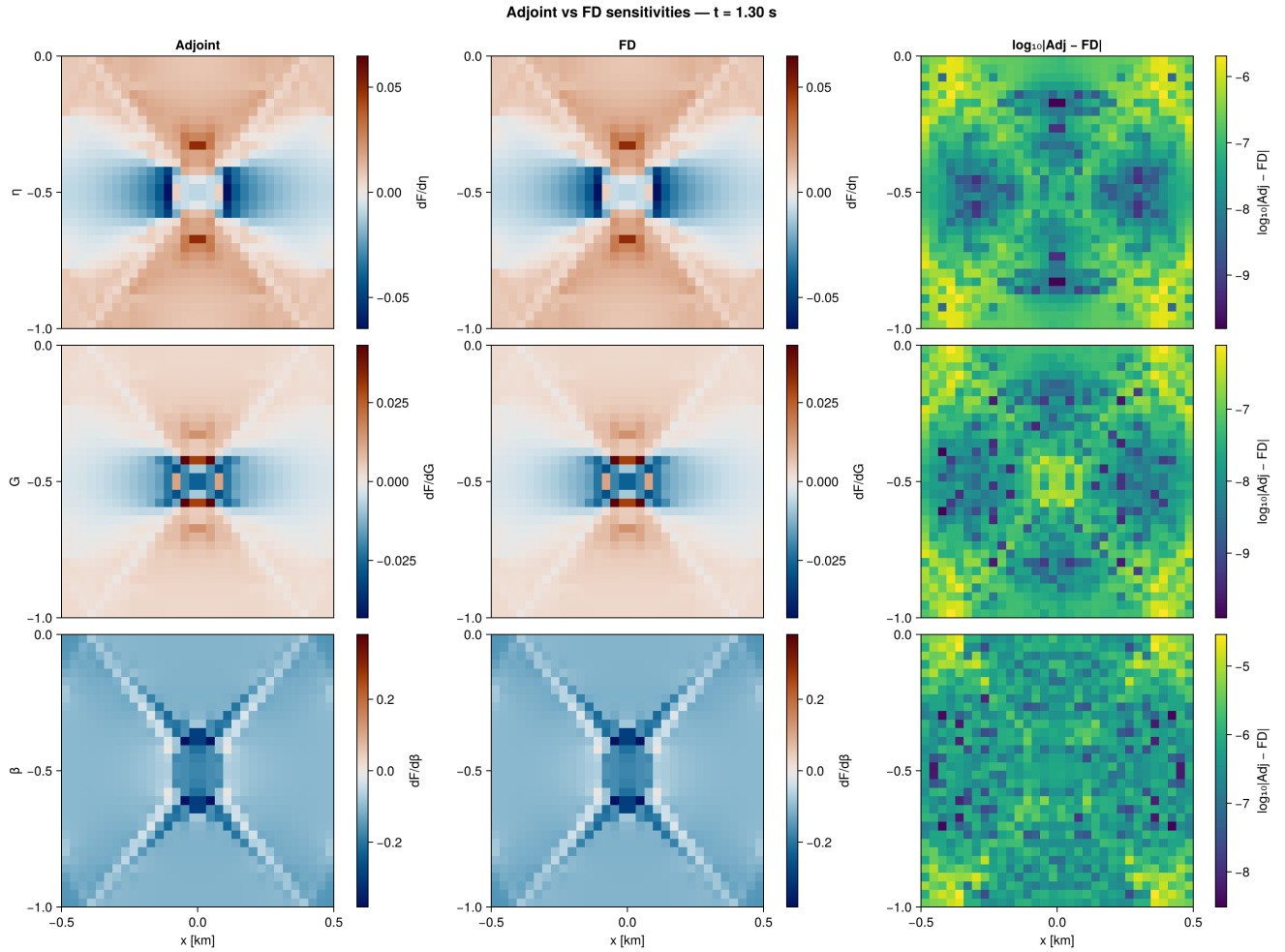


Figure A1. Cell-by-cell comparison of adjoint and finite-difference sensitivity kernels at $t = 1.3$ s and a resolution of 32^2 , after the shear bands have formed. Rows show the sensitivity of the integrated inclusion pressure F with respect to viscosity η (top), shear modulus G (middle), and compressibility β (bottom). The left column shows the adjoint kernels, the center column the finite-difference kernels, and the right column the base-10 logarithm of their absolute difference. The two methods agree to between seven and eight significant digits across most of the domain; the largest differences ($\sim 10^{-5}$) occur for the β kernel, whose magnitude is roughly an order larger than the η and G kernels.



Adjoint vs Finite Differences (all timesteps)

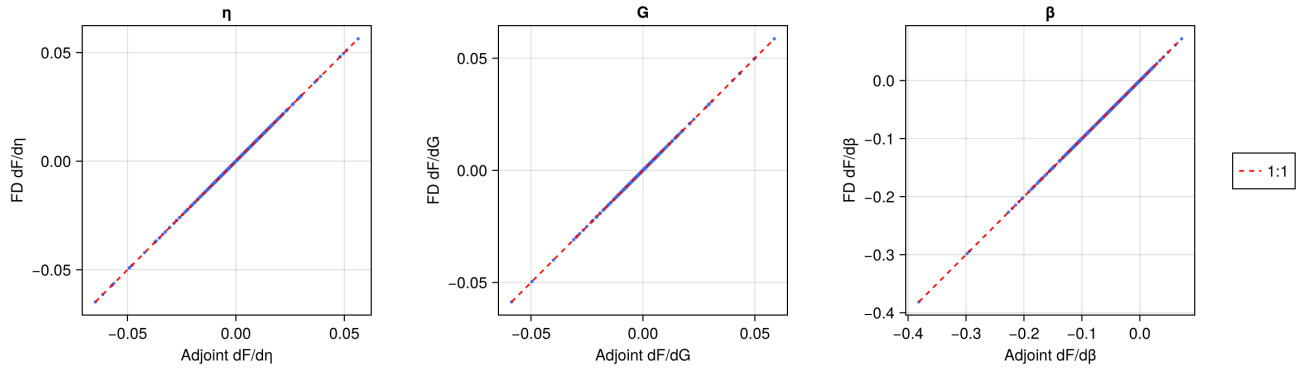


Figure A2. Adjoint versus finite-difference sensitivities for every grid cell at every time step, for viscosity (left), shear modulus (center), and compressibility (right). Each point is one cell at one time step, the dashed red line is the 1:1 relation. All points fall on this line, confirming agreement between the two methods across the full simulation.

Figure A1 compares the two methods cell by cell at $t = 1.3$ s, after the shear bands have localized. The fields are visually indistinguishable, with absolute differences of order 10^{-7} for the η and G kernels and up to 10^{-5} for β , whose magnitude is roughly an order larger. Figure A2 extends this to all cells at all time steps where every point lies on the 1:1 line.

Finally, Figure A3 shows the gradient test for both gradients at $t = 1.3$ s. Both decrease at the expected first order over several decades of ϵ . The adjoint gradient converges to a low round-off floor ($\sim 10^{-7}$), whereas the finite-difference gradient flattens at a higher floor set by its own $\mathcal{O}(h)$ truncation error. Testing both gradients confirms not only that the adjoint kernels are correct, but also that the finite-difference reference used in Figures A1 and A2 is itself trustworthy.

Author contributions. J. Frasunkiewicz developed the adjoint framework, implemented all examples, conducted the validation experiments, and wrote the manuscript. C. Schuler contributed to the development of the rising plume example and the general adjoint framework, the validation thereof and feedback on the manuscript. E. Moulas helped develop the underlying theoretical understanding and provided feedback on the manuscript. B.J.P Kaus developed the proof of concept for the forward modeling, supervised the work and contributed to writing and editing the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This work has been funded by the Federal Ministry of Education and Research (BMBF; project DEGREE, grant no. 03G0925B). The authors thank Jana David for the setup for the petrological experiment.

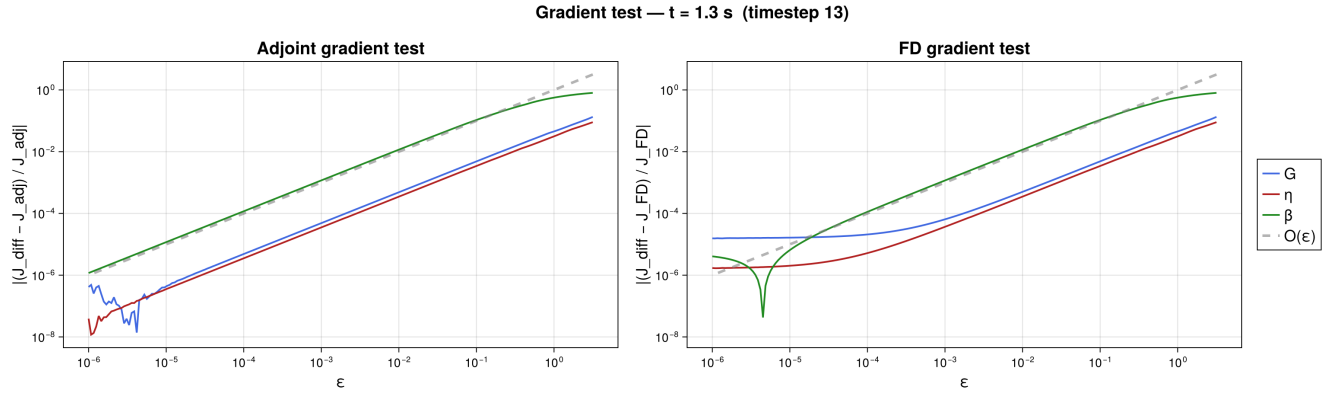


Figure A3. Gradient convergence test at $t = 1.3$ s for the adjoint gradient (left) and the finite-difference gradient (right). Each curve shows the relative error of Eq. (A3) between the directional derivative from a perturbed forward solve and the gradient projected onto the random direction $\delta \mathbf{m}$, for η (red), G (blue), and β (green). The dashed grey line marks the expected first-order $\mathcal{O}(\epsilon)$ slope. For the adjoint gradient the error decreases at first order down to $\epsilon \approx 10^{-5}$, below which round-off in the finite-difference directional derivative dominates. The finite-difference gradient flattens at a higher floor set by its own $\mathcal{O}(h)$ truncation error; the sharp dip in the β curve is a sign change in the error rather than a feature of the convergence.

References



References

- 775 Alexanderian, A.: Optimal Experimental Design for Infinite-dimensional Bayesian Inverse Problems Governed by PDEs: A Review, <https://doi.org/10.48550/arXiv.2005.12998>, 2021.
- Alexanderian, A., Petra, N., Stadler, G., and Ghattas, O.: A-Optimal Design of Experiments for Infinite-Dimensional Bayesian Linear Inverse Problems with Regularized ℓ_0 -Sparsification, <https://doi.org/10.48550/arXiv.1308.4084>, 2014.
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M.: Automatic Differentiation in Machine Learning: A Survey, <https://doi.org/10.48550/arXiv.1502.05767>, 2018.
- 780 Budiansky, B.: Thermal and Thermoelastic Properties of Isotropic Composites, *Journal of Composite Materials*, 4, 286–295, <https://doi.org/10.1177/002199837000400301>, 1970.
- Bunge, H.-P., Hagelberg, C. R., and Travis, B. J.: Mantle Circulation Models with Variational Data Assimilation: Inferring Past Mantle Flow and Structure from Plate Motion Histories and Seismic Tomography, 2003.
- 785 Campolongo, F., Saltelli, A., and Tarantola, S.: Sensitivity Analysis as an Ingredient of Modeling, *Statistical Science*, 15, 377–395, <https://doi.org/10.1214/ss/1009213004>, 2000.
- Colli, L., Bunge, H.-P., and Oeser, J.: Impact of Model Inconsistencies on Reconstructions of Past Mantle Flow Obtained Using the Adjoint Method, *Geophysical Journal International*, 221, 617–639, <https://doi.org/10.1093/gji/ggaa023>, 2020.
- Degen, D. and Wellmann, F.: Guidelines for Sensitivity Analyses in Process Simulations for Solid Earth Geosciences, <https://doi.org/10.31223/X58H83>, 2025.
- 790 Dominguez, H., Riel, N., and Lanari, P.: Modelling Chemical Advection during Magma Ascent, *Geoscientific Model Development*, 17, 6105–6122, <https://doi.org/10.5194/gmd-17-6105-2024>, 2024.
- Errico, R. M.: What Is an Adjoint Model?, *Bulletin of the American Meteorological Society*, 78, 2577–2591, [https://doi.org/10.1175/1520-0477\(1997\)078<2577:WIAAM>2.0.CO;2](https://doi.org/10.1175/1520-0477(1997)078<2577:WIAAM>2.0.CO;2), 1997.
- 795 Fichtner, A., Bunge, H.-P., and Igel, H.: The Adjoint Method in Seismology, *Physics of the Earth and Planetary Interiors*, 157, 86–104, <https://doi.org/10.1016/j.pepi.2006.03.016>, 2006.
- Frasunkiewicz, J., Schuler, C., Moulas, E., and Kaus, B. J.: Automatic adjoints without the algebra: flexible sensitivity analysis of nonlinear problems in the geosciences – model code, <https://doi.org/10.5281/zenodo.21157952>, 2026.
- Gerya, T. V.: *Introduction to Numerical Geodynamic Modelling*, Cambridge University Press, Cambridge, 2 edn., <https://doi.org/10.1017/9781316534243>, 2019.
- 800 Ghattas, O. and Willcox, K.: Learning Physics-Based Models from Data: Perspectives from Inverse Problems and Model Reduction, *Acta Numerica*, 30, 445–554, <https://doi.org/10.1017/S0962492921000064>, 2021.
- Ghelichkhan, S. and Bunge, H.-P.: The Compressible Adjoint Equations in Geodynamics: Derivation and Numerical Assessment, *GEM - International Journal on Geomathematics*, 7, 1–30, <https://doi.org/10.1007/s13137-016-0080-5>, 2016.
- 805 Ghelichkhan, S., Gibson, A., Davies, D. R., Kramer, S. C., and Ham, D. A.: Automatic Adjoint-Based Inversion Schemes for Geodynamics: Reconstructing the Evolution of Earth’s Mantle in Space and Time, *Geoscientific Model Development*, 17, 5057–5086, <https://doi.org/10.5194/gmd-17-5057-2024>, 2024.
- Gibert, B., Seipold, U., Tommasi, A., and Mainprice, D.: Thermal Diffusivity of Upper Mantle Rocks: Influence of Temperature, Pressure, and the Deformation Fabric, *Journal of Geophysical Research: Solid Earth*, 108, 2002JB002108, <https://doi.org/10.1029/2002JB002108>, 2003.
- 810



- Giles, M. B. and Pierce, N. A.: An Introduction to the Adjoint Approach to Design, Flow, Turbulence and Combustion, 65, 393–415, <https://doi.org/10.1023/A:1011430410075>, 2000.
- Givoli, D.: A Tutorial on the Adjoint Method for Inverse Problems, *Computer Methods in Applied Mechanics and Engineering*, 380, 113 810, <https://doi.org/10.1016/j.cma.2021.113810>, 2021.
- 815 Griewank, A. and Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2 edn., ISBN 978-0-89871-659-7, <https://doi.org/10.1137/1.9780898717761>, 2008.
- Gunzburger, M. D.: Adjoint Equation-Based Methods for Control Problems in Incompressible, Viscous Flows, Flow, Turbulence and Combustion, 65, 249–272, <https://doi.org/10.1023/A:1011455900396>, 2000.
- Haber, E., Horesh, L., and Tenorio, L.: Numerical Methods for Experimental Design of Large-Scale Linear Ill-Posed Inverse Problems, 820 *Inverse Problems*, 24, 055 012, <https://doi.org/10.1088/0266-5611/24/5/055012>, 2008.
- Hasegawa, A., Ohta, K., Yagi, T., and Hirose, K.: Thermal Conductivity of Platinum and Periclase under Extreme Conditions of Pressure and Temperature, *High Pressure Research*, 43, 68–80, <https://doi.org/10.1080/08957959.2023.2193892>, 2023.
- Hofmeister, A. M.: Mantle Values of Thermal Conductivity and the Geotherm from Phonon Lifetimes, *Science*, 283, 1699–1706, <https://doi.org/10.1126/science.283.5408.1699>, 1999.
- 825 Horbach, A., Bunge, H.-P., and Oeser, J.: The Adjoint Method in Geodynamics: Derivation from a General Operator Formulation and Application to the Initial Condition Problem in a High Resolution Mantle Circulation Model, *GEM - International Journal on Geomathematics*, 5, 163–194, <https://doi.org/10.1007/s13137-014-0061-5>, 2014.
- Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V. B., and Tebbutt, W.: A Differentiable Programming System to Bridge Machine Learning and Scientific Computing, <https://doi.org/10.48550/arXiv.1907.07587>, 2019.
- 830 Jansen, J.: Adjoint-Based Optimization of Multi-Phase Flow through Porous Media – A Review, *Computers & Fluids*, 46, 40–51, <https://doi.org/10.1016/j.compfluid.2010.09.039>, 2011.
- Li, D., Gurnis, M., and Stadler, G.: Towards Adjoint-Based Inversion of Time-Dependent Mantle Convection with Non-Linear Viscosity, *Geophysical Journal International*, p. ggw493, <https://doi.org/10.1093/gji/ggw493>, 2017.
- Liu, L. and Gurnis, M.: Simultaneous Inversion of Mantle Properties and Initial Conditions Using an Adjoint of Mantle Convection, 835 *Journal of Geophysical Research: Solid Earth*, 113, 2008JB005 594, <https://doi.org/10.1029/2008JB005594>, 2008.
- MacAyeal, D. R.: The Basal Stress Distribution of Ice Stream E, Antarctica, Inferred by Control Methods, *Journal of Geophysical Research: Solid Earth*, 97, 595–603, <https://doi.org/10.1029/91JB02454>, 1992.
- Margossian, C. C.: A Review of Automatic Differentiation and Its Efficient Implementation, *WIREs Data Mining and Knowledge Discovery*, 9, e1305, <https://doi.org/10.1002/WIDM.1305>, 2019.
- 840 Martin, N. and Monnier, J.: Adjoint Accuracy for the Full Stokes Ice Flow Model: Limits to the Transmission of Basal Friction Variability to the Surface, *The Cryosphere*, 8, 721–741, <https://doi.org/10.5194/tc-8-721-2014>, 2014.
- Moses, W. S. and Churavy, V.: Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients, <https://doi.org/10.5555/3495724.3496770>, 2020.
- Moutzouris, D., Stroh, A., Schorn, S., and Moulas, E.: Hamiltonian Monte Carlo Applied to Inverse Petrological Problems, *EarthArXiv preprint*, <https://doi.org/10.31223/X58V0H>, preprint, 2026.
- 845 Plessix, R.-E.: A Review of the Adjoint-State Method for Computing the Gradient of a Functional with Geophysical Applications, *Geophysical Journal International*, 167, 495–503, <https://doi.org/10.1111/j.1365-246X.2006.02978.x>, 2006.



- Popov, A. A., Berlie, N., and Kaus, B. J. P.: A Dilatant Visco-Elasto-Viscoplasticity Model with Globally Continuous Tensile Cap: Stable Two-Field Mixed Formulation, *Geoscientific Model Development*, 18, 7035–7058, <https://doi.org/10.5194/gmd-18-7035-2025>, 2025.
- 850 Powell, R. W. and Touloukian, Y. S.: Thermal Conductivities of the Elements: Results of a Critical Evaluation., *Science*, 181, 999–1008, <https://doi.org/10.1126/science.181.4104.999>, 1973.
- Puel, S., Becker, T. W., Villa, U., Ghattas, O., and Liu, D.: An Adjoint-Based Optimization Method for Jointly Inverting Heterogeneous Material Properties and Fault Slip from Earthquake Surface Deformation Data, *Geophysical Journal International*, 236, 778–797, <https://doi.org/10.1093/gji/ggad442>, 2024.
- 855 Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A.: Universal Differential Equations for Scientific Machine Learning, <https://doi.org/10.48550/arXiv.2001.04385>, 2021.
- Räss, L., Utkin, I., Duretz, T., Omlin, S., and Podladchikov, Y. Y.: Assessing the Robustness and Scalability of the Accelerated Pseudo-Transient Method, *Geoscientific Model Development*, 15, 5757–5786, <https://doi.org/10.5194/gmd-15-5757-2022>, 2022.
- Ratnaswamy, V., Stadler, G., and Gurnis, M.: Adjoint-Based Estimation of Plate Coupling in a Non-Linear Mantle Flow Model: Theory and Examples, *Geophysical Journal International*, 202, 768–786, <https://doi.org/10.1093/gji/ggv166>, 2015.
- 860 Reuber, G. S.: Statistical and Deterministic Inverse Methods in the Geosciences: Introduction, Review, and Application to the Nonlinear Diffusion Equation, *GEM - International Journal on Geomathematics*, 12, 19, <https://doi.org/10.1007/s13137-021-00186-y>, 2021.
- Reuber, G. S., Popov, A. A., and Kaus, B. J.: Deriving Scaling Laws in Geodynamics Using Adjoint Gradients, *Tectonophysics*, 746, 352–363, <https://doi.org/10.1016/j.tecto.2017.07.017>, 2018.
- 865 Reuber, G. S., Holbach, L., Popov, A. A., Hanke, M., and Kaus, B. J. P.: Inferring Rheology and Geometry of Subsurface Structures by Adjoint-Based Inversion of Principal Stress Directions, *Geophysical Journal International*, 223, 851–861, <https://doi.org/10.1093/gji/ggaa344>, 2020.
- Revels, J., Lubin, M., and Papamarkou, T.: Forward-Mode Automatic Differentiation in Julia, <https://doi.org/10.48550/arXiv.1607.07892>, 2016.
- 870 Schintgen, T.: Exploration for Deep Geothermal Reservoirs in Luxembourg and the Surroundings - Perspectives of Geothermal Energy Use, *Geothermal Energy*, 3, 9, <https://doi.org/10.1186/s40517-015-0028-2>, 2015.
- Schuler, C.: Thermomechanical Modeling of Mantle Dynamics and Plate Motion in the Mediterranean, Ph.D. thesis, Johannes Gutenberg-Universität Mainz, Mainz, Germany, <https://doi.org/10.25358/openscience-13957>, 2025.
- Scott, W., Hoggard, M., Duvernay, T., Ghelichkhan, S., Gibson, A., Roberts, D., Kramer, S. C., and Davies, D. R.: Automated Forward and Adjoint Modelling of Viscoelastic Deformation of the Solid Earth, *Geoscientific Model Development*, 19, 2717–2745, <https://doi.org/10.5194/gmd-19-2717-2026>, 2026.
- 875 Siripunvaraporn, W.: Three-Dimensional Magnetotelluric Inversion: An Introductory Guide for Developers and Users, *Surveys in Geophysics*, 33, 5–27, <https://doi.org/10.1007/s10712-011-9122-6>, 2012.
- Smith, J. T. and Vasco, D. W.: Adjoint-Based Inversion of Geodetic Data for Sources of Deformation and Strain, *Journal of Geophysical Research: Solid Earth*, 126, e2021JB021735, <https://doi.org/10.1029/2021JB021735>, 2021.
- 880 Tian, X., Volkov, O., and Voskov, D.: An Advanced Inverse Modeling Framework for Efficient and Flexible Adjoint-Based History Matching of Geothermal Fields, *Geothermics*, 116, 102849, <https://doi.org/10.1016/j.geothermics.2023.102849>, 2024.
- Tromp, J., Tape, C., and Liu, Q.: Seismic Tomography, Adjoint Methods, Time Reversal and Banana-Doughnut Kernels: Seismic Tomography, Adjoint Methods, Time Reversal and Banana-Doughnut Kernels, *Geophysical Journal International*, 160, 195–216, <https://doi.org/10.1111/j.1365-246X.2004.02453.x>, 2004.
- 885



Turcotte, D. L. and Schubert, G.: *Geodynamics*, Cambridge University Press, Cambridge, 3 edn., <https://doi.org/10.1017/CBO9780511843877>, 2014.

Utkin, I., Chen, Y., Räss, L., and Werder, M. A.: Snapshot and Time-Dependent Inversions of Basal Sliding Using Automatic Generation of Adjoint Code on Graphics Processing Units, *Journal of Glaciology*, 71, e72, <https://doi.org/10.1017/jog.2025.40>, 2025.

890 Vasco, D. W. and Mali, G.: On the Use of Adjoints in the Inversion of Observed Quasi-Static Deformation, *Geophysical Journal International*, 224, 896–908, <https://doi.org/10.1093/gji/ggaa481>, 2020.

Villa, U., Petra, N., and Ghattas, O.: hiPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs: Part I: Deterministic Inversion and Linearized Bayesian Inference, *ACM Transactions on Mathematical Software*, 47, 1–34, <https://doi.org/10.1145/3428447>, 2021.

895 Virieux, J. and Operto, S.: An Overview of Full-Waveform Inversion in Exploration Geophysics, *Geophysics*, 74, WCC1–WCC26, <https://doi.org/10.1190/1.3238367>, 2009.