



A new method to invert InSAR data to resolve stress changes on a fracture embedded in a 3D heterogeneous crust

Valérie Cayol, Olivier Bodart, Farshid Dabaghi

► To cite this version:

Valérie Cayol, Olivier Bodart, Farshid Dabaghi. A new method to invert InSAR data to resolve stress changes on a fracture embedded in a 3D heterogeneous crust. 2022. hal-03885687

HAL Id: hal-03885687

<https://uca.hal.science/hal-03885687>

Preprint submitted on 6 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new method to invert InSAR data to resolve stress changes on a fracture embedded in a 3D heterogeneous crust

Oliver Bodart^{*}, Valérie Cayol[†] & Farshid Dabaghi[‡]

1 INTRODUCTION

2 PHYSICAL AND COMPUTATIONAL MODEL

2.1 Mathematical modelling of the solid

From a mathematical point of view, a volcano is a bounded open set $\Omega \subset \mathbb{R}^3$, occupied by an elastic solid. This set is assumed to have a smooth boundary $\partial\Omega$ which we separate in two (non empty) parts $\partial\Omega := \Gamma_D \cup \overline{\Gamma_N}$, with $\Gamma_D \cap \Gamma_N = \emptyset$. As depicted on Figure 1, the subset Γ_N is actually the ground of the volcano and free to move. The subset Γ_D is an artificial boundary defined in order to work on a finite size object and is assumed to satisfy a 0 displacement condition. We assume that the elastic solid occupying Ω is subject to a body force field \mathbf{f} . In the sequel, boldface letters will be used to denote (usually 3-dimensional) vector fields. Plain letters will represent scalar quantities.

We denote by $\mathbf{x} = (x, y, z)$ the generic point in \mathbb{R}^3 , and by $\mathbf{u} = \mathbf{u}(\mathbf{x})$ the displacement field of

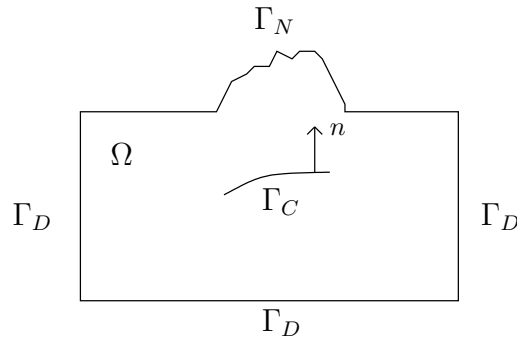


Figure 1. Volcanic domain Ω and its boundaries

the solid Ω . The Cauchy stress $\sigma(\mathbf{u})$ and strain $\varepsilon(\mathbf{u})$ are given by

$$\sigma(\mathbf{u}) = \lambda \text{Tr}(\varepsilon(\mathbf{u})) \mathbf{I}_{\mathbb{R}^3} + 2\mu \varepsilon(\mathbf{u}) \quad \text{and} \quad \varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top),$$

where (λ, μ) are the Lamé coefficients of the material, $\mathbf{I}_{\mathbb{R}^3}$ the identity matrix, and $\text{Tr}(\cdot)$ the matrix trace.

We also assume the presence of a crack in the volcano, which is represented mathematically by a 2-D surface $\Gamma_C \subset \Omega$ on which a traction or pressure force will be exerted (see also Figure 1). In a volcanic context, Γ_C might represent a magma-filled crack on a fault. As already said in the introduction, the shape and position of the crack are supposed to be known in this work. The deformation field \mathbf{u} of the solid is supposed to satisfy the following elastostatic system:

$$-\text{div } \sigma(\mathbf{u}) = \mathbf{f} \quad \text{in } \Omega, \tag{1}$$

$$\mathbf{u} = 0 \quad \text{in } \Gamma_D, \tag{2}$$

$$\sigma(\mathbf{u}) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_N, \tag{3}$$

$$\sigma(\mathbf{u}) \cdot \mathbf{n}^\pm = \mathbf{t}^\pm \quad \text{on } \Gamma_C. \tag{4}$$

The first equation is the equilibrium law describing the (linear) elastic behaviour of the material. Condition (2) describes the fact that the displacement vanishes on the underground boundary of the solid. In equation (3), and in the sequel, \mathbf{n} denotes the unit normal vector on the boundary of Ω , oriented externally. This condition describes the free movement of the ground part of the volcano. Finally, equation (4) describes the force acting on the crack Γ_C . The vectors \mathbf{n}^+ and $\mathbf{n}^- = -\mathbf{n}^+$ are unit normal vectors on the crack Γ_C (see Figure 2). The vector functions $\mathbf{t}^\pm(\mathbf{x})$ denote the force exerted on the crack. They are such that $\mathbf{t}^+ = -\mathbf{t}^-$. When this force acts normally to the crack, that is $\mathbf{t}^\pm(\mathbf{x}) = p(\mathbf{x})\mathbf{n}^\pm$, the force is called a pressure, otherwise in general it is called a traction force.

This traction force is the actual unknown of our problem. From the mathematical point of view it is supposed to be quadratically integrable as well as its gradient.

In order to derive a finite element approximation of the state equations (1)–(4), we need a weak formulation of the system. This is done by multiplying the equation (1) by a regular vector field \mathbf{v} such that $\mathbf{v} = 0$, and then integrating by parts over Ω . Using the conditions (2)–(4) then yields:

$$\int_{\Omega} \sigma(\mathbf{u}) : \varepsilon(\mathbf{v}) \, d\Omega = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_C} (\mathbf{t}^\pm) \cdot \mathbf{v} \, d\Gamma_C, \tag{5}$$

where $:$ is the double inner product for matrices, i.e. for $A = (a_{ij})_{i,j=1,2,3}$ and $B = (b_{ij})_{i,j=1,2,3}$,

$$A : B = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} b_{ij}.$$

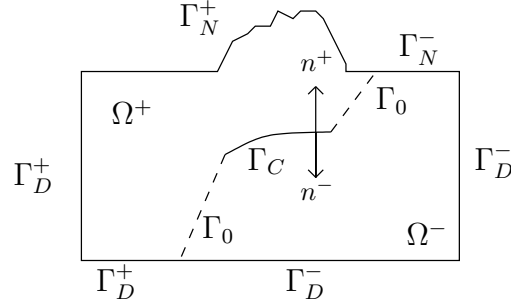


Figure 2. Splitting a volcanic cracked domain by domain decomposition method

Notice that this approach allow us to deal with situations where the elastic material is nonhomogeneous and anisotropic, that is when the Lamé coefficients are not constant.

2.2 Finite element model

The direct problem (1)–(4) efficiently will have to be solved iteratively during the inversion procedure. Therefore we need an efficient algorithm to do so. To this aim, we use a domain decomposition method. More precisely, following Bodart et al. (2016), the domain Ω is split into two sub-domains such that each point of the domain lies on one side of the crack or on the crack. For this purpose, we use an artificial extension Γ_0 of the crack Γ_C . Assuming that $\Gamma_F = \Gamma_C \cup \Gamma_0$ splits the domain into two subdomains Ω^+ and Ω^- , we have $\Omega = \Omega^+ \cup \Gamma_F \cup \Omega^-$, $\Gamma_N = \Gamma_N^+ \cup \Gamma_N^-$ and $\Gamma_D = \Gamma_D^+ \cup \Gamma_D^-$. We define on Γ_C two opposite unit outward normal vectors \mathbf{n}^+ (from Ω^+) and \mathbf{n}^- (from Ω^-). The global unknown \mathbf{u} is split into $\mathbf{u}^+ = \mathbf{u}|_{\Omega^+}$ and $\mathbf{u}^- = \mathbf{u}|_{\Omega^-}$.

Therefore, the equations (1)–(4) can be rewritten with \mathbf{u}^+ and \mathbf{u}^- as unknowns:

$$-\text{div } \sigma(\mathbf{u}^\pm) = \mathbf{f}^\pm \quad \text{in } \Omega^\pm, \quad (6)$$

$$\mathbf{u}^\pm = 0 \quad \text{on } \Gamma_D \cap \partial\Omega^\pm, \quad (7)$$

$$(\sigma(\mathbf{u}) \cdot \mathbf{n})^\pm = 0 \quad \text{on } \Gamma_N \cap \partial\Omega^\pm, \quad (8)$$

$$(\sigma(\mathbf{u}) \cdot \mathbf{n})^\pm = \mathbf{t}^\pm \quad \text{on } \Gamma_C, \quad (9)$$

$$[[\mathbf{u}]] = 0 \quad \text{on } \Gamma_0, \quad (10)$$

$$[[\sigma(\mathbf{u})]] \cdot \mathbf{n}^+ = 0 \quad \text{on } \Gamma_0, \quad (11)$$

where $[[\mathbf{v}]] = \mathbf{v}^+ - \mathbf{v}^-$ denotes the jump of a function across Γ_0 . The equations and conditions (6)–(9) are a straightforward rewriting of (1)–(4). The two last conditions (10) and (11) are imposed to enforce the continuity of displacement and stress across Γ_0 to ensure that $\mathbf{u} = (\mathbf{u}^-, \mathbf{u}^+)$ solves the original problem (1)–(4).

Let us now describe the discretization of this problem via the finite element method. A lagrange

multiplier λ defined on Γ_0 is introduced to enforce the continuity of the displacement across the Γ_0 . Defining the associated lagrangian functional and expressing the associated saddle point conditions give a weak formulation for Problem (6)–(11) which unknown is in the form $\mathbf{X} = (\mathbf{u}^+, \mathbf{u}^-, \boldsymbol{\lambda})^\top$ (see Bodart et al. (2016) for details).

In order to numerically approximate the system, consider a tetrahedral mesh of Ω (as e.g. in Figure 3). Let φ_i^\pm be a finite element basis defined on this mesh. Also defining a mesh of the extended crack, we build finite elements bases ψ_i and θ_j on Γ_0 and Γ_C respectively.

Identifying the unknowns \mathbf{u}^+ , \mathbf{u}^- and $\boldsymbol{\lambda}$ with there value on the nodes of the mesh, the discretized form of Problem (6)–(11), then reads

$$\mathbf{K}\mathbf{X} = \mathbf{F}. \quad (12)$$

The matrix \mathbf{K} of this system is built from the stiffness matrices A^\pm , and the constraint coupling matrices on Γ_0 denoted by B^\pm . More precisely, we have

$$\mathbf{K} = \begin{pmatrix} A^+ & 0 & B^{+T} \\ 0 & A^- & -B^{-T} \\ B^+ & -B^- & 0 \end{pmatrix},$$

where

$$A^\pm := \left[\int_{\Omega^\pm} \sigma(\varphi_i^\pm) : \varepsilon(\varphi_j^\pm) d\Omega^\pm \right]_{ij}, \quad B^\pm := \left[\int_{\Gamma_0} \varphi_i^\pm \cdot \psi_j d\Gamma_0 \right]_{ij}$$

The right hand side vector \mathbf{F}^\pm is given by

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}^+ \\ \mathbf{F}^- \\ 0 \end{pmatrix}, \quad \mathbf{F}^\pm := \left[\int_{\Omega^\pm} \mathbf{f}^\pm \cdot \varphi_i^\pm d\Omega^\pm + \int_{\Gamma_C} \mathbf{t}^\pm \cdot \varphi_i^\pm d\Gamma_C \right]_i,$$

which boils down to the algebraic formulation

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}^+ \\ \mathbf{F}^- \\ 0 \end{pmatrix} = \begin{pmatrix} M_\Omega^+ \mathbf{f}^+ \\ M_\Omega^- \mathbf{f}^- \\ 0 \end{pmatrix} + \begin{pmatrix} +M_C \mathbf{t} \\ -M_C \mathbf{t} \\ 0 \end{pmatrix},$$

where the mass matrices M_Ω^\pm and the coupling matrix on Γ_C , denoted by M_C , are given by

$$[M_\Omega^\pm]_{ij} = \int_{\Omega^\pm} \varphi_i^\pm \cdot \varphi_j^\pm d\Omega^\pm, \quad [M_C]_{ij} = \int_{\Gamma_C} \varphi_i^\pm \cdot \theta_j d\Gamma_C.$$

Eventually, we can naturally define two matrices L_Ω and L_C such that we have

$$\mathbf{F} = L_\Omega \mathbf{f} + L_C \mathbf{t}.$$

The matrix \mathbf{K} is symmetric and positive definite, which allows the use of powerful classical

solvers. The implementation of this model uses the finite element library GetFem++ Renard & Pomier (<http://home.gna.org/getfem>). The library provides all the necessary tools to handle various types of PDEs, and links to classical powerful solvers (conjugate gradient, SuperLU, MUMPS) as well as other routines for mesh management, definition of cracks and boundaries via level set functions) which make it quite versatile and powerful. We refer the reader to Bodart et al. (2016) for a detailed mathematical and computational analysis of this method. See also Bodart et al. (2020, 2021) for a mathematical analysis of similar optimal control problems aking use of the domains decomposition technique.

3 INVERSION FROM FULL GROUND MEASUREMENTS

3.1 Mathematical formulation of the problem

Let us assume the displacement field of the volcano to be fully measured on the ground Γ_N , and call it \mathbf{u}_d . The inversion problem consists in finding a traction vector \mathbf{t} such that the solution of Problem (1)–(4) satisfies $\mathbf{u} = \mathbf{u}_d$ on Γ_N . However this is known to be an ill-posed problem (e.g. the solution is not unique), and moreover the measurements are perturbed by random uncertainties. Therefore we adopt a least squares approach, combined with a regularization (i.e. smoothing) technique to build the inversion process.

Consider the following cost function:

$$J(\mathbf{t}) = \frac{1}{2} \int_{\Gamma_N} (\mathbf{u} - \mathbf{u}_d)^\top \mathbf{C}^{-1} (\mathbf{u} - \mathbf{u}_d) d\Gamma_N + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 d\Gamma_C, \quad (13)$$

where \mathbf{C} denotes the covariance operator of the measurements uncertainties (see e.g. Tarantola (2005)), and is assumed to be positive definite, \top denotes the transpose operator, and finally, the non-negative constants α_0 and α_1 are the Tikhonov regularization (or smoothing) parameters.

We aim at minimizing this cost function over the space of feasible traction vectors. This space consists in quadratically integrable vectors, which gradient is also quadratically integrable.

In Bodart et al. (2020, 2021), a simplified version of this problem is mathematically studied. The cost function only features the first smoothing term. As it will be shown later on, our tests have proved the necessity of smoothing the gradient, in a practical framework, to obtain a physically admissible solution \mathbf{t} . The optimal choices for the values of α_0 and α_1 will be discussed further.

To derive the optimality conditions for the Problem (13), we introduce the following Lagrangian:

$$\mathcal{L}(\mathbf{u}, \mathbf{t}, \phi) = J(\mathbf{t}) - \int_{\Omega} (\sigma(\mathbf{u}) : \varepsilon(\phi) - f) \cdot \phi d\Omega + \int_{\Gamma_C} \mathbf{t}^\pm \cdot \phi d\Gamma_C,$$

where ϕ is the Lagrange multiplier for the constraint (1)–(4) in the weak form (5). When $\nabla_{\mathbf{u}} \mathcal{L} = 0$ and $\nabla_{\phi} \mathcal{L} = 0$, we have $\nabla_{\mathbf{t}} \mathcal{L} = \nabla J$. Cancelling $\nabla_{\phi} \mathcal{L}$ gives problem (1)–(4). Cancelling $\nabla_{\mathbf{u}} \mathcal{L}$ and

performing integration by parts yields that ϕ is the solution of

$$\begin{cases} -\operatorname{div} \sigma(\phi) = 0 & \text{in } \Omega, \\ \phi = 0 & \text{in } \Gamma_D, \\ \sigma(\phi) \cdot \mathbf{n} = \mathbf{C}^{-1}(\mathbf{u} - \mathbf{u}_d) & \text{on } \Gamma_N, \end{cases} \quad (14)$$

called the *adjoint state* system. Finally, computing $\nabla_{\mathbf{u}} \mathcal{L}$, for \mathbf{u} and ϕ satisfying (1)–(4) and (14), gives:

$$\nabla J(\mathbf{t}) = \alpha_0 \mathbf{t} + \alpha_1 \nabla \mathbf{t} + (\phi \cdot \mathbf{n}^\pm). \quad (15)$$

We refer the reader to Bodart et al. (2020, 2021), for further details.

After splitting the computational domain Ω , the cost function J in the optimization problem (13) becomes:

$$\begin{aligned} J(\mathbf{t}) &= \frac{1}{2} \int_{\Gamma_N^+} (\mathbf{u}^+ - \mathbf{u}_d^+)^\top \mathbf{C}^{-1} (\mathbf{u}^+ - \mathbf{u}_d^+) d\Gamma_N^+ + \frac{1}{2} \int_{\Gamma_N^-} (\mathbf{u}^- - \mathbf{u}_d^-)^\top \mathbf{C}^{-1} (\mathbf{u}^- - \mathbf{u}_d^-) d\Gamma_N^- \\ &\quad + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 d\Gamma_C. \end{aligned} \quad (16)$$

The previous optimality conditions rewrite naturally from this new formulation.

3.2 The Discrete Problem

The next step is to adapt the cost function, the adjoint state and the gradient of the cost function to the discrete system (12). To this aim, it has to be noticed that all terms in (16) are symmetric. Discretizing the two first terms will involve the introduction of mass matrices on Γ_N^\pm . One needs to keep the symmetry of the discrete form. Since the covariance matrix \mathbf{C} is positive definite, it is also the same for its inverse and we can write

$$\mathbf{C} = QDQ^{-1}, \quad \mathbf{C}^{-1} = QD^{-1}Q^{-1},$$

where Q is the matrix which columns are the eigenvectors of \mathbf{C} and D the diagonal matrix of the eigenvalues of \mathbf{C} . This diagonalized form allows us to write $\mathbf{C}^{-1} = \mathbf{C}^{-\frac{1}{2}} \mathbf{C}^{-\frac{1}{2}}$, where

$$\mathbf{C}^{-\frac{1}{2}} = QD^{-\frac{1}{2}}Q^{-1}.$$

In view of this, we rewrite the continuous cost function J as

$$\begin{aligned} J(\mathbf{t}) &= \frac{1}{2} \int_{\Gamma_N^+} (\mathbf{u}^+ - \mathbf{u}_d^+)^\top \mathbf{C}^{-\frac{1}{2}} \mathbf{C}^{-\frac{1}{2}} (\mathbf{u}^+ - \mathbf{u}_d^+) d\Gamma_N^+ + \frac{1}{2} \int_{\Gamma_N^-} (\mathbf{u}^- - \mathbf{u}_d^-)^\top \mathbf{C}^{-\frac{1}{2}} \mathbf{C}^{-\frac{1}{2}} (\mathbf{u}^- - \mathbf{u}_d^-) d\Gamma_N^- \\ &\quad + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 d\Gamma_C. \end{aligned} \quad (17)$$

As in Section 2.2, we will identify the previous functions with their values at the mesh nodes. Define a set of finite element basis functions χ_i^\pm (with 3 degrees of freedom at each mesh node) defined on Γ_N^\pm . Then, define the ground mass matrix

$$M_G = \begin{pmatrix} M_G^+ & 0 \\ 0 & M_G^- \end{pmatrix}, \quad (18)$$

with $[M_G^\pm]_{ij} = \int_{\Gamma_N} \chi_i^\pm \cdot \chi_j^\pm d\Gamma_N$. Let us also define the matrices M_{F_0} and M_{F_1} as

$$[M_{F_0}]_{ij} = \int_{\Gamma_C} \theta_i \cdot \theta_j d\Gamma_C, \quad [M_{F_1}]_{ij} = \int_{\Gamma_C} \nabla \theta_i \cdot \nabla \theta_j d\Gamma_C.$$

Recall that θ_j are the finite element basis functions defined on Γ_C .

Finally, in view of the domain decomposition method, the mesh nodes located on the ground will be on either side of the crack. Therefore, the measured displacement and the restriction to the ground nodes of the computed displacement (denoted \mathbf{u}_G) can be written as

$$\mathbf{u}_d = \begin{pmatrix} \mathbf{u}_d^+ \\ \mathbf{u}_d^- \end{pmatrix}, \quad \mathbf{u}_G = \begin{pmatrix} \mathbf{u}_G^+ \\ \mathbf{u}_G^- \end{pmatrix}.$$

Consider then the reduction matrix \mathcal{O}_R such that

$$\mathcal{O}_R \mathbf{X} = \mathbf{u}_G,$$

where \mathbf{X} is the solution of (12).

The discrete version of the cost function J defined by (17) is then

$$J_d(\mathbf{t}) = \frac{1}{2}(\mathcal{O}_R \mathbf{X} - \mathbf{u}_d)^\top \mathbf{C}^{-\frac{1}{2}} M_G \mathbf{C}^{-\frac{1}{2}} (\mathcal{O}_R \mathbf{X} - \mathbf{u}_d) + \frac{\alpha_0}{2}(\mathbf{t}^\top M_{F_0} \mathbf{t}) + \frac{\alpha_1}{2}(\mathbf{t}^\top M_{F_1} \mathbf{t}). \quad (19)$$

The optimality conditions for the minimizer of the discrete cost function J_d are obtained via the characterization of the saddle point of the following (discrete) Lagrangian function:

$$\mathcal{L}_d(\mathbf{X}, \mathbf{t}, \phi) = J_d(\mathbf{t}) - \langle \mathbf{K} \mathbf{X} - (L_\Omega \mathbf{f} + L_C \mathbf{t}), \phi \rangle.$$

Computing the partial derivative of \mathcal{L}_d and then cancelling them, lead to the discrete counterpart of the adjoint problem (14) as:

$$\mathbf{K} \phi = \mathcal{O}_R^\top \mathbf{C}^{-\frac{1}{2}} M_G \mathbf{C}^{-\frac{1}{2}} (\mathcal{O}_R \mathbf{X} - \mathbf{u}_d),$$

which rewrites, denoting $\mathbf{u}_G = \mathcal{O}_R \mathbf{X}$,

$$\mathbf{K} \phi = \mathcal{O}_R^\top \mathbf{C}^{-\frac{1}{2}} M_G \mathbf{C}^{-\frac{1}{2}} (\mathbf{u}_G - \mathbf{u}_d), \quad (20)$$

where \mathbf{K} is the (symmetric) matrix of System (12). The gradient of J_d at any point \mathbf{t} is then:

$$\nabla J_d(\mathbf{t}) = \alpha_0 M_{F_0} \mathbf{t} + \alpha_1 M_{F_1} \mathbf{t} + L_C^\top \phi. \quad (21)$$

3.3 Practical minimization algorithm

In a previous work (see Bodart et al. (2021)), we presented two methods to minimize a simpler version of our cost function J_a : the conjugate gradient algorithm and a quasi-Newton method (low storage BFGS). These two techniques are natural choices due to the quadratic structure of the cost function. They were studied in depth in terms of computational performance and accuracy. The BFGS method appeared to be converging faster when the mesh of the domain Ω is rather coarse. When the mesh becomes finer, the number of unknowns in the problem increases and the two methods tend to give similar results. In the framework of applications which we are interested in, we aim at fast processes, that is we will most of the time consider meshes that are rather coarse except in the neighborhood of the crack Γ_C . Therefore we will use a BFGS algorithm (see BROYDEN (1970); Fletcher (1970); Goldfarb (1970); Shanno (1970)) in our numerical tests. It involves the construction of a sequence of approximation of the inverse of the Hessian matrix of the cost function J_a . We use a limited storage version of the algorithm called L-BFGS (see Nocedal (1980)). The method is presented in Algorithm 1, with the necessary adaptations to our problem.

As usual with this type of methods sequences are built and will be denoted as follows:

- \mathbf{t}^k : traction force vector,
- \mathbf{X}^k : solution of the state equation (12),
- \mathbf{u}^k : displacement field **on the ground**,
- ϕ^k : solution of the adjoint state equation (20),
- \mathbf{g}^k : gradient of the cost function J_a at point \mathbf{t}^k ,
- $\mathbf{d}^k, \mathbf{w}^k$: displacement directions for the optimization,
- ρ^k : optimal displacement step.

The number $k \geq 0$ represents the iteration number.

Notice that the underlying quadratic form in the cost function allow to compute explicitly the optimal step size at each iteration, via formula (28). The detailed computation is presented in the Appendix.

In the algorithm, the domain decomposition technique is used to solve the discrete state and adjoint state equations (12) and (20). Notice that the matrix K is symmetric and does not change during the iterative process, which means that the computation cost due to matrix factorization is reduced.

The algorithm iterates until the gradient of J_a becomes *sufficiently small*, as precised in the algorithm below.

Algorithm 1 Algorithm to minimize J_d

Require: \mathbf{t}^0 , \mathbf{u}_d and $\epsilon > 0$

 $k \leftarrow 0$

 Solve (12) with $\mathbf{t} = \mathbf{t}^0 \rightarrow \mathbf{X}^0$
 $\mathbf{u}^0 \leftarrow \mathcal{O}_R \mathbf{X}^0$

 Solve (20) with $\mathbf{u} = \mathbf{u}^0 \rightarrow \phi^0$

 Initial gradient: $\mathbf{g}^0 \leftarrow \alpha_0 M_{F_0} \mathbf{t}^0 + \alpha_1 M_{F_1} \mathbf{t}^0 + L_C^\top \phi^0$

 Initial direction: $\mathbf{d}^0 \leftarrow -\mathbf{g}^0$
 $H^0 \leftarrow \mathbf{I}_C$
while $\|\mathbf{g}^{k+1}\|_{\Gamma_C} < \epsilon \|\mathbf{g}^0\|_{\Gamma_C}$ **do**

 Solve (12) with $\mathbf{t} = \mathbf{t}^k \rightarrow \mathbf{W}^k$
 $\mathbf{w}^k \leftarrow \mathcal{O}_R \mathbf{W}^k$

 Compute step size ρ^k by (28) with $\mathbf{u} = \mathbf{u}^k$, $\mathbf{t} = \mathbf{t}^k$, $\mathbf{d} = \mathbf{d}^k$, $\mathbf{w} = \mathbf{w}^k$
 $\mathbf{t}^{k+1} = \mathbf{t}^k + \rho^k \mathbf{d}^k$
 $\mathbf{u}^{k+1} = \mathbf{u}^k + \rho^k \mathbf{w}^k$

 Solve (20) with $\mathbf{u} = \mathbf{u}^{k+1} \rightarrow \phi^{k+1}$
 \mathbf{g}^{k+1} in (15) with \mathbf{t}^{k+1} and ϕ^{k+1}

 Compute H^{k+1} via formula (30)

 $\mathbf{d}^{k+1} = -H^{k+1} \mathbf{g}^{k+1}$
 $k \leftarrow k + 1$
end while

3.4 Numerical tests and validation with synthetic data

The numerical application is performed by a synthetic test via simulation of a flat volcano as follows: A semi-infinite elastic domain at center with radius 100 km and extending down to $[0, 20]$ km (see Figure 3(a)), with the elasticity parameters: the Young modulus $E = 5000.0$ MPa and Poisson ratio $\nu = 0.25$. A horizontal circular fracture is defined with radius $r = 1$ km and located at $(x, y, z) = (0, 0, 0.3)$ km in the topo flat. The mesh generation and level set implementation is explained in Bodart et al. (2020, 2021). The ground surface is free to move and the other sides is fixed i.e. satisfy the boundary conditions on Γ_N and Γ_D respectively. We generated a synthetic solution \mathbf{u}_d associated to a discontinuous traction $\mathbf{t}^\pm = (0, 0, \pm 1.5)$ MPa, applied on a part of the fracture Γ_C , as shown in Figure

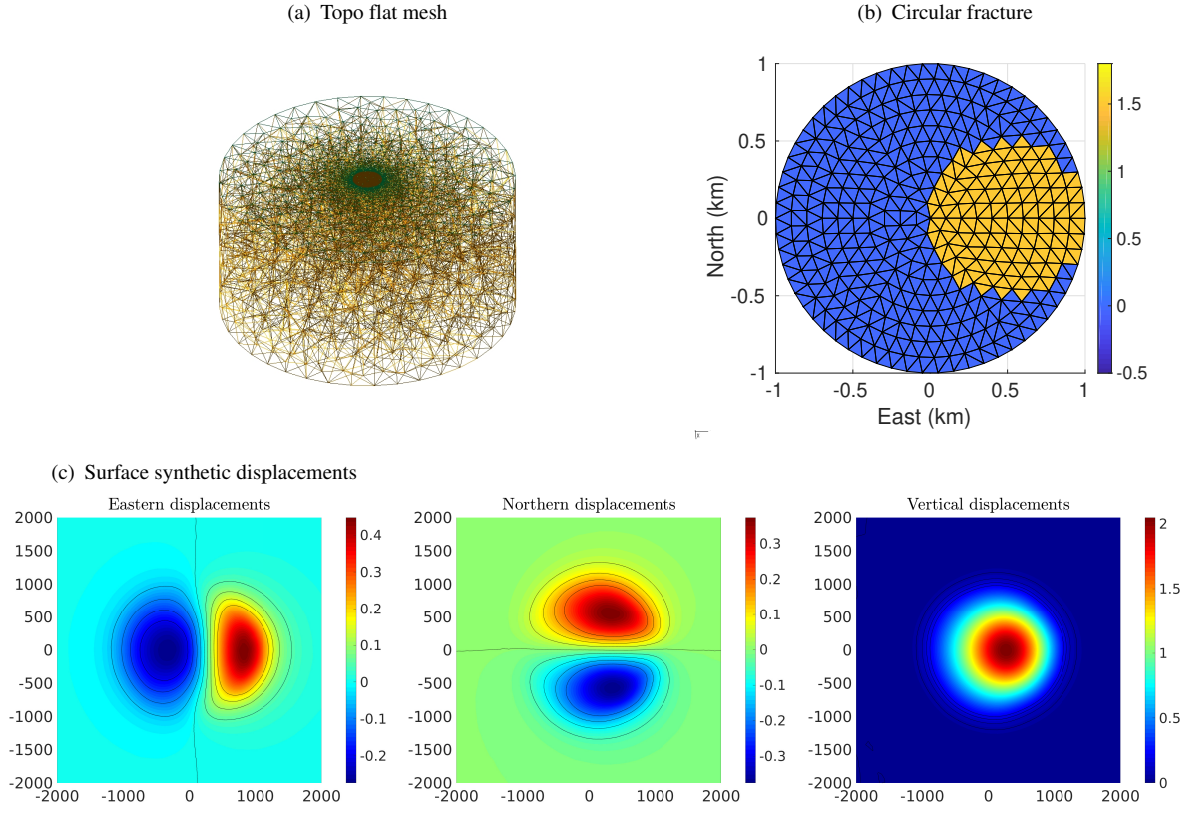


Figure 3. Configuration of the synthetic test corresponding to a horizontal circular fracture beneath a flat topography. (a) Progressive mesh used. The mesh has a radius of 100 km. The fine mesh region has a 1.5 km radius; the intermediate mesh size goes from 1.5 to 15 km radius; further away the mesh gets coarser. (b) The source is a pressurized disk with a 1 km radius located at different depths. The yellow patch is submitted to a normal traction $\mathbf{t}_{exact} = (0, 0, 1.5)$ MPa and the blue patch has a null traction $\mathbf{t}_{exact} = (0, 0, 0)$. (c) Surface synthetic displacements obtained by this pressure.

3(b) (the yellow patches) and $\mathbf{t}^\pm = (0, 0, 0)$ on the rest of the fracture (the blue patches). In some numerical experiments, this traction is interpreted as a pressure $\mathbf{p}^\pm = 1.5$ MPa applied on the yellow zone.

Finding a robust and efficient method to compute appropriate regularization parameters α_0 and α_1 for given mesh and noisy data, is not straightforward. The admissible solution \mathbf{t} is dependent on the chosen α_0 and α_1 and it is crucial to obtain a good approximation of the solution to the optimization problem. For choosing the optimal parameters, a series of tests is organized by using the graphical tools. Indeed, three important criteria are compared by varying α_0 and α_1 as follows: we set $\alpha_0 \in \{1.0E-12, 1.0E-11, \dots, 1.0E0\}$ and $\alpha_1 \in \{1.0E-5, 1.0E-4, \dots, 1.0E5\}$. Then we compare:

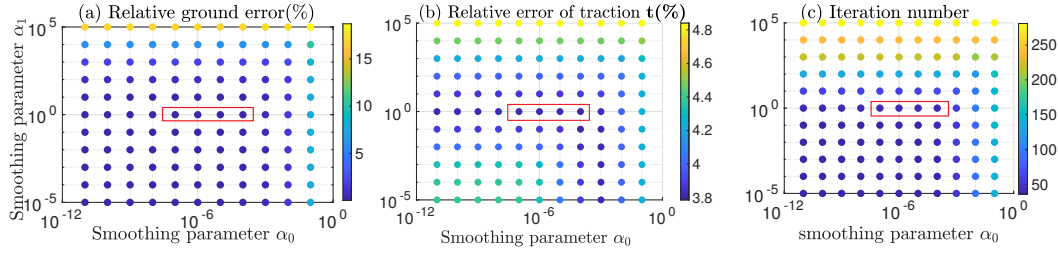


Figure 4. Systematic exploration of the smoothing parameters α_0 and α_1 for minimizing the cost function in equation (13). The source is located at -0.3 km depth beneath the flat topography. The acceptable combination of smoothing parameters is found by comparing (a) relative ground error, (b) relative error of traction on the disk and (c) iteration number. The source has 291 unknowns. The best compromises are indicated by the red boxes.

A relative ground error defined by:

$$E_{\mathbf{u}} = \frac{\int_{\Gamma_N} |\mathbf{u} - \mathbf{u}_d|^2 d\Gamma_N}{\int_{\Gamma_N} |\mathbf{u}_d|^2 d\Gamma_N} \times 100.$$

A relative error of traction \mathbf{t} on the fracture source, defined by:

$$E_{\mathbf{t}} = \frac{\sum |\mathbf{t} - \mathbf{t}_{exact}|^2}{\sum |\mathbf{t}_{exact}|^2 d\Gamma_C} \times 100.$$

Note that an inevitable numerical error is already produced to compute the traction \mathbf{t} on the fracture source. Using the non-conformal mesh to implement the fracture by level set method, is the reason of this error which can be reduced by using a finer mesh around the fracture.

finally, the iteration number for a given $\epsilon = 1.0E - 14$ in Algorithm 1 to satisfying the convergence criteria

$$\frac{\int_{\Gamma_C} |\mathbf{g}^{k+1}|^2 d\Gamma_C}{\int_{\Gamma_C} |\mathbf{g}^0|^2 d\Gamma_C} < \epsilon.$$

The numerical tests presented in Figure 4, show that, $\alpha_1 = 1$ can be chosen as an acceptable parameter. Moreover, it seems, the above criteria are less sensitive to α_0 between $1.0E - 7$ and $1.0E - 4$. Accordingly, we employ L-curve, another graphical tool, to trade off between two criteria: the misfit

$$\|\mathbf{u} - \mathbf{u}_d\| = \left(\int_{\Gamma_N} |\mathbf{u} - \mathbf{u}_d|^2 d\Gamma_N \right)^{\frac{1}{2}}, \quad (22)$$

and the norm of the traction or smoothing

$$\|\mathbf{t}\| = \left(\int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C \right)^{\frac{1}{2}}. \quad (23)$$

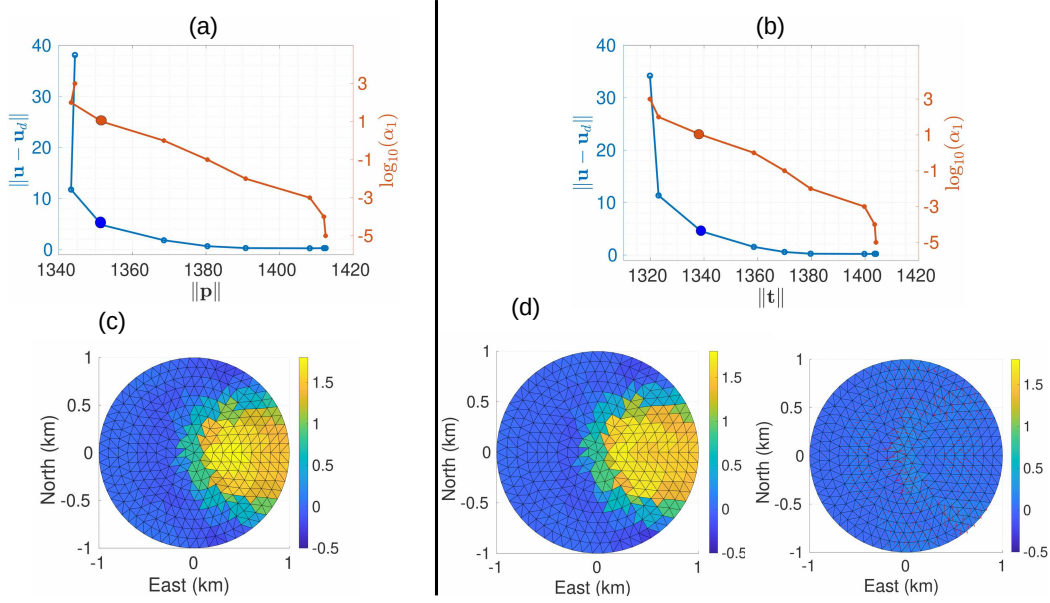


Figure 5. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (13)). The fracture is located at -0.3 km. (a) L-curve when solving for pressure \mathbf{p} . The larger points indicate the best compromise. (b) L-curve when solving for traction \mathbf{t} . The larger points indicate the best compromise. (c) Fracture pressure corresponding to the best $\alpha_1 = 1.0\text{E}1$. (d) Normal and (e) tangential tractions projected on the fracture corresponding to the best $\alpha_1 = 1.0\text{E}1$. Here, we set $\alpha_0 = 1.0\text{E} - 7$.

The L-curves are depicted by setting $\alpha_0 = 1.0\text{E} - 7$ and varying α_1 . Closest point to the origin shows the best compromise between misfit and smoothing, and it will be the acceptable α_1 . In particular case when we aim to find the pressure, the traction \mathbf{t} may be replaced by the pressure \mathbf{p} . In Figure 5, for the above fracture and mesh, two L-curves corresponding to both pressure \mathbf{p} and traction \mathbf{t} are illustrated.

Fracture pressure corresponding to $\alpha_1 = 1.0\text{E}1$ as an appropriate choice is presented in Figure 5(c). As shown in Figures 6 and 7, in 3D realistic volcanoes where we are interested in large-scale problems, the values around the $1.0\text{E}1$ are yet acceptable. Normal and tangential tractions are depicted in Figure 7. Note that we suppose $(0, 0, -1)$ as the unit normal vector \mathbf{n} .

We then aim to consider the impact of the fracture depth to the approximated admissible solution. To do that, the numerical experiments are repeated for the fracture at -0.9 km of depth. The best combination for $\alpha_1 = 1.0\text{E} - 1$ and α_0 between $1.0\text{E} - 7$ and $1.0\text{E} - 4$, in red box are shown in Figure 8. The L-curves depicted for \mathbf{p} and \mathbf{t} in Figure 9(a, b) show that the closest point to the origin for the 0.9 km is $\alpha_1 = 1.0\text{E}0$. The normal stress projected on the fracture presented in Figure 9(c)

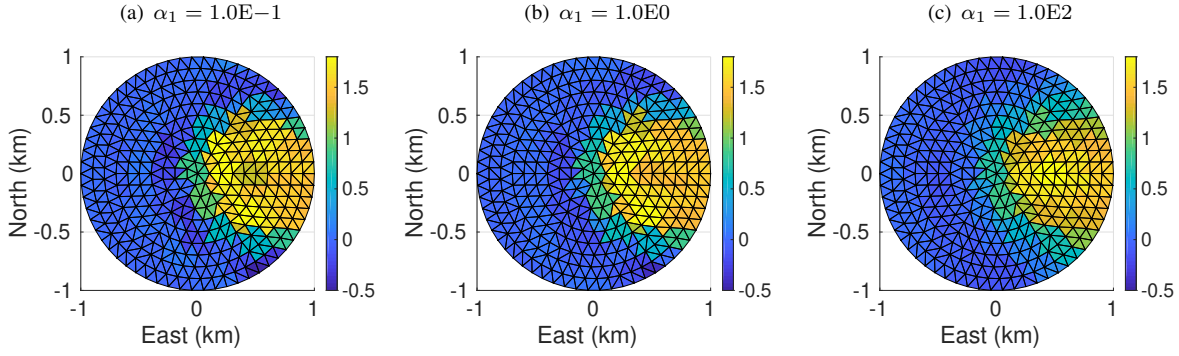


Figure 6. Fracture pressure located at -0.3 km, corresponding to (a) $\alpha_1 = 1.0E-1$, (b) $\alpha_1 = 1.0E0$ and (c) $\alpha_1 = 1.0E2$

compare to exact solution confirms an acceptable approximated solution. As previous case the values around the $1.0E0$ for α_1 are acceptable (see Figures ?? and ??). A comparison of the iteration number presented in Table 1, indicates that the number of iterations is augmented for the greater values of α_1 .

As seen in the numerical simulations, we may trust to the L-curves results to find an appropriate α_1 for more complicated geometry of fracture and mesh, which is clearly help us to computational cost.

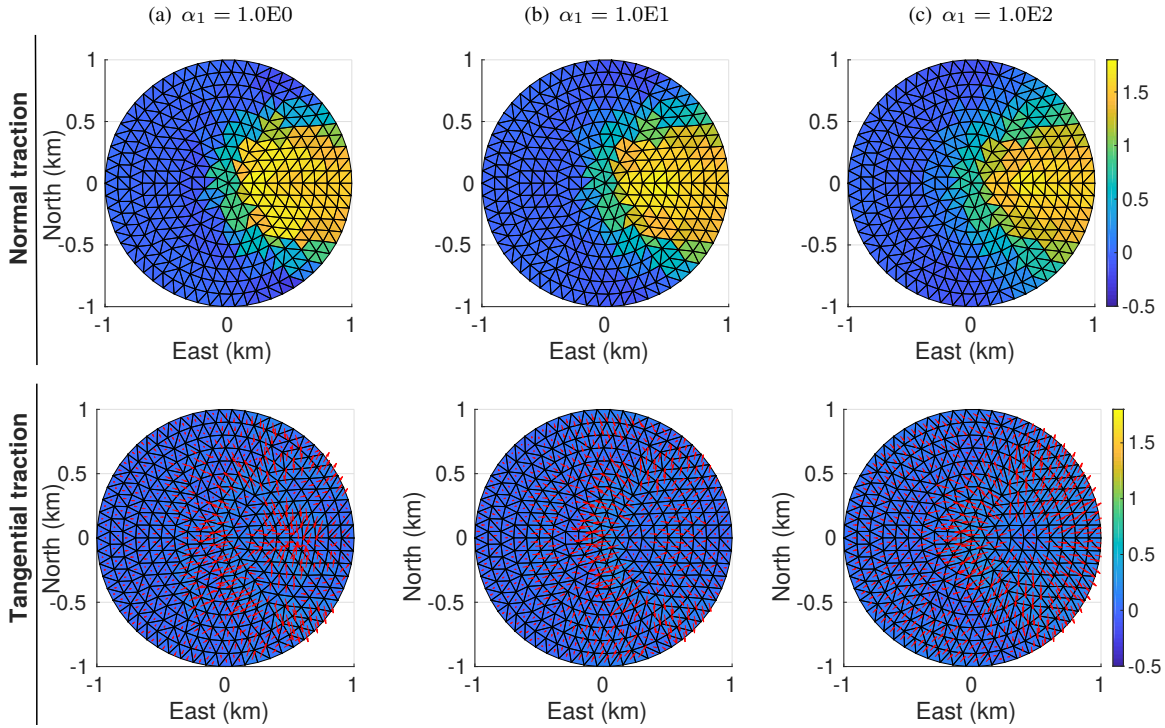


Figure 7. Normal and tangential tractions projected on the fracture located at -0.3 km, corresponding to (a) $\alpha_1 = 1.0E0$, (b) $\alpha_1 = 1.0E1$ and (c) $\alpha_1 = 1.0E2$. The directions of the tangential stress are shown by red vectors.

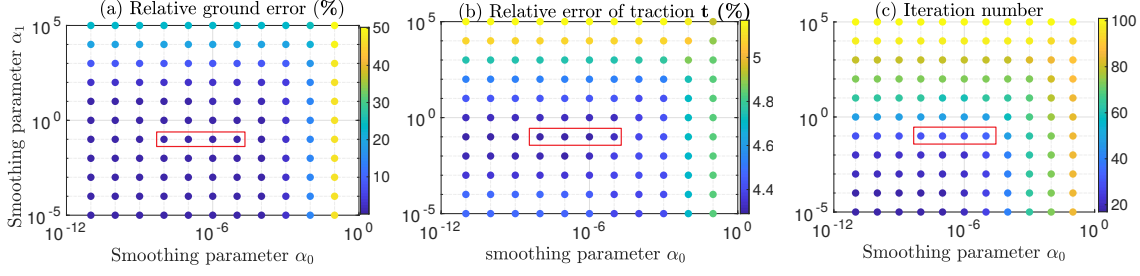


Figure 8. Systematic exploration of the smoothing parameters α_0 and α_1 for minimizing the cost function in equation (13). The source is located at -0.9 km depth beneath the flat topography. The acceptable combination of smoothing parameters is found by comparing (a) relative ground error, (b) relative error of traction on the disk and (c) iteration number. The source has 107 unknowns. The best compromises are indicated by the red boxes.

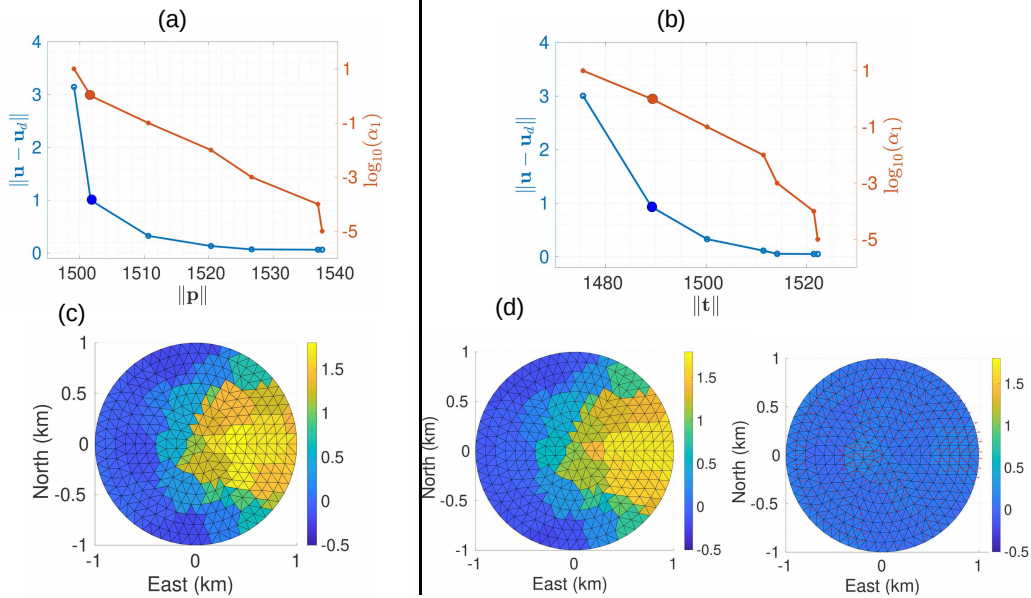


Figure 9. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (13)). The fracture is located at -0.9 km. (a) L-curve when solving for pressure \mathbf{p} . The larger points indicate the best compromise. (b) L-curve when solving for traction \mathbf{t} . The larger points indicate the best compromise. (c) Fracture pressure corresponding to the best $\alpha_1 = 1.0\text{E}1$. (d) Normal and (e) tangential tractions projected on the fracture corresponding to the best $\alpha_1 = 1.0\text{E}0$. Here, we set $\alpha_0 = 1.0\text{E} - 7$.

| Depths | α_1 | Unknown | Number of iteration | Number of unknowns |
|---------|------------|-----------------------|---------------------|--------------------|
| -0.3 km | 1.0E1 | Pressure \mathbf{p} | 71 | 291 |
| | | Traction \mathbf{t} | 84 | 873 |
| -0.9 km | 1.0E0 | Pressure \mathbf{p} | 48 | 107 |
| | | Traction \mathbf{t} | 67 | 321 |

Table 1. Comparison between the number of iterations for two depths -0.3 and -0.9 km and when the unknowns are pressure \mathbf{p} and traction \mathbf{t} for $\alpha_0 = 1.0\text{E}-7$.

4 TAKING INTO ACCOUNT THE EARTH-SATELLITE DIRECTIONS

4.1 Mathematical framework and new optimization problem

Before going further, let us first remark that the previous section can easily be adapted to the case where the measurements of the displacement field are performed only on a part of the ground $\widetilde{\Gamma}_N \subset \Gamma_N$. The cost function to be minimized becomes

$$J(\mathbf{t}) = \frac{1}{2} \int_{\widetilde{\Gamma}_N} (\mathbf{u} - \mathbf{u}_d)^\top \mathbf{C}^{-1} (\mathbf{u} - \mathbf{u}_d) d\Gamma_N + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 d\Gamma_C.$$

Rewriting the optimality conditions and adapting the algorithms is straightforward. This is what is actually implemented in our software.

However, assuming the displacement field to be known in cartesian coordinates is not realistic for the applications. In this section we aim at adapting our work to the case where measurements are provided by satellite radar interferometry (as e.g. in Fukushima et al. (????)). We will first state precisely what measurements are made and then adapt our methods to this new framework.

Let $\mathbf{u}_d = (u_x, u_y, u_z)^\top$ be a displacement field on Γ_N , written in cartesian coordinates. A satellite will aim at the ground to measure a displacement, and the resulting measurement will be in the form $\mathbf{p}^\top \mathbf{u}_d$ where the aiming direction (also called earth-satellite direction) $\mathbf{p} = (p_x, p_y, p_z)$ is a unit vector oriented from the ground to the satellite. Let then $N \geq 1$ be the number of satellites, each associated to a direction $\mathbf{p}_i = (p_{ix}, p_{iy}, p_{iz})$ for $i = 1, \dots, N$. We can build a matrix $\mathbf{P} : \mathbb{R}^3 \rightarrow \mathbb{R}^N$ as $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)^\top$, so that the actual measurement is

$$\mathbf{R}_d = \mathbf{P} \mathbf{u}_d.$$

The vector field $\mathbf{R}_d \in \mathbb{R}^N$ is defined on Γ_N , and the matrix \mathbf{P} also operates on the solutions of System (1)–(4). Hence, given the field \mathbf{R}_d , our new optimization problem consists now in finding the traction vector \mathbf{t} minimizing the following cost function:

$$J(\mathbf{t}) = \frac{1}{2} \int_{\Gamma_N} (\mathbf{P} \mathbf{u} - \mathbf{R}_d)^\top \mathbf{C}^{-\frac{1}{2}} \mathbf{C}^{-\frac{1}{2}} (\mathbf{P} \mathbf{u} - \mathbf{R}_d) d\Gamma_N + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 d\Gamma_C, \quad (24)$$

where $\alpha_0 > 0$ and $\alpha_1 > 0$ are the regularization parameters. The optimality conditions for this problem are derived in the same way than in Section 3.1. We will not go into details about this and focus on the discrete version of the problem.

Using the definitions of Sections 2.2 and 3.2, and in view of (19) and (24) the discrete cost function becomes

$$J_d(\mathbf{t}) = \frac{1}{2}(\mathbf{P}\mathcal{O}_R\mathbf{X} - \mathbf{R}_d)^\top \mathbf{C}^{-\frac{1}{2}} M_G \mathbf{C}^{-\frac{1}{2}} (\mathbf{P}\mathcal{O}_R\mathbf{X} - \mathbf{R}_d) + \frac{\alpha_0}{2}(\mathbf{t}^\top M_{F_0} \mathbf{t}) + \frac{\alpha_1}{2}(\mathbf{t}^\top M_{F_1} \mathbf{t}). \quad (25)$$

where X is the solution of the discrete system (12). This finite dimensional problem then boils down to finding the saddle point of the following Lagrangian

$$\mathcal{L}_d(\mathbf{X}, \mathbf{t}, \phi) = J_d(\mathbf{t}) - \langle \mathbf{K}\mathbf{X} - (L_\Omega \mathbf{f} + L_C \mathbf{t}), \phi \rangle.$$

Computing the partial derivative of \mathcal{L}_d and then cancelling them, leads to the discrete adjoint problem (14):

$$\mathbf{K}\phi = \mathcal{O}_R^\top \mathbf{P}^\top \mathbf{C}^{-\frac{1}{2}} M_G \mathbf{C}^{-\frac{1}{2}} (\mathbf{P}\mathcal{O}_R\mathbf{X} - \mathbf{R}_d), \quad (26)$$

and the gradient of J_d is:

$$\nabla J_d = \alpha_0 M_{F_0} \mathbf{t} + \alpha_1 M_{F_1} \mathbf{t} + L_C^\top \phi. \quad (27)$$

The minimization Algorithm 1 can then be adapted to the new adjoint state and gradient to obtain the numerical approximation of the traction \mathbf{t} on the crack by using the surface measurements provided by radar interferometry \mathbf{r}_d .

4.2 Applications to synthetic test

The numerical applications here are proceeded in three different steps: First, we aim to the adaptation of the theoretical results in Earth-Satellite directions. For the sake of simplicity, we assume the covariance is an identity matrix. Next, in order to reduce the observed data, the identity covariance matrix is reduced to the nodal mesh points. We then consider a problem with dense covariance matrix adapted to a limited number of observed data.

The vector directions are chosen by using InSAR satellite directions, listed in Table 2. The numerical tests are performed with one to four radar looks to confirm the theoretical results. We set the same configuration as previous section for the problems in Cartesian coordinate. The horizontal circular fracture beneath the flat topography located at 0.3 km (see Figure 3). We still take $\alpha_0 = 10^{-7}$ and by depicting the L-curves, we find the best α_1 with the best compromise between misfit and the traction norm \mathbf{t} or pressure \mathbf{p} (see Figure 10) Normal and tangential tractions projected on the fracture corresponding to the best $\alpha_1 = 1$ are presented in Section supplementary Material, Figure ???. A comparison between the iteration numbers listed in Table 3 for the observed synthetic data provided

| | | |
|-------------|---------------|--------------------|
| S4: | D R 151 0.056 | [0.56 -0.14 0.80] |
| S6: | A R 144 0.056 | [-0.66 -0.17 0.73] |
| TSX: | A R 036 0.031 | [-0.54 0.11 0.83] |
| TSX: | D R 036 0.031 | [0.59 -0.13 0.80] |

Table 2. InSAR Unit vector directions

by different number of radar looks, show that the use of more radar looks leads to a decrease in the number of iteration.

In a more realistic scenario, the number of observed data are limited. In the previous numerical experiments, we assumed a synthetic \mathbf{u}_d in P2 finite elements. The degrees of freedom's nodes of a classical P2 Lagrange elements on a 3D tetrahedron, are located on the vertices or nodal mesh points and the midpoints of the edges. In realistic volcano phenomenons (three-dimensional elasticity problem), using the P2 finite element guarantees a good approximated solution. However, we aim to reduce the synthetic \mathbf{u}_d to nodal mesh points on the ground surface as a P1 finite elements. On the other side, to conserve the P2 finite elements for the elasticity problem, we should conserve the dimension of the covariance $\mathbf{C}^{-\frac{1}{2}}$ and the mass matrix M_G in a P2 elements. Therefore, from implementation

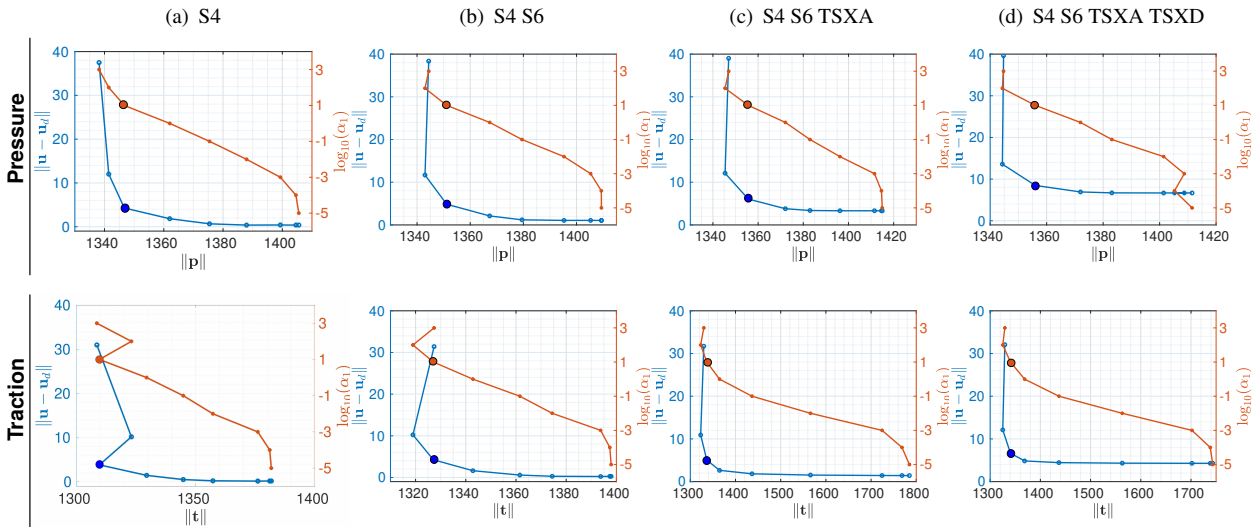


Figure 10. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (24)) taking into account the earth-satellite directions with identity covariance matrix. The fracture is located at -0.3 km, when solving for pressure \mathbf{p} and traction \mathbf{t} . The larger points indicate the best compromise. (a) in S4 (b) in S4 and S6 (c) in S4, S6 and TSXA (d) in S4, S6, TSXA and TSXD directions. Here, we set $\alpha_0 = 1.0\text{E} - 7$.

| Radar look | α_1 | Unknown | Number of iteration | Number of unknowns |
|-----------------|------------|-----------------------|---------------------|--------------------|
| S4 | 1.0E1 | Pressure \mathbf{p} | 93 | 291 |
| | | Traction \mathbf{t} | 108 | 873 |
| S4 S6 | 1.0E1 | Pressure \mathbf{p} | 59 | 291 |
| | | Traction \mathbf{t} | 80 | 873 |
| S4 S6 TSXA | 1.0E1 | Pressure \mathbf{p} | 51 | 291 |
| | | Traction \mathbf{t} | 58 | 873 |
| S4 S6 TSXA TSXD | 1.0E1 | Pressure \mathbf{p} | 50 | 291 |
| | | Traction \mathbf{t} | 58 | 873 |

Table 3. Comparison between the number of iterations taking into account the earth-satellite directions with identity covariance matrix in InSAR unit vector directions. The fracture is located at -0.3 km, when solving for pressure \mathbf{p} and traction \mathbf{t} . Here, we set $\alpha_0 = 1.0\text{E}-7$.

point of view, the degrees of freedom's related to the midpoints of the edges in $\mathbf{C}^{-\frac{1}{2}}$ and M_G are set to zero. Despite the increase in the iterations number, the numerical experiments presented still show very satisfactory results (see Section Supplementary Material Figure ??).

Another step to approach the reality is using the dense covariance matrices for limited number of data, obtained by InSAR and cGNSS.

To do that, we are using the Defvolc interfaces The dense covariance matrices are provided by DefVolc, pre and post-processor software. We started by creating the synthetic \mathbf{u}_d and the dense covariance matrix in S4 radar look. The results obtained in Figure 11 and first row of Table 4, confirm the adaptability of our method to the dense covariance matrices. In reality, most of the time, the atmospheric contribution caused masked and noisy InSAR data. Therefore, some numerical experiments is performed by creating masked and noisy synthetic \mathbf{u}_d and covariance matrix. The numerical experiments are presented in Figure 12. The L-curves are depicted for synthetic \mathbf{u}_d projected to S4 an S6 directions. First without any mask and noise in the data and then different possible cases by adding the mask and noise to the synthetic data. We summarized appropriate choices for α_1 and the number of iterations to achieve the convergence with these α_1 , for different cases of synthetic \mathbf{u}_d , in Table 4. Despite of choosing a greater α_1 , for the masked data, the number of iteration is reduced. However, for the noisy data the convergence of the inversion process is much more harder. Moreover the vector size of the synthetic \mathbf{u}_d for each case are listed in Table 4.

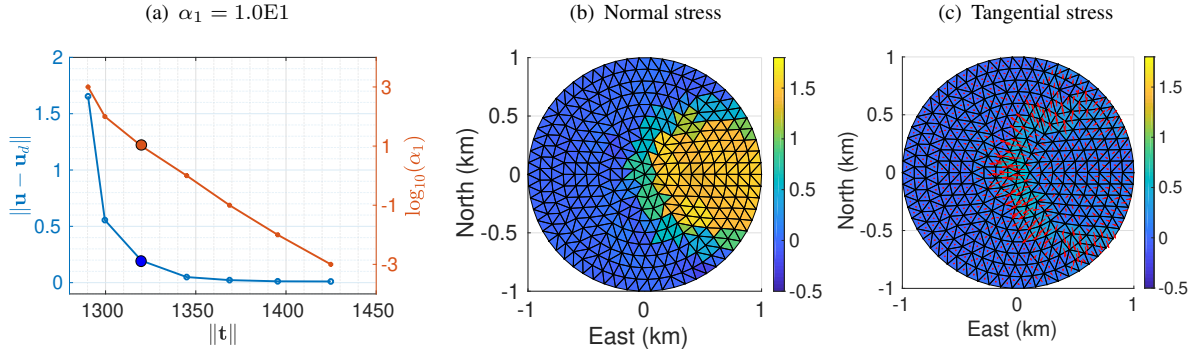


Figure 11. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (24)) taking into account the earth-satellite directions with the dense covariance matrix in S4 radar look direction. The fracture is located at -0.3 km, when solving for the traction \mathbf{t} . The larger points indicate the best compromise. (b) Normal (c) tangential tractions projected on the fracture corresponding to the best $\alpha_1 = 1.0E1$. Here, we set $\alpha_0 = 1.0E - 7$.

4.2.1 Noise influence

| Radar look | α_1 | Number of iteration | Vector size \mathbf{u}_d |
|---------------------------|------------|---------------------|----------------------------|
| S4 | 1.0E1 | 98 | 1616 |
| S4 S6 | 1.0E1 | 102 | 3232 |
| S4 S6 with mask | 1.0E2 | 56 | 1087 |
| S4 S6 with noise | 1.0E2.5 | 344 | 3232 |
| S4 S6 with mask and noise | 1.0E3 | 306 | 1087 |

Table 4. Comparison between the number of iterations for fracture at 0.3 km depth and when the unknowns are the traction \mathbf{t} for different radar looks with dense covariance matrix.

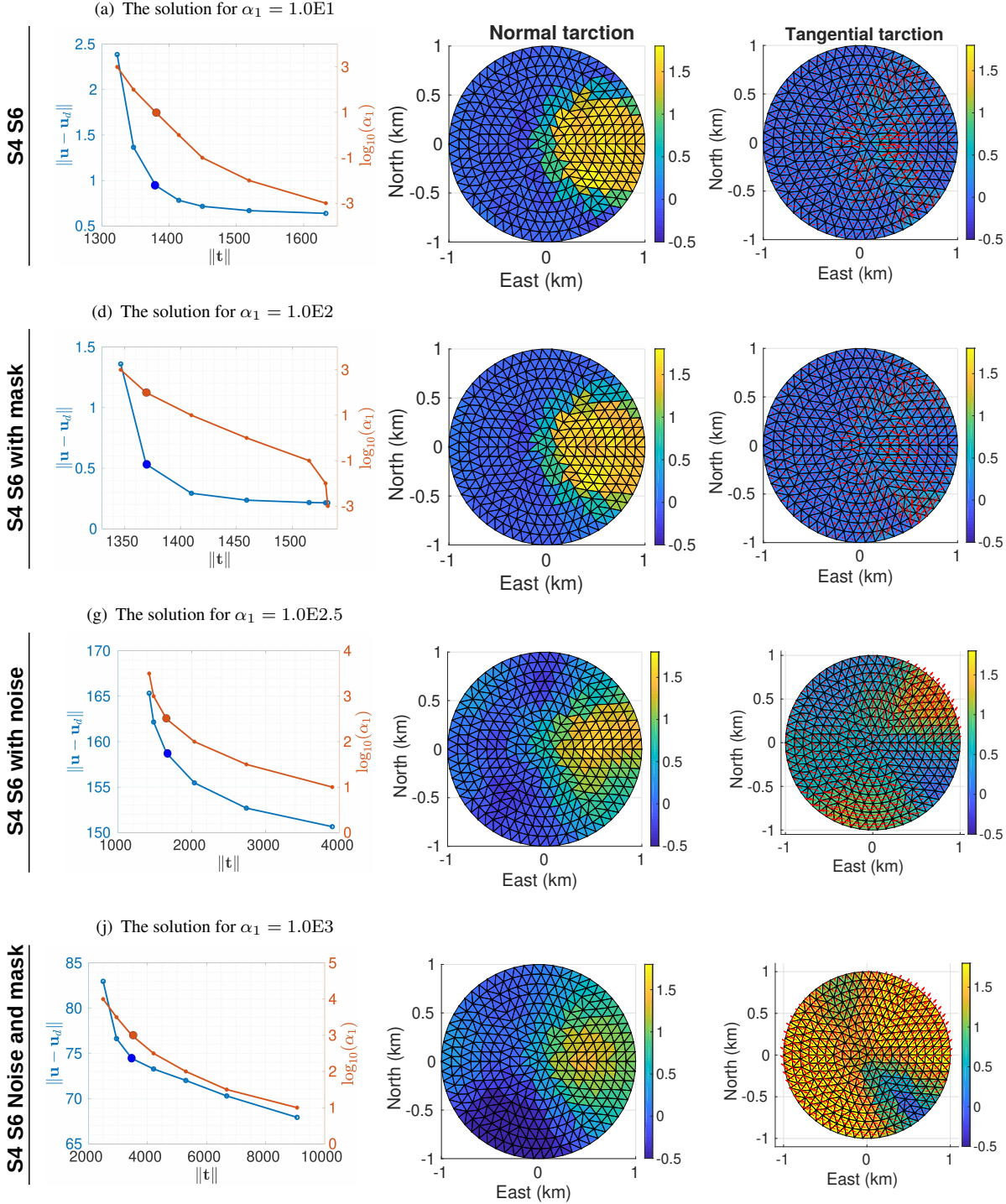


Figure 12. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (24)) taking into account the earth-satellite directions with the dense covariance matrix. The fracture is located at -0.3 km, when solving for the traction t . The larger points indicate the best compromise. The normal and tangential traction are presented for the appropriate α_1 . The synthetic data u_d is projected in S4 and S6 radar looks. (a) without any noise and mask, (d) with mask, (g) with noise (j) with noise and mask. Here, we set $\alpha_0 = 1.0\text{E} - 7$.

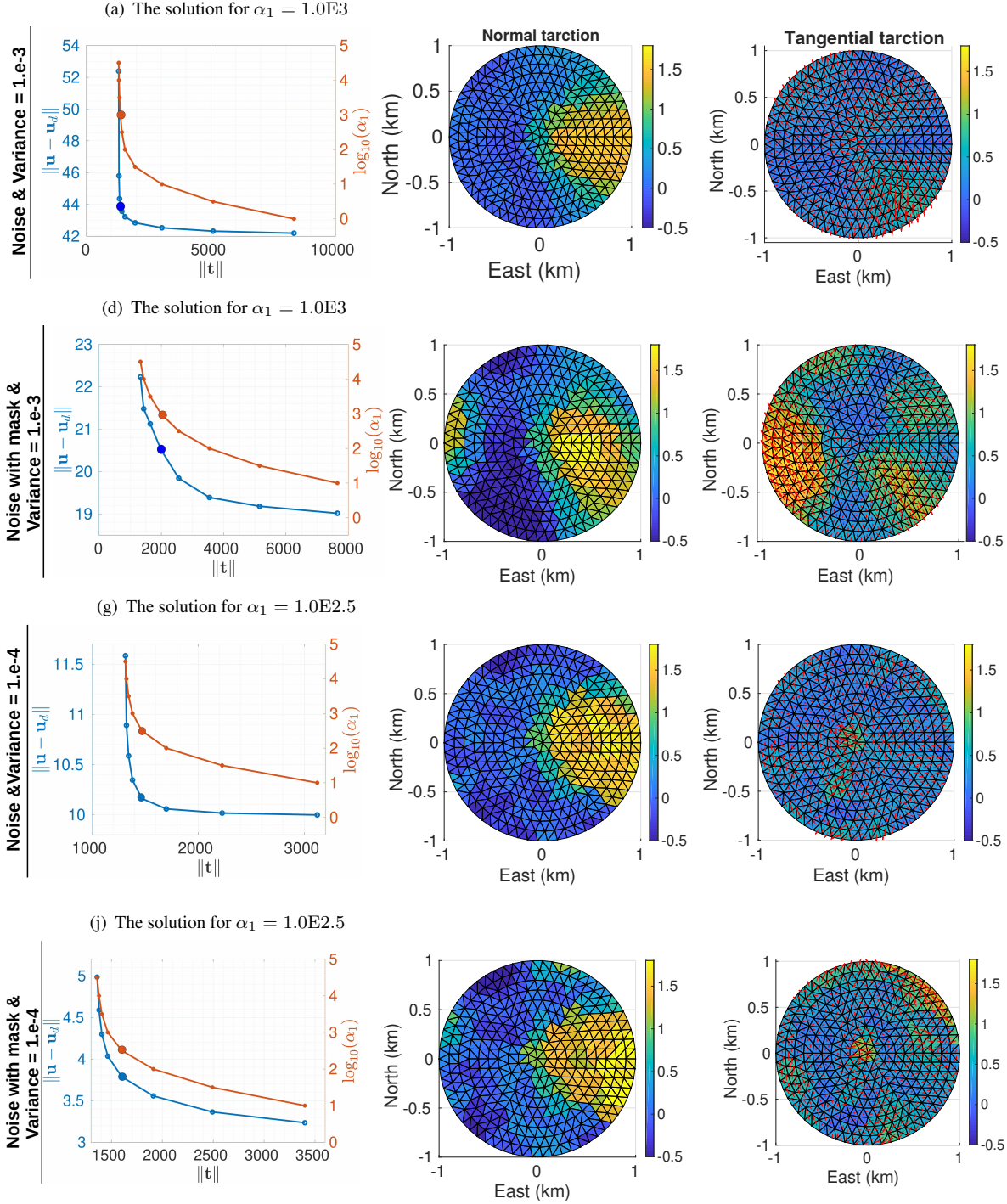


Figure 13. L-curves used to find α_1 representing the best compromise between the data fit (equation (22)) and the smoothing (equation (23)) in the cost function (equation (24)) taking into account the earth-satellite directions with the dense covariance matrix to consider the noise effect to the solutions. The fracture is located at -0.3 km, when solving for the traction t . The larger points indicate the best compromise. The normal and tangential traction are presented for the appropriate α_1 . The synthetic data u_d is projected in S4 and S6 radar looks. (a) with noise and variance 1.e-3 (d) with noise and mask and variance 1.e-3, (g) with noise and variance 1.e-4, (j) with noise and mask and variance 1.e-4 Here, we set $\alpha_0 = 1.0E - 7$.

| Radar look | Variance | α_1 | Number of iteration | Vector size \mathbf{u}_d |
|---------------------------|-------------------|------------------|---------------------|----------------------------|
| S4 S6 with noise | $1.0\text{E} - 2$ | $1.0\text{E}2.5$ | 344 | 3232 |
| S4 S6 with mask and noise | $1.0\text{E} - 2$ | $1.0\text{E}3$ | 306 | 1087 |
| S4 S6 with noise | $1.0\text{E} - 3$ | $1.0\text{E}3$ | 315 | 3232 |
| S4 S6 with mask and noise | $1.0\text{E} - 3$ | $1.0\text{E}3$ | 117 | 1087 |
| S4 S6 with noise | $1.0\text{E} - 4$ | $1.0\text{E}2.5$ | 191 | 3232 |
| S4 S6 with mask and noise | $1.0\text{E} - 4$ | $1.0\text{E}2.5$ | 55 | 1087 |

Table 5. Comparison between the number of iterations for fracture at 0.3 km depth and when the unknowns are the traction \mathbf{t} for different radar looks with dense covariance matrix for noise produced by different variance.

ACKNOWLEDGMENTS

CONCLUSION

5 APPENDIX

5.1 The optimal step size

This section is dedicated to some mathematical details for interested readers. First part is concerned to compute the optimal step size. To simplify the presentation, let us introduce c_N defined on Γ_C

$$c_N(\mathbf{w}, \mathbf{w}) = \int_{\Gamma_N} \mathbf{w}^\top \mathbf{C}^{-1} \mathbf{w} \, d\Gamma_N,$$

so that the cost functional in (13) becomes

$$J(\mathbf{t}) := \frac{1}{2} c_N(\mathbf{u} - \mathbf{u}_d, \mathbf{u} - \mathbf{u}_d) + \frac{\alpha_0}{2} \int_{\Gamma_C} |\mathbf{t}|^2 \, d\Gamma_C + \frac{\alpha_1}{2} \int_{\Gamma_C} |\nabla \mathbf{t}|^2 \, d\Gamma_C.$$

Then the directional derivative of J in the direction of a given \mathbf{d} reads

$$\left\langle \frac{\partial J}{\partial \mathbf{t}}(\mathbf{t}), \mathbf{d} \right\rangle_{\alpha_0, \alpha_1} = c_N(\mathbf{u} - \mathbf{u}_d, \mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{t} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{t} \cdot \nabla \mathbf{d}) \, d\Gamma_C,$$

where \mathbf{w} is the solution to (5) with $\mathbf{t} = \mathbf{d}$. Therefore, we may compute the optimal step size ρ , with a search direction d by solving

$$\left\langle \frac{\partial J}{\partial \mathbf{t}}(\mathbf{t} + \rho \mathbf{d}), \mathbf{d} \right\rangle_{\alpha_0, \alpha_1} = 0,$$

that is

$$c_N(\mathbf{u} + \rho \mathbf{w} - \mathbf{u}_d, \mathbf{w}) + \alpha_0 \int_{\Gamma_C} ((\mathbf{t} + \rho \mathbf{d}) \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} ((\nabla \mathbf{t} + \rho \nabla \mathbf{d}) \cdot \nabla \mathbf{d}) \, d\Gamma_C = 0,$$

which gives

$$\begin{aligned} & \rho \left[c_N(\mathbf{w}, \mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{d} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{d} \cdot \nabla \mathbf{d}) \, d\Gamma_C \right] + c_N(\mathbf{u} - \mathbf{u}_d, \mathbf{w}) \\ & + \alpha_0 \int_{\Gamma_C} (\mathbf{t} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{t} \cdot \nabla \mathbf{d}) \, d\Gamma_C = 0. \end{aligned}$$

The optimal step size is therefore

$$\rho^* = - \frac{c_N(\mathbf{u} - \mathbf{u}_d, \mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{t} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{t} \cdot \nabla \mathbf{d}) \, d\Gamma_C}{c_N(\mathbf{w}, \mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{d} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{d} \cdot \nabla \mathbf{d}) \, d\Gamma_C}. \quad (28)$$

5.2 The optimal step size for Earth-Satellite direction

The optimal step size for Earth-Satellite direction as previous, is still obtained by solving

$$\left\langle \frac{\partial J}{\partial \mathbf{t}}(\mathbf{t} + \rho \mathbf{d}), \mathbf{d} \right\rangle_{\alpha_0, \alpha_1} = 0$$

that is

$$c_N(\mathbf{P}\mathbf{u} + \rho \mathbf{P}\mathbf{w} - \mathbf{r}_d, \mathbf{P}\mathbf{w}) + \alpha_0 \int_{\Gamma_C} ((\mathbf{t} + \rho \mathbf{d}) \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} ((\nabla \mathbf{t} + \rho \nabla \mathbf{d}) \cdot \nabla \mathbf{d}) \, d\Gamma_C = 0,$$

where \mathbf{w} is still the solution to (5) with $\mathbf{t} = \mathbf{d}$. This gives the optimal step size

$$\rho^* = - \frac{c_N(\mathbf{P}\mathbf{u} - \mathbf{r}_d, \mathbf{P}\mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{t} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{t} \cdot \nabla \mathbf{d}) \, d\Gamma_C}{c_N(\mathbf{P}\mathbf{w}, \mathbf{P}\mathbf{w}) + \alpha_0 \int_{\Gamma_C} (\mathbf{d} \cdot \mathbf{d}) \, d\Gamma_C + \alpha_1 \int_{\Gamma_C} (\nabla \mathbf{d} \cdot \nabla \mathbf{d}) \, d\Gamma_C}. \quad (29)$$

5.3 L-BFGS Update

The second part is concerned to L-BFGS Update in minimization Algorithm 1

$$H^{k+1} = (I - \theta^k \mathbf{s}^k \mathbf{y}^{k\top}) H^k (I - \theta^k \mathbf{y}^k \mathbf{s}^{k\top}) + \theta^k \mathbf{s}^k \mathbf{s}^{k\top}, \quad (30)$$

with

$$\mathbf{s}^k = \mathbf{t}^{k+1} - \mathbf{t}^k, \quad \mathbf{y}^k = \mathbf{g}^{k+1} - \mathbf{g}^k, \quad \theta^k = \frac{1}{\mathbf{y}^{k\top} \mathbf{s}^k}.$$

REFERENCES

- Bodart, O., Cayol, V., Court, S., & Koko, J., 2016. Xfem-based fictitious domain method for linear elasticity model with crack, *SIAM Journal on Scientific Computing*, **38**(2), B219–B246.
- Bodart, O., Cayol, V., Dabaghi, F., & Koko, J., 2020. Fictitious domain method for an inverse problem in volcanoes., *Springer, Domain Decomposition Methods in Science and Engineering XXV. DD 2018. Lecture Notes in Computational Science and Engineering*, **138**.
- Bodart, O., Cayol, V., Dabaghi, F., & Koko, J., 2021. An inverse problem in an elastic domain with a crack : a fictitious domain approach, *submitted*.
- BROYDEN, C. G., 1970. The convergence of a class of double-rank minimization algorithms 1. general considerations, *IMA Journal of Applied Mathematics*, **6**(1), 76–90.
- Fletcher, R., 1970. A new approach to variable metric algorithms, *The Computer Journal*, **13**(3), 317–322.
- Fukushima, Y., Cayol, V., & Durand, P., 2000. Finding realistic dike models from interferometric synthetic aperture radar data: The february 2000 eruption at piton de la fournaise, *Journal of Geophysical Research: Solid Earth*, **110**(B3).
- Goldfarb, D., 1970. A family of variable-metric methods derived by variational means., *Math. Comp.*, **24**, 23–26.
- Nocedal, J., 1980. Updating quasi-newton matrices with limited storage, *Mathematics of Computation*, **35**(151), 773–782.
- Renard, Y. & Pommier, J., <http://home.gna.org/getfem>. Getfem++. An Open Source generic C++ library for finite element methods.
- Shanno, D. F., 1970. Conditioning of quasi-newton methods for function minimization., *Math. Comp.*, **24**, 647–656.
- Tarantola, A., 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation*, Society for Industrial and Applied Mathematics.