An inverse problem in an elastic domain with a crack : a fictitious domain approach

Oliver Bodart, Valérie Cayol, Farshid Dabaghi, Jonas Koko

An inverse problem in an elastic domain with a crack : a fictitious domain approach

Oliver Bodart*, Valérie Cayol †, Farshid Dabaghi‡ and Jonas Koko§

Post-print, *Computational Geosciences*, 2022
https://doi.org/10.1007/s10596-021-10121-7
Published online: 2022 January 07; Accepted: 2021 November 29. Received: 2020 November 28

Abstract

An inverse problem applied to volcanology is studied. It consists in the determination of the variable pressure applied to a crack in order to fit observed ground displacements. The deformation of the volcano is assumed to be governed by linear elasticity. The direct problem is solved via a fictitious domain method, using a finite element discretization of XFEM type. The ground misfit is minimized using a combination of a domain decomposition and optimatily conditions. The gradient of the cost function is derived from a sensitivity analysis. Discretization of the problem is studied. Numerical tests (in 2D and 3D) are presented to illustrate the effectiveness of the proposed approach. In particular, we find that a quasi-Newton method is more efficient than a conjugate gradient method for solving the optimization problem.

keywords: Inverse problem – Crack – Fictitious domain – Linear elasticity – Conjugate gradient

1 Introduction

The computational cost is a key issue for recovering information on cracks (e.g., location, propagation, applied traction) for several problems in Geophysics: magma-filled cracks in volcanoes (e.g., [23]), seismogenic faults in tectonics (e.g., [18]), etc. The determination of crack characteristics from ground surface observations leads to a large scale constrained optimization problem. The cost functional controls the residual between the observed values and the computed values. The constraints are the direct problem, that is, the mechanical equilibrium equations. The geophysics scientific community has been addressing this type of problem for a long time, assuming the crust of the Earth to be linear elastic.

*Institut Camille Jordan, Université Jean Monnet Saint-Étienne, CNRS UMR 5208, F-42023 Saint-Etienne, France, email: olivier.bodart@univ-st-etienne.fr

†Université Clermont Auvergne, CNRS, IRD, OPGC, Laboratoire Magmas et Volcans, F-63000 Clermont-Ferrand, France, email: valerie.cayol@uca.fr

‡Institut Camille Jordan, Université Jean Monnet, F-42023 Saint-Etienne, France, email: farshid.dabaghi@univ-st-etienne.fr

§LIMOS, Université Clermont Auvergne – CNRS UMR 6158, Campus des Cézeaux, BP 10448, F-63173 Aubière Cedex, France, email: koko@isima.fr.

In a vast majority of works, domain decomposition or fictitious domains methods are not used to solve the direct problem. In [10], boundary elements are used to solve the direct problem, and the crack and a constant pressure inside the crack are identified *via* a Monte Carlo type method. The drawback of the method is that taking heterogeneities in the solid into account is not straightforward. In [8] and [24], the authors solve the direct problem with a finite element technique, combined with a so-called "Green Function" method to build an approximation of the operator of the ground surface displacements with respect to the pressurized crack. In both approaches, the crack location is fixed. In [1], a domain decomposition method is used with a conformal mesh for the domain and the crack. As a consequence, the domain has to be re-meshed if the crack configuration changes. In [6], a fictitious domain method using XFEM is proposed for solving the direct problem without re-meshing the domain when the crack configuration changes. In this work, we use this method to solve the systems involved in our optimal control problem. This method is inspired by XFEM, since it consists partially in cutting the basis functions near or around the interfaces. But, unlike XFEM, the finite element spaces are not enriched with singular functions, for several reasons. First, we aim at dealing with 3D realistic problems in Earth sciences. The basis enrichment yields a high computational cost. As solving inverse problems imply that the direct problem and its adjoint are solved iteratively, we need a fast solver. Second, the extra precision induced by the enrichment is not needed in our case: we are not interested in tracking the crack propagation, nor in small-scale phenomenons. Third, not performing the enrichment helps us avoid another computational cost issue: minimizing the number of updates when the position of the crack is modified. Since the degrees of freedoms considered on the underlying mesh do not match the crack and its extension, a good approximation of the solution is not guaranteed around these surfaces. However, we implemented a local refinement strategy in the area of the interface. This has improved the approximated solution, especially the ground displacement which we are interested in. Therefore, the cost versus precision ratio of our method is satisfactory for this application.

Instead of the original XFEM, an artificial extension of the crack is considered. We impose a homogeneous displacement jump condition with a Lagrange multiplier to ensure the continuity across this extension. In this paper, we focus on the case where the crack location and shape are known, and we aim at identifying the (non constant) pressure exerted on the crack. This problem writes as an optimal control problem. In [7] a preliminary version of this work was presented. The proceeding discussed the possibility to apply a domain decomposition method for such a problem. Therefore, the focus was on the discrete problem and its adjoint. Only an elementary numerical example was presented. In this article a full sensitivity analysis is written, efficient algorithms for the optimal control problem are studied and the numerical tests deal with non constant pressures.

The paper is organized as follows : in Section 2, the general framework and the problem under consideration are presented. In Section 3, optimality conditions are derived, and the gradient of the cost function J is computed. Section 4 presents the domain decomposition method for the state equation and its consequences on the formulation of the optimal control problem. Section 6 presents the optimization methods which will be tested on the problem. Section 5 is devoted to more practical aspects of our problem: discretization and domain decomposition. Numerical results are then presented in Section 7. Finally a brief conclusion is made in Section 8.

2 Problem setting

Let Ω be a bounded open set in $\mathbb{R}^d$, $d = 2, 3$ with smooth boundary $\partial\Omega := \Gamma_{\rm D} \cup \overline{\Gamma_{\rm N}}$ where $\Gamma_{\rm D}$ and $\Gamma_{\rm N}$ are of non-zero measure and $\Gamma_{\rm D} \cap \Gamma_{\rm N} = \emptyset$. We assume that Ω is occupied by an elastic solid and subject to external forces of density $\mathbf{f} \in \mathbf{L}^2(\Omega)$. we denote by $\mathbf{u}$ the displacement field of the solid. The Cauchy stress $\sigma(\mathbf{u})$ and strain $\varepsilon(\mathbf{u})$ are given by

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

where (λ, μ) are the Lamé coefficients, $\mathbf{I}_{\mathbb{R}^d}$ the identity tensor, and $\mathrm{Tr}(\cdot)$ the matrix trace. Consider a crack $\Gamma_{\rm C} \subset \Omega$ represented by a curve ($d = 2$) or a surface ($d = 3$) parametrized by an injective mapping. The deformation field of the solid is supposed to satisfy the following elastostatic system:

Find $\mathbf{u} \in \mathbf{V} = \{\mathbf{v} \in \mathbf{H}^1(\Omega) \mid \mathbf{v} = 0 \text{ on } \Gamma_{\rm D}\}$, such that:

$$\begin{aligned} -\mathrm{div}\ \sigma(\mathbf{u}) &= \mathbf{f} \text{ in } \Omega, && (2.1)\\ \mathbf{u} &= 0 \text{ in } \Gamma_{\rm D}, && (2.2)\\ \sigma(\mathbf{u})\mathbf{n} &= 0 \text{ on } \Gamma_{\rm N}, && (2.3)\\ \sigma(\mathbf{u})\mathbf{n} &= -p\mathbf{n} \text{ on } \Gamma_{\rm C}. && (2.4) \end{aligned}$$

On $\Gamma_{\rm N}$, we denote by $\mathbf{n}$ the outward unit normal vector. On $\Gamma_{\rm C}$, we denote by $\mathbf{n}^+$ and $\mathbf{n}^-$ the two possible orientations of the unit normal vector. We will see later on that this can be related to the domain decomposition method we use. In our problem, $\Gamma_{\rm N}$ is free to move. For our application, it represents the ground surface. The displacement field can be observed on $\Gamma_{\rm N}$, whereas the pressure $p \in L^2(\Gamma_C)$ exerted on the crack is unknown most of the time (as well as the shape and location of the crack Γ_C). Our works aims at applications in the field of volcanology, where Γ_C is a magma intrusion, i.e. a thin crack filled with magma.

In this paper, we assume that the shape and the location of the crack Γ_C are known. Consider a displacement field $\mathbf{u}_d \in \mathbf{L}^2(\Gamma_{\rm N})$ measured on Γ_N. The identification of a pressure force p such that that the solution $\mathbf{u} = \mathbf{u}(p)$ of the direct problem (2.1)-(2.4) satisfies $\mathbf{u} = \mathbf{u}_d$ on Γ_N is an ill-posed problem. We then consider the following optimization problem

$$\min_{p \in \mathbf{L}^2(\Gamma_{\rm C})} J(p) := \frac{1}{2}\int_{\Gamma_{\rm N}} (\mathbf{u} - \mathbf{u}_d)^\top \mathbf{C}^{-1}(\mathbf{u} - \mathbf{u}_d)\, \mathrm{d}\Gamma_{\rm N} + \frac{\alpha}{2}\|p\|^2_{\mathbf{L}^2(\Gamma_{\rm C})}, \tag{2.5}$$

where $\mathbf{C}$ denotes the covariance operator of the measurements uncertainties (see e.g. [22]), assumed to be positive definite. Finally $\alpha > 0$ is a Tikhonov regularization parameter. The purpose of our work is to extract the pressure from surface measurements such as those provided by radar interferometry [10]. To simplify the presentation, let us introduce the bilinear form c_N defined on $\Gamma_{\rm N}$

$$c_N(\mathbf{w}, \mathbf{w}) = \int_{\Gamma_{\rm N}} \mathbf{w}^\top \mathbf{C}^{-1}\mathbf{w}\, \mathrm{d}\Gamma_{\rm N},$$

so that the cost functional in (2.5) becomes

$$J(p) := \frac{1}{2}c_N(\mathbf{u} - \mathbf{u}_d, \mathbf{u} - \mathbf{u}_d) + \frac{\alpha}{2}\|p\|^2_{\mathbf{L}^2(\Gamma_{\rm C})}.$$

We have the following existence and uniqueness result:

Proposition 2.1. *For any $\alpha > 0$, the problem* (2.5) *admits a unique solution p^* in* $\mathbf{L}^2(\Gamma_C)$.

Proof. The proof is classical: applying the same method as in [14], one easily shows that J is strictly convex and coercive on $\mathbf{L}^2(\Gamma_{\mathrm{C}})$. □

3 Sensitivity analysis

To derive a minimization algorithm for (2.5) we need some information on the derivative of J with respect to p. Let us introduce the linear and bilinear forms

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega \sigma(\mathbf{u}) : \varepsilon(\mathbf{v})\,\mathrm{d}\Omega, \tag{3.1}$$

$$l(\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{C}}} (p\mathbf{n})\cdot\mathbf{v}\,\mathrm{d}\Gamma_{\mathrm{C}}. \tag{3.2}$$

The variational formulation of (2.1)-(2.4) is then

Find $\mathbf{u}\in\mathbf{V}$ such that

$$a(\mathbf{u},\mathbf{v}) = l(\mathbf{v}), \quad \forall \mathbf{v}\in\mathbf{V}. \tag{3.3}$$

¿From (3.3), we deduce that the mapping $p\mapsto\mathbf{u}(p)$ is affine and $\mathbf{u}(p+\rho d)=\mathbf{u}(p)+\rho\mathbf{w}$, where $\mathbf{w}\in\mathbf{V}$ is the solution of the following *sensitivity problem* (with d as crack pressure)

Find $\mathbf{w}\in\mathbf{V}$ such that

$$a(\mathbf{w},\mathbf{v}) = (d\mathbf{n},\mathbf{v})_{\Gamma_{\mathrm{C}}}, \quad \forall \mathbf{v}\in\mathbf{V}, \tag{3.4}$$

where $(\cdot,\cdot)_{\Gamma_{\mathrm{C}}}$ stands for the standard scalar product on $\mathbf{L}^2(\Gamma_{\mathrm{C}})$.

If $\mathbf{u}$ is a solution of (3.3), the *adjoint problem* is given by

Find $\mathbf{z}\in V$

$$a(\mathbf{z},\mathbf{v}) = c_N(\mathbf{u}-\mathbf{u}_d,\mathbf{v}), \quad \forall \mathbf{v}\in\mathbf{V}. \tag{3.5}$$

For a given d in $\mathbf{L}^2(\Gamma_{\mathrm{C}})$, the directional derivative of J is given by

$$\langle\frac{\partial J}{\partial p}(p), d\rangle = c_N(\mathbf{u}-\mathbf{u}_d,\mathbf{w}) + \alpha(p,d), \tag{3.6}$$

where $\mathbf{w}$ is the solution of the sensitivity problem (3.4). Setting $\mathbf{v}=\mathbf{z}$ in(3.4) and $\mathbf{v}=\mathbf{w}$ in (3.5) we get (thanks to the symmetry of a)

$$(d\mathbf{n},\mathbf{z})_{\Gamma_{\mathrm{C}}} = c_N(\mathbf{u}-\mathbf{u}_d,\mathbf{w}). \tag{3.7}$$

Substituting (3.7) into (3.6) we get

$$\langle\frac{\partial J}{\partial p}(p), d\rangle = (d\mathbf{n},\mathbf{z})_{\Gamma_{\mathrm{C}}} + \alpha(p,d)_{\Gamma_{\mathrm{C}}} = (\mathbf{z}\cdot\mathbf{n}+\alpha p, d)_{\Gamma_{\mathrm{C}}}, \quad \forall d\in\mathbf{L}^2(\Gamma_{\mathrm{C}}). \tag{3.8}$$

We deduce from (3.8) that the gradient of J with respect to p in $\mathbf{L}^2(\Gamma_{\mathrm{C}})$ is

$$\nabla J(p) = \mathbf{z}\cdot\mathbf{n} + \alpha p. \tag{3.9}$$

With a search direction d, we compute the optimal stepsize ρ by solving

$$\langle \frac{\partial J}{\partial p}(p+\rho d), d\rangle = c_N(\mathbf{u}+\rho\mathbf{w}-\mathbf{u}_d, \mathbf{w}) + \alpha(p+\rho d, d)_{\Gamma_C} = 0,$$

that is

$$\rho\left[c_N(\mathbf{w},\mathbf{w}) + \alpha(d,d)\right] + c_N(\mathbf{u}-\mathbf{u}_d,\mathbf{w}) + \alpha(p,d)_{\Gamma_C} = 0.$$

The optimal stepsize is therefore

$$\rho^* = -\left[c_N(\mathbf{u}-\mathbf{u}_d,\mathbf{w}) + \alpha(p,d)_{\Gamma_C}\right]\left[c_N(\mathbf{w},\mathbf{w}) + \alpha(d,d)\right]^{-1}.$$

4 Domain decomposition method for the direct problem

As shown in the previous section, each evaluation of the derivative of p needs to solve two direct problems (3.4) and (3.7). To solve the direct problem (2.1)-(2.4) efficiently, we use a domain decomposition method. More precisely, following [6], 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 crack Γ_C, as shown in Figure 1. ¿From a theoretical point of view, we just need to assume that the crack is represented by an injective curve or a surface to ensure the existence of such an artificial extension.

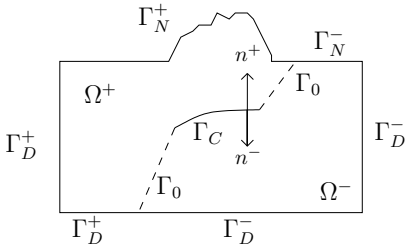

Figure 1: Splitting of the cracked domain

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 $\mathbf{n}^+$ and $\mathbf{n}^-$. The global unknown $\mathbf{u}$ is split into $\mathbf{u}^+ = \mathbf{u}_{|\Omega^+}$ and $\mathbf{u}^- = \mathbf{u}_{|\Omega^-}$.

With the notations above, Problem (2.1)-(2.4) can be reformulated as follows.

Find $\mathbf{u}^\pm \in \mathbf{H}^1(\Omega)$ such that :

$$\begin{aligned}
-\text{div } \sigma(\mathbf{u}^\pm) &= \mathbf{f}^\pm \text{ in } \Omega^\pm, && (4.1)\\
\mathbf{u}^\pm &= 0 \text{ on } \Gamma_D^\pm, && (4.2)\\
(\sigma(\mathbf{u})\mathbf{n})^\pm &= 0 \text{ on } \Gamma_N^\pm, && (4.3)\\
(\sigma(\mathbf{u})\mathbf{n})^\pm &= p\mathbf{n}^\pm \text{ on } \Gamma_C, && (4.4)\\
[\![\mathbf{u}]\!] &= 0 \text{ on } \Gamma_0, && (4.5)\\
[\![\sigma(\mathbf{u})]\!]\mathbf{n}^\pm &= 0 \text{ on } \Gamma_0, && (4.6)
\end{aligned}$$

where $[\![\cdot]\!]$ denote jumps of a vector field accross Γ_0. Equations (4.5)-(4.6) enforce the continuity of displacements and stress across Γ_0 to ensure that $\mathbf{u} = (\mathbf{u}^-, \mathbf{u}^+)$ solves the original problem (2.1)-(2.4).

For the weak formulation of the problem (4.1)-(4.6) we introduce the following Hilbert spaces

$$\begin{aligned} \mathbf{V}^+ &= \{\mathbf{v} \in \mathbf{H}^1(\Omega^+) \mid \mathbf{v} = 0 \text{ on } \Gamma_{\mathrm{D}} \cap \partial\Omega^+\}, \\ \mathbf{V}^- &= \{\mathbf{v} \in \mathbf{H}^1(\Omega^-) \mid \mathbf{v} = 0 \text{ on } \Gamma_{\mathrm{D}} \cap \partial\Omega^-\}, \\ \mathbf{W} &= (\mathbf{L}^2(\Gamma_0))^2. \end{aligned}$$

The linear and bilinear forms (3.1)-(3.2) become in the subdomains

$$\begin{aligned} a^{\pm}(\mathbf{u}^{\pm}, \mathbf{v}^{\pm}) &= \int_{\Omega^{\pm}} \sigma(\mathbf{u}^{\pm}) : \varepsilon(\mathbf{v}^{\pm}) \, \mathrm{d}\Omega^{\pm}, \\ l^{\pm}(\mathbf{v}^{\pm}) &= \int_{\Omega^{\pm}} \mathbf{f} \cdot \mathbf{v}^{\pm} \, \mathrm{d}\Omega^{\pm} + \int_{\Gamma_{\mathrm{C}}} (p\mathbf{n})^{\pm} \cdot \mathbf{v}^{\pm} \, \mathrm{d}\Gamma_{\mathrm{C}} \end{aligned}$$

We also define the bilinear form

$$b(\boldsymbol{\lambda}, \mathbf{v}) = (\boldsymbol{\lambda}, \mathbf{v})_{\Gamma_0} = \int_{\Gamma_0} \boldsymbol{\lambda} \cdot \mathbf{v} \, \mathrm{d}\Gamma_0.$$

Since the bilinear form a is symmetric and coercive, (4.1)-(4.6) can be reformulated as a (convex) constrained optimization problem with (4.5) as the linear constraint. The terms involving the Lagrange multiplier λ ensure the continuity condition (4.6). The Lagrangian functional associated with (4.1)-(4.6) is

$$\mathcal{L}(\mathbf{u}^+, \mathbf{u}^-, \boldsymbol{\lambda}) = \frac{1}{2} a^+(\mathbf{u}^+, \mathbf{u}^+) + \frac{1}{2} a^-(\mathbf{u}^-, \mathbf{u}^-) - l^+(\mathbf{u}^+) - l^-(\mathbf{u}^-) + b(\boldsymbol{\lambda}, \mathbf{u}^+) - b(\boldsymbol{\lambda}, \mathbf{u}^-) \quad (4.7)$$

and the variational formulation of (4.1)-(4.6) is the saddle-point of $\mathcal{L}$, that is, the solution of the following problem:

Find $(\mathbf{u}^+, \mathbf{u}^-, \boldsymbol{\lambda}) \in \mathbf{V}^+ \times \mathbf{V}^- \times \mathbf{W}$ such that

$$\begin{aligned} a^+(\mathbf{u}^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}, \mathbf{v}^+) &= l^+(\mathbf{v}^+), \quad \forall \mathbf{v}^+ \in \mathbf{V}^+, & (4.8) \\ a^-(\mathbf{u}^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}, \mathbf{v}^-) &= l^-(\mathbf{v}^-), \quad \forall \mathbf{v}^- \in \mathbf{V}^-, & (4.9) \\ b(\boldsymbol{\mu}, \mathbf{u}^+) - b(\boldsymbol{\mu}, \mathbf{u}^-) &= 0, \quad \forall \boldsymbol{\mu} \in \mathbf{W}. & (4.10) \end{aligned}$$

The Uzawa conjugate gradient domain decomposition method to solve (4.8)-(4.10), proposed by [6], is presented in Algorithm 1. Note that the matrices involved in Algorithm 1 do not change during the iterative process. Therefore a factorization can be performed once and for all in the initialization step to save computational time. We refer to [6] for a detailed study of this method.

Algorithm 1 Uzawa conjugate gradient algorithm for the direct problem

$k = 0$. **Initialization** $\epsilon > 0$ and $\boldsymbol{\lambda}_0$ are given. Compute $\mathbf{u}_0^{\pm}$ such that

$$a^+(\mathbf{u}_0^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_0, \mathbf{v}^+) = l^+(\mathbf{v}^+) \quad \forall \mathbf{v}^+ \in \mathbf{V}^+, \tag{4.11}$$

$$a^-(\mathbf{u}_0^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_0, \mathbf{v}^-) = l^-(\mathbf{v}^-) \quad \forall \mathbf{v}^- \in \mathbf{V}^-, \tag{4.12}$$

$\boldsymbol{\gamma}_0 = [\![\mathbf{u}_0]\!]$ (gradient) and $\boldsymbol{\mu}_0 = \boldsymbol{\gamma}_0$ (search direction).

$k \geq 0$. While $(\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k)_{\Gamma_0} > \epsilon(\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_0)_{\Gamma_0}$

- Sensitivity: Compute $\boldsymbol{\omega}_k^{\pm}$ such that

$$a^+(\boldsymbol{\omega}_k^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_k, \mathbf{v}^+) = l^+(\mathbf{v}^+) \quad \forall \mathbf{v}^+ \in \mathbf{V}^+, \tag{4.13}$$

$$a^-(\boldsymbol{\omega}_k^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_k, \mathbf{v}^-) = l^-(\mathbf{v}^-) \quad \forall \mathbf{v}^- \in \mathbf{V}^-, \tag{4.14}$$

- Stepsize
$$\rho_k = -(\boldsymbol{\mu}_k, [\![\mathbf{u}_k]\!])_{\Gamma_0} \,/\, (\boldsymbol{\mu}_k, [\![\boldsymbol{\omega}_k]\!])_{\Gamma_0}$$
- Update: $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho_k \boldsymbol{\mu}_k$, $\mathbf{u}_{k+1}^{\pm} = \mathbf{u}_k^{\pm} + \rho_k \boldsymbol{\omega}^{\pm}$
- New gradient: $\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \rho_k [\![\boldsymbol{\omega}_k]\!]$
- New direction: $\boldsymbol{\mu}_{k+1} = \boldsymbol{\gamma}_{k+1} + \beta_k \boldsymbol{\mu}_k$, with $\beta_k = (\boldsymbol{\gamma}_{k+1}, \boldsymbol{\gamma}_{k+1})_{\Gamma_0} \,/\, (\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k)_{\Gamma_0}$ (Fletcher-Reeves version).

5 Practical implementation aspects for the direct problem

5.1 Discretization of the fictitious domain method

Let us first describe the discrete formulation of Problem (4.8)–(4.10). We use a method of XFEM type (eXtended Finite Element Method). For further details, we refer the reader to [2, 6, 13, 15]. In this approach, the mesh of the extended crack Γ_0 (and therefore the degrees of freedom for the Lagrange multiplier and the pressure p) are independent of the original mesh of $\Omega^{\pm}$ and could be chosen independently. Hence we introduce two discrete finite element spaces $\widetilde{\mathbf{V}}_h \subset \mathbf{H}^1(\Omega)$ and $\widetilde{\mathbf{W}}_h \subset \mathbf{L}^2(\Omega)$ and define the discretization spaces

$$\mathbf{V}_h^+ = \widetilde{\mathbf{V}}_h|_{\Omega^+}, \qquad \mathbf{V}_h^- = \widetilde{\mathbf{V}}_h|_{\Omega^-}, \qquad \mathbf{W}_h = \widetilde{\mathbf{W}}_h|_{\Gamma_0}.$$

A finite element discrete formulation of (4.8)–(4.10) consists then in finding $(\mathbf{u}_h^+, \mathbf{u}_h^-, \boldsymbol{\lambda}_h) \in \mathbf{V}_h^+ \times \mathbf{V}_h^- \times \mathbf{W}_h$ such that :

$$a^+(\mathbf{u}_h^+, \mathbf{v}_h^+) + b(\boldsymbol{\lambda}_h, \mathbf{v}_h^+) = l(\mathbf{v}_h^+) \quad \forall \mathbf{v}_h^+ \in \mathbf{V}_h^+, \tag{5.1}$$

$$a^-(\mathbf{u}_h^-, \mathbf{v}_h^-) - b(\boldsymbol{\lambda}_h, \mathbf{v}_h^-) = l(\mathbf{v}_h^-) \quad \forall \mathbf{v}_h^- \in \mathbf{V}_h^-, \tag{5.2}$$

$$b(\boldsymbol{\mu}_h, \mathbf{u}_h^+) - b(\boldsymbol{\mu}_h, \mathbf{u}_h^-) = 0 \quad \forall \boldsymbol{\mu}_h \in \mathbf{W}_h, \tag{5.3}$$

where

$$\begin{aligned} a^{\pm}(\mathbf{u}_h^{\pm}, \mathbf{v}_h^{\pm}) &= \int_{\Omega^{\pm}} \varepsilon(\mathbf{u}_h^{\pm}) : \sigma(\mathbf{v}_h^{\pm}) \, \mathrm{d}\Omega^{\pm}, \\ l(\mathbf{v}_h^{\pm}) &= \int_{\Omega^{\pm}} \mathbf{f}^{\pm} \cdot \mathbf{v}_h^{\pm} \, \mathrm{d}\Omega^{\pm} + \int_{\Gamma_{\mathrm{C}}} \mathbf{v}_h^{\pm} \cdot p\mathbf{n}^{\pm} \, \mathrm{d}\Gamma_{\mathrm{C}}, \\ b(\boldsymbol{\mu}_h, \mathbf{u}_h^{\pm}) &= \int_{\Gamma_0} \boldsymbol{\mu}_h \cdot \mathbf{u}_h^{\pm} \, \mathrm{d}\Gamma_0. \end{aligned}$$

Let us finally add that the integrations on the fictitious domains $\Omega^{\pm}$ and Γ_{F} are made by multiplying the basis functions by Heaviside functions

$$H^{\pm}(x) = \begin{cases} 1 & \text{if } x \in \Omega^{\pm}, \\ 0 & \text{otherwise.} \end{cases}$$

as introduced in [12]. Doing this, technically differs from the classical XFEM technique, where the basis functions across the Γ_{C} are enriched by singular functions as done e.g. in [2, 15]. However, in [3], the two methods are shown to be mathematically equivalent. More precisely, the decomposition of the basis functions are simply a linear combination of the original XFEM basis.

The existence and uniqueness of discrete Problem (5.1)–(5.3) has been established in [6], by using standard techniques. The proof is made under the following assumption: let $\overline{\boldsymbol{\mu}}_h \in \mathbf{W}_h$ be such that for all $\mathbf{v}^{\pm} \in \mathbf{V}^{\pm}$, if $b(\overline{\boldsymbol{\mu}}_h, \mathbf{v}^{\pm}) = 0$ then $\overline{\boldsymbol{\mu}}_h = 0$. Notice that this assumption is weaker that the classical *inf–sup* condition. See [6] for details.

The implementation of this method uses the finite element library GetFEM (see [20]). The library provides tools to handle various types of PDEs, classical powerful solvers (conjugate gradient, SuperLU, MUMPS), and routines for mesh management, definition of boundaries via level set functions, which make it quite versatile and powerful.

5.2 Level set implementation for the crack

Since we use the GetFem library [20], the simplest way to take the crack into account is to define two level set functions. If a regular parametrization of the crack is available, we define two explicit functions as follows: a primary function $ls_1(\mathbf{x}) = 0$ splits Ω into to Ω^+ and Ω^-, thus describing $\Gamma_{\mathrm{F}} = \Gamma_{\mathrm{C}} \cup \Gamma_0$. Then, a secondary function $ls_2(\mathbf{x}) < 0$ delimits the crack Γ_{C} itself (see sections 7.1 and 7.2 for examples).

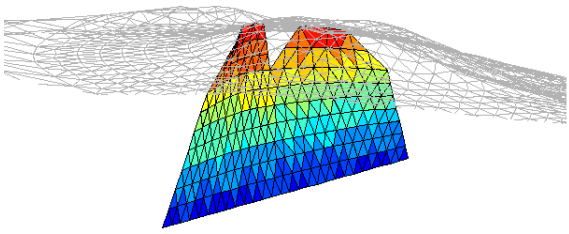

Figure 2: Triangular surface mesh representing the crack

In some practical applications (i.e. in 3D cases), the crack is not explicitly defined. It is approximated by a triangle mesh which is independent of the mesh of the domain Ω. Figure 2 depicts such a case. In this example, we used the crack representation introduced in [10].

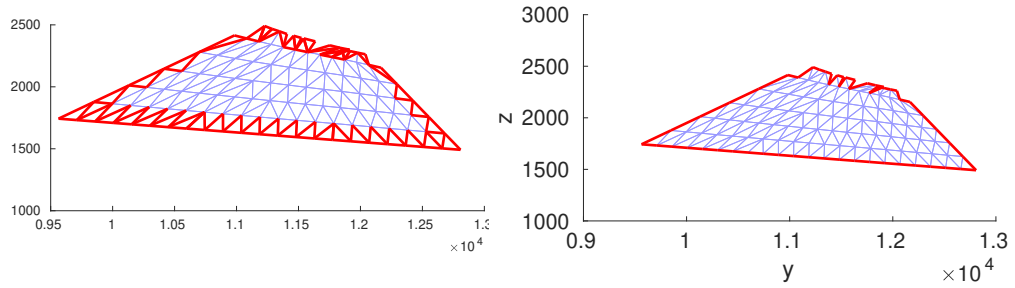

Figure 3: Crack triangle mesh with triangles located in the crack front are shown in red (left). Front segments of the crack are shown in red (right)

In such a situation, the level set functions ls_1 and ls_2 need to be defined on the main mesh. In [6], the level set function ls_1 is built to take values in $\{-1, 0, 1\}$. In this work, this construction has been improved by the use of signed distances, allowing to work with more complex crack shapes. More precisely, for every node $\mathbf{x}$ of the mesh, let $(ABC)_{\mathbf{x}}$ be the crack triangle nearest to $\mathbf{x}$. Let $d_{\mathbf{x}}$ be the euclidean distance between $\mathbf{x}$ and $(ABC)_{\mathbf{x}}$. This is easily computed via an orthogonal projection of $\mathbf{x}$ onto the plane generated by $(ABC)_{\mathbf{x}}$. Choosing an orientation for the normal on the crack, it remains to define $ls_1(\mathbf{x}) = \pm d_{\mathbf{x}}$. The real technical point here lies in the extension of Γ_C to $\Gamma_F = \Gamma_C \cup \Gamma_0$. This extension is built from the planes generated by boundary dike triangles (see Figures 3 (left) and 4).

Let us now define the secondary level set function ls_2: let Γ_P be the front segments of the crack Γ_C (see Figure 3 (right)). For any point $\mathbf{x}$ in Ω, , consider its projected point $\mathbf{p_x}$ on the extended crack. We then define $ls_2(\mathbf{x})$ as the signed distance between $\mathbf{p_x}$ and Γ_P. It is be computed as follows: for each segment $[\mathbf{s_1}, \mathbf{s_2}]$ of Γ_P, where $\mathbf{s_1}$ and $\mathbf{s_2}$ are the end points of the segment, the points of the line through $\mathbf{s_1}$ and $\mathbf{s_2}$ are given by $s(t) = \mathbf{s_1} + t(\mathbf{s_2} - \mathbf{s_1})$, $t \in \mathbb{R}$. The points in the segment are obtained for $t \in [0, 1]$. Minimizing the function $\|s(t) - \mathbf{p_x}\|$ over $t \in \mathbb{R}$ gives

$$\widetilde{t} = \frac{\langle \mathbf{p} - \mathbf{s_1}, \mathbf{s_2} - \mathbf{s_1} \rangle}{\|\mathbf{s_2} - \mathbf{s_1}\|^2}.$$

We project back to the segment ($t \in [0, 1]$) by setting

$$t^* = \min(\max(\widetilde{t}, 0), 1).$$

The distance between $\mathbf{p_x}$ and the segment $[\mathbf{s_1}, \mathbf{s_2}]$ is then given by $\|s(t^*) - \mathbf{p_x}\|$. The level set function $ls_2(\mathbf{x})$ is then defined by taking the minimum of this quantity over the segments. The sign is computed by considering the unit outward normal vector to Γ_P.

It is noteworthy to mention another similar approach of the problem, called explicit–implicit fracture surface representation. It is proposed in [9, 19]. In this method, the crack is extended along the line segments generated by the a family of vectors $\mathbf{t_m}$ at all nodes on the crack front. These vectors are interpreted by the authors as tangential extensions of the crack surface, normal to the crack front. At each node of the crack front, the vector $\mathbf{t_m}$ is defined as the cross product between an averaged normal vector $\mathbf{n_m}$ at the node and an averaged tangent vector $\mathbf{q_m}$ in the direction of each segment of the crack front (see Figure 5 in [9]). The authors then combine the explicit crack descriptions with the implicit description via the definition of three level set functions.

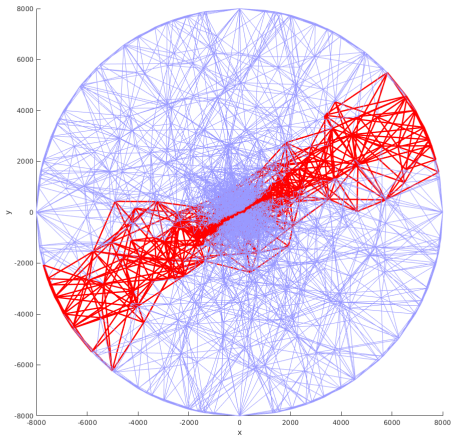

Figure 4: Elements of a volcano mesh, cut by a crack and its extension (in red).

5.3 Stabilized variational formulation

Without a stabilization technique, the optimality *inf–sup* condition is not always satisfied for the discrete variational formulation (5.1)-(5.3), yielding convergence problems for the discretized direct problem. The Barbosa–Hughes stabilization consists in modifying the Lagrangian (4.7) as follows:

$$\mathcal{L}_\gamma(\mathbf{u}^+,\mathbf{u}^-,\boldsymbol{\lambda}) = \mathcal{L}(\mathbf{u}^+,\mathbf{u}^-,\boldsymbol{\lambda}) - \frac{\gamma}{2}\int_{\Gamma_0}|\boldsymbol{\lambda}+\sigma(\mathbf{u}^+)\mathbf{n}^+|^2\,\mathrm{d}\Gamma_0 - \frac{\gamma}{2}\int_{\Gamma_0}|\boldsymbol{\lambda}-\sigma(\mathbf{u}^-)\mathbf{n}^-|^2\,\mathrm{d}\Gamma_0, \tag{5.4}$$

with $\gamma > 0$. We refer to [4, 5] for a detailed justification of the added stabilization terms. This technique ensures an unconditional optimal convergence for the multiplier and then an optimality *inf–sup* condition (see [6] and the references therein).

The saddle point of this stabilized Lagrangian functional (5.4) satisfies the following discrete variational formulation:

Find $(\mathbf{u}_h^+,\mathbf{u}_h^-,\boldsymbol{\lambda}_h) \in \mathbf{V}_h^+ \times \mathbf{V}_h^- \times \mathbf{W}_h$ such that :

$$\begin{aligned}
a_\gamma^+(\mathbf{u}_h^+,\mathbf{v}_h^+) + b_\gamma^+(\boldsymbol{\lambda}_h,\mathbf{v}_h^+) &= l(\mathbf{v}_h^+), \quad \forall \mathbf{v}_h^+ \in \mathbf{V}_h^+, &(5.5)\\
a_\gamma^-(\mathbf{u}_h^-,\mathbf{v}_h^-) - b_\gamma^-(\boldsymbol{\lambda}_h,\mathbf{v}_h^-) &= l(\mathbf{v}_h^-), \quad \forall \mathbf{v}_h^- \in \mathbf{V}_h^-, &(5.6)\\
b_\gamma^+(\boldsymbol{\mu}_h,\mathbf{u}_h^+) - b_\gamma^-(\boldsymbol{\mu}_h,\mathbf{u}_h^-) - c_\gamma(\boldsymbol{\lambda}_h,\boldsymbol{\mu}_h,) &= 0, \quad \forall \boldsymbol{\mu}_h \in \mathbf{W}_h, &(5.7)
\end{aligned}$$

where we have set

$$\begin{aligned}
a_\gamma^\pm(\mathbf{u}_h^+,\mathbf{v}_h^+) &= a^\pm(\mathbf{u}_h^\pm,\mathbf{v}_h^\pm) - \gamma(\sigma(\mathbf{u}_h^\pm)\mathbf{n}^\pm,\sigma(\mathbf{v}_h^\pm)\mathbf{n}^\pm)_{\Gamma_0},\\
b_\gamma^\pm(\boldsymbol{\lambda}_h,\mathbf{v}_h^\pm) &= b(\boldsymbol{\lambda}_h,\mathbf{v}_h^\pm) - \gamma(\boldsymbol{\lambda}_h,\sigma(\mathbf{v}_h^\pm)\mathbf{n}^\pm)_{\Gamma_0},\\
c_\gamma(\boldsymbol{\lambda}_h,\boldsymbol{\mu}_h,) &= 2\gamma(\boldsymbol{\lambda}_h,\boldsymbol{\mu}_h,)_{\Gamma_0}.
\end{aligned}$$

The stabilization has an impact on the algorithms described in the previous sections.

In Algorithm 1, which is used as inner solver for the first domain decomposition algorithm, the bilinear forms $a^\pm$ and b must be replaced by $a_\gamma^\pm$ and $b_\gamma^\pm$ in systems (4.11)-(4.12)

and (4.13)-(4.14). At each iteration, k the gradient $\boldsymbol{\gamma}_k$ and the stepsize ρ_k become (dropping discretization subscript h)

$$\begin{aligned}\boldsymbol{\gamma}_k &= [\![\mathbf{u}_k]\!] - \gamma(\boldsymbol{\lambda}_k + \sigma(\mathbf{u}_k^+)\mathbf{n}^+) - \gamma(\boldsymbol{\lambda}_k - \sigma(\mathbf{u}_k^-)\mathbf{n}^-),\\ \rho_k &= -(\boldsymbol{\mu}_k, \boldsymbol{\gamma}_k)_{\Gamma_0} \left(\boldsymbol{\mu}_k, [\![\mathbf{w}_k]\!] - \gamma(\boldsymbol{\mu}_k + \sigma(\mathbf{w}_k^+)\mathbf{n}^+) - \gamma(\boldsymbol{\mu}_k - \sigma(\mathbf{w}_k^-)\mathbf{n}^-\right)_{\Gamma_0}^{-1}.\end{aligned}$$

In Algorithm 2, the bilinear forms $a^{\pm}$ and b must be replaced by $a_\gamma^{\pm}$ and $b_\gamma^{\pm}$ in (6.2)-(6.3), (6.5)-(6.6), (6.8)-(6.9) and (6.11)-(6.12). The interface condition equations (6.4), (6.7), (6.10) and (6.13) must be replaced by (5.7).

The stabilization of the direct problem is thoroughly studied in [6]. It allows to perform a convergence analysis of the numerical schemes. The choice of the parameter γ is rather empirical, and also described in this same article. Thus, we won't go into more details on this question.

6 Minimization algorithms for the optimal control problem

In this section we present two minimization algorithms that were implemented to solve problem (2.5). These two methods are rather natural in this framework, and were adapted to the context of domain decomposition. We will first describe the conjugate algorithm, then a quasi-Newton method.

6.1 Conjugate gradient

After splitting the computational domain Ω, the cost function J in optimization problem (2.5) becomes

$$J(p) = \frac{1}{2}c_N(\mathbf{u}^+ - \mathbf{u}_d^+, \mathbf{u}^+ - \mathbf{u}_d^+) + \frac{1}{2}c_N(\mathbf{u}^- - \mathbf{u}_d^-, \mathbf{u}^- - \mathbf{u}_d^-) + \frac{\alpha}{2}\|p\|^2_{\mathbf{L}^2(\Gamma_C)}. \qquad (6.1)$$

Our first domain decomposition algorithm to solve the inverse problem (2.5) is described in Algorithm 2.

Algorithm 2 First algorithm for (2.5): conjugate gradient with domain decomposition

$k = 0$. p_0 is given, Compute $(\mathbf{u}_0^{\pm}, \boldsymbol{\lambda}_0^e)$ such that

$$a^+(\mathbf{u}_0^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_0, \mathbf{v}^+) = l(\mathbf{v}^+), \quad \forall \mathbf{v}^+ \in \mathbf{V}_h^+, \tag{6.2}$$

$$a^-(\mathbf{u}_0^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_0, \mathbf{v}^-) = l(\mathbf{v}^-), \quad \forall \mathbf{v}^- \in \mathbf{V}_h^-, \tag{6.3}$$

$$b(\boldsymbol{\mu}, \mathbf{u}_0^+) - b(\boldsymbol{\mu}, \mathbf{u}_0^-) = 0, \quad \forall \boldsymbol{\mu} \in \mathbf{W}_h. \tag{6.4}$$

Compute $(\mathbf{z}_0^{\pm}, \boldsymbol{\lambda}_0^a)$ such that

$$a^+(\mathbf{z}_0^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_0^a, \mathbf{v}^+) = c_N(\mathbf{u}_0^+ - \mathbf{u}_d^+, \mathbf{v}^+), \quad \forall \mathbf{v}^+ \in \mathbf{V}_h^+, \tag{6.5}$$

$$a^-(\mathbf{z}_0^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_0^a, \mathbf{v}^-) = c_N(\mathbf{u}_0^- - \mathbf{u}_d^-, \mathbf{v}^-), \quad \forall \mathbf{v}^- \in \mathbf{V}_h^-, \tag{6.6}$$

$$b(\boldsymbol{\mu}, \mathbf{z}_0^+) - b(\boldsymbol{\mu}, \mathbf{z}_0^-) = 0, \quad \forall \boldsymbol{\mu} \in \mathbf{W}_h. \tag{6.7}$$

$g_0 = \mathbf{z}_0^{\pm} \cdot \mathbf{n}^{\pm} + \alpha p_0$ (gradient) and $d_0 = -g_0$ (search direction)

$k \geq 0$. **Sensitivity:** Compute $(\mathbf{w}_k, \boldsymbol{\lambda}_k^s)$ such that

$$a^+(\mathbf{w}_k^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_k^s, \mathbf{v}^+) = (d_k \mathbf{n}^+, \mathbf{v}^+)_{\Gamma_C}, \quad \forall \mathbf{v}^+ \in \mathbf{V}_h^+, \tag{6.8}$$

$$a^-(\mathbf{w}_k^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_k^s, \mathbf{v}^-) = (d_k \mathbf{n}^-, \mathbf{v}^-)_{\Gamma_C}, \quad \forall \mathbf{v}^- \in \mathbf{V}_h^-, \tag{6.9}$$

$$b(\boldsymbol{\mu}, \mathbf{w}_k^+) - b(\boldsymbol{\mu}, \mathbf{w}_k^-) = 0, \quad \forall \boldsymbol{\mu} \in \mathbf{W}_h. \tag{6.10}$$

Stepsize:

$$\rho_k = -\left(c_N(\mathbf{u}_k^{\pm} - \mathbf{u}_d^{\pm}, \mathbf{w}_k^{\pm}) + \alpha(p_k, d_k)\right)\left(c_N(\mathbf{w}_k^{\pm}, \mathbf{w}_k^{\pm}) + \alpha(d_k, d_k)_{Gc}\right)^{-1}$$

Update $p_{k+1} = p_k + \rho_k d_k$, $\mathbf{u}_{k+1}^{\pm} = \mathbf{u}_k^{\pm} + \rho_k \mathbf{w}_k^{\pm}$

Adjoint: Compute $(\mathbf{z}_k^{\pm}, \boldsymbol{\lambda}_k^a)$ such that

$$a^+(\mathbf{z}_k^+, \mathbf{v}^+) + b(\boldsymbol{\lambda}_k^a, \mathbf{v}^+) = c_N(\mathbf{u}_k^+ - \mathbf{u}_d^+, \mathbf{v}^+), \quad \forall \mathbf{v}^+ \in \mathbf{V}_h^+, \tag{6.11}$$

$$a^-(\mathbf{z}_k^-, \mathbf{v}^-) - b(\boldsymbol{\lambda}_k^a, \mathbf{v}^-) = c_N(\mathbf{u}_k^- - \mathbf{u}_d^-, \mathbf{v}^-), \quad \forall \mathbf{v}^- \in \mathbf{V}_h^-, \tag{6.12}$$

$$b(\boldsymbol{\mu}, \mathbf{z}_k^+) - b(\boldsymbol{\mu}, \mathbf{z}_k^-) = 0, \quad \forall \boldsymbol{\mu} \in \mathbf{W}_h. \tag{6.13}$$

New gradient: $g_{k+1} = \mathbf{z}_{k+1}^{\pm} \cdot \mathbf{n}^{\pm} + \alpha p_{k+1}$

New direction: $d_{k+1} = -g_{k+1} + \beta_k d_k$, with $\beta_k = (g_{k+1}, g_{k+1})_{\Gamma_C}(g_k, g_k)_{\Gamma_C}^{-1}$

We iterate until the gradient of J becomes "sufficiently small", that is

$$\| g_k \|_{\Gamma_C} < \epsilon \| g_0 \|_{\Gamma_C} .$$

Algorithm 2 uses Algorithm 1 as a inner solver for solving (6.2)-(6.4), (6.5)-(6.7), (6.8)-(6.10) and (6.11)-(6.13).

Notice also here, that this algorithm can be written in a fully continuous way, making it theoretically insensitive to the discretization parameter. In Algorithm 2, we give the discrete form, to make things consistent with the quasi-Newton method described in the next subsection.

6.2 BFGS quasi-Newton method

The problem under consideration being quadratic, the conjugate gradient should be the natural choice. However, in some practical cases, the fracture can be quite small with respect to the global domain, leading to a small number of unknowns for the pressure p. In such a case, the cost function (2.5) can be profitably minimized via a BFGS (Broyden, Fletcher, Goldfarb and Shanno) quasi-Newton algorithm. The update formulas for quadratic problems are well documented (see e.g. [16, 17] for more details). Moreover, as for the conjugate gradient technique presented before, the underlying quadratic form allows to compute explicitly the optimal step in the search direction at any iteration. Therefore, no line search routine is needed to implement this algorithm. The main differ-

Algorithm 3 Second algorithm for (2.5): BFGS method with domain decomposition

$k = 0$. p_0 is given, Compute $(\mathbf{u}_0^{\pm}, \boldsymbol{\lambda}_0^e)$ solving(6.2),(6.3) and(6.4).

Compute $(\mathbf{z}_0^{\pm}, \boldsymbol{\lambda}_0^a)$ solving(6.5), (6.6) and (6.7).

Set H^0 to the identity matrix.

$g_0 = \mathbf{z}_0^{\pm} \cdot \mathbf{n}^{\pm} + \alpha p_0$ (gradient) and $d_0 = -g_0$ (initial search direction)

$k \geq 0$. **Sensitivity:** Compute $(\mathbf{w}_k, \boldsymbol{\lambda}_k^s)$ solving (6.8), (6.9)and (6.10).

Stepsize:

$$\rho_k = -\left(c_N(\mathbf{u}_k^{\pm} - \mathbf{u}_d^{\pm}, \mathbf{w}_k^{\pm}) + \alpha(p_k, d_k)\right)\left(c_N(\mathbf{w}_k^{\pm}, \mathbf{w}_k^{\pm}) + \alpha(d_k, d_k)_{Gc}\right)^{-1}$$

Update: $p_{k+1} = p_k + \rho_k d_k$, $\mathbf{u}_{k+1}^{\pm} = \mathbf{u}_k^{\pm} + \rho_k \mathbf{w}_k^{\pm}$

Adjoint: Compute $(\mathbf{z}_k^{\pm}, \boldsymbol{\lambda}_k^a)$ solving (6.11), (6.12) and (6.13).

New gradient: $g_{k+1} = \mathbf{z}_{k+1}^{\pm} \cdot \mathbf{n}^{\pm} + \alpha p_{k+1}$

Update of the approximation of $(\nabla^2 J)^{-1}$:

$$H^{k+1} = (I - \theta^k s^k y^{k\mathsf{T}}) H^k (I - \theta^k y^k s^{k\mathsf{T}}) + \theta^k s^k s^{k\mathsf{T}}, \tag{6.14}$$

with

$$s^k = p^{k+1} - p^k, \quad y^k = g^{k+1} - g^k, \quad \theta^k = \frac{1}{y^{k\mathsf{T}} s^k}.$$

New direction: $d^{k+1} = -H^{k+1} g^{k+1}$.

ence with the conjugate gradient algorithm lies in the computation of the search direction d_k, which involves the construction of a sequence (H^k) of approximations of the (inverse) Hessian Matrix of the cost function J. The algorithm 3 describes the method.

In formula (6.14), the functions p^k, g^k are identified by their coordinate vectors on the approximation basis used for W_h. Finally, notice that we used a limited storage algorithm, called L–BFGS (see [16]) for the update of the matrices H^k.

7 Numerical results

7.1 2D test

In this example, we consider the open set $\Omega =]-1,1[\times]-1,1[$. The Lamé parameters are $\lambda = 1$ and $\mu = 1$. We set $\Gamma_{\mathrm{D}} =]0,1[\times\{1\}$, and consequently $\Gamma_{\mathrm{N}} = \partial\Omega \setminus \Gamma_{\mathrm{D}}$.

The crack Γ_{C} is the following line:

$$\Gamma_{\mathrm{C}} = \{(x,y) \in \Omega\,;\, x = 0.05,\ -0.2 < y < 0.2\}\,,$$

hence the level set functions ls_1 and ls_2 are

$$ls_1(x,y) = x - 0.05, \qquad ls_2(x,y) = |y| - 0.2.$$

Therefore, the extended crack is given by

$$\Gamma_{\mathrm{C}} = \{(x,y) \in \Omega\,;\, x = 0.05\}\,.$$

We suppose that there are no body forces i.e. $\mathbf{f} = 0$. The domain is uniformly meshed, and in the following test the meshing parameters with respect to x and y are $h_x = h_y = 0.1$. The crack being small, there are only 8 unknowns for the pressure force p. In this example, a synthetic solution is built from a known force p (depicted in Figure 7), and its value on the top of the domain u_d is used in the cost function. The regularization parameter is intently chosen very small ($\alpha = 10^{-13}$). Figure 5 presents the convergence of the gradient of the cost function. This convergence is achieved at the expected rate for all the methods. Then, Figure 6 presents the decay of the ground error (also called misfit), that is the L^∞ norm of $\mathbf{u} - \mathbf{u}_d$. Finally, Figure 7 shows the pressure p at the beginning and the end of the iterative process. As one can see, both optimization methods are efficient in terms of gradient and misfit decay. The BFGS method is a bit faster. Notice that the numerical values for the errors are quite small: indeed in linear elasticity with realistic parameters, we deal with small displacements which requires extra precision on the errors.

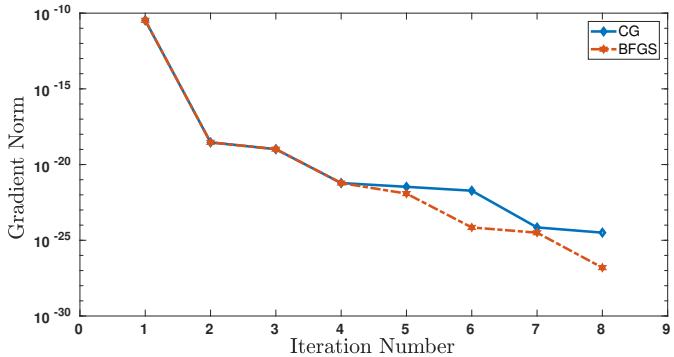

Figure 5: L^2-norm of the gradient of the cost function for the 2D test.

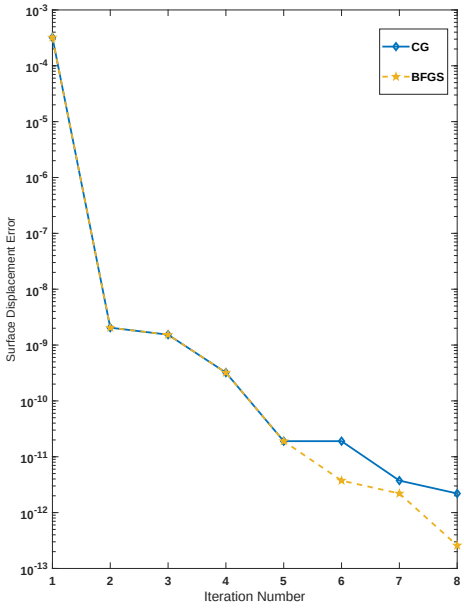

Figure 6: Evolution of the ground misfit in L^∞-norm for the 2D test.

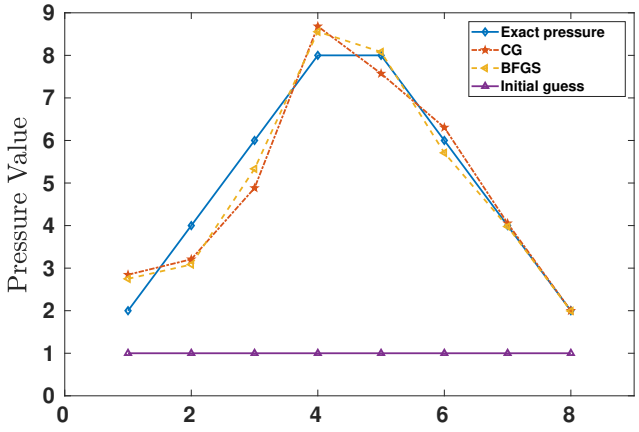

Figure 7: Computed and exact crack pressures for the 2D test.

7.2 3D test

Here we consider $\Omega =]-1,1[^3$, $\lambda = \mu = 1$ and $\Gamma_D =]0,1[\times]0,1[\times\{1\}$. The crack is given by

$$\Gamma_C = \left\{(x,y,z) \in \Omega\,;\, x = 0.05,\, y^2 + z^2 - 0.2^2 < 0\right\},$$

that is

$$ls_1(x,y) = x - 0.05, \qquad ls_2(x,y) = y^2 + z^2 - 0.2^2.$$

The simulations were conducted in the same way as for the 2D case. The mesh is uniform in all the directions, with a mesh parameter $h = 0.2$. There are 49 unknowns for the pressure p. For test purposes, the number of iterations has been set equal to the number of unknowns, but after 8 iterations, the convergence is achieved.

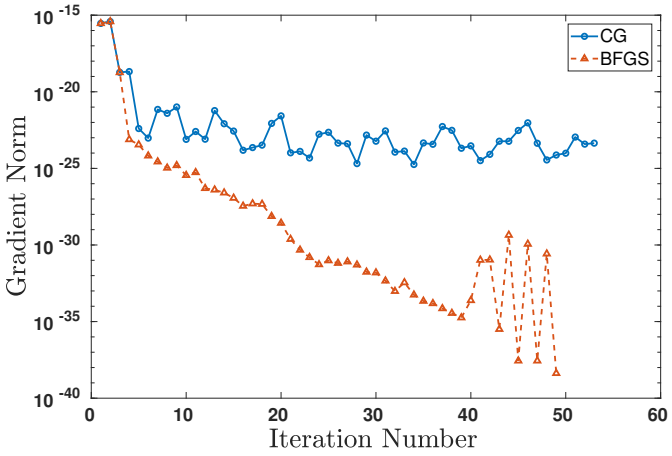

Figure 8: L^2-norm of the gradient of the cost function for the 3D test.

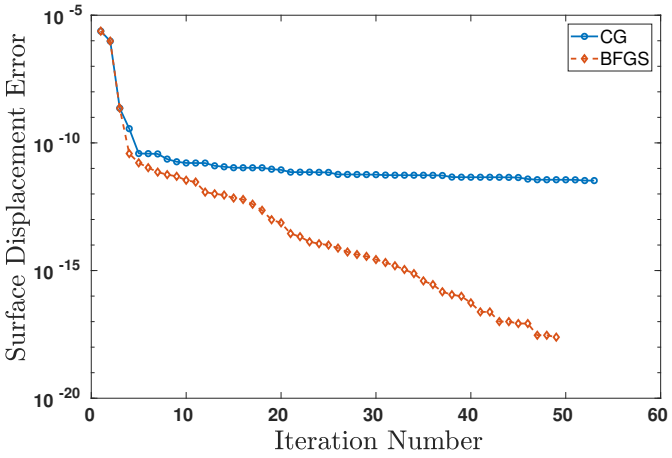

Figure 9: Evolution of the ground misfit (in L^∞-norm) for the 3D test.

7.3 Conjugate gradient Versus Quasi-Newton

The first remark about these two first "academic" examples is that both methods perform well. However, the BFGS method seems more efficient than the Conjugate Gradient method, which is quite unexpected. Therefore we performed a convergence test for the 2-dimensional situation with a slightly larger crack (in order to be able to sufficiently refine the mesh on it). The crack is thus given by

$$\Gamma_C = \{(x, y, z) \in \Omega\,;\, x = 0.05,\, |y| - 0.5 < 0\}\,,$$

Table 1 compares the two minimization algorithms as the mesh is refined. As one can see the BFGS and the Conjugate Gradient method achieve the same results when the mesh gets finer.

h	Number of unknowns	BFGS				CG			
		Iter.	GE	$\|\|g_k\|\|$	CPU	Iter.	GE	$\|\|g_k\|\|$	CPU
1/2	8	6	8.3E−11	4.2E−16	0.11	1000	2.9E−9	2.4E−8	2.58
1/4	16	7	1.5E−10	7.9E−14	0.19	262	1.2E−10	7.5E−13	1.38
1/8	32	6	9.4E−11	1.1E−13	0.28	74	9.3E−11	2.8E−13	0.83
1/16	64	6	1.4E−10	1.7E−13	0.69	36	1.4E−10	2.3E−13	1.30
1/32	128	6	1.4E−10	2.1E−13	2.34	34	1.4E−10	4.5E−13	4.56
1/128	256	6	1.6E−10	2.3E−13	11.47	19	1.6E−10	3.0E−13	14.90

Table 1: Performances of the Conjugate Gradient (CG) and BFGS methods. Iter : number of iterations; GE: Ground error; $||g_k||$: final norm of the gradient; CPU: CPU time in Seconds.

7.4 An Application to Synthetic Data on the Piton de la Fournaise volcano.

In this application, we used a digital elevation model (DEM) of the Piton de la Fournaise volcano (Île de la réunion, France), provided by the french institute IGN (Institut Géographique National), to build a 3D mesh of the volcano. This is done by using the Gmsh software (see [11]). The resulting mesh is depicted on Figure 10.

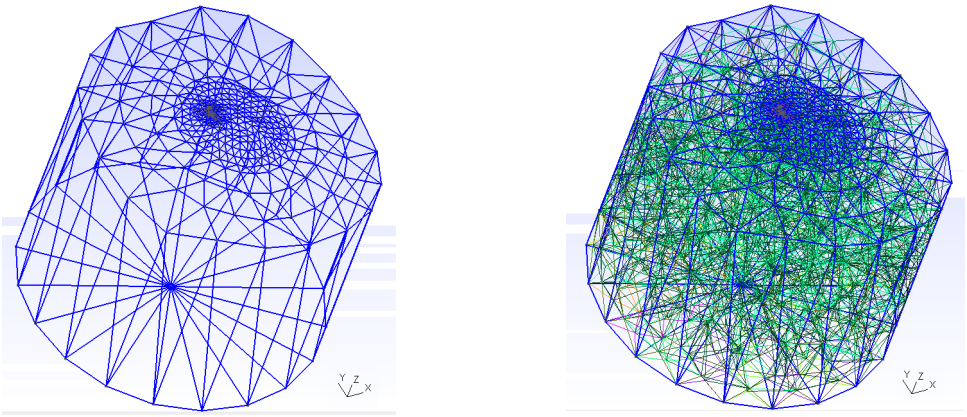

Figure 10: Realistic volcano meshes generated from a Piton de la Fournaise DEM by Gmsh. Surface mesh (left) and volume mesh (right).

The crack is a disk as displayed on Figure 11. It is located 300 meters under the ground (Figure 12). In this example, in order to validate the algorithms, we generated a synthetic solution associated to a discontinuous pressure p which is also displayed on Figure 11: the yellow patches indicate a value of 1.5 MPa and the blue patches a 0 value. The Young modulus of the material is $E = 5000$ MPa and the Poisson coefficient is $\nu = 0.25$. The Lamé coefficients are easily computed:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}.$$

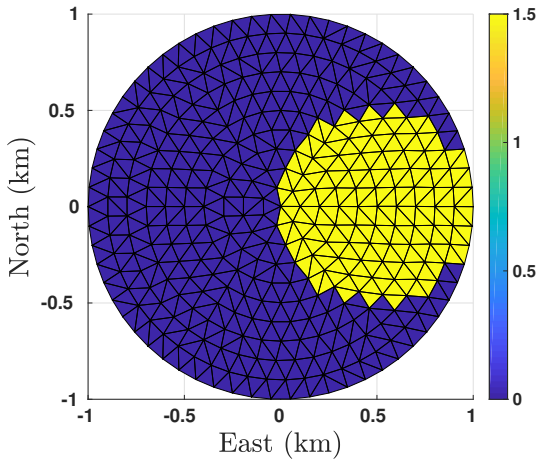

Figure 11: Crack shape and theoretical pressure p

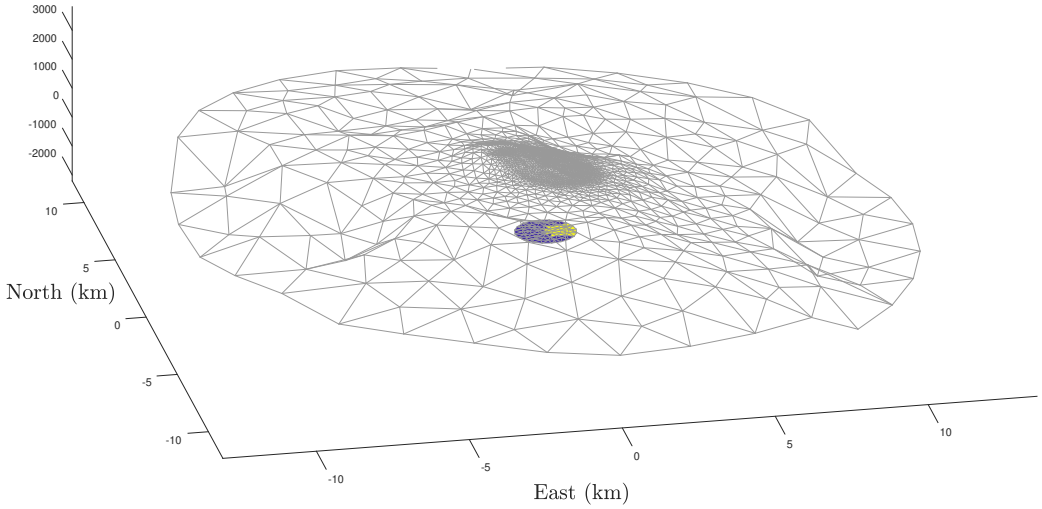

Figure 12: Crack location.

In this case, the mesh being rather coarse, the BFGS method was employed. Figures 13 and 14 show the convergence of the gradient (in L^2 norm) and the ground misfit which is very low (with respect to the size of the domain) at the end of the iterations. Finally, Figure 15 displays a node-by-node graph of the theoretical pressure, showing the initial guess and the final result. The obtained solution is practically superimposed on the theoretical one. This shows the effectiveness of the method.

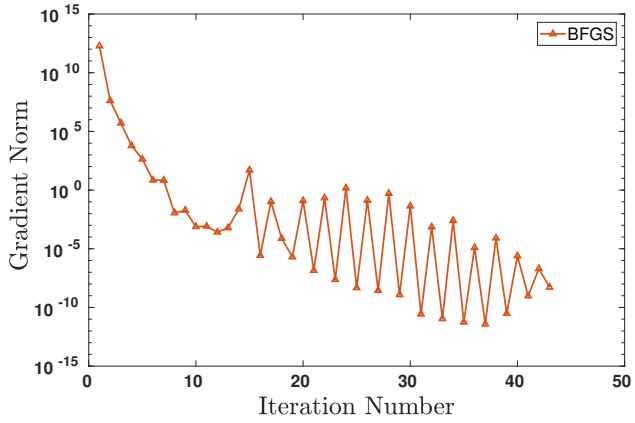

Figure 13: L^2-norm of the gradient of the cost function for the Piton de la Fournaise case.

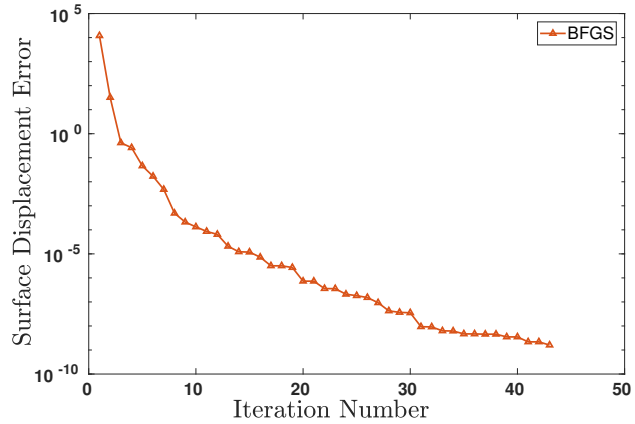

Figure 14: Evolution of the ground misfit (in L^∞-norm) for the Piton de la Fournaise case.

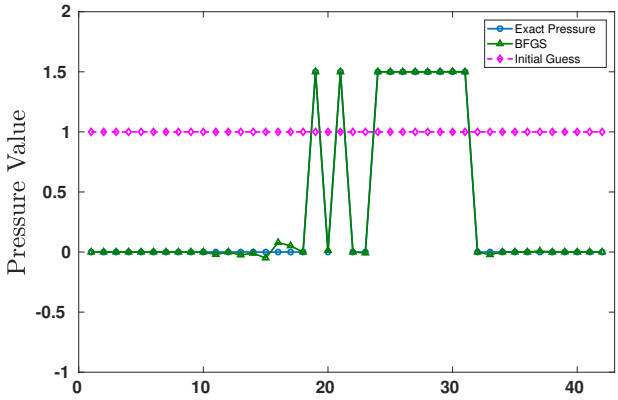

Figure 15: Convergence to the theoretical pressure for the Piton de la Fournaise case.

8 Conclusion

We have investigated methods for the determination of optimal pressure values on a crack in order to fit ground displacement in a realistic volcanologic framework. The combination of the domain decomposition and optimality conditions gives pertinent results in terms of the ground misfit minimization. Moreover we used only open source software, allowing us to make our programs available to the scientific community in the future. In a forthcoming work, we will focus on situations with physical measurements (ground displacements obtained via radar interferometry or GNSS measurements, see e.g. [21]). The problem studied here can be generalized to the identification of non-zero shear forces (full traction vectors). In a future article, we will address the inversion of the full traction vector, along with some applications. This work will be followed by the study of topological optimization questions related to this problem, leading to a full inversion process, i.e. the determination of both the crack and the pressure applied on it.

Acknowledgments

The National Centre for Space Studies (CNES) and the Laboratory of Excellence Clervolc funded the post-doctorate of Farshid Dabaghi. This is Laboratory of Excellence ClerVolc contribution number 474. This work is a contribution to the EUROVOLC project, under the EU Horizon 2020 and Innovation Action, grant No. 731070.

References

[1] B. T. Aagaard, M. G. Knepley, and C. A. Williams. A domain decomposition approach to implementing fault slip in finite-element models of quasi-static and dynamic crustal deformation. *Journal of Geophysical Research: Solid Earth*, 118(6):3059–3079, 2013.

[2] S. Amdouni, M. Moakher, and Y. Renard. A local projection stabilization of fictitious domain method for elliptic boundary value problems. *Applied Numerical Mathematics*, 76:60 – 75, 2014.

[3] P. Areias and T. Belytschko. A comment on the article "a finite element method for simulation of strong and weak discontinuities in solid mechanics" by a. hansbo and p. hansbo [comput. methods appl. mech. engrg. 193 (2004) 3523-3540]. *Computer Methods in Applied Mechanics and Engineering*, 195:1275–1276, 2006.

[4] H. J. Barbosa and T. J. Hughes. The finite element method with lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1):109 – 128, 1991.

[5] H. J. Barbosa and T. J. Hughes. Boundary lagrange multipliers in finite element methods: Error analysis in natural norms. *Numerische Mathematik*, 62(1):1–15, Dec 1992.

[6] O. Bodart, V. Cayol, S. Court, and J. Koko. XFEM-based fictitious domain method for linear elasticity model with crack. *SIAM Journal on Scientific Computing*, 38(2):B219–B246, 2016.

[7] O. Bodart, V. Cayol, F. Dabaghi, and J. Koko. An optimal control problem based on a fictitious domain method for inversion of the pressure contribution on the crack in the volcanic concepts. In R. D. Haynes, S. MacLachlan, X.-C. Cai, L. Halpern, H. H. Kim, A. Klawonn, and O. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXV*, volume 138 of *Lecture Notes in Computational Science and Engineering*, pages 401–408. Springer Verlag, 2020.

[8] G. Currenti, A. Bonaccorso, C. Del Negro, F. Guglielmino, D. Scandura, and E. Boschi. FEM-based inversion for heterogeneous fault mechanisms: application at etna volcano by DInSAR data. *Geophysical Journal International*, 183(2):765–773, Nov 2010.

[9] T.-P. Fries and M. Baydoun. Crack propagation with the extended finite element method and a hybrid explicit–implicit crack description. *International Journal for Numerical Methods in Engineering*, 89(12):1527–1558, 2012.

[10] Y. Fukushima, V. Cayol, and P. Durand. 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), 2005.

[11] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309 – 1331, 2009.

[12] A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(33):3523–3540, 2004.

[13] J. Haslinger and Y. Renard. A new fictitious domain approach inspired by the extended finite element method. *SIAM Journal on Numerical Analysis*, 47(2):1474–1499, 2009.

[14] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Grundlehren Der Mathematishen Wissenschaften 170. Springer Verlag, 1971.

[15] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[16] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, New York, NY, USA, first edition, 1999.

[18] S. Ozawa, T. Nishimura, H. Suito, T. Kobayashi, T. Mikio, and T. Imakire. Coseismic and postseismic slip of the 2011 magnitude-9 Tohoku-Oki earthquake. *Nature*, 475:373–376, 2011.

[19] B. Prabel, A. Simatos, T. Yuritzinn, and T. Charras. Propagation de fissures tridimensionnelles dans des matériaux inélastiques avec x-fem dans cast3m. *CSMA*, 2011.

[20] Y. Renard and J. Pommier. An open source generic C++ library for finite element methods. `http://home.gna.org/getfem/`.

[21] D. Smittarello, V. Cayol, V. Pinel, A. Peltier, J.-L. Froger, and V. Ferrazzini. Magma propagation at piton de la fournaise from joint inversion of insar and gnss. *Journal of Geophysical Research: Solid Earth*, 124(2):1361–1387, 2019.

[22] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.

[23] C. Wauthier, V. Cayol, K. François, and d'Oreye Nicolas. Magma sources involved in the 2002 Nyiragongo eruption, as inferred from an INSAR analysis. *Journal of Geophysical Research*, 117, 2022.

[24] C. A. Williams and L. M. Wallace. Effects of material property variations on slip estimates for subduction interface slow-slip events. *Geophysical Research Letters*, 42(4):1113–1121, 2015.