



HAL
open science

Alternating direction method of multiplier for the unilateral contact problem with an automatic penalty parameter selection

Jonas Koko

► **To cite this version:**

Jonas Koko. Alternating direction method of multiplier for the unilateral contact problem with an automatic penalty parameter selection. 2019. hal-02047516

HAL Id: hal-02047516

<https://hal.science/hal-02047516v1>

Preprint submitted on 25 Feb 2019

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.

Alternating direction method of multiplier for the unilateral contact problem with an automatic penalty parameter selection

Jonas KOKO*

Abstract

We propose an alternating direction method of multiplier (ADMM) for the unilateral (frictionless) contact problem with an optimal parameter selection. We first introduce an auxiliary unknown to separate the linear elasticity subproblem from the unilateral contact condition. Then an alternating direction is applied to the corresponding augmented Lagrangian. By eliminating the primal and auxiliary unknowns, at the discrete level, we derive a pure dual algorithm, starting point for the convergence analysis and the optimal parameter approximation. Numerical experiments are proposed to illustrate the efficiency of the proposed (optimal) penalty parameter selection method.

Keywords: Linear elasticity, unilateral contact problem, alternating direction, augmented Lagrangian.

1 Introduction

Unilateral contact (frictionless) problem is a very common problem in engineering and poses serious challenges. It differs from the classical linear elasticity only by the presence of a linear constraint (the non-penetration condition). Various numerical methods have been developed, see, e.g., [8, 6, 7, 9, 13, 14] and references therein. The method proposed in [9] is an alternating direction method of multiplier (ADMM) and it is related to augmented Lagrangian operator splitting methods [2, 3]. The main idea is to separate the differentiable part of the problem (i.e. linear elasticity) from the nondifferentiable part (i.e. the non-penetration condition) by introducing an auxiliary unknown. Applying an alternating direction method to the corresponding augmented Lagrangian leads to a simple two-step iterative method in which the matrix of the linear elasticity is constant. This property makes ADMM algorithm outperforms, see [9], another optimization based method, the popular active-set semi-smooth Newton method [6, 7, 13].

Alternating direction method of multiplier is a powerful operator-splitting algorithm for solving structured optimization problems encountered in a wide range of areas such as compressed sensing [15], image restoration [12], machine learning [1], etc. The method was initiated by Glowinski et al. [4, 2, 3], who systematically developed ADMM algorithms (also known as Uzawa block relaxation methods) for solving nonlinear partial differential equations. But ADMM based algorithms have a severe drawback: the choice of the penalty

*LIMOS, UMR 6158 CNRS – Université Clermont Auvergne, BP 10448, F-63000 Clermont-Ferrand, France

parameter. Indeed, the rate of convergence of the algorithm is strongly related to the penalty parameter and the choice of this parameter is a non trivial problem. We propose, in this paper, a simple procedure for the numerical approximation of the optimal penalty parameter for use with ADMM algorithm in the case of a frictionless contact problem. To this end, we carefully derived a convergence analysis on a pure dual ADMM algorithm. The pure dual version is obtained by eliminating the primal and auxiliary unknowns. Note that the approximate formula is based on extreme eigenvalues, and is therefore suitable for coarse meshes only.

The paper is organized as follows. In Section 2 the ADMM algorithm for the unilateral frictionless contact problem is presented. In Section 3 the finite dimensional versions are presented. The convergence analysis is carried out in Section 4. The penalty parameter approximation is outlined in Section 5, followed by the numerical experiments in Section 6.

2 The model problem and ADMM algorithm

We consider an elastic body occupying in its initial (undeformed) configuration a bounded domain Ω of \mathbb{R}^d ($d = 2, 3$) with a boundary $\Gamma = \Gamma_D \cup \Gamma_C$. We assume that the elastic body is fixed along Γ_D with $\text{meas}(\Gamma_D) > 0$. Γ_C denotes a portion of Γ which is a candidate contact surface between Ω and a rigid foundation. The normalized gap between Γ_C and the rigid foundation is denoted by g . In this paper, we consider the small strains hypothesis so that the strain tensor is $\mathcal{E}(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^t)/2$, where $\mathbf{u} = (u_1(x), \dots, u_d(x))$ is the displacement field. Hooke's law is assumed, i.e. the stress tensor is linked to the displacement through the linear relation

$$\sigma(\mathbf{u}) = C\mathcal{E}(\mathbf{u})$$

where $C = (C_{ijkl})$ is the (fourth order) elastic moduli tensor, assumed to be symmetric positive definite. Let \mathbf{n} be the outward unit normal to Ω on Γ . We consider the normal component of the displacement field and the stress tensor given by

$$\mathbf{u}_n = \mathbf{u} \cdot \mathbf{n}, \quad \sigma_n(\mathbf{u}) = (\sigma(\mathbf{u})\mathbf{n}) \cdot \mathbf{n}.$$

The unilateral contact problem consists, for a given volume force \mathbf{f} , of finding the displacement field \mathbf{u} satisfying

(i) the equilibrium equations

$$-\text{div}\sigma(\mathbf{u}) = \mathbf{f} \quad \text{in } \Omega, \quad (2.1)$$

$$\mathbf{u} = 0 \quad \text{on } \Gamma_D, \quad (2.2)$$

(ii) and the contact (i.e. non-penetration) conditions

$$\mathbf{u}_n - g \leq 0, \quad \sigma_n(\mathbf{u}) \leq 0, \quad (\mathbf{u}_n - g)\sigma_n(\mathbf{u}) = 0, \quad \text{on } \Gamma_C. \quad (2.3)$$

Let us introduce the functions space $V = \{v \in H^1(\Omega)^d, \mathbf{v} = 0 \text{ on } \Gamma_D\}$ and the set of admissible displacements $K = \{\mathbf{v} \in V, \mathbf{v}_n - g \leq 0 \text{ on } \Gamma_C\}$. Let $a(\cdot, \cdot)$ be the symmetric, continuous and coercive bilinear form which corresponds to the virtual work in the elastic body

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \sigma(\mathbf{u}) : \mathcal{E}(\mathbf{v}) dx.$$

We denote by $\mathbf{f}(\cdot)$ the linear form of external forces

$$\mathbf{f}(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} dx.$$

Using the potential energy functional

$$J(\mathbf{v}) = \frac{1}{2}a(\mathbf{v}, \mathbf{v}) - f(\mathbf{v})$$

we can formulate the unilateral contact problem as the constrained minimization problem

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

$$J(\mathbf{u}) \leq J(\mathbf{v}), \quad \forall \mathbf{v} \in K. \quad (2.4)$$

To achieve a solution of (2.4) by an ADMM algorithm, we need additional steps. Following Glowinski and Le Tallec [3], we introduce the set

$$C = \{p \in L^2(\Gamma_C), p - g \leq 0 \text{ on } \Gamma_c\}$$

and its characteristic functional $\mathcal{I}_C : L^2(\Gamma_c) \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by

$$\mathcal{I}_C(p) = \begin{cases} 0 & \text{if } p \in C, \\ +\infty & \text{if } p \notin C. \end{cases}$$

It is clear that (2.4) is equivalent to the following constrained minimization problem

Find $(\mathbf{u}, p) \in V \times L^2(\Gamma_C)^2$ such that

$$J(\mathbf{u}) + \mathcal{I}_C(p) \leq J(\mathbf{v}) + \mathcal{I}_C(q) \quad \forall (\mathbf{v}, q) \in V \times L^2(\Gamma_C)^2, \quad (2.5)$$

$$\mathbf{u}_n - p = 0 \text{ on } \Gamma_C. \quad (2.6)$$

With (2.5)-(2.6) we associate the augmented Lagrangian functional

$$\mathcal{L}_r(\mathbf{v}, q; \lambda) = J(\mathbf{v}) + \mathcal{I}_C(q) + (\mathbf{v}_n - q, \lambda)_{\Gamma_C} + \frac{r}{2} \|\mathbf{v}_n - q\|_{\Gamma_C}^2, \quad (2.7)$$

where $r > 0$ is the penalty parameter. ADMM algorithm for (2.4) is based on (2.7). The method performs successive minimizations in \mathbf{u} and p followed by the multiplier λ update as follows

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{v}} \mathcal{L}_r(\mathbf{v}, p^k; \lambda^k), \quad (2.8)$$

$$p^{k+1} = \arg \min_p \mathcal{L}_r(\mathbf{u}^{k+1}; q; \lambda^k), \quad (2.9)$$

$$\lambda^{k+1} = \lambda^k + r(\mathbf{u}_n^{k+1} - p^{k+1}). \quad (2.10)$$

Minimization (2.8) leads to an equilibrium problem which always has a unique solution even though the original problem allows rigid body motions. Minimization (2.9) is solved explicitly. The corresponding ADMM algorithm is presented in Algorithm 1 (see [9] for detailed derivation). We iterate until the relative error on \mathbf{u}^k and p^k becomes "sufficiently" small.

Algorithm 1 ADMM algorithm for the unilateral contact problem

Initialization. ϕ^0 and λ^0 are given.

Iteration $k \geq 0$. Compute successively \mathbf{u}^{k+1} , p^{k+1} and λ^{k+1} as follows

- Find $\mathbf{u}^{k+1} \in V$ such that

$$a(\mathbf{u}^{k+1}, \mathbf{v}) + r(\mathbf{u}_n^{k+1}, \mathbf{v}_n)_{\Gamma_C} = \mathbf{f}(\mathbf{v}) + (rp^k - \lambda^k, \mathbf{v}_n)_{\Gamma_C}, \quad \forall \mathbf{v} \in V.$$

- Compute the auxiliary unknown

$$p^{k+1} = \mathbf{u}_n^{k+1} + \frac{1}{r} \left[\lambda^k - \left(\lambda^k + r(\mathbf{u}_n^{k+1} - g) \right)^+ \right].$$

- Update the Lagrange multiplier

$$\lambda^{k+1} = \lambda^k + r(\mathbf{u}_n^{k+1} - p^{k+1}).$$

3 Finite dimensional algorithms

We assume that $\Omega \subset \mathbb{R}^d$ is a polyhedral domain and therefore can be exactly triangulated. We consider a (continuous piecewise linear) finite element triangulation \mathcal{T}_h of Ω consistent with the decomposition of its boundary Γ into Γ_D and Γ_C . If n is the number of nodes of the triangulation, the dimension of the finite element subspace $V_h \subset V$ is $\dim V_h = dn$.

Let \mathbf{x}_j ($j = 1, \dots, m$) be a contact node (i.e. \mathbf{x}_j lies on Γ_C). The displacement vector and the unit outward normal at \mathbf{x}_j are denoted by \mathbf{u}_j and \mathbf{n}_j , respectively. We introduce the linear mappings $\mathbf{N} : \mathbb{R}^{dn} \rightarrow \mathbb{R}^m$, such that \mathbf{Nu} is the vector of the normal components of \mathbf{u} at contact nodes, that is, $(\mathbf{Nu})_j = \mathbf{u}_j^\top \mathbf{n}_j$, $j = 1, \dots, m$.

The finite element discretization leads to the following matrices and vectors

- \mathbf{A} , $(dn) \times (dn)$ stiffness matrix (symmetric positive definite), i.e. from the bilinear form $a(u, v)$;
- \mathbf{M} normal mass matrices ($m \times m$ symmetric positive definite);
- $\mathbf{f} \in \mathbb{R}^{dn}$ the (discrete) external forces;
- $\mathbf{g} \in \mathbb{R}^m$ the (discrete) normalized gap;
- $\mathbf{p} \in \mathbb{R}^m$ the (discrete) auxiliary unknown;
- $\boldsymbol{\lambda} \in \mathbb{R}^m$ the (discrete) lagrange multiplier.

With the notations above, the augmented Lagrangian (2.7) now reads in the discrete setting

$$\mathcal{L}_r(\mathbf{u}, \mathbf{p}; \boldsymbol{\lambda}) = J(\mathbf{u}) + \boldsymbol{\lambda}^\top \mathbf{M}(\mathbf{Nu} - \mathbf{p}) + \frac{r}{2} (\mathbf{Nu} - \mathbf{p})^\top \mathbf{M}(\mathbf{Nu} - \mathbf{p}), \quad (3.1)$$

where

$$J(\mathbf{u}) = \frac{1}{2} \mathbf{u}^\top \mathbf{A} \mathbf{u} - \mathbf{f}^\top \mathbf{u}.$$

3.1 Discrete ADMM algorithm

The derivation of the discrete version of Algorithm 1 from the augmented Lagrangian function (3.1) is straightforward, Algorithm 2.

Algorithm 2 Algebraic ADMM algorithm for the unilateral contact problem

Initialization $k = 0$ $r > 0$, \mathbf{p}^0 and $\boldsymbol{\lambda}^0$ are given.

Iteration $k \geq 0$. Compute successively \mathbf{u}^{k+1} , \mathbf{p}^{k+1} and $\boldsymbol{\lambda}^{k+1}$ as follows.

Step 1 Compute $\mathbf{u}^{k+1} \in V_h$ such that

$$(\mathbf{A} + r\mathbf{N}^\top \mathbf{M}\mathbf{N})\mathbf{u}^{k+1} = \mathbf{f} + \mathbf{N}^\top \mathbf{M}(r\mathbf{p}^k - \boldsymbol{\lambda}^k) \quad (3.2)$$

Step 2 Compute $\mathbf{p}^{k+1} \in P_h$

$$\mathbf{p}^{k+1} = \mathbf{N}\mathbf{u}^{k+1} + \frac{1}{r} \left(\boldsymbol{\lambda}^k - (\boldsymbol{\lambda}^k + r(\mathbf{N}\mathbf{u}^{k+1} - g))^+ \right) \quad (3.3)$$

Step 3 Update Lagrange multiplier

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + r(\mathbf{N}\mathbf{u}^{k+1} - \mathbf{p}^{k+1})$$

3.2 ADMM algorithm without the auxiliary unknown

Algorithm 2 can be simplified by eliminating the auxiliary unknown \mathbf{p} . Indeed, using the update formula (3.3) we have

$$\begin{aligned} r\mathbf{p}^k &= r(\mathbf{p}^k - \mathbf{N}\mathbf{u}^k) + r\mathbf{N}\mathbf{u}^k \\ &= -(\boldsymbol{\lambda}^{k-1} + r(\mathbf{N}\mathbf{u}^k - \mathbf{p}^k)) + r\mathbf{N}\mathbf{u}^k + \boldsymbol{\lambda}^{k-1} \\ &= r\mathbf{N}\mathbf{u}^k + \boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^k, \end{aligned}$$

that is

$$r\mathbf{p}^k - \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} - 2\boldsymbol{\lambda}^k + r\mathbf{N}\mathbf{u}^k. \quad (3.4)$$

If we set $\boldsymbol{\lambda}_-^k = \min(0, \boldsymbol{\lambda}^{k-1} + r(\mathbf{N}\mathbf{u}^k - g))$, the right-hand side of (3.4) becomes

$$\begin{aligned} \boldsymbol{\lambda}^{k-1} + r\mathbf{N}\mathbf{u}^k - 2\boldsymbol{\lambda}^k &= \boldsymbol{\lambda}^{k-1} + r(\mathbf{N}\mathbf{u}^k - \mathbf{g}) + r\mathbf{g} - 2\boldsymbol{\lambda}^k \\ &= \boldsymbol{\lambda}_-^k - \boldsymbol{\lambda}^k + r\mathbf{g}. \end{aligned}$$

It follows that

$$r\mathbf{p}^k - \boldsymbol{\lambda}^k = \boldsymbol{\lambda}_-^k - \boldsymbol{\lambda}^k + r\mathbf{g}$$

i.e., we have eliminated the auxiliary unknown \mathbf{p} from the right-hand side of (3.2). It remains to remove \mathbf{p} from the multiplier update formula. From (3.3) we deduce that

$$r(\mathbf{p}^{k+1} - \mathbf{N}\mathbf{u}^{k+1}) = \boldsymbol{\lambda}^k - (\boldsymbol{\lambda}^k + r(\mathbf{N}\mathbf{u}^{k+1} - g))^+. \quad (3.5)$$

Substituting (3.5) into the Lagrange multiplier update formula, we obtain the new multiplier update formula

$$\boldsymbol{\lambda}_+^{k+1} := (\boldsymbol{\lambda}^k + r(\mathbf{N}\mathbf{u}^{k+1} - g))^+.$$

Gathering the results above, the ADMM algorithm without the auxiliary unknown \mathbf{p} is described in Algorithm 3. In Algorithm 3, $\boldsymbol{\lambda}_+^0$ and $\boldsymbol{\lambda}_-^0$ must be consistent. Indeed, if, e.g., $\mathbf{u}^0 = 0$, then we can set $\boldsymbol{\lambda}_+^0 = r(\mathbf{N}\mathbf{u}^0 - \mathbf{g})^+ = 0$. It follows that $\boldsymbol{\lambda}_-^0 = r(\mathbf{N}\mathbf{u}^0 - \mathbf{g})^- = -r\mathbf{g}$.

Algorithm 3 ADMM without auxiliary unknown for the unilateral contact problem

Initialization $k = 0$ $r > 0$, $\boldsymbol{\lambda}_+^0$ and $\boldsymbol{\lambda}_-^0$ are given.

Iteration $k \geq 0$. Compute successively \mathbf{u}^{k+1} , $\boldsymbol{\lambda}_+^{k+1}$ and $\boldsymbol{\lambda}_-^{k+1}$ as follows.

Step 1 Compute \mathbf{u}^{k+1} such that

$$(\mathbf{A} + r\mathbf{N}^\top \mathbf{M}\mathbf{N})\mathbf{u}^{k+1} = \mathbf{f} + r\mathbf{N}^\top \mathbf{M}\mathbf{g} + \mathbf{N}^\top \mathbf{M}(\boldsymbol{\lambda}_-^k - \boldsymbol{\lambda}_+^k) \quad (3.6)$$

Step 2 Update the Lagrange multipliers

$$\begin{aligned} \boldsymbol{\lambda}_+^{k+1} &= (\boldsymbol{\lambda}_+^k + r(\mathbf{N}\mathbf{u}^{k+1} - \mathbf{g}))^+ \\ \boldsymbol{\lambda}_-^{k+1} &= (\boldsymbol{\lambda}_-^k + r(\mathbf{N}\mathbf{u}^{k+1} - \mathbf{g}))^- \end{aligned}$$

3.3 Pure dual version

The pure dual version of the ADMM algorithm for the unilateral contact problem is obtained by eliminating the displacements vector \mathbf{u} . From (3.6) we have

$$\mathbf{u}^{k+1} = \mathbf{A}_r^{-1} \left(\mathbf{f} + r\mathbf{N}^\top \mathbf{M}\mathbf{g} + \mathbf{N}^\top \mathbf{M}(\boldsymbol{\lambda}_-^k - \boldsymbol{\lambda}_+^k) \right)$$

where we have set

$$\mathbf{A}_r = \mathbf{A} + r\mathbf{N}^\top \mathbf{M}\mathbf{N}.$$

Substituting \mathbf{u}^{k+1} into the Lagrange multiplier update formulas, we obtain Algorithm 4.

Algorithm 4 Dual ADMM algorithm for the unilateral contact problem

Initialization $k = 0$ $r > 0$, $\boldsymbol{\lambda}_+^0$ and $\boldsymbol{\lambda}_-^0$ are given.

Set $\mathbf{A}_r = \mathbf{A} + r\mathbf{N}^\top \mathbf{M}\mathbf{N}$ and $\mathbf{a} = \mathbf{A}_r^{-1}(\mathbf{f} + r\mathbf{N}^\top \mathbf{M}\mathbf{g})$

Iteration $k \geq 0$. Compute successively $\boldsymbol{\lambda}_+^{k+1}$ and $\boldsymbol{\lambda}_-^{k+1}$ as follows.

$$\boldsymbol{\lambda}_+^{k+1} = \left[(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top \mathbf{M})\boldsymbol{\lambda}_+^k + r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top \mathbf{M}\boldsymbol{\lambda}_-^k + r(\mathbf{N}\mathbf{a} - \mathbf{g}) \right]^+ \quad (3.7)$$

$$\boldsymbol{\lambda}_-^{k+1} = \left[(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top \mathbf{M})\boldsymbol{\lambda}_-^k + r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top \mathbf{M}\boldsymbol{\lambda}_+^k + r(\mathbf{N}\mathbf{a} - \mathbf{g}) \right]^- \quad (3.8)$$

Algorithm 4 is unpracticable since in \mathbf{A}_r^{-1} can be a full matrix of large size. However, the recursive relation (3.7)-(3.8) is important because it gives an easy way to prove the convergence of the ADMM algorithm. Indeed, Algorithm 4 is useful as a theoretical basis for the study of the influence of r .

4 Convergence

Since $\min(0, x) = -\max(0, -x)$, the recursive relation (3.7)-(3.8) can be rewritten in the useful form

$$\boldsymbol{\lambda}_+^{k+1} = \left[(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k + r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k + r(\mathbf{N}\mathbf{a} - \mathbf{g}) \right]^+ \quad (4.1)$$

$$\boldsymbol{\lambda}_-^{k+1} = - \left[-(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k - r(\mathbf{N}\mathbf{a} - \mathbf{g}) \right]^+ . \quad (4.2)$$

Since for any $x \in \mathbb{R}^n$ $\|x^+\| \leq \|x\|$ we have

$$\|\boldsymbol{\lambda}_+^{k+1}\|^2 \leq \|(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k + r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k\|^2 + \|r(\mathbf{N}\mathbf{a} - \mathbf{g})\|^2 \quad (4.3)$$

$$\|\boldsymbol{\lambda}_-^{k+1}\| \leq \|-(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k\| + \|r(\mathbf{N}\mathbf{a} - \mathbf{g})\| \quad (4.4)$$

so that

$$\begin{aligned} \|\boldsymbol{\lambda}_+^{k+1}\|^2 + \|\boldsymbol{\lambda}_-^{k+1}\|^2 &\leq \|(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k + r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k\|^2 \\ &\quad + \|-(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M})\boldsymbol{\lambda}_+^k - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}\boldsymbol{\lambda}_-^k\|^2 + 2\|r(\mathbf{N}\mathbf{a} - \mathbf{g})\|^2 \end{aligned}$$

that is

$$\|\boldsymbol{\lambda}_\pm^{k+1}\|^2 \leq \|\mathcal{A}\boldsymbol{\lambda}_\pm^{k+1}\|^2 + 2\|r(\mathbf{N}\mathbf{a} - \mathbf{g})\|^2,$$

where we have set

$$\boldsymbol{\lambda}_\pm^{k+1} = \begin{bmatrix} \boldsymbol{\lambda}_+^{k+1} \\ \boldsymbol{\lambda}_-^{k+1} \end{bmatrix} \quad \text{and} \quad \mathcal{A} = \begin{bmatrix} \mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M} & r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M} \\ -(\mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}) & -r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M} \end{bmatrix}.$$

The convergence of Algorithm 4 therefore depends on the spectral radius of \mathcal{A} . We have the following proposition for the non-zero eigenvalues of \mathcal{A} .

Proposition 4.1 *The non-zero eigenvalues of \mathcal{A} are eigenvalues of $\mathbb{I} - 2r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}$ and conversely.*

PROOF. Let us set

$$\mathcal{A}_1 = \mathbb{I} - r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M} \quad \text{and} \quad \mathcal{A}_2 = r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}$$

so that

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \\ -\mathcal{A}_1 & -\mathcal{A}_2 \end{bmatrix}.$$

We have to show that the non-zero eigenvalues of \mathcal{A} are eigenvalues of $\mathcal{A}_1 - \mathcal{A}_2 = \mathbb{I} - 2r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^T\mathbf{M}$ and conversely. Let α be an eigenvalue of \mathcal{A} and $(\mathcal{X}_1, \mathcal{X}_2)$ the corresponding eigenvector. We have

$$\mathcal{A}_1\mathcal{X}_1 + \mathcal{A}_2\mathcal{X}_2 = \alpha\mathcal{X}_1, \quad (4.5)$$

$$-\mathcal{A}_1\mathcal{X}_1 - \mathcal{A}_2\mathcal{X}_2 = \alpha\mathcal{X}_2 \quad (4.6)$$

that is, $\alpha(\mathcal{X}_1 + \mathcal{X}_2) = 0$. For non-zero eigenvalues we deduce that the eigenvectors are such that $\mathcal{X}_1 = -\mathcal{X}_2$. Substituting in (4.5)-(4.6) we obtain

$$(\mathcal{A}_1 - \mathcal{A}_2)\mathcal{X}_1 = \alpha\mathcal{X}_1,$$

$$(\mathcal{A}_1 - \mathcal{A}_2)\mathcal{X}_2 = \alpha\mathcal{X}_2.$$

We deduce that α is an eigenvalue of $\mathcal{A}_1 - \mathcal{A}_2$. From the relations above, it is obvious that non-zero eigenvalues of $\mathcal{A}_1 - \mathcal{A}_2$ are also eigenvalues of \mathcal{A} . \square

To study the eigenvalues of $\mathbb{I} - 2r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top\mathbf{M}$, we consider the following sequence

$$\boldsymbol{\mu}^{k+1} = (\mathbb{I} - 2r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top\mathbf{M})\boldsymbol{\mu}^k = \boldsymbol{\mu}^k - 2r\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k, \quad k \geq 0. \quad (4.7)$$

We follow the same procedure as [3, p. 46] for a quadratic programming problem with equality constraints. From (4.7), we deduce that

$$\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^{k+1} = \mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k - 2r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}\mathbf{A}_r^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k. \quad (4.8)$$

Since

$$\mathbf{A}_r^{-1} = (\mathbf{A} + r\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1} = (\mathbb{I} + r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1}\mathbf{A}^{-1}$$

it follows that

$$\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^{k+1} = \mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k - 2r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}(\mathbb{I} + r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1}\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k \quad (4.9)$$

By introducing the auxiliary unknown $\boldsymbol{\eta}^k = \mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\boldsymbol{\mu}^k$, the sequence (4.9) becomes

$$\boldsymbol{\eta}^{k+1} = (\mathbb{I} - 2r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}(\mathbb{I} + r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1})\boldsymbol{\eta}^k, \quad \forall k \geq 0. \quad (4.10)$$

The convergence of the sequence $\{\boldsymbol{\eta}^k\}$ will give us informations about the convergence of the sequence $\{\boldsymbol{\mu}^k\}$ and then $\{\boldsymbol{\lambda}_+^k, \boldsymbol{\lambda}_-^k\}$. From [5, th. 1.1.6 and cor. 2.4.3.4], $\mathbf{B} = (\mathbb{I} - 2r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}(\mathbb{I} + r\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1})$ is a polynomial function of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$, with $p(t) = (1 - 2rt(1 + rt)^{-1})$ as polynomial. Then the eigenvalues of \mathbf{B} are polynomial functions of the eigenvalues of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ and thus $\lambda_i(\mathbf{B}) = p(\lambda_i(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}))$, that is

$$\lambda_i(\mathbf{B}) = 1 - 2r \frac{\lambda_i(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})}{1 + r\lambda_i(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})} = \frac{1 - r\lambda_i(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})}{1 + r\lambda_i(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N})}. \quad (4.11)$$

The eigenvalues (and eigenvectors) of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ are solution to the generalized symmetric eigenvalue problem

$$\mathbf{N}^\top\mathbf{M}\mathbf{N}\mathbf{w} = \lambda\mathbf{A}\mathbf{w}$$

Since \mathbf{A} is symmetric positive definite and $\mathbf{N}^\top\mathbf{M}\mathbf{N}$ is symmetric positive semi-definite, the eigenvalues of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ are non-negative, and the eigenvectors corresponding to two distinct eigenvalues are A -orthogonal. It follows that the eigenvalues of \mathbf{B} are such that

$$0 \leq \lambda_i(\mathbf{B}) \leq 1, \quad \forall i. \quad (4.12)$$

Note that \mathbf{A} and \mathbf{M} are square and non singular matrices. Then if 0 is an eigenvalue of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$, then the corresponding eigen-subspace is $\ker(\mathbf{N})$. Thus $R(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M})$ is spanned by the eigenvectors of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ associated with strictly positive eigenvalues (using the property $R(\mathbf{N}) = (\ker(\mathbf{N}))^\perp$). Since $\{\boldsymbol{\eta}^k\} \subset R(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M})$, we can write (4.10) in a basis of $R(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M})$ formed with eigenvectors \mathbf{w}_i of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ associated with strictly positive eigenvalues λ_i . We get

$$\boldsymbol{\eta}_i^{k+1} = \frac{1 - r\lambda_i}{1 + r\lambda_i} \boldsymbol{\eta}_i^k, \quad \forall i = 1, \dots, N_1 \text{ and } k \geq 0.$$

We deduce the following convergence theorem.

Theorem 4.2 (Convergence) *ADMM Algorithm 4 converges for any $r > 0$.*

Corollary 4.3 (optimal step-size and convergence rate) *The optimal penalty parameter for r is*

$$r^* = \frac{1}{\sqrt{\lambda_m \lambda_M}} \quad (4.13)$$

where λ_m and λ_M are, respectively, the smallest nonzero eigenvalue and the largest eigenvalue of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$. With the choice (4.13), the convergence of Algorithm 4 is linear with an asymptotic constant θ satisfying

$$\theta \leq \frac{1 - (\lambda_m/\lambda_M)^{1/2}}{1 + (\lambda_m/\lambda_M)^{1/2}}. \quad (4.14)$$

PROOF: For (4.13), see, e.g., [3, ch. 3]. The convergence factor (4.14) is deduced from the behavior of the function $\xi \mapsto (1 - \xi)/(1 + \xi)$. \square

If $\mathbf{N}^\top\mathbf{M}\mathbf{N}$ is nonsingular, we can express (4.14) using the (Euclidean) condition number of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ given by

$$\kappa_2(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}) = \|\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}\|_2 \|(\mathbf{N}^\top\mathbf{M}\mathbf{N})^{-1}\mathbf{A}\|_2 = \lambda_M/\lambda_m.$$

We obtain the following corollary.

Corollary 4.4 *If $\mathbf{N}^\top\mathbf{M}\mathbf{N}$ is nonsingular, then the eigenvalues of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ are strictly positive and Algorithm 4 with the optimal choice (4.13) converges linearly with an asymptotic constant θ satisfying*

$$\theta \leq \frac{\kappa_2(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}) - 1}{\kappa_2(\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}) + 1}$$

5 Penalty parameter approximation

We now study how to compute λ_m and λ_M , the smallest nonzero eigenvalue and the largest eigenvalue of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$. For this purpose, it is convenient to arrange the indices in such a way that the contact indices C and the interior indices I occur in a consecutive order. This leads to the block matrix representation of the stiffness matrix as

$$\mathbf{A} = \begin{pmatrix} A_{II} & A_{IC} \\ A_{CI} & A_{CC} \end{pmatrix}$$

and

$$\mathbf{N}^\top\mathbf{M}\mathbf{N} = \begin{pmatrix} O_{II} & O_{IC} \\ O_{CI} & M_{CC} \end{pmatrix},$$

where O_{II} , O_{IC} and O_{CI} are matrices of zeros; and M_{CC} is the contact boundary (diagonal) mass matrix. Consequently, if we set

$$\mathbf{A}^{-1} = \begin{pmatrix} \tilde{A}_{II} & \tilde{A}_{IC} \\ \tilde{A}_{CI} & \tilde{A}_{CC} \end{pmatrix},$$

then the nonzero eigenvalues of $\mathbf{A}^{-1}\mathbf{N}^\top\mathbf{M}\mathbf{N}$ are eigenvalues of $\tilde{A}_{CC}M_{CC}$. To compute \tilde{A}_{CC} , we can apply block-Gaussian elimination to \mathbf{A} with A_{II} as block-pivot

$$\tilde{A}_{CC} = (A_{CC} - A_{CI}A_{II}^{-1}A_{IC})^{-1}.$$

It follows that \tilde{A}_{CC} is the Schur complement of A_{II} in \mathbf{A} .

We now detail the practical steps for the approximation of the penalty parameter. Since computing A_{II}^{-1} is unpracticable even for medium size problem, we compute \tilde{A}_{CC} by solving equivalent linear systems. We obtain Algorithm 5.

Algorithm 5 Algorithm for computing the approximate penalty parameter r^*

Step 1. Solve for \tilde{X} the system

$$A_{II}\tilde{X} = A_{IC} \quad (5.1)$$

Step 2. Compute the schur complement of A_{II}

$$S = A_{CC} - A_{CI}\tilde{X}$$

Step 3. Solve for X

$$SX = M_{CC} \quad (5.2)$$

Step 4. Compute λ_m and λ_M the extreme eigenvalues of X and the optimal penalty parameter approximation (4.13).

Solving (5.1) or (5.2) is equivalent to solving several linear systems that have the same matrix but different right-hand sides. Since the different right-hand sides are available, all the systems of equations can be solved at the same time using a single Gaussian elimination (e.g. using MATLAB \ operator : $X = S \setminus M_{CC}$ for (5.2)).

6 Numerical experiments

We have implemented the algorithms described in the previous sections in MATLAB, using piece-wise linear finite element, vectorized assembling functions and the mesh generator provided in [10, 11], on a computer equipped running Linux (Ubuntu 16.04) with 3.00GHz clock frequency and 32GB RAM. The ADMM solver used in the experiments is based on Algorithm 1. The test problems used are designed in order to illustrate the behavior of the algorithms more than to model contact actual phenomena. Since our approximation procedure combines matrix inversion and computation of extreme eigenvalues, we restrict our penalty parameter estimation to small size problems.

We use the following notations

- r^* approximate optimal penalty parameter obtained by (4.13).
- \bar{r} standard numerical optimal penalty parameter obtained using the uniform sampling of interval $(.1, \alpha E)$, where $\alpha > 1$ and E is the Young modulus.
- m_c the size of matrix $\tilde{A}_{CC}M_{CC}$ used for computing r^* . To reduce the computational cost $m_c \leq 100$ in the numerical experiments.
- Iter. The number of iterations required for convergence in Algorithm 2.
- CPU Time. CPUT time in Seconds.

6.1 Example 1: 2D problem

We consider a rectangular elastic body $\Omega = (0, 2) \times (0, 1)$ with the boundary partition $\Gamma_D = \{0\} \times (0, 1)$, $\Gamma_N = (0, 2) \times \{1\}$ and $\Gamma_C = (0, 2) \times \{0\}$. On Γ_D the body is clamped ($\mathbf{u}_D = 0$) and on Γ_N non-homogeneous Neumann boundary conditions are prescribed $\sigma(\mathbf{u}) \cdot \mathbf{n} = -10$. On Γ_C , the normalized gap between Ω and the rigid foundation is $g(x) = 0.01$. The material constants are $E = 2000$ and $\nu = 0.3$.

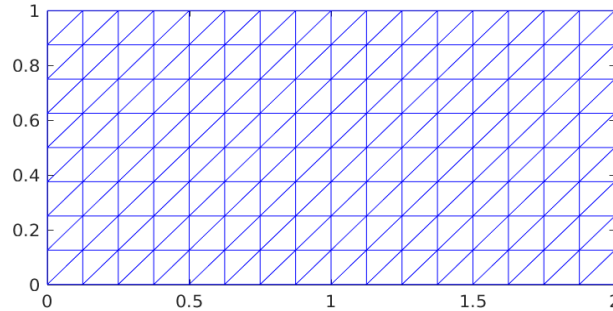


Figure 1: Mesh sample for $\Omega = (0, 2) \times (0, 1)$ of size $h = 1/8$

We first consider the meshes of size $h=1/8$ (Figure 1), $1/16$ and $1/32$. We compare the approximate penalty parameter r^* obtained with (4.13) with the numerical penalty parameter \bar{r} obtained by sampling the interval $(0.1, 10) * E$ with 200 points uniformly spaced. The evolution of the number of iterations with respect to the penalty parameter is shown in Figure 2. The normal stress distribution on Γ_C is shown in Figure 3

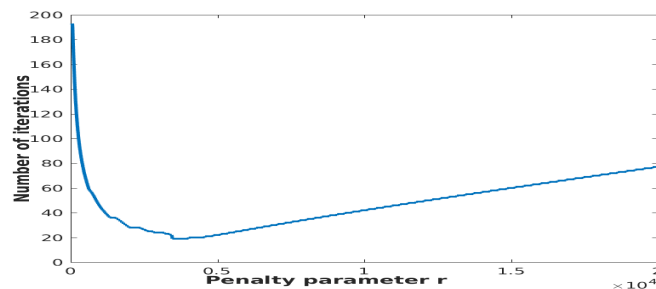


Figure 2: Number of iterations versus the penalty parameter for Example 1.

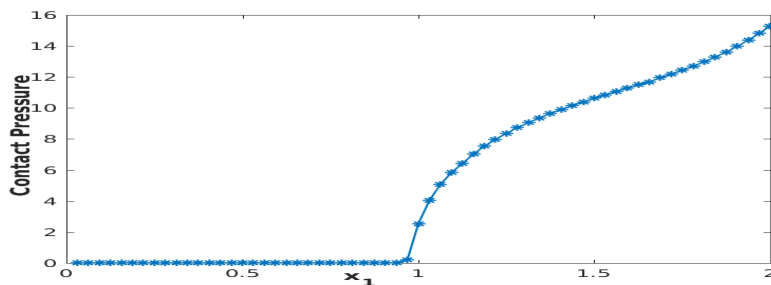


Figure 3: Normal Stress distribution on Γ_C for Example 1.

We summarize in Table 1 the (optimal) penalty parameters and the behavior of Al-

gorithm 2 in terms of CPU time and number of iterations. Note that the CPU time for r^* includes the time spent in both Algorithm 2 and Algorithm 5. The CPU time for \bar{r} is the computational time of the whole sampling procedure. We can notice that the sampling process gives better results in terms of the number of iterations required by the ADMM solver. But sampling is a time-consuming process. For the largest problem ($h = 1/32$), obtaining the optimal penalty parameter (4.13) with Algorithm 5 is about 80 times faster than the standard sampling procedure. Table 2 shows that, for $r^* = 3134.1289$ and $\bar{r} = 3510$ (obtained using a mesh of size $1/32$), the performances of Algorithm 2 are almost comparable for large size meshes.

h	1/8	1/16	1/32
size of m_c	17	33	65
r^*	1591.4878	2220.5867	3134.1289
Iter.	35	27	27
CPU Time	0.057	0.177	0.269
\bar{r}	3510	2810	3510
Iter.	19	18	19
CPU Time	1.797	4.957	23.766

Table 1: \bar{r} obtained by sampling Versus r^* obtained by (4.13).

h		1/8	1/16	1/32	1/64	1/128	1/256
$r^* = 3134.1289$	Iterations	24	19	27	25	19	21
	CPU Time	0.009	0.018	0.096	0.479	1.603	9.127
$\bar{r} = 3510$	Iterations	19	20	19	20	20	20
	CPU Time	0.008	0.014	0.078	0.339	1.646	8.713

Table 2: Performances of Algorithm 2 with $r^* = 3134.1289$ and $\bar{r} = 3510$ for Example 1.

7 Example 2: Hertz problem

The Hertz problem is a classical test problem in the numerical simulation of unilateral contact problem. It consists of an infinitely long cylinder resting in a rigid foundation, and subjected to a uniform load along its top of intensity $f = -(0, 1600)$. The radius of the cylinder is $R = 8$. The material constants for the cylinder are $E = 2000$ and $\nu = 0.3$. For symmetry reason, only quarter of the cylinder is consider (Figure 4) and we set $u_1 = 0$ on $\Gamma_D = \{0\} \times (0, 8)$. The contact surface is $\Gamma_C = \{\mathbf{x} \in (0, 8)^2 \mid x_1^2 + x_2^2 = 16\}$.

We first consider meshes with 442 and 1692 with 35 and 69 nodes on Γ_C , respectively. Note that the problem allows a rigid body motion in the vertical direction. In ADMM Algorithm 2, the mass terms provided by the normal integral prevent infinite displacements at the initial step. But for Algorithm 5, A_{II} is only positive semidefinite and then non invertible. We simply add a small positive constant on the vertical components of Γ_D . As in Example 1, we compute \bar{r} by sampling $(.1, 10) * E$ with 200 points, uniformly spaced. Figure 5 shows the variation of the number of iterations with respect to the penalty parameter. The deformed configuration and the stress distribution on the contact Γ_C are shown in Figure 6 and 7, respectively.

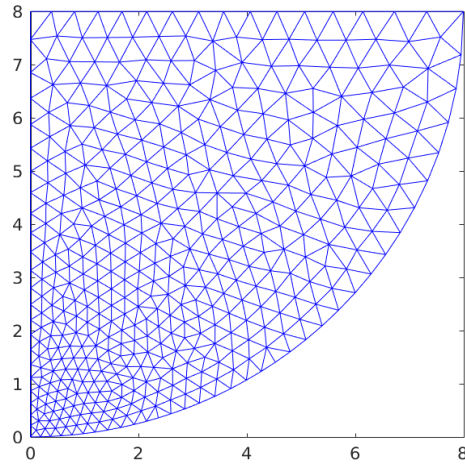


Figure 4: Mesh sample for the Hertz problem.

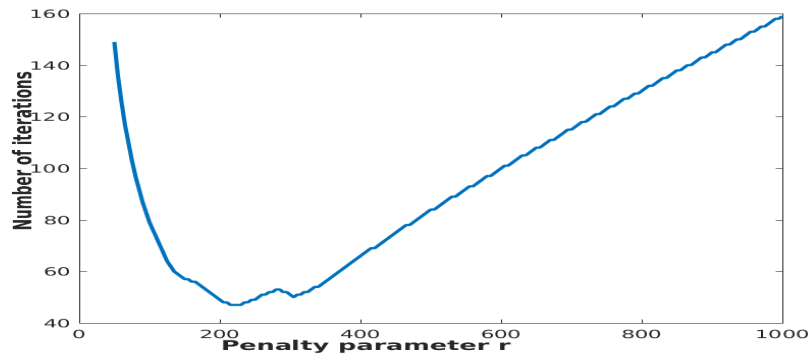


Figure 5: Number of iterations versus penalty parameter for the hertz problem.

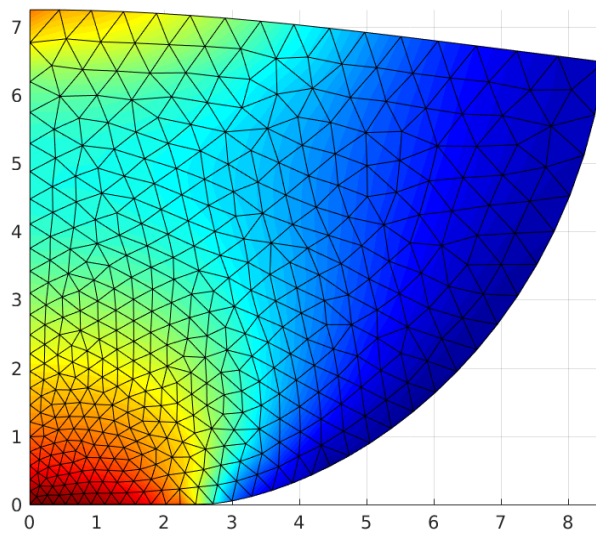
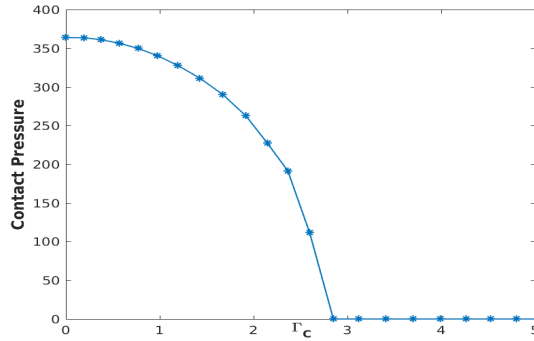


Figure 6: Deformed configuration and Von Mises effective stress for the Hertz problem

Figure 7: Normal Stress distribution on Γ_C for the Hertz problem

We summarize in Table 3 the (optimal) penalty parameters and the behavior of Algorithm 2 in terms of CPU time and number of iterations. For $m_c = 35$, the sampling procedure gives better results in terms of the number of iterations for the ADMM solver but the computational time is much greater. For $m_c = 69$ the number of iterations is almost equivalent for both \bar{r} and r^* but the computational time is again much greater for \bar{r} . We report in Table 4 the performances of Algorithm 2 using $r^* = 201.7825$ and $\bar{r} = 200$. We can notice that the performances of Algorithm 2 using both penalty parameters are almost equivalent.

m_c	35	69
r^*	137.1789	201.7825
Iter.	59	43
CPU Time	0.084	0.166
\bar{r}	200	200
Iter.	49	44
CPU Time	46.357	215.029

Table 3: \bar{r} obtained by sampling Versus r^* obtained by (4.13), for the Hertz problem.

Number of nodes on Ω/Γ_C	442/35	1692/69	6619/125	26181/229	104137/457	
$r^* = 201.7825$	Iter.	49	43	54	51	50
	CPU Time	0.029	0.097	0.592	4.485	27.109
$\bar{r} = 200$	Iter.	49	44	56	51	51
	CPU Time	0.025	0.083	0.597	4.483	27.560

Table 4: Performances of Algorithm 2 with $r^* = 201.7825$ and $\bar{r} = 200$ for the Hertz problem

7.1 Example 3: 3D problem

We now study the behavior of a 3D rectangular elastic body pressed onto a solid hemisphere, Figure 8 ([13, 9]). The elastic body occupies the domain $\Omega = (-0.5, 0.5) \times (-1, 1) \times (-0.5, 0.5)$ and the obstacle is a half-ball with radius $r = 0.5$ and center $(-0.3, 0, -1)$. The

contact surface is the lower surface $\Gamma_C = (-0.5, 0.5) \times (-1, 1) \times \{-0.5\}$. A displacement $u_D = (0, 0, -0.2)$ is prescribed on the upper surface $\Gamma_D = (-0.5, 0.5) \times (-1, 1) \times \{0.5\}$. The Young modulus is $E = 10^6$ and the Poisson ratio $\nu = 0.45$. Figure 8 shows geometry and the obstacle.

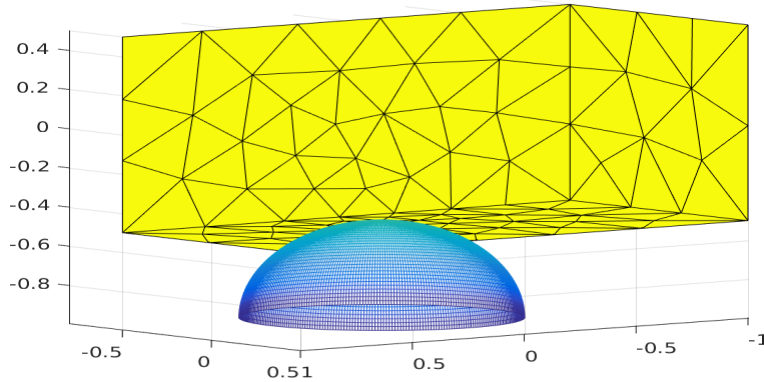


Figure 8: Geometry of the 3D obstacle problem

We first consider a non uniform mesh with 198 nodes, 773 tetrahedrons and 45 nodes on Γ_C . The deformed configuration obtained using ADMM Algorithm 2 is shown in Figure 9 and the normal (contact) stress distribution in Figure 10. For \bar{r} , we sample $(0.1 \cdot 10 \cdot E)$ with 200 points, uniformly spaced. We summarize in Table 5 the computed penalty parameters r^* and \bar{r} and the behavior of Algorithm 2 in terms of number of iterations and CPU time. Even though r^* is 20% larger than \bar{r} , the number of iterations required by the ADMM solver with both values is almost the same. We notice again that computing the penalty parameter with Algorithm 5 is about 100 times faster.

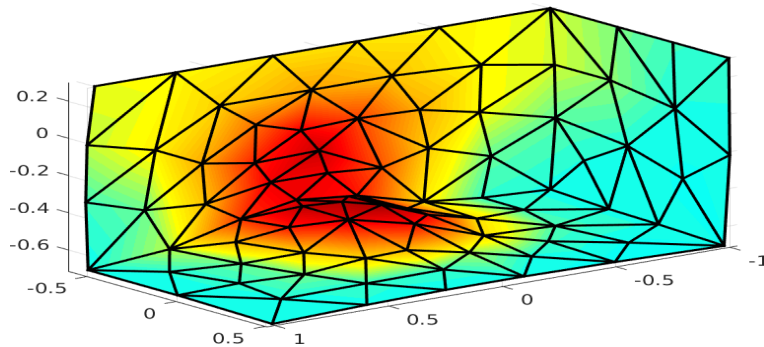
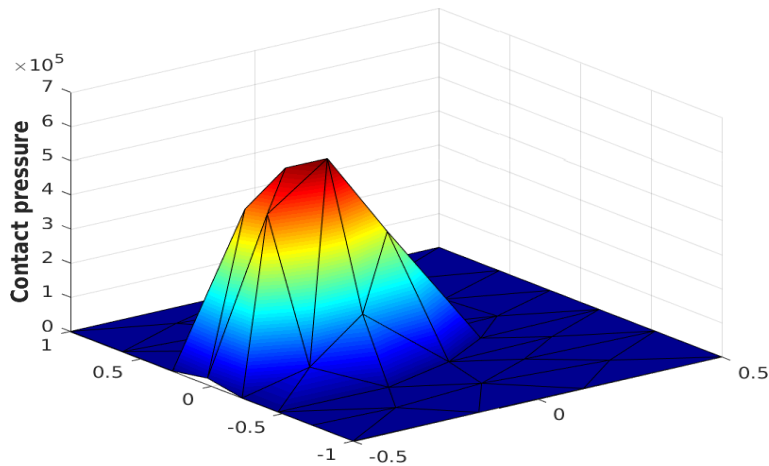


Figure 9: Deformed configuration and Von Mises effective stress for Example 3.

	Iter.	CPU Time
$r^* = 3.1026 \times 10^6$	28	0.083
$\bar{r} = 2.6000 \times 10^6$	28	8.035

Table 5: \bar{r} obtained by sampling Versus r^* obtained by (4.13) for the 3D obstacle problem

To study the behavior of Algorithm 2, the initial mesh of 198 nodes and 773 tetrahedrons is successively refined to produce meshes with 1289, 9245 and 69897 nodes and 6144, 49472 and 395776 tetrahedrons, respectively. The performances of Algorithm 2 are

Figure 10: Normal stress distribution on Γ_C

Number of nodes on Ω/Γ_C		198/45	1289/153	9245/603	69897/2341
$r^* = 3.1026 \times 10^6$	Iter.	28	30	31	32
	CPU Time	0.053	0.613	10.267	363.534
$\bar{r} = 2.6000 \times 10^6$	Iter.	26	28	33	30
	CPU Time	0.046	0.577	10.568	349.122

Table 6: Performances of Algorithm 2 with \bar{r} and r^* for the 3D obstacle problem.

reported in Table 6. As in the 2D case, the number of iterations is virtually independent of the mesh size, for both penalty parameters. We can notice that, for both penalty parameter, the difference of computational time for the largest problem is less than 5%. The performances of Algorithm 2 with r^* and \bar{r} are therefore comparable.

8 Conclusion

We have investigated the automatic parameter selection for the alternating direction method of multiplier applied to the unilateral contact problem. Numerical experiments have shown that the proposed procedure for the parameter selection is a good alternative to the standard sampling procedure.

References

- [1] BOYD S., PARIKH N., CHU E., PELEATO B. and ECKSTEIN J. Distributed optimization and statistical learning via alternating direction method of multipliers. *Foundation and Trends in Machine Learning*, 3:1–122, 2011.
- [2] FORTIN M. and GLOWINSKI R. *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems*. North-Holland, Amsterdam, 1983.
- [3] GLOWINSKI R. and LE TALLEC P. *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*. Studies in Applied Mathematics. SIAM, Philadelphia, 1989.

- [4] GLOWINSKI R. and MAROCCO A. Sur l'approximation par éléments finis d'ordre un, et la résolution par pénalisation–dualité, d'une classe de problèmes de Dirichlet non linéaires. *RAIRO Anal. Num.*, 9(2):41–76, 1975.
- [5] HORN R.A. and JOHNSON C.R. *Matrix Analysis*. Cambridge University Press, 2013.
- [6] HÜEBER S., STADLER G. and WOHLMUTH B. I. A primal-dual active set algorithm for three-dimensional contact problems with Coulomb friction. *SIAM J. Sci. Comput.*, 30:572–596, 2008.
- [7] ITO K. and KUNISCH K. Semi-smooth Newton methods for the Signorini problem. *Applications of Mathematics*, 53:455–568, 2008.
- [8] KIKUCHI N. and ODEN J.T. *Contact problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. Studies in Applied Mathematics. SIAM, Philadelphia, 1988.
- [9] KOKO J. Uzawa block relaxation for the unilateral contact problem. *J. Comput. Appl. Math.*, 235:2343–2356, 2011.
- [10] KOKO J. A MATLAB mesh generator for the two-dimensional finite element method. *Appl. Math. Comput.*, 250:650–664, 2015.
- [11] KOKO J. Fast MATLAB assembly of fem matrices in 2d and 3d using cell array approach. *Int. J. Model. Simul. Sci. Comput.*, 7, 2016.
- [12] KOKO J. and JEHAN-BESSON S. An augmented lagrangian method for TV_g+L^1 -norm minimization. *J. Math. Imaging Vis.*, 38:182–196, 2010.
- [13] STADLER G. Path-following and augmented Lagrangian methods for contact problems in linear elasticity. *J. Comput. Appl. Math.*, 203:533–547, 2007.
- [14] WRIGGERS P. *Computational Contact Mechanics*. John Wiley & Sons, 2002.
- [15] YANG Y. and ZHANG Y. Alternating direction algorithms for l_1 -problems in compressive sensing. *SIAM J. Sci. Comput.*, 33:250–278, 2011.