



ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Mathematics and Computers in Simulation xxx (xxxx) xxx


 MATHEMATICS
AND
COMPUTERS
IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Dual strategies for solving the Stokes problem with stick–slip boundary conditions in 3D

 Jaroslav Haslinger^a, Radek Kučera^{a,b,*}, Taoufik Sassi^c, Václav Šátek^{d,b}
^a Faculty of Mechanical Engineering, VŠB-TUO, 17. listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic

^b IT4Innovations, VŠB-TUO, 17. listopadu 2172/15, 708 33 Ostrava-Poruba, Czech Republic

^c Laboratory of Mathematics Nicolas Oresme, CNRS UMR 6139, University of Caen - Normandy, 14032 Caen Cedex, France

^d Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, 612 66 Brno, Czech Republic

Received 3 March 2020; received in revised form 7 December 2020; accepted 15 December 2020

Available online xxxx

Abstract

The paper deals with the numerical realization of the 3D Stokes flow subject to threshold slip boundary conditions. The weak velocity–pressure formulation leads to an inequality type problem that is approximated by a mixed finite element method. The resulting algebraic system is non-smooth. Besides the pressure, three additional Lagrange multipliers are introduced: the discrete normal stress releasing the impermeability condition and two discrete shear stresses regularizing the non-smooth slip term. Eliminating the discrete velocity component we obtain the minimization problem for the smooth functional, expressed in terms of the pressure, the normal, and the shear stresses. This problem is solved either by a path following variant of the interior point method or by the semi-smooth Newton method. Numerical scalability is illustrated by computational experiments. © 2020 International Association for Mathematics and Computers in Simulation (IMACS). Published by Elsevier B.V. All rights reserved.

Keywords: Stokes problem; Stick–slip boundary conditions; Interior-point method; Semi-smooth Newton method

1. Introduction

The no-slip condition is the standard boundary condition in fluid flow models. It characterizes the stick of a fluid on a solid wall, i.e., the tangential velocity on the boundary of a computational domain vanishes. However, a slip of a fluid is observed along a wall in many real situations (e.g. water flow along hydrophobic surfaces). The simplest stick–slip condition is the Navier one [17]:

$$\boldsymbol{\sigma}_t = -\kappa \mathbf{u}_t, \quad (1)$$

where $\boldsymbol{\sigma}_t$ is the shear stress, \mathbf{u}_t is the tangential component of the velocity \mathbf{u} , and $\kappa \geq 0$ is an adhesive coefficient. One can see from (1) that a slip appears whenever $\boldsymbol{\sigma}_t \neq \mathbf{0}$. The stick–slip conditions introduced by Fujita [6] consider their threshold character using the slip bound function $g \geq 0$. They read as follows:

$$\left. \begin{aligned} \|\boldsymbol{\sigma}_t\| &\leq g, \\ \|\boldsymbol{\sigma}_t\| < g &\Rightarrow \mathbf{u}_t = \mathbf{0}, \\ \|\boldsymbol{\sigma}_t\| = g &\Rightarrow \exists c \geq 0 : \mathbf{u}_t = -c\boldsymbol{\sigma}_t. \end{aligned} \right\} \quad (2)$$

* Corresponding author at: IT4Innovations, VŠB-TUO, 17. listopadu 2172/15, 708 33 Ostrava-Poruba, Czech Republic.

E-mail address: radek.kucera@vsb.cz (R. Kučera).

<https://doi.org/10.1016/j.matcom.2020.12.015>

0378-4754/© 2020 International Association for Mathematics and Computers in Simulation (IMACS). Published by Elsevier B.V. All rights reserved.

Hence, the slip may occur only if the bound g is attained and the tangential velocity and the shear stress have the opposite direction. This condition is well-known in contact problems of solid mechanics as the Tresca friction law [7].

In this paper we combine (1) and (2) in one boundary condition formulated for the Stokes flow model. For the mathematical analysis of this problem we refer to [8,19]. Computational experiments based on an alternating direction method can be found in [4]. Another way of solving non-smooth problems using the primal–dual set strategy which is associated to a limit case of the semi-smooth Newton (SSN) method has been introduced and analysed in [9]. Our aim is to develop efficient algorithms for solving large-scale algebraic systems arising from an appropriate finite element discretization of 3D problems that were successful in 2D cases. In particular, we extend the path-following (PF) variant of the interior point method [13] and the SSN method [16]. Note that the extensions from 2D to 3D are not straightforward, since the norm appearing in (2) is represented by the absolute value in 2D while by the Euclidean norm in \mathbb{R}^2 for 3D problems. This fact changes the structure of the resulting (saddle-point or dual) algebraic problems since the simple bounds in 2D are replaced by the separable spherical constraints in 3D. The PF algorithm is an appropriate modification of that one proposed in [14], while the implementation of the SSN method is based on similar ideas as in [15]. Note that original versions of these algorithms were developed and successfully tested in context of 3D contact problems of solid mechanics. In contrast to [15], it is not necessary to symmetrize the (generalized) Jacobian matrices in the SSN algorithm if the adhesive coefficient κ is positive. A considerable difference between problems of solid and flow mechanics is the presence of the incompressibility and impermeability conditions in the latter. Consequently, the dual algebraic formulations involve a relatively small number of constrained unknowns which excludes the efficient use of some types of algorithms. For this reason we do not use the active-set minimization algorithms [12]. It has been observed that for our type of 2D problems [13] they are less efficient, especially for large-scale computations. This property is much more significant in 3D.

The rest of the paper is organized as follows. In Section 2 we introduce different formulations of our problem. Section 3 deals with the finite element approximation based on the P1-bubble/P1 finite element pair. The respective algebraic problem is expressed in the saddle-point and the dual form. In Section 4 the algorithms are presented. The PF algorithm is assembled as the solver of the dual problem with the separable spherical constraints. The SSN algorithm solves the projective form of the saddle-point problem in which the constraints are expressed by the projections on circles in \mathbb{R}^2 . Section 5 summarizes results of numerical experiments. Finally, Section 6 gives several concluding remarks.

2. Formulation

Let $\Omega \subset \mathbb{R}^3$ be a bounded domain with a sufficiently smooth boundary $\partial\Omega$ that is split into three disjoint parts: $\partial\Omega = \overline{\gamma}_D \cup \overline{\gamma}_N \cup \overline{\gamma}_S$, $\gamma_D \neq \emptyset$, $\gamma_S \neq \emptyset$. We consider the viscous flow of an incompressible Newtonian fluid modelled by the Stokes system in Ω with the Dirichlet and Neumann boundary conditions on γ_D and γ_N , respectively, and with the impermeability and the stick–slip boundary conditions on γ_S :

$$\left. \begin{aligned} -2\nu \operatorname{div} D(\mathbf{u}) + \nabla p &= \mathbf{f} && \text{in } \Omega, \\ \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{0} && \text{on } \gamma_D, \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}_N && \text{on } \gamma_N, \\ u_n &= 0 && \text{on } \gamma_S, \\ \|\boldsymbol{\sigma}_t + \kappa \mathbf{u}_t\| &\leq g && \text{on } \gamma_S, \\ \boldsymbol{\sigma}_t \cdot \mathbf{u}_t + g \|\mathbf{u}_t\| + \kappa (\mathbf{u}_t \cdot \mathbf{u}_t) &= 0 && \text{on } \gamma_S. \end{aligned} \right\} \quad (3)$$

The unknowns \mathbf{u} , p represent the velocity and the pressure field in Ω , respectively, \mathbf{f} is a force acting on the fluid, $\nu > 0$ is the kinematic viscosity, and $\boldsymbol{\sigma}_N$ is a prescribed stress on γ_N . Further $\boldsymbol{\sigma} = 2\nu D(\mathbf{u})\mathbf{n} - p\mathbf{n}$ is the stress on $\partial\Omega$ and $D(\mathbf{u}) = 1/2(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the symmetric part of the gradient of \mathbf{u} . The unit outer normal and tangential vectors at $\mathbf{x} \in \partial\Omega$ will be denoted by $\mathbf{n} := \mathbf{n}(\mathbf{x})$ and $\mathbf{t}_1 := \mathbf{t}_1(\mathbf{x})$, $\mathbf{t}_2 := \mathbf{t}_2(\mathbf{x})$, respectively, and we will assume that the triplet $\{\mathbf{n}, \mathbf{t}_1, \mathbf{t}_2\}$ is an orthonormal basis at all $\mathbf{x} \in \partial\Omega$. The normal, tangential components of the velocity \mathbf{u} along $\partial\Omega$ are denoted by $u_n = \mathbf{u} \cdot \mathbf{n}$, and $\mathbf{u}_t = (\mathbf{u} \cdot \mathbf{t}_1, \mathbf{u} \cdot \mathbf{t}_2)$, respectively, and analogously for the stress $\boldsymbol{\sigma}$. Finally, $g \geq 0$, $\kappa \geq 0$ is the slip bound, and the adhesive function, respectively, on γ_S . The symbol $\|\cdot\|$ in (3) stands for the Euclidean norm in \mathbb{R}^2 .

The stick–slip boundary conditions (3)_{6,7} become the Navier slip law if $g = 0, \kappa > 0$, and the Tresca stick–slip law if $g > 0, \kappa = 0$. Introducing the new variable $\tilde{\sigma}_t = \sigma_t + \kappa \mathbf{u}_t$, (3)_{6,7} can be written as follows:

$$\left. \begin{aligned} \|\tilde{\sigma}_t\| &\leq g \\ \tilde{\sigma}_t \cdot \mathbf{u}_t + g \|\mathbf{u}_t\| &= 0 \end{aligned} \right\} \text{on } \gamma_S,$$

or, equivalently,

$$\mathbf{u}_t(\mathbf{x}) \neq \mathbf{0} \Rightarrow \|\tilde{\sigma}_t(\mathbf{x})\| = g(\mathbf{x}) \ \& \ \exists c := c(\mathbf{x}) \geq 0 : \tilde{\sigma}_t(\mathbf{x}) = -c \mathbf{u}_t(\mathbf{x}), \quad \left. \vphantom{\|\tilde{\sigma}_t(\mathbf{x})\|} \right\} \mathbf{x} \in \gamma_S. \quad (4)$$

Thus, (3)_{6,7} is formally expressed as the Tresca stick–slip law also for $\kappa > 0$. Consequently, algorithms developed for this type of the stick–slip law can be used, as well.

To give the weak velocity–pressure formulation of (3) we introduce the following spaces and forms:

$$\begin{aligned} V(\Omega) &= \{\mathbf{v} \in (H^1(\Omega))^3 : \mathbf{v} = \mathbf{0} \text{ on } \gamma_D, v_n = 0 \text{ on } \gamma_S\}, \\ V_0(\Omega) &= \{\mathbf{v} \in V(\Omega) : \nabla \cdot \mathbf{v} = 0 \text{ in } \Omega\} \end{aligned}$$

and

$$\begin{aligned} a(\mathbf{w}, \mathbf{v}) &= \nu \int_{\Omega} D(\mathbf{w}) : D(\mathbf{v}) \, d\mathbf{x}, \quad j_{g,\kappa}(\mathbf{w}, \mathbf{v}) = \int_{\gamma_S} (g \|\mathbf{v}_t\| + \kappa \mathbf{w}_t \cdot \mathbf{v}_t) \, ds, \\ b(\mathbf{v}, q) &= - \int_{\Omega} q(\nabla \cdot \mathbf{v}) \, d\mathbf{x}, \quad l(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\gamma_N} \sigma_N \cdot \mathbf{v} \, ds, \end{aligned}$$

where $\mathbf{w}, \mathbf{v} \in (H^1(\Omega))^3$, $q \in L^2(\Omega)$, and $D(\mathbf{w}) : D(\mathbf{v}) = \sum_{i=1}^3 \sum_{j=1}^3 D_{ij}(\mathbf{w}) D_{ij}(\mathbf{v})$, $D(\mathbf{v}) = (D_{ij}(\mathbf{v}))_{i,j=1}^3$.

The weak velocity–pressure formulation of (3) is defined by:

$$\left. \begin{aligned} \text{Find } (\mathbf{u}, p) &\in V(\Omega) \times L^2(\Omega) \text{ such that } \forall (\mathbf{v}, q) \in V(\Omega) \times L^2(\Omega) \\ a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(\mathbf{v} - \mathbf{u}, p) + j_{g,\kappa}(\mathbf{u}, \mathbf{v}) - j_{g,\kappa}(\mathbf{u}, \mathbf{u}) &\geq l(\mathbf{v} - \mathbf{u}), \\ b(\mathbf{u}, q) &= 0. \end{aligned} \right\} \quad (5)$$

The existence and uniqueness of a weak solution follow from the next theorem.

Theorem 1 ([3,5]). *Let $\mathbf{f} \in (L^2(\Omega))^3$, $\sigma_N \in (L^2(\gamma_N))^3$, and $g, \kappa \in L^\infty(\gamma_S)$, $g \geq 0, \kappa \geq 0$. Then the solution (\mathbf{u}, p) to (5) exists and the velocity component \mathbf{u} is unique. If $\gamma_N \neq \emptyset$, then the pressure component p is unique as well, otherwise it is defined up to an additive constant.*

Remark 1. To guarantee uniqueness of p in the case when $\gamma_N = \emptyset$ it is sufficient to replace the space $L^2(\Omega)$ by its subspace $L_0^2(\Omega)$ of functions whose integral mean value in Ω is equal to zero. Computational aspects are discussed in more detail in Section 5, Example 3.

We will also use (5) with the modified bilinear form a . We set:

$$a_\kappa(\mathbf{w}, \mathbf{v}) = a(\mathbf{w}, \mathbf{v}) + \int_{\gamma_S} \kappa \mathbf{w}_t \cdot \mathbf{v}_t \, ds, \quad j_g(\mathbf{v}) = \int_{\gamma_S} g \|\mathbf{v}_t\| \, ds.$$

Then the weak velocity–pressure formulation (5) can be written in the following equivalent form:

$$\left. \begin{aligned} \text{Find } (\mathbf{u}, p) &\in V(\Omega) \times L^2(\Omega) \text{ such that } \forall (\mathbf{v}, q) \in V(\Omega) \times L^2(\Omega) \\ a_\kappa(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(\mathbf{v} - \mathbf{u}, p) + j_g(\mathbf{v}) - j_g(\mathbf{u}) &\geq l(\mathbf{v} - \mathbf{u}), \\ b(\mathbf{u}, q) &= 0. \end{aligned} \right\} \quad (6)$$

It is easy to see that the first component \mathbf{u} solves the following velocity formulation of (3):

$$\mathbf{u} \in V_0(\Omega) : a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + j_{g,\kappa}(\mathbf{u}, \mathbf{v}) - j_{g,\kappa}(\mathbf{u}, \mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in V_0(\Omega), \quad (7)$$

or, equivalently,

$$\mathbf{u} = \operatorname{argmin}\{J(\mathbf{v}) := \frac{1}{2} \nu \int_{\Omega} D(\mathbf{v}) : D(\mathbf{v}) \, d\mathbf{x} + j_{g,\kappa}(\mathbf{v}, \mathbf{v}) - l(\mathbf{v}), \mathbf{v} \in V_0(\Omega)\}. \quad (8)$$

For numerical realization of (3) we use the so-called *four-field formulation*, which is based on a dualization of the impermeability condition $v_n = 0$ on γ_S and a regularization of the non-smooth slip term j_g . To this end we shall need the function space X_n on γ_S defined by

$$X_n = \{\varphi \in L^2(\gamma_S) : \exists \mathbf{v} \in W(\Omega) \text{ such that } \varphi = v_n \text{ on } \gamma_S\}$$

and its dual X'_n , where

$$W(\Omega) = \{\mathbf{v} \in (H^1(\Omega))^3 : \mathbf{v} = \mathbf{0} \text{ on } \gamma_D\}.$$

The duality pairing between X_n and X'_n will be denoted by $\langle \cdot, \cdot \rangle$. The slip term j_g will be regularized using that

$$j_g(\mathbf{v}) = \int_{\gamma_S} g \|\mathbf{v}_t\| ds = \sup_{\boldsymbol{\mu}_t \in K(g)} \int_{\gamma_S} \boldsymbol{\mu}_t \cdot \mathbf{v}_t ds,$$

where

$$K(g) = \{\boldsymbol{\mu}_t \in (L^\infty(\partial\Omega))^2 : \|\boldsymbol{\mu}_t\| \leq g \text{ a.e. on } \gamma_S\}.$$

The four-field formulation of (3) reads as follows:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, p, \lambda_n, \lambda_t) \in W(\Omega) \times L^2(\Omega) \times X'_n \times K(g) \text{ such that} \\ & a_\kappa(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + \langle \lambda_n, \mathbf{v}_n \rangle + \int_{\gamma_S} \lambda_t \cdot \mathbf{v}_t ds = l(\mathbf{v}) \quad \forall \mathbf{v} \in W(\Omega), \\ & b(\mathbf{u}, q) = 0 \quad \forall q \in L^2(\Omega), \\ & \langle \mu_n, u_n \rangle = 0 \quad \forall \mu_n \in X'_n, \\ & \int_{\gamma_S} (\boldsymbol{\mu}_t - \lambda_t) \cdot \mathbf{u}_t ds \leq 0 \quad \forall \boldsymbol{\mu}_t \in K(g). \end{aligned} \right\} \quad (9)$$

Such formulation has been studied in [8]. It was proven that (9) has a solution, the couple (\mathbf{u}, p) solves (5) and $\lambda_n = -\sigma_n$, $\lambda_t = -\tilde{\sigma}_t$ on γ_S . It is readily seen that (9) represents the necessary and sufficient conditions for $(\mathbf{u}, p, \lambda_n, \lambda_t)$ to be a saddle-point of the Lagrangian L on $W(\Omega) \times L^2(\Omega) \times X'_n \times K(g)$, where

$$L(\mathbf{v}, q, \mu_n, \boldsymbol{\mu}_t) = \frac{1}{2} a_\kappa(\mathbf{v}, \mathbf{v}) + b(\mathbf{v}, q) + \langle \mu_n, \mathbf{v}_n \rangle + \int_{\gamma_S} \boldsymbol{\mu}_t \cdot \mathbf{v}_t ds - l(\mathbf{v}).$$

3. Finite element approximation and algebraic problems

Here and in what follows we shall suppose that $\gamma_N \neq \emptyset$ so that p is unique. To discretize (5) we use the mixed finite element method. Let $V_h \subset W_h$, Q_h be finite element approximations of $V(\Omega)$, $W(\Omega)$, and $L^2(\Omega)$, respectively, chosen in such a way that the bilinear form $b : W_h \times Q_h \rightarrow \mathbb{R}$ satisfies the following *inf-sup* condition: there exists a constant $\beta > 0$ which does not depend on the discretization parameter h such that

$$\sup_{\mathbf{v}_h \in W_h} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{1,\Omega}} \geq \beta \|q_h\|_{0,\Omega}$$

holds for every $q_h \in Q_h$.

The discretization of (5) reads as follows:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}_h, p_h) \in V_h \times Q_h \text{ such that } \forall (\mathbf{v}_h, q_h) \in V_h \times Q_h \\ & a(\mathbf{u}_h, \mathbf{v}_h - \mathbf{u}_h) + b(\mathbf{v}_h - \mathbf{u}_h, p_h) + j_{g,\kappa}(\mathbf{u}_h, \mathbf{v}_h) - j_{g,\kappa}(\mathbf{u}_h, \mathbf{u}_h) \geq l(\mathbf{v}_h - \mathbf{u}_h), \\ & b(\mathbf{u}_h, q_h) = 0. \end{aligned} \right\} \quad (10)$$

Replacing a and $j_{g,\kappa}$ in (10) by a_κ and j_g , respectively, we get the discretization of (6).

For the discretization and in computations we use the P1-bubble/P1 finite element pair on a regular tetrahedral partition \mathcal{T}_h of $\bar{\Omega}$. On \mathcal{T}_h we define the finite element spaces:

$$\begin{aligned} V_h &= \{\mathbf{v}_h \in V(\Omega) : \mathbf{v}_h|_T \in (P_1(T))^3 \oplus (B(T))^3 \quad \forall T \in \mathcal{T}_h\}, \\ Q_h &= \{q_h \in C(\bar{\Omega}) : q_h|_T \in P_1(T) \quad \forall T \in \mathcal{T}_h\}, \end{aligned}$$

where $P_1(T)$, $B(T)$ are the spaces of polynomials of degree one and of bubble functions of degree four on $T \in \mathcal{T}_h$, respectively.

The algebraic counterpart of (10) reads as follows:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^{n_p} \text{ such that for all } \mathbf{v} \in \mathbb{V} \\ & \mathbf{u}^T \mathbf{A}(\mathbf{v} - \mathbf{u}) + (\mathbf{v} - \mathbf{u})^T \mathbf{B}^T \mathbf{p} + j_{g,\kappa}^h(\mathbf{u}, \mathbf{v}) - j_{g,\kappa}^h(\mathbf{u}, \mathbf{u}) \geq \mathbf{b}^T(\mathbf{v} - \mathbf{u}), \\ & \mathbf{B}\mathbf{u} = \mathbf{0}, \end{aligned} \right\} \quad (11)$$

where

$$\begin{aligned} \mathbb{V} &= \{\mathbf{v} \in \mathbb{R}^{3(n_u+n_q)} : \mathbf{N}\mathbf{v} = \mathbf{0}\}, \\ j_{g,\kappa}^h(\mathbf{w}, \mathbf{v}) &= \sum_{i \in \mathcal{N}} (g_i \|\mathbf{v}_{t,i}\| + \kappa_i \mathbf{w}_{t,i}^T \mathbf{v}_{t,i}). \end{aligned}$$

Here, n_p , n_u stand for the number of the finite element nodes of \mathcal{T}_h in $\bar{\Omega}$, and $\bar{\Omega} \setminus \bar{\gamma}_D$, respectively, while n_q denotes the number of the tetrahedras in \mathcal{T}_h . Further, $\mathbf{A} \in \mathbb{R}^{3(n_u+n_q) \times 3(n_u+n_q)}$ is a symmetric, positive definite diffusion matrix, $\mathbf{B} \in \mathbb{R}^{n_p \times 3(n_u+n_q)}$ is a full row-rank divergence matrix, $\mathbf{N}, \mathbf{T}_1, \mathbf{T}_2 \in \mathbb{R}^{n_s \times 3(n_u+n_q)}$ are full row-rank matrices whose rows are defined by the normal \mathbf{n} and the tangential vectors $\mathbf{t}_1, \mathbf{t}_2$, respectively, at the nodes belonging to $\bar{\gamma}_S \setminus \bar{\gamma}_D$, n_s is the number of the nodes of \mathcal{T}_h on $\bar{\gamma}_S \setminus \bar{\gamma}_D$, and $\mathbf{b} \in \mathbb{R}^{3(n_u+n_q)}$ is a vector of nodal forces. Finally, $\mathbf{v}_{t,i} = ((\mathbf{T}_1 \mathbf{v})_i, (\mathbf{T}_2 \mathbf{v})_i)^T \in \mathbb{R}^2$, $i \in \mathcal{N} := \{1, \dots, n_s\}$.

The algebraic version $j_{g,\kappa}^h$ of $j_{g,\kappa}$ introduced above is obtained by using the following quadrature formula on a triangle $\tau \in \mathcal{T}_h|_{\bar{\gamma}_S}$:

$$\int_{\tau} g \|\mathbf{v}_t\| ds \approx |\tau| g(\mathbf{x}_\tau) \frac{1}{3} (\|\mathbf{v}_{t,i}\| + \|\mathbf{v}_{t,j}\| + \|\mathbf{v}_{t,k}\|) \quad (12)$$

assuming that $g \in C(\bar{\gamma}_S)$, where $|\tau| = \text{meas } \tau$, \mathbf{x}_τ is the centre of gravity of τ and $\mathbf{v}_{t,i}, \mathbf{v}_{t,j}, \mathbf{v}_{t,k}$ are the values of \mathbf{v}_t at the vertices $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ of τ . Summing up (12) over all $\tau \in \mathcal{T}_h|_{\bar{\gamma}_S}$ we see that

$$\int_{\bar{\gamma}_S} g \|\mathbf{v}_t\| ds \approx \sum_{i \in \mathcal{N}} g_i \|\mathbf{v}_{t,i}\| \quad \text{with} \quad g_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{3} g(\mathbf{x}_{\tau_j^i}), \quad i \in \mathcal{N},$$

where n_i is the number of the triangles $\tau_j^i \in \mathcal{T}_h|_{\bar{\gamma}_S}$ sharing $\mathbf{x}_i \in \bar{\gamma}_S \setminus \bar{\gamma}_D$ as the common vertex. In the same way we obtain:

$$\int_{\bar{\gamma}_S} \kappa \mathbf{w}_t \cdot \mathbf{v}_t ds \approx \sum_{i \in \mathcal{N}} \kappa_i \mathbf{w}_{t,i}^T \mathbf{v}_{t,i} \quad \text{with} \quad \kappa_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{3} \kappa(\mathbf{x}_{\tau_j^i}), \quad i \in \mathcal{N}. \quad (13)$$

Since

$$j_{g,\kappa}^h(\mathbf{w}, \mathbf{v}) = \sum_{i \in \mathcal{N}} g_i \|\mathbf{v}_{t,i}\| + \mathbf{w}^T (\mathbf{T}_1^T \mathbf{D}_\kappa \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{D}_\kappa \mathbf{T}_2) \mathbf{v}$$

with $\mathbf{D}_\kappa = \text{diag}\{\kappa_1, \dots, \kappa_{n_s}\}$, one can write (11) in the following equivalent form:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^{n_p} \text{ such that for all } \mathbf{v} \in \mathbb{V} \\ & \mathbf{u}^T \mathbf{A}_\kappa(\mathbf{v} - \mathbf{u}) + (\mathbf{v} - \mathbf{u})^T \mathbf{B}^T \mathbf{p} + j_g^h(\mathbf{v}) - j_g^h(\mathbf{u}) \geq \mathbf{b}^T(\mathbf{v} - \mathbf{u}), \\ & \mathbf{B}\mathbf{u} = \mathbf{0}, \end{aligned} \right\} \quad (14)$$

where

$$\mathbf{A}_\kappa = \mathbf{A} + \mathbf{T}_1^T \mathbf{D}_\kappa \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{D}_\kappa \mathbf{T}_2 \quad (15)$$

and

$$j_g^h(\mathbf{v}) = \sum_{i \in \mathcal{N}} g_i \|\mathbf{v}_{t,i}\|.$$

Problem (14) is the algebraic counterpart of (6).

It is easy to show that (14) is equivalent to the discrete velocity formulation:

$$\left. \begin{array}{l} \text{Find } \mathbf{u} \in \mathbb{V}_B \text{ such that} \\ \mathcal{J}(\mathbf{u}) \leq \mathcal{J}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{V}_B, \end{array} \right\} \quad (16)$$

where

$$\begin{aligned} \mathbb{V}_B &= \{\mathbf{v} \in \mathbb{R}^{3(n_u+n_q)} : \mathbf{N}\mathbf{v} = \mathbf{0}, \mathbf{B}\mathbf{v} = \mathbf{0}\}, \\ \mathcal{J}(\mathbf{v}) &= \frac{1}{2} \mathbf{v}^T \mathbf{A}_\kappa \mathbf{v} - \mathbf{v}^T \mathbf{b} + j_g^h(\mathbf{v}). \end{aligned}$$

Formulation (16) is not suitable for direct computations, as the constraints in \mathbb{V}_B can be hardly handled for large-scale problems. Moreover, the function \mathcal{J} is non-differentiable due to the term j_g^h . To overcome these difficulties, we will use the dual formulation of (16) and derive the discrete counterpart of (9).

We introduce four Lagrange multipliers: $\boldsymbol{\mu}_{t_1}, \boldsymbol{\mu}_{t_2} \in \mathbb{R}^{n_s}$ to regularize j_g^h and $\boldsymbol{\mu}_n \in \mathbb{R}^{n_s}$, $\mathbf{q} \in \mathbb{R}^{n_p}$ to release the discrete impermeability and incompressibility conditions, respectively. The set of the Lagrange multipliers $\boldsymbol{\mu} = (\boldsymbol{\mu}_{t_1}^T, \boldsymbol{\mu}_{t_2}^T, \boldsymbol{\mu}_n^T, \mathbf{q}^T)^T$ is defined by:

$$\begin{aligned} \mathbb{X} &= \mathbb{X}_t \times \mathbb{R}^{n_s+n_p}, \\ \mathbb{X}_t &= \{(\boldsymbol{\mu}_{t_1}^T, \boldsymbol{\mu}_{t_2}^T)^T \in \mathbb{R}^{2n_s} : \|\boldsymbol{\mu}_{t,i}\| \leq g_i, i \in \mathcal{N}\}, \end{aligned}$$

where $\boldsymbol{\mu}_{t,i} = (\mu_{t_1,i}, \mu_{t_2,i})^T \in \mathbb{R}^2$ and $\mu_{t_k,i}$ is the i th component of $\boldsymbol{\mu}_{t_k} \in \mathbb{R}^{n_s}$, $k = 1, 2$. The term j_g^h can be written as follows:

$$j_g^h(\mathbf{v}) = \sum_{i \in \mathcal{N}} \max_{\substack{\boldsymbol{\mu}_{t,i} \in \mathbb{R}^2 \\ \|\boldsymbol{\mu}_{t,i}\| \leq g_i}} \boldsymbol{\mu}_{t,i}^T \mathbf{v}_{t,i} = \max_{\boldsymbol{\mu} \in \mathbb{X}_t} \sum_{i \in \mathcal{N}} \boldsymbol{\mu}_{t,i}^T \mathbf{v}_{t,i}.$$

Thus

$$\min_{\mathbf{v} \in \mathbb{V}_B} \mathcal{J}(\mathbf{v}) = \min_{\mathbf{v} \in \mathbb{R}^{3(n_u+n_q)}} \max_{\boldsymbol{\mu} \in \mathbb{X}} \mathcal{L}^\#(\mathbf{v}, \boldsymbol{\mu}),$$

where the Lagrangian $\mathcal{L}^\# : \mathbb{R}^{3(n_u+n_q)} \times \mathbb{X} \rightarrow \mathbb{R}$ associated with the problem (16) is defined for $(\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{3(n_u+n_q)} \times \mathbb{X}$ by:

$$\mathcal{L}^\#(\mathbf{v}, \boldsymbol{\mu}) = \frac{1}{2} \mathbf{v}^T \mathbf{A}_\kappa \mathbf{v} - \mathbf{v}^T \mathbf{b} + \boldsymbol{\mu}_{t_1}^T \mathbf{T}_1 \mathbf{v} + \boldsymbol{\mu}_{t_2}^T \mathbf{T}_2 \mathbf{v} + \boldsymbol{\mu}_n^T \mathbf{N} \mathbf{v} + \mathbf{q}^T \mathbf{B} \mathbf{v}.$$

Until now, the velocity vectors $\mathbf{v} \in \mathbb{R}^{3(n_u+n_q)}$ incorporate $3n_q$ bubble components. These components are usually eliminated before the computational process. This elimination is easy in linear problems, when it is done in a linear saddle-point system [11]. In our case, we perform this elimination in a saddle-point formulation for $\mathcal{L}^\#$, which leads to the reduced Lagrangian $\mathcal{L} : \mathbb{R}^{3n_u} \times \mathbb{X} \rightarrow \mathbb{R}$ defined by:

$$\begin{aligned} \mathcal{L}(\mathbf{v}, \boldsymbol{\mu}) &= \frac{1}{2} \mathbf{v}^T \mathbf{A}_\kappa \mathbf{v} - \mathbf{v}^T \mathbf{b} + \boldsymbol{\mu}_{t_1}^T \mathbf{T}_1 \mathbf{v} + \boldsymbol{\mu}_{t_2}^T \mathbf{T}_2 \mathbf{v} + \boldsymbol{\mu}_n^T \mathbf{N} \mathbf{v} + \mathbf{q}^T \mathbf{B} \mathbf{v} \\ &\quad - \frac{1}{2} \mathbf{q}^T \mathbf{E} \mathbf{q} - \mathbf{c}^T \mathbf{q}, \quad (\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{3n_u} \times \mathbb{X}. \end{aligned}$$

To simplify notation here and in what follows, we use the same symbols for the corresponding matrices and vectors before and after the elimination of the bubble components. The dimensions of the reduced matrices are: $\mathbf{A}_\kappa \in \mathbb{R}^{3n_u \times 3n_u}$, $\mathbf{T}_1, \mathbf{T}_2, \mathbf{N} \in \mathbb{R}^{n_s \times 3n_u}$, $\mathbf{B} \in \mathbb{R}^{n_p \times 3n_u}$ and $\mathbf{b} \in \mathbb{R}^{3n_u}$. Note that these matrices exhibit the same properties as before the elimination, especially, the expression (15) remains valid with $\mathbf{A} \in \mathbb{R}^{3n_u \times 3n_u}$ being symmetric, positive definite. The presence of the symmetric, positive semidefinite matrix $\mathbf{E} \in \mathbb{R}^{n_p \times n_p}$ and of the vector $\mathbf{c} \in \mathbb{R}^{n_p}$ are due to this elimination.

The saddle-point formulation of (16) reads as follows:

$$\left. \begin{array}{l} \text{Find } (\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{3n_u} \times \mathbb{X} \text{ such that} \\ \mathcal{L}(\mathbf{u}, \boldsymbol{\mu}) \leq \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) \leq \mathcal{L}(\mathbf{v}, \boldsymbol{\lambda}) \quad \forall (\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{3n_u} \times \mathbb{X}, \end{array} \right\} \quad (17)$$

or, equivalently,

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{3n_u} \times \mathbb{X} \text{ such that} \\ & \mathbf{A}_\kappa \mathbf{u} + \mathbf{T}_1^T \boldsymbol{\lambda}_{t_1} + \mathbf{T}_2^T \boldsymbol{\lambda}_{t_2} + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{B}^T \mathbf{p} - \mathbf{b} = \mathbf{0}, \\ & \mathbf{B} \mathbf{u} - \mathbf{E} \mathbf{p} - \mathbf{c} = \mathbf{0}, \\ & \mathbf{N} \mathbf{u} = \mathbf{0}, \\ & \|\boldsymbol{\lambda}_{t,i}\| \leq g_i, \\ & \mathbf{u}_{t,i} \neq \mathbf{0} \Rightarrow \|\boldsymbol{\lambda}_{t,i}\| = g_i \ \& \ \exists c_i \geq 0 : \boldsymbol{\lambda}_{t,i} = c_i \mathbf{u}_{t,i}, \end{aligned} \right\} i \in \mathcal{N}, \quad (18)$$

where $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{t_1}^T, \boldsymbol{\lambda}_{t_2}^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T$ and $(\boldsymbol{\lambda}_{t_1}, \boldsymbol{\lambda}_{t_2}), \boldsymbol{\lambda}_n, \mathbf{p}$ are the discrete counterparts of $-\tilde{\sigma}_t, -\sigma_n$, and p , respectively.

Now we eliminate the unknown \mathbf{u} . Denoting $\mathbf{C} = (\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{N}^T, \mathbf{B}^T)^T$, the first equality in (18) yields $\mathbf{u} = \mathbf{A}_\kappa^{-1}(\mathbf{b} - \mathbf{C}^T \boldsymbol{\lambda})$. Inserting \mathbf{u} into the first inequality in (17), we arrive at the dual formulation of (16):

$$\left. \begin{aligned} & \text{Find } \boldsymbol{\lambda} \in \mathbb{X} \text{ such that} \\ & \mathcal{S}(\boldsymbol{\lambda}) \leq \mathcal{S}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathbb{X}, \end{aligned} \right\} \quad (19)$$

where

$$\mathcal{S}(\boldsymbol{\mu}) = \frac{1}{2} \boldsymbol{\mu}^T \mathbf{F}_\kappa \boldsymbol{\mu} - \boldsymbol{\mu}^T \mathbf{d}$$

with $\mathbf{F}_\kappa = \mathbf{C} \mathbf{A}_\kappa^{-1} \mathbf{C}^T + \text{diag}(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{E})$ being symmetric, positive definite and $\mathbf{d} = \mathbf{C} \mathbf{A}_\kappa^{-1} \mathbf{b} - (\mathbf{0}^T, \mathbf{0}^T, \mathbf{0}^T, \mathbf{c}^T)^T$.

It should be noted that (19) is more convenient for numerical solution unlike (16) as the function \mathcal{S} is differentiable (quadratic) and the feasible set \mathbb{X} is defined by the separable spherical constraints.

A path-following variant of the interior point method developed for such type of problems in contact mechanics of solids will be shortly presented in the next section. Another way how to solve our problem originates from (18) with the algebraic stick–slip boundary conditions expressed by means of projections on appropriate convex sets. Then a Newton-type method can be used, for which an implementation based on the dual formulation is efficient.

4. Algorithms

In this section, we present main ideas of the above mentioned algorithms that turned out to be highly efficient for solving 3D contact problems of solid mechanics [14,15].

4.1. Path-following method

Let $L : \mathbb{R}^{3n_s+n_p} \times \mathbb{R}_+^{n_s} \rightarrow \mathbb{R}$ be the Lagrangian to (19):

$$L(\boldsymbol{\mu}, \mathbf{v}) = \mathcal{S}(\boldsymbol{\mu}) + \sum_{i \in \mathcal{N}} v_i (\mu_{t_1,i}^2 + \mu_{t_2,i}^2 - g_i^2), \quad (20)$$

where $\mathbf{v} \in \mathbb{R}^{n_s}$, $\mathbf{v} \geq \mathbf{0}$, is the vector of Lagrange multipliers releasing the spherical constraints appearing in \mathbb{X} . Let $\mathbf{z} := -\nabla_{\mathbf{v}} L(\boldsymbol{\mu}, \mathbf{v})$ be the new variable and define the function $\mathbf{H} : \mathbb{R}^{5n_s+n_p} \rightarrow \mathbb{R}^{5n_s+n_p}$,

$$\mathbf{H}(\boldsymbol{\omega}) := (\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \mathbf{v})^T, (\nabla_{\mathbf{v}} L(\boldsymbol{\mu}, \mathbf{v}) + \mathbf{z})^T, \mathbf{e}^T \mathbf{M} \mathbf{Z})^T,$$

where $\boldsymbol{\omega} = (\boldsymbol{\mu}^T, \mathbf{v}^T, \mathbf{z}^T)^T \in \mathbb{R}^{5n_s+n_p}$, $\mathbf{M} = \text{diag}(\mathbf{v})$, $\mathbf{Z} = \text{diag}(\mathbf{z})$, and $\mathbf{e} \in \mathbb{R}^{n_s}$ is the vector whose all components are equal to 1. The solution $\boldsymbol{\lambda}$ to (19) is the first component of the vector $\bar{\boldsymbol{\omega}} = (\boldsymbol{\lambda}^T, \bar{\mathbf{v}}^T, \bar{\mathbf{z}}^T)^T$, which satisfies

$$\mathbf{H}(\bar{\boldsymbol{\omega}}) = \mathbf{0}, \quad \bar{\mathbf{v}} \geq \mathbf{0}, \quad \bar{\mathbf{z}} \geq \mathbf{0}, \quad (21)$$

since (21) is equivalent to the Karush–Kuhn–Tucker conditions for $(\boldsymbol{\lambda}, \bar{\mathbf{v}})$ to be a saddle-point of (20) on $\mathbb{R}^{3n_s+n_p} \times \mathbb{R}_+^{n_s}$.

To derive the path-following algorithm, we replace (21) by the following perturbed problem:

$$\left. \begin{aligned} & \text{Find } \boldsymbol{\omega}^\tau = ((\boldsymbol{\lambda}^\tau)^T, (\mathbf{v}^\tau)^T, (\mathbf{z}^\tau)^T)^T \in \mathbb{R}^{5n_s+n_p} \text{ such that} \\ & \mathbf{H}(\boldsymbol{\omega}^\tau) = (\mathbf{0}^T, \mathbf{0}^T, \tau \mathbf{e}^T)^T, \quad \mathbf{v}^\tau > \mathbf{0}, \quad \mathbf{z}^\tau > \mathbf{0}, \end{aligned} \right\} \quad (22)$$

where $\tau \in \mathbb{R}_+$. Solutions $\boldsymbol{\omega}^\tau$ to (22) define a curve $\mathcal{C}(\tau)$ in $\mathbb{R}^{5n_s+n_p}$ called the *central path*. This curve approaches $\bar{\boldsymbol{\omega}}$ when τ tends to zero. We combine the damped Newton method used for solving the equation in (22) with an appropriate change of τ which guarantees that the iterations belong to a neighbourhood $\mathcal{N}(c_1, c_2)$ of $\mathcal{C}(\tau)$ defined by

$$\mathcal{N}(c_1, c_2) = \{\boldsymbol{\omega} = (\boldsymbol{\mu}^T, \mathbf{v}^T, \mathbf{z}^T)^T \in \mathbb{R}^{5n_s+n_p} : v_i z_i \geq c_1 \vartheta, i = 1, \dots, n_s, \\ \mathbf{v} \geq \mathbf{0}, \mathbf{z} \geq \mathbf{0}, \|\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \mathbf{v})\| \leq c_2 \vartheta, \|\nabla_{\mathbf{v}} L(\boldsymbol{\mu}, \mathbf{v}) + \mathbf{z}\| \leq c_2 \vartheta\},$$

where $c_1 \in (0, 1]$, $c_2 \geq 1$, and $\vartheta := \vartheta(\boldsymbol{\omega}) = \mathbf{v}^T \mathbf{z} / n_s$. In the k th iteration, we replace $\tau := \tau^{(k)}$ by the product of $\vartheta^{(k)} := \vartheta(\boldsymbol{\omega}^{(k)})$ with the centring parameter $c^{(k)}$ chosen as in [18]. The algorithm uses also the *Armijo-type condition* (24) ensuring that the sequence $\{\vartheta^{(k)}\}$ is monotonically decreasing. By $\mathbf{J}(\boldsymbol{\omega})$ in (23), we denote the Jacobian matrix of \mathbf{H} at $\boldsymbol{\omega}$.

ALGORITHM PF: Given $c_1 \in (0, 1]$, $c_2 \geq 1$, $0 < c_{\min} \leq c_{\max} \leq 1/2$, $c_3 \in (0, 1)$, and $\varepsilon \geq 0$. Let $\boldsymbol{\omega}^{(0)} \in \mathcal{N}(c_1, c_2)$ and set $k := 0$.

(i) Choose $c^{(k)} \in [c_{\min}, c_{\max}]$.

(ii) Solve

$$\mathbf{J}(\boldsymbol{\omega}^{(k)}) \Delta \boldsymbol{\omega}^{(k+1)} = -\mathbf{H}(\boldsymbol{\omega}^{(k)}) + (\mathbf{0}^T, \mathbf{0}^T, c^{(k)} \vartheta^{(k)} \mathbf{e}^T)^T. \quad (23)$$

(iii) Set $\boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \alpha^{(k)} \Delta \boldsymbol{\omega}^{(k+1)}$ with the largest $\alpha^{(k)} \in (0, 1]$ such that $\boldsymbol{\omega}^{(k+1)} \in \mathcal{N}(c_1, c_2)$ and

$$\vartheta^{(k+1)} \leq c_3 (1 - \alpha^{(k)} (1 - c^{(k)})) \vartheta^{(k)}. \quad (24)$$

(iv) Return $\bar{\boldsymbol{\omega}} = \boldsymbol{\omega}^{(k+1)}$, if $err^{(k)} := \|\boldsymbol{\omega}^{(k+1)} - \boldsymbol{\omega}^{(k)}\| / \|\boldsymbol{\omega}^{(k+1)}\| \leq \varepsilon$, else set $k := k + 1$ and go to step (i).

The bounds imposed on the parameters mentioned in the initialization section follow from the convergence analysis presented in [14].

The computational efficiency depends on the way how the inner linear systems (23) are solved. The Jacobian matrix is non-symmetric and indefinite with the following block structure:

$$\mathbf{J}(\boldsymbol{\omega}^{(k)}) = \begin{pmatrix} \mathbf{F}_k + \mathbf{D}_M & \mathbf{J}_{12} & \mathbf{0} \\ \mathbf{J}_{12}^T & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{Z} & \mathbf{M} \end{pmatrix}, \quad \mathbf{J}_{12} = \begin{pmatrix} 2\mathbf{X}_1 & \mathbf{0} \\ 2\mathbf{X}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $\mathbf{D}_M = \text{diag}(2\mathbf{M}, 2\mathbf{M}, \mathbf{0})$ and $\mathbf{X}_k = \text{diag}(\boldsymbol{\mu}_{1k}, k = 1, 2)$. Eliminating the 2nd and 3rd components of $\Delta \boldsymbol{\omega}^{(k+1)}$, we get the reduced linear system for $\Delta \boldsymbol{\mu}^{(k+1)}$ with the Schur complement

$$\mathbf{J}_{SC} = \mathbf{F}_k + \mathbf{D}_M + \mathbf{J}_{12} \mathbf{M} \mathbf{Z}^{-1} \mathbf{J}_{12}^T,$$

where $\mathbf{Z} = \text{diag}(\mathbf{Z}_1, \mathbf{Z}_2)$ and $\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2)$. As $\boldsymbol{\mu}^{(k)} > \mathbf{0}$, $\mathbf{z}^{(k)} > \mathbf{0}$, the matrix \mathbf{J}_{SC} is symmetric, positive definite and the reduced linear system can be solved by the conjugate gradient method. In order to guarantee its convergence, we use the preconditioner:

$$\mathbf{P}_{SC} = \mathbf{D}_{F_k} + \mathbf{D}_M + \mathbf{J}_{12} \mathbf{M} \mathbf{Z}^{-1} \mathbf{J}_{12}^T,$$

where $\mathbf{D}_{F_k} = \text{diag}(\mathbf{F}_k)$. The eigenvalues of the preconditioned matrix $\mathbf{P}_{SC}^{-1} \mathbf{J}_{SC}$ belong to an interval which does not depend on the iteration and the spectral condition number $cond$ is bounded by (see [14]):

$$cond(\mathbf{P}_{SC}^{-1} \mathbf{J}_{SC}) \leq cond(\mathbf{D}_{F_k}) cond(\mathbf{F}_k). \quad (25)$$

In computations, we approximate \mathbf{D}_{F_k} replacing \mathbf{A}_k^{-1} in \mathbf{F}_k by $\text{diag}(\mathbf{A}_k)^{-1}$.

The conjugate gradient method used in the k th step of ALGORITHM PF is initialized and terminated adaptively. The initial iteration is taken as the computed result in the previous iteration and the (inner) iterations are terminated, if the relative residuum is less than the stopping tolerance given by

$$tol^{(k)} = \min\{r_{tol} \times err^{(k-1)}, c_{fact} \times tol^{(k-1)}\},$$

where $0 < r_{tol} < 1$, $0 < c_{fact} < 1$, $err^{(-1)} = 1$, and $tol^{(-1)} = r_{tol} / c_{fact}$.

4.2. Semi-smooth Newton method

Let $C(r) = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| \leq r\}$ be the circle of radius $r \geq 0$ and $\mathbf{P}_r : \mathbb{R}^2 \rightarrow C(r)$ be the projection on $C(r)$. Clearly

$$\mathbf{P}_r(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{for } \|\mathbf{x}\| \leq r, \\ \frac{r}{\|\mathbf{x}\|} \mathbf{x} & \text{for } \|\mathbf{x}\| > r. \end{cases} \quad (26)$$

We will use the generalized Jacobian matrix $\mathbf{J}_{P_r} : \mathbb{R}^2 \rightarrow \mathbb{R}^{2 \times 2}$ of \mathbf{P}_r defined by:

$$\mathbf{J}_{P_r}(\mathbf{x}) = \begin{cases} \mathbf{I} & \text{for } \|\mathbf{x}\| \leq r, \\ \frac{r}{\|\mathbf{x}\|} \left(\mathbf{I} - \frac{1}{\|\mathbf{x}\|^2} \mathbf{x}\mathbf{x}^T \right) & \text{for } \|\mathbf{x}\| > r. \end{cases} \quad (27)$$

It is readily seen that the matrix $\mathbf{J}_{P_r}(\mathbf{x})$ is positive semidefinite for all \mathbf{x} .

The last two lines in (18) representing the stick–slip law can be equivalently written as

$$\boldsymbol{\lambda}_{t,i} = \mathbf{P}_{g_i}(\boldsymbol{\lambda}_{t,i} + \rho_i \mathbf{u}_{t,i}), \quad i \in \mathcal{N}, \quad (28)$$

where $\rho_i > 0$, $i \in \mathcal{N}$ are arbitrary but fixed parameters. Let the adhesive function $\kappa > 0$ be positive on γ_S . Introducing the new variables $\mathbf{s}_{t_1}, \mathbf{s}_{t_2} \in \mathbb{R}^{n_s}$ and $\mathbf{s}_{t,i} = (s_{t_1,i}, s_{t_2,i})^T \in \mathbb{R}^2$ such that¹

$$\mathbf{s}_{t,i} = \kappa_i \mathbf{u}_{t,i} + \boldsymbol{\lambda}_{t,i}, \quad i \in \mathcal{N}$$

and setting $\rho_i = \kappa_i$ for all $i \in \mathcal{N}$, we get

$$\mathbf{u}_{t,i} - \kappa_i^{-1}(\mathbf{s}_{t,i} - \mathbf{P}_{g_i}(\mathbf{s}_{t,i})) = \mathbf{0}, \quad i \in \mathcal{N}. \quad (29)$$

This form of (28) leads to the symmetric generalized Jacobian matrix of the whole system (18). Note that the vectors $\mathbf{s}_{t,i}$ approximate $-\boldsymbol{\sigma}_t(\mathbf{x}_i)$ at the nodes $\mathbf{x}_i \in \bar{\gamma}_S \setminus \bar{\gamma}_D$, $i \in \mathcal{N}$.

From (15) and (29) we see that (18) can be written as the system of non-smooth equations:

$$\mathbf{G}(\mathbf{y}) = \mathbf{0} \quad (30)$$

with $\mathbf{G} : \mathbb{R}^{3n_u + 3n_s + n_p} \rightarrow \mathbb{R}^{3n_u + 3n_s + n_p}$ defined at $\mathbf{y} = (\mathbf{u}^T, \mathbf{s}_{t_1}^T, \mathbf{s}_{t_2}^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T$ by

$$\mathbf{G}(\mathbf{y}) := \begin{pmatrix} \mathbf{A}\mathbf{u} + \mathbf{T}_1^T \mathbf{s}_{t_1} + \mathbf{T}_2^T \mathbf{s}_{t_2} + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{B}^T \mathbf{p} - \mathbf{b} \\ \mathbf{T}_1 \mathbf{u} - \mathbf{D}_\kappa^{-1}(\mathbf{s}_{t_1} - \mathbf{H}_1(\mathbf{s}_{t_1}, \mathbf{s}_{t_2})) \\ \mathbf{T}_2 \mathbf{u} - \mathbf{D}_\kappa^{-1}(\mathbf{s}_{t_2} - \mathbf{H}_2(\mathbf{s}_{t_1}, \mathbf{s}_{t_2})) \\ \mathbf{N}\mathbf{u} \\ \mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c} \end{pmatrix}$$

and $\mathbf{D}_\kappa = \text{diag}(\kappa_1, \dots, \kappa_{n_s})$, $\mathbf{H}_j(\mathbf{s}_{t_1}, \mathbf{s}_{t_2}) = (\mathbf{P}_{g_{1,j}}(\mathbf{s}_{t_1,1}), \dots, \mathbf{P}_{g_{n_s,j}}(\mathbf{s}_{t_1,n_s}))^T$, where $\mathbf{P}_{g_{i,j}}$, $i \in \mathcal{N}$ stands for the j th component of \mathbf{P}_{g_i} , $j = 1, 2$.

Eq. (30) will be solved by the Newton-type iterations:

$$\mathbf{J}_G(\mathbf{y}^{(k)})\mathbf{y}^{(k+1)} = \mathbf{J}_G(\mathbf{y}^{(k)})\mathbf{y}^{(k)} - \mathbf{G}(\mathbf{y}^{(k)}), \quad k = 0, 1, \dots, \quad (31)$$

where $\mathbf{J}_G(\mathbf{y})$ is an arbitrary generalized Jacobian matrix of \mathbf{G} at \mathbf{y} and $\mathbf{y}^{(0)}$ is an initial approximation. We use the active/inactive set terminology. Let $\mathcal{A}, \mathcal{I} \subseteq \mathcal{N}$ be the *active*, and *inactive* set at \mathbf{y} , respectively:

$$\mathcal{A} := \mathcal{A}(\mathbf{y}) = \{i \in \mathcal{N} : \|\mathbf{s}_{t,i}\| \leq g_i\}, \quad \mathcal{I} := \mathcal{I}(\mathbf{y}) = \mathcal{N} \setminus \mathcal{A}.$$

Let $\mathbf{D}_\mathcal{A}, \mathbf{D}_\mathcal{I} \in \mathbb{R}^{n_s \times n_s}$ be the *indicator* matrices defined by:

$$\mathbf{D}_\mathcal{A} = \text{diag}(d_1, \dots, d_{n_s}), \quad d_i = 1 \text{ if } i \in \mathcal{A}, \quad d_i = 0 \text{ if } i \notin \mathcal{A},$$

$$\mathbf{D}_\mathcal{I} = \mathbf{I} - \mathbf{D}_\mathcal{A}.$$

¹ Recall that $s_{t_j,i}$ is the i th component of the vector \mathbf{s}_{t_j} , $j = 1, 2$ and similarly for other vectors.

It follows from (26) that

$$\mathbf{G}(\mathbf{y}) = \begin{pmatrix} \mathbf{A}\mathbf{u} + \mathbf{T}_1^T \mathbf{s}_{t_1} + \mathbf{T}_2^T \mathbf{s}_{t_2} + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{B}^T \mathbf{p} - \mathbf{b} \\ \mathbf{T}_1 \mathbf{u} - \mathbf{D}_\kappa^{-1} \mathbf{D}_\mathcal{I} (\mathbf{I} - \mathbf{D}_{g/\|\mathbf{s}_t\|}) \mathbf{s}_{t_1} \\ \mathbf{T}_2 \mathbf{u} - \mathbf{D}_\kappa^{-1} \mathbf{D}_\mathcal{I} (\mathbf{I} - \mathbf{D}_{g/\|\mathbf{s}_t\|}) \mathbf{s}_{t_2} \\ \mathbf{N}\mathbf{u} \\ \mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c} \end{pmatrix},$$

where $\mathbf{D}_{g/\|\mathbf{s}_t\|} = \text{diag}(d_1, \dots, d_{n_s})$, $d_i = g_i/\|\mathbf{s}_{t,i}\|$ if $i \in \mathcal{I}$, $d_i = 0$ if $i \notin \mathcal{I}$. Standard differentiation rules and (27) yield:

$$\mathbf{J}_G(\mathbf{y}) = \begin{pmatrix} \mathbf{A} & \mathbf{T}_1^T & \mathbf{T}_2^T & \mathbf{N}^T & \mathbf{B}^T \\ \mathbf{T}_1 & -\mathbf{D}_{11} & -\mathbf{D}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_2 & -\mathbf{D}_{12} & -\mathbf{D}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{E} \end{pmatrix}, \quad (32)$$

where

$$\begin{aligned} \mathbf{D}_{jj} &= \mathbf{D}_\kappa^{-1} \mathbf{D}_\mathcal{I} (\mathbf{I} - \mathbf{D}_{g/\|\mathbf{s}_t\|} (\mathbf{I} - \mathbf{D}_{1/\|\mathbf{s}_t\|^2} \mathbf{D}_{s_{t_j}^2})), \quad j = 1, 2, \\ \mathbf{D}_{12} &= \mathbf{D}_\kappa^{-1} \mathbf{D}_\mathcal{I} \mathbf{D}_{g/\|\mathbf{s}_t\|} \mathbf{D}_{1/\|\mathbf{s}_t\|^2} \mathbf{D}_{s_{t_1}} \mathbf{D}_{s_{t_2}}, \\ \mathbf{D}_{1/\|\mathbf{s}_t\|^2} &= \text{diag}(d_1, \dots, d_{n_s}), \quad d_i = 1/\|\mathbf{s}_{t,i}\|^2 \text{ if } i \in \mathcal{I}, \quad d_i = 0 \text{ if } i \notin \mathcal{I}, \\ \mathbf{D}_{s_{t_j}^2} &= \text{diag}(s_{t_j,1}^2, \dots, s_{t_j,n_s}^2), \quad j = 1, 2, \\ \mathbf{D}_{s_{t_j}} &= \text{diag}(\mathbf{s}_{t_j}), \quad j = 1, 2. \end{aligned}$$

To solve the linear systems (31), we use the Schur complement \mathbf{S} to \mathbf{A} -block in (32) defined by:

$$\mathbf{S} = \mathbf{F} + \mathbf{D}, \quad (33)$$

where $\mathbf{F} = \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T$ with the same \mathbf{C} as in (19) and

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{12} & \mathbf{D}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} \end{pmatrix}.$$

The right hand-sides of the Schur complement linear systems are given by $\mathbf{d} = \mathbf{C}\mathbf{A}^{-1}\mathbf{b} - (\mathbf{0}^T, \mathbf{0}^T, \mathbf{0}^T, \mathbf{c}^T)^T$. We arrive at the implementation of (31), in which the iterations are performed only with last four components of \mathbf{y} collected in the vector $\hat{\boldsymbol{\lambda}} = (\mathbf{s}_{t_1}^T, \mathbf{s}_{t_2}^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T$.

ALGORITHM SSN: Let $\hat{\boldsymbol{\lambda}}^{(0)} \in \mathbb{R}^{3n_s+n_p}$, $\varepsilon \geq 0$ and set $k := 0$.

(i) Assembly the active/inactive sets \mathcal{A} and \mathcal{I} at $\hat{\boldsymbol{\lambda}}^{(k)}$ and the respective matrix \mathbf{D} to build \mathbf{S} in (33).

(ii) Solve the linear system:

$$\mathbf{S}\hat{\boldsymbol{\lambda}}^{(k+1)} = \mathbf{d}. \quad (34)$$

(iii) Return $\bar{\boldsymbol{\lambda}} = \hat{\boldsymbol{\lambda}}^{(k+1)}$ and $\bar{\mathbf{u}} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{C}^T \bar{\boldsymbol{\lambda}})$, if $\text{err}^{(k)} := \|\hat{\boldsymbol{\lambda}}^{(k+1)} - \hat{\boldsymbol{\lambda}}^{(k)}\|/\|\hat{\boldsymbol{\lambda}}^{(k+1)}\| \leq \varepsilon$, else set $k := k + 1$ and go to step (i).

From the definition of the matrix \mathbf{D} we see that it is positive semidefinite and consequently \mathbf{S} is positive definite. Hence, the linear systems (34) can be solved by the conjugate gradient method. It is well-known that this method is inefficient, if the spectral condition number of the system matrix is large. In our case $\text{cond}(\mathbf{S})$ tends to infinity, when the finite element mesh norm approaches zero. This follows from the presence of \mathbf{D}_κ^{-1} in \mathbf{D} and the fact that

κ_i , $i \in \mathcal{N}$ depend on the area of the respective finite element triangles (see (13)). In order to improve conditioning of the solved linear systems, we use the following preconditioner:

$$\mathbf{P}_S = \mathbf{D}_F + \begin{pmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{12} & \mathbf{D}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \text{diag}(\mathbf{E}) \end{pmatrix},$$

where $\mathbf{D}_F = \text{diag} \mathbf{F}$. Let us note that the result analogous to (25) can be established, i.e.,

$$\text{cond}(\mathbf{P}_S^{-1} \mathbf{S}) \leq \text{cond}(\mathbf{D}_F) \text{cond}(\mathbf{F})$$

so that the spectral condition number of the preconditioned Schur complement \mathbf{S} does not depend on κ_i , $i \in \mathcal{N}$.

Finally note that conjugate gradient method in the k th step of ALGORITHM SSN is initialized and terminated adaptively using the same ideas as in ALGORITHM PF.

5. Numerical experiments

The computations were performed by the supercomputer Salomon at IT4I VŠB-TU Ostrava [20]. The Salomon cluster consists of 1009 compute nodes. Each node is a powerful x86-64 computer with Intel Xeon E5-2680v3 processors equipped with 24 cores and at least 128 GB RAM. All codes are implemented in Matlab R2020a. The velocity component is eliminated in both algorithms implicitly by solving auxiliary linear systems involving \mathbf{A} with the preliminary Cholesky factorization of \mathbf{A} . To this end we use the Matlab function `chol`. We use ALGORITHM PF with $c_1 = 10^{-3}$, $c_2 = 10^9$, $c_{\min} = 10^{-12}$, $c_{\max} = 0.5$, $c_3 = 10^{-2}$, $\varepsilon = 10^{-3}$, $r_{\text{tol}} = 0.9$, $c_{\text{fact}} = 0.9$. These values turned out to be optimal, as it follows from the tests in [14]. ALGORITHM SSN uses $r_{\text{tol}} = 0.01$, $c_{\text{fact}} = 0.5$, and $\varepsilon = 10^{-3}$. The termination tolerances ε lead to the relative residua of order 10^{-5} . In the tables below we observe the numbers n_{it} , n_F of the outer iterations, and of matrix–vector multiplications by \mathbf{F}_κ or \mathbf{F} , respectively. Note that n_F determines overall complexities of computations. In all examples the computational domain Ω is represented by the unit cube $(0, 1)^3$. To construct its partition \mathcal{T}_h we first cut Ω into small cubes and then each cube is split into five tetrahedras, see Fig. 1. The partition \mathcal{T}_h is generated by Iso2mesh toolbox [10]. As we have already mentioned, the finite element spaces use P1-bubble/P1 element pairs on \mathcal{T}_h . The resulting mesh will be characterized by values of the parameters n_u , n_p , n_s introduced in Section 3. The stiffness matrices are assembled by the vectorized code [2,11].

Example 1. The boundary $\partial\Omega$ consists of three parts γ_D , γ_N , and γ_S defined by $\gamma_D = \gamma_{\text{top}} \cup \gamma_{\text{front}} \cup \gamma_{\text{back}}$, $\gamma_N = \gamma_{\text{left}} \cup \gamma_{\text{right}}$, $\gamma_S = (0, 1) \times (0, 1) \times \{0\}$, where $\gamma_{\text{top}} = (0, 1) \times (0, 1) \times \{1\}$, $\gamma_{\text{front}} = \{0\} \times (0, 1) \times (0, 1)$, $\gamma_{\text{back}} = \{1\} \times (0, 1) \times (0, 1)$, $\gamma_{\text{left}} = (0, 1) \times \{0\} \times (0, 1)$, $\gamma_{\text{right}} = (0, 1) \times \{1\} \times (0, 1)$. Data of problem (3) are as follows: $\mathbf{f} = -2\nu \text{div} D(\mathbf{u}_{\text{exp}}) + \nabla p_{\text{exp}}$, $\nu = 1/2$, $\boldsymbol{\sigma}_N = 2\nu D(\mathbf{u}_{\text{exp}})\mathbf{n} - p_{\text{exp}}\mathbf{n}$, $g = 50$, and $\kappa = 500$, where $\mathbf{u}_{\text{exp}} = (u_{\text{exp},1}, u_{\text{exp},2}, u_{\text{exp},3})$,

$$\begin{aligned} u_{\text{exp},1}(x, y, z) &= 4(1 - \cos(2\pi x)) \sin(2\pi y)z(1 - z), \\ u_{\text{exp},2}(x, y, z) &= 4 \sin(2\pi x)(\cos(2\pi y) - 1)z(1 - z), \\ u_{\text{exp},3}(x, y, z) &= 0, \\ p_{\text{exp}}(x, y, z) &= 2\pi(-\cos(2\pi x) + 2\cos(2\pi y) - \cos(2\pi z)). \end{aligned}$$

It is easy to verify that the couple $(\mathbf{u}_{\text{exp}}, p_{\text{exp}})$ solves the Stokes system with the no-slip condition on $\gamma_D \cup \gamma_S$ and the Neumann condition with $\boldsymbol{\sigma}_N$ on γ_N . Therefore for an appropriate choice of g it solves also problem (3). Data are chosen in such a way that both, slip and stick zones appear on γ_S . The tangential velocity field \mathbf{u}_t and the pressure on γ_S are seen in Fig. 2, while the distribution of $\|\mathbf{u}_t\|$ and $\max\{\|\boldsymbol{\sigma}_t\|, g\}$ on γ_S are depicted in Fig. 3. Finally Fig. 4 represents $\boldsymbol{\sigma}_t$ and $\tilde{\boldsymbol{\sigma}}_t$ on γ_S .

In Table 1 we summarize the resulting values of n_{it} , n_F , and CPU time (in seconds) for different finite element meshes with increasing n_u , n_p , and n_s . Looking at the values of n_F , one can deduce that both algorithms are scalable, i.e., n_F changes moderately with growing size of the problem. The computations without preconditioning are considerably less efficient as it is seen from Table 2.

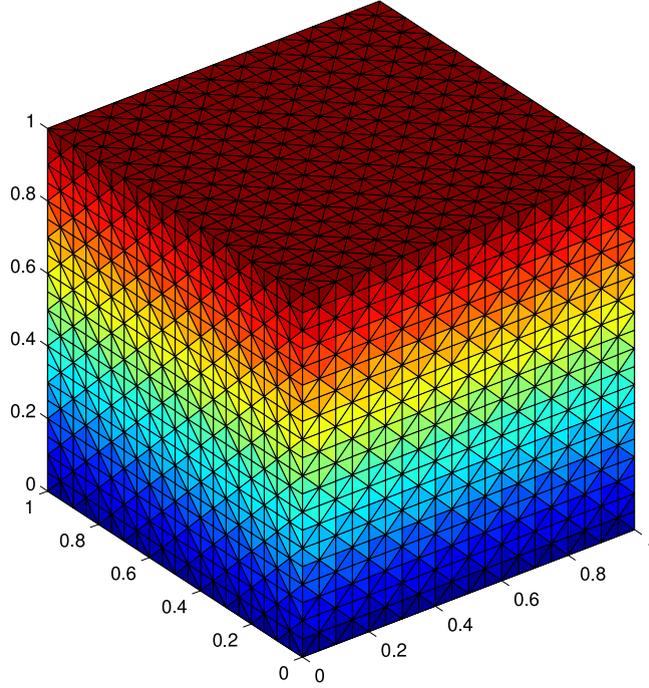


Fig. 1. Partition \mathcal{T}_h of $\bar{\Omega} = [0, 1]^3$.

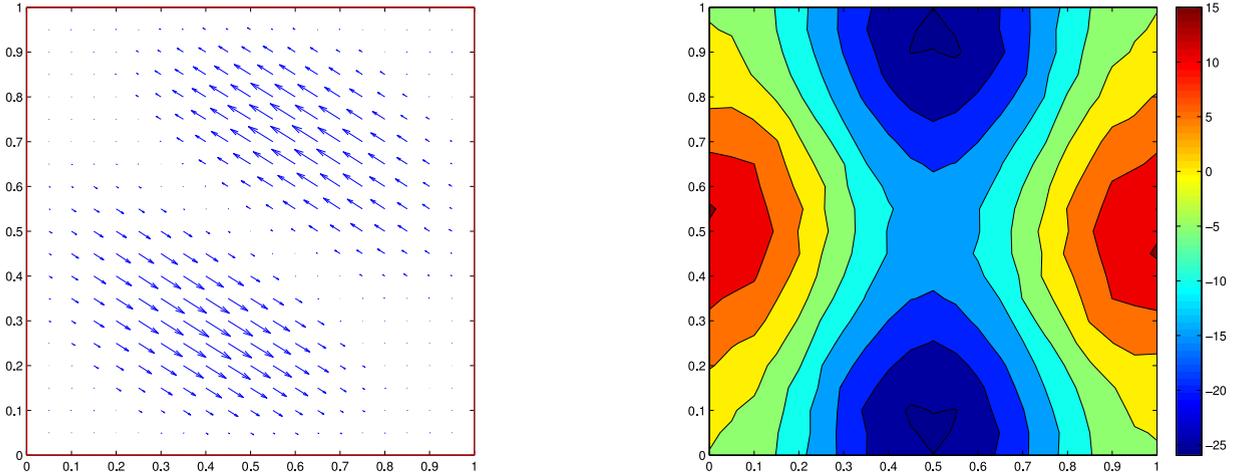


Fig. 2. Tangential velocity field (left), pressure (right) on γ_S .

Example 2. Data are the same as in the previous example with the exception of g . By an appropriate choice of the slip bound one can affect the behaviour of the fluid on γ_S . Our aim is to choose g in such a way that either the slip or the stick occurs on the whole or at least on a substantial part of γ_S . First we set $g = 1$. In this case the upper bound in (3)₆ is attained and the slip appears almost everywhere on γ_S except for small neighbourhoods around the corners. On the other hand for $g = 500$ the upper bound in (3)₆ is not reached and $\mathbf{u}_t = \mathbf{0}$ and in view of (3)₅ $\mathbf{u} = \mathbf{0}$ on γ_S . The computational attributes for both cases are summarized in Tables 4 and 5. From them one can observe the scalability of the methods also for these limit situations. In the case $g = 500$, our experimental data \mathbf{u}_{exp} and p_{exp} coincide with the analytic solution of the problem (3). Therefore convergence rate of the finite element approximation may be evaluated. In Table 3 we introduce $Err_1(h) = \|\mathbf{u}_h - \mathbf{u}_{exp}\|_{(L^2(\Omega))^3}$,

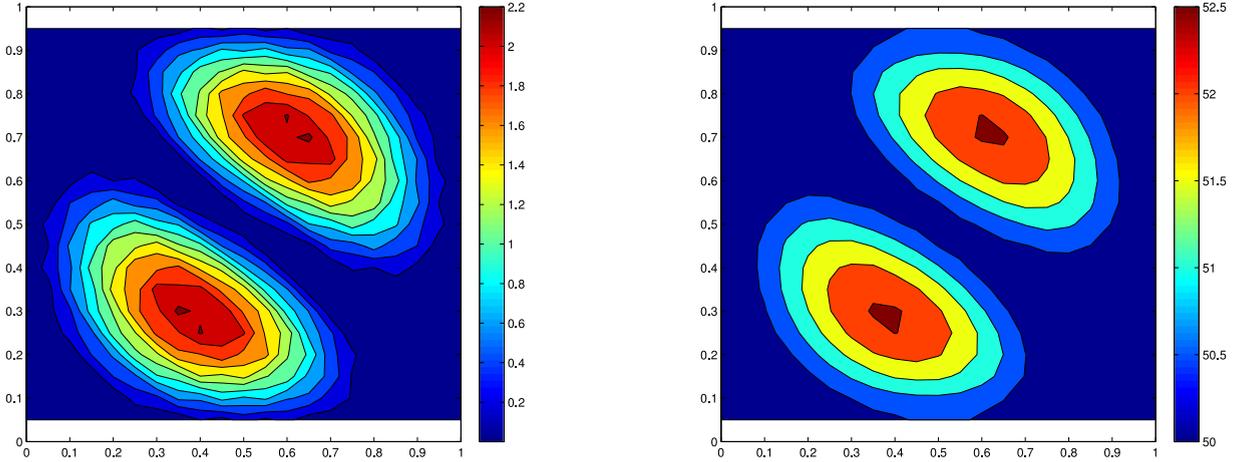


Fig. 3. Tangential velocity norm (left), $\max\{\|\sigma_t\|, g\}$ (right) on γ_S .

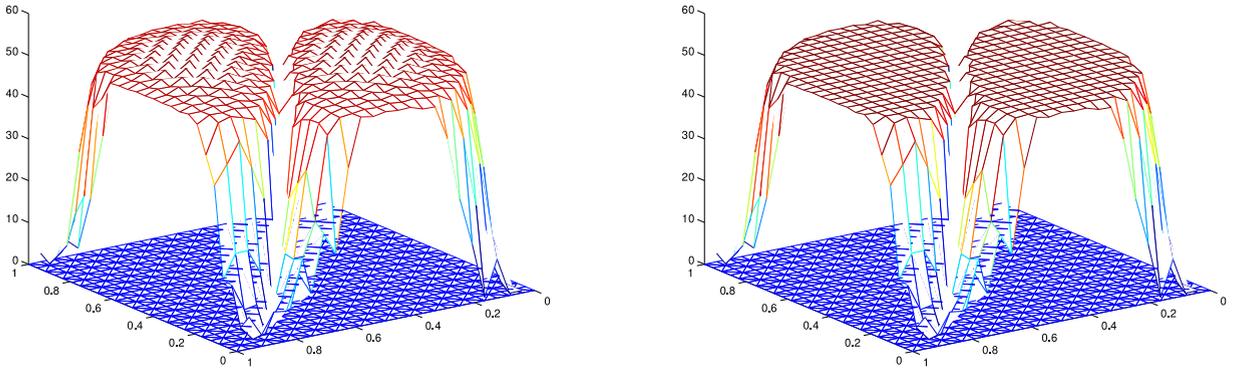


Fig. 4. Distribution of σ_t (left) and $\tilde{\sigma}_t$ (right) on γ_S .

Table 1

Computations with preconditioner for $g = 50$, $\kappa = 500$.

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time (s.)	n_{it}/n_F	time (s.)	n_{it}/n_F
1512/729/63	1	9/397	0.2	7/189
5148/2197/143	3	11/443	1	7/203
12240/4913/255	7	12/319	4	8/252
23940/9261/399	19	11/351	8	8/238
41400/15625/575	44	12/327	16	8/235
65772/24389/783	70	11/293	30	8/231
98208/35937/1023	124	11/265	66	8/262
139860/50653/1295	249	11/257	135	8/270

$Err_2(h) = \|\mathbf{u}_h - \mathbf{u}_{exp}\|_{(H^1(\Omega))^3} + \|p_h - p_{exp}\|_{L^2(\Omega)}$, and $Rate_q(h_{j-1}, h_j) = \log_2(Err_q(h_{j-1})/Err_q(h_j))$ for $q = 1, 2$. Thus, the convergence rate is estimated from each two adjacent discretizations. One can see that these estimates are better than it is predicted by the theory [1,5].

Example 3. Up to now we supposed that γ_N is non-empty ensuring uniqueness of the pressure p . If $\gamma_N = \emptyset$ then the pressure is determined up to an arbitrary constant. The easiest way how to overcome this difficulty in computations is to fix the value of p at some point in Ω . We use the same data as in Example 1 excluding the boundary condition

Table 2

Computations without preconditioner for $g = 50$, $\kappa = 500$.

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time (s.)	n_{it}/n_F	time (s.)	n_{it}/n_F
1512/729/63	4	12/ 1512	2	11/ 1871
5148/2197/ 143	33	15/ 7360	10	10/ 3227
12240/4913/255	186	16/ 15908	83	15/ 7103
23940/9261/399	> 3600	–/–	201	9/ 6695

Table 3

Convergence rate of FE approximation for P1-bubble/P1.

j	h_j	$Err_1(h_j)$	$Rate_1(h_{j-1}, h_j)$	$Err_2(h_j)$	$Rate_2(h_{j-1}, h_j)$
1	1/4	0.3425	–	3.1748	–
2	1/8	0.1368	1.32	1.3565	1.22
3	1/16	0.0483	1.5	0.5308	1.35
4	1/32	0.0244	0.98	0.2360	1.16

Table 4

Computations with preconditioner for $g = 1$, $\kappa = 500$.

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time (s.)	n_{it}/n_F	time (s.)	n_{it}/n_F
1512/729/63	0.3	12/ 140	0.1	5/ 87
5148/2197/143	1	16/ 121	0.4	5/ 76
12240/4913/255	4	22/ 139	1	5/ 70
23940/9261/399	13	24/ 159	2	5/ 74
41400/15625/575	33	30/ 223	5	5/ 69
65772/24389/783	94	42/ 308	8	5/ 70
98208/35937/1023	145	32/ 232	19	5/ 70
139860/50653/1295	302	34/ 251	32	5/ 68

Table 5

Computations with preconditioner for $g = 500$, $\kappa = 500$.

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time (s.)	n_{it}/n_F	time (s.)	n_{it}/n_F
1512/729/63	0.6	12/ 240	0.3	7/ 175
5148/2197/143	2	11/ 423	4	6/ 176
12240/4913/255	7	9/ 365	2	6/ 171
23940/9261/399	29	9/ 577	7	7/ 205
41400/15625/575	58	8/ 541	17	8/ 219
65772/24389/783	107	7/ 415	30	7/ 202
98208/35937/1023	283	8/ 678	49	6/ 171
139860/50653/1295	540	8/ 624	104	6/ 209

on γ_N and with the following minor change, namely the boundary $\partial\Omega$ is now decomposed into two parts γ_D and $\gamma_S = (0, 1) \times (0, 1) \times \{0\}$. To define the discrete pressure \mathbf{p} unequivocally, we prescribe $p(\mathbf{x}) = 0$ at the point $\mathbf{x} = (1, 1, 1)$. This is realized by omitting last row in the matrix \mathbf{B} and the last row and the last column in the matrix \mathbf{E} . Consequently, the matrix \mathbf{C} appearing in both algorithms has full row-rank. Complexity of computations is seen in Table 6. Also in this case one can observe the scalability of the methods.

6. Conclusions

We proposed two algorithms for solving the Stokes flow with the stick–slip boundary conditions in 3D, both based on dual strategies. ALGORITHM PF is the path-following variant of the interior point method whose outer loop

Table 6Computations with preconditioner for $g = 50$, $\kappa = 500$, and $\gamma_N = \emptyset$.

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time (s.)	n_{it}/n_F	time (s.)	n_{it}/n_F
1176/728/49	0.5	10/ 349	0.3	6/ 133
4356/2196/121	1.9	12/ 400	0.6	7/ 172
10800/4912/225	6	14/ 327	2	7/ 194
21660/9260/361	16	14/ 340	6	7/ 191
38088/15624/529	35	12/ 293	11	7/ 189
61236/24388/729	67	12/ 310	22	7/ 188
92256/35936/961	127	11/ 262	45	7/ 204
132300/50652/1225	240	12/ 274	97	8/ 217

uses the dumped Newton iterations. The inner linear systems are solved by the preconditioned conjugate gradient method with the adaptive precision control. The preconditioner (with diagonal blocks) removes ill-conditioning of system matrices in the later iterations. ALGORITHM SSN is the active-set implementation of the semi-smooth Newton method whose inner system matrices are symmetric owing to the presence of the non-zero adhesive coefficient. The inner linear systems are solved similarly as in ALGORITHM PF. Now the preconditioner removes ill-conditioning caused by small finite element mesh norms. Numerical experiments indicate scalability of both algorithms. ALGORITHM SSN is more efficient than ALGORITHM PF also in the case of the comparable complexity characteristic n_F , since its implementation is easier, requires less accompanying computations and programming manipulations. The classical (minimization) active-set strategy algorithms are not efficient by our experiences for these kinds of problems.

Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NSU II) project “IT4Innovations excellence in science - LQ1602” (RK,VS) and by the project No. 17-01747S of the Czech Science Foundation (HK,RK). The paper also includes the results of the internal BUT FIT project FIT-S-20-6427 (VS). A part of this work was prepared during the stay of the second author at LMNO University of Caen Normandy in June 2018.

References

- [1] D.N. Arnold, F. Brezzi, M. Fortin, A stable finite element for the Stokes equations, *Calcolo* 22 (4) (1984) 337–344.
- [2] V. Arzt, Finite Element Meshes and Assembling of Stiffness Matrices (Master’s thesis), VŠB-TU Ostrava, Czech Republic, 2019.
- [3] M. Ayadi, L. Baffico, M.K. Gdoura, T. Sassi, Error estimates for Stokes problem with Tresca friction conditions, *ESAIM Math. Model. Numer. Anal.* 48 (2014) 1413–1429.
- [4] J.K. Djoko, J. Koko, Numerical methods for the Stokes and Navier–Stokes equations driven by threshold slip boundary conditions, *Comput. Methods Appl. Mech. Engrg.* 305 (2016) 936–958.
- [5] H.C. Elman, D.J. Silvester, A.J. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.
- [6] H. Fujita, A mathematical analysis of motions of viscous incompressible fluid under leak and slip boundary conditions, *RIMS Kokyuroku* 888 (1994) 199–216.
- [7] J. Haslinger, I. Hlaváček, J. Nečas, Numerical methods for unilateral problems in solid mechanics, in: *Handbook of Numerical Analysis, Volume IV, Part 2, Vol. 4*, North Holland, Amsterdam, 1996, pp. 313–485.
- [8] J. Haslinger, J. Stebel, Stokes problem with a solution dependent slip bound: Stability of solutions with respect to domains, *ZAMM Z. Angew. Math. Mech.* 96 (9) (2016) 1049–1060.
- [9] M. Hintermüller, V.A. Kovtunencko, K. Kunisch, Obstacle problems with cohesion: A hemi-variational inequality approach and its efficient numerical solution, *SIAM J. Optim.* 21 (2) (2011) 491–516.
- [10] Iso2mesh: A 3D surface and volumetric mesh generator for MATLAB/Octave, 2018, url: <http://iso2mesh.sourceforge.net> [online].
- [11] J. Koko, Efficient MATLAB codes for the 2D/3D Stokes equation with the mini-element, *Informatica* 30 (2) (2019) 243–268.
- [12] R. Kučera, Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints, *SIAM J. Optim.* 19 (2) (2008) 846–862.
- [13] R. Kučera, J. Haslinger, V. Šátek, M. Jarošová, Efficient methods for solving the Stokes problem with slip boundary conditions, *Math. Comput. Simulation* 145 (2018) 114–124.

- [14] R. Kučera, J. Machalová, H. Netuka, P. Ženčák, An interior point algorithm for the minimization arising from 3D contact problems with friction, *Optim. Methods Softw.* 28 (6) (2013) 1195–1217.
- [15] R. Kučera, K. Motyčková, A. Markopoulos, J. Haslinger, On the inexact symmetrized globally convergent semi-smooth Newton method for 3D contact problems with Tresca friction: the R-linear convergence rate, *Optim. Methods Softw.* 35 (1) (2020) 65–86.
- [16] R. Kučera, K. Motyčková, J. Pacholek, T. Sassi, The semi-smooth Newton method for solving the Stokes problem with the stick-slip boundary condition, *AIP Conf. Proc.* 1978 (1) (2018) 360003.
- [17] C.L.M.H. Navier, Sur les lois du mouvement des fluides, *Mem. Acad. R. Sci. Inst. Fr.* 6 (1823) 389–440.
- [18] J. Nocedal, A. Wächter, R.A. Waltz, Adaptive barrier update strategies for nonlinear interior methods, *SIAM J. Optim.* 19 (4) (2009) 1674–1693.
- [19] C.L. Roux, Steady Stokes flows with threshold slip boundary conditions, *Math. Models Methods Appl. Sci.* 15 (2005) 1141–1168.
- [20] Salomon hardware overview, 2020, url: <https://docs.it4i.cz/salomon/hardware-overview/> [online].