

Article

# Numerical Modeling of the Leak through Semipermeable Walls for 2D/3D Stokes Flow: Experimental Scalability of Dual Algorithms

Jaroslav Haslinger <sup>1,2,†</sup> , Radek Kučera <sup>1,2,†</sup> , Kristina Motyčková <sup>2,\*,†</sup>  and Václav Šátek <sup>2,3,†</sup> 

<sup>1</sup> Faculty of Mechanical Engineering, VŠB-TUO, 17. Listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic; jaroslav.haslinger@vsb.cz (J.H.); radek.kucera@vsb.cz (R.K.)

<sup>2</sup> IT4Innovations, VŠB-TUO, Studentská 6231/1B, 708 00 Ostrava-Poruba, Czech Republic; vaclav.satek@vsb.cz

<sup>3</sup> Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, 612 66 Brno, Czech Republic

\* Correspondence: kristina.motyckova@vsb.cz; Tel.: +420-597-329-656

† These authors contributed equally to this work.

**Abstract:** The paper deals with the Stokes flow subject to the threshold leak boundary conditions in two and three space dimensions. The velocity–pressure formulation leads to the inequality type problem that is approximated by the P1-bubble/P1 mixed finite elements. The resulting algebraic system is nonsmooth. It is solved by the path-following variant of the interior point method, and by the active-set implementation of the semi-smooth Newton method. Inner linear systems are solved by the preconditioned conjugate gradient method. Numerical experiments illustrate scalability of the algorithms. The novelty of this work consists in applying dual strategies for solving the problem.

**Keywords:** Stokes problem; threshold leak boundary conditions; interior-point method; semi-smooth Newton method



**Citation:** Haslinger, J.; Kučera, R.; Motyčková, K.; Šátek, V. Numerical Modeling of the Leak through Semipermeable Walls for 2D/3D Stokes Flow: Experimental Scalability of Dual Algorithms. *Mathematics* **2021**, *9*, 2906. <https://doi.org/10.3390/math9222906>

Academic Editor: Marek Lampart

Received: 6 October 2021

Accepted: 8 November 2021

Published: 15 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The no-slip condition is the standard boundary condition in mathematical fluid flow models. It expresses the fact that a fluid adheres to the surface  $S$ , i.e., the velocity  $u = 0$  on  $S$ . Nevertheless, in many situations a motion of the fluid on  $S$  such as a slip along impermeable  $S$  or a leak through semipermeable  $S$  can be observed. The Navier slip condition is the simplest one [1]:  $\sigma_t = -ku_t$ , where  $k > 0$  is the adhesive coefficient,  $\sigma_t$ ,  $u_t$  denotes the shear stress, and the tangential component of  $u$ , respectively. From its form we see that the fluid starts to slip whenever  $\sigma_t \neq 0$ . However, this is not the case, in general. For example, a water drop on an inclined plane which is coated by a non-wetting (hydrophobic) material (teflon, e.g.,) slips only if the angle inclination reaches a specific value. Similarly, a leak of a fluid through semipermeable  $S$  may occur only if the magnitude of the stress vector  $\sigma$  (or some of its components) on  $S$  attains a certain critical value. The true slip or leak boundary conditions should reflect this threshold behavior. To this end one can use tools of convex analysis and write them in a compact form as inclusions for set-valued mappings, defined by the subgradient of appropriate convex functionals [2]. The overview of different threshold slip laws is presented and discussed in [3]. Since “slip” in fluid mechanics is a synonym for “friction” in solid mechanics (in particular in contact mechanics), it is not surprising that friction and slip models use the same terminology. The weak form of mathematical models under threshold slip or leak conditions leads to a variational inequality type problem, intricacy of which depends on the used law. The simplest one is the threshold slip and leak condition of the Tresca type defined by:

$$\text{(slip)} \quad |\sigma_t| \leq g, \quad u_t \neq 0 \quad \text{then} \quad \sigma_t = -g \frac{u_t}{|u_t|} \quad \text{on } S, \quad (1)$$

and

$$\text{(leak)} \quad |\sigma_n| \leq g, \quad u_n \neq 0 \quad \text{then} \quad \sigma_n = -g \frac{u_n}{|u_n|} \quad \text{on } S, \quad (2)$$

respectively, where  $g$  is an a-priori given threshold bound,  $|\cdot|$  stands for the absolute value of a real number or the Euclidean norm in (1) for 3D problems. The Stokes system under (1) and (2) has been studied by Fujita [4,5] and by Roux [6] who considered more general slip models, in which the bound  $g$  may depend on  $|u_t|$ . Despite the fact that the mathematical analysis of this class of problems began relatively late (compared with friction problems of contact mechanics), a lot of theoretical papers on this topic has appeared during past twenty years. Results have been extended to other fluid models and slip conditions such as the steady and unsteady Stokes and Navier–Stokes equations, Coulomb’s type friction law [7–11] and the references therein. The Signorini unilateral boundary conditions [12]:

$$u_n + g \geq 0, \quad \sigma_n + \sigma^* \geq 0, \quad (u_n + g)(\sigma_n + \sigma^*) = 0 \quad \text{on } S, \quad (3)$$

with given  $g$ ,  $\sigma^*$  is another type of conditions which leads to an inequality formulation. The authors used (3) in [13,14] as the artificial outflow conditions prescribed on a part of the boundary to model the blood flow in large arteries.

One of difficulties we face in numerical solution of variational inequalities is the fact that problems are generally nonsmooth and consequently, appropriate computational methods have to be used [15,16]. For the Stokes or Navier–Stokes system with slip or leak conditions nonsmoothness is caused by the presence of the nondifferentiable functional  $j$  whose subgradient defines (1) or (2). In the case of the Signorini conditions nonsmoothness is due to the kinematical constraint  $u_n + g \geq 0$  on  $S$  which imposes additional restrictions on the set of admissible velocity fields. The velocity–pressure formulation of such problems is discretized using a pair of finite element spaces satisfying the Babuska–Brezzi condition [17,18]. One way to handle the nondifferentiable term  $j$ , the kinematical constraint in (3), is to use a regularization of  $j$  and a smooth penalization of the constraint, respectively in order to transform originally the nonsmooth problem into a sequence of smooth ones, which can be solved by standard methods. The regularization of  $j$  has been used in [19], and the penalty method in [13]. Another and the most frequent way of releasing constraints and the nonsmooth character of mathematical models is based on a dualization approach. In the case of the Stokes and Navier–Stokes system, this leads to the weak velocity–pressure type formulation which is enhanced by the additional Lagrange multipliers. The resulting weak formulation is formally the same as the Karush–Kuhn–Tucker conditions in saddle point problems. The majority of papers devoted to the convergence/error analysis of this class of problems uses just the dualization approach [20–27]. Moreover, there exists a wide range of numerical methods for efficient computational realization of the resulting algebraic models [16].

The present paper is devoted to computational aspects of the Stokes problem with threshold leak conditions. We use the dual strategy, i.e., the velocity component is eliminated from an appropriate mixed finite element discretization of 2D and 3D problems. Our aim is to develop efficient algorithms for solving resulting algebraic systems which are satisfied by the dual variables, namely by the discrete pressure, shear and the normal stress on  $S$ , in our case. Let us note that this approach is frequently used in contact problems of solid mechanics. However, its simple transfer to problems of fluid mechanics is not possible due to the presence of the incompressibility condition prescribed in the whole computational domain. The modified path–following (PF) variant of the interior point method has been used in [27] for solving 2D Stokes system with threshold slip conditions, and in [28] for 3D problems. In addition, the latter paper uses also the semi-smooth Newton (SSN) and compares it with the PF method. Although numerical experiments demonstrated the scalable behavior of both approaches, the SSN method turned out to be more efficient because of its simpler implementation. Similar numerical tests and comparisons of SSN and PF will be done in the present paper. It is hard to predict the computational efficiency in advance, since the discrete threshold leak conditions in 2D and 3D problems always

lead to simple box constraints unlike the separable spherical constraints in the case of the threshold slip in 3D. Another important feature is the fact that the dual algebraic formulations involve relatively a small number of constrained unknowns. This excludes the efficient use of some types of algorithms such as strictly feasible minimization methods based on an active-set strategy [29]. It has been found in [27] for 2D problems that they are less efficient than the PF method. The inefficiency is more significant for large-scale 3D problems. The main benefit of this work is an application of dual strategies for solving the problem. The methods presented in this paper may be easily modified for parallelization based on the FETI domain decomposition technique [30].

The paper is organized as follows. In Section 2 we present the classical and several weak formulations of the problem. The mixed finite element approximation based on the P1-bubble/P1 finite element pair is introduced in Section 3. The resulting algebraic problems are starting point for constructing of algorithms in Section 4. Finally, Section 5 contains results of numerical experiments, including the problems with non-unique solutions. Concluding remarks are summarized in Section 6.

The following notation is used in the paper:  $H^k(\Omega)$ ,  $k \geq 0$  integer, stands for the Sobolev space of functions defined in  $\Omega$  which are together with their generalized derivatives up to order  $k$  square integrable in a domain  $\Omega \subset \mathbb{R}^d$  ( $H^0(\Omega) = L^2(\Omega)$ ). The scalar product in  $L^2(\Omega)$  will be denoted by  $(\cdot, \cdot)$ .  $L^2_+(\Omega)$  denotes the set of non-negative functions from  $L^2(\Omega)$ . If  $\gamma$  is a part of the boundary of  $\Omega$  then  $H^{\frac{1}{2}}(\gamma)$  is the trace space on  $\gamma$  of functions belonging to  $H^1(\Omega)$ . The space of  $p \times q$  matrices is denoted by  $\mathbb{R}^{p \times q}$ ,  $\mathbb{R}^p := \mathbb{R}^{p \times 1}$  is the space of  $p$ -dimensional vectors, and  $\mathbb{R}^p_+$  denotes the non-negative orthant of  $\mathbb{R}^p$ . Bold characters are used for vectors and matrices. By  $\mathbf{0}$  we denote the zero matrix, or vector. Further  $\mathbf{I} \in \mathbb{R}^{p \times p}$  stands for the identity matrix. The scalar product of two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$ , is denoted by  $\mathbf{a} \cdot \mathbf{b}$ . If  $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{p \times q}$ ,  $\mathbf{B} = (b_{ij}) \in \mathbb{R}^{p \times q}$  are two matrices then  $\mathbf{A} : \mathbf{B} = a_{ij}b_{ij}$  (summation convention is used). Caligraphic symbols will be used for index sets, for instance:  $\mathcal{N} = \{1, 2, \dots, n\}$  and  $\mathcal{A}, \mathcal{I} \subseteq \mathcal{N}$ . If  $\mathbf{N} \in \mathbb{R}^{p \times q}$  and  $\mathcal{A} \neq \emptyset$ , then  $\mathbf{N}_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}| \times q}$  stands for the matrix given by the rows of  $\mathbf{N}$  whose indices belong to  $\mathcal{A}$ . If  $\mathbf{A} \in \mathbb{R}^{p \times p}$  is symmetric, positive definite with the smallest, and largest eigenvalues  $0 < \alpha_{\min} \leq \alpha_{\max}$ , respectively, then the spectral condition number of  $\mathbf{A}$  is  $cond(\mathbf{A}) = \alpha_{\max}/\alpha_{\min}$ . The indicator matrix of a subset  $\mathcal{S} \subseteq \mathcal{N}$  is the diagonal matrix  $\mathbf{D}(\mathcal{S}) = diag(s_1, s_2, \dots, s_p) \in \mathbb{R}^{p \times p}$ , where  $s_i = 1$  for  $i \in \mathcal{S}$  and  $s_i = 0$  if  $i \notin \mathcal{S}$ . Finally, the symbol  $\|\cdot\|$  stands for the Euclidean norm in  $\mathbb{R}^p$ ,  $p \geq 2$ .

## 2. Setting of the Problem

The aim of this section is to recall the classical formulation of the Stokes system with leak boundary conditions and to present several weak formulations of this problem together with main existence/uniqueness results.

Let  $\Omega \subset \mathbb{R}^d$ ,  $d \in \{2, 3\}$  be a bounded domain with the Lipschitz boundary  $\partial\Omega$  which is decomposed into two nonempty, nonoverlapping parts  $\Gamma$  and  $S$  both open in  $\partial\Omega$ . The classical formulation of the problem reads as follows: find the velocity vector  $u : \Omega \rightarrow \mathbb{R}^d$  and the pressure  $p : \Omega \rightarrow \mathbb{R}$  satisfying the following system of differential equations and boundary conditions:

$$\left. \begin{aligned} -2\mu \operatorname{div}(\mathbb{D}u) + \nabla p &= f && \text{in } \Omega, \\ \operatorname{div} u &= 0 && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma, \\ u_t &= 0 && \text{on } S, \\ |\sigma_n| &\leq g && \text{on } S, \\ u_n \neq 0 \Rightarrow |\sigma_n| = g \ \&\ \ u_n \sigma_n \leq 0 && \text{on } S. \end{aligned} \right\} \tag{4}$$

Here  $\mu > 0$  is the viscosity of the fluid,  $f : \Omega \rightarrow \mathbb{R}^d$ ,  $g : S \rightarrow \mathbb{R}_+$  denote an external force, and a nonnegative leak threshold, respectively. Further  $\mathbb{D}u = \frac{1}{2}(\nabla u + (\nabla u)^T)$  is the symmetric part of the gradient of  $u$ ,  $n$  stands for the unit, outward normal vector to  $\partial\Omega$ ,  $u_n = u \cdot n$ ,  $u_t = u - u_n n$  is the normal, and the tangential component of  $u$  on  $\partial\Omega$ ,

respectively. Similarly,  $\sigma_n = 2\mu \mathbb{D}un \cdot n - p$  is the normal component of the stress vector  $\sigma = 2\mu \mathbb{D}un - pn$  on  $\partial\Omega$ . It is worth noticing that unlike the shear stress  $\sigma_t = \sigma - \sigma_n n$ , the normal component  $\sigma_n$  depends explicitly on the pressure  $p$ . The boundary conditions on  $S$  express the fact that there is no slip along  $S$  and the fluid can escape from  $\Omega$  only if the threshold  $g$  is attained, i.e.,  $|\sigma_n| = g$ . In addition, the sign of  $\sigma_n$  is opposite to the one of  $u_n$ .

**Remark 1.** The last two conditions in (4) can be written in the following equivalent way:

$$|\sigma_n| \leq g, \quad \sigma_n u_n + g|u_n| = 0 \quad \text{on } S \tag{5}$$

or

$$-\sigma_n \in g\partial|u_n| \quad \text{on } S,$$

where the symbol  $\partial$  stands for the subgradient of convex functions.

To present variational formulations of (4) we shall need the following function spaces and forms:

$$\begin{aligned} W(\Omega) &= \{v \in (H^1(\Omega))^d \mid v = 0 \text{ on } \Gamma\}, \\ V(\Omega) &= \{v \in W(\Omega) \mid v_t = 0 \text{ on } S\}, \\ V_{div}(\Omega) &= \{v \in V(\Omega) \mid \operatorname{div} v = 0 \text{ in } \Omega\}, \end{aligned}$$

and

$$\begin{aligned} a(u, v) &= 2\mu \int_{\Omega} \mathbb{D}u : \mathbb{D}v \, dx, \quad u, v \in (H^1(\Omega))^d, \\ b(v, q) &= - \int_{\Omega} q \operatorname{div} v \, dx, \quad q \in L^2(\Omega), v \in (H^1(\Omega))^d, \\ (f, v) &= \int_{\Omega} f \cdot v \, dx, \quad f \in (L^2(\Omega))^d, v \in (H^1(\Omega))^d, \\ j(v_n) &= \int_S g|v_n| \, ds, \quad g \in L^2_+(S), v \in (H^1(\Omega))^d. \end{aligned}$$

By the weak velocity–pressure formulation of (4) we call a problem of finding a pair  $(u, p) \in V(\Omega) \times L^2(\Omega)$  such that

$$\left. \begin{aligned} a(u, v - u) + b(v - u, p) + j(v_n) - j(u_n) &\geq (f, v - u) \quad \forall v \in V(\Omega), \\ b(u, q) &= 0 \quad \forall q \in L^2(\Omega). \end{aligned} \right\} \tag{6}$$

From Green’s formula with an appropriate choice of test functions  $v$  one can verify that the classical and weak velocity–pressure formulations are equivalent for sufficiently smooth solutions  $(u, p)$  ([5]).

Restricting to test functions  $v \in V_{div}(\Omega)$  we arrive at the weak velocity formulation of (4):

$$u \in V_{div}(\Omega) : \quad a(u, v - u) + j(v_n) - j(u_n) \geq (f, v - u) \quad v \in V_{div}(\Omega), \tag{7}$$

which is equivalent to the following minimization problem:

$$u \in \operatorname{argmin}\{J(v), v \in V_{div}(\Omega)\}, \tag{8}$$

where

$$J(v) = \frac{1}{2}a(v, v) + j(v_n) - (f, v).$$

Problem (8) is the variational inequality of the second kind and it has a unique solution  $u$  [15].

The following existence/uniqueness result has been established in [4,5].

**Theorem 1.** *Problem (6) has a solution  $(u, p)$  for any  $f \in (L^2(\Omega))^d$  and  $g \in L^2_+(S)$ . The velocity vector  $u$  is unique. If there is a set  $\omega \subseteq S$ ,  $meas_{d-1}\omega > 0$  such that  $|\sigma_n| = g$  on  $\omega$  then the pressure  $p$  is unique, as well, otherwise  $p$  is determined up to constants  $c$  satisfying  $|\sigma_n + c| \leq g$  a.e. on  $S$ .*

**Remark 2.** *If, besides  $\Gamma$  and  $S$  there is a portion  $\Gamma_N \subseteq \partial\Omega$ ,  $meas_{d-1}\Gamma_N > 0$ , where the value of  $\sigma$  is prescribed, i.e.,  $\sigma = \sigma_N \in (L^2(\Gamma_N))^d$  given, then also  $p$  is unique.*

**Remark 3.** *It has been shown in [5] that, if  $p$  is not unique, then the range of the admissible constants  $c$  satisfying  $|\sigma_n + c| \leq g$  a.e. on  $S$  is the interval  $\langle k_1, k_2 \rangle$ , where  $k_1 = \sup_S(\sigma_n - g)$ ,  $k_2 = \inf_S(\sigma_n + g)$ , provided that the solution  $(u, p)$  is smooth enough.*

The last formulation presented in this section is the variant of the velocity–pressure formulation (6) in which the Lagrange multipliers are used to release the no-slip condition  $u_t = 0$  on  $S$  and to regularize the nonsmooth term  $j$ . To this end we shall need the following trace spaces:

$$\begin{aligned} X_t &= \{ \phi \in (L^2(S))^{d-1} \mid \exists v \in W(\Omega) : \phi = v_t \text{ on } S \}, \\ X_n &= \{ \mu \in L^2(S) \mid \exists v \in W(\Omega) : \mu = v_n \text{ on } S \} \end{aligned}$$

and their duals  $X'_t, X'_n$ . If  $\Omega$  is the domain with a smooth boundary ( $C^{1,1}$ , e.g.), then  $X_t = (H^{\frac{1}{2}}(S))^{d-1}$  and  $X_n = H^{\frac{1}{2}}(S)$ .

**Remark 4.** *If the solution  $(u, p)$  is sufficiently smooth, then the normal and shear components of  $\sigma$  on  $S$  have been defined pointwisely at the beginning of this section. If no regularity assumptions are imposed on  $(u, p)$  then  $\sigma_n$  and  $\sigma_t$  have to be understood as elements of  $X'_n$  and  $X'_t$ , respectively, defined by*

$$\langle \sigma_t, v_t \rangle := a(u, v) + b(v, p) - (f, v) \quad \forall v \in W(\Omega), v_n = 0 \text{ on } S, \tag{9}$$

$$\langle \sigma_n, v_n \rangle := a(u, v) + b(v, p) - (f, v) \quad \forall v \in W(\Omega), v_t = 0 \text{ on } S, \tag{10}$$

where the  $\langle \cdot, \cdot \rangle$  stands for the duality pairings. It is easy to see that the right hand sides of (9) and (10) depend only on  $v_t$ , and  $v_n$  on  $S$ , respectively, since  $(u, p) \in V_{div}(\Omega) \times L^2(\Omega)$  solves the Stokes system in a weak sense. Moreover,  $\sigma_n$  belongs to  $L^2(S)$  and it satisfies the last two conditions on  $S$  in (4). Indeed,

$$j(v_n) - j(u_n) \geq \langle -\sigma_n, v_n - u_n \rangle \quad \forall v \in V(\Omega) \Leftrightarrow -\sigma_n \in \partial j(u_n) \text{ a.e. on } S.$$

The duality pairing  $\langle \sigma_n, v_n \rangle$  is defined by the  $L^2(S)$ -scalar product in this case.

Finally, let

$$K = \{ v \in L^2(S) \mid |v| \leq g \text{ a.e. on } S \}$$

be a closed, convex subset of  $L^2(S)$ . To derive the new formulation we use the saddle-point approach. It holds:

$$\begin{aligned} J(u) &= \min_{v \in V_{div}(\Omega)} J(v) = \min_{v \in V(\Omega)} \sup_{q \in L^2(\Omega)} \{ J(v) + b(v, q) \} \\ &= \min_{v \in W(\Omega)} \sup_{q \in L^2(\Omega)} \sup_{\substack{\tau \in X'_t \\ v \in K}} \{ J(v) + b(v, q) - \langle \tau, v_t \rangle - \langle v, v_n \rangle \} \\ &:= \min_{v \in W(\Omega)} \sup_{q \in L^2(\Omega)} \sup_{\substack{\tau \in X'_t \\ v \in K}} \mathcal{L}(v, q, \tau, v), \end{aligned}$$

where  $\mathcal{L} : Q \rightarrow \mathbb{R}$ ,  $Q = W(\Omega) \times L^2(\Omega) \times X'_t \times K$  is the Lagrangian. Suppose there exists a saddle-point  $(u^*, p^*, \tau^*, \nu^*) \in Q$  of  $\mathcal{L}$  on  $Q$ :

$$\mathcal{L}(u^*, q, \tau, \nu) \leq \mathcal{L}(u^*, p^*, \tau^*, \nu^*) \leq \mathcal{L}(v, p^*, \tau^*, \nu^*),$$

that holds for every  $(v, p, \tau, \nu) \in Q$ .

It is known (see [2]) that the necessary and sufficient condition for  $(u^*, p^*, \tau^*, \nu^*)$  to be a saddle-point of  $\mathcal{L}$  on  $Q$  is the satisfaction of the Karush–Kuhn–Tucker conditions:

$$\left. \begin{aligned} a(u^*, v) + b(v, p^*) &= (f, v) + \langle \tau^*, v_t \rangle + \langle \nu^*, \nu_n \rangle \quad \forall v \in W(\Omega), \\ b(u^*, q) &= 0 \quad \forall q \in L^2(\Omega), \\ \langle \tau, u^*_t \rangle &= 0 \quad \forall \tau \in X'_t, \\ \langle \nu + \nu^*, u^*_n \rangle &\leq 0 \quad \forall \nu \in K. \end{aligned} \right\} \quad (11)$$

It is easy to see that  $(u^*, p^*) \in V_{div}(\Omega) \times L^2(\Omega)$  satisfies the Stokes system and the boundary conditions on  $\partial\Omega$ , i.e.,  $(u^*, p^*)$  solves (6). Moreover,  $\tau^* = \sigma^*_t$ ,  $\nu^* = \sigma^*_n$  is the corresponding shear, and normal stress on  $S$ , respectively. The opposite assertion is also trivial: if  $(u, p) \in V(\Omega) \times L^2(\Omega)$  is a solution to (6), then the quadruplet  $(u, p, \sigma_t, \sigma_n) \in W(\Omega) \times L^2(\Omega) \times X'_t \times K$  solves (11), where  $\sigma_t, \sigma_n$  is the corresponding shear, and normal stress on  $S$ , respectively.

**Theorem 2.** A couple  $(u, p) \in V(\Omega) \times L^2(\Omega)$  solves (6) if and only if the quadruplet  $(u, p, \sigma_t, \sigma_n) \in Q$  is a solution to (11).

Proof is straightforward. For more details we refer to [31].

**Remark 5.** In the discrete counterpart of the problem we shall use the following more general form  $j_\kappa$  instead of  $j$ :

$$j_\kappa(v_n, w_n) = \int_S g|v_n| + \kappa w_n v_n ds, \quad g, \kappa \in L^2_+(S), \quad v, w \in (H^1(\Omega))^d.$$

Then the weak formulation (6) is modified as follows: find a pair  $(u, p) \in V(\Omega) \times L^2(\Omega)$  such that

$$\left. \begin{aligned} a(u, v - u) + b(v - u, p) + j_\kappa(v_n, u_n) - j_\kappa(u_n, u_n) &\geq (f, v - u) \quad \forall v \in V(\Omega), \\ b(u, q) &= 0 \quad \forall q \in L^2(\Omega). \end{aligned} \right\} \quad (12)$$

The function  $\kappa : S \rightarrow \mathbb{R}_+$  characterizes pore opening. The respective leak condition reads as

$$|\sigma_n + \kappa u_n| \leq g, \quad (\sigma_n + \kappa u_n)u_n + g|u_n| = 0 \quad \text{on } S. \quad (13)$$

Therefore the substitution  $\tilde{\sigma}_n = \sigma_n + \kappa u_n$  transforms the leak condition (13) formally into (5) so that all results of this section remain valid also for problem (12).

### 3. Mixed Finite Element Method and Algebraic Formulations

In this section we present the algebraic form of the problem, which will be used in the subsequent parts of the paper for numerical solution. It is based on the discretization of the velocity–pressure formulation (12) by a mixed finite element method. Next we shall suppose that  $\Omega$  is a polygonal ( $d = 2$ ), or polyhedral ( $d = 3$ ) domain. If not, then  $\Omega$  is approximated by them. Let  $V_h \subset W_h, Q_h$  be finite element approximations of  $V(\Omega), W(\Omega)$ , and  $L^2(\Omega)$ , respectively, such that the bilinear form  $b$  satisfies the Babuska–Brezzi condition:

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

where  $\beta > 0$  does not depend on the discretization parameter  $h$ .

The mixed finite element approximation of (12) reads as follows:

$$\left. \begin{aligned} & \text{Find } (u_h, p_h) \in V_h \times Q_h \text{ such that} \\ & a(u_h, v_h - u_h) + b(v_h - u_h, p_h) + j_\kappa(v_h, u_h) - j_\kappa(u_h, u_h) \geq (f, v_h - u_h) \quad \forall v_h \in V_h, \\ & b(u_h, q_h) = 0 \quad \forall q_h \in Q_h. \end{aligned} \right\} \quad (14)$$

In computations we use the P1-bubble/P1 finite element pair [32] on a regular partition  $\mathcal{T}_h$  of  $\bar{\Omega}$  into triangles, tetrahedras for  $d = 2$ , and 3, respectively. On  $\mathcal{T}_h$  we define the finite element spaces:

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

where  $P_1(T)$ ,  $B(T)$  are the spaces of polynomials of degree one and of bubble functions of degree three ( $d = 2$ ) and four ( $d = 3$ ) on  $T \in \mathcal{T}_h$ , respectively.

The algebraic counterpart of (14) reads as follows:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^{n_p} \text{ such that} \\ & \mathbf{u}^T \mathbf{A}(\mathbf{v} - \mathbf{u}) + (\mathbf{v} - \mathbf{u})^T \mathbf{B}^T \mathbf{p} + j_\kappa^h(\mathbf{v}, \mathbf{u}) - j_\kappa^h(\mathbf{u}, \mathbf{u}) \geq \mathbf{b}^T(\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}, \\ & \mathbf{B}\mathbf{u} = \mathbf{0} \end{aligned} \right\} \quad (15)$$

with

$$\begin{aligned} \mathbb{V} &= \{\mathbf{v} \in \mathbb{R}^{d(n_u+n_q)} : \mathbf{T}\mathbf{v} = \mathbf{0}\}, \\ j_\kappa^h(\mathbf{v}, \mathbf{w}) &= \mathbf{g}^T |\mathbf{N}\mathbf{v}| + \mathbf{w}^T \mathbf{N}^T \mathbf{D}_\kappa \mathbf{N}\mathbf{v}, \end{aligned} \quad (16)$$

where the absolute value is understood componentwisely,  $\mathbf{D}_\kappa = \text{diag}(\kappa)$ ,  $\mathbf{g}, \kappa \in \mathbb{R}_+^{n_s}$  with  $n_s$  being the number of nodes belonging to  $\bar{S} \setminus \bar{\Gamma}$ . Here,  $n_p, n_u$  stand for the number of the finite element nodes of  $\mathcal{T}_h$  in  $\bar{\Omega}$ , and  $\bar{\Omega} \setminus \bar{\Gamma}$ , respectively, while  $n_q$  denotes the number of the triangles/tetrahedras in  $\mathcal{T}_h$ . Further,  $\mathbf{A} \in \mathbb{R}^{d(n_u+n_q) \times d(n_u+n_q)}$  is a symmetric, positive definite diffusion matrix,  $\mathbf{B} \in \mathbb{R}^{n_p \times d(n_u+n_q)}$  is a full row-rank divergence matrix, and  $\mathbf{b} \in \mathbb{R}^{d(n_u+n_q)}$  is a vector of nodal forces. The unit outward, normal vector at the  $i$ -th node of  $\bar{S} \setminus \bar{\Gamma}$  defines nonzero entries in the  $i$ -th row of the matrix  $\mathbf{N} \in \mathbb{R}^{n_s \times d(n_u+n_q)}$ . For  $d = 2$ , the tangential vector defines nonzero entries in the  $i$ -th row of the matrix  $\mathbf{T} \in \mathbb{R}^{n_s \times 2(n_u+n_q)}$ . For  $d = 3$ , there are two mutually orthonormal tangential vectors that define nonzero entries in the  $i$ -th rows of the matrices  $\mathbf{T}_1, \mathbf{T}_2 \in \mathbb{R}^{n_s \times 3(n_u+n_q)}$ . In this case,  $\mathbf{T} = (\mathbf{T}_1^T, \mathbf{T}_2^T)^T$ .

Denote

$$\mathcal{N} = \{1, \dots, n_s\}$$

that is the local index set of the nodes belonging to  $\bar{S} \setminus \bar{\Gamma}$ . The algebraic version  $j_\kappa^h$  of  $j$  introduced above is the result of numerical integration, assuming that  $g, \kappa \in C(\bar{S})$ . For  $d = 3$  we use the following integration formula on the triangular element  $\tau \in \mathcal{T}_h|_{\bar{S}}$  with the vertices  $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \bar{S}$ :

$$\begin{aligned} \int_\tau g |v_n| ds &\approx |\tau| g(\mathbf{x}_\tau) \frac{1}{3} (|v_n(\mathbf{x}_i)| + |v_n(\mathbf{x}_j)| + |v_n(\mathbf{x}_k)|) \\ &= |\tau| g(\mathbf{x}_\tau) \frac{1}{3} (|(\mathbf{N}\mathbf{v})_i| + |(\mathbf{N}\mathbf{v})_j| + |(\mathbf{N}\mathbf{v})_k|), \end{aligned} \quad (17)$$

where  $|\tau| = \text{meas } \tau$ ,  $\mathbf{x}_\tau$  is the centre of gravity of  $\tau$ , and  $(\mathbf{Nv})_i$  is the  $i$ -th component of  $\mathbf{Nv}$ . Summing up (17) over all  $\tau \in \mathcal{T}_{h|\bar{S}}$  we arrive at:

$$\int_S g|v_n| ds \approx \mathbf{g}^T |\mathbf{Nv}| \quad \text{with} \quad g_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{3} g(\mathbf{x}_{\tau_j^i}), \quad i \in \mathcal{N},$$

where  $\mathbf{g} = (g_1, \dots, g_{n_s})^T$  and  $n_i$  is the number of the triangles  $\tau_j^i \in \mathcal{T}_{h|\bar{S}}$  sharing  $\mathbf{x}_i \in \bar{S} \setminus \bar{\Gamma}$  as the common vertex. In the same way we get:

$$\int_S \kappa w_n v_n ds \approx \mathbf{w}^T \mathbf{N}^T \mathbf{D}_\kappa \mathbf{Nv} \quad \text{with} \quad \kappa_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{3} \kappa(\mathbf{x}_{\tau_j^i}), \quad i \in \mathcal{N}, \tag{18}$$

where  $\kappa = (\kappa_1, \dots, \kappa_{n_s})^T$ . We proceed analogously for  $d = 2$  when  $\tau \in \mathcal{T}_{h|\bar{S}}$  is a line segment,  $n_i \leq 2$ , and

$$g_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{2} g(\mathbf{x}_{\tau_j^i}), \quad \kappa_i := \sum_{j=1}^{n_i} \frac{|\tau_j^i|}{2} \kappa(\mathbf{x}_{\tau_j^i}), \quad i \in \mathcal{N}. \tag{19}$$

From this and (16)

$$j_\kappa^h(\mathbf{v}, \mathbf{u}) - j_\kappa^h(\mathbf{u}, \mathbf{u}) = \mathbf{g}^T |\mathbf{Nv}| - \mathbf{g}^T |\mathbf{Nu}| + \mathbf{u}^T \mathbf{N}^T \mathbf{D}_\kappa \mathbf{N}(\mathbf{v} - \mathbf{u}).$$

Thus problem (15) can be written in the following equivalent form:

$$\left. \begin{aligned} & \text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^{n_p} \text{ such that} \\ & \mathbf{u}^T \mathbf{A}_\kappa (\mathbf{v} - \mathbf{u}) + (\mathbf{v} - \mathbf{u})^T \mathbf{B}^T \mathbf{p} + j^h(\mathbf{v}) - j^h(\mathbf{u}) \geq \mathbf{b}^T (\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}, \\ & \mathbf{B}\mathbf{u} = \mathbf{0}, \end{aligned} \right\} \tag{20}$$

where

$$\mathbf{A}_\kappa = \mathbf{A} + \mathbf{N}^T \mathbf{D}_\kappa \mathbf{N} \tag{21}$$

and

$$j^h(\mathbf{v}) = \mathbf{g}^T |\mathbf{Nv}|.$$

It is easy to show that the first component  $\mathbf{u}$  in (20) solves the discrete velocity formulation being the algebraic version of (8):

$$\left. \begin{aligned} & \text{Find } \mathbf{u} \in \mathbb{V}_\mathbf{B} \text{ such that} \\ & \mathcal{J}(\mathbf{u}) \leq \mathcal{J}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{V}_\mathbf{B}, \end{aligned} \right\} \tag{22}$$

where

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

Formulation (22) is not suited for direct computations, as the constraints in  $\mathbb{V}_\mathbf{B}$  can be hardly handled for large-scale problems. Moreover, the function  $\mathcal{J}$  is nondifferentiable caused by the term  $j^h$ . To overcome these difficulties, we will use the dual formulation of (22) and derive the discrete counterpart of (11).



We introduce three Lagrange multipliers:  $\boldsymbol{\mu}_n \in \mathbb{R}^{n_s}$  to regularize  $j^h$ ,  $\boldsymbol{\mu}_t \in \mathbb{R}^{(d-1)n_s}$ ,  $\mathbf{q} \in \mathbb{R}^{n_p}$  to release the discrete stick, and incompressibility conditions, respectively and denote  $\boldsymbol{\mu} = (\boldsymbol{\mu}_n^T, \boldsymbol{\mu}_t^T, \mathbf{q}^T)^T$ . Let

$$\begin{aligned} \mathbb{X} &= \mathbb{X}_n \times \mathbb{R}^{(d-1)n_s+n_p}, \\ \mathbb{X}_n &= \{\boldsymbol{\mu}_n \in \mathbb{R}^{n_s} : |\mu_{n,i}| \leq g_i, i \in \mathcal{N}\}, \end{aligned}$$

where  $\mu_{n,i}$  is the  $i$ -th component of  $\boldsymbol{\mu}_n \in \mathbb{R}^{n_s}$ . The term  $j^h$  can be written as follows:

$$j^h(\mathbf{v}) = \sum_{i \in \mathcal{N}} \max_{|\mu_{n,i}| \leq g_i} \mu_{n,i}(\mathbf{N}\mathbf{v})_i = \max_{\boldsymbol{\mu}_n \in \mathbb{X}_n} \boldsymbol{\mu}_n^T \mathbf{N}\mathbf{v}.$$

Thus:

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

where the Lagrangian  $\mathcal{L}^\# : \mathbb{R}^{d(n_u+n_q)} \times \mathbb{X} \rightarrow \mathbb{R}$  associated with the velocity Formulation (22) is defined for  $(\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{d(n_u+n_q)} \times \mathbb{X}$  by:

$$\mathcal{L}^\#(\mathbf{v}, \boldsymbol{\mu}) = \frac{1}{2} \mathbf{v}^T \mathbf{A}_k \mathbf{v} - \mathbf{v}^T \mathbf{b} + \boldsymbol{\mu}_n^T \mathbf{N}\mathbf{v} + \boldsymbol{\mu}_t^T \mathbf{T}\mathbf{v} + \mathbf{q}^T \mathbf{B}\mathbf{v}.$$

Until now, the velocity vector  $\mathbf{v} \in \mathbb{R}^{d(n_u+n_q)}$  incorporates  $dn_q$  bubble components. These components are usually eliminated before the computational process [33]. In our case, we perform this elimination in a saddle-point formulation for  $\mathcal{L}^\#$ , which leads to the reduced Lagrangian  $\mathcal{L} : \mathbb{R}^{dn_u} \times \mathbb{X} \rightarrow \mathbb{R}$  defined by:

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\mu}) = \frac{1}{2} \mathbf{v}^T \mathbf{A}_k \mathbf{v} - \mathbf{v}^T \mathbf{b} + \boldsymbol{\mu}_n^T \mathbf{N}\mathbf{v} + \boldsymbol{\mu}_t^T \mathbf{T}\mathbf{v} + \mathbf{q}^T \mathbf{B}\mathbf{v} - \frac{1}{2} \mathbf{q}^T \mathbf{E}\mathbf{q} - \mathbf{c}^T \mathbf{q}, \quad (\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{dn_u} \times \mathbb{X}.$$

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:  $\mathbf{A}_k \in \mathbb{R}^{dn_u \times dn_u}$ ,  $\mathbf{N} \in \mathbb{R}^{n_s \times dn_u}$ ,  $\mathbf{T} \in \mathbb{R}^{(d-1)n_s \times dn_u}$ ,  $\mathbf{B} \in \mathbb{R}^{n_p \times dn_u}$ , and  $\mathbf{b} \in \mathbb{R}^{dn_u}$ . Note that these matrices preserve the same properties as before the elimination, especially, the expression (21) remains valid with  $\mathbf{A} \in \mathbb{R}^{dn_u \times dn_u}$  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  $\mathbf{c} \in \mathbb{R}^{n_p}$  is due to this elimination. Note that  $\mathbf{E}$  has the defect one and the eigenvector whose all components are equal to 1 corresponds to zero eigenvalue.

The new saddle-point formulation of (22) reads as follows:

$$\left. \begin{aligned} \text{Find } (\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{dn_u} \times \mathbb{X} \text{ such that} \\ \mathcal{L}(\mathbf{u}, \boldsymbol{\mu}) \leq \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) \leq \mathcal{L}(\mathbf{v}, \boldsymbol{\lambda}) \quad \forall (\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{dn_u} \times \mathbb{X}, \end{aligned} \right\} \quad (23)$$

or, equivalently,

$$\left. \begin{aligned} \text{Find } (\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{dn_u} \times \mathbb{X} \text{ such that} \\ \mathbf{A}_k \mathbf{u} + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{T}^T \boldsymbol{\lambda}_t + \mathbf{B}^T \mathbf{p} - \mathbf{b} = \mathbf{0}, \\ \mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c} = \mathbf{0}, \\ \mathbf{T}\mathbf{u} = \mathbf{0}, \\ |\lambda_{n,i}| \leq g_i, \\ (\mathbf{N}\mathbf{u})_i \neq 0 \Rightarrow |\lambda_{n,i}| = g_i \ \& \ (\mathbf{N}\mathbf{u})_i \lambda_{n,i} \geq 0, \end{aligned} \right\} i \in \mathcal{N}, \quad (24)$$

where  $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_n^T, \boldsymbol{\lambda}_t^T, \mathbf{p}^T)^T$ .

**Remark 6.** Here,  $\lambda_n, \lambda_t, \mathbf{p}$  are the discrete counterparts of  $-\tilde{\sigma}_n$  (see Remark 5),  $-\sigma_t$ , and  $p$ , respectively. The change of the sign at  $\tilde{\sigma}_n$  and  $\sigma_t$  is for convenience of readers. One can see at a glance that the dual Hessian will be positive definite, as  $\mathbf{C}$  introduced below is assembled from matrices without any negative signs.

Now we eliminate the unknown  $\mathbf{u}$ . Denoting  $\mathbf{C} = (\mathbf{N}^T, \mathbf{T}^T, \mathbf{B}^T)^T$ , the first equation in (24) yields  $\mathbf{u} = \mathbf{A}_\kappa^{-1}(\mathbf{b} - \mathbf{C}^T \lambda)$ . Inserting  $\mathbf{u}$  into the first inequality in (23), we arrive at the dual formulation of (22):

$$\left. \begin{aligned} & \text{Find } \lambda \in \mathbb{X} \text{ such that} \\ & \mathcal{S}(\lambda) \leq \mathcal{S}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathbb{X}, \end{aligned} \right\} \tag{25}$$

where  $\mathcal{S} : \mathbb{R}^{dn_s+n_p} \rightarrow \mathbb{R}$ ,

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

with  $\mathbf{F}_\kappa = \mathbf{C} \mathbf{A}_\kappa^{-1} \mathbf{C}^T + \text{diag}(\mathbf{0}, \mathbf{0}, \mathbf{E})$  being symmetric, positive definite provided that  $\Gamma_N$  from Remark 2 is nonempty, and  $\mathbf{d} = \mathbf{C} \mathbf{A}_\kappa^{-1} \mathbf{b} - (\mathbf{0}^T, \mathbf{0}^T, \mathbf{c}^T)^T$ . In what it follows we will assume that  $\mathbf{F}_\kappa$  is non-singular. The case  $\Gamma_N = \emptyset$  will be discussed in Example 3.

It should be noted that (25) is more convenient for numerical solution unlike (22) as the function  $\mathcal{S}$  is differentiable (quadratic) and the feasible set  $\mathbb{X}$  is defined by the box constraints. The solution to (25) may be computed by an appropriate algorithm of the constrained minimization. Another way how to solve our problem originates from (24) but with the algebraic leak boundary conditions expressed by projections on appropriate convex sets. The resulting system of nonsmooth algebraic equations can be solved by the nonsmooth Newton type method.

#### 4. Algorithms

The aim of this section is to present main ideas of the path-following and the semismooth Newton methods that turned out to be highly efficient for solving frictional contact problems of solid mechanics [34–36]. These algorithms have been adapted for the Stokes system with the threshold slip boundary conditions in 2D [27,37] and 3D [28]. The situation with the threshold leak conditions is different since only box constraints are imposed on the individual components of  $\lambda_n$  in 2D, as well as in 3D problems. In view of this fact the same implementation of the path-following method can be used in both cases. Also the semismooth Newton method works only with projections on compact intervals unlike the threshold slip boundary conditions in 3D in which case the components of  $\lambda_t$  are pairwise subject to spherical constraints and projections on circles are needed. Consequently, the finite termination property for the semismooth Newton method can be expected also for 3D problems.

##### 4.1. Path-Following Method

Let  $L : \mathbb{R}^{dn_s+n_p} \times \mathbb{R}_+^{2n_s} \rightarrow \mathbb{R}$  be the Lagrangian to (25):

$$L(\boldsymbol{\mu}, \mathbf{v}) = \mathcal{S}(\boldsymbol{\mu}) + \mathbf{v}_1^T (-\mathbf{g} - \boldsymbol{\mu}_n) + \mathbf{v}_2^T (\boldsymbol{\mu}_n - \mathbf{g}), \tag{26}$$

where  $\mathbf{v} := (\mathbf{v}_1^T, \mathbf{v}_2^T)^T \in \mathbb{R}^{2n_s}$ ,  $\mathbf{v} \geq \mathbf{0}$ , is the Lagrange multiplier vector releasing the box constraints appearing in  $\mathbb{X}$ . Let  $\mathbf{z}_k := -\nabla_{\mathbf{v}_k} L(\boldsymbol{\mu}, \mathbf{v})$ ,  $k = 1, 2$ ,  $\mathbf{z} := (\mathbf{z}_1^T, \mathbf{z}_2^T)^T \in \mathbb{R}^{2n_s}$  be the new variables and define the function  $\mathbf{H} : \mathbb{R}^{(d+4)n_s+n_p} \rightarrow \mathbb{R}^{(d+4)n_s+n_p}$ ,

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

where  $\boldsymbol{\omega} = (\boldsymbol{\mu}^T, \mathbf{v}^T, \mathbf{z}^T)^T \in \mathbb{R}^{(d+4)n_s+n_p}$ ,  $\mathbf{M} = \text{diag}(\mathbf{v})$ ,  $\mathbf{Z} = \text{diag}(\mathbf{z})$ , and  $\mathbf{e} \in \mathbb{R}^{n_s}$  is the vector whose all components are equal to 1. The solution  $\lambda$  to (25) is the first component of the vector  $\bar{\boldsymbol{\omega}} = (\lambda^T, \bar{\mathbf{v}}^T, \bar{\mathbf{z}}^T)^T$ , which satisfies

$$\mathbf{H}(\bar{\boldsymbol{\omega}}) = \mathbf{0}, \quad \bar{\mathbf{v}} \geq \mathbf{0}, \quad \bar{\mathbf{z}} \geq \mathbf{0}, \tag{27}$$

since (27) is equivalent to the Karush-Khun-Tucker conditions for  $(\lambda, \bar{v})$  to be a saddle-point of (26) on  $\mathbb{R}^{d n_s + n_p} \times \mathbb{R}_+^{2 n_s}$ .

To derive the path-following algorithm, we replace (27) by the following perturbed problem:

$$\left. \begin{aligned} \text{Find } \omega_\tau &= (\lambda_\tau^T, \nu_\tau^T, \mathbf{z}_\tau^T)^T \in \mathbb{R}^{(d+4)n_s+n_p} \text{ such that} \\ \mathbf{H}(\omega_\tau) &= (\mathbf{0}^T, \mathbf{0}^T, \tau \mathbf{e}^T)^T, \nu_\tau > \mathbf{0}, \mathbf{z}_\tau > \mathbf{0}, \end{aligned} \right\} \tag{28}$$

where  $\tau \in \mathbb{R}_+$ . Solutions  $\omega_\tau$  to (28) define a curve  $\mathcal{C}(\tau)$  in  $\mathbb{R}^{(d+4)n_s+n_p}$  called the *central path*. This curve approaches  $\bar{\omega}$  when  $\tau$  tends to zero. We combine the damped Newton method used for solving the equation in (28) with an appropriate change of  $\tau$  which guarantees that the iterations belong to a neighbourhood  $\mathcal{N}(c_1, c_2)$  of  $\mathcal{C}(\tau)$  defined by:

$$\begin{aligned} \mathcal{N}(c_1, c_2) &= \{ \omega = (\boldsymbol{\mu}^T, \boldsymbol{\nu}^T, \mathbf{z}^T)^T \in \mathbb{R}^{(d+4)n_s+n_p} : \nu_i z_i \geq c_1 \vartheta, i = 1, \dots, 2n_s, \\ &\quad \boldsymbol{\nu} \geq \mathbf{0}, \mathbf{z} \geq \mathbf{0}, \|\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \boldsymbol{\nu})\| \leq c_2 \vartheta, \|\nabla_{\boldsymbol{\nu}} L(\boldsymbol{\mu}, \boldsymbol{\nu}) + \mathbf{z}\| \leq c_2 \vartheta \}, \end{aligned}$$

where  $c_1 \in (0, 1]$ ,  $c_2 \geq 1$ , and  $\vartheta := \vartheta(\omega) = \boldsymbol{\nu}^T \mathbf{z} / (2n_s)$ . In the  $k$ -th iteration, we replace  $\tau := \tau^{(k)}$  by the product of  $\vartheta^{(k)} := \vartheta(\omega^{(k)})$  with the centering parameter  $c^{(k)}$  chosen as in [38]. The algorithm uses also the *Armijo-type condition* (30) ensuring that the sequence  $\{\vartheta^{(k)}\}$  is monotonically decreasing. By  $\mathbf{J}(\omega)$  in (29), we denote the Jacobian matrix of  $\mathbf{H}$  at  $\omega$ .

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  $\omega^{(0)} \in \mathcal{N}(c_1, c_2)$  and set  $k := 0$ .

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

(ii) Solve

$$\mathbf{J}(\omega^{(k)}) \Delta \omega^{(k+1)} = -\mathbf{H}(\omega^{(k)}) + (\mathbf{0}^T, \mathbf{0}^T, c^{(k)} \vartheta^{(k)} \mathbf{e}^T)^T. \tag{29}$$

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

$$\vartheta^{(k+1)} \leq c_3 (1 - \alpha^{(k)} (1 - c^{(k)})) \vartheta^{(k)}. \tag{30}$$

(iv) Return  $\bar{\omega} = \omega^{(k+1)}$ , if  $err^{(k)} := \|\omega^{(k+1)} - \omega^{(k)}\| / \|\omega^{(k+1)}\| \leq \varepsilon$ , 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 [34]. The computational efficiency depends on the way how the inner linear systems (29) are solved. The Jacobian matrix is non-symmetric and indefinite with the following block structure:

$$\mathbf{J}(\omega^{(k)}) = \begin{pmatrix} \mathbf{F}_\kappa & \mathbf{J}_{12} & \mathbf{0} \\ \mathbf{J}_{12}^T & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{Z}^{(k)} & \mathbf{M}^{(k)} \end{pmatrix}, \quad \mathbf{J}_{12} = \begin{pmatrix} -\mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Eliminating the 2nd and 3rd component of  $\Delta \omega^{(k+1)}$ , we get the reduced linear system for  $\Delta \boldsymbol{\mu}^{(k+1)}$  with the Schur complement

$$\mathbf{S}_\kappa^{(k)} = \mathbf{F}_\kappa + \mathbf{J}_{12} \mathbf{M}^{(k)} (\mathbf{Z}^{(k)})^{-1} \mathbf{J}_{12}^T. \tag{31}$$

As  $\boldsymbol{\mu}^{(k)} > \mathbf{0}$ ,  $\mathbf{z}^{(k)} > \mathbf{0}$ , the matrix  $\mathbf{S}_\kappa^{(k)}$  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:

$$\mathbf{P}_\kappa^{(k)} = \mathbf{D}_{\mathbf{F}_\kappa} + \mathbf{J}_{12} \mathbf{M}^{(k)} (\mathbf{Z}^{(k)})^{-1} \mathbf{J}_{12}^T,$$

where  $\mathbf{D}_{F_k} = \text{diag}(\mathbf{F}_k)$ . Although  $\mathbf{S}_k^{(k)}$  and  $\mathbf{P}_k^{(k)}$  depend on the  $k$  iteration, the eigenvalue bounds of the preconditioned matrix  $(\mathbf{P}_k^{(k)})^{-1}\mathbf{S}_k^{(k)}$  are positive and do not depend on  $k$ . Consequently, the spectral condition number  $\text{cond}$  is bounded by (see [34]):

$$\text{cond}((\mathbf{P}_k^{(k)})^{-1}\mathbf{S}_k^{(k)}) \leq \text{cond}(\mathbf{D}_{F_k})\text{cond}(\mathbf{F}_k). \tag{32}$$

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

$$\tilde{\mathbf{D}}_{F_k} = \text{diag}(\mathbf{C}\text{diag}(\mathbf{A}_k)^{-1}\mathbf{C}^T) + \text{diag}(\mathbf{0}, \mathbf{0}, \text{diag}(\mathbf{E})).$$

The inequality (32) remains valid also for this choice of  $\tilde{\mathbf{D}}_{F_k}$ .

The conjugate gradient method (CGM) used in the  $k$ -th step of ALGORITHM PF is initialized and terminated adaptively. The initial CGM iteration is taken as the computed result in the previous iteration. The CGM iterations are terminated if the relative residuum is less than the bound, depending on the precision achieved in the outer loop  $\text{err}^{(k-1)}$  and determined by:

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

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

#### 4.2. Semismooth Newton Method

Let  $P_{[a,b]} : \mathbb{R} \rightarrow [a, b]$  be the projection of  $\mathbb{R}$  on the interval  $[a, b]$ ,  $a \leq b$ , defined by the max-function:

$$P_{[a,b]}(x) = x - \max\{0, x - b\} + \max\{0, -x + a\}, \quad x \in \mathbb{R}. \tag{33}$$

The last two lines in (24) representing the leak condition can be equivalently written as:

$$\lambda_{n,i} = P_{[-g_i, g_i]}(\lambda_{n,i} + \rho_i(\mathbf{N}\mathbf{u})_i), \quad i \in \mathcal{N}, \tag{34}$$

where  $\rho_i > 0$ ,  $i \in \mathcal{N}$ , are arbitrary but fixed parameters. We introduce the new variable  $\mathbf{s}_n \in \mathbb{R}^{n_s}$  whose components  $s_{n,i} = \kappa_i(\mathbf{N}\mathbf{u})_i + \lambda_{n,i}$  approximate  $-\sigma_n(\mathbf{x}_i)$  at the nodes  $\mathbf{x}_i \in \bar{\gamma}_S \setminus \bar{\gamma}_D$ ,  $i \in \mathcal{N}$ . Then (34) takes the form:

$$0 = \kappa_i(\mathbf{N}\mathbf{u})_i - s_{n,i} + P_{[-g_i, g_i]}(s_{n,i} + (\rho_i - \kappa_i)(\mathbf{N}\mathbf{u})_i), \quad i \in \mathcal{N}.$$

We distinguish two cases:

$$\left. \begin{aligned} \text{if } \kappa_i = 0, \rho_i = \rho : \quad & 0 = -s_{n,i} + P_{[-g_i, g_i]}(s_{n,i} + \rho(\mathbf{N}\mathbf{u})_i), \\ \text{if } \kappa_i > 0, \rho_i = \kappa_i : \quad & 0 = \kappa_i(\mathbf{N}\mathbf{u})_i - s_{n,i} + P_{[-g_i, g_i]}(s_{n,i}), \end{aligned} \right\} i \in \mathcal{N}. \tag{35}$$

The equations in (35) can be reformulated using (33). As we prefer the vector notation, we split  $\mathcal{N}$  into two subsets:

$$\mathcal{N}_0 = \{i \in \mathcal{N} : \kappa_i = 0\}, \quad \mathcal{N}_+ = \{i \in \mathcal{N} : \kappa_i > 0\}$$

and introduce the sub-vectors  $\mathbf{s}_{n, \mathcal{N}_0}$ ,  $\mathbf{s}_{n, \mathcal{N}_+}$ ,  $\mathbf{g}_{\mathcal{N}_0}$ ,  $\mathbf{g}_{\mathcal{N}_+}$  and the sub-matrices  $\mathbf{N}_{\mathcal{N}_0}$ ,  $\mathbf{N}_{\mathcal{N}_+}$ ,  $\mathbf{D}_{\kappa_+} = \text{diag}(\kappa_{\mathcal{N}_+})$ . Then (35) is equivalent to:

$$\left. \begin{aligned} \mathbf{0} &= \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \max\{\mathbf{0}, \mathbf{s}_{n, \mathcal{N}_0} + \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \mathbf{g}_{\mathcal{N}_0}\} + \max\{\mathbf{0}, -\mathbf{s}_{n, \mathcal{N}_0} - \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \mathbf{g}_{\mathcal{N}_0}\}, \\ \mathbf{0} &= \mathbf{N}_{\mathcal{N}_+} \mathbf{u} - \max\{\mathbf{0}, \mathbf{D}_{\kappa_+}^{-1}(\mathbf{s}_{n, \mathcal{N}_+} - \mathbf{g}_{\mathcal{N}_+})\} + \max\{\mathbf{0}, \mathbf{D}_{\kappa_+}^{-1}(-\mathbf{s}_{n, \mathcal{N}_+} - \mathbf{g}_{\mathcal{N}_+})\}, \end{aligned} \right\} \tag{36}$$

where the max-function is understood component-wisely. The max-function can be efficiently replaced by indicator matrices. First we define *inactive/active sets* for the second arguments of the max-functions in (36):

$$\begin{aligned} \mathcal{I}_0^+ &:= \{i \in \mathcal{N}_0 : s_{n,i} + \rho(\mathbf{N}\mathbf{u})_i - g_i > 0\}, \\ \mathcal{I}_0^- &:= \{i \in \mathcal{N}_0 : -s_{n,i} - \rho(\mathbf{N}\mathbf{u})_i - g_i > 0\}, \\ \mathcal{A}_0 &:= \mathcal{N}_0 \setminus (\mathcal{I}_0^+ \cup \mathcal{I}_0^-), \\ \mathcal{I}_+^+ &:= \{i \in \mathcal{N}_+ : s_{n,i} - g_i > 0\}, \\ \mathcal{I}_+^- &:= \{i \in \mathcal{N}_+ : -s_{n,i} - g_i > 0\} \\ \mathcal{A}_+ &:= \mathcal{N}_+ \setminus (\mathcal{I}_+^+ \cup \mathcal{I}_+^-). \end{aligned}$$

According to this notation, we introduce the indicator matrices  $\mathbf{D}(\mathcal{I}_0^+)$ ,  $\mathbf{D}(\mathcal{I}_0^-)$ ,  $\mathbf{D}(\mathcal{A}_0)$ ,  $\mathbf{D}(\mathcal{I}_+^+)$ ,  $\mathbf{D}(\mathcal{I}_+^-)$ ,  $\mathbf{D}(\mathcal{A}_+)$ , respectively. Summarizing, (24) can be written as one equation:

$$\mathbf{G}(\mathbf{y}) = \mathbf{0}, \tag{37}$$

where  $\mathbf{G} : \mathbb{R}^{d(n_u+n_s)+n_p} \rightarrow \mathbb{R}^{d(n_u+n_s)+n_p}$  is defined at  $\mathbf{y} = (\mathbf{u}^T, \mathbf{s}_{n,\mathcal{N}_0}^T, \mathbf{s}_{n,\mathcal{N}_+}^T, \boldsymbol{\lambda}_t^T, \mathbf{p}^T)^T$  by

$$\mathbf{G}(\mathbf{y}) := \begin{pmatrix} \mathbf{A}\mathbf{u} + \mathbf{N}_{\mathcal{N}_0}^T \mathbf{s}_{n,\mathcal{N}_0} + \mathbf{N}_{\mathcal{N}_+}^T \mathbf{s}_{n,\mathcal{N}_+} + \mathbf{T}^T \boldsymbol{\lambda}_t + \mathbf{B}^T \mathbf{p} - \mathbf{b} \\ \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \mathbf{D}(\mathcal{I}_0^+) (\mathbf{s}_{n,\mathcal{N}_0} + \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \mathbf{g}_{\mathcal{N}_0}) + \mathbf{D}(\mathcal{I}_0^-) (-\mathbf{s}_{n,\mathcal{N}_0} - \rho \mathbf{N}_{\mathcal{N}_0} \mathbf{u} - \mathbf{g}_{\mathcal{N}_0}) \\ \mathbf{N}_{\mathcal{N}_+} \mathbf{u} - \mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^+) (\mathbf{s}_{n,\mathcal{N}_+} - \mathbf{g}_{\mathcal{N}_+}) + \mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^-) (-\mathbf{s}_{n,\mathcal{N}_+} - \mathbf{g}_{\mathcal{N}_+}) \\ \mathbf{T}\mathbf{u} \\ \mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c} \end{pmatrix}$$

using also (21). The standard differentiation rules lead to the following (generalized) Jacobian matrix of  $\mathbf{G}$  at  $\mathbf{y}$ :

$$\mathbf{J}_G(\mathbf{y}) = \begin{pmatrix} \mathbf{A} & | & \mathbf{N}_{\mathcal{N}_0}^T & \mathbf{N}_{\mathcal{N}_+}^T & \mathbf{T}^T & \mathbf{B}^T \\ \hline \rho \mathbf{D}(\mathcal{A}_0) \mathbf{N}_{\mathcal{N}_0} & | & -\mathbf{D}(\mathcal{I}_0^+ \cup \mathcal{I}_0^-) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N}_{\mathcal{N}_+} & | & \mathbf{0} & -\mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^+ \cup \mathcal{I}_+^-) & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & | & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & | & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{E} \end{pmatrix}.$$

The Equation (37) will be solved by the Newton-type iterations:

$$\mathbf{J}_G(\mathbf{y}^{(k)}) \mathbf{y}^{(k+1)} = \mathbf{J}_G(\mathbf{y}^{(k)}) \mathbf{y}^{(k)} - \mathbf{G}(\mathbf{y}^{(k)}), \quad k = 0, 1, \dots \tag{38}$$

This iterative process generates a sequence  $\{\mathbf{y}^{(k)}\}$  starting from  $\mathbf{y}^{(0)}$  as an initial approximation. The right hand-side in (38) reads as follows:

$$\mathbf{J}_G(\mathbf{y}^{(k)}) \mathbf{y}^{(k)} - \mathbf{G}(\mathbf{y}^{(k)}) = \begin{pmatrix} \mathbf{b} \\ \hline (\mathbf{D}(\mathcal{I}_0^-) - \mathbf{D}(\mathcal{I}_0^+)) \mathbf{g}_{\mathcal{N}_0} \\ \mathbf{D}_{\kappa_+}^{-1} (\mathbf{D}(\mathcal{I}_+^-) - \mathbf{D}(\mathcal{I}_+^+)) \mathbf{g}_{\mathcal{N}_+} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}.$$

From this it is easy to show that  $\mathbf{s}_{n, \mathcal{I}_0^+}^{(k+1)} = \mathbf{g}_{\mathcal{I}_0^+}$ ,  $\mathbf{s}_{n, \mathcal{I}_0^-}^{(k+1)} = -\mathbf{g}_{\mathcal{I}_0^-}$  and that the remaining components of the new iteration  $\mathbf{y}^{(k+1)}$  in (38) solve the following linear system:

$$\left( \begin{array}{c|ccc} \mathbf{A} & \mathbf{N}_{\mathcal{A}_0}^T & \mathbf{N}_{\mathcal{N}_+}^T & \mathbf{T}^T & \mathbf{B}^T \\ \hline \mathbf{N}_{\mathcal{A}_0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N}_{\mathcal{N}_+} & \mathbf{0} & -\mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^+ \cup \mathcal{I}_+^-) & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{E} \end{array} \right) \begin{pmatrix} \mathbf{u}^{(k+1)} \\ \mathbf{s}_{n, \mathcal{A}_0}^{(k+1)} \\ \mathbf{s}_{n, \mathcal{N}_+}^{(k+1)} \\ \boldsymbol{\lambda}_t^{(k+1)} \\ \mathbf{p}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{b}} \\ \mathbf{0} \\ \tilde{\mathbf{g}}_{\mathcal{N}_+} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}, \quad (39)$$

where  $\tilde{\mathbf{b}} = \mathbf{b} - \mathbf{N}_{\mathcal{I}_0^+}^T \mathbf{g}_{\mathcal{I}_0^+} + \mathbf{N}_{\mathcal{I}_0^-}^T \mathbf{g}_{\mathcal{I}_0^-}$  and  $\tilde{\mathbf{g}}_{\mathcal{N}_+} = \mathbf{D}_{\kappa_+}^{-1} (\mathbf{D}(\mathcal{I}_+^-) - \mathbf{D}(\mathcal{I}_+^+)) \mathbf{g}_{\mathcal{N}_+}$ .

To solve the linear systems (39), we use the Schur complement  $\mathbf{S}^{(k)}$  to  $\mathbf{A}$ -block defined by:

$$\mathbf{S}^{(k)} = \mathbf{F}^{(k)} + \mathbf{D}^{(k)}, \quad (40)$$

where  $\mathbf{F}^{(k)} = \mathbf{C}^{(k)} \mathbf{A}^{-1} (\mathbf{C}^{(k)})^T$  and  $\mathbf{D}^{(k)} = \text{diag}(\mathbf{0}, \mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^+ \cup \mathcal{I}_+^-), \mathbf{0}, \mathbf{E})$  with  $\mathbf{C}^{(k)} = (\mathbf{N}_{\mathcal{A}_0}^T, \mathbf{N}_{\mathcal{N}_+}^T, \mathbf{T}^T, \mathbf{B}^T)^T$ . The right hand-sides of the Schur complement linear systems are given by

$$\mathbf{d}^{(k)} = \mathbf{C}^{(k)} \mathbf{A}^{-1} \tilde{\mathbf{b}} - (\mathbf{0}^T, \tilde{\mathbf{g}}_{\mathcal{N}_+}^T, \mathbf{0}^T, \mathbf{c}^T)^T.$$

Note that the dependence on the  $k$  iteration is through the active set  $\mathcal{A}_0$ . We arrive at the implementation of (38), in which the iterations are performed only with the last four components of  $\mathbf{y}$  assembled in the vector  $\hat{\boldsymbol{\lambda}} = (\mathbf{s}_{n, \mathcal{N}_0}^T, \mathbf{s}_{n, \mathcal{N}_+}^T, \boldsymbol{\lambda}_t^T, \mathbf{p}^T)^T$ .

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

- (i) Determine the inactive/active sets  $\mathcal{I}_0^+, \mathcal{I}_0^-, \mathcal{A}_0, \mathcal{I}_+^+, \mathcal{I}_+^-$  at  $\hat{\boldsymbol{\lambda}}^{(k)}$  to build  $\mathbf{S}^{(k)}$  and  $\mathbf{d}^{(k)}$ .
- (ii) Set  $\mathbf{s}_{n, \mathcal{I}_0^+}^{(k+1)} = \mathbf{g}_{\mathcal{I}_0^+}$ ,  $\mathbf{s}_{n, \mathcal{I}_0^-}^{(k+1)} = -\mathbf{g}_{\mathcal{I}_0^-}$  and solve the Schur complement linear system

$$\mathbf{S}^{(k)} \tilde{\boldsymbol{\lambda}}^{(k+1)} = \mathbf{d}^{(k)} \quad (41)$$

for the remaining components of  $\hat{\boldsymbol{\lambda}}^{(k+1)}$  assembled in the vector in  $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ .

- (iii) Return  $\hat{\boldsymbol{\lambda}}^{(k+1)}$  and  $\mathbf{u}^{(k+1)} = \mathbf{A}^{-1} (\tilde{\mathbf{b}} - \mathbf{C}^T \hat{\boldsymbol{\lambda}}^{(k+1)})$  with  $\mathbf{C} = (\mathbf{N}_{\mathcal{N}_0}^T, \mathbf{N}_{\mathcal{N}_+}^T, \mathbf{T}^T, \mathbf{B}^T)^T$ , 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).

The algorithm is the dual variant of the semismooth Newton method with the super-linear convergence rate established in [39] provided that the initial iteration is sufficiently close to the solution. The overall computational efficiency depends on the way how the (inner) linear systems are solved. Although the size of  $\mathbf{S}^{(k)}$  depends on the cardinality of  $\mathcal{A}_0$  (i.e., on the  $k$  iteration), it is easily seen that these matrices are always positive definite. Hence, the linear systems (41) can be solved by the conjugate gradient method. This method in the  $k$ -th step of ALGORITHM SSN is initialized and terminated adaptively using the same ideas as in ALGORITHM PF. Note that the preconditioning of the conjugate gradient iterations is needed. This follows from the presence of  $\mathbf{D}_{\kappa_+}^{-1}$  in  $\mathbf{D}^{(k)}$  and the fact that  $\kappa_i, i \in \mathcal{N}_+$ , depend on the mesh norms (on the area of the respective triangles in (18) for  $d = 3$  and the length of line segments in (19) for  $d = 2$ ). Hence,  $\text{cond}(\mathbf{S}^{(k)})$  tends to infinity, when the finite element mesh norm approaches zero. As  $\mathbf{S}^{(k)}$  is ill-conditioned through its diagonal, we use the diagonal preconditioner:

$$\mathbf{P}^{(k)} = \mathbf{D}_{\mathbf{F}^{(k)}} + \mathbf{D}_{\mathbf{D}^{(k)}},$$

where  $\mathbf{D}_{\mathbf{F}^{(k)}} = \text{diag}(\mathbf{F}^{(k)})$  and  $\mathbf{D}_{\mathbf{D}^{(k)}} = \text{diag}(\mathbf{0}, \mathbf{D}_{\kappa_+}^{-1} \mathbf{D}(\mathcal{I}_+^+ \cup \mathcal{I}_+^-), \mathbf{0}, \text{diag}(\mathbf{E}))$ . From spectral analysis presented in [34], it follows that although  $\mathbf{S}^{(k)}$  and  $\mathbf{P}^{(k)}$  depend on  $\kappa_i, i \in \mathcal{N}_+$ , the

eigenvalue bounds of the preconditioned matrix  $(\mathbf{P}^{(k)})^{-1}\mathbf{S}^{(k)}$  are positive and independent of  $\kappa_i, i \in \mathcal{N}_+$ . Consequently,

$$\text{cond}((\mathbf{P}^{(k)})^{-1}\mathbf{S}^{(k)}) \leq \text{cond}(\mathbf{D}_{F^{(k)}})\text{cond}(\mathbf{F}^{(k)}). \tag{42}$$

In computations, we approximate  $\mathbf{D}_{F^{(k)}}$  using analogous idea as in ALGORITHM PF.

### 5. Numerical Experiments

The computations were performed by the supercomputer Salomon at IT4I VŠB-TUO [40]. 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 by solving auxiliary linear systems involving  $\mathbf{A}_\kappa$  or  $\mathbf{A}$  with the preliminary Cholesky factorization realized by 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 [34]. ALGORITHM SSN uses  $r_{\text{tol}} = 0.01, c_{\text{fact}} = 0.5$ , and  $\varepsilon = 10^{-3}$ . The termination tolerance  $\varepsilon$  leads to the relative residua of order  $10^{-5}$ . In the tables below the numbers  $n_{it}, n_F$  of the outer iterations, and the matrix-vector multiplications by  $\mathbf{F}_\kappa$  or  $\mathbf{F}^{(k)}$ , respectively, are monitored. Note that  $n_F$  determines overall complexities of computations. The partitions  $\mathcal{T}_h$  of  $\Omega$  are generated by Iso2mesh toolbox and ANSYS software [41,42]. 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 [33,43]. To guarantee the uniqueness of the solution to (4) we will consider that  $\partial\Omega$  is decomposed into three non-overlapping and non-empty parts  $\Gamma, S$ , and  $\Gamma_N$  open in  $\partial\Omega$  (see Remark 2). Nevertheless, one example and comments when  $\Gamma_N = \emptyset$  will be presented, as well. Finally note that all physical quantities are considered in the SI system so that the units of  $g$  and  $\kappa$  are Pa and Pa.s.m<sup>-1</sup>, respectively.

**Example 1.** (Square in 2D). Let  $\Omega = (0, 1)^2$ . To construct the triangulation  $\mathcal{T}_h$  we first cut  $\Omega$  into small squares and then each square into two triangles, see Figure 1(left). The decomposition of  $\partial\Omega$  is defined as follows:  $\Gamma = (0, 1) \times \{1\}, \Gamma_N = \gamma_{\text{left}} \cup \gamma_{\text{right}}, S = (0, 1) \times \{0\}$ , where  $\gamma_{\text{left}} = \{0\} \times (0, 1), \gamma_{\text{right}} = \{1\} \times (0, 1)$ . Further  $f = -2\mu \text{div } \mathbb{D}(u_{\text{exp}}) + \nabla p_{\text{exp}}$  in  $\Omega, \mu = 1/2, u = 0$  on  $\Gamma, \sigma_N = 2\mu \mathbb{D}(u_{\text{exp}})n - p_{\text{exp}}n$  on  $\Gamma_N, g = 15$ , and  $\kappa = 30$  on  $S$  (see Remark 5), where  $u_{\text{exp}} = (u_{\text{exp},1}, u_{\text{exp},2})$ , and

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

Note that  $u_{\text{exp}}$  satisfies the homogeneous Dirichlet boundary condition on  $\partial\Omega$ . Data are chosen in such a way that both, leak and no leak zones appear on  $S$ . Figure 1 (right) presents the streamlines of the computed solution. One can see that the leak on  $S$  generates the fluid suction on  $\Gamma_N$ . Figure 2 shows the pressure and velocity distribution in  $\Omega$ . The magnitudes of the normal velocity  $|u_n|$  and of the normal stress  $|\sigma_n|$  on  $S$  are seen in Figure 3 (left), while  $|\tilde{\sigma}_n|$  is depicted in Figure 3 (right). In the tables below the values  $n_{it}, n_F$ , and CPU time (in seconds) for different finite element meshes with increasing  $n_u, n_p$ , and  $n_s$ , and different values of  $g$  are compared. The computational complexities for  $g = 15$  are shown in Table 1. The analogous characteristics for  $g = 0.1$  (leak everywhere) and  $g = 100$  (no leak) are summarized in Tables 2 and 3, respectively. In view of the values of  $n_F$ , one can see that both algorithms are scalable. The computations without preconditioning are considerably less efficient as it is seen from Table 4.

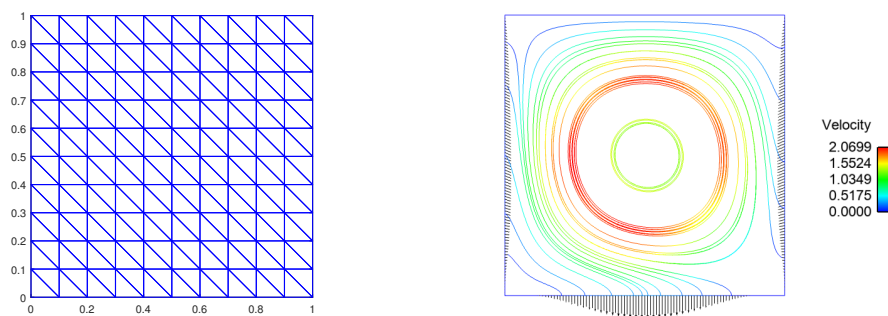


Figure 1. Two-dimensional (2D) mesh (left) and streamlines (right);  $g = 15, \kappa = 30$ .

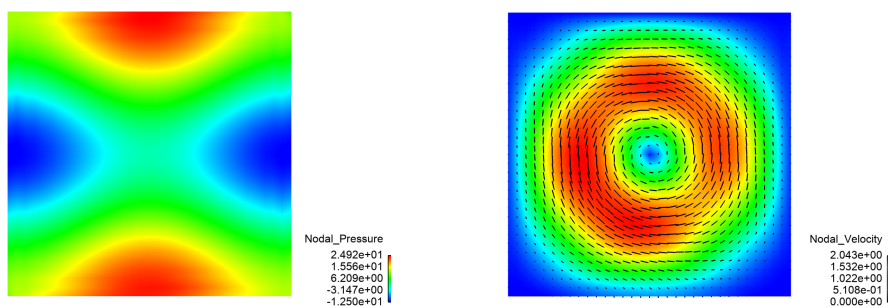


Figure 2. The pressure (left) and velocity (right) distribution;  $g = 15, \kappa = 30$ .

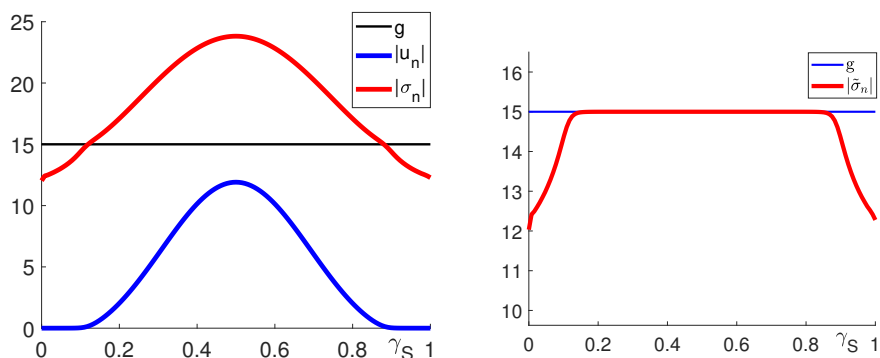


Figure 3. Distribution of  $|u_n|, |\sigma_n|$  (left) and  $|\bar{\sigma}_n|$  (right) on  $S$ ;  $g = 15, \kappa = 30$ .

Table 1. Computations with preconditioner for  $g = 15, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
8320/4225/65	1.2	12/171	0.7	6/97
18,624/9409/97	3.7	12/194	2.8	8/139
33,024/16,641/129	8.4	12/209	4.8	7/129
51,520/25,921/161	15.5	12/231	10.2	8/150
74,112/37,249/193	23.8	12/228	15.7	7/139
100,800/50,625/225	42.4	12/260	25.5	7/147
131,584/66,049/257	62.4	12/266	38.5	7/147
166,464/83,521/289	82.9	12/251	58.3	7/156
205,440/103,041/321	118.0	12/261	78.3	7/153
248,512/124,609/353	172.0	13/295	110.0	7/164



**Table 2.** Computations with preconditioner for  $g = 0.1, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
8320/4225/65	1.1	10/131	0.8	6/88
18,624/9409/97	3.0	11/144	2.3	6/97
33,024/16,641/129	6.4	12/178	3.7	6/100
51,520/25,921/161	12.0	12/181	8.3	7/120
74,112/37,249/193	16.6	10/164	16.1	7/131
100,800/50,625/225	32.0	12/190	18.5	5/89
131,584/66,049/257	48.7	12/196	35.3	7/137
166,464/83,521/289	76.1	13/221	50.4	7/134
205,440/103,041/321	110.5	14/237	72.5	7/142
248,512/124,609/353	128.0	12/211	99.8	7/147

**Table 3.** Computations with preconditioner for  $g = 100, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
8320/4225/65	1.2	9/150	0.8	6/87
18,624/9409/97	3.9	9/190	2.1	7/116
33,024/16,641/129	12.6	11/294	6.6	8/142
51,520/25,921/161	15.4	11/275	8.1	6/102
74,112/37,249/193	28.2	10/244	20.2	9/172
100,800/50,625/225	42.7	10/251	24.5	7/134
131,584/66,049/257	61.4	10/258	26.2	5/88
166,464/83,521/289	86.8	10/263	54.7	7/143
205,440/103,041/321	121.6	10/271	85.3	8/168
248,512/124,609/353	164.8	10/281	95.1	7/139

**Table 4.** Computations without preconditioner for  $g = 15, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
8320/4225/65	13.0	27/1554	9.1	13/1450
18,624/9409/97	36.3	31/1865	30.8	20/2036
33,024/16,641/129	103.0	33/2484	82.0	20/2352
51,520/25,921/161	239.6	39/3371	188.5	15/3164
74,112/37,249/193	459.8	42/4089	343.6	14/3672

**Example 2.** (Cube in 3D). Let  $\Omega = (0, 1)^3$ . To construct the partition  $\mathcal{T}_h$  we first cut  $\Omega$  into small cubes and then each cube is split into five tetrahedras, see Figure 4. The partition of  $\partial\Omega$  into  $\Gamma, S,$  and  $\Gamma_N$  is defined as follows:  $\Gamma = \gamma_{top} \cup \gamma_{bottom} \cup \gamma_{back}, \Gamma_N = \gamma_{left} \cup \gamma_{right}, S = \gamma_{front}$ , where  $\gamma_{bottom} = (0, 1) \times (0, 1) \times \{0\}, \gamma_{top} = (0, 1) \times (0, 1) \times \{1\}, \gamma_{back} = \{1\} \times (0, 1) \times (0, 1), \gamma_{left} = (0, 1) \times \{0\} \times (0, 1), \gamma_{right} = (0, 1) \times \{1\} \times (0, 1), \gamma_{front} = \{0\} \times (0, 1) \times (0, 1)$ . Further  $f = -2\mu \operatorname{div} \mathbb{D}(u_{exp}) + \nabla p_{exp}$  in  $\Omega, \mu = 1/2, u = 0$  on  $\Gamma, \sigma_N = 2\mu \mathbb{D}(u_{exp})n - p_{exp}n$  on  $\Gamma_N, g = 15,$  and  $\kappa = 30,$  where  $u_{exp} = (u_{exp,1}, u_{exp,2}, u_{exp,3}),$  and

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

Again,  $u_{exp}$  satisfies the homogeneous Dirichlet boundary condition on  $\partial\Omega$  and data are chosen in such a way that both, leak and no leak zones appear on  $S$ . Figure 5 shows the distribution of the velocity field on  $S$  and  $\Gamma_N$ . One can see the leak zone in the middle of  $S$  and the fluid suction on  $\gamma_{left}$ . The same suction effect appears on  $\gamma_{right}$ . The normal stress  $|\sigma_n|$  and  $|\tilde{\sigma}_n|$  on  $S$  are seen in Figure 6. In Tables 5–8 we summarize analogous characteristics as in Example 1. The conclusions are also analogous, i.e., both algorithms turn out to be scalable also in 3D and the preconditioner plays an important role. Notice that the numbers  $n_F$  of the matrix-vector multiplications are lower compared to 2D case.

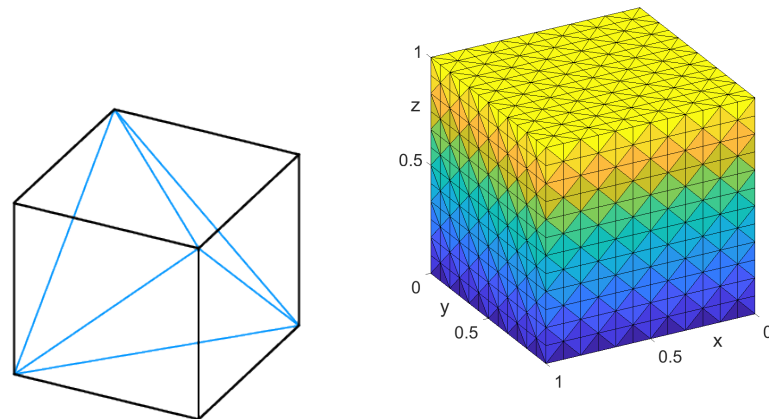


Figure 4. Splitting of the small cube (left) and 3D mesh (right); 1728 cubes and 8640 tetrahedras.

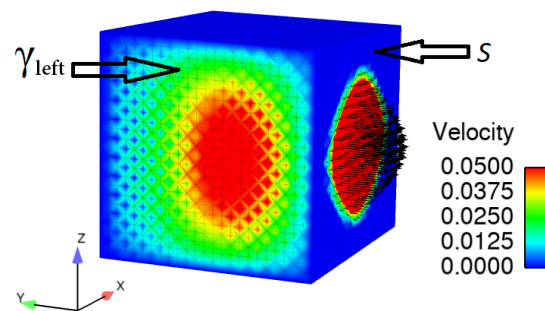


Figure 5. Velocity field on  $S$  and  $\Gamma_N$ ;  $g = 15, \kappa = 30$ .

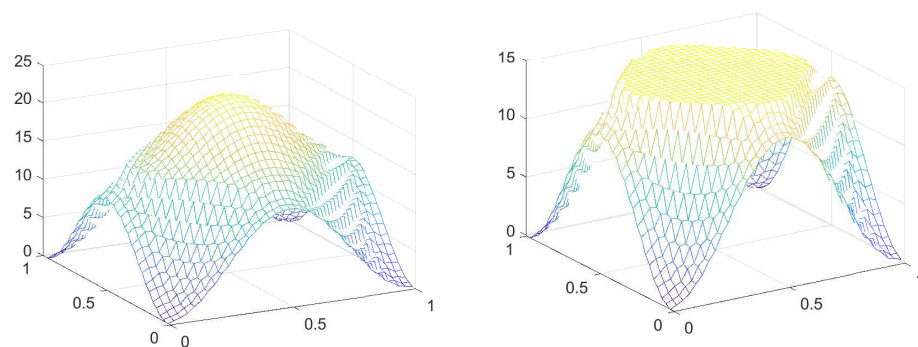


Figure 6. Distribution of  $|\sigma_n|$  (left) and  $|\tilde{\sigma}_n|$  (right) on  $S$ ;  $g = 15, \kappa = 30$ .

**Table 5.** Computations with preconditioner for  $g = 15, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
5148/2197/143	0.9	11/100	0.4	6/67
12,240/4913/255	3.2	11/107	1.3	6/66
23,940/9261/399	12.5	11/120	6.3	6/77
41,400/15,625/575	31.3	11/128	16.2	6/79
65,772/24,389/783	73.7	11/140	34.7	6/77
98,208/35,937/1023	136.8	11/137	68.5	6/77
139,860/50,653/1295	280.8	12/155	124.0	6/77
191,880/68,921/1599	502.9	12/162	235.6	6/83

**Table 6.** Computations with preconditioner for  $g = 0.1, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
5148/2197/143	0.8	11/87	0.4	6/66
12,240/4913/255	3.9	13/95	1.7	7/76
23,940/9261/399	9.8	11/90	5.7	6/72
41,400/15,625/575	26.5	13/101	12.9	6/64
65,772/24,389/783	55.7	11/101	23.9	5/54
98,208/35,937/1023	137.0	15/127	45.8	5/54
139,860/50,653/1295	246.1	14/126	82.7	5/55
191,880/68,921/1599	400.6	13/119	145.5	5/56

**Table 7.** Computations with preconditioner for  $g = 100, \kappa = 30$ .

$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
5148/2197/143	1.0	8/92	0.4	6/71
12,240/4913/255	3.7	8/99	2	7/77
23,940/9261/399	12.5	8/134	6.2	6/76
41,400/15,625/575	33.6	8/138	16.1	6/77
65,772/24,389/783	57.7	7/116	34.5	6/76
98,208/35,937/1023	139.0	8/141	68.7	6/77
139,860/50,653/1295	249.7	8/145	140	7/92
191,880/68,921/1599	400.7	8/133	166.9	4/64

**Table 8.** Computations without preconditioner for  $g = 15, \kappa = 30$ .

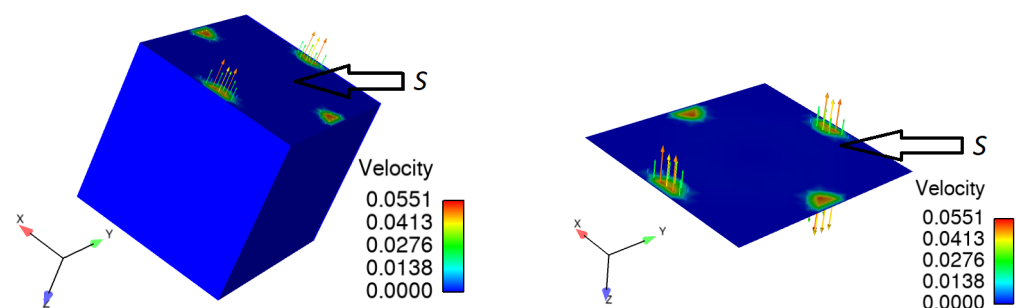
$n_u/n_p/n_s$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
5148/2197/143	21.7	41/2963	14.0	9/1940
12,240/4913/255	86.5	32/2700	40.6	9/2198
23,940/9261/399	688.7	59/7857	273.7	10/3281
41,400/15,625/575	1549.6	46/6909	796.5	10/3673

**Example 3.** (Cube in 3D, non-unique solution). We change the partition of  $\partial\Omega$  from Example 2 as follows:  $\Gamma_N = \emptyset, \Gamma = \gamma_{left} \cup \gamma_{right} \cup \gamma_{back} \cup \gamma_{front} \cup \gamma_{top}$ , and  $S = \gamma_{bottom}$ . Let us remind that, if  $\Gamma_N = \emptyset$ , the pressure might be non-unique depending on the choice of  $g$ . Next we suppose that  $g$  is constant and  $\kappa = 0$  on  $S$ . To establish the critical value of  $g$  we use the Stokes system with the homogeneous Dirichlet data prescribed on the whole boundary  $\partial\Omega$ . Assuming that its solution is smooth enough,  $g_{crit} = (\sup_S(\sigma_n) - \inf_S(\sigma_n))/2$ , where

$\sigma_n$  is the corresponding normal stress. However since the whole procedure is applied to the finite element discretization, the regularity assumption is superfluous. For given physical data and the partition  $\mathcal{T}_h$  characterized by the parameters  $n_u/n_p/n_s = 38,088/15,625/529$  the value  $g_{crit} = 18.31$ . It is easy to show that if  $g < g_{crit}$ , the problem has a unique solution with both, leak and no-leak zones on  $S$ . If  $g > g_{crit}$ , the solution is not unique and only no-leak zone appears on  $S$ . The behavior of the algorithms for different  $g$  is summarized in Table 9. One can see that both algorithms are stable in all cases. The table presents also the values  $\sup_S(\sigma_n^g)$  and  $\inf_S(\sigma_n^g)$  computed for the respective  $g$ . One can easily check that  $(\sup_S(\sigma_n^g) - \inf_S(\sigma_n^g))/2 = g_{crit}$  for  $g \geq g_{crit}$ , while  $(\sup_S(\sigma_n^g) - \inf_S(\sigma_n^g))/2 = g$  if  $g < g_{crit}$ . The distribution of the velocity field on  $S$  for  $g = 16.31$  is seen from Figure 7. Note that the matrix  $\mathbf{C}$  used in (25), especially its part  $(\mathbf{N}^T, \mathbf{B}^T)^T$ , has not the full row-rank due to  $\Gamma_N = \emptyset$ . Consequently,  $\mathbf{F}_\kappa$  and  $\mathbf{F}^{(k)}$  for  $\mathcal{A}_0 = \mathcal{N}_0$  are singular with the defect one and the eigenvector whose all components are equal to 1 corresponds to the zero eigenvalue. The matrix  $\mathbf{S}_\kappa^{(k)}$  of the inner linear systems in ALGORITHM PF is regularized by adding the diagonal matrix  $\mathbf{J}_{12}\mathbf{M}^{(k)}(\mathbf{Z}^{(k)})^{-1}\mathbf{J}_{12}^T$  to  $\mathbf{F}_\kappa$  as seen from (31) (there are non-zero diagonal entries on the level of  $\mathbf{N}$ ). Analogously, the matrix  $\mathbf{S}^{(k)}$  in ALGORITHM SSN is regularized by adding  $\mathbf{D}^{(k)}$  to  $\mathbf{F}^{(k)}$  (see (41)) and by the choice of the active set  $\mathcal{A}_0 \neq \mathcal{N}_0$ . These heuristic arguments show that the matrices of the solved linear systems can be regularized by the construction of the algorithms. Finally note that the condition number bounds of the preconditioned Schur complements (32) and (42) are no longer valid. Nevertheless, the preconditioners can still be used and play the same important role as in the previous examples.

**Table 9.** Computations with different  $g$  and  $\kappa = 0$  for  $\Gamma_N = \emptyset$  ( $n_u/n_p/n_s = 38,088/15,625/529$ ).

$g$	ALGORITHM PF		ALGORITHM SSN		
	$\inf_S(\sigma_n^g)/\sup_S(\sigma_n^g)$	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
$g_{crit} + 2$	-19.86/16.76	106.1	18/ <b>615</b>	32.3	7/ <b>192</b>
$g_{crit} + 0.5$	-18.60/18.02	104.9	20/ <b>581</b>	39.7	8/ <b>232</b>
$g_{crit} + 0.2$	-18.40/18.22	113.9	21/ <b>662</b>	39.1	8/ <b>234</b>
$g_{crit}$	-18.31/18.31	118.7	21/ <b>682</b>	38.5	8/ <b>231</b>
$g_{crit} - 0.2$	-18.11/18.11	105.8	21/ <b>619</b>	40.3	8/ <b>234</b>
$g_{crit} - 0.5$	-17.81/17.81	83.5	18/ <b>462</b>	34.3	8/ <b>201</b>
$g_{crit} - 2$	-16.31/16.31	53.9	15/ <b>189</b>	24.4	7/ <b>156</b>



**Figure 7.** Distribution of the velocity field on  $S$  for  $g = 16.31$ ,  $\kappa = 0$ , and  $\Gamma_N = \emptyset$ ; the whole cube (left) and  $S$  (right).

**Example 4.** (3D branched tube). Let  $\Omega$  be the branched tube as in Figure 8, where the partition  $\mathcal{T}_h$  and the partition of the boundary  $\partial\Omega = \bar{S} \cup \bar{\gamma}_{in} \cup \bar{\gamma}_{out1} \cup \bar{\gamma}_{out2}$  is also depicted. This tube may represent a final part of the capillary in the human body. On  $\Gamma = \gamma_{in}$  we prescribe the input velocity  $u|_{\Gamma,1} = a - a/r^2(y^2 + z^2)$ ,  $u|_{\Gamma,2} = u|_{\Gamma,3} = 0$  with  $a = 5 \cdot 10^{-4}$ ,  $r = 5.94 \cdot 10^{-4}$  while  $\Gamma_N = \gamma_{out1} \cup \gamma_{out2}$  consists of two outputs of the tube with the natural outflow condition  $\sigma_N = 0$ . The leak condition is prescribed on  $S = \partial\Omega \setminus \bar{\Gamma} \cup \bar{\Gamma}_N$ . Further  $f = 0$  in  $\Omega$ ,  $\mu = 1/2$ , and  $\kappa = 30$ . The partition  $\mathcal{T}_h$  is characterized by the following values

of the parameters:  $n_u = 27,408$ ,  $n_p = 9178$ , and  $n_s = 4079$ . The behavior of ALGORITHM PF and ALGORITHM SSN for different  $g$  is reported in Table 10. The distribution of the velocity and the pressure for three selected  $g$  is depicted in Figure 9. Almost all liquid leak through the wall at the vicinity of  $\gamma_{in}$  for the smallest value  $g = 10$  can be observed, see Figure 9 (top). The leak through the wall is minimal for  $g = 50$ , see Figure 9 (bottom). The pressure is also growing for higher values of  $g$ . The zoomed parts of the velocity fields near  $\gamma_{in}$  and  $\gamma_{out2}$  are depicted in Figure 10 (the velocity distribution inside the tube looks like the Poiseuille distribution). We used the reorthogonalization technique of the descent vectors inside the CGM solver [44] that increases considerably efficiency of computations; compare Table 10 with Table 11.

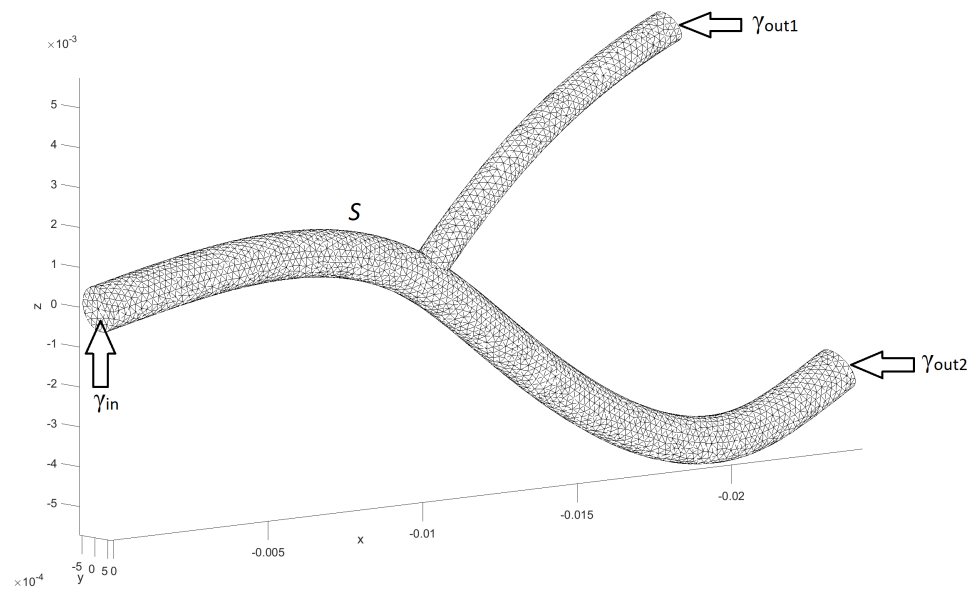


Figure 8. Branched tube: mesh.

Table 10. Computations for different values of  $g$  with reorthogonalization.

$g$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
10	3018	26/3511	1369	15/1882
20	3122	26/3805	1348	15/1885
30	3146	25/3934	1370	15/1882
40	3622	26/4291	1096	13/1522
50	3948	26/4508	995	12/1377

Table 11. Computations for different values of  $g$  without reorthogonalization.

$g$	ALGORITHM PF		ALGORITHM SSN	
	time(s.)	$n_{it}/n_F$	time(s.)	$n_{it}/n_F$
10	8314	25/11,017	15,584	15/21,214
20	9927	25/12,594	14,934	14/21,344
30	10,866	25/13,861	11,156	14/16,653
40	10,815	26/15,520	11,696	12/16,473
50	12,801	26/16,622	18,673	9/24,300

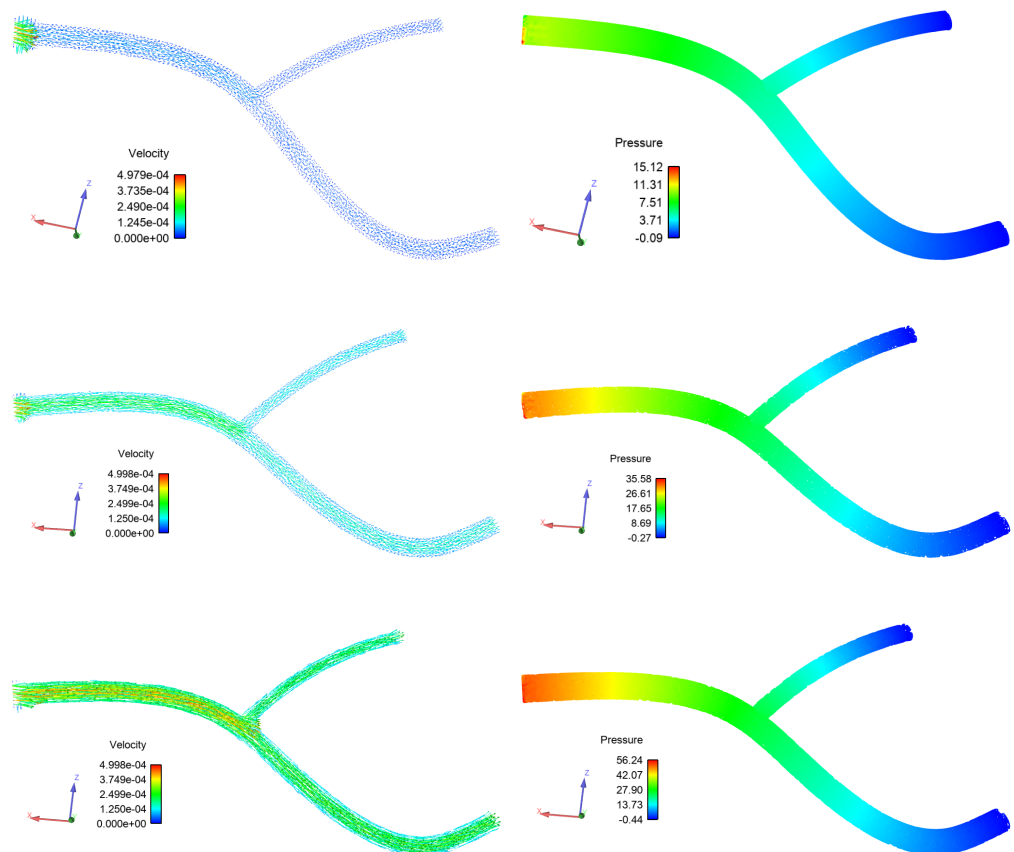


Figure 9. Distribution of the velocity (left) and the pressure (right) for  $g = 10, 30, 50$ .

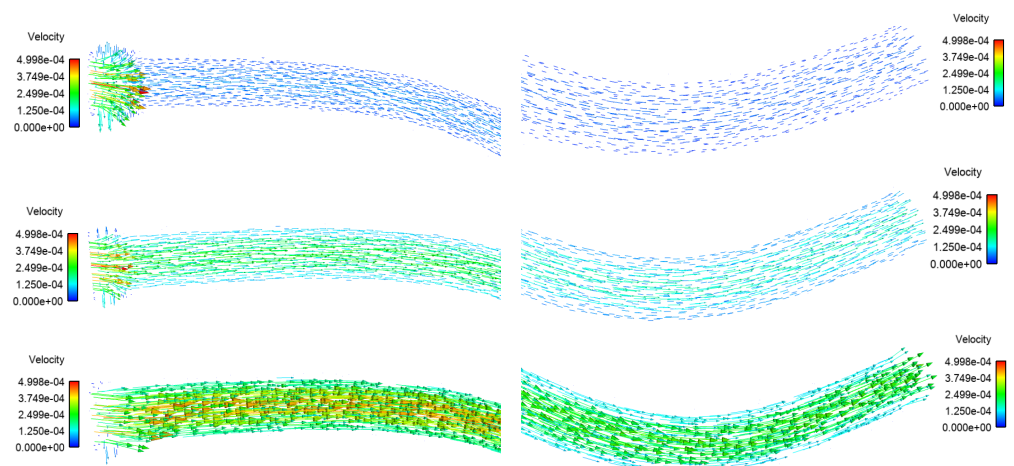


Figure 10. Distribution of the velocity near  $\gamma_{in}$  (left) and  $\gamma_{out2}$  (right) for  $g = 10, 30, 50$ .

### 6. Conclusions and Comments

We have proposed two algorithms for solving the Stokes flow with the threshold leak boundary condition, both based on dual strategies. Unlike the threshold slip boundary conditions [28], the algorithms are identical in 2D and 3D. ALGORITHM PF is based on the path-following variant of the interior point method that generates strictly feasible iterations. This algorithm combines the dumped Newton method in the outer loop with the preconditioned conjugate gradient method used for solving the inner linear systems. The precision control of the inner loop is driven adaptively with respect to the accuracy achieved in the outer level. The crucial ingredient of the algorithm is its diagonal preconditioner that improves ill-conditioning of the system matrices in the later iterations of the outer loop.

ALGORITHM SSN uses the active-set implementation of the semi-smooth Newton method adapted for our type of boundary conditions. The inner linear systems are solved using the same ideas as in ALGORITHM PF. In this case, the diagonal preconditioner improves ill-conditioning caused by small finite element mesh norms. Experimental scalability of both algorithms is indicated by numerical experiments. Indeed, one can conclude that ALGORITHM SSN is more efficient than ALGORITHM PF. Problems with multiple solutions can be also solved since the singular dual Hessian is implicitly regularized by the construction of the algorithms.

**Author Contributions:** Conceptualization, J.H. and R.K.; methodology, J.H. and R.K.; software, K.M. and V.Š.; validation, J.H., R.K., K.M. and V.Š.; formal analysis, J.H. and R.K.; investigation, J.H., R.K., K.M. and V.Š.; resources, J.H. and R.K.; data curation, J.H.; writing—original draft preparation, J.H. and R.K.; writing—review and editing, R.K., K.M. and V.Š.; visualization, K.M. and V.Š.; supervision, J.H. and R.K.; project administration, K.M.; funding acquisition, R.K. and V.Š. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140). The paper also includes the results of the internal BUT FIT project FIT-S-20-6427 (VS). The first author acknowledges the support of FW01010096 of the Czech Technological Agency.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Navier, C.L. Mémoires sur les lois du mouvement des fluides. *Mem. Acad. R. Paris* **1823**, *6*, 389–416.
2. Ekeland, I.; Temam, R. *Analyse Convex et Problèmes Variationnels*; Dunod, Gauthier-Villars: Paris, France, 1974.
3. Rao, J.; Rajagopal, K. The effect of the slip boundary conditions on the flow of fluids in a channel. *Acta Mech.* **1999**, *135*, 113–126. [[CrossRef](#)]
4. Fujita, H. A coherent analysis of Stokes flows under boundary conditions of friction type. *J. Comput. Appl. Math.* **2002**, *149*, 57–69. [[CrossRef](#)]
5. Fujita, H. A mathematical analysis of motions of viscous incompressible fluid under leak and slip boundary conditions. *RIMS Kokyuroku* **1994**, *888*, 199–216.
6. Roux, C.L. Steady Stokes flows with Threshold Slip Boundary Conditions. *Math. Models Meth. Appl. Sci.* **2005**, *15*, 1141–1168. [[CrossRef](#)]
7. Roux, C.L.; Tani, A. Steady solutions of the Navier-Stokes equations with Threshold Slip Boundary Conditions. *Math. Meth. Appl. Sci.* **2007**, *30*, 595–624. [[CrossRef](#)]
8. Saito, N. On the Stokes Equation with the Leak and Slip Boundary conditions of Friction type: Regularity of Solutions. *Publ. RIMS Kyoto Univ.* **2004**, *40*, 345–383. [[CrossRef](#)]
9. Consiglieri, L. A non-local Friction Problem for a class of Non-Newtonian Flows. *Porth. Math.* **2003**, *60*, 237–251.
10. Boukrouche, M.; Paoli, L. Global existence for a 3D non-stationary Stokes flow with Coulomb's type friction boundary conditions. *Appl. Anal.* **2017**, *1–31*. [[CrossRef](#)]
11. Balilescu, L.; Martín, J.S.; Takahashi, T. On the Navier-Stokes system with the Coulomb friction law boundary condition. *Math. Phys.* **2017**, *68*, 1–25. [[CrossRef](#)]
12. Haslinger, J.; Hlaváček, I.; Nečas, J. Numerical methods for unilateral problems in solid mechanics. *Handb. Numer. Anal. IV North-Holl.* **1996**. [[CrossRef](#)]
13. Saito, N.; Sugitani, Y.; Zhou, G. Unilateral problem for the Stokes equations: The well-posedness and finite element approximation. *Appl. Num. Math.* **2016**, *105*, 124–147. [[CrossRef](#)]
14. Zhou, G.; Saito, N. The Navier-Stokes equations under a unilateral boundary condition of Signorini's type. *Math. Fluid Mech.* **2016**, *18*, 481–510. [[CrossRef](#)]
15. Glowinski, R. *Numerical Methods for Nonlinear Variational Problems*; Springer Series in Computational Physics; Springer: Berlin, Germany, 1983. [[CrossRef](#)]
16. Fortin, M.; Glowinski, R. *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary Value Problems*. In *Studies in Mathematics and Its Applications*; Elsevier: North-Holland, The Netherlands, 1983; Volume 15. [[CrossRef](#)]
17. Girault, V.; Raviart, P.A. *Finite Element Methods for Navier-Stokes Equations, Theory and Algorithms*; Springer: Berlin, Germany, 1986. [[CrossRef](#)]
18. Brezzi, F.; Fortin, M. *Mixed and Hybrid Finite Element Methods*; Springer Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 1991; Volume 15. [[CrossRef](#)]

19. Haslinger, J.; Mäkinen, R.A.E. The parameter identification in the Stokes system with threshold slip boundary conditions. *ZAMM* **2020**, *100*, 5. [[CrossRef](#)]
20. Ayadi, M.; Gdoura, M.K.; Sassi, T. Error estimates for Stokes problem with Tresca friction conditions. *ESAIM Math. Model. Numer. Anal.* **2014**, *48*, 1413–1429. [[CrossRef](#)]
21. Kashiwabara, T. On a finite element approximation of the Stokes equations under a slip boundary condition of the friction type. *Jpn. J. Industr. Appl. Math.* **2013**, *30*, 227–261. [[CrossRef](#)]
22. Kashiwabara, T. Finite element method for the Stokes equations under a leak boundary condition of friction type. *SIAM J. Numer. Anal.* **2013**, *51*, 2448–2469. [[CrossRef](#)]
23. Djoko, J.K.; Mbehou, M. Finite element analysis for Stokes and Navier Stokes equations driven by threshold slip boundary conditions. *Int. J. Numer. Anal. Model.* **2013**, *B4*, 235–255.
24. Haslinger, J.; Kučera, R.; Šátek, V. Stokes system with local Coulomb's boundary conditions: Analysis of discretized models and Implementation. *Comp. Math. Appl.* **2019**, *77*, 1655–1667. [[CrossRef](#)]
25. Djoko, J.K.; Koko, J. Numerical methods for the Stokes and Navier-Stokes equations driven by slip boundary conditions. *Comput. Methods Appl. Mech. Engrg.* **2016**, *305*, 936–958. [[CrossRef](#)]
26. Djoko, J.K.; Koko, J.; Kučera, R. Power law Stokes equations with threshold slip boundary conditions: Numerical methods and implementation. *Math. Methods Appl. Sci.* **2019**, *42*, 1488–1511. [[CrossRef](#)]
27. Kučera, R.; Haslinger, J.; Šátek, V.; Jarošová, M. Efficient methods for solving the Stokes problem with slip boundary conditions. *Math. Comput. Simul.* **2018**, *145*, 114–124. [[CrossRef](#)]
28. Haslinger, J.; Kučera, R.; Sassi, T.; Šátek, V. Dual strategies for solving the Stokes problem with stick-slip boundary conditions in 3D. *Math. Comput. Simul.* **2021**, *189*, 191–206. [[CrossRef](#)]
29. Kučera, R. Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints. *SIAM J. Optim.* **2008**, *19*, 846–862. [[CrossRef](#)]
30. Brzobohatý, T.; Jarošová, M.; Kučera, R.; Šátek, V. Path-following interior point method: Theory and applications for the Stokes flow with a stick-slip boundary condition. *Adv. Eng. Softw.* **2019**, *129*, 35–43. [[CrossRef](#)]
31. Haslinger, J.; Stebel, J. Stokes problem with a solution dependent slip bound: Stability of solutions with respect to domains. *ZAMM—J. Appl. Math. Mech./Z. Angew. Math. Mech.* **2016**, *96*, 1049–1060. [[CrossRef](#)]
32. Arnold, D.N.; Brezzi, F.; Fortin, M. A stable finite element for the Stokes Equations. *Calcolo* **1984**, *21*, 337–344. [[CrossRef](#)]
33. Koko, J. Efficient MATLAB codes for the 2D/3D Stokes equation with the mini-element. *Informatica* **2019**, *30*, 243–268. [[CrossRef](#)]
34. Kučera, R.; Machalová, J.; Netuka, H.; Ženčák, P. An interior point algorithm for the minimization arising from 3D contact problems with friction. *Optim. Methods Softw.* **2013**, *28*, 1195–1217. [[CrossRef](#)]
35. Kučera, R.; Motyčková, K.; Markopoulos, A. The R-linear convergence rate of an algorithm arising from the semi-smooth Newton method applied to 2D contact problems with friction. *Comput. Optim. Appl.* **2015**, *61*, 437–461. [[CrossRef](#)]
36. Kučera, R.; Motyčková, K.; Markopoulos, A.; Haslinger, J. 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.* **2020**, *35*, 65–86. [[CrossRef](#)]
37. Kučera, R.; Motyčková, K.; Pacholek, J.; Sassi, T. The semi-smooth Newton method for solving the Stokes problem with the stick-slip boundary Condition. *AIP Conf. Proc.* **2018**, *1978*, 360003. [[CrossRef](#)]
38. Nocedal, J.; Wächter, A.; Waltz, R.A. Adaptive barrier update strategies for nonlinear interior methods. *SIAM J. Optim.* **2009**, *19*, 4, 1674–1693. [[CrossRef](#)]
39. Hintermüller, M.; Ito, K.; Kunisch, K. The primal-dual active set strategy as a semismooth Newton method. *SIAM J. Optim.* **2002**, *13*, 3, 865–888. [[CrossRef](#)]
40. Salomon Hardware Overview. Available online: <https://docs.it4i.cz/salomon/hardware-overview/> (accessed on 1 October 2021).
41. Iso2mesh: A 3D Surface and Volumetric Mesh Generator for MATLAB/Octave. Available online: <http://iso2mesh.sourceforge.net> (accessed on 13 October 2021).
42. Ansys: Engineering Simulation Software. Available online: <https://www.ansys.com/> (accessed on 16 September 2021).
43. Arzt, V. Finite Element Meshes and Assembling of Stiffness Matrices. Master's Thesis, VŠB-TU Ostrava, Ostrava, Czech Republic, 2019.
44. Kamath, C. *The FETI Level 1 Method: Theory and Implementation*; Technical Report UCRL-ID-138075; Lawrence Livermore National Laboratory: Livermore, CA, USA, 2000; pp. 1–31. [[CrossRef](#)]