An inexact Newton method for solving complementarity problems in hydrodynamic lubrication

F. Mezzadri * [1] and E. Galligani †[1]

[1]Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, via P. Vivarelli 10/1, building 26, I-41125, Modena

April 12, 2022

Received: date / Accepted: date

Abstract

We present an iterative procedure based on a damped inexact Newton iteration for solving Linear Complementarity Problems. We introduce the method in the framework of a popular problem arising in mechanical engineering: the analysis of cavitation in lubricated contacts. In this context, we show how the perturbation and the damping parameter are chosen in our method and we prove the global convergence of the entire procedure. A Fortran implementation of the method is finally analyzed. First, we validate the procedure and analyze all its components, performing also a comparison with a recently proposed technique based on the Fischer-Burmeister-Newton iteration. Then, we solve a 2D problem and provide some insights on an efficient implementation of the method exploiting routines of the Lapack and of the PETSc packages for the solution of inner linear systems.

Keywords: *Complementarity Problem, Damped Inexact Newton algorithm, Cavitation Lubrication*
MSC2010: 65H10, 65K05, 65L12, 90C51, 76B10

1 Introduction

Linear Complementarity Problems (LCPs) began to be extensively studied in the mid 1960's (e.g. see [1]) and they immediately excited much interest for both their mathematical properties and their applications. Indeed, on the mathematical side, the LCP unifies the formulation of linear programming, quadratic programming and bimatrix game problems, leading to important discoveries in all these fields [2, p. 3]. Several applications ranging from engineering, to economics or computer science, nonetheless, exist.

For instance, an interesting problem that often arises in mechanical engineering and that can be formulated as a complementarity problem consists in the analysis of cavitation in lubricated contacts. For a comprehensive review of earlier studies on this problem, the reader may refer to [3]. Other relevant contributions include [4, 5, 6, 7, 8]. Other studies, like [9], have then considered the case of cavitation in elasto-hydrodynamic lubrication, while, more recently, also the effect of fluid compressibility, piezoviscosity and non-Newtonian fluid behavior has been analyzed by [10].

The aim of this paper is to propose an iterative procedure based on a Damped Inexact Newton (DIN) iteration for solving LCPs. Approaches of this kind are characterized by the presence of a perturbation, which prevents the algorithm from stalling, and of a damping parameter, which ensures the global convergence of the procedure. The choice of these two parameters is, thus, of fundamental importance and it is the distinctive feature of different damped inexact Newton methods. Therefore, we show how the perturbation

*francesco.mezzadri@unimore.it
†emanuele.galligani@unimore.it

and the damping parameters are chosen in our procedure and provide a proof of the global convergence of the method. We also analyze how different perturbations affect the efficiency of the method itself.

Although the procedure we propose is general, it seems appropriate to present it with special regard to a specific problem, which, in our case, is the aforementioned cavitation in hydrodynamic lubrication. This idea follows [11], where a procedure based on an inexact Newton iteration with Armijo backtracking condition has been used for solving practical problems concerning oxygen diffusion and combustion. In particular, we refer to the model of cavitation in hydrodynamic lubrication presented in [12], where the authors reformulated the problem so that pressure and a variable related to density are complementary in the entire domain ([13]).

We also compare our method with a similar procedure based on the Fischer-Burmeister-Newton (FBN) iteration. Algorithms of this kind are efficient and have been recently applied to lubrication problems in [14]. However, we show that the DIN-based algorithm better enforces the non-negativity conditions imposed by the complementarity.

Lastly, we also analyze the efficiency of the method. In particular, we make some considerations on how efficiency can be improved and compare different inner solvers, all implemented through the Lapack [15] or the PETSc [16, 17] packages.

The paper is structured as follows. In Section 2 we outline the problem of our concern, which is described by the Reynolds equation. For a more detailed description of this equation in bearing systems, the reader is referred to [18, Chapt. 6]. We then rewrite the problem in a complementarity formulation, present its discretized form and introduce the general layout of our method applied to the resulting nonlinear system.

In Section 3 we analyze the method. We prove that the conditions for it to be a damped inexact Newton method hold and we provide a proof of its convergence. In this regard, we also show how the perturbation and the damping parameter are chosen and which criteria they must satisfy. Finally, we briefly present the FBN method, which we later compare with the DIN iteration.

Sections 4, 5 and 6 are devoted to numerical experiments. First, in Section 4 we provide the numerical data which define the test problems and the setting of the algorithm. We also summarize the configurations of lubricated bearings which are considered afterwards. Then, in Section 5 we validate and analyze the procedure by a Fortran implementation of the DIN method. We consider the behavior of the method in different situations (e.g. changing the starting vectors) and when some parts of the algorithm are removed, so to better show the role of each component. We also compare the results obtained by the DIN method with those computed by the FBN iteration. Finally, Section 6 presents some insights on a faster implementation and provides a comparison of various linear solvers called through Lapack or PETSc. Lastly, we present the solution of a 2D example.

Conclusions are then given in Section 7.

2 Layout of the algorithm applied to a complementarity problem

2.1 Description of the problem

The problem of our interest is described by the Reynolds equation (1886), which governs the pressure distribution of the thin, fluid films typical of bearing systems. When the thickness of the film is constant in time, the equation is stationary and it reads

$$\frac{1}{12\mu}\frac{\partial}{\partial x}\left(\rho h^3\frac{\partial p}{\partial x}\right)+\frac{1}{12\mu}\frac{\partial}{\partial y}\left(\rho h^3\frac{\partial p}{\partial y}\right)=\frac{1}{2}U\frac{\partial(\rho h)}{\partial x}, \tag{1}$$

where

- the unknowns are the pressure $p=p(x,y)$ and the density $\rho=\rho(x,y)$;
- $h=h(x,y)>0$ is the thickness of the film;
- $\mu>0$ is the viscosity of the fluid;
- $U>0$ is the velocity of the fluid.

Cavitation consists in cavities forming in the fluid when pressure gets low. Equation (1) remains valid also in this case, but pressure and density behave differently where cavitation does and does not occur. Indeed, in the zone where cavitation does not occur (called *active region*), p is always greater than or equal to zero. Density ρ is instead equal to the density of the liquid, ρ_0, implying $\rho_0 - \rho = 0$. On the other hand, in the cavitated region (also called *inactive region*), ρ is not equal to ρ_0, since the fluid is now made of liquid, vapor and gas. Hence $\rho \leq \rho_0$, or, equivalently, $\rho_0 - \rho \geq 0$. Simultaneously, we have $p = 0$. Thus we can write the complementarity condition

$$\begin{aligned} (\rho_0 - \rho)p &= 0 \\ (\rho_0 - \rho) \geq 0; \qquad p &\geq 0. \end{aligned} \tag{2}$$

Following [12], we then pose $r = (\rho_0 - \rho)/\rho_0$ and write (1) as

$$\frac{\partial}{\partial x}\left(\frac{h^3}{6\mu}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{h^3}{6\mu}\frac{\partial p}{\partial y}\right) - U\frac{\partial h}{\partial x} + U\frac{\partial (rh)}{\partial x} = 0 \tag{3}$$

subject, in cavitation, to the complementarity condition $pr = 0$, with $p \geq 0$, $r \geq 0$.

We now discretize the domain by a grid of n inner points and approximate the derivatives of Equation (3) by a finite difference scheme [19, Chap. 6] or by a finite element method [20, §14.5]. In this paper we use the box-discretization in [19, pp. 196-199] and discretize the derivative in r by backwards finite difference quotients. It is worth noticing that this discretization method works directly on the self-adjoint equation. Therefore, it is sufficient to require the continuity of $h^3\partial p/\partial x$ and of $h^3\partial p/\partial y$, while the partial derivatives of p can be piece-wise continuous.

After discretization, we obtain the algebraic system

$$\begin{aligned} & A\boldsymbol{p} + B\boldsymbol{r} = \boldsymbol{c} \\ \text{subject to:} \quad & \\ & \boldsymbol{p}^T\boldsymbol{r} = 0 \\ & \boldsymbol{p} \geq \boldsymbol{0}; \qquad \boldsymbol{r} \geq \boldsymbol{0}, \end{aligned} \tag{4}$$

where $\boldsymbol{0} \in \mathbb{R}^n$ denotes the null vector, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$ and $\boldsymbol{c}, \boldsymbol{p}, \boldsymbol{r} \in \mathbb{R}^n$. This is the formulation given in [2, p. 30] for the generalized complementarity problem.

More specifically, considering, for simplicity, a 1D problem and defining

$$\alpha_i = \frac{h(x_i)^3}{6\mu}, \qquad \beta_i = Uh(x_i) \tag{5}$$

the matrix A is the tridiagonal matrix of elements $a_{i,j}$

$$\begin{aligned} a_{i,i-1} &= -\frac{1}{\Delta x}\alpha_{i-1/2}, && i = 2, 3, \ldots, n; \\ a_{i,i} &= \frac{1}{\Delta x}(\alpha_{i-1/2} + \alpha_{i+1/2}), && i = 1, 2, \ldots, n; \\ a_{i,i+1} &= -\frac{1}{\Delta x}\alpha_{i+1/2}, && i = 1, 2, \ldots, n-1. \end{aligned} \tag{6}$$

It is easy to verify that A is irreducible [19, p.18] and diagonally dominant (by rows and by columns, strictly at the first and at the last row/column). Then, A is irreducibly diagonally dominant [19, p.23]. Having also positive diagonal elements and non-positive off-diagonal elements, it is an M-matrix [19, p.91]. Finally, due to its symmetry, it is a Stieltjes matrix, and, thus, positive definite [19, p.91].

On the other hand, B is the bidiagonal matrix of elements $b_{i,j}$

$$\begin{aligned} b_{i,i-1} &= \beta_{i-1}, && i = 2, 3, \ldots, n; \\ b_{i,i} &= -\beta_i && i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

Then, B is diagonally dominant by columns (strictly at the last column) and it has negative diagonal elements and non-negative off-diagonal elements. Moreover, it is nonsingular.

In the 2D case, similar considerations apply, with A block-tridiagonal matrix (with diagonal blocks that are tridiagonal and sub- and super-diagonal blocks that are diagonal) and B block-diagonal (with lower-bidiagonal blocks).

2.2 The damped inexact Newton iteration

Problem (4) is the LCP that we use to present our damped inexact Newton iteration. In this regard, first we write problem (4) as a system of $2n$ nonlinear algebraic equations with restriction on the sign of the components of $\boldsymbol{p}$ and of $\boldsymbol{r}$:

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{p},\boldsymbol{r}) &= \boldsymbol{0} \\ \boldsymbol{p} \geq \boldsymbol{0}; \qquad \boldsymbol{r} &\geq \boldsymbol{0} \end{aligned} \tag{8}$$

with

$$\boldsymbol{F}(\boldsymbol{p},\boldsymbol{r}) = \begin{pmatrix} \boldsymbol{F}_1(\boldsymbol{p},\boldsymbol{r}) \\ PR\boldsymbol{e} \end{pmatrix}, \tag{9}$$

where $\boldsymbol{F}_1(\boldsymbol{p},\boldsymbol{r}) \in \mathbb{R}^n$ is $A\boldsymbol{p} + B\boldsymbol{r} - \boldsymbol{c}$, $\boldsymbol{e} \in \mathbb{R}^n$ is the unity vector and $P, R \in \mathbb{R}^{n\times n}$ are diagonal matrices whose non-zero entries are the elements of $\boldsymbol{p}$ and of $\boldsymbol{r}$ respectively.

Then, we write the Jacobian matrix

$$F'(\boldsymbol{p},\boldsymbol{r}) = \begin{pmatrix} A & B \\ R & P \end{pmatrix} \tag{10}$$

and set the Newton iteration to solve (4): chosen an initial iterate $(\boldsymbol{p}^{(0)}, \boldsymbol{r}^{(0)})$ sufficiently close to the solution, we define

$$\begin{pmatrix} \boldsymbol{p}^{(k+1)} \\ \boldsymbol{r}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}^{(k)} \\ \boldsymbol{r}^{(k)} \end{pmatrix} + \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix}, \qquad k = 0, 1, \ldots, \tag{11}$$

where $(\Delta\boldsymbol{p}^{(k)}, \Delta\boldsymbol{r}^{(k)})$ is the solution of the linear system

$$F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \begin{pmatrix} \Delta\boldsymbol{p} \\ \Delta\boldsymbol{r} \end{pmatrix} = -\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}). \tag{12}$$

A typical problem with this method is that the iteration on $PR\boldsymbol{e} = 0$ can lead the algorithm to stall. This has been observed, for example, in [21] with regard to the application of the Newton method to the Karush-Kuhn-Tucker conditions of a nonlinear programming problem. We can see this easily by considering a single i-th equation, $n < i \leq 2n$, of system (12)

$$r_i^{(k)} \Delta p_i + p_i^{(k)} \Delta r_i = -p_i^{(k)} r_i^{(k)} \tag{13}$$

and supposing, for example, $r_i^{(k)} = 0$. This implies $\Delta r_i^{(k)} = 0$, which, in turn, implies $r_i^{(k+1)} = 0$, and so on for any following iteration. The same obviously applies to p_i if we suppose $p_i^{(k)} = 0$. Then, if an iterate reaches the boundary of the feasible region, it sticks to it also in the subsequent iterations.

To solve this problem, we perturb the complementarity equations in order to force the iterates sufficiently far from the boundary of the nonnegative orthant $\boldsymbol{p} \geq \boldsymbol{0}$, $\boldsymbol{r} \geq \boldsymbol{0}$. Calling $\tilde{\rho}$ a positive scalar, the perturbed system becomes

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{p},\boldsymbol{r}) &= \tilde{\rho}\tilde{\boldsymbol{e}} \\ \boldsymbol{p} \geq \boldsymbol{0}; \qquad \boldsymbol{r} &\geq \boldsymbol{0} \end{aligned} \tag{14}$$

with

$$\tilde{\boldsymbol{e}} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{e} \end{pmatrix}. \tag{15}$$

Therefore, now we have to solve the perturbed Newton equation

$$F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \begin{pmatrix} \Delta \boldsymbol{p} \\ \Delta \boldsymbol{r} \end{pmatrix} = -\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) + \tilde{\rho}_k \tilde{\boldsymbol{e}} \tag{16}$$

at each iteration, with $\tilde{\rho}_k \to 0$ for $k \to \infty$. In this way, Equation (16) replaces (12) and, together with Equation (11), it generates a sequence $\{\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}\}$ which strictly satisfies the non-negativity conditions in (14). Equation (8) is instead satisfied in the limit.

Finally, we complete the method by introducing a damping parameter α_k, $0 < \alpha_k < 1$, which enforces convergence, as described in Section 3. Denoting the solution of (16) by $(\Delta \boldsymbol{p}^{(k)}, \Delta \boldsymbol{r}^{(k)})$, Equation (11) is therefore replaced by

$$\begin{pmatrix} \boldsymbol{p}^{(k+1)} \\ \boldsymbol{r}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}^{(k)} \\ \boldsymbol{r}^{(k)} \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta \boldsymbol{p}^{(k)} \\ \Delta \boldsymbol{r}^{(k)} \end{pmatrix}. \tag{17}$$

3 Analysis of the method

For an initial guess $(\boldsymbol{p}^{(0)}, \boldsymbol{r}^{(0)})$ and for $k = 0, 1, \ldots$, Equations (16) and (17) make up the *Damped Inexact Newton method* (DIN) if some conditions hold. More specifically,

1. the vector $(\Delta \boldsymbol{p}^{(k)}, \Delta \boldsymbol{r}^{(k)})$ solution of (16) must be a descent direction for the merit function $\Phi(\boldsymbol{p}, \boldsymbol{r}) = \|\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})\|^2$, where $\|\cdot\|$ denotes the Euclidean norm;
2. α_k must guarantee the reduction of the merit function at each iteration, while the components of $\boldsymbol{p}^{(k)}$ and $\boldsymbol{r}^{(k)}$ remain positive for all $k = 0, 1, \ldots$. This means defining a *path following condition* and a *backtracking condition* which α_k must satisfy.

To satisfy these conditions, we act on the choice of the perturbation $\tilde{\rho}_k$ and of the damping parameter α_k.

3.1 Choice of $\tilde{\rho}_k$ and descent condition

With a suitable choice of the perturbation, we can interpret Equation (16) as the k-th iteration of an inexact Newton method [22, 23, 24]. Let us set

$$\tilde{\rho}_k = \sigma_k \mu_k \tag{18}$$

where

- σ_k is the forcing term, $0 < \sigma_{min} \leq \sigma_k \leq \sigma_{max} < 1$;
- μ_k is a perturbation parameter which satisfies

$$\|\mu_k \tilde{\boldsymbol{e}}\| \leq \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|. \tag{19}$$

If we isolate $\tilde{\rho}_k \tilde{\boldsymbol{e}}$ in (16), take the norm of both sides of the equation and replace $\tilde{\rho}_k$ with its definition in (18), applying condition (19) we get

$$\left\| F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \begin{pmatrix} \Delta \boldsymbol{p}^{(k)} \\ \Delta \boldsymbol{r}^{(k)} \end{pmatrix} + \boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \right\| \leq \sigma_k \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|. \tag{20}$$

Thus, $\sigma_k \mu_k \tilde{\boldsymbol{e}}$ has the meaning of a residual. Therefore, we call (20) *residual condition* of the inexact Newton method with forcing term σ_k.

Now we must then choose a μ_k for which the residual condition is satisfied. In this regard, in [25] it is suggested to choose μ_k in the interval

$$\mu_k^{(1)} \equiv \frac{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{n} \leq \mu_k \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|}{\sqrt{n}} \equiv \mu_k^{(2)}. \tag{21}$$

With this choice, the residual condition is satisfied and the following theorem holds.

Theorem 1. *The vector* $(\Delta \boldsymbol{p}^{(k)}, \Delta \boldsymbol{r}^{(k)})$ *is a descent direction for the merit function* $\Phi(\boldsymbol{p}, \boldsymbol{r}) = \|\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})\|^2$, *i.e.*

$$\nabla \Phi(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^T \begin{pmatrix} \Delta \boldsymbol{p}^{(k)} \\ \Delta \boldsymbol{r}^{(k)} \end{pmatrix} \leq 0. \tag{22}$$

Proof. First let us determine how μ_k must be for (22) to be satisfied. In this regard, let us substitute

$$\nabla \Phi(\boldsymbol{p}, \boldsymbol{r}) = 2F'(\boldsymbol{p}, \boldsymbol{r})^T \boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})$$

in (22). By (16) and by $\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})^T \tilde{\boldsymbol{e}} = \boldsymbol{e}^T R P \boldsymbol{e} = \boldsymbol{p}^T \boldsymbol{r}$, with a few simple algebraic passages we can write

$$\nabla \Phi(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^T \begin{pmatrix} \Delta \boldsymbol{p}^{(k)} \\ \Delta \boldsymbol{r}^{(k)} \end{pmatrix} = -2\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2 + 2\sigma_k \mu_k {\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}. \tag{23}$$

It follows that condition (22) is satisfied if

$$\mu_k \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{\sigma_k({\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)})}. \tag{24}$$

We now have to prove that $\mu_k^{(1)}$ and $\mu_k^{(2)}$ satisfy (24) and that inequality (21) holds. To this aim, in the following we use the chain of inequalities (e.g., see [26, pp. 278-279])

$$\frac{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{n} = \frac{\|P^{(k)} R^{(k)} \boldsymbol{e}\|_1}{n} \leq \frac{\|P^{(k)} R^{(k)} \boldsymbol{e}\|}{\sqrt{n}} \leq \frac{\|P^{(k)} R^{(k)} \boldsymbol{e}\|_1}{\sqrt{n}} = \frac{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{\sqrt{n}},$$

(where $\|\cdot\|_1$ indicates the 1-norm) and

$$\|P^{(k)} R^{(k)} \boldsymbol{e}\| \leq \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|.$$

It follows

$$\mu_k^{(1)} = \frac{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{n} \leq \frac{\|P^{(k)} R^{(k)} \boldsymbol{e}\|}{\sqrt{n}} \frac{\sqrt{n} {\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{\sqrt{n} {\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}} \leq \frac{(\sqrt{n}\|P^{(k)} R^{(k)} \boldsymbol{e}\|)^2}{n({\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)})} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}},$$

and

$$\mu_k^{(2)} = \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|}{\sqrt{n}} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|\sqrt{n}} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{\sqrt{n}\|P^{(k)} R^{(k)} \boldsymbol{e}\|} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|^2}{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}},$$

proving that (22) is satisfied for $\mu_k^{(1)}$ and $\mu_k^{(2)}$. Finally, we also have

$$\mu_k^{(1)} = \frac{{\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)}}{n} \leq \frac{\|P^{(k)} R^{(k)} \boldsymbol{e}\|}{\sqrt{n}} \leq \frac{\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|}{\sqrt{n}} = \mu_k^{(2)},$$

which proves that (21) holds as well. □

3.2 Solution of the perturbed Newton equation

We now have to solve the perturbed Newton equation (16). This can be done exactly by a direct solver or approximately by an iterative one. This last option is useful for reducing the computational cost of the procedure when the Jacobian matrix is of large dimensions. Moreover, the tolerance of the iterative solver can be set adaptively, so to require less inner iterations when $\boldsymbol{p}^{(k)}$ and $\boldsymbol{r}^{(k)}$ are still far from the solution.

In order to do so, we need to introduce a new parameter δ_k, which is needed to define adaptively the stopping condition of the iterative solver, but which also affects all the following analysis. Every consideration involving δ_k, however, applies also to case of direct inner solver: in this case, we simply pose $\delta_k = 0$.

The iterative inner solver is stopped when the residual $\hat{\boldsymbol{r}}^{(k)}$ of Equation (16),

$$\hat{\boldsymbol{r}}^{(k)} = F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \begin{pmatrix} \Delta \boldsymbol{p}^{(k)} \\ \Delta \boldsymbol{r}^{(k)} \end{pmatrix} + \boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) - \sigma_k \mu_k \tilde{\boldsymbol{e}}, \tag{25}$$

satisfies

$$\|\hat{\boldsymbol{r}}^{(k)}\| \leq \delta_k(\sqrt{n}\mu_k). \tag{26}$$

Here, $(\Delta \boldsymbol{p}^{(k)}, \Delta \boldsymbol{r}^{(k)})$ denotes the solution of system (16) computed stopping the inner solver when (26) is satisfied. For $\mu_k^{(1)} \leq \mu_k \leq \mu_k^{(2)}$ and $0 < \sigma_{\min} \leq \sigma_k \leq \sigma_{\max} < 1$, it is possible to prove that the vector $(\Delta \boldsymbol{p}^{(k)}, \Delta \boldsymbol{r}^{(k)})$ is a descent direction for the merit function $\Phi(\boldsymbol{p}, \boldsymbol{r}) = \|\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})\|^2$ when also $0 \leq \delta_k \leq \delta_{max} < 1$ and $\sigma_{max} + \delta_{max} < 1$ are satisfied. The proof runs as in [27, Theorem 2].

3.3 Choice of α_k

After solving the perturbed Newton equation, we need to compute the new iterate using (17). The damping parameter α_k is used to enforce the global convergence of the method and its choice is performed by verifying three conditions: feasibility, centrality and backtracking.

3.3.1 The feasibility condition

The feasibility condition determines the initial value of α, which will then be passed to the centrality conditions. For $k = 0, 1, \ldots,$ the feasibility condition reads

$$\alpha = \min \left\{ \min_{\Delta p_i^{(k)} < 0} \frac{-p_i^{(k)}}{\Delta p_i^{(k)}}, \min_{\Delta r_i^{(k)} < 0} \frac{-r_i^{(k)}}{\Delta r_i^{(k)}}, 1 \right\}. \tag{27}$$

The value of α computed by (27) guarantees the feasibility of

$$\begin{aligned} \boldsymbol{p}^{(k)}(\alpha) &= \boldsymbol{p}^{(k)} + \alpha \Delta \boldsymbol{p}^{(k)} \geq 0; \\ \boldsymbol{r}^{(k)}(\alpha) &= \boldsymbol{r}^{(k)} + \alpha \Delta \boldsymbol{r}^{(k)} \geq 0, \end{aligned}$$

with $\boldsymbol{p}^{(0)} > 0$, $\boldsymbol{r}^{(0)} > 0$, where $\Delta \boldsymbol{p}^{(k)}$ and $\Delta \boldsymbol{r}^{(k)}$ are the approximate solutions of (16) with residual condition (19).

3.3.2 The centrality conditions

The centrality conditions force the iterates $\boldsymbol{p}^{(k)}$ and $\boldsymbol{r}^{(k)}$ to adhere to the central path, forcing them sufficiently far from the boundary of the nonnegative orthant. In this regard, following [21], let us define the functions

$$\varphi^{(k)}(\alpha) \equiv \min_{i=1,\ldots,n} \left(P^{(k)}(\alpha) R^{(k)}(\alpha) \boldsymbol{e} \right) - \gamma_k \tau_1 \left(\frac{\boldsymbol{p}^{(k)}(\alpha)^T \boldsymbol{r}^{(k)}(\alpha)}{n} \right); \tag{28}$$

$$\psi^{(k)}(\alpha) \equiv \boldsymbol{p}^{(k)}(\alpha)^T \boldsymbol{r}^{(k)}(\alpha) - \gamma_k \tau_2 \|\boldsymbol{F}_1(\boldsymbol{p}^{(k)}(\alpha), \boldsymbol{r}^{(k)}(\alpha))\|, \tag{29}$$

where $\gamma_k \in [0.5, 1)$ and

$$\tau_1 \leq \frac{\min_{i=1,\ldots,n} \left(P^{(0)} R^{(0)} \boldsymbol{e} \right)}{(\frac{\boldsymbol{p}^{(0)T} \boldsymbol{r}^{(0)}}{n})}; \qquad \tau_2 \leq \frac{\boldsymbol{p}^{(0)T} \boldsymbol{r}^{(0)}}{\|\boldsymbol{F}_1(\boldsymbol{p}^{(0)}, \boldsymbol{r}^{(0)})\|}, \tag{30}$$

with $\boldsymbol{p}^{(0)} > \boldsymbol{0}$ and $\boldsymbol{r}^{(0)} > \boldsymbol{0}$. Imposing also

$$\sigma_k > \max \left\{ \delta_k \frac{\sqrt{n} + \tau_1 \gamma_k}{1 - \tau_1 \gamma_k}, \delta_k \frac{\sqrt{n} + \tau_2 \gamma_k}{\sqrt{n}} \right\}, \tag{31}$$

it is possible to prove (see [27]) the existence of a value of α satisfying $\varphi^{(k)}(\alpha) \geq 0$ and $\psi^{(k)}(\alpha) \geq 0$. These two inequalities are the centrality conditions[1] and we denote by $\tilde{\alpha}_k$ an α which satisfies them both. In practice, $\tilde{\alpha}_k$ is usually computed recursively by dividing α (provided by the feasibility condition) by a factor larger than 1 until both the centrality conditions are satisfied.

We now need to prove the boundedness of the sequences $\{\boldsymbol{p}^{(k)}\}$ and $\{\boldsymbol{r}^{(k)}\}$, which is required also to prove that $\tilde{\alpha}_k$ is bounded away from zero. To do so, given $\varepsilon \geq 0$, let us define the level set

$$\Omega(\varepsilon) = \left\{ (\boldsymbol{p}, \boldsymbol{r}) : \varepsilon \leq \Phi(\boldsymbol{p}, \boldsymbol{r}) \leq \Phi(\boldsymbol{p}^{(0)}, \boldsymbol{r}^{(0)}); \min_{1 \leq i \leq n} (PR\boldsymbol{e}) \geq \frac{\tau_2}{2} \left(\frac{\boldsymbol{p}^T \boldsymbol{r}}{n} \right); \boldsymbol{p}^T \boldsymbol{r} \geq \frac{\tau_2}{2} \|\boldsymbol{F}_1(\boldsymbol{p}, \boldsymbol{r})\| \right\},$$

with $\boldsymbol{p}^{(0)} > \boldsymbol{0}$ and $\boldsymbol{r}^{(0)} > \boldsymbol{0}$. This last condition is required by Equation (30). The following theorem holds.

Theorem 2. *If $(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \in \Omega(\varepsilon)$, $\varepsilon > 0$, then*

1) the sequence $\left\{ \begin{pmatrix} \boldsymbol{p}^{(k)} \\ \boldsymbol{r}^{(k)} \end{pmatrix} \right\}$ is component-wise bounded away from zero;

2) the sequence of matrices $\left\{ F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^{-1} \right\}$ is bounded;

3) the sequence $\left\{ \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} \right\}$ is bounded.

Proof. In $\Omega(\varepsilon)$, $\varepsilon > 0$, by definition of $\Phi(\boldsymbol{p}, \boldsymbol{r})$, we have

$$\left\| \begin{pmatrix} \boldsymbol{F}_1(\boldsymbol{p}, \boldsymbol{r}) \\ PR\boldsymbol{e} \end{pmatrix} \right\|^2 > 0.$$

So, if $\boldsymbol{F}_1(\boldsymbol{p}, \boldsymbol{r}) = \boldsymbol{0}$, it must hold $PR\boldsymbol{e} \neq \boldsymbol{0}$, which implies $\boldsymbol{p}^T \boldsymbol{r} > 0$. Conversely, $PR\boldsymbol{e} = \boldsymbol{0}$ would imply $\|\boldsymbol{F}_1(\boldsymbol{p}, \boldsymbol{r})\| > 0$ but, by definition of $\Omega(\varepsilon)$, this would mean $\boldsymbol{p}^T \boldsymbol{r} > 0$ and $\min_{1 \leq i \leq n} PR\boldsymbol{e} > 0$, contradicting the hypothesis. Therefore, $PR\boldsymbol{e} = \boldsymbol{0}$ is not possible. The sequences $\{p_i^{(k)}\}$ and $\{r_i^{(k)}\}$ are then bounded away from zero, proving proposition 1.

Let us then consider the properties of A and of B in Section 2.1 and the positivity of $p_j > 0$ and $r_j > 0$, $j = 1, \ldots, n$ (which makes P and R nonsingular). We can prove that $A - BP^{-1}R$ is nonsingular. Indeed, the matrix $-B$ is a column diagonally dominant matrix with positive diagonal elements and non-positive off-diagonal elements. These properties are conserved when $-B$ is multiplied to the right by diagonal matrices with positive diagonal entries, such as P^{-1} and R. It follows that $A - BP^{-1}R$ is the sum between two column diagonally dominant matrices with positive diagonal entries and non-positive off-diagonal entries. Moreover, strict diagonal dominance holds for at least a column index. Hence, $A - BP^{-1}R$ is nonsingular. Indeed, it is irreducibly diagonally dominant and (by the sign of its elements) it is also a nonsingular M-matrix.

Considering a matrix like the Jacobian $F'(\boldsymbol{p}, \boldsymbol{r})$ in (10), the nonsingularity of $A - BP^{-1}R$ and of P implies that the inverse of the Jacobian is (e.g. see [28, p.108])

$$F'(\boldsymbol{p}, \boldsymbol{r})^{-1} = \begin{pmatrix} (A - BP^{-1}R)^{-1} & -(A - BP^{-1}R)^{-1}BP^{-1} \\ -P^{-1}R(A - BP^{-1}R)^{-1} & P^{-1} + P^{-1}R(A - BP^{-1}R)^{-1}BP^{-1} \end{pmatrix}.$$

Since every sub-matrix is bounded, so is also $F'(\boldsymbol{p}, \boldsymbol{r})^{-1}$. Then, $\{F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^{-1}\}$ is uniformly bounded in $\Omega(\varepsilon)$, $\varepsilon > 0$, proving proposition 2.

Finally, let us consider that, by the definition of $\hat{\boldsymbol{r}}^{(k)}$ in (25), we have

$$\begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} = -F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^{-1} \left(\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) + \sigma_k \mu_k \tilde{\boldsymbol{e}} + \hat{\boldsymbol{r}}^{(k)} \right).$$

[1] If we replace $\varphi^{(k)}(\alpha)$ and $\psi^{(k)}(\alpha)$ with their expressions in (28) and (29), it is easy to note that the condition $\varphi^{(k)}(\alpha) \geq 0$ forces the iterates far from $(\boldsymbol{p}, \boldsymbol{r}) \geq 0$, while $\psi^{(k)}(\alpha) \geq 0$ makes the sequence $\{\boldsymbol{p}^{(k)}(\alpha)^T \boldsymbol{r}^{(k)}(\alpha)\}$ to converge to zero slower than $\{\|\boldsymbol{F}_1(\boldsymbol{p}^{(k)}(\alpha), \boldsymbol{r}^{(k)}(\alpha))\|\}$.

Then, let us consider that the boundedness of $\{F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^{-1}\}$ in $\Omega(\varepsilon)$, $\varepsilon > 0$, means that there exists a positive scalar M such that $\|F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^{-1}\| \leq M$ for $(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \in \Omega(\varepsilon)$, $\varepsilon > 0$ and $k \geq 0$. Moreover, the inequalities in (19) and in (26) pose similar conditions on $\sigma_k \mu_k \tilde{\boldsymbol{e}}$ and on $\hat{\boldsymbol{r}}^{(k)}$, respectively. Using these relationships and remembering $\sigma_k + \delta_k \leq \sigma_{\max} + \delta_{\max} < 1$ in $\Omega(\varepsilon)$, $\varepsilon > 0$, we get

$$\begin{aligned}\left\| \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} \right\| &\leq M(1 + \sigma_k + \delta_k)\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\| \\ &\leq M(1 + \sigma_{\max} + \delta_{\max})\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\| \leq 2M\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|.\end{aligned}$$

Therefore, the sequence $\left\{ \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} \right\}$ is bounded for $(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \in \Omega(\varepsilon)$, $\varepsilon > 0$, completing the proof. □

Using this result, it is possible to prove that $\tilde{\alpha}_k$ is bounded away from zero as well. In this regard, see [27, Theorem 6].

3.3.3 The backtracking condition

The value of $\tilde{\alpha}_k$ computed by the feasibility and by the centrality conditions could still be too large to assure a reduction of the merit function $\Phi(\boldsymbol{p}, \boldsymbol{r})$. Thus, we introduce the *backtracking condition*. In this regard, we use the Inexact Newton Backtracking algorithm (INB), which is a line search strategy described in [23] that recursively reduces $\tilde{\alpha}_k$ by the relationship $\alpha_k = \theta^t \tilde{\alpha}_k$, with $\theta \in (0, 1)$ and t nonnegative integer which is gradually increased. The procedure stops as α_k satisfies

$$\|\boldsymbol{F}(\boldsymbol{p}^{(k)} + \alpha_k \Delta\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)} + \alpha_k \Delta\boldsymbol{r}^{(k)})\| \leq (1 - \beta\alpha_k(1 - (\sigma_k + \delta_k)))\, \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|, \tag{32}$$

with $\beta \in (0, 1)$. The proof that t is finite, and, therefore, α_k remains bounded away from zero, runs as that in [27, pp. 364-366].

3.3.4 Convergence of the algorithm

Let us set $\xi_k = 1 - \beta\alpha_k(1 - (\sigma_k + \delta_k))$. By $\beta, \alpha_k \in (0, 1)$ and by $0 < \sigma_k + \delta_k \leq \sigma_{\max} + \delta_{\max} < 1$, we have that ξ_k is uniformly less than 1. Thus, if we replace $1 - \beta\alpha_k(1 - (\sigma_k + \delta_k))$ by ξ_k in (32), the backtracking condition directly gives

$$\|\boldsymbol{F}(\boldsymbol{p}^{(k+1)}, \boldsymbol{r}^{(k+1)})\| \leq \xi_k \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\| \qquad 0 < \xi_k \leq \bar{\xi} < 1. \tag{33}$$

Next, consider the definition of the residual $\hat{\boldsymbol{r}}^{(k)}$ in (25) and its boundedness in (26). Using also (19) and (21) to evaluate, respectively, $\|\mu_k \tilde{\boldsymbol{e}}\|$ and $\|\hat{\boldsymbol{r}}^{(k)}\|$ from above, we find that the step $\alpha_k \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix}$ satisfies the condition

$$\begin{aligned}\left\| F'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \alpha_k \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} + \boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \right\| &= \left\| \alpha_k \left(-\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) + \sigma_k \mu_k \tilde{\boldsymbol{e}} + \hat{\boldsymbol{r}}^{(k)} \right) + \boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \right\| \\ &\leq (1 - \alpha_k)\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\| + \alpha_k \left(\sigma_k \|\mu_k \tilde{\boldsymbol{e}}\| + \|\hat{\boldsymbol{r}}^{(k)}\| \right) \\ &\leq (1 - \alpha_k(1 - (\sigma_k + \delta_k)))\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\| \\ &= \eta_k \|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|,\end{aligned} \tag{34}$$

where $\eta_k \coloneqq 1 - \alpha_k(1 - (\sigma_k + \delta_k))$. From the considerations made on ξ_k, we notice that $\eta_k \in (0, 1)$ as well.

Inequalities (33) and (34) correspond to the convergence conditions of an inexact Newton method [29, §6.4]:

if the sequence $\{(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}\}$ has limit point $(\boldsymbol{p}^*, \boldsymbol{r}^*)$ where $F'(\boldsymbol{p}^*, \boldsymbol{r}^*)$ is nonsingular then

$$\lim_{k\to\infty} \boldsymbol{p}^{(k)} = \boldsymbol{p}^*; \qquad \lim_{k\to\infty} \boldsymbol{r}^{(k)} = \boldsymbol{r}^*,$$

and $\boldsymbol{F}(\boldsymbol{p}^*, \boldsymbol{r}^*) = \boldsymbol{0}$.

Moreover, it is also possible to show (see [30, p. 9]) that the damped inexact Newton method has a super-linear local convergence.

3.4 A note on the Fischer-Burmeister-Newton iteration

As mentioned earlier, complementarity problems connected with hydrodynamic lubrication have been recently solved in [14] using a solution algorithm based on the Fischer-Burmeister-Newton (FBN) method, which has been introduced in [31].

In the FBN approach, $\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})$ in (8) is replaced by

$$\tilde{\boldsymbol{F}}(\boldsymbol{p}, \boldsymbol{r}) = \begin{pmatrix} A\boldsymbol{p} + B\boldsymbol{r} - \boldsymbol{c} \\ \sqrt{p_1^2 + r_1^2} - p_1 - r_1 \\ \vdots \\ \sqrt{p_n^2 + r_n^2} - p_n - r_n \end{pmatrix}. \tag{35}$$

and the Newton method is then applied to the nonlinear system $\tilde{\boldsymbol{F}}(\boldsymbol{p}, \boldsymbol{r}) = \boldsymbol{0}$.

The definition of the method follows, then, (11) and (12) with $\tilde{\boldsymbol{F}}$ instead of $\boldsymbol{F}$. Here we call $(\Delta\boldsymbol{p}^{(k)}, \Delta\boldsymbol{r}^{(k)})$ the solution of the linear system (12) applied to $\tilde{\boldsymbol{F}}(\boldsymbol{p}, \boldsymbol{r}) = \boldsymbol{0}$.

It is easy to show that also the FBN method suffers of the problem that, if a component of $\boldsymbol{p}^{(k)}$ or of $\boldsymbol{r}^{(k)}$ reaches the boundary of the feasible region, it sticks to it also in the successive iterations.

In order to ensure global convergence, we assume the non-singularity of the Jacobian of $\tilde{\boldsymbol{F}}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})$ and we consider two approaches:

1. Introduce a damping parameter α_k satisfying the Armijo rule [32]

$$\tilde{\Phi}(\boldsymbol{p}^{(k+1)}, \boldsymbol{r}^{(k+1)}) \leq \tilde{\Phi}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) + \tilde{\beta}\alpha_k \nabla\tilde{\Phi}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})^T \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix}, \tag{36}$$

with $\tilde{\beta} \in (0, 1)$, e.g., $\tilde{\beta} = 10^{-4}$. In this case, the global convergence can be proved by showing that $(\Delta\boldsymbol{p}^{(k)}, \Delta\boldsymbol{r}^{(k)})$ is a descent direction for the merit function $\tilde{\Phi}(\boldsymbol{p}, \boldsymbol{r}) = \|\tilde{\boldsymbol{F}}(\boldsymbol{p}, \boldsymbol{r})\|^2$ and then following the convergence scheme for the line search method in [33, §3.1, §3.2].

2. Consider an inner iterative solver with stopping rule $\|\hat{\boldsymbol{r}}^{(k)}\| \leq \delta_k \|\tilde{\boldsymbol{F}}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|$, where $\delta_k \leq \delta_{max} < 1$ and $\hat{\boldsymbol{r}}^{(k)}$ is the residual

$$\hat{\boldsymbol{r}}^{(k)} = \tilde{F}'(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)}) \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} + \tilde{\boldsymbol{F}}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})$$

and use the backtracking condition (32) with $\sigma_k = 0$. In this case, the method becomes an inexact Newton method. Therefore, one has to prove that the inequalities (33) and (34) are satisfied. In this regard, defined the compact set

$$\tilde{\Omega}(\varepsilon) = \{(\boldsymbol{p}, \boldsymbol{r}) \quad : \varepsilon \leq \tilde{\Phi}(\boldsymbol{p}, \boldsymbol{r}) \leq \tilde{\Phi}(\boldsymbol{p}^{(0)}, \boldsymbol{r}^{(0)})$$

it is easy to prove the boundedness of the sequence $\left\{ \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix} \right\}$ in $\tilde{\Omega}(\varepsilon)$, $\varepsilon > 0$, by taking into account the non-singularity of the Jacobian matrix. We can then make the same considerations of the DIN method and, setting $\xi_k = 1 - \tilde{\beta}\alpha_k(1 - \delta_k)$ and $\eta_k = 1 - \alpha_k(1 - \delta_k)$, we can show that both the convergence inequalities (33) and (34) are satisfied with $\tilde{F}$ and $\tilde{F}'$ instead of F and F'.

It is possible to show that this method converges also for $\delta_k = 0$ (which is, for a direct inner solver) by following the theorem [34, §6.3], which regards the convergence of a sequence $\{\boldsymbol{v}^{(k)}\}$ (where, in our case, $\boldsymbol{v}^{(k)}$ is $(\boldsymbol{p}^{(k)^T}, \boldsymbol{r}^{(k)^T})^T$).

4 Numerical experiments

We now introduce the numerical experiments, performed using a Fortran implementation of the method. In the following, we define the analyzed problems and the numerical parameters used in the experiments.

4.1 Setting of the algorithm

The parameters of the solver are reported in Table 1, where $\tau_1^{(\max)}$ and $\tau_2^{(\max)}$ are the upper bounds for τ_1 and τ_2 as in (30) and θ is the parameter multiplying α_k until feasibility, centrality and backtracking condition are satisfied.

Table 1 Data used in the numerical experiments for setting the DIN solver

τ_1	τ_2	μ_k	β	θ	γ_k
$\min\left(\frac{\tau_1^{(\max)}\cdot 10^{-7}}{2}, 0.99\right)$	$\tau_2^{(\max)}\cdot 10^{-7}$	$\mu_k^{(2)}$	10^{-4}	0.5	$0.5, \forall k$

Other parameters depend on the type of problem we are solving. Indeed, in 1D cases with coarser discretizations, we use a direct inner solver (Gaussian elimination) so to analyze the behavior of the DIN algorithm itself, without having to deal with the additional residual given by an iterative inner solver, which would require further remarks. We therefore set

$$\sigma_k = 0.01\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|; \quad \sigma_{\max} = 0.9; \quad \delta_k = 0.$$

The same applies to the more efficient direct solvers used in Section 6.

When we employ iterative solvers, on the other hand, $\delta_k + \sigma_k < 1$ and (31) must hold. We thus set

$$\sigma_k = 0.01\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|; \quad \sigma_{\max} = 0.9; \quad \delta_k = \min\left(0.1, \frac{\hat{\epsilon}\sigma_k}{\max\left(\frac{\sqrt{n}+\tau_1+\gamma_k}{1-\tau_1\gamma_k}, \frac{\sqrt{n}+\tau_2\gamma_k}{\sqrt{n}}\right)}\right)$$

with $\hat{\epsilon} < 1$. When not otherwise specified, we set $\hat{\epsilon} = 1/n$. Since σ_k can become very small, we also impose a lower bound to the tolerance, which at the kth iteration is then chosen as

$$tol_{inner}^k = \max\left(tol_{\min}, \delta_k\|\boldsymbol{F}(\boldsymbol{p}^{(k)}, \boldsymbol{r}^{(k)})\|\right).$$

Finally, the Newton iteration is stopped when the stopping criteria

$$c_1 \equiv \|\boldsymbol{F}(\boldsymbol{p}^{(k+1)}, \boldsymbol{r}^{(k+1)})\| \leq \varepsilon_1 \quad \text{and} \quad c_2 \equiv \left\|\alpha_k \begin{pmatrix} \Delta\boldsymbol{p}^{(k)} \\ \Delta\boldsymbol{r}^{(k)} \end{pmatrix}\right\| \leq \varepsilon_2 \tag{37}$$

are satisfied, where $\epsilon_1 = \epsilon_2 = 10^{-8}$ when not otherwise specified. The maximum number of DIN iterations is set at 500, while the maximum number of inner iterations is set at $10,000$.

4.2 Numerical data of the test problems

In 1D cases, denoting by $h_{\min}$ and $h_{\max}$, respectively, the minimum and the maximum thickness of the fluid film and by L the length of the interval $[a, b]$, we set

$$h_{\min} = 0.015; \quad h_{\max} = 0.025; \quad L = 100; \quad \mu = 0.015; \quad U = 5. \tag{38}$$

We consider both homogeneous and non-homogeneous Dirichlet boundary conditions on p. This describes the two physical situations that can occur: in the first case, we are starting in a cavitated situation, while in

the second the entire phenomenon happens inside the domain. We also modify $h(x)$, changing the shape of the film. In particular, we reproduce the *Convergent-Divergent* (C-D) and the *Divergent-Convergent*(D-C) schematics (see [12]). We can reproduce the C-D schematics by defining $h(x)$ as a convex function passing by $h_{\max}$ in $x = a$ and in $x = b$ and by $h_{\min}$ in $x = (a + b)/2$. In this regard, we define $h(x)$ as a parabola $h(x) = ax^2 + bx + c$ or as a sinusoid

$$h(x) = A \sin\left(\frac{2\pi}{\lambda}x + \phi\right) + K$$

of wavelength $\lambda = L$. The D-C schematics is reproduced analogously with concave functions. In the following, when we speak of *divergent* and *convergent* parts of the domain, we thus refer to the reducing or increasing thickness of the film, respectively.

Depending on the boundary conditions and on the configuration of the film, we define the following three test problems, embedding the cases most commonly analyzed in the literature:

Problem 1: C-D configuration with $p(a) = p(b) = 0$, $r(a) = 0$;

Problem 2: C-D configuration with $p(a) = p(b) = 1$, $r(a) = 0$;

Problem 3: D-C configuration with $p(a) = p(b) = 1$, $r(a) = 0$.

Finally, regarding the 2D case, we consider a square domain Ω $[0, L] \times [0, L]$ where $h(x, y)$ is described by a sinusoid with C-D configuration from $x = 3L/8$ to $x = 5L/8$ and by $h(x, y) = h_{\max}$ elsewhere. We use the same numerical values of (38). Calling $\partial\Omega$ the boundary of the domain, we thus define

Problem 4: C-D configuration with $p(\partial\Omega) = 0$, $r(0, y) = 0$.

5 Results and analysis

5.1 Validation of the method

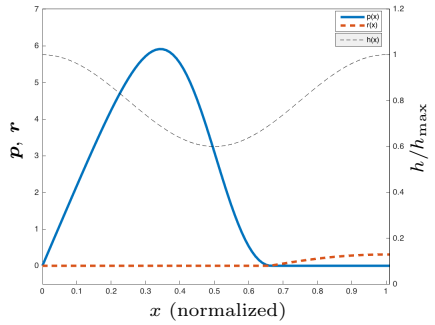
In this subsection, we present the results obtained by solving *Problem 1*, *Problem 2* and *Problem 3* with the numerical data in (38) (scaled so that the result for $\boldsymbol{p}$ is in MegaPascal, as commonly done in the literature) by the DIN method. The film thickness $h(x)$ is here described by a sinusoid as in Section 4.2 and the domain is discretized by $n = 100$ inner points. Figure 1 provides the plots of the pressure for all these cases, along with the plot of r and of $h(x)$.

The C-D configuration with homogeneous Dirichlet conditions on $\boldsymbol{p}$ is the simplest case, whose solution is provided by several articles in the literature (e.g. see [12], where also [35, 36] are cited). The trend of the pressure computed with our code is the same as in these works, as it can be seen in Figure 1a. Moreover, we also see that the complementarity condition is always respected.

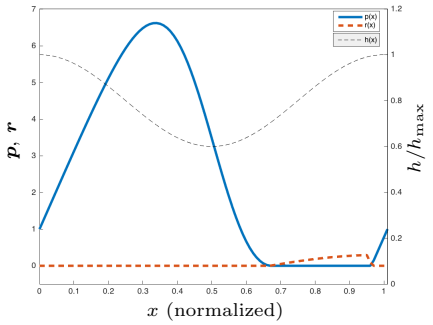
The same can be said for *Problem 2* (Figure 1b) and for *Problem 3* (Figure 1c). This latter situation is probably the most interesting one: indeed, as stated in [12], some formulations (like those in [9] and in [35]) fail to give correct results for *Problem 3* since they assume that the cavitation can occur only in the divergent parts of the profile. This leads to some inaccuracies at the boundary of *Problem 2* as well. On the contrary, the used cavitation model leads to physically realistic solutions: in Figure 1c, for instance, cavitation occurs in the converging part of the film, as in [12]. All the computed profiles are, thus, in accordance with the most recent results found in the literature.

To further assess the validity of our approach, we also formulate $h(x)$ in a different way. As mentioned in Section 4.2, we consider $h(x)$ described by a parabola, as well. Considering, for example, *Problem 1*, in Figure 2 we compare the results obtained for a parabolic and a sinusoidal $h(x)$.

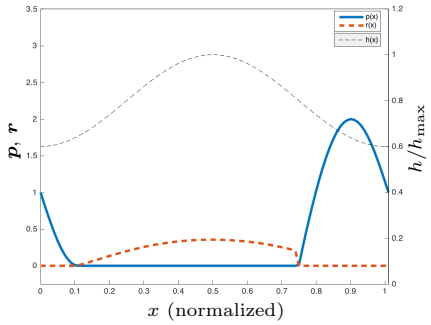
Qualitatively, the pressure profiles are really similar. However, as it can be intuitively expected, a less abrupt decrease in thickness leads to lower peaks of pressure. The behavior is the same also for *Problem 2* and *Problem 3*.



(a) Results for Problem 1



(b) Results for Problem 2



(c) Results for Problem 3

Figure 1 Plots of $h(x)$ and of the solutions $\boldsymbol{p}$ and $\boldsymbol{r}$ of *Problem 1*, *Problem 2* and *Problem 3* solved with the DIN method and Gaussian elimination as inner solver.

Finally, in Table 2 we report the number of DIN, centrality and backtracking iterations, together with the value of the stopping criteria c_1 and c_2 (defined like in (37)) at convergence. In Table 3 we instead report the results obtained by solving *Problem 1* with various discretizations using GMRES as inner solver with the parameters in Section 4 with $tol_{\min} = 10^{-12}$. Here the implementation of the GMRES follows [37, p.45] with Givens rotations and re-orthogonalization. We remark that we are now not concerned with the efficiency of the method, but only with its validation with direct and iterative solvers. Efficient implementations are instead studied in Section 6.

Table 2 Results obtained with the DIN method

Problem	*h(x)*	*it*	*back*	*cent*	c_1	c_2
1	Sinusoid	30	0	26	$1.55 \cdot 10^{-18}$	$5.96 \cdot 10^{-9}$
	Parabola	30	0	26	$4.68 \cdot 10^{-17}$	$2.99 \cdot 10^{-13}$
2	Sinusoid	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	Parabola	46	0	46	$5.02 \cdot 10^{-12}$	$5.24 \cdot 10^{-9}$
3	Sinusoid	58	0	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	Parabola	56	0	56	$1.11 \cdot 10^{-11}$	$5.58 \cdot 10^{-9}$

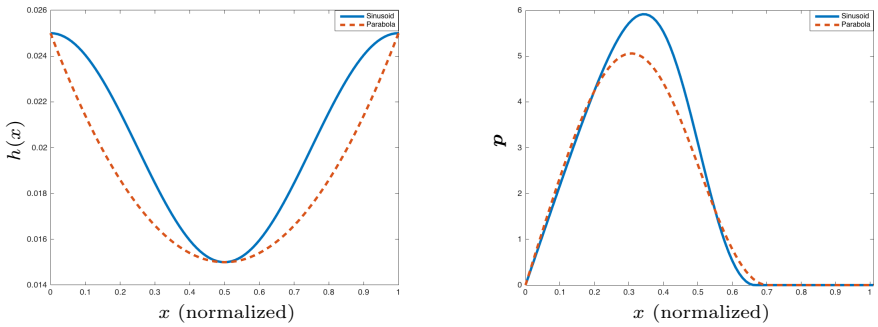


Figure 2 Comparison of the results obtained for $h(x)$ described by a parabola or by a sinusoid. Left: plot of $h(x)$. Right: plot of solution $\boldsymbol{p}$.

Table 3 Results obtained with different discretizations for Problem 1 with sinusoidal $h(x)$ and adaptive linear tolerance with $\hat{\epsilon} = 1/n$ and $tol_{\min} = 10^{-12}$

n	it	$back$	$cent$	GMRES	c_1	c_2
50	52	0	52	4,114	$9.90 \cdot 10^{-13}$	$5.82 \cdot 10^{-9}$
100	29	0	26	5,130	$1.87 \cdot 10^{-11}$	$2.48 \cdot 10^{-11}$
200	44	0	48	15,256	$6.49 \cdot 10^{-10}$	$1.94 \cdot 10^{-10}$
500	49	0	48	43,006	$4.09 \cdot 10^{-12}$	$3.30 \cdot 10^{-9}$
1,000	50	0	49	86,546	$1.04 \cdot 10^{-11}$	$2.96 \cdot 10^{-9}$

5.2 Role of the conditions on α_k

We now analyze what happens when we remove some conditions (like feasibility, centrality or backtracking) in the choice of the step-size α_k. We also consider what happens if the reduction of Argaez, Tapia and Velazquez[2] is introduced. The results of this analysis are reported in Table 4, where the different conditions on α_k are marked in the following way:

- F marks the presence of the feasibility condition;
- C marks the presence of the centrality condition;
- B marks the presence of the backtracking condition;
- R marks the presence of the Argaez, Tapia and Velazquez reduction.

Finally, the star $*$ next to the value of the stopping conditions c_1 and c_2 indicates that the algorithm converged, but to a wrong solution.

We notice that the reduction of Argaez, Tapia and Velazquez does indeed reduce the number of iterations: indeed, the centrality conditions are not triggered anymore in any of the considered problems, and the number of the DIN iterations is almost halved.

Although both centrality and backtracking are, in general, needed for the algorithm to converge, we notice that backtracking is never triggered in the analyzed cases. Therefore, the solution does not change when we remove it. Similarly, Table 3 shows not only that the algorithm converges when we remove the centrality

[2]This reduction, introduced in [38], consists in reducing the value of α_k computed by the feasibility conditions multiplying it by a factor $\hat{\theta}$ defined as

$$\hat{\theta} = \begin{cases} \max(0.8, 1 - 100(\boldsymbol{p}^{(k)^T} \boldsymbol{r}^{(k)})) & \text{if } \alpha_k = 1 \\ \max(0.8, \min(0.9995, 1 - 100(\boldsymbol{p}^{(k)^T} \boldsymbol{r}^{(k)}))) & \text{if } \alpha_k < 1 \end{cases}$$

Table 4 Results obtained adding or removing conditions on the choice of α_k

Problem	*Conditions*	*it*	*back*	*cent*	c_1	c_2
1	FCB	30	0	26	$1.55 \cdot 10^{-18}$	$5.96 \cdot 10^{-9}$
	FRCB	19	0	0	$3.88 \cdot 10^{-19}$	$2.39 \cdot 10^{-10}$
	FB	16	0	-	$4.78 \cdot 10^{-19}$	$5.12 \cdot 10^{-10}$
	FC	30	0	26	$1.55 \cdot 10^{-18}$	$5.96 \cdot 10^{-9}$
	CB	26	0	29	$2.10 \cdot 10^{-15}$	$3.95 \cdot 10^{-12}$
	B	15	7	-	$4.40 \cdot 10^{-19}*$	$5.02 \cdot 10^{-11}*$
	C	26	-	29	$2.10 \cdot 10^{-15}$	$3.95 \cdot 10^{-12}$
2	FCB	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	FRCB	20	0	1	$2.14 \cdot 10^{-15}$	$3.73 \cdot 10^{-12}$
	FB	17	0	-	$8.10 \cdot 10^{-19}$	$6.47 \cdot 10^{-13}$
	FC	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	CB	41	0	49	$7.71 \cdot 10^{-12}$	$8.30 \cdot 10^{-9}$
	B	15	26	-	$1.61 \cdot 10^{-18}*$	$6.02 \cdot 10^{-9}*$
	C	41	-	49	$7.71 \cdot 10^{-12}$	$8.30 \cdot 10^{-9}$
3	FCB	58	0	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	FRCB	36	0	7	$3.73 \cdot 10^{-16}$	$1.18 \cdot 10^{-9}$
	FB	19	0	-	$8.17 \cdot 10^{-20}$	$8.36 \cdot 10^{-13}$
	FC	58	-	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	CB	51	0	79	$5.84 \cdot 10^{-12}$	$6.82 \cdot 10^{-9}$
	B	23	124	-	$3.68 \cdot 10^{-19}*$	$2.36 \cdot 10^{-11}*$
	C	51	-	79	$5.84 \cdot 10^{-12}$	$6.82 \cdot 10^{-9}$

conditions, but also that α_k still does not trigger backtracking. Therefore, α_k is never reduced after the feasibility condition. This results in a larger step, which justifies the smaller number of iterations. However, we remark that this applies to the analyzed problems and it does not mean that, in general, disregarding the centrality conditions leads to higher efficiency or to faster convergence: both centrality and backtracking are required for ensuring convergence as in Section 3.

If we also remove the feasibility condition, the centrality conditions become fundamental in order to compute the correct solution. Otherwise, the backtracking condition still ensures that the algorithm converges, but the computed solution is not correct. Indeed, the solution gets unstable in the parts of $\boldsymbol{p}$ and of $\boldsymbol{r}$ close to zero, where we start having also large oscillations and negative values. The reductions performed by the backtracking conditions alone are, therefore, not sufficient, since the adherence to the central path is granted no more.

On the other hand, removing the feasibility condition when preserving the centrality does not largely affect the results, but it only increases the number of required centrality iterations. This could be expected also from the analysis of the algorithm: indeed, the centrality satisfies the feasibility, which is introduced in order to pass a "better", already feasible α_k to the centrality.

5.3 Role of the perturbation parameter μ_k and of the starting vectors $\boldsymbol{p}^{(0)}$ and $\boldsymbol{r}^{(0)}$

In Table 1 we chose μ_k equal to the upper bound $\mu_k^{(2)}$, thus obtaining a greater perturbation. Arguably, this helps avoiding stagnation when we are close to the boundary. In order to see this, let us see what happens when we change μ_k and the starting vectors $\boldsymbol{p}^{(0)}$ and $\boldsymbol{r}^{(0)}$. The results are reported in Table 5, where *maxit* indicates that the maximum number of DIN iterations has been reached before convergence.

We notice that the convergence is never faster for $\mu_k = \mu_k^{(1)}$. On the contrary, it gets much slower as the initial iterates $\boldsymbol{p}^{(0)}$ and $\boldsymbol{r}^{(0)}$ are reduced. Indeed, when the initial iterates are large, the difference is

Table 5 Results with different starting vectors $\boldsymbol{p}^{(0)}$ and $\boldsymbol{r}^{(0)}$ and different choices of μ_k

Problem	$\boldsymbol{p}^{(0)}$	$\boldsymbol{r}^{(0)}$	μ_k	*it*	*back*	*cent*	c_1	c_2
1	1	0.1	$\mu_k^{(1)}$	43	0	42	$2.78 \cdot 10^{-12}$	$5.44 \cdot 10^{-9}$
	1	0.1	$\mu_k^{(2)}$	30	0	26	$1.55 \cdot 10^{-18}$	$5.96 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(1)}$	74	0	74	$1.91 \cdot 10^{-15}$	$5.35 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(2)}$	74	0	74	$1.82 \cdot 10^{-15}$	$5.09 \cdot 10^{-9}$
	10^{-15}	10^{-15}	$\mu_k^{(1)}$	*maxit*	0	536	$1.11 \cdot 10^{-5}$	$3.68 \cdot 10^{-7}$
	10^{-15}	10^{-15}	$\mu_k^{(2)}$	83	21	82	$2.80 \cdot 10^{-15}$	$7.86 \cdot 10^{-9}$
2	1	0.1	$\mu_k^{(1)}$	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	1	0.1	$\mu_k^{(2)}$	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(1)}$	70	0	70	$4.12 \cdot 10^{-14}$	$5.96 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(2)}$	35	0	34	$5.14 \cdot 10^{-12}$	$7.55 \cdot 10^{-9}$
	10^{-15}	10^{-15}	$\mu_k^{(1)}$	*maxit*	0	503	$2.48 \cdot 10^{-4}$	$1.06 \cdot 10^{-10}$
	10^{-15}	10^{-15}	$\mu_k^{(2)}$	71	30	70	$3.97 \cdot 10^{-14}$	$5.74 \cdot 10^{-9}$
3	1	0.1	$\mu_k^{(1)}$	58	0	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	1	0.1	$\mu_k^{(2)}$	58	0	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(1)}$	89	0	89	$4.41 \cdot 10^{-14}$	$8.11 \cdot 10^{-9}$
	10^{-5}	10^{-5}	$\mu_k^{(2)}$	88	0	88	$5.15 \cdot 10^{-14}$	$7.73 \cdot 10^{-9}$
	10^{-15}	10^{-15}	$\mu_k^{(1)}$	*maxit*	0	21,059	$5.47 \cdot 10^{-5}$	$2.15 \cdot 10^{-32}$
	10^{-15}	10^{-15}	$\mu_k^{(2)}$	78	26	77	$4.76 \cdot 10^{-14}$	$7.13 \cdot 10^{-9}$

negligible, especially for *Problem 2* and for *Problem 3*, for which the results are indistinguishable in the two cases. However, for $\boldsymbol{p}^{(0)} = \boldsymbol{r}^{(0)} = 10^{-15}$, the algorithms do not converge if the perturbation parameter is too small: we are indeed so near to the non-negative orthant that several centrality iterations are triggered, leading the algorithm to stall. This is particularly evident for *Problem 3*.

Another effective choice is to set the perturbation parameter in an adaptive way. This can be done using the condition

$$\begin{aligned}&\text{If } {\boldsymbol{p}^{(k)}}^T \boldsymbol{r}^{(k)} > 0.1\\ &\mu_k = \mu_k^{(1)}\\ &\text{else}\\ &\mu_k = \mu_k^{(2)}.\end{aligned}$$

In the considered cases, this leads exactly to the same results computed choosing $\mu_k = \mu_k^{(2)}$. In other cases, however, the accuracy of the solution can increase if we choose $\mu_k = \mu_k^{(1)}$ when possible.

5.4 Comparison with the Fischer-Burmeister-Newton method

Finally, we compare the DIN method with the FBN method. As mentioned before, in [14] the FBN iteration has been used for the solution of lubrication problems and it is remarked that this algorithm is very efficient. However, it has to be noted that it also accepts iterates $\boldsymbol{p}^{(k)}$ and $\boldsymbol{r}^{(k)}$ whose components are negative: indeed, there is nothing preventing this, while in DIN the non-negativity conditions are enforced by the perturbation $\tilde{\rho}_k$. We now show that this difference is actually visible in the numerical experiments.

Solving the one-dimensional problems by the FBN method, we get the results in Table 6, where FBN-A denotes the FBN method with Armijo condition (36) and FBN-I denotes the FBN method with inexact Newton backtracking condition (26) with $\tilde{\boldsymbol{F}}$ instead of $\boldsymbol{F}$ and $\sigma_k = \delta_k = 0$. In all the cases, we set $n = 100$.

Table 6 Results for *Problem 1*,*Problem 2* and *Problem 3* computed with the DIN and FBN methods

Problem	*Method*	*it*	*back*	*cent*	c_1	c_2
1	DIN	30	0	26	$1.55 \cdot 10^{-18}$	$5.96 \cdot 10^{-9}$
	FBN-A	13	15	-	$3.59 \cdot 10^{-19}$	$5.57 \cdot 10^{-13}$
	FBN-I	13	15	-	$3.59 \cdot 10^{-19}$	$5.57 \cdot 10^{-13}$
2	DIN	46	0	46	$6.63 \cdot 10^{-12}$	$6.38 \cdot 10^{-9}$
	FBN-A	13	13	-	$6.83 \cdot 10^{-19}$	$8.56 \cdot 10^{-13}$
	FBN-I	13	13	-	$6.83 \cdot 10^{-19}$	$8.56 \cdot 10^{-13}$
3	DIN	58	0	58	$9.31 \cdot 10^{-12}$	$7.83 \cdot 10^{-9}$
	FBN-A	24	62	-	$9.26 \cdot 10^{-20}$	$1.05 \cdot 10^{-13}$
	FBN-I	24	62	-	$9.26 \cdot 10^{-20}$	$1.05 \cdot 10^{-13}$

We see that both the FBN methods converge in less iterations than the DIN method. This is a good measure of the computational cost as well: indeed, as better described in the next section, the solution of the linear systems arising at each Newton iteration is arguably the most expensive part of the procedure. However, the faster convergence of the FBN iterations is likely linked to the specific problems we analyzed: for these particular problems, similar and even better performances were obtained in Table 4 with the DIN method when only feasibility and backtracking conditions were applied.

Moreover, the conditions $\boldsymbol{p} \geq \mathbf{0}$, $\boldsymbol{r} \geq \mathbf{0}$ are not always respected. This is shown in Table 7, which represents the value of some components of $\boldsymbol{p}$ in points where $\boldsymbol{p}$ should be equal to zero. The table refers to *Problem 1*, but the same applies to the other analyzed problems as well. We see that $\boldsymbol{p}$ computed with DIN is bounded

Table 7 Comparison of some values p_i of $\boldsymbol{p}$ obtained solving *Problem 1* with DIN and FBN methods

i	p_i DIN	p_i FBN-A	p_i FBN-I
70	$1.59 \cdot 10^{-25}$	$3.11 \cdot 10^{-18}$	$3.11 \cdot 10^{-18}$
71	$1.27 \cdot 10^{-25}$	$-6.42 \cdot 10^{-18}$	$-6.42 \cdot 10^{-18}$
72	$1.06 \cdot 10^{-25}$	$2.10 \cdot 10^{-18}$	$2.10 \cdot 10^{-18}$
73	$9.08 \cdot 10^{-26}$	$4.33 \cdot 10^{-18}$	$4.33 \cdot 10^{-18}$
74	$7.99 \cdot 10^{-26}$	$5.65 \cdot 10^{-18}$	$5.65 \cdot 10^{-18}$
75	$7.15 \cdot 10^{-26}$	$-1.19 \cdot 10^{-18}$	$-1.19 \cdot 10^{-18}$

away from zero: its components never get smaller than 10^{-26}, ensuring that the non-negativity conditions of the complementarity are satisfied. On the other hand, the FBN implementations oscillate around zero, allowing both positive and negative values for the components of the vector. This violation is here below machine precision and hence it is not concerning for the analyzed problems and for the chosen tolerance. Nonetheless, it is interesting to notice the effect of the absence of something which strongly enforces the non-negativity of the iterates. Moreover, if we choose $\epsilon_1 = \epsilon_2 = 10^{-4}$ (as done for the DIN in the next section) FBN iterations present a few values in the order of -10^{-9}. The DIN iteration is, thus, more stable and the solutions it computes approach zero smoothly and without oscillations, as in Table 7.

6 An efficient implementation and a 2D example

In the previous Section, we validated the DIN method and we analyzed the procedure in different situations. However, we were not concerned with the efficiency of the implementation. This is testified, for instance, by the large number of linear iterations required in Table 3, which, in turn, greatly affects computational times.

In this subsection, we instead focus on an efficient solution of the problem. The heart of a fast implementation of the DIN method relies in the efficient solution of the inner linear system arising at each DIN iteration. Indeed, this is arguably the most onerous part of the procedure, since the computation of the Jacobian is not onerous (see (10)) and the choice of an appropriate step-length α_k is performed by algebraic operations which are not much computationally expensive. We also relax the outer tolerances to $\epsilon_1 = \epsilon_2 = 10^{-4}$, which is enough to compute accurate results. Indeed, with this choice, in the analyzed cases the norm of $\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{r})$ at convergence does not exceed 10^{-6} (and it usually is in the range $10^{-9} \div 10^{-11}$), while $c_2 < \epsilon_2 = 10^{-4}$ is sufficient to ensure that the solution does not change rapidly between two successive iterations.

We consider different kinds of inner solvers:

- direct methods for full matrices;
- direct methods for sparse matrices;
- preconditioned iterative methods.

In order to guarantee an efficient implementation and to enhance reproducibility, we employ well-known software libraries. In particular, the used direct solver for full matrices is given by the LU solver of the Lapack libraries. Regarding sparse direct solvers, we consider SuperLU [39, 40] and MUMPS [41, 42] called through the PETSc suite[3]. Finally, we use preconditioners and iterative solvers of the PETSc package. In this last case, the inner systems are solved inexactly and the tolerance of the linear solver is chosen as in Section 4 with $tol_{\min} = \min\{\epsilon_1, \epsilon_2\} = 10^{-4}$. We always used left preconditioning, which is also the default setting for most Krylov solvers in PETSc.

In the following, we consider *Problem 1* (representing a 1D example) and *Problem 4* (2D example). In tables, the columns titled *time* report computational times (expressed in seconds) for running the entire algorithm in a Unix environment on a laptop with a dual core 2.7 GHz Intel "Core i5" processor ("Broadwell" series).

Table 8 Solution of the inner systems by LU factorization

n	*it*	*back*	*cent*	*time*
100	29	0	26	0.17
500	38	0	37	2.47
1000	39	0	38	17.74

Starting from the 1D case, in Table 8 we report the results obtained with the LU factorization. When n is small, the solver works efficiently, but computational times rapidly increase as the order of the Jacobian matrix increases. Thus, exploiting the sparsity of the Jacobian becomes crucial for problems of large dimensions.

To this end, we can use direct solvers for sparse matrices (Table 9) or iterative solvers (Table 10). We see that direct solvers for sparse matrices behave much better than those for full matrices and are still efficient also when we have thousands of variables.

Regarding iterative solvers, in Table 10 we report the results obtained by using GMRES or BiCG-STAB with ILU preconditioner. Other common preconditioners (such as Jacobi, SOR, etc.) did not behave well, with the sole exception of the block-Jacobi preconditioner. On the other hand, ILU preconditioning reduces the number of linear iterations consistently, as we can see in Table 10, where also computational times approach those of direct methods for sparse matrices.

Passing to the 2D case, the order of the Jacobian matrix rapidly increases with n: each block of the Jacobian matrix is of order n^2. Moreover, A is block-tridiagonal and it is more sparse (although we have more nonzero elements in each row). Therefore, we now only consider solvers for sparse matrices and iterative

[3]The use of PETSc allows also to easily change solver and it provides a parallel implementation of the linear solvers through MPI. For better reproducibility, all the result here reported have however been obtained by running the programs sequentially on a single core.

Table 9 Solution of inner systems by a sparse direct solver. SuperLU (left) and MUMPS (right)

n	*it*	*back*	*cent*	*time*
100	29	0	26	0.14
1,000	39	0	38	0.73
5,000	45	0	44	3.79
10,000	49	0	48	8.46

n	*it*	*back*	*cent*	*time*
100	29	0	26	0.19
1,000	39	0	38	1.11
5,000	45	0	44	10.22
10,000	49	0	48	21.81

Table 10 Solution of inner systems by a preconditioned iteartive solver. GMRES solver with ILU preconditioner (left) and BiCG-STAB solver with ILU preconditioner (right)

n	*it*	*back*	*cent*	GMRES	*time*
1,000	41	0	41	1,779	1.22
10,000	50	0	50	14,064	30.30

n	*it*	*back*	*cent*	BiCGS	*time*
1,000	41	0	41	2,572	1.72
10,000	35	61	35	9,849	23.54

solvers. The results obtained using SuperLU and ILU-preconditioned GMRES are reported in Table 11, while Figure 3 represents the solution computed by ILU-preconditioned GMRES with $n = 100$. We notice that the iterative solver becomes more competitive as n increases: computational times are practically equal when $n = 100$. For yet larger systems, it is also advisable to employ multigrid preconditioners, in order to make the number of linear iterations more independent of the order of the problem.

Table 11 Comparison between sparse direct solver and iterative solver. SuperLU (left) and GMRES solver with ILU preconditioner (right)

n	*it*	*back*	*cent*	*time*
50	65	0	65	15.55
100	101	0	101	609.96

n	*it*	*back*	*cent*	GMRES	*time*
50	66	0	66	9,029	19.44
100	100	0	100	35,367	625.77

Lastly, we conclude with some information on saddle-point (e.g. see [43]) and on augmentation-based preconditioners (e.g. see [44]), which are commonly used in the solution of systems similar to those we need to solve at each DIN iteration. Since these preconditioners are not natively implemented in PETSc, a direct comparison with the results previously reported would be inconsistent. It appears nonetheless suitable to provide some remarks on their implementation for the problems of our concern.

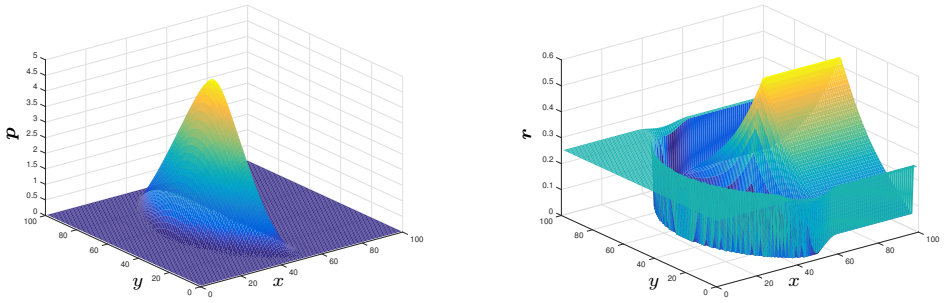
Starting from augmentation-based preconditioners, they are generally applied to solving systems where the $(2, 2)$ block of the coefficient matrix is null or symmetric negative (semi)definite. Remembering the form of the Jacobian (10), we can put ourselves in this situation by left-multiplying both sides of our Newton's system by the matrix

$$\tilde{I} = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix of order n. In the application of the preconditioner, a positive definite weight matrix W must then be chosen. A common choice is to set it as the $(2, 2)$ block of the coefficient matrix changed of sign, which is $W = P$. This, however, can give some numerical difficulties as some diagonal elements of P approach zero, especially when we want to solve the complementarity problem using a small tolerance. In this case, it is therefore advisable to use other diagonal weight matrices, such as the identity matrix.

Passing to saddle-point preconditioners, if $\|P\|_2$ is small enough, the problem preserves the characteristics of a generalized saddle-point problem. The preconditioners analyzed, for instance, in [45] can then be of interest. Here, the $(1, 1)$ block of the coefficient matrix is first split in $A = F - E$. Then, considering the Jacobian (10), the saddle-point preconditioner P with exact Schur complement is

$$P = \begin{pmatrix} F^{-1} & 0 \\ 0 & -(P - RF^{-1}B)^{-1} \end{pmatrix}$$



(a) Plot of $\boldsymbol{p}$ for Problem 4 **(b)** Plot of $\boldsymbol{r}$ for Problem 4

Figure 3 Plots of the profiles of $\boldsymbol{p}$ and $\boldsymbol{r}$ of Problem 4 computed by ILU-preconditioned GMRES with $n = 100$ and adaptive linear tolerance with $tol_{\min} = 10^4$

where P, R are diagonal and B is lower bidiagonal. It is worth noticing that the Schur complement can be computed exactly when A is split by Jacobi or Gauss-Seidel splittings. Indeed, if F is the diagonal of A (Jacobi splitting), $P - RF^{-1}B$ is lower bidiagonal and its inverse is readily available. In case of Gauss-Seidel splitting, since A is tridiagonal, F is lower bidiagonal. Hence, $P - RF^{-1}B$ is lower triangular and can be inverted by forward substitution. Alternatively, preconditioners with approximate Schur complement [45] can be used as well.

7 Conclusions

We introduced a DIN algorithm for solving complementarity problems and we presented it with special regard to LCPs arising in hydrodynamic lubrication. We proved the global convergence of the proposed method and we demonstrated its applicability for solving the problems of our concern. In the numerical experiments we also analyzed the components of the algorithm, highlighting, for example, the advantage of choosing a larger perturbation parameter μ_k. We also compared the DIN and the FBN iteration, confirming the ability of the DIN method of strongly enforcing the non-negativity conditions of the complementarity problem, while the FBN also accepts negative values in the solution vectors (although within machine precision). Lastly, we remarked the importance of solving efficiently the inner linear systems in order to have a fast implementation of the method. In particular, we compared the effect of different linear solvers, all efficiently implemented through the Lapack and the PETSc packages. We then concluded our analysis by solving a 2D example.

References

[1] R. W. Cottle, G. B. Dantzig, Complementarity pivot theory of mathematical programming, Linear Algebra Appl. 1 (1968) 103–125.

[2] R. W. Cottle, J.-S. Pang, R. E. Stone, The Linear Complementarity Problem, Classics in Applied Mathematics, SIAM, 2009.

[3] G. Capriz, G. Cimatti, Free boundary problems in the theory of hydrodynamic lubrication: A survey, Tech. Rep. Nota Informatica C81-7, Istituto di Matematica, University of Pisa (1981).

[4] A. Laratta, O. Menchi, Approssimazione della soluzione di una disequazione variazionale. applicazione ad un problema di frontiera libera, Calcolo 11 (1974) 243–267.

[5] G. McAllister, S. Rohde, An optimization problem in hydrodynamic lubrication theory, Appl. Math. Opt. 2 (1976) 223–235.

[6] G. Cimatti, O. Menchi, On the numerical solution of a variational inequality connected with the hydrodynamica lubrication of a complete journal bearing, Calcolo 15 (1978) 249–258.

[7] C. Cryer, The numerical solution of a degenerate variational inequality, in: S. V. Parter (Ed.), Numerical Methods for Partial Differential Equations, 1979.

[8] C. Cryer, A. Dempster, Equivalence of linear complementarity problems and linear programs in vector lattice Hilbert spaces, SIAM J. Control Optim. 18 (1) (1980) 76–90.

[9] M. Kostreva, Elasto-hydrodynamic lubrication: a nonlinear complementarity problem, Int. J. Numer. Meth. Fl. 4 (1984) 377–397.

[10] L. Bertocchi, D. Dini, M. Giacopini, M. Fowell, A. Baldini, Fluid film lubrication in the presence of cavitation: a mass-conserving two-dimensional formulation for compressible, piezoviscous and non-Newtonian fluids, Tribol. Int. 67 (2013) 61–71.

[11] G. Chapiro, A. E. Gutierrez, J. Herskovits, S. R. Mazorche, W. S. Pereira, Numerical solution of a class of moving boundary problems with a nonlinear complementarity approach, J. Optimiz. Theory App. 168 (2) (2016) 534–550.

[12] M. Giacopini, M. Fowell, D. Dini, A. Strozzi, A mass-conserving complementarity formulation to study lubricant films in the presence of cavitation, J. Tribol. 132 (2010).

[13] A. Almqvist, P. Wall, Modelling cavitation in (elasto)hydrodynamic lubrication, Adv. Tribol. (2016).

[14] T. Woloszynski, P. Podsiadlo, G. W. Stachowiak, Efficient solution to the cavitation problem in hydrodynamic lubrication, Tribol. Lett. 58 (2015) 1–11.

[15] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[16] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[17] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory (2017).
URL `http://www.mcs.anl.gov/petsc`

[18] M. Khonsari, E. Booser, Applied tribology: bearing, design and lubrication, 2nd Edition, John Wiley & Sons, Chichester, 2008.

[19] R. Varga, Matrix Iterative Analysis, 2nd Edition, Springer, Berlin, 2000.

[20] E. Süli, D. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, Cambridge, 2003.

[21] A. El-Bakri, R. Tapia, T. Tsuchiya, Y. Zhang, On the formulation and theory of the Newton interior point method for nonlinear programming, J. Optimiz. Theory App. 89 (1996) 507–541.

[22] R. Dembo, S. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (1982) 400–408.

[23] S. Eisenstat, H. Walker, Globally convergent inexact Newton methods, SIAM J. Optimiz. 4 (1994) 393–422.

[24] D. Fokkema, G. Sleijpen, H. Van der Vost, Accelerated inexact Newton schemes for large systems of nonlinear equations, SIAM J. Sci. Comput. 19 (1998) 657–674.

[25] C. Durazzi, On the Newton interior-point method for nonlinear programming problems, J. Optimiz. Theory App. 104 (2000) 73–90.

[26] R. Horn, C. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[27] E. Galligani, Analysis of the convergence of an inexact Newton method for solving Karush-Kuhn-Tucker systems, in: Atti del Seminario Matematico e Fisico dell'Università di Modena e Reggio Emilia LII, 2004, pp. 331–368.

[28] D. S. Bernstein, Matrix Mathematics: theory, facts and formulas, 2nd Edition, Princeton University Press, Princeton, NJ (U.S.A.), 2009.

[29] W. Rheinboldt, Methods for Solving Systems of Nonlinear Equations, 2nd Edition, SIAM, Philadelphia, 1998.

[30] S. Bonettini, E. Galligani, V. Ruggiero, Inner solvers for interior point methods for large scale nonlinear programming, Comput. Optim. Appl. 37 (2007) 1–34.

[31] A. Fischer, A special Newton-type optimization method, Optimization 24 (1992) 269–284.

[32] L. Armijo, Minimization of functions having Lipschitz-continuous first partial derivatives, Pac. J. Math. 16 (1966) 1–3.

[33] J. Nocedal, S. Wright, Numerical Optimization, Springer-Verlag, New York, 1999.

[34] V. Lakshmikantham, D. Trigiante, Theory of Difference Equations: Numerical Methods and Applications, 2nd Edition, Marcel Dekker Inc., New York, 2002.

[35] K. Oh, P. Goenka, The elastohydrodynamic solution of journal bearings under dynamic loading, J. Tribol. 107 (3) (1985) 389–395.

[36] D. Bonneau, M. Hajjam, Modélisation de la rupture et de la reformation des films lubrifiants dans les contacts élastohydrodynamiques, Eur. J. Comput. Mech. 10 (2001) 679–704.

[37] C. Kelley, Iterative Methods for Linear and Nonlinear Equations, Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[38] M. Argaez, R. Tapia, L. Velazquez, Numerical comparisons of path-following strategies for a primal-dual interior-point method for nonlinear programming, J. Optimiz. Theory App. 114 (2002) 255–272.

[39] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. H. Liu, A supernodal approach to sparse partial pivoting, SIAM J. Matrix Anal. A. 20 (3) (1999) 720–755.

[40] X. Li, J. Demmel, J. Gilbert, L. Grigori, M. Shao, I. Yamazaki, SuperLU Users' Guide, Tech. Rep. LBNL-44289, Lawrence Berkeley National Laboratory, `http://crd.lbl.gov/ xiaoye/SuperLU/`. Last update: August 2011 (September 1999).

[41] P. R. Amestoy, I. S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix. Anal. A. 23 (1) (2001) 15–41.

[42] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Comput. 32 (2) (2006) 136–156.

[43] M. Benzi, A. J. Wathen, Some preconditioning techniques for saddle point problems, in: W. H. A. Schilders, H. A. van der Vorst, J. Rommes (Eds.), Model Order Reduction: Theory, Research Aspects and Applications, Vol. 13, Springer, Berlin, Heidelberg, 2008, pp. 195–211.

[44] B. Morini, V. Simoncini, M. Tani, A comparison of reduced and unreduced kkt systems arising from interior point methods, Comput. Optim. Appl. 68 (1) (2017) 1–27.

[45] C. Siefert, E. De Sturler, Preconditioners for generalized saddle-point problems, SIAM J. Numer. Anal. 44 (3) (2006) 1275–1296.