# Numerical optimization for rolling frictional contact problems

Minh Hoang Nguyen

UNIVERSITY OF LIMOGES
&
INRIA GRENOBLE RHÔNE-ALPES

**Intership report**
to obtain the degree of
MASTER ACSYON

---

# Numerical optimization for rolling frictional contact problems

---

Hoang Minh NGUYEN

**Supervisors:**
Prof. Vincent ACARY, Prof. Paul ARMAND

Limoges, France
September, 2021

# Acknowledgements

*To the indispensable people who helped me accomplish effectually this master thesis.*

First and foremost, I should like to express wholeheartedly my sincere gratitude and appreciation for Professor Paul ARMAND, Professor Vincent ACARY, my prodigious internship supervisors. Under their proper and seasoned direction, I have successfully passed the 6-month internship at research centers Inria and Xlim and accumulated loads of diverse knowledge in optimization and mechanics as well as in-depth research experience. First off, Professor Armand gave me an overview of applying optimization theory to a real rolling friction problem while also teaching me dominant techniques for solving the considerable difficulties that the problem poses. Second, Professor Acary encouraged me to use the powerful and widely used high-level programming language Python as well as giving me feedback on their appropriate use. Lastly, both professors have been closely monitoring my progress in finding solutions and always give me immediate consultation and guidance so that I can spot the inappropriate solutions and how to improve them.

Besides my tremendous mentors, it is a pleasure to show my profound gratitude to professors in Vietnam who have always supported and brought me to the master program ACSYON of the University of Limoges, as well as professors who have taught and inspired me to learn and research the applications of math during this program. These special thanks go to Professor Huynh Van Ngai, Professor Nguyen An Khuong, Professor Samir Adly, Professor Paul Armand, Professor Noureddine Igbida, Professor Moulay Barkatou, Professor Simone Naldi, Professor Loïc Bourdin, Professor Olivier Ruatta, Professor Olivier Prot, Professor Francisco José Silva Álvarez.

Moreover, I desire to dedicate this separate paragraph to express my sincere thanks to a respected teacher who taught me French in Vietnam. She is the one who not only gave me solid intellectual baggage on the way to this beautiful country to study and develop myself, but as well always deliver me motivation and share with me feelings during the journey of accumulating knowledge in France.

Last but not least, I must mention close people have always been by my side to constantly encourage and directly share with me all the difficulties, knowledge, spirit, enthusiam, energy and joy during my learning journey. They are my family, especially my mother, and friends who are dear and lovely Le Le, Viet Sang, Duc Anh, Tuna, Khawla and Jmili. It would be truly difficult to attain this accomplishment without them.

# Abstract

The scope of this report is to study the optimal solution of a model for the problem of unilateral contact between solid bodies with rolling friction, by means of an interior-point algorithm. The model is based on the second-order cone programming (also SOCP) but with a conical constraint given by the rolling friction cone that is not a standard cone in convex optimization as the Lorentz cone, for instance. The difficulties are that the KKT system related to this model is not square (the number of variables and equations are not equal) and the rolling friction cone is not symmetric. That means it is necessary to find out the way that transforms the KKT system into a square version, and as well as symmetrizes the conical constraint in order to be able to apply the solving techniques for the standard SOCP. Some prerequisites will be firstly known in annex, then one of the best possibility is proposed to solve the model in this report, along with difficulties and their solutions. In the last section, more than 200 experiments are introduced to apply this proposed approach and then present the results and observations of exceptional phenomena.

***Keywords :*** optimization, Jordan algebra, SOCP, rolling friction contact, rolling friction cone, Nesterov and Todd scaling.

# Contents


**Acknowledgements** **ii**

**Abstract** **iii**

**Abbreviations and Symbols** **1**

**Introduction** **3**

**1 Interior-point algorithm for the rolling friction problem** **5**

1.1 Conical constraint symmetrization . . . 5

1.1.1 Mathematical model to study . . . 5

1.1.2 Retrieve self-dual cones . . . 8

1.2 Formulation for KKT system . . . 10

1.2.1 Complementarity conditions . . . 10

1.2.2 KKT conditions . . . 10

1.2.3 Strict complementarity . . . 10

1.3 Perturbed formulation for KKT system . . . 11

1.3.1 Perturbed KKT system . . . 11

1.3.2 Linear system of the IP algorithm . . . 12

1.3.3 Step-length computation . . . 13

1.3.4 A dual infeasibility phenomenon . . . 13

1.3.5 Simple IP algorithm . . . 15

1.3.6 IP algorithm with Mehrotra scheme . . . 16

1.3.7 First simple example. . . . 17

1.4 Symmetrization of linearized system . . . 17

1.4.1 Nesterov-Todd scaling strategy . . . 17

1.4.2 Scaling technique algorithm for IP method . . . 20

1.4.3 An alternative scaling operator . . . 22

**2 Numerical experiments** **23**

2.1 Experiments . . . 23

2.1.1 Simple IP algorithm . . . 23

2.1.2 IP algorithm with predictor-corrector scheme . . . 25

2.1.3 IP algorithm with predictor-corrector scheme and NT scaling technique . 26

2.1.4 IP algorithm with predictor-corrector scheme and NT scaling technique using matrix F . . . 27

2.2 Performance profiles . . . 29

2.3 Jump phenomenon . . . 32

**Annex** **36**

# Abbreviations and Symbols

| | |
|---|---|
| $(x; y; z)$ | Column vector $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (x^\top, y^\top, z^\top)$ |
| IP | Interior-point |
| KKT | Karush–Kuhn–Tucker |
| NT | Nesterov-Todd |
| RHS | Right-hand-side |
| SC | Strict complementarity |
| SOCP | Second-order cone programming |

# Introduction

In [1], Acary and Bourrier formulate and analyse a model for the problem of unilateral contact between solid bodies with rolling friction. Accordingly, let us introduce the rolling friction cone, as the cone of admissible reaction forces and torques, which is denoted by

$$K = \left\{p = (r_{\text{N}}; r_{\text{T}}; m_{\text{R}}) \in \mathbb{R}^5 : \|r_{\text{T}}\| \leq \mu r_{\text{N}}, \|m_{\text{R}}\| \leq \mu_{\text{R}} r_{\text{N}}\right\}, \tag{1}$$

where $\mu > 0$ and $\mu_{\text{R}} > 0$ are the friction and rolling friction coefficients. The vector $r$ is the reaction force at contact, with a normal component $r_{\text{N}}$ and the tangential components represented by the vector $r_{\text{T}} \in \mathbb{R}^2$. The vector $m_{\text{R}} \in \mathbb{R}^2$ is the rolling friction reaction moment at contact. Then, the dual cone $K_r^* = \{y \in \mathbb{R}^5 : \forall p \in K_r, y^\top p \geq 0\}$ is given by [1, Lemma 3.1]

$$K^* = \left\{y = (u_{\text{N}}; u_{\text{T}}; \omega_{\text{R}}) \in \mathbb{R}^5 : \mu\|u_{\text{T}}\| + \mu_{\text{R}}\|\omega_{\text{R}}\| \leq u_{\text{N}}\right\}, \tag{2}$$

where $u = (u_{\text{N}}; u_{\text{T}})$ is the velocity vector of the contact point, with normal and tangential components, and $\omega_{\text{R}}$ is the relative angular velocity.

The general friction contact problem with rolling resistance [1] is formulated as[1]

$$\begin{aligned} Mv + f - H^\top r &= 0 \\ Hv + w &= y \\ K^* \ni y + \Phi(y) &\perp r \in K, \end{aligned}$$

where $M \in \mathbb{R}^{m\times m}$ is symmetric and positive-definite, $f \in \mathbb{R}^m$, $H \in \mathbb{R}^{d\times m}$, $w \in \mathbb{R}^d$ and

$$\Phi(y) = \begin{pmatrix} \mu\|u_{\text{T}}\| + \mu_{\text{R}}\|\omega_{\text{R}}\| \\ \mathbf{0}_{d-1} \end{pmatrix} \in \mathbb{R}^d.$$

Noting that $\Phi(\hat{y}) = \Phi(y + \Phi(y))$, the change of variable $\hat{y} = y + \Phi(y)$ allows to reformulate the problem as

$$\begin{aligned} Mv + f - H^\top r &= 0 \\ Hv + w + \Phi(\hat{y}) &= \hat{y} \\ K^* \ni \hat{y} &\perp r \in K. \end{aligned} \tag{3}$$

In [3], a sequential solution of this system is proposed as follows. The second equation is written as

$$Hv + w + \phi(s) = y \quad \text{and} \quad s = \mu\|u_{\text{T}}\| + \mu_{\text{R}}\|\omega_{\text{R}}\|$$

where $\phi(s) = (s, 0, 0, 0, 0)^\top$. The idea is to solve the nonlinear system as a parametric system of the form

$$\begin{aligned} Mv + f - H^\top r &= 0 \\ Hv + w + \phi(s) &= y \\ K^* \ni y &\perp r \in K, \end{aligned} \tag{4}$$

[1] To simplify, here we consider only one contact point.

where $s$ is updated, either at the end of each iteration or after a convergence criterion is satisfied, according to

$$s = \mu\|u_{\mathrm{T}}\| + \mu_{\mathrm{R}}\|\omega_{\mathrm{R}}\|.$$

The advantage of this formulation is that, for a fixed $s$, the parametric system is now the first order optimality conditions of a convex optimization problem, which can be transformed into the class of second order cone optimization problems. The rest of the report focuses on the numerical solution of the related optimization problem.

Let $n \in \mathbb{N}$ be the number of contact points, $d = 5$ be the size of the unknown reaction and torque vector at contact, and $m \in \mathbb{N}$ be the number of degrees of freedom. In the relaxed formulation of (4) where $s$ is fixed and where we set $w = w + \phi(s)$, the rolling friction contact problem leads to a convex optimization problems of the form (see [2])

$$\begin{array}{ll} \min\limits_{v,y} & \frac{1}{2}v^\top Mv + f^\top v \\ \text{s.t.} & Hv + w = y, \\ & y \in \mathcal{K}^*, \end{array} \tag{5}$$

where $M \in \mathbb{R}^{m\times m}$ is symmetric and positive-definite, $f \in \mathbb{R}^m$, $H \in \mathbb{R}^{nd\times m}$, $w \in \mathbb{R}^{nd}$ and $\mathcal{K} = \prod_{i=1}^n K_i$, each cone $K_i$ is of the form (1). We then have $\mathcal{K}^* = \prod_{i=1}^n K_i^*$. The dual problem of (5) is then

$$\begin{array}{ll} \min\limits_{v,p} & \frac{1}{2}v^\top Mv - w^\top p \\ \text{s.t.} & H^\top p - Mv = f, \\ & p \in \mathcal{K}. \end{array} \tag{6}$$

The goal of this internship is to compute numerical solutions $(v, y, p) \in \mathbb{R}^m \times \mathbb{R}^{nd} \times \mathbb{R}^{nd}$ of optimization problems related to the primal-dual model (5)-(6), by means of an interior-point algorithm.

The original interior-point method has been proposed by Karmarkar [9] for solving linear programming. Then, Nesterov and Nemirovski [12], Nesterov and Todd [11] generalized efficient primal-dual interior-point method for convex conical programming problems, only for self-scaled cones [14, §3.5.1] (or symmetric cones. Unfortunately, the rolling friction cone $K$ (1) is not self-dual, i.e., $K \neq K^*$, so $K$ is also not symmetric. Therefore, the main motivation for the intership is to show how to symmetrize the rolling friction cone $K$ in order to apply efficiently interior-point method by using a Euclidean Jordan algebra.

In this report, we present in Chapter 1 the mathematical model for rolling friction problem as well as the strategy by which we transform the rolling friction cone into product of two Lorentz cones, which are symmetric cones [7, §2], by means of a change of variables. We also show that interior-point method efficiently works for a such strategy. At the same time, we describe a special phenomenon concerning dual feasibility in solving the problem. In Chapter 2, many numerical experiments are applied to our approach in order to demonstrate its efficiency and robustness by several different schemes for interior-point method.

# Chapter 1

# Interior-point algorithm for the rolling friction problem

In this chapter, we will present how to apply interior-point method for the rolling friction problem which is more complicated than the standard SOCP because of its conical constraint that involves non symmetric cones. As shown in introduction, the conical constraint of primal-dual model (5)-(6) is asymmetric. Thus, the first thing to do is to symmetrize these cones in order to be able to introduce an appropriate formulation for the rolling friction problem that enables the application of interior-point method.

## 1.1 Conical constraint symmetrization

### 1.1.1 Mathematical model to study

Let us denote by $\mu_i$ and $\mu_{\mathrm{R},i}$ for $i \in \{1,\dots,n\}$ the friction coefficients related to a cone $K_i$. Let $i \in \{1,\dots,n\}$, to eliminate these coefficients from (6), a change of variables is done:

$$u_{i,0} = u_{i,\mathrm{N}}, \quad \bar{u}_i = \mu_i u_{i,\mathrm{T}} \quad \text{and} \quad \tilde{u}_i = \mu_{\mathrm{R},i} w_{i,\mathrm{R}}.$$

The matrix $H$ and the vector $w$ are left-multiplied by a block-diagonal matrix $D$ with $n$ blocks of the form

$$\mathrm{diag}(1, \mu_i, \mu_i, \mu_{\mathrm{R},i}, \mu_{\mathrm{R},i}).$$

We then have $u = Dy$ and the linear constraint of (5) becomes $DHv + Dw = u$. To simplify the notation, we denote again by $H$ and $w$ the new matrix $DH$ and the new vector $Dw$. In the same way, we set $r = D^{-1}p$, so that

$$r_{i,0} = r_{i,\mathrm{N}}, \quad \bar{r}_i = \frac{1}{\mu_i} r_{i,\mathrm{T}} \quad \text{and} \quad \tilde{r}_i = \frac{1}{\mu_{i,\mathrm{R}}} m_{i,\mathrm{R}}.$$

Let us now define the rolling friction cone

$$\mathcal{F}_i = \{r_i = (r_{i,0}; \bar{r}_i; \tilde{r}_i) \in \mathbb{R}^d : \max\{\|\bar{r}_i\|, \|\tilde{r}_i\|\} \le r_{i,0}\}.$$

**Lemma 1.1.1.** The dual cone of $\mathcal{F}_i$ is given by

$$\mathcal{F}_i^* = \{u_i = (u_{i,0}; \bar{u}_i; \tilde{u}_i) \in \mathbb{R}^d : \|\bar{u}_i\| + \|\tilde{u}_i\| \le u_{i,0}\}.$$

*Proof.* To simplify, we will give a dual cone for only one contact point, i.e., $n = 1$. In this case, the rolling friction cone and its dual are

$$\begin{aligned} \mathcal{F} &= \{r = (r_0; \bar{r}; \tilde{r}) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 : \max\{\|\bar{r}\|, \|\tilde{r}\|\} \le r_0\}, \\ \mathcal{F}^* &= \{u = (u_0; \bar{u}; \tilde{u}) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 : \|\bar{u}\| + \|\tilde{u}\| \le u_0\}. \end{aligned}$$

Note that the dual cone $\mathcal{F}^* = \{u \in \mathbb{R}^d : u^\top r \geq 0 \text{ for all } r \in \mathcal{F}\}$. We now let $u = (u_0; \bar{u}; \tilde{u}) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2$ such that $u_0 \geq \|\bar{u}\| + \|\tilde{u}\|$. For all $r \in \mathcal{F}$, we get

$$u^\top r = u_0 r_0 + \bar{u}^\top \bar{r} + \tilde{u}^\top \tilde{r} \geq u_0 r_0 - \|\bar{u}\|\|\bar{r}\| - \|\tilde{u}\|\|\tilde{r}\|$$

by Cauchy-Schwarz inequalities $\bar{u}^\top \bar{r} \geq -\|\bar{u}\|\|\bar{r}\|$ and $\tilde{u}^\top \tilde{r} \geq -\|\tilde{u}\|\|\tilde{r}\|$. Since $r \in \mathcal{F}$, we then obtain

$$u^\top r \geq r_0 u_0 - r_0\|\bar{u}\| - r_0\|\tilde{u}\| = r_0(u_0 - \|\bar{u}\| - \|\tilde{u}\|) \geq 0$$

which implies $u \in \mathcal{F}^*$. Conversely, let $u \in \mathcal{F}^* = \{u \in \mathbb{R}^d : u^\top r \geq 0 \text{ for all } r \in \mathcal{F}\}$ and let $\mathbf{0} = (0; 0) \in \mathbb{R}^2$ be the zero vector. We now define $r = (1; \bar{r}; \tilde{r})$ such that

$$\bar{r} = \begin{cases} -\frac{\bar{u}}{\|\bar{u}\|} & \text{if } \bar{u} \neq \mathbf{0} \\ \mathbf{0} & \text{if } \bar{u} = \mathbf{0} \end{cases} \quad \text{and} \quad \tilde{r} = \begin{cases} -\frac{\tilde{u}}{\|\tilde{u}\|} & \text{if } \tilde{u} \neq \mathbf{0} \\ \mathbf{0} & \text{if } \tilde{u} = \mathbf{0} \end{cases}.$$

We acknowledge that $r \in \mathcal{F}$. Four cases are possible:

- $\bar{u} = \tilde{u} = \mathbf{0}$ then $r = (1; \mathbf{0}; \mathbf{0}) \in \mathbb{R}^5$. Since $u^\top r \geq 0$, we obtain $u_0 \geq 0 = \|\bar{u}\| + \|\tilde{u}\|$.
- $\bar{u} = \mathbf{0}, \tilde{u} \neq \mathbf{0}$ then $r = \left(1; \mathbf{0}; -\frac{\tilde{u}}{\|\tilde{u}\|}\right) \in \mathbb{R}^5$. We have $0 \leq u^\top r = u_0 - \|\tilde{u}\|$ then $u_0 \geq \|\tilde{u}\| = \|\bar{u}\| + \|\tilde{u}\|$.
- $\bar{u} \neq \mathbf{0}, \tilde{u} = \mathbf{0}$ then $r = \left(1; -\frac{\bar{u}}{\|\bar{u}\|}; \mathbf{0}\right) \in \mathbb{R}^5$. We have $0 \leq u^\top r = u_0 - \|\bar{u}\|$ then $u_0 \geq \|\bar{u}\| = \|\bar{u}\| + \|\tilde{u}\|$.
- $\bar{u} \neq \mathbf{0}, \tilde{u} \neq \mathbf{0}$ then $r = \left(1; -\frac{\bar{u}}{\|\bar{u}\|}; -\frac{\tilde{u}}{\|\tilde{u}\|}\right)$. We have $0 \leq u^\top r = u_0 - \|\bar{u}\| - \|\tilde{u}\|$ then $u_0 \geq \|\bar{u}\| + \|\tilde{u}\|$.

Therefore, $u$ always satisfies $u_0 \geq \|\bar{u}\| + \|\tilde{u}\|$ for all $u \in \mathcal{F}^*$. The proof is done. □

We have

$$\mathcal{F} = \prod_{i=0}^{n} \mathcal{F}_i \quad \text{and} \quad \mathcal{F}^* = \prod_{i=0}^{n} \mathcal{F^*}_i.$$

The primal-dual pair (5)-(6) becomes

$$\begin{aligned} \min_{v,u} \quad & \tfrac{1}{2} v^\top M v + f^\top v \\ \text{s.t.} \quad & Hv + w = u, \\ & u \in \mathcal{F}^*, \end{aligned} \tag{1.1}$$

and

$$\begin{aligned} \min_{v,r} \quad & \tfrac{1}{2} v^\top M v + w^\top r \\ \text{s.t.} \quad & H^\top r - Mv = f, \\ & r \in \mathcal{F}. \end{aligned} \tag{1.2}$$

where

- $v \in \mathbb{R}^m$,
- $u = (u_1; \ldots; u_n) \in \mathbb{R}^{nd}$ and $u_i = (u_{i,0}; \bar{u}_i; \tilde{u}_i) \in \mathbb{R}^d$ for $i \in \{1, \ldots, n\}$ are the primal variables,
- $r = (r_1; \ldots; r_n) \in \mathbb{R}^{nd}$ and $r_i = (r_{i,0}; \bar{r}_i; \tilde{r}_i) \in \mathbb{R}^d$ for $i \in \{1, \ldots, n\}$ are the dual variables.

**Proposition 1.1.2.** The optimization problems (1.1)-(1.2) are the primal-dual pair of the problem.

*Proof.* The Lagrangian function associated to (1.1) is

$$\begin{array}{rcl} L(v,u,r) & = & \frac{1}{2}v^\top Mv + f^\top v + r^\top(u - Hv - w) \\ & = & \frac{1}{2}v^\top Mv + (f - H^\top r)^\top v + r^\top u - w^\top r, \end{array}$$

for $(v,u,r) \in \mathbb{R}^m \times \mathbb{R}^{nd} \times \mathbb{R}^{nd}$. The dual function is then

$$d(r) = \inf_{(v,u)\in\mathbb{R}^m\times\mathcal{F}^*} L(v,u,r). \tag{1.3}$$

The Lagrangian is separable in $v$ and $u$, so we have

$$\inf_{(v,u)\in\mathbb{R}^m\times\mathcal{F}^*} L(v,u,r) = \inf_{v\in\mathbb{R}^m} L_1(v,r) + \inf_{u\in\mathcal{F}^*} r^\top u$$

where

$$L_1(v,r) = \frac{1}{2}v^\top Mv + (f - H^\top r)^\top v.$$

Since $L_1(v,r)$ is convex in $v$, its infimum is then achieved when $\nabla_v L_1(v,r) = 0$, i.e.,

$$Mv + f - H^\top r = 0.$$

Besides, by the definition of the dual cone, we have

$$\inf_{u\in\mathcal{F}^*} r^\top u = \begin{cases} 0 & \text{if } r \in \mathcal{F}^{**} = \mathcal{F} \\ -\infty & \text{otherwise} \end{cases}.$$

Hence, (1.3) becomes

$$d(r) = \begin{cases} -\frac{1}{2}v^\top Mv - w^\top r & \text{if } Mv + f - H^\top r = 0 \quad \text{and} \quad r \in \mathcal{F} \\ -\infty & \text{otherwise} \end{cases}.$$

The dual problem follows. □

**Remark 1.1.3.** Since $M$ is positive definite, by Proposition (1.1.2), we can formulate the reduced form of the dual problem (1.2) as

$$\begin{array}{ll} \min & \frac{1}{2}r^\top HM^{-1}H^\top r - (HM^{-1}f - w)^\top r, \\ \text{s.t.} & r \in \mathcal{F}. \end{array} \tag{1.4}$$

Note that a vector $r \in \mathbb{R}^{nd}$ is an optimal solution if and only if

$$\mathcal{F}^* \ni HM^{-1}H^\top r - HM^{-1}f + w \perp r \in \mathcal{F}.$$

However, we do not solve directly the reduced problem despite its apparent simplification. This problem is not strictly convex because of the rank-deficiency of the matrix $H$. It means that if $\text{rank}(H) = nd$ then $HM^{-1}H^\top \succ 0$; and if $\text{rank}(H) < nd$ then $HM^{-1}H^\top$ is singular. Thus, the solution is not unique. The Jacobian matrix of the KKT system related to (1.4), in this case, becomes singular and then causes numerical difficulties. Moreover, it is possible to lose the structure of the problem since if $H$ is a sparse matrix, which is usually the case, $HM^{-1}H^\top$ is no more sparse.

**Assumption 1.1.4.** *(Slater's condition)* There exists a vector $v \in \mathbb{R}^m$ satisfying $Hv + w \in \text{int}(\mathcal{F}^*)$.

**Theorem 1.1.5.** (i) Let $(v,u,r) \in \mathbb{R}^{m+2nd}$ be a solution of (1.5). Then $(v,u)$ is an optimal solution of (1.1) and $(v,r)$ is an optimal solution of (1.2).

(ii) Conversely, if the problem (1.1) is feasible, then it has a unique optimal solution $(v, u) \in \mathbb{R}^{m+nd}$. In addition, if (1.1) is strictly feasible (i.e., the Slater's conditions are satisfied), then there exists $r \in \mathbb{R}^{nd}$ such that $(v, u, r)$ is a solution of (1.5). The primal-dual solution of (1.1) is characterized by

$$\begin{array}{lcl} Hv + w - u & = & 0, \\ Mv + f - H^\top r & = & 0, \\ u^\top r & = & 0, \\ (u, r) \in \mathcal{F}^* \times \mathcal{F}. & & \end{array} \tag{1.5}$$

Before going into the proof, we need the following lemma.

**Lemma 1.1.6.** *(Weak duality)*
Let $(v, u, r) \in \mathbb{R}^m \times \mathcal{F}^* \times \mathcal{F}$ be a feasible primal-dual triplet for the problems (1.1)-(1.2), i.e., $Hv + w = u$ and $Mv + f - H^\top r = 0$, then

$$d(v, r) := -\frac{1}{2} v^\top M v - w^\top r \leq \frac{1}{2} v^\top M v + f^\top v =: p(v).$$

*Proof.* Let $(v, u, r) \in \mathbb{R}^m \times \mathcal{F}^* \times \mathcal{F}$, we have

$$0 \leq u^\top r = (Hv + w)^\top r = v^\top (Mv + f) + w^\top r = v^\top M v + f^\top v + w^\top r,$$

then $-w^\top r \leq v^\top M v + f^\top v$. By this inequality, we obtain $d(v, r) \leq p(v)$. □

We now go to the proof for Theorem (1.1.5). An elegant proof using arguments from convex analysis can be found in [3]. We give here a proof using the classical weak and strong duality properties of a convex optimization problem.

*Proof.* The statement $(i)$ is a directy consequence of weak duality. To prove $(ii)$, suppose that (1.1) is feasible. Because the objective function of (1.1) is strongly convex, and thus coercive, there exists an optimal solution and it is unique. If we assume that the Slater's conditions are satisfied, then from the strong duality Theorem of convex programming [8, Theorem 11.15] (see also [8, Theorem 11.23]) there exists $r \in \mathbb{R}^{nd}$ such that $p(v) = d(v, r)$[1]. Since $u^\top r = p(v) - d(v, r)$, the KKT conditions (1.5) are satisfied. □

### 1.1.2 Retrieve self-dual cones

One difficulty with the formulation (1.1) and (1.2) is that the cone $\mathcal{F}$ is not self-dual and thus not symmetric. Therefore, it seems to be difficult to find out a Jordan product such that the complementarity condition of (1.5) can be formulated as $u \circ r = 0$. Let us see now how to reformulate these problems by means of Lorentz cones which are self-dual. We denote the Lorentz cone in $\mathbb{R}^3$ by

$$\mathcal{L} = \{x = (x_0; \bar{x}) \in \mathbb{R} \times \mathbb{R}^2 : \|\bar{x}\| \leq x_0\}.$$

The idea being to reformulate the inequality $u_0 \geq \|\bar{u}\| + \|\tilde{u}\|$ (which belongs to $\mathcal{F}_i^*$) into two equivalent inequalities presents in the following remark.

**Remark 1.1.7.** For a triplet $(a, b, c)$ of real numbers, we have $a \geq b + c$ if and only if there exists $t, t' \in \mathbb{R}$ such that $t \geq b, t' \geq c$ and $a = t + t'$.

By letting $u_{i,0} = t_i + t_i'$ for all $i \in \{1, \ldots, n\}$, the problem (1.1) can be reformulated as

$$(P_{t,t'}) \qquad \begin{array}{ll} \min\limits_{v, u_{t,t'}} & \frac{1}{2} v^\top M v + f^\top v \\ \text{s.t.} & Hv + w = J u_{t,t'}, \\ & ((t_i; \bar{u}_i), (t_i'; \tilde{u}_i)) \in \mathcal{L} \times \mathcal{L}, i \in \{1, \ldots, n\}, \end{array} \tag{1.6}$$

where

[1] $p(v) =$ optimal value of (1.1) and $d(v, r) =$ optimal value of (1.2).

- $J = \begin{pmatrix} j & & \\ & \ddots & \\ & & j \end{pmatrix} \in \mathbb{R}^{5n\times 6n}$ and $j = \begin{pmatrix} 1 & & 1 & \\ & I_2 & & \\ & & & I_2 \end{pmatrix} \in \mathbb{R}^{5\times 6}$,
- $u_{t,t'} = (u_{1,t,t'};\dots;u_{n,t,t'}) \in \mathbb{R}^{n(d+1)}$ and $u_{i,t,t'} = (t_i;\bar{u}_i,t'_i;\tilde{u}_i) \in \mathbb{R}^{d+1}$ for all $i \in \{1,\dots,n\}$.

So, we can see that an explicit formula of $Ju_{t,t'}$ given by

$$Ju_{t,t'} = (ju_{1,t,t'};\dots;ju_{n,t,t'}) \quad \text{and} \quad ju_{i,t,t'} = \begin{pmatrix} t_i + t'_i \\ \bar{u}_i \\ \tilde{u}_i \end{pmatrix}, i \in \{1,\dots,n\}.$$

We have that $(v,u)$ is an optimal solution of (1.1) if and only if there exists vectors $t,t' \in \mathbb{R}^n$ such that $(v,u_{t,t'})$ is an optimal solution of (1.6).

**Proposition 1.1.8.** The dual of (1.6) is as follows

$$(D_{t,t'}) \qquad \begin{array}{ll} \min\limits_{v,r} & \frac{1}{2}v^\top Mv + w^\top r \\ \text{s.t.} & H^\top r - Mv = f, \\ & ((r_{i,0};\bar{r}_i),(r_{i,0};\tilde{r}_i)) \in \mathcal{L}\times\mathcal{L}, i \in \{1,\dots,n\}. \end{array} \tag{1.7}$$

*Proof.* Let $i \in \{1,\dots,n\}$. The Lagrangian function associated to (1.6) is

$$\begin{array}{rcl} L(v,u_{t,t'},r) & = & \frac{1}{2}v^\top Mv + f^\top v + r^\top(Ju_{t,t'} - Hv - w) \\ & = & \frac{1}{2}v^\top Mv + (f - H^\top r)^\top v + r^\top Ju_{t,t'} - w^\top r, \end{array} \tag{1.8}$$

for $(v,u_{t,t'},r) \in \mathbb{R}^m \times \mathbb{R}^{n(d+1)} \times \mathbb{R}^{nd}$. The dual function is then

$$d(r) = \inf_{\substack{v\in\mathbb{R}^m,\\ ((t_i;\bar{u}_i),(t'_i;\tilde{u}_i))\in\mathcal{L}\times\mathcal{L}}} L(v,u_{t,t'},r). \tag{1.9}$$

Since

$$r^\top Ju_{t,t'} = (r_0^\top t + \bar{r}^\top\bar{u}) + (r_0^\top t' + \tilde{r}^\top\tilde{u}) = (r_0;\bar{r})^\top(t;\bar{u}) + (r_0;\tilde{r})^\top(t';\tilde{u}),$$

where $r_0 = (r_{1,0};\dots;r_{n,0})$, $t = (t_1;\dots;t_n)$, $t' = (t'_1;\dots;t'_n)$, $\bar{r} = (\bar{r}_1;\dots;\bar{r}_n)$ and $\tilde{r} = (\tilde{r}_1;\dots;\tilde{r}_n)$, the Lagrangian is separable in $v$, $(t;\bar{u})$ and $(t;\tilde{u})$. So, we have

$$\inf_{(v,(t_i;\bar{u}_i),(t'_i;\tilde{u}_i))\in\mathbb{R}^m\times\mathcal{L}\times\mathcal{L}} L(v,u_{t,t'},r) = \inf_{v\in\mathbb{R}^m} L_1(v,r) + \inf_{(t_i;\bar{u}_i)\in\mathcal{L}} (r_0;\bar{r})^\top(t;\bar{u}) + \inf_{(t'_i;\tilde{u}_i)\in\mathcal{L}} (r_0;\tilde{r})^\top(t';\tilde{u})$$

where

$$L_1(v,r) = \frac{1}{2}v^\top Mv + (f - H^\top r)^\top v.$$

Since $L_1$ is convex in $v$, its infimum is then achieved when $\nabla_v L_1(v,r) = 0$, i.e.,

$$Mv + f - H^\top r = 0.$$

By the definition of the dual cone, we have

$$\inf_{(t_i;\bar{u}_i)\in\mathcal{L}} (r_0;\bar{r})^\top(t;\bar{u}) = \begin{cases} 0 & \text{if } (r_{i,0};\bar{r}_i) \in \mathcal{L} \\ -\infty & \text{otherwise} \end{cases},$$

and

$$\inf_{(t'_i;\tilde{u}_i)\in\mathcal{L}} (r_0;\tilde{r})^\top(t';\tilde{u}) = \begin{cases} 0 & \text{if } (r_{i,0};\tilde{r}_i) \in \mathcal{L} \\ -\infty & \text{otherwise} \end{cases}.$$

Hence, we obtain

$$d(r) = \begin{cases} -\frac{1}{2}v^\top Mv - w^\top r & \text{if } Mv + f - H^\top r = 0 \quad \text{and} \quad ((r_{i,0};\bar{r}_i),(r_{i,0};\tilde{r}_i)) \in \mathcal{L}\times\mathcal{L} \\ -\infty & \text{otherwise} \end{cases}. \tag{1.10}$$

The dual problem follows. ☐

## 1.2 Formulation for KKT system

### 1.2.1 Complementarity conditions

The following proposition will be useful to state the KKT conditions, see [4, Lemma 15].

**Proposition 1.2.1.** Assume that $u \in \mathcal{F}^*$ and $r \in \mathcal{F}$. The following assertions are equivalent:

(i) $u^\top r = 0$.

(ii) There exists $t_i, t_i' \in \mathbb{R}$ such that for $i \in \{1, \dots, n\}$,

$$(t_i; \bar{u}_i) \circ (r_{i,0}; \bar{r}_i) = 0 \quad \text{and} \quad (t_i'; \tilde{u}_i) \circ (r_{i,0}; \tilde{r}_i) = 0. \tag{1.11}$$

*Proof.* We let $u = (u_1; \dots; u_n) \in \mathbb{R}^{nd}, r = (r_1; \dots; r_n) \in \mathbb{R}^{nd}$ such that $u_i = (u_{i,0}; \bar{u}_i; \tilde{u}_i) \in \mathcal{F}_i^*$ and $r_i = (r_{i,0}; \bar{r}_i; \tilde{r}_i) \in \mathcal{F}_i$. There exists $t_i, t_i' \in \mathbb{R}$ such that $u_{i,0} = t_i + t_i'$ and $(t_i; \bar{u}_i)$, $(r_{i,0}; \bar{r}_i)$, $(t_i'; \tilde{u}_i)$, $(r_{i,0}; \tilde{r}_i)$ are in $\mathcal{L}$ for all $i \in \{1, \dots, n\}$. In particular, because the Lorentz cone is self dual, the scalar product of any two pairs of these vectors is nonnegative, i.e.,

$$(t_i; \bar{u}_i)^\top (r_{i,0}; \bar{r}_i) \geq 0 \quad \text{and} \quad (t_i'; \tilde{u}_i)^\top (r_{i,0}; \tilde{r}_i) \geq 0.$$

Therefore, for all $i \in \{1, \dots, n\}$, we have

$$\begin{array}{rcl} u_i^\top r_i & = & (u_{i,0}; \bar{u}_i; \tilde{u}_i)^\top (r_{i,0}; \bar{r}_i; \tilde{r}_i) \\ & = & (t_i + t_i'; \bar{u}_i; \tilde{u}_i)^\top (r_{i,0}; \bar{r}_i; \tilde{r}_i) \\ & = & (t_i; \bar{u}_i)^\top (r_{i,0}; \bar{r}_i) + (t_i'; \tilde{u}_i)^\top (r_{i,0}; \tilde{r}_i) \geq 0. \end{array} \tag{1.12}$$

It follows that $u^\top r = \sum_{i=1}^n u_i^\top r_i = 0$ if and only if $u_i^\top r_i = 0$ for all $i \in \{1, \dots, n\}$. By virtue of [4, Lemma 15] and (1.12), $u_i^\top r_i = 0$ if and only if (1.11) holds. $\square$

### 1.2.2 KKT conditions

We can now formulate the first order optimality conditions of the primal-dual pair of the problem (1.6) by using Theorem (1.1.5). Assume that (1.6) is strictly feasible solutions, then $(v, u_{t,t'}, r) \in \mathbb{R}^m \times \mathbb{R}^{n(d+1)} \times \mathbb{R}^{nd}$ is a primal-dual optimal solution for this problem if and only if

$$\begin{array}{rcll} Mv + f - H^\top r & = & 0, & \\ Ju_{t,t'} - Hv - w & = & 0, & \\ (t_i; \bar{u}_i) \circ (r_{i,0}; \bar{r}_i) & = & 0, & i \in \{1, \dots n\}, \\ (t_i'; \tilde{u}_i) \circ (r_{i,0}; \tilde{r}_i) & = & 0, & i \in \{1, \dots n\}, \\ t_i & \geq & \|\bar{u}_i\|, & i \in \{1, \dots n\}, \\ t_i' & \geq & \|\tilde{u}_i\|, & i \in \{1, \dots n\}, \\ r_{i,0} & \geq & \|\bar{r}_i\|, & i \in \{1, \dots n\}, \\ r_{i,0} & \geq & \|\tilde{r}_i\|, & i \in \{1, \dots n\}. \end{array} \tag{1.13}$$

### 1.2.3 Strict complementarity

Let us define strict complementarity (SC) for the primal-dual formulation:

$$\begin{array}{rcl} Mv + f - H^\top r & = & 0 \\ Hv + w & = & Ju_{t,t'} \\ (t; \bar{u}) \circ (r_0; \bar{r}) & = & 0 \\ (t'; \tilde{u}) \circ (r_0; \tilde{r}) & = & 0, \\ (t; \bar{u}), (t'; \tilde{u}), (r_0; \bar{r}), (r_0; \tilde{r}) & \in & \mathcal{L}, \end{array}$$

where $\mathcal{L}$ is the Lorentz cone. To simplify, we consider only one contact point $n = 1$, the generalization to several contact points is straightforward. Following [4, Definition 23], a solution satisfies SC if and only if

$$(t;\bar{u}) + (r_0;\bar{r}) \in \mathrm{int}\mathcal{L} \quad \text{and} \quad (t';\tilde{u}) + (r_0;\tilde{r}) \in \mathrm{int}\mathcal{L}.$$

For each pair, there are three possibilities:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (1) | $t > \|\bar{u}\|$ | and | $(r_0;\bar{r}) = \mathbf{0}_{\mathbb{R}^3}$, | (a) | $t' > \|\tilde{u}\|$ | and | $(r_0;\tilde{r}) = \mathbf{0}_{\mathbb{R}^3}$ |
| (2) | $(t;\bar{u}) = \mathbf{0}_{\mathbb{R}^3}$ | and | $r_0 > \|\bar{r}\|$, | (b) | $(t';\tilde{u}) = \mathbf{0}_{\mathbb{R}^3}$ | and | $r_0 > \|\tilde{r}\|$ |
| (3) | $t = \|\bar{u}\|$ | and | $r_0 = \|\bar{r}\|$, | (c) | $t' = \|\tilde{u}\|$ | and | $r_0 = \|\tilde{r}\|$ |

Table 1.1: This must be read as ((1) or (2) or (3)) and ((a) or (b) or (c)).

Some possibilities are incompatible. Using the fact that $u_0 = t + t'$, this leads to five possibilities, which we list relatively to the original cones:

- $u = 0$ and $r_0 > \max\{\|\bar{r}\|, \|\tilde{r}\|\}$.
- $u_0 > \|\bar{u}\| + \|\tilde{u}\|$ and $r = 0$.
- $u_0 = \|\bar{u}\|$, $\tilde{u} = 0$ and $r_0 = \|\bar{r}\|$, $r_0 > \|\tilde{r}\|$.
- $u_0 = \|\tilde{u}\|$, $\bar{u} = 0$ and $r_0 = \|\tilde{r}\|$, $r_0 > \|\bar{r}\|$.
- $u_0 = \|\bar{u}\| + \|\tilde{u}\|$, $\bar{u} \neq 0$, $\tilde{u} \neq 0$ and $r_0 = \|\bar{r}\| = \|\tilde{r}\|$.

## 1.3 Perturbed formulation for KKT system

### 1.3.1 Perturbed KKT system

The system (1.13) is solved by means of an interior-point algorithm. To do that, a perturbation of the complementarity equations is introduced. Let us consider the barrier problem associated to (1.6):

$$(P_{t,t',\mu}) \qquad \begin{array}{ll} \min\limits_{v,u_{t,t'}} & \frac{1}{2}v^\top Mv + f^\top v - \mu\sum_{i=1}^n \ln\det(t_i;\bar{u}_i) - \mu\sum_{i=1}^n \ln\det(t'_i;\tilde{u}_i) \\ \text{s.t.} & Hv + w = Ju_{t,t'}, \end{array} \tag{1.14}$$

where $\mu > 0$ is called the barrier parameter and $\det(\cdot)$ is defined in Euclidean Jordan algebra, see Annex.

**Lemma 1.3.1.** Let $e = (1,0,0)^\top \in \mathbb{R}^3$ be the unit vector related to the Jordan product. The pertubed KKT optimality conditions for (1.14) are:

$$\begin{array}{rcl} Mv + f - H^\top r &=& 0, \\ Ju_{t,t'} - Hv - w &=& 0, \\ (t_i;\bar{u}_i)\circ(r_{i,0};\bar{r}_i) &=& 2\mu e, \quad i \in \{1,\dots n\}, \\ (t'_i;\tilde{u}_i)\circ(r_{i,0};\tilde{r}_i) &=& 2\mu e, \quad i \in \{1,\dots n\}. \end{array} \tag{1.15}$$

*Proof.* The Lagrangian function associated to (1.14) is

$$\begin{array}{l} L(v,u_{t,t'},r) \\ = \quad \dfrac{1}{2}v^\top Mv + f^\top v - \mu\displaystyle\sum_{i=1}^n \ln\det(t_i;\bar{u}_i) - \mu\sum_{i=1}^n \ln\det(t'_i;\tilde{u}_i) + r^\top(Ju_{t,t'} - Hv - w), \end{array}$$

for $(v, u_{t,t'}, r) \in \mathbb{R}^m \times \mathbb{R}^{n(d+1)} \times \mathbb{R}^{nd}$. The KKT optimality conditions are

$$\nabla L(v, u_{t,t'}, r) = 0, \tag{1.16}$$

where

- $\nabla_v L(v, u_{t,t'}, r) = Mv + f - H^\top r$,
- $\nabla_{u_{t,t'}} L(v, u_{t,t'}, r) = J^\top r - 2\mu u_{t,t'}^{-1}$,
  where $u_{t,t'}^{-1} = \begin{pmatrix} (t_1; \bar{u}_1)^{-1} \\ (t'_1; \tilde{u}_1)^{-1} \\ \vdots \\ (t_n; \bar{u}_n)^{-1} \\ (t'_n; \tilde{u}_n)^{-1} \end{pmatrix}$ and $(t_i; \bar{u}_i)^{-1}$ (also $(t'_i; \tilde{u}_i)^{-1}$) for $i \in \{1, \ldots, n\}$ are defined in Euclidean Jordan algebra, see Annex,
- $\nabla_r L(v, u_{t,t'}, r) = Ju_{t,t'} - Hv - w$.

From the condition $\nabla_{u_{t,t'}} L(v, u_{t,t'}, r) = 0$ in (1.16), we have

$$\begin{pmatrix} r_{1,0} \\ \bar{r}_1 \\ r_{1,0} \\ \tilde{r}_1 \\ \vdots \\ r_{n,0} \\ \bar{r}_n \\ r_{n,0} \\ \tilde{r}_n \end{pmatrix} - 2\mu \begin{pmatrix} \begin{pmatrix} t_1 \\ \bar{u}_1 \end{pmatrix}^{-1} \\ \begin{pmatrix} t'_1 \\ \tilde{u}_1 \end{pmatrix}^{-1} \\ \vdots \\ \begin{pmatrix} t_n \\ \bar{u}_n \end{pmatrix}^{-1} \\ \begin{pmatrix} t'_n \\ \tilde{u}_n \end{pmatrix}^{-1} \end{pmatrix} = 0,$$

then

$$\begin{array}{l} (r_{i,0}; \bar{r}_i) = 2\mu(t_i; \bar{u}_i)^{-1}, \quad i \in \{1, \ldots n\}, \\ (r_{i,0}; \tilde{r}_i) = 2\mu(t'_i; \tilde{u}_i)^{-1}, \quad i \in \{1, \ldots n\}. \end{array} \tag{1.17}$$

We multiply accordingly from the right of (1.17) by $(t_i; \bar{u}_i)^{-1}, (t'_i; \tilde{u}_i)^{-1}$. Using the Jordan product, we obtain

$$\begin{array}{rcl} (t_i; \bar{u}_i) \circ (r_{i,0}; \bar{r}_i) &=& 2\mu e, \quad i \in \{1, \ldots n\}, \\ (t'_i; \tilde{u}_i) \circ (r_{i,0}; \tilde{r}_i) &=& 2\mu e, \quad i \in \{1, \ldots n\}. \end{array}$$

The proof is done. □

### 1.3.2 Linear system of the IP algorithm

The IP algorithm is to apply the Newton method for (1.15), while driving $\mu$ to zero and maintaining the iterates $(t_i; \bar{u}_i)$, $(t'_i; \tilde{u}_i)$, $(r_{i,0}; \bar{r}_i)$ and $(r_{i,0}; \tilde{r}_i)$ in the interior of the Lorentz cones. The linear system to solve at each iteration is of the form

$$\begin{pmatrix} M & -H_0^\top & -\bar{H}^\top & -\tilde{H}^\top & & & & \\ -H_0 & & & & 1 & & 1 & \\ -\bar{H} & & & & & I & & \\ -\tilde{H} & & & & & & & I \\ & t & \bar{u}^\top & & r_0 & \bar{r}^\top & & \\ & \bar{u} & tI & & \bar{r} & r_0 I & & \\ & t' & & \tilde{u}^\top & & & r_0 & \tilde{r}^\top \\ & \tilde{u} & & t'I & & & \tilde{r} & r_0 I \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta r_0 \\ \Delta \bar{r} \\ \Delta \tilde{r} \\ \Delta t \\ \Delta \bar{u} \\ \Delta t' \\ \Delta \tilde{u} \end{pmatrix} = - \begin{pmatrix} Mv + f - H^\top r \\ t + t' - H_0 v - w_0 \\ \bar{u} - \bar{H}v - \bar{w} \\ \tilde{u} - \tilde{H}v - \tilde{w} \\ tr_0 + \bar{u}^\top \bar{r} - 2\sigma\mu \\ t\bar{r} + r_0 \bar{u} \\ t' r_0 + \tilde{u}^\top \tilde{r} - 2\sigma\mu \\ t' \tilde{r} + r_0 \tilde{u} \end{pmatrix}. \tag{1.18}$$

The linear system (1.18) is represented for the case $n = 1$. The case $n > 1$ can be easily deduced from this one by using a vector of unknows of the form

$$(\Delta v; \Delta r; \Delta u_{t,t'}),$$

where

$$\Delta u_{t,t'} = \begin{pmatrix} \Delta u_{1,t,t'} \\ \vdots \\ \Delta u_{n,t,t'} \end{pmatrix}, \text{ with } \Delta u_{i,t,t'} = \begin{pmatrix} \Delta t_i \\ \Delta \bar{u}_i \\ \Delta t'_i \\ \Delta \tilde{u}_i \end{pmatrix} \quad \text{and} \quad \Delta r = \begin{pmatrix} \Delta r_1 \\ \vdots \\ \Delta r_n \end{pmatrix}, \text{ with } \Delta r_i = \begin{pmatrix} \Delta r_{i,0} \\ \Delta \bar{r}_i \\ \Delta \tilde{r}_i \end{pmatrix},$$

for all $i \in \{1, \ldots, n\}$. In IP algorithms, the barrier parameter is set to $\mu = \frac{\mathrm{trace}(x \circ z)}{2n}$, when the complementarity condition is of the form $x \circ z$, where $x$ and $z$ are in the product of $n$ Lorentz cones, see [4, page 42]. For the system (1.15), we could set the barrier parameter to

$$\mu = \frac{\mathrm{trace}((t; \bar{u}) \circ (r_0; \bar{r})) + \mathrm{trace}((t'; \tilde{u}) \circ (r_0; \tilde{r}))}{2n} = \frac{(Ju_{t,t'})^\top r}{n} = \frac{u^\top r}{n}.$$

The parameter $\sigma \in (0, 1)$ is the centralization parameter. It is chosen according to different strategies which are presented in the next sections.

### 1.3.3 Step-length computation

Once the linear sytem is solved, we have update the variables $v$, $u$, $r$ and $t$ so that the next iteration remains in the interior of the second order cones, i.e., the inequalities of (1.15) must strictly hold. Usually, a primal step-length $\alpha_{\mathrm{P}} \in (0, 1]$ and a dual step-length $\alpha_{\mathrm{D}} \in (0, 1]$ are used. Howerver, in the rolling friction case, we decide to choose the same step-lengths. The triple $(v, u_{t,t'}, r)$ is updated to $(v^+, u^+_{t,t'}, r^+)$ by

$$v^+ = v + \alpha \Delta v, \quad u^+_{t,t'} = u_{t,t'} + \alpha \Delta u_{t,t'}, \quad r^+ = r + \alpha \Delta r.$$

For the detailed calculations of $\alpha$, see [16, §2.4]. In the next section, a characteristic phenomenon associated to the dual infeasibility will be presented to clarify the fact that step-lengths should be identical in our case.

### 1.3.4 A dual infeasibility phenomenon

Assuming that the system is solved exactly, i.e., $\Delta v$, $\Delta u$ and $\Delta r$ are exactly computed by (1.18), then, theoretically, primal and dual infeasibilities have to decrease monotonically along the iterations. In fact, the primal infeasibility for the next iteration is given by

$$\begin{aligned} \text{pinfeas}^+ &= Hv^+ + w - Ju^+_{t,t'} \\ &= H(v + \alpha_{\mathrm{P}} \Delta v) + w - J(u_{t,t'} + \alpha_{\mathrm{P}} \Delta u_{t,t'}) \\ &= (Hv + w - Ju_{t,t'}) + \alpha_{\mathrm{P}} H \Delta v - \alpha_{\mathrm{P}} J \Delta u_{t,t'} \\ &= \text{pinfeas} + (-\alpha_{\mathrm{P}} \text{pinfeas} + \alpha_{\mathrm{P}} \text{pinfeas}) + \alpha_{\mathrm{P}} H \Delta v - \alpha_{\mathrm{P}} J \Delta u_{t,t'} \\ &= (1 - \alpha_{\mathrm{P}}) \text{pinfeas} + \alpha_{\mathrm{P}} (\text{pinfeas} - (-H \Delta v + J \Delta u_{t,t'})) \\ &= (1 - \alpha_{\mathrm{P}}) \text{pinfeas} + \alpha_{\mathrm{P}} \xi_{\mathrm{P}}, \end{aligned} \tag{1.19}$$

where $\xi_{\mathrm{P}} = \text{pinfeas} - (-H\Delta v + J\Delta u_{t,t'})$ is the residual vector in solving primal feasibility equations in (1.18). Since (1.18) is exactly solved, then $\xi_{\mathrm{P}} = 0$ and $\|\text{pinfeas}^+\| = (1 - \alpha_{\mathrm{P}})\|\text{pinfeas}\|$ which is monotonically decreasing. However, this is not totally the case for the dual residual since

$$\text{dinfeas}^+ = (1 - \alpha_{\mathrm{D}}) \text{dinfeas} + \alpha_{\mathrm{D}} \xi_{\mathrm{D}} + (\alpha_{\mathrm{D}} - \alpha_{\mathrm{P}}) M \Delta v,$$

which depends also on the term $(\alpha_{\mathrm{D}} - \alpha_{\mathrm{P}})M\Delta v$. This means that if the steplengths $\alpha_{\mathrm{D}}$ and $\alpha_{\mathrm{P}}$ are identical at each iteration then the dual infeasibility behavior occurs normally. On the other hand, its value could increase abruptly and abnormally, so the direction to update the variables $v$ and $r$ is no longer the Newton direction. We call this phenomenon as a $jump$. This comes from the fact that the vector $v$, which is initially a primal variable, appears in the dual problem. By experiments presented in 2.3, we will observe the jump phenomenon which leads to the slower convergence. Therefore, we have to take $\alpha_{\mathrm{P}} = \alpha_{\mathrm{D}}$ in the algorithm.

**Remark 1.3.2.** An additional remark on the linearization of a linear equation.
Consider a system of equations of the form $F(X) = 0$, solved by means of the Newton method, i.e., solve the linear equation $F'(X)\Delta X = -F(X)$, then set $X^+ = X + \alpha\Delta X$, where $\alpha \in (0, 1]$ is a step-length. Suppose that a part of the system is a linear equation of the form

$$Ax + By = c,$$

where $x$ and $y$ are parts of the variable $X$. The linearization of the equation $F = 0$ leads to

$$A\Delta x + B\Delta y = -(Ax + By - c).$$

Suppose that different step-lengths are used to update the variables $x$ and $y$, that is

$$x^+ = x + \alpha\Delta x \quad \text{and} \quad y^+ = y + \beta\Delta y.$$

We then have

$$\begin{aligned} A(x + \alpha\Delta x) + B(y + \beta\Delta y) - c &= (1 - \alpha)(Ax + By - c) + (\beta - \alpha)B\Delta y, \\ &= (1 - \beta)(Ax + By - c) + (\alpha - \beta)A\Delta x. \end{aligned}$$

If $\alpha = \beta$, then we see that the residual of the linear equation is reduced by a factor $(1 - \alpha)$. Moreover, if $\alpha = \beta = 1$ then the residual is reduced to zero. But when $\alpha \neq \beta$, we see that even if the residual is zero at the current iteration (i.e., $Ax + By - c = 0$), at the next iteration the norm of the residual is equal to $|\alpha - \beta|\|A\Delta x\| = |\alpha - \beta|\|B\Delta y\|$, and thus does not vanish.

### 1.3.5 Simple IP algorithm

We will now give a simple version of IP algorithm (Algorithm 1) applied for the rolling friction problem with the tools presented in the previous subsections.

**Algorithm 1** Simple IP algorithm

```
Choose some tolerance ε > 0, σ ∈ (0,1) and set the starting iteration number k = 0
Set a starting point (v^0, u^0_{t,t'}, r^0) ∈ ℝ^m × ℝ^{n(d+1)} × ℝ^{nd}
for k = 0, 1, 2, ... do
    pinfeas^k ← ‖J u^k_{t,t'} − H v^k − w‖_F / (1 + ‖w‖_2)            ▷ See [16, page 191]
    dinfeas^k ← ‖M v^k + f − H^⊤ r^k‖_F / (1 + ‖f‖_2)
    μ^k = (J u^k_{t,t'})^⊤ r^k / n
    if max{pinfeas^k, dinfeas^k, μ^k} < ε then                         ▷ STOP condition
        STOP
    Solve (1.18) to obtain (Δv^k, Δu^k_{t,t'}, Δr^k)
    α_{p1} ← get_step_length((t^k; ū^k), (Δt^k; Δū^k))                 ▷ See section 1.3.3
    α_{p2} ← get_step_length((t'^k; ũ^k), (Δt'^k; Δũ^k))
    α_{d1} ← get_step_length((r^k_0; r̄^k), (Δr^k_0; Δr̄^k))
    α_{d2} ← get_step_length((r^k_0; r̃^k), (Δr^k_0; Δr̃^k))
    Set α ← min{α_{p1}, α_{p2}, α_{d1}, α_{d2}}
    Set (v^{k+1}, u^{k+1}_{t,t'}, r^{k+1}) = (v^k, u^k_{t,t'}, r^k) + α(Δv^k, Δu^k_{t,t'}, Δr^k)
```

### 1.3.6 IP algorithm with Mehrotra scheme

In pratice, Mehrotra proposed an efficient primal-dual interior-point algorithm [10] with the preditor-corrector scheme. There are two linear systems to solve with the same coefficients matrix. The first system is solved with $\sigma = 0$, the second system is slightly different. Following the SDPT3 implementation in [16, page 192] to define the optimal centering parameter $\sigma$, we apply this for the rolling friction case by Algorithm 2. Note that the modified RHS (1.20) is used in the corrector step to solve (1.18)

$$\begin{pmatrix} Mv + f - H^\top r \\ Ju_{t,t'} - Hv - w \\ (t;\bar{u}) \circ (r_0;\bar{r}) + (\Delta t;\Delta\bar{u}) \circ (\Delta r_0;\Delta\bar{r}) - 2\sigma\mu e \\ (t';\tilde{u}) \circ (r_0;\tilde{r}) + (\Delta t';\Delta\tilde{u}) \circ (\Delta r_0;\Delta\tilde{r}) - 2\sigma\mu e \end{pmatrix}. \tag{1.20}$$

**Algorithm 2** IP algorithm with predictor-corrector scheme

Choose some tolerance $\epsilon > 0$ and set the starting iteration number $k = 0$<br>
Set a starting point $(v^0, u^0_{t,t'}, r^0) \in \mathbb{R}^m \times \mathbb{R}^{n(d+1)} \times \mathbb{R}^{nd}$<br>
**for** $k = 0, 1, 2, \dots$ **do**<br>
    pinfeas$^k \leftarrow \|Ju^k_{t,t'} - Hv^k - w\|_F/(1 + \|w\|_2)$<br>
    dinfeas$^k \leftarrow \|Mv^k + f - H^\top r^k\|_F/(1 + \|f\|_2)$<br>
    $\mu^k = (Ju^k_{t,t'})^\top r^k/n$<br>
    **if** $\max\{\text{pinfeas}^k, \text{dinfeas}^k, \mu^k\} < \epsilon$ **then** ▷ STOP condition<br>
        STOP<br>
    ———- PREDICTOR Step ———-<br>
    $\sigma^k \leftarrow 0$<br>
    Solve (1.18) to obtain $(\Delta v^k, \Delta u^k_{t,t'}, \Delta r^k)$<br>
    $\alpha_{p_1} \leftarrow$ get_step_length$((t^k;\bar{u}^k), (\Delta t^k;\Delta\bar{u}^k))$ ▷ See section 1.3.3<br>
    $\alpha_{p_2} \leftarrow$ get_step_length$((t'^k;\tilde{u}^k), (\Delta t'^k;\Delta\tilde{u}^k))$<br>
    $\alpha_{d_1} \leftarrow$ get_step_length$((r^k_0;\bar{r}^k), (\Delta r^k_0;\Delta\bar{r}^k))$<br>
    $\alpha_{d_2} \leftarrow$ get_step_length$((r^k_0;\tilde{r}^k), (\Delta r^k_0;\Delta\tilde{r}^k))$<br>
    Set $\alpha_{\text{P}} \leftarrow \min\{\alpha_{p_1}, \alpha_{p_2}\}, \alpha_{\text{D}} \leftarrow \min\{\alpha_{d_1}, \alpha_{d_2}\}$<br>
    Set $\gamma \leftarrow 0.9 + 0.09 * \min\{\alpha_{\text{P}}, \alpha_{\text{D}}\}$<br>
    ———- CORRECTOR Step ———-<br>
    $\mu_a \leftarrow (Ju_{t,t'} + \alpha_{\text{P}} J\Delta u_{t,t'})^\top (r + \alpha_{\text{D}}\Delta r)/nd$<br>
    **if** $\mu^k < 10^{-5}$ **then**<br>
        $e \leftarrow \max\{1, 3 * \min\{\alpha_{\text{P}}, \alpha_{\text{D}}\}\}$<br>
    **else**<br>
        $e \leftarrow 1$<br>
    $\sigma^k \leftarrow \min\{1, (\mu_a/\mu^k)^e\}$<br>
    Solve (1.18) again with the modified RHS (1.20)<br>
    $\alpha_{p_1} \leftarrow \gamma *$ get_step_length$((t^k;\bar{u}^k), (\Delta t^k;\Delta\bar{u}^k))$ ▷ Compute again the step-lengths<br>
    $\alpha_{p_2} \leftarrow \gamma *$ get_step_length$((t'^k;\tilde{u}^k), (\Delta t'^k;\Delta\tilde{u}^k))$<br>
    $\alpha_{d_1} \leftarrow \gamma *$ get_step_length$((r^k_0;\bar{r}^k), (\Delta r^k_0;\Delta\bar{r}^k))$<br>
    $\alpha_{d_2} \leftarrow \gamma *$ get_step_length$((r^k_0;\tilde{r}^k), (\Delta r^k_0;\Delta\tilde{r}^k))$<br>
    Set $\alpha \leftarrow \min\{\alpha_{p_1}, \alpha_{p_2}, \alpha_{d_1}, \alpha_{d_2}\}$<br>
    Set $(v^{k+1}, u^{k+1}_{t,t'}, r^{k+1}) = (v^k, u^k_{t,t'}, r^k) + \alpha(\Delta v^k, \Delta u^k_{t,t'}, \Delta r^k)$

### 1.3.7 First simple example.

Consider a sphere of mass $m$ and radius $R$ rolling on a plan. Let $(x, y, z)$ be the coordinates of the contact point, $z$ being the vertical one. The matrix $M$ is defined by $M = \text{diag}(m, m, m, I_1, I_1, I_1)$, where $I_1 = \frac{2}{5}mR^2$. The global velocity vector is $v = (\dot{x}; \dot{y}; \dot{z}; \dot{\theta}; \dot{\psi}; \dot{\phi})$, where $\dot{\theta}, \dot{\phi}, \dot{\psi}$ are respectively the angular velocities arround $x, y, z$. The vector $f$ is set to $f = (0; 0; g; M_x; M_y; M_z)$, with, e.g., $m = 1, g = 10$, $M_x = M_y = M_z = 1$. The matrix $H$ is defined by

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

The vector $w$ is set to zero in the constraint $Hv + w = u$. The velocity vector is then $u = (\dot{z}; \dot{x}; \dot{y}; \dot{\theta}; \dot{\phi})$. The friction coefficients are set to $\mu = 0.5$ and $\mu_{\text{R}} = 0.01$.

## 1.4 Symmetrization of linearized system

We see that the predictor-corrector algorithm 2 utilizes the solution of two linear systems being identical along the iterations. Thus, an implementation that solves only once the linear system by LDL factorization should be introduced, see [13, page 54-55]. However, the linear system is not symmetrizable by means of standard transformation used in IP method, even the coefficient matrix can be singular. This is due to the fact that the Jordan product is not commutative. Hence, this section presents the scaling technique to retrieve a symmetric non-singular system. The main reason comes from the fact that the solution of a symmetric system is cheaper than a nonsymmetric one. Moreover, LDL factorization allows to control the inertia of a matrix, i.e., the number of positive, negative and null eigenvalues of the factorized matrix, an important information that can be used in an optimization process.

### 1.4.1 Nesterov-Todd scaling strategy

By left multiplying the last four equations of (1.18) by $\text{arw}(t; \bar{u})^{-1}$ and $\text{arw}(t'; \tilde{u})^{-1}$, the matrix of the linear system becomes

$$\begin{pmatrix} M & -H^\top & \\ -H & & J \\ & J^\top & A \end{pmatrix}, \tag{1.21}$$

where

$$A = \begin{pmatrix} \text{arw}(t; \bar{u})^{-1}\text{arw}(r_0; \bar{r}) & \\ & \text{arw}(t'; \tilde{u})^{-1}\text{arw}(r_0; \tilde{r}) \end{pmatrix}.$$

Suppose that the matrices $\text{arw}(t; \bar{u})$ and $\text{arw}(r_0; \bar{r})$ commute, and so do $\text{arw}(t'; \tilde{u})$ and $\text{arw}(r_0; \tilde{r})$. In that case, the matrix $A$ is symmetric, and so do the matrix (1.21). In the general case, the matrix $A$ is not symmetric. This is why we have to introduce a scaling strategy (which is a local change of variables) so that the arrow matrices commute and then the result system is symmetric, see [4, page 42]. The transformation of the nonsymmetric (scaled) system (1.18) to a symmetric one remains exactly the same.

Let us now introduce two quadratic representation operators, see Annex, $Q_{\bar{p}}, Q_{\tilde{p}}$ by which we change locally the variables for the rolling friction case. Let $\bar{p}, \tilde{p}$ be the Nesterov and Todd (NT) directions given by

$$\begin{aligned}\bar{p} &= \left\{Q_{(t;\bar{u})^{1/2}}[Q_{(t;\bar{u})^{1/2}}(r_0;\bar{r})]^{-1/2}\right\}^{-1/2}, \\ \tilde{p} &= \left\{Q_{(t';\tilde{u})^{1/2}}[Q_{(t';\tilde{u})^{1/2}}(r_0;\tilde{r})]^{-1/2}\right\}^{-1/2},\end{aligned} \tag{1.22}$$

we define

$$\begin{aligned}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix} &= Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}, &\quad \begin{pmatrix}\check{r}_0\\ \check{\bar{r}}\end{pmatrix} &= Q_{\bar{p}^{-1}}\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix},\\ \begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix} &= Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}, &\quad \begin{pmatrix}\check{r}'_0\\ \check{\tilde{r}}\end{pmatrix} &= Q_{\tilde{p}^{-1}}\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix},\end{aligned}$$

where

$$\begin{aligned}\begin{pmatrix}t\\ \bar{u}\end{pmatrix} &= \left(\begin{pmatrix}t_1\\ \bar{u}_1\end{pmatrix};\dots;\begin{pmatrix}t_n\\ \bar{u}_n\end{pmatrix}\right)\in\mathbb{R}^{(d-2)n}, &\quad \begin{pmatrix}r_0\\ \bar{r}\end{pmatrix} &= \left(\begin{pmatrix}r_{1,0}\\ \bar{r}_1\end{pmatrix};\dots;\begin{pmatrix}r_{n,0}\\ \bar{r}_n\end{pmatrix}\right)\in\mathbb{R}^{(d-2)n},\\ \begin{pmatrix}t'\\ \tilde{u}\end{pmatrix} &= \left(\begin{pmatrix}t'_1\\ \tilde{u}_1\end{pmatrix};\dots;\begin{pmatrix}t'_n\\ \tilde{u}_n\end{pmatrix}\right)\in\mathbb{R}^{(d-2)n}, &\quad \begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix} &= \left(\begin{pmatrix}r_{1,0}\\ \tilde{r}_1\end{pmatrix};\dots;\begin{pmatrix}r_{n,0}\\ \tilde{r}_n\end{pmatrix}\right)\in\mathbb{R}^{(d-2)n}.\end{aligned}$$

**Remark 1.4.1.** For the scaling technique, we can choose the directions $\bar{p},\tilde{p}$ in different ways so that $(\hat{t}_i;\hat{\bar{u}}_i)$ and $(\check{r}_{i,0};\check{\bar{r}}_i)$ commute; also for the pair $(\check{r}'_{i,0};\check{\tilde{r}}_i)$. In the rolling friction case, we select NT direction to get this. Indeed, we have[2]

$$Q_{\bar{p}}^2\begin{pmatrix}t\\ \bar{u}\end{pmatrix} = \begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}, \quad \text{and} \quad Q_{\tilde{p}}^2\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix} = \begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}.$$

Hence,

$$\begin{aligned}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix} &= Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix} = Q_{\bar{p}^{-1}}\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix} = \begin{pmatrix}\check{r}_0\\ \check{\bar{r}}\end{pmatrix},\\ \begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix} &= Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix} = Q_{\tilde{p}^{-1}}\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix} = \begin{pmatrix}\check{r}'_0\\ \check{\tilde{r}}\end{pmatrix}.\end{aligned}$$

In practice, IP method using NT direction is faster than the others directions for the second-order cones. Computational results and explanations shown in [16, page 212] indicate this fact.

Using the fact that $Q_{\bar{p}^{-1}}Q_{\bar{p}} = I$, $Q_{\tilde{p}^{-1}}Q_{\tilde{p}} = I$, the definitions $Q_p := \begin{pmatrix}Q_{\bar{p}} & \\ & Q_{\tilde{p}}\end{pmatrix}$ and

$$\begin{aligned}u_{t,t'} &= \begin{pmatrix}t\\ \bar{u}\\ t'\\ \tilde{u}\end{pmatrix} = Q_p^{-1}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\\ \hat{t}'\\ \hat{\tilde{u}}\end{pmatrix} = Q_p^{-1}\hat{u}_{t,t'},\\ r_{\mathrm{J}} = J^\top r &= \begin{pmatrix}r_0\\ \bar{r}\\ r_0\\ \tilde{r}\end{pmatrix} = Q_p\begin{pmatrix}\check{r}_0\\ \check{\bar{r}}\\ \check{r}'_0\\ \check{\tilde{r}}\end{pmatrix} = Q_p\check{r}_{\mathrm{J}},\end{aligned}$$

since $Q_p(\mathcal{L}\times\mathcal{L}) = \mathcal{L}\times\mathcal{L}$, see [4, theorem 9], the primal-dual problems (1.6) and (1.7) become

$$\begin{array}{ll}\min\limits_{v,u_{t,t'}} & \frac{1}{2}v^\top Mv + f^\top v\\ \text{s.t.} & Hv + w = Q_p^{-1}\hat{u}_{t,t'},\\ & \left(\begin{pmatrix}\hat{t}_i\\ \hat{\bar{u}}_i\end{pmatrix},\begin{pmatrix}\hat{t}'_i\\ \hat{\tilde{u}}_i\end{pmatrix}\right)\in\mathcal{L}\times\mathcal{L}, i\in\{1,\dots,n\},\end{array} \tag{1.23}$$

[2]See [4, page 42] for a proof.

and

$$\begin{array}{ll} \min\limits_{v,r} & \frac{1}{2}v^\top Mv + w^\top(KQ_p\check{r}_{\text{J}}) \\ \text{s.t.} & Mv + f - H^\top KQ_p\check{r}_{\text{J}} = 0, \\ & \left(\begin{pmatrix}\check{r}_{i,0}\\ \check{\bar{r}}_i\end{pmatrix}, \begin{pmatrix}\check{r}'_{i,0}\\ \check{\tilde{r}}_i\end{pmatrix}\right) \in \mathcal{L}\times\mathcal{L}, i\in\{1,\dots,n\}, \end{array} \tag{1.24}$$

where

$$K = \begin{pmatrix} k & & \\ & \ddots & \\ & & k \end{pmatrix} \in \mathbb{R}^{5n\times 6n} \quad \text{and} \quad k = \begin{pmatrix} 1 & 0 & \\ & I_2 & \\ & & I_2 \end{pmatrix} \in \mathbb{R}^{5\times 6}.$$

Note that we take the matrix $K$ such that $KJ^\top = I$. Thus, the new pertubed KKT system associated to (1.6) - (1.7) presents

$$\begin{array}{rcll} Mv + f - H^\top KQ_p\check{r}_{\text{J}} &=& 0, & \\ -Hv - w + Q_p^{-1}\hat{u}_t &=& 0, & \\ \left(\begin{pmatrix}\hat{t}_i\\ \hat{\bar{u}}_i\end{pmatrix}, \begin{pmatrix}\check{r}_{i,0}\\ \check{\bar{r}}_i\end{pmatrix}\right) &=& 2\mu e, & i\in\{1,\dots,n\} \\ \left(\begin{pmatrix}\hat{t}'_i\\ \hat{\tilde{u}}_i\end{pmatrix}, \begin{pmatrix}\check{r}'_{i,0}\\ \check{\tilde{r}}_i\end{pmatrix}\right) &=& 2\mu e, & i\in\{1,\dots,n\} \\ ((t_i;\bar{u}_i),(t'_i;\tilde{u}_i)) &\in& \text{int}(\mathcal{L}\times\mathcal{L}), & i\in\{1,\dots,n\}. \end{array} \tag{1.25}$$

The key lemma is following.

**Lemma 1.4.2.** Let $\mathbf{e} = (e_1,\dots,e_n)$ and $e_i = (1;0;0)\in\mathbb{R}^3, i\in\{1,\dots,n\}$.
$(\Delta v, \Delta\hat{u}_t, \Delta\check{r}_{\text{J}})$ solves the linearized system of equations (1.25)

$$\begin{array}{rcl} M\Delta v - H^\top KQ_p\Delta\check{r}_{\text{J}} &=& -(Mv + f - H^\top KQ_p\check{r}_{\text{J}}), \\ -H\Delta v + Q_p^{-1}\Delta\hat{u}_t &=& -(Q_p^{-1}\hat{u}_t - Hv - w), \\ \begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}\circ\begin{pmatrix}\Delta\check{r}_0\\ \Delta\check{\bar{r}}\end{pmatrix} + \begin{pmatrix}\Delta\hat{t}\\ \Delta\hat{\bar{u}}\end{pmatrix}\circ\begin{pmatrix}\check{r}_0\\ \check{\bar{r}}\end{pmatrix} &=& -\left[\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}\circ\begin{pmatrix}\check{r}_0\\ \check{\bar{r}}\end{pmatrix} - 2\mu\mathbf{e}\right], \\ \begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}\circ\begin{pmatrix}\Delta\check{r}'_0\\ \Delta\check{\tilde{r}}\end{pmatrix} + \begin{pmatrix}\Delta\hat{t}'\\ \Delta\hat{\tilde{u}}\end{pmatrix}\circ\begin{pmatrix}\check{r}_0\\ \check{\tilde{r}}\end{pmatrix} &=& -\left[\begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}\circ\begin{pmatrix}\check{r}'_0\\ \check{\tilde{r}}\end{pmatrix} - 2\mu\mathbf{e}\right], \end{array} \tag{1.26}$$

if and only if $(\Delta v, \Delta u, \Delta r)$ solves

$$\begin{array}{rcl} M\Delta v - H^\top\Delta r &=& -(Mv + f - H^\top r), \\ -H\Delta v + J\Delta u &=& -(Ju_{t,t'} - Hv - w), \\ \multicolumn{3}{l}{\text{arw}\left(Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}\right)Q_{\bar{p}^{-1}}\begin{pmatrix}\Delta r_0\\ \Delta\bar{r}\end{pmatrix} + \text{arw}\left(Q_{\bar{p}^{-1}}\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}\right)Q_{\bar{p}}\begin{pmatrix}\Delta t\\ \Delta\bar{u}\end{pmatrix}} \\ &=& -\left[\text{arw}\left(Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}\right)\text{arw}\left(Q_{\bar{p}^{-1}}\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}\right) - 2\mu\mathbf{e}\right], \\ \multicolumn{3}{l}{\text{arw}\left(Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}\right)Q_{\tilde{p}^{-1}}\begin{pmatrix}\Delta r_0\\ \Delta\tilde{r}\end{pmatrix} + \text{arw}\left(Q_{\tilde{p}^{-1}}\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}\right)Q_{\tilde{p}}\begin{pmatrix}\Delta t'\\ \Delta\tilde{u}\end{pmatrix}} \\ &=& -\left[\text{arw}\left(Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}\right)\text{arw}\left(Q_{\tilde{p}^{-1}}\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}\right) - 2\mu\mathbf{e}\right]. \end{array} \tag{1.27}$$

The proof is directly produced. We observe that only blocks of last 4 equations of (1.27) change from the corresponding ones of (1.18). Thus, we just consider these blocks of 4 equations when the scaling technique is applied by means of a transform multiplied respectively by

$$\left[\text{arw}\left(Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}\right)Q_{\bar{p}^{-1}}\right]^{-1} = Q_{\bar{p}}\text{arw}\left(Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}\right)^{-1},$$

$$\left[\text{arw}\left(Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}\right)Q_{\tilde{p}^{-1}}\right]^{-1} = Q_{\tilde{p}}\text{arw}\left(Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}\right)^{-1}.$$

Using the fact that

$$\operatorname{arw}\left(Q_{\bar{p}}\begin{pmatrix}t\\ \bar{u}\end{pmatrix}\right)^{-1}\operatorname{arw}\left(Q_{\bar{p}^{-1}}\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}\right)=\operatorname{arw}\left(Q_{\tilde{p}}\begin{pmatrix}t'\\ \tilde{u}\end{pmatrix}\right)^{-1}\operatorname{arw}\left(Q_{\tilde{p}^{-1}}\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}\right)=I,$$

we obtain

$$\begin{aligned}\begin{pmatrix}\Delta r_0\\ \Delta\bar{r}\end{pmatrix}+Q_{\bar{p}^2}\begin{pmatrix}\Delta t\\ \Delta\bar{u}\end{pmatrix}&=-\begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}+2\mu Q_{\bar{p}}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}^{-1},\\ \begin{pmatrix}\Delta r_0\\ \Delta\tilde{r}\end{pmatrix}+Q_{\tilde{p}^2}\begin{pmatrix}\Delta t'\\ \Delta\tilde{u}\end{pmatrix}&=-\begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}+2\mu Q_{\tilde{p}}\begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}^{-1}.\end{aligned}\tag{1.28}$$

Eventually, the matrix of linear system (1.27) becomes

$$\begin{pmatrix}M & -H^\top & \\ -H & & J\\ & J^\top & Q_p^2\end{pmatrix},\tag{1.29}$$

and the right-hand-side (RHS) shows

$$-\begin{pmatrix}Mv+f-H^\top r\\ Ju_{t,t'}-Hv-w\\ \begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}-2\mu Q_{\bar{p}}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}^{-1}\\ \begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}-2\mu Q_{\tilde{p}}\begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}^{-1}\end{pmatrix}.\tag{1.30}$$

### 1.4.2 Scaling technique algorithm for IP method

The algorithm 3 now presents for (1.31)

$$\begin{pmatrix}M & -H^\top & \\ -H & & J\\ & J^\top & Q_p^2\end{pmatrix}\begin{pmatrix}\Delta v\\ \Delta r\\ \Delta t\\ \Delta\bar{u}\\ \Delta t'\\ \Delta\tilde{u}\end{pmatrix}=-\begin{pmatrix}Mv+f-H^\top r\\ Ju_{t,t'}-Hv-w\\ \begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}-2\mu\sigma Q_{\bar{p}}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}^{-1}\\ \begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}-2\mu\sigma Q_{\tilde{p}}\begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}^{-1}\end{pmatrix}.\tag{1.31}$$

Note that the modified RHS (1.32) is used in the corrector step to solve (1.31)

$$\begin{pmatrix}Mv+f-H^\top r\\ Ju_{t,t'}-Hv-w\\ \begin{pmatrix}r_0\\ \bar{r}\end{pmatrix}+\left(Q_{\bar{p}}\begin{pmatrix}\Delta t\\ \Delta\bar{u}\end{pmatrix}\right)\circ\left(Q_{\bar{p}}^{-1}\begin{pmatrix}\Delta r_0\\ \Delta\bar{r}\end{pmatrix}\right)-2\mu\sigma Q_{\bar{p}}\begin{pmatrix}\hat{t}\\ \hat{\bar{u}}\end{pmatrix}^{-1}\\ \begin{pmatrix}r_0\\ \tilde{r}\end{pmatrix}+\left(Q_{\tilde{p}}\begin{pmatrix}\Delta t'\\ \Delta\tilde{u}\end{pmatrix}\right)\circ\left(Q_{\tilde{p}}^{-1}\begin{pmatrix}\Delta r_0\\ \Delta\tilde{r}\end{pmatrix}\right)-2\mu\sigma Q_{\tilde{p}}\begin{pmatrix}\hat{t}'\\ \hat{\tilde{u}}\end{pmatrix}^{-1}\end{pmatrix}.\tag{1.32}$$

**Algorithm 3** IP algorithm with predictor-corrector scheme and scaling technique

Choose some tolerance $\epsilon > 0$ and set the starting iteration number $k = 0$
Set the matrix $J$
Set a starting point $(v^0, u^0_{t,t'}, r^0) \in \mathbb{R}^m \times \mathbb{R}^{n(d+1)} \times \mathbb{R}^{nd}$
**for** $k = 0, 1, 2, \dots$ **do**
  pinfeas$^k \leftarrow \|u^k_{t,t'} - Hv^k - w\|_F/(1 + \|w\|_2)$
  dinfeas$^k \leftarrow \|Mv^k + f - H^\top r^k\|_F/(1 + \|f\|_2)$
  $\mu^k = (u^k_{t,t'})^\top r^k/n$
  **if** $\max\{\text{pinfeas}^k, \text{dinfeas}^k, \mu^k\} < \epsilon$ **then** ▷ STOP condition
    STOP
  $\bar{p}^k \leftarrow \text{NT}((t^k; \bar{u}^k), (r^k_0; \bar{r}^k))$ ▷ Compute NT directions
  $\tilde{p}^k \leftarrow \text{NT}((t'^k; \tilde{u}^k), (r^k_0; \tilde{r}^k))$
  $Q_{\bar{p}^k} \leftarrow$ quad_repr$(\bar{p}^k)$ ▷ Compute quadratic representations
  $Q_{\tilde{p}^k} \leftarrow$ quad_repr$(\tilde{p}^k)$
  ——— PREDICTOR Step ———
  $\sigma^k \leftarrow 0$
  Solve (1.31)
  $\alpha_{p_1} \leftarrow$ get_step_length$((t^k; \bar{u}^k), (\Delta t^k; \Delta\bar{u}^k))$ ▷ See section 1.3.3
  $\alpha_{p_2} \leftarrow$ get_step_length$((t'^k; \tilde{u}^k), (\Delta t'^k; \Delta\tilde{u}^k))$
  $\alpha_{d_1} \leftarrow$ get_step_length$((r^k_0; \bar{r}^k), (\Delta r^k_0; \Delta\bar{r}^k))$
  $\alpha_{d_2} \leftarrow$ get_step_length$((r^k_0; \tilde{r}^k), (\Delta r^k_0; \Delta\tilde{r}^k))$
  Set $\alpha_{\text{P}} \leftarrow \min\{\alpha_{p_1}, \alpha_{p_2}\}, \quad \alpha_{\text{D}} \leftarrow \min\{\alpha_{d_1}, \alpha_{d_2}\}$
  Set $\gamma \leftarrow 0.9 + 0.09 * \min\{\alpha_{\text{P}}, \alpha_{\text{D}}\}$
  ——— CORRECTOR Step ———
  $\mu_a \leftarrow (u_{t,t'} + \alpha_{\text{P}}\Delta u_{t,t'})^\top (r + \alpha_{\text{D}}\Delta r)/nd$
  **if** $\mu^k < 10^{-5}$ **then**
    $e \leftarrow \max\{1, 3 * \min\{\alpha_{\text{P}}, \alpha_{\text{D}}\}\}$
  **else**
    $e \leftarrow 1$
  $\sigma^k \leftarrow \min\{1, (\mu_a/\mu^k)^e\}$
  Solve (1.31) again with the modified RHS (1.32)
  $\alpha_{p_1} \leftarrow \gamma *$ get_step_length$((t^k; \bar{u}^k), (\Delta t^k; \Delta\bar{u}^k))$ ▷ Compute again the step-lengths
  $\alpha_{p_2} \leftarrow \gamma *$ get_step_length$((t'^k; \tilde{u}^k), (\Delta t'^k; \Delta\tilde{u}^k))$
  $\alpha_{d_1} \leftarrow \gamma *$ get_step_length$((r^k_0; \bar{r}^k), (\Delta r^k_0; \Delta\bar{r}^k))$
  $\alpha_{d_2} \leftarrow \gamma *$ get_step_length$((r^k_0; \tilde{r}^k), (\Delta r^k_0; \Delta\tilde{r}^k))$
  Set $\alpha \leftarrow \min\{\alpha_{p_1}, \alpha_{p_2}, \alpha_{d_1}, \alpha_{d_2}\}$
  Set $(v^{k+1}, u^{k+1}_{t,t'}, r^{k+1}) = (v^k, u^k_{t,t'}, r^k) + \alpha(\Delta v^k, \Delta u^k_{t,t'}, \Delta r^k)$

### 1.4.3 An alternative scaling operator

As proposed by Cai, Zhi and Toh, Kim-Chuan in [5, §2], in stead of using the scaling operator $Q_p$, we can introduce the NT scaling technique using a block diagonal scaling matrix $F = \mathrm{diag}(F_1, \ldots, F_n)$ such that $Fx = F^{-1}z$ for variables $x = (x_1; \ldots; x_n), z = (z_1; \ldots; z_n) \in \mathbb{R}^{nd}$ ($x_i, z_i \in \mathbb{R}^d, i \in \{1, \ldots, n\}$) belonging to the second-order cones. The elements $F_i$ for all $i \in \{1, \ldots, n\}$ are given by

$$F_i = \omega_i \begin{pmatrix} f_{i,0} & \bar{f}_i^\top \\ \bar{f}_i & I + \frac{1}{1+f_{i,0}} \bar{f}_i \bar{f}_i^\top \end{pmatrix}, \tag{1.33}$$

where

$$\omega_i = \sqrt[4]{\frac{\det(z_i)}{\det(x_i)}} \quad \text{and} \quad f_i = \begin{pmatrix} f_{i,0} \\ \bar{f}_i \end{pmatrix} = \frac{1}{\sqrt{2(\sqrt{\det(x_i)\det(z_i)} + x_i^\top z_i)}} \begin{pmatrix} \frac{1}{\omega_i} z_{i,0} + \omega_i x_{i,0} \\ \frac{1}{\omega_i} \bar{z}_i - \omega_i \bar{x}_i \end{pmatrix}.$$

Note that $\det(f_i) = 1$. In addition, Tsuchiya presents in [15, Proposition 2.1] an explicit formula for the inverse of $F$ given by

$$F_i^{-1} = \frac{1}{\omega_i} \begin{pmatrix} f_{i,0} & -\bar{f}_i^\top \\ -\bar{f}_i & I + \frac{1}{1+f_{i,0}} \bar{f}_i \bar{f}_i^\top \end{pmatrix}. \tag{1.34}$$

Tsuchiya also shown in [15, Proposition 7.6] that this such matrix $F$ is the unique symmetric positive definite matrix for the NT scaling technique, i.e., $F = Q_p$. Hence, we can apply these scaling operators for the NT scaling technique instead of quadratic representation operators $Q_{\bar{p}}, Q_{\tilde{p}}$ as introduced in the previous section. Numerically computation in practice by Python shows that this approach is faster than the predecessor. See section (2.1.4) for the summary.

# Chapter 2

# Numerical experiments

In this chapter, we will test the algorithms on a data set consisting of 201 experiments of the problem of unilateral contact between solid bodies with rolling friction. Each experiment will include a different number of contact points $n$ and degrees of freedom $m$. We also analyse and evaluate the results delivered by each algorithm in order to demonstrate their efficiency. Finally, we present a special phenomenon turning out for dual infeasibility as mentioned before.

## 2.1 Experiments

The following four algorithms will be tested in turn.

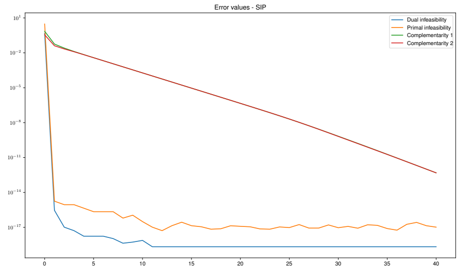
### 2.1.1 Simple IP algorithm

We follow directly the algorithm (1). In brief, we need to solve the following linear system at each iterations with centralization parameter $\sigma = 0.6$

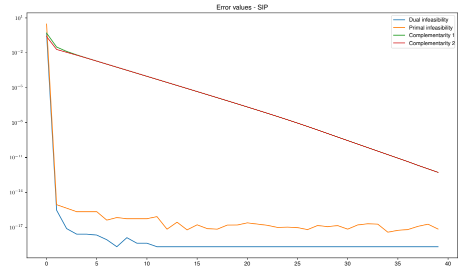
$$\begin{pmatrix} M & -H_0^\top & -\bar{H}^\top & -\tilde{H}^\top & & & & & \\ -H & & & & & J & & & \\ & t & \bar{u}^\top & & r_0 & \bar{r}^\top & & & \\ & \bar{u} & tI & & \bar{r} & r_0 I & & & \\ & t' & & \tilde{u}^\top & & & r_0 & \tilde{r}^\top \\ & \tilde{u} & & t'I & & & \tilde{r} & r_0 I \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta r \\ \Delta t \\ \Delta \bar{u} \\ \Delta t' \\ \Delta \tilde{u} \end{pmatrix} = - \begin{pmatrix} Mv + f - H^\top r \\ Ju_{t,t'} - Hv - w \\ tr_0 + \bar{u}^\top \bar{r} - 2\sigma\mu \\ (t;\bar{u}) \circ (r_0'; \bar{r}) - 2\sigma\mu e \\ (t';\tilde{u}) \circ (r_0;\tilde{r}) - 2\sigma\mu e \end{pmatrix}. \tag{2.1}$$

In practice, $\sigma \in (0, 1)$ will be chosen so that we can reach the optimality as quickly as possible. If the Newton steps toward the optimal point is so slow, $\sigma$ should be increased. If the algorithm fails in that the Newton steps tend to go beyond the cones, $\sigma$ should be decreased. Intermediate values of $\sigma \in (0, 1)$ would be priority choice, see [13, page 398].

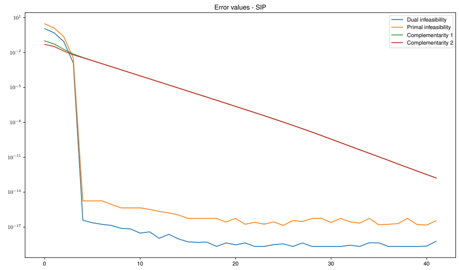
For each algorithm, we present the results for 4 experiments with the number of contact points $n$ respectively: 2, 4, 100 and 351. In each experiment, 4 graphs are displayed correspondingly to 4 error values are **Primal infeasibility**, **Dual infeasibility**, **Complementarity 1** and **Complementarity 2** along iterations. These complementarity values are corresponding complementarity conditions stated in (1.15). Obviously we expect the error values to decrease through iterations until the tolerance we want is reached. In the scope of this topic, we take the tolerance as $10^{-12}$. Figure 2.1 shows the evolution of error values. We especially interest in the robust decrease of **Primal-Dual infeasibilities** after only the first few iterations.



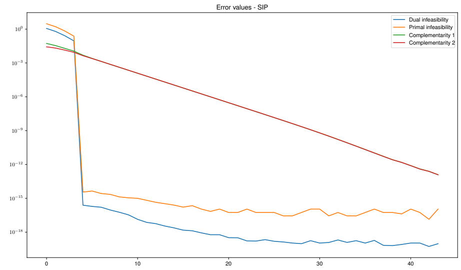

**(a)** $n = 2$




**(b)** $n = 4$




**(c)** $n = 100$



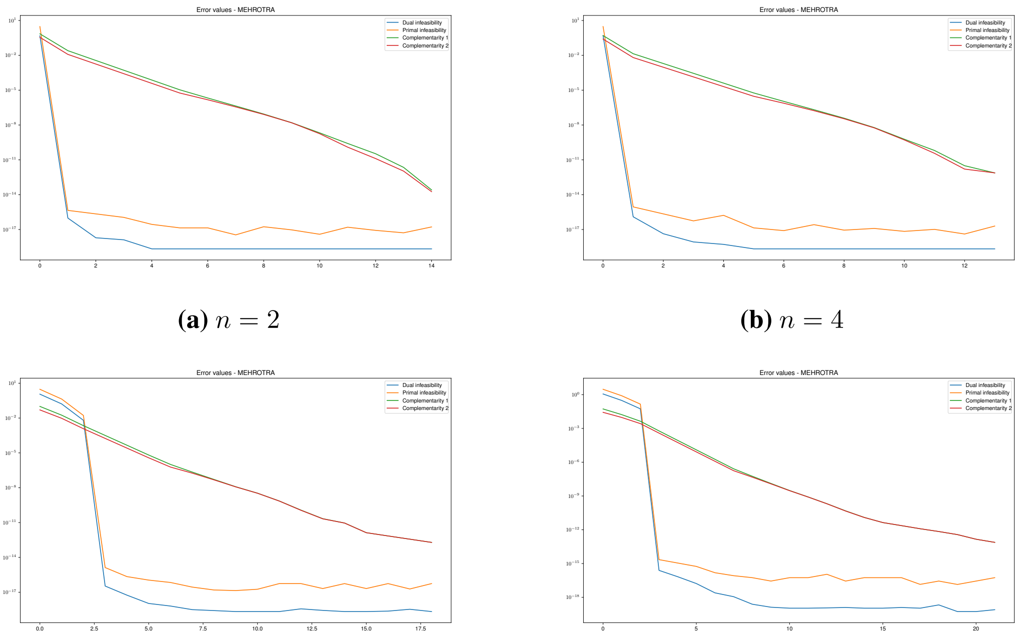

**(d)** $n = 351$

**Figure 2.1:** *Error values for Simple Interior-point algorithm* (1).

From these experiments solved by the simplest version of IP algorithm, we have achieved initial success in solving rolling friction problem with the proposed strategy.

### 2.1.2 IP algorithm with predictor-corrector scheme

By using the practical predictor-corrector algorithm proposed by Mehrotra, the optimal solution should be expeditiously reached. Following the algorithm (2), we firstly solve (2.1) with $\sigma = 0$ in order to get the prediction for a better Newton step at the next iteration. Then, we resolve (2.1) with the chosen $\sigma$. Since we solve twice the linear system with the same coefficient matrix, an LU factorization for this matrix is applied to accelerate the algorithm. Figure 2.2 gives us the demonstration for acceleration.




**(a)** $n = 2$ **(b)** $n = 4$

**(c)** $n = 100$ **(d)** $n = 351$

**Figure 2.2:** *Error values for Interior-point algorithm with Mehrotra scheme.*

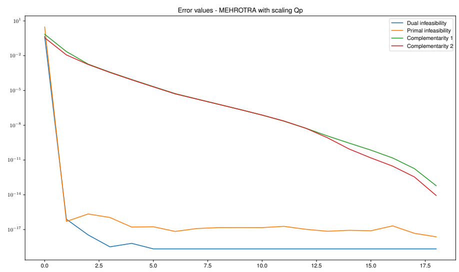
We can see that the optimal solutions are obtained after a number of iterations only half of simple IP algorithm. Despite the possibility of non-convergence [13, page 411], Mehrotra's practical algorithm still works efficiently for the rolling friction case.

### 2.1.3 IP algorithm with predictor-corrector scheme and NT scaling technique

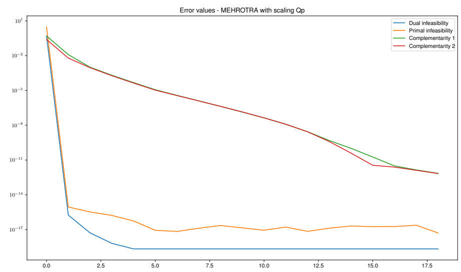
As mentioned in algorithm (3), we need to solve the following linear system along the iterations

$$\begin{pmatrix} M & -H^\top & \\ -H & & J \\ & J^\top & Q_p^2 \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta r \\ \Delta t \\ \Delta \bar{u} \\ \Delta t' \\ \Delta \tilde{u} \end{pmatrix} = - \begin{pmatrix} Mv + f - H^\top r \\ Ju_{t,t'} - Hv - w \\ \begin{pmatrix} r_0 \\ \bar{r} \end{pmatrix} - 2\mu\sigma Q_{\bar{p}} \begin{pmatrix} \hat{t} \\ \hat{\bar{u}} \end{pmatrix}^{-1} \\ \begin{pmatrix} r_0 \\ \tilde{r} \end{pmatrix} - 2\mu\sigma Q_{\tilde{p}} \begin{pmatrix} \hat{t}' \\ \hat{\tilde{u}} \end{pmatrix}^{-1} \end{pmatrix}. \tag{2.2}$$

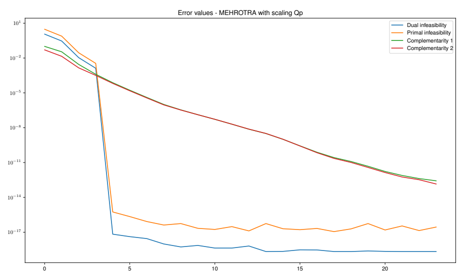
Some results are shown in Figure 2.3.



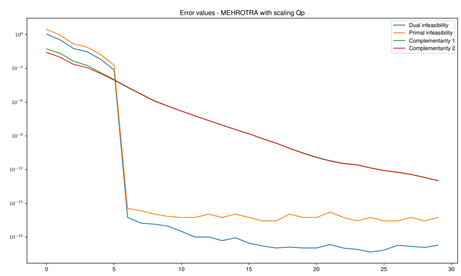

**(a)** $n = 2$




**(b)** $n = 4$




**(c)** $n = 100$



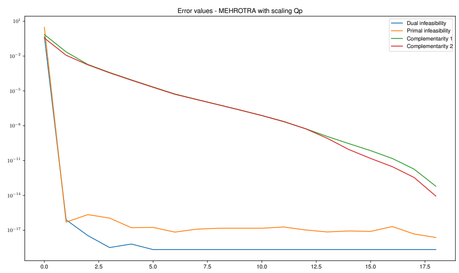

**(d)** $n = 351$

**Figure 2.3:** *Error values for Interior-point algorithm with Mehrotra scheme and scaling $Q_p$.*

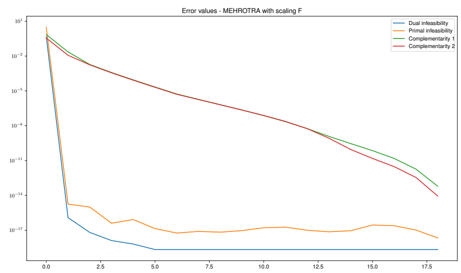
Through these figures, NT scaling technique works adequately for the rolling friction problem despite its slightly longer execution time. It is straightforward that the calculations for matrix $Q_p$ in which quadratic representation operators $Q_{\bar{p}}, Q_{\tilde{p}}$ and NT directions $\bar{p}, \tilde{p}$ satisfy (1.22) are so complex.

### 2.1.4 IP algorithm with predictor-corrector scheme and NT scaling technique using matrix F

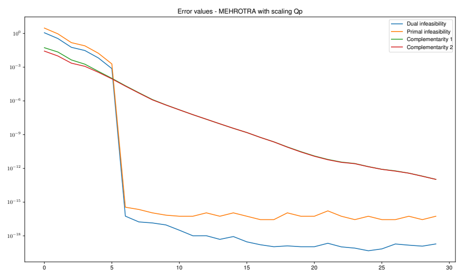
To accelerate Mehrotra algorithm with NT scaling technique, we replace the operator $Q_p$ by $F$ characterized by (1.33). Theoretically, $Q_p$ and $F$ are totally identical form the analytical point of view. However, explicit formulas of $F$ help us compute less than its predecessor. Figure 2.4 shows the results in pairs whereby we can produce a comparison between error values obtained by two scaling operators $Q_p$ and $F$. The figures on the left correspond to $Q_p$, the figures on the right correspond to the other.



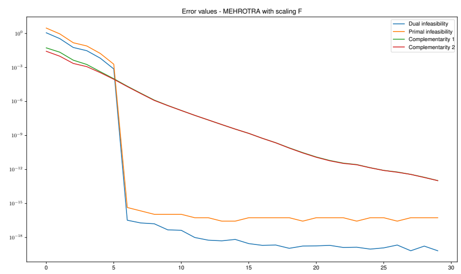

**(a)** $n = 2$, *NT scaling using matrix* $Q_p$




**(b)** $n = 2$, *NT scaling using matrix* $F$




**(c)** $n = 351$, *NT scaling using matrix* $Q_p$




**(d)** $n = 351$, *NT scaling using matrix* $F$

**Figure 2.4:** *Error values for Interior-point algorithm with Mehrotra scheme and scaling* $F$.

Taking a closer look at these figures, we hardly notice the obvious differences between the graphs. We also show a bit more detail about error values associated to the experiment $n = 351$ through iterations in order to illustrate the differences in the following table.

| ite | $Q_p$ | | | | $F$ | | | |
|---|---|---|---|---|---|---|---|---|
| | comp 1 | comp 2 | pinfeas | dinfeas | comp 1 | comp 2 | pinfeas | dinfeas |
| 0 | 5.37e-02 | 2.69e-02 | 3.00e+00 | 1.15e+00 | 5.37e-02 | 2.69e-02 | 3.00e+00 | 1.15e+00 |
| 1 | 2.23e-02 | 9.91e-03 | 9.22e-01 | 3.53e-01 | 2.23e-02 | 9.91e-03 | 9.22e-01 | 3.53e-01 |
| 2 | 4.41e-03 | 2.27e-03 | 1.52e-01 | 5.82e-02 | 4.41e-03 | 2.27e-03 | 1.52e-01 | 5.82e-02 |
| 3 | 1.84e-03 | 1.19e-03 | 7.92e-02 | 3.03e-02 | 1.84e-03 | 1.19e-03 | 7.92e-02 | 3.03e-02 |
| 4 | 4.26e-04 | 3.50e-04 | 1.73e-02 | 6.63e-03 | 4.26e-04 | 3.50e-04 | 1.73e-02 | 6.63e-03 |
| . | ... | ... | ... | ... | ... | ... | ... | ... |
| 25 | 8.32e-13 | 8.19e-13 | 2.78e-17 | 6.93e-20 | 8.32e-13 | 8.19e-13 | 5.55e-17 | 1.22e-19 |
| 26 | 6.01e-13 | 5.71e-13 | 2.78e-17 | 1.89e-19 | 6.01e-13 | 5.71e-13 | 2.78e-17 | 2.16e-19 |
| 27 | 3.77e-13 | 3.73e-13 | 5.55e-17 | 1.49e-19 | 3.72e-13 | 3.68e-13 | 5.55e-17 | 6.76e-20 |
| 28 | 2.06e-13 | 1.99e-13 | 2.78e-17 | 1.22e-19 | 2.05e-13 | 1.97e-13 | 5.55e-17 | 1.74e-19 |
| 29 | 1.06e-13 | 1.05e-13 | 5.55e-17 | 1.91e-19 | 1.05e-13 | 1.04e-13 | 5.55e-17 | 6.76e-20 |
| | Execution time: 17.825270 | | | | Execution time: 14.445220 | | | |

Table 2.1: Comparison of results obtained by NT scaling technique using operators $Q_p$ and $F$.

For convenience, we only specify the five first and five last iterations. For the first iterations, we observe that all error values are absolutely identical for both operators, meanwhile they become slightly different for the last ones. These distinctions come from errors in the calculations such as inverse, square root, etc since the formulas of $Q_p$ and $F$ are different. The distinction will be more pronounced when the matrix size is larger, and when it has gone through many iterations after which the accuracy is extremely remarkable. Moreover, the operator $F$ accelerate the NT scaling technique in comparison with the operator $Q_p$. We will take a closer look at this through performance profile in the next section. Besides, we applied an LU factorization to solve all the linear systems and we did not take the advantage of the symmetry of the matrix in (2.2). Indeed, (2.2) should be solved by means of an LDL factorization which would improve the numerical results related to the NT scaling strategy. The next stage after this intership will accomplish this thing.

## 2.2 Performance profiles

We now take a glance at an overall evalution for 4 algorithms with the tolerance $10^{-12}$ in the following table.

| SIP | MEHROTRA | MEHROTRA scaling $Q_p$ | MEHROTRA scaling $F$ | |
|---|---|---|---|---|
| 8358 | 3808 | 4801 | 4795 | Total iterations |
| 8.086672 | 5.853297 | 8.902764 | 7.034623 | Average execution time |

Table 2.2: Summary specification for 4 algorithms.

To be able to get the benchmarking optimization solvers, Dolan and Moré [6] introduced the performance profile by which we can evaluate our algorithms with the best features on a large dataset. Assume that we possess a set $P$ of $n_p$ problems which will be solved by a set $S$ of $n_s$ solvers. For each problem $p$, each solver $s$ and a required precision $tol$, we define a performance criterion by

$$t_{p,s} = \text{computing time required to solve problem } p \text{ by solver } s \text{ at precision } tol.$$

To compare the performance on problem $p$ by solver $s$ with the others solvers, we define a performance ratio by

$$r_{p,s} = \frac{t_{p,s}}{\min t_{p,s} : s \in S} \geq 1.$$

A cumulative distribution function $\rho_s$ for the performance ratio for a solver $s$ is defined by

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : r_{p,s} \leq \tau\} \leq 1,$$

for each $\tau \geq 1$. The parameter $\rho_s(\tau)$ is the probability for solver $s$ whose performance ratio $r_{p,s}$ is within a factor $\tau$ of the best solver. In other words, the higher $\rho_s(\tau)$ is, the better the algorithm is. We now present the performance profiles for 201 experiments with some different tolerances $10^{-12}$, $10^{-14}$, $10^{-16}$, $10^{-17}$ in the following Figures.

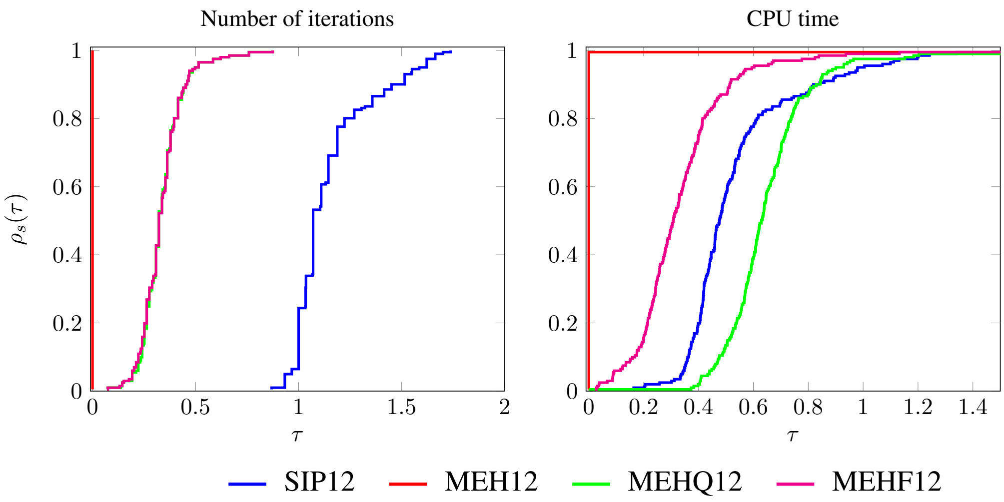



**Figure 2.5:** *Performance profile for* ***tol*** $= 10^{-12}$.

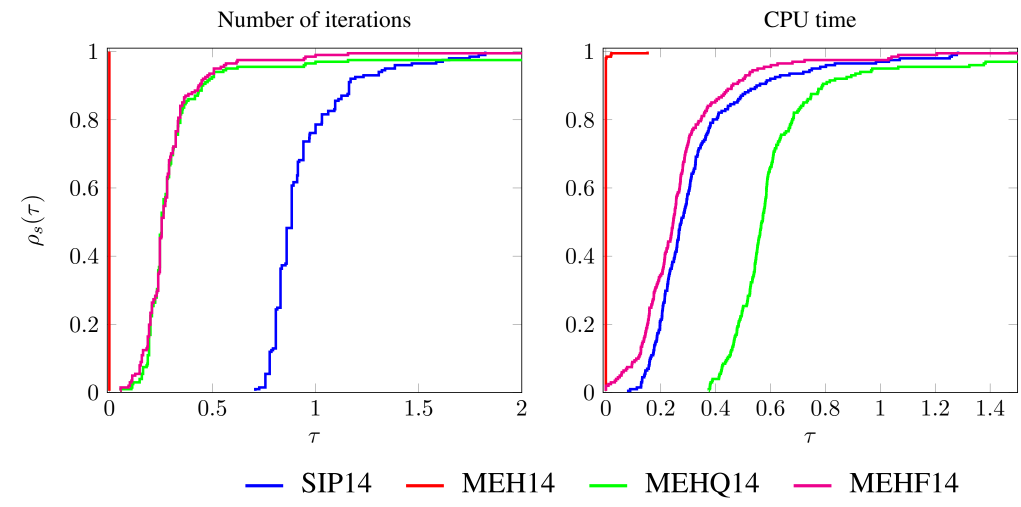



**Figure 2.6:** *Performance profile for* ***tol*** $= 10^{-14}$.

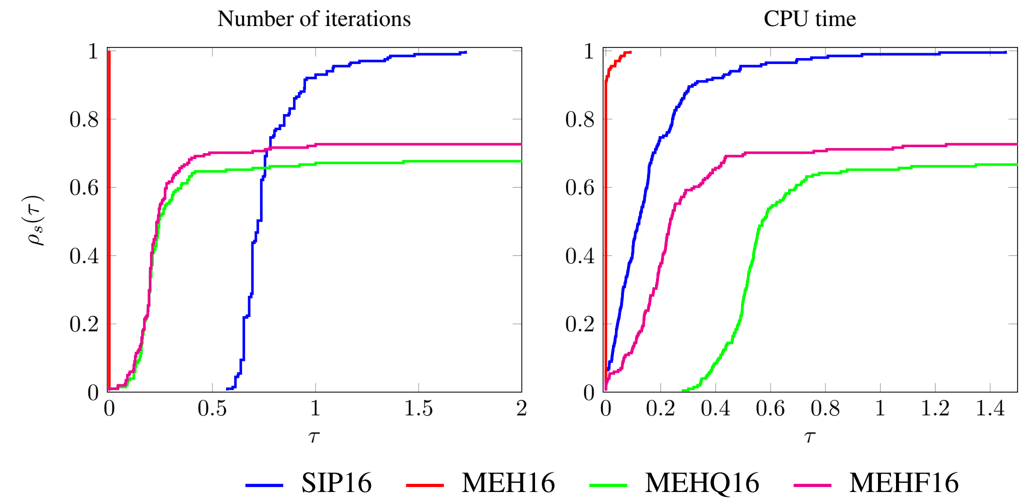



**Figure 2.7:** *Performance profile for* ***tol*** $= 10^{-16}$.

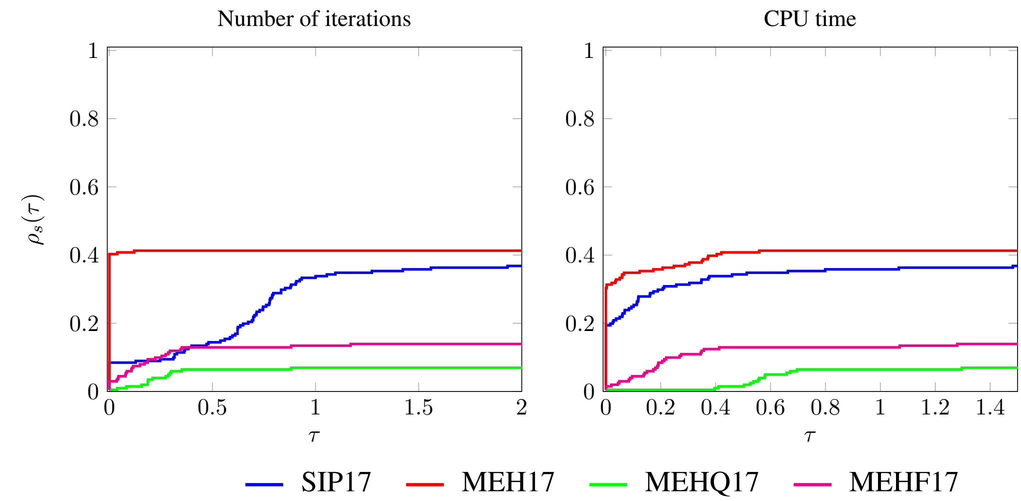



**Figure 2.8:** *Performance profile for* ***tol*** $= 10^{-17}$.

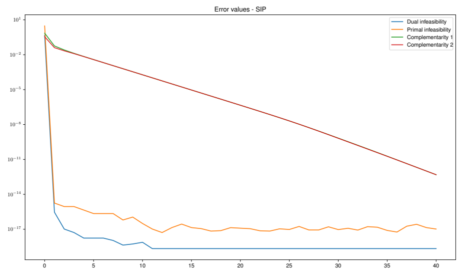
With these performance profiles, we observe that the original algorithm of Mehrotra (without scaling) has the highest performance. We also see that Mehrotra algorithm applying NT scaling technique reduces the speed of algorithms because of its computational complexity, the total number of iterations to reach the optimal solution is about 25% more than the one without scaling. In addition, using the operator $F$ for the NT scaling technique provides a 25% faster speedup than using the operator $Q_p$.

## 2.3 Jump phenomenon

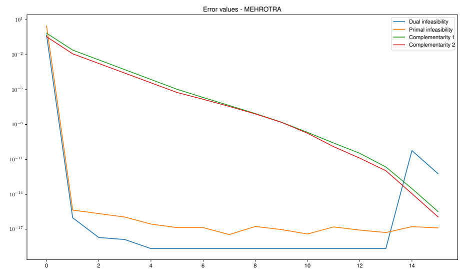
As mentioned in (1.3.4), we can see the $jump$ phenomenon for the dual infeasibility. In Figure 2.9, with the number of contact points $n = 2$, the $jump$ phenomenon has not appeared clearly (actually it appears only a little in the case of Mehrotra algorithm). As described in section 1.3.4, it is simply understandable that the number of contact points is insignificant, the step-lengths obtained from the calculation are all equal to 1 through iterations for almost algorithms. Thus, the term $(\alpha_{\text{D}} - \alpha_{\text{P}})M\Delta v$ disappears in the dual residual

$$\text{dinfeas}^{+} = (1 - \alpha_{\text{D}})\text{dinfeas} + \alpha_{\text{D}}\xi_{\text{D}} + (\alpha_{\text{D}} - \alpha_{\text{P}})M\Delta v,$$

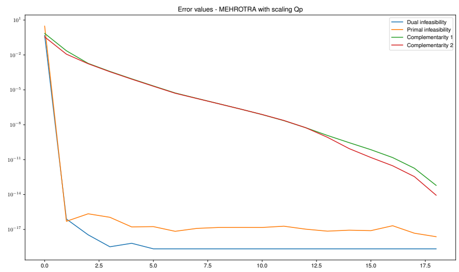
algorithms work smoothly as usual.



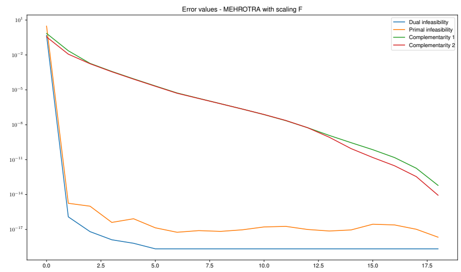

**(a)** $n = 2$, *SIP*




**(b)** $n = 2$, *MEHROTRA without scaling*




**(c)** $n = 2$, *MEHROTRA scaling* $Q_p$




**(d)** $n = 2$, *MEHROTRA scaling* $F$

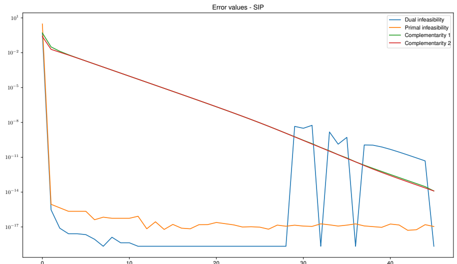
**Figure 2.9:** $jump$ *appears faint.*

We observe that the $jump$ appears in $2.9(b)$, only for the original Mehrotra's algorithm. It is explained that in the last iterations, Mehrotra's algorithm produced some different step-lengths $\alpha_{\text{P}}$ and $\alpha_{\text{D}}$. We also show detailed information in the following table.

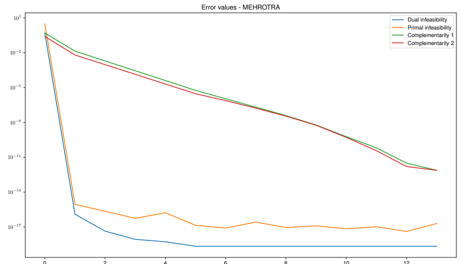
| ite | comp 1 | comp 2 | pinfeas | dinfeas | $\alpha_{\text{P}}$ | $\alpha_{\text{D}}$ |
|---|---|---|---|---|---|---|
| . | ... | ... | ... | ... | ... | ... |
| 11 | 2.54e-10 | 1.14e-10 | 1.36e-17 | 2.16e-19 | 1.00e+00 | 1.00e+00 |
| 12 | 3.31e-11 | 1.25e-11 | 1.91e-17 | 2.16e-19 | 1.00e+00 | 1.00e+00 |
| 13 | 2.19e-12 | 1.05e-12 | 1.41e-17 | 2.16e-19 | 1.00e+00 | 1.00e+00 |
| 14 | 2.99e-14 | 1.12e-14 | 1.08e-17 | 5.73e-11 | 9.94e-01 | 1.00e+00 |
| 15 | 3.15e-16 | 1.13e-16 | 1.99e-17 | 5.92e-13 | 9.90e-01 | 9.91e-01 |
| | Execution time: 0.103712 | | | | | |

Table 2.3: $jump$ occurs when $\alpha_{\text{P}}$ and $\alpha_{\text{D}}$ are different at the two last iterations.

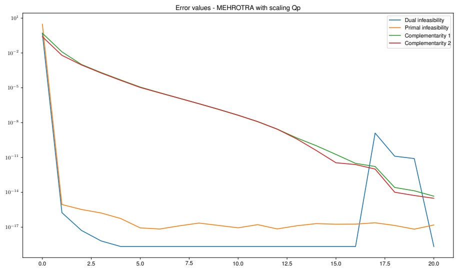
However, when the number of contact points $n$ inscrease, step-lengths are frequently distinct. The $jump$ appears as in figure 2.10 and even occurs densely as in figure 2.11.



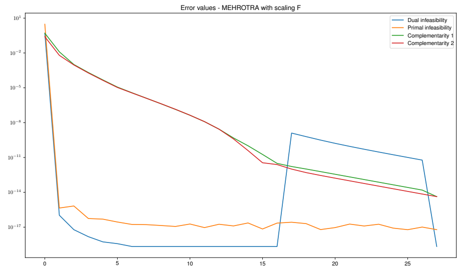

**(a)** $n = 4$, *SIP*




**(b)** $n = 4$, *MEHROTRA without scaling*




**(c)** $n = 4$, *MEHROTRA scaling* $Q_p$




**(d)** $n = 4$, *MEHROTRA scaling* $F$

**Figure 2.10:** *jump occurs clearly.*

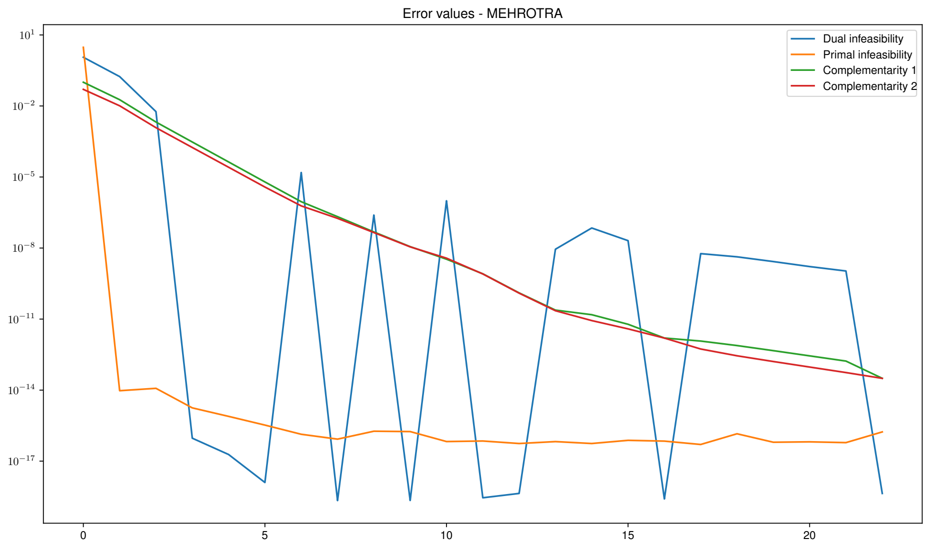


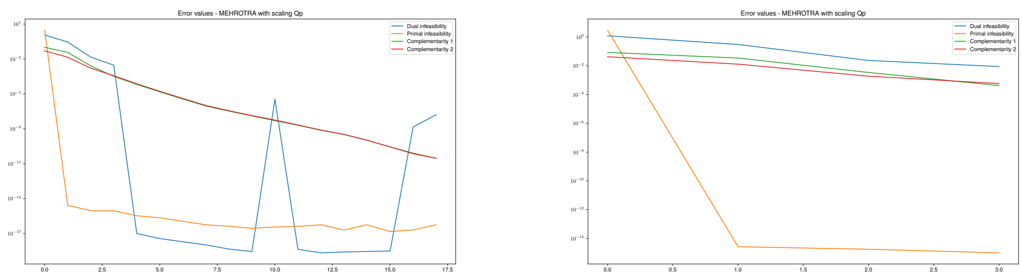

**Figure 2.11:** $n = 100$, *MEHROTRA - jump occurs with a dense frequency.*

Jumps appear more as the number of contact points increases. The following table corresponding to $n = 100$ by original Mehrotra's algorithm shows that the number of step-lengths varies greatly.

| ite | comp 1 | comp 2 | pinfeas | dinfeas | $\alpha_{\mathrm{P}}$ | $\alpha_{\mathrm{D}}$ |
|---|---|---|---|---|---|---|
| . | ... | ... | ... | ... | ... | ... |
| 5 | 6.19e-06 | 3.77e-06 | 3.33e-16 | 1.25e-18 | 1.00e+00 | 1.00e+00 |
| 6 | 9.06e-07 | 6.03e-07 | 1.36e-16 | 1.55e-05 | 9.91e-01 | 9.68e-01 |
| 7 | 2.11e-07 | 1.80e-07 | 8.48e-17 | 2.16e-19 | 1.00e+00 | 1.00e+00 |
| 8 | 4.78e-08 | 4.51e-08 | 1.84e-16 | 2.46e-07 | 1.00e+00 | 9.98e-01 |
| 9 | 1.14e-08 | 1.12e-08 | 1.77e-16 | 2.19e-19 | 1.00e+00 | 1.00e+00 |
| 10 | 3.39e-09 | 3.74e-09 | 6.72e-17 | 9.89e-07 | 1.00e+00 | 9.52e-01 |
| 11 | 8.16e-10 | 8.05e-10 | 7.08e-17 | 2.84e-19 | 1.00e+00 | 1.00e+00 |
| 12 | 1.28e-10 | 1.24e-10 | 5.55e-17 | 4.32e-19 | 1.00e+00 | 1.00e+00 |
| 13 | 2.39e-11 | 2.25e-11 | 6.68e-17 | 8.91e-09 | 9.94e-01 | 1.00e+00 |
| 14 | 1.53e-11 | 8.63e-12 | 5.55e-17 | 7.03e-08 | 7.19e-01 | 9.40e-01 |
| 15 | 6.03e-12 | 3.85e-12 | 7.56e-17 | 2.05e-08 | 8.58e-01 | 1.00e+00 |
| 16 | 1.58e-12 | 1.58e-12 | 7.01e-17 | 2.53e-19 | 1.00e+00 | 1.00e+00 |
| 17 | 1.18e-12 | 5.44e-13 | 5.05e-17 | 5.81e-09 | 7.16e-01 | 1.00e+00 |
| 18 | 7.61e-13 | 2.81e-13 | 1.43e-16 | 4.26e-09 | 6.25e-01 | 1.00e+00 |
| 19 | 4.64e-13 | 1.61e-13 | 6.28e-17 | 2.68e-09 | 6.11e-01 | 1.00e+00 |
| 20 | 2.79e-13 | 9.41e-14 | 6.53e-17 | 1.65e-09 | 6.08e-01 | 1.00e+00 |
| 21 | 1.68e-13 | 5.51e-14 | 6.06e-17 | 1.07e-09 | 6.07e-01 | 1.00e+00 |
| 22 | 3.15e-14 | 3.14e-14 | 1.72e-16 | 4.32e-19 | 1.00e+00 | 1.00e+00 |
| | Execution time: 2.861987 | | | | | |

Table 2.4: $\alpha_{\mathrm{P}}$ and $\alpha_{\mathrm{D}}$ are different many times.

We observe that the $jump$ not only slows down the convergence but as well causes the algorithm to fail because the points go directly to outside the cones after 16 iterations as in figure 2.12$(a)$, or almost instantaneously from the begining as in figure 2.12$(b)$.




**(a)** $n = 100$, *MEHROTRA scaling* $Q_p$ **(b)** $n = 140$, *MEHROTRA scaling* $Q_p$

**Figure 2.12:** *Failure from* $jump$.

# Conclusion

In the scope of this report, we have successfully solved the rolling friction problem related to the primal-dual model (5)-(6) (a simplified formulation) by means of interior-point algorithms, as well as effectively applied the NT scaling technique to improve the performance for the optimal solutions. We have also ironed out the most difficulty of this problem in which the cones are not self-dual by the change of variables introduced in remark (1.1.7). Moreover, we can generalize this approach to the case of a cone $F$ of the form

$$F = \left\{ x = (x_0; x_1; \dots; x_p) \in \mathbb{R} \times \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p} : \sum_{i=1}^{p} \|x_i\| \leq x_0 \quad \text{and} \quad \sum_{i=1}^{p} n_i = d-1 \right\}.$$

For the next stage of development after this topic,

1. we plan to implement the algorithms in C with a solution of the linear system by means of an LDL symmetric factorization,
2. and we will deal with the general rolling friction problem (3) in which the optimization problem is no more convex.

# Annex

This section is mainly following [4, §4]. We will take a glance at the Euclidean Jordan algebra which is a well-known framework associated with SOCP. For simplicity, we consider all vectors consisting of a single block, i.e. $x = (x_0; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$. For each such vector $x \in \mathbb{R}^n$, we define an $arrow - shaped$ matrix Arw($x$) given by

$$\text{arw}(x) = \begin{pmatrix} x_0 & \bar{x} \\ \bar{x}^\top & x_0 I \end{pmatrix}.$$

Note that $x \in \mathcal{L} = \{v = (v_0; \bar{v}) \in \mathbb{R} \times \mathbb{R}^{n-1} : \|\bar{v}\| \leq v_0\}$ (known as Lorentz cone) if and only if Arw($x$) is positive definite. In SOCP, it is useful to introduce the identity vector $e = (1; \mathbf{0}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the reflection matrix

$$R = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{pmatrix} \in \mathbb{R}^{n \times n},$$

and the direct sum $\oplus$ for two matrices $A$ and $B$

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

**Jordan product** Let us now define Jordan product of two vectors $x, y$:

$$x \circ y := \begin{pmatrix} x^\top y \\ x_0 y_1 + y_0 x_1 \\ \vdots \\ x_0 y_{n-1} + y_0 x_{n-1} \end{pmatrix} = \begin{pmatrix} x^\top y \\ x_0 \bar{y} + y_0 \bar{x} \end{pmatrix} = \text{arw}(x) y = \text{arw}(x)\text{arw}(y) e.$$

It is useful to know some relations involving the operation $\circ$.

1. Commmutative property $x \circ y = y \circ x$.
2. Unique identity element $x \circ e = x$.
3. Let $x^2 = x \circ x$. $\circ$ does not possess the assosiative property, i.e. $x \circ (y \circ z) \neq (x \circ y) \circ z$, but the power assosiative property $x^{p+q} = x^p \circ x^q$ for all $p, q \in \mathbb{N}$.

**Spectral decomposition** Every element in Jordan algebra can be factorized as spectral decomposition given by

$$x = \lambda_1 c_1 + \lambda_2 c_2,$$

where

- $\lambda_1 = x_0 + \|\bar{x}\|$, $\lambda_2 = x_0 - \|\bar{x}\|$ are eigenvalues,
- $c_1 = \frac{1}{2}\begin{pmatrix} 1 \\ \frac{\bar{x}}{\|\bar{x}\|} \end{pmatrix}$, $c_2 = \frac{1}{2}\begin{pmatrix} 1 \\ -\frac{\bar{x}}{\|\bar{x}\|} \end{pmatrix}$ are eigenvectors.

**Bacsic properties in Euclidean Jordan algebra** We also need the following definitions in the application of Jordan algebra.

1. $\mathrm{tr}(x) = \lambda_1 + \lambda_2 = 2x_0$,
2. $\det(x) = \lambda_1\lambda_2 = x_0^2 - \|\bar{x}\|^2$,
3. $x^{-1} = \lambda_1^{-1}c_1 + \lambda_2^{-1}c_2 = \frac{Rx}{\det(x)}$ if $\det(x) \neq 0$,
4. $x^{1/2} = \lambda_1^{1/2}c_1 + \lambda_2^{1/2}c_2$ for $x \in \mathcal{L}$,
5. $\nabla_x \ln\det(x) = 2x^{-1}$.

**Quadratic representation** Let us now introduce another vital linear transformation known as $quadratic - representation$ associated to $x$ given by

$$Q_x = 2\mathrm{arw}^2(x) - \mathrm{arw}(x^2) = \begin{pmatrix} \|x\|^2 & 2x_0\bar{x}^\top \\ 2x_0\bar{x} & \det(x)I + 2\bar{x}\bar{x}^\top \end{pmatrix} = 2xx^\top - \det(x)R.$$

For $x \in \mathbb{R}^n$ non-singular, $y \in \mathbb{R}^n$ and $t \in \mathbb{Z}$, we have the following properties

1. $Q_x y = 2x \circ (x \circ y) - x^2 \circ y = 2(x^\top y)x - \det(x)Ry$,
2. $Q_{x^t} = Q_x^t$,
3. $(Q_x y)^t = Q_{x^t} y^t$ if $x, y$ commute,
4. $Q_{Q_y x} = Q_y Q_x Q_y$.

**Multi-blocks vector** We let $i \in \{1, \dots, n\}$. If vectors $x, y$ consists of many blocks, i.e. $x = (x_1; \dots; x_n) \in \mathbb{R}^{nd}$, $x_i \in \mathbb{R}^d$ and $y = (y_1; \dots; y_n) \in \mathbb{R}^{nd}$, $y_i \in \mathbb{R}^d$, then

1. $x \circ y = (x_1 \circ y_1; \dots; x_n \circ y_n)$,
2. $x^{-1} = (x_1^{-1}; \dots; x_n^{-1})$,
3. $\mathrm{arw}(x) = \mathrm{arw}(x_1) \oplus \cdots \oplus \mathrm{arw}(x_n)$,
4. $Q_x = Q_{x_1} \oplus \cdots \oplus Q_{x_n}$.

# Bibliography


[1] Vincent Acary and Franck Bourrier. Coulomb friction with rolling resistance as a cone complementarity problem. *European Journal of Mechanics - A/Solids*, 85:104046, 2021.

[2] Vincent Acary, Maurice Brémond, and Olivier Huber. *On Solving Contact Problems with Coulomb Friction: Formulations and Numerical Comparisons*, pages 375–457. Springer International Publishing, Cham, 2018.

[3] Vincent Acary, Florent Cadoux, Claude Lemaréchal, and Jérôme Malick. A formulation of the linear discrete Coulomb friction problem via convex optimization. *ZAMM Z. Angew. Math. Mech.*, 91(2):155–175, 2011.

[4] F. Alizadeh and D. Goldfarb. Second-order cone programming. volume 95, pages 3–51. 2003. ISMP 2000, Part 3 (Atlanta, GA).

[5] Zhi Cai and Kim-Chuan Toh. Solving second order cone programming via a reduced augmented system approach. *SIAM Journal on Optimization*, 17(3):711–737, 2006.

[6] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[7] Osman Güler. Barrier functions in interior point methods. *Mathematics of Operations Research*, 21(4):860–885, 1996.

[8] Osman Güler. *Foundations of optimization*, volume 258. Springer Science & Business Media, 2010.

[9] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[10] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optim.*, 2(4):575–601, 1992.

[11] Yu E Nesterov and Michael J Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on optimization*, 8(2):324–364, 1998.

[12] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[13] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[14] James Renegar. *A mathematical view of interior-point methods in convex optimization*. SIAM, 2001.

[15] Takashi Tsuchiya. A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. *Optimization Methods and Software*, 11, 06 1998.

[16] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2, Ser. B):189–217, 2003.