Richard Andrášik

Nonlinear Rescaling Method and Self-concordant Functions

Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica, Vol. 52 (2013), No. 2, 5--19

Persistent URL: http://dml.cz/dmlcz/143533

Acta Univ. Palacki. Olomuc., Fac. rer. nat.,
Mathematica **52**, 2 (2013) 5–19

Nonlinear Rescaling Method and Self-concordant Functions

Richard ANDRÁŠIK

Department of Mathematical Analysis and Applications of Mathematics
Faculty of Science, Palacký University
17. listopadu 12, 771 46 Olomouc, Czech Republic
e-mail: andrasik.richard@gmail.com

(Received May 27, 2013)

Abstract

Nonlinear rescaling is a tool for solving large-scale nonlinear programming problems. The primal-dual nonlinear rescaling method was used to solve two quadratic programming problems with quadratic constraints. Based on the performance of primal-dual nonlinear rescaling method on testing problems, the conclusions about setting up the parameters are made. Next, the connection between nonlinear rescaling methods and self-concordant functions is discussed and modified logarithmic barrier function is recommended as a suitable nonlinear rescaling function.

Key words: convex optimization, nonlinear rescaling method, self-concordant functions

2010 Mathematics Subject Classification: 46N10, 47N10, 65K05

1 Introduction

The basic idea of nonlinear rescaling (NR) methods is a nonlinear transformation of constraint functions. Originally, the modified barrier methods (see [5]) were introduced along with few modified barrier functions. Afterwards, log-sigmoid function was also considered usable for NR and log-sigmoid multipliers method was described in [6] and [7]. Consequently, the pieces of knowledge were refined, put together and generalization of these techniques led to the concept of NR methods and NR functions. Similar to progress with interior-point methods, the primal-dual nonlinear rescaling (PDNR) method was developed (see [9]).

PDNR method is locally convergent with Q-linear convergence rate. PDNR method can be combined with another optimization method (e.g. primal-dual

path-following method) to obtain global convergence (see [9]). Another way how to improve convergence of PDNR method is dynamic scaling parameter update (see [10]) together with some globalization strategy (e.g. step length computation). This approach leads to the primal-dual nonlinear rescaling method with dynamic scaling parameter update (PDNRD). Recently, generalizations and other improvements (see [11] and [12]) were developed to improve asymptotic convergence rate and to reduce computational effort.

Several issues still remain for research in NR theory (see [9] and [11]). The first puprpose of this paper is to describe the parameters of PDNRD method. The second purpose is to give a recommendation about which NR function to use in computations.

The paper is organized as follows. First, the convex optimization problem is considered and basic assumptions are discussed. Then, NR functions are defined and the key idea of NR method is explained. Afterwards, PDNRD method is presented and its parameters are described. The connection between self-concordant functions and a modified logarithmic barrier function is discussed in Section 4 as a reason for using the modified logarithmic barrier function as NR function. Finally, numerical experiments with different parameter settings were made. The results are presented in Section 5.

2 Statement of the problem

We have the convex optimization problem

$$\begin{cases} \text{minimize } f(x),\ x \in \mathbb{R}^n, \\ \text{subject to } c_i(x) \geq 0,\ i = 1, \ldots, r. \end{cases} \tag{2.1}$$

Function f is convex and functions c_i are concave, $\forall i = 1, \ldots, r$. For simplicity we define mapping $c\colon \mathbb{R}^n \to \mathbb{R}^r$ as

$$c(x) = (c_1(x), c_2(x), \ldots, c_r(x))^{\mathrm{T}}, \quad \forall x \in \mathbb{R}^n.$$

We assume that:

(A) Functions f, c_i, $\forall i = 1, \ldots, r$, are at least twice continuously differentiable on the set $\mathbb{R}^n$.
(B) The optimal set $X^* = \operatorname{Argmin}\{f(x); x \in S\}$ is bounded and not empty.
(C) The Slater condition holds.

For problem (2.1) we define Lagrangian

$$L(x; \lambda) = f(x) - \sum_{i=1}^{r} \lambda_i c_i(x) \tag{2.2}$$

and we consider the dual problem

$$\begin{cases} \text{maximize } \theta(\lambda),\ \lambda \in \mathbb{R}^r, \\ \text{subject to } \lambda_i \geq 0,\ i = 1, \ldots, r, \end{cases} \tag{2.3}$$

where $\theta(\lambda) = \inf\limits_{x \in \mathbb{R}^n} L(x; \lambda)$ is the dual function.

Let $x^* \in X^*$. Due to assumption (C) Karush–Kuhn–Tucker's conditions can be used.

3 Nonlinear rescaling

3.1 Nonlinear rescaling functions

First, we define functions which are used to transform constraints of the problem (2.1).

Definition 3.1 Twice continuously differentiable function $\psi\colon (t_0; +\infty) \to \mathbb{R}$, where $-\infty < t_0 < 0$, satisfying conditions

(i) $\psi(0) = 0$, $\psi'(0) = 1$,
(ii) $\psi'(t) > 0$, $\forall t \in (t_0; +\infty)$,
(iii) $\psi''(t) < 0$, $\forall t \in (t_0; +\infty)$,
(iv) $\exists a > 0\colon \psi(t) \le -at^2$, $\forall t \in (t_0; 0)$,
(v) $\exists b > 0\colon \psi'(t) \le bt^{-1}$, $\forall t > 0$,
(vi) $\exists c > 0\colon \psi''(t) \ge -ct^{-2}$, $\forall t > 0$

is called *NR function*. The symbol Ψ denotes the set of NR functions.

For example, exponential transformation, modified logarithmic barrier function and modified hyperbolic barrier function are NR functions. These functions are defined by the following formulas:

$$\begin{aligned} \psi_1(t) &= 1 - \mathrm{e}^{-t} \\ \psi_2(t) &= \ln(t+1), \\ \psi_3(t) &= \frac{t}{t+1}. \end{aligned}$$

Functions ψ_i, $i = 2, 3$, can be modified so that $\psi_i \in \mathcal{C}^2(\mathbb{R})$. The function ψ_1 is already twice continuously differentiable and the following modification is not necessary, of course it can be done. The modification is realized by quadratic extrapolation of ψ_i for a given parameter $\tau \in (-1; 0)$ on the interval $(-\infty; \tau)$. We denote these extrapolated functions ψ_{q_i}, $i = 1, 2, 3$.

Few examples of NR functions were presented, but which to use in computations? Which NR function is the most suitable? We deal with this question in Section 4 from a theoretical point of view and in Section 5 from a practical point of view (computational efficiency).

3.2 Equivalent problem

NR methods are based on the idea to transform problem (2.1) using a function $\psi \in \Psi$ to the equivalent problem

$$\begin{cases} \text{minimize } f(x), \ x \in \mathbb{R}^n, \\ \text{subject to } k^{-1}\psi(kc_i(x)) \ge 0, \ i = 1, \dots, r, \end{cases} \tag{3.1}$$

From the definition of NR function, it is obvious that problems (2.1) and (3.1) have the same admissible sets and also the same optimal sets. Positive real number k is *scaling parameter*.

The Lagrangian for the equivalent problem (3.1) is given by formula

$$\mathcal{L}(x;\lambda,k) = f(x) - k^{-1}\sum_{i=1}^{r}\lambda_i\psi_i(kc_i(x)). \tag{3.2}$$

Suppose for a while that we know the solution of dual problem $\lambda^* \in \mathbb{R}^r_+$. Then it is sufficient to minimize the function $\mathcal{L}(x;\lambda^*,k)$ in primal variable x. Constrained optimization problem would be transformed to unconstrained optimization problem.

Since the Lagrange multipliers λ^* are not known, we estimate them and update them in every step of the method—just like the solution of the primal problem. In consequence, constrained optimization problem is converted to a sequence of unconstrained optimization problems. Newton's method or its variant is applied in each step to minimize the Lagrangian $\mathcal{L}$ in primal variable.

Algorithm 3.1 (Basic concept of NR methods)
Let $k > 0$ be a scaling parameter. An initial approximations $x^0 \in \mathbb{R}^n$ and $\lambda^0 \in \mathbb{R}^r_{++}$ are given. We suppose that approximation $(x^s,\lambda^s) \in \mathbb{R}^n \times \mathbb{R}^r_{++}$, $s \in \mathbb{N}_0$, is known already. We find next primal-dual pair (x^{s+1},λ^{s+1}) using the following formulas

$$\begin{aligned} &x^{s+1}\colon \nabla_x\mathcal{L}(x^{s+1};\lambda^s,k) = 0,\\ &\lambda_i^{s+1} = \psi'\left(kc_i(x^{s+1})\right)\lambda_i^s, \quad i = 1,\dots,r. \end{aligned} \tag{3.3}$$

The Algorithm 3.1 is well defined due to the following theorem.

Theorem 3.1 *For any given $(\lambda,k) \in \mathbb{R}^r_{++}\times\mathbb{R}_{++}$ there exists $\hat{x} \in \mathbb{R}^n$ such that*

$$\mathcal{L}(\hat{x};\lambda,k) = \min_{x\in\mathbb{R}^n}\mathcal{L}(x;\lambda,k).$$

Proof see [7, p. 206].

The main purpose of NR is to improve properties of Lagrangian. Classical Lagrangian L does not always work, because the existence of the unconstrained Lagrange minimizer is unknown in general. On the other hand, the unconstrained minimizer of the Lagrangian $\mathcal{L}$ always exists (according to Theorem 3.1). Moreover, NR dramatically sharpens the reaction of Lagrangian to the constraint violation, which has an impact on the computations.

3.3 Primal-dual nonlinear rescaling method with dynamic scaling parameter update

To obtain higher convergence rate of the method, we dynamically change the scaling parameter. Moreover, we use Newton's method with step length (e.g. backtracking line search algorithm) to solve the formulas (3.3).

We introduce a function which measures the distance between approximation (x, λ) and solution (x^*, λ^*).

Definition 3.2 Function $\nu\colon \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}_+$, defined as follows

$$\nu(x, \lambda) = \max\left\{\|\nabla_x L(x; \lambda)\|,\ -\min_{1 \le i \le r} c_i(x),\ \sum_{i=1}^{r} |\lambda_i c_i(x)|\right\}, \tag{3.4}$$

is called *the merit function*.

From first order optimality conditions it follows that

$$\nu(\hat{x}, \hat{\lambda}) = 0 \Leftrightarrow (\hat{x}, \hat{\lambda}) \in X^*.$$

The primal-dual system derived from (3.3) has the form (see [10])

$$\begin{bmatrix} \nabla^2_{xx} L(x; \bar{\lambda}) + \frac{1}{k^2}\mathbf{I}_n & -\nabla c(x)^{\mathrm{T}} \\ -k\Psi''\left(kc(x)\right)\Lambda \nabla c(x) & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x L(x; \bar{\lambda}) \\ 0 \end{bmatrix}, \tag{3.5}$$

where primal-dual pair (x, λ) is an approximation of solution, k is the scaling parameter and $\bar{\lambda} = \Psi'\left(kc(x)\right)\lambda$ is the dual predictor of Lagrange multipliers. We denote

$$N_k(\cdot) = \begin{bmatrix} \nabla^2_{xx} L(x; \bar{\lambda}) + \frac{1}{k^2}\mathbf{I}_n & -\nabla c(x)^{\mathrm{T}} \\ -k\Psi''\left(kc(x)\right)\Lambda \nabla c(x) & \mathbf{I}_r \end{bmatrix}.$$

Primal-dual system can be solved in two ways using techniques for sparse matrices or for positive definite matrices (see [9]).

Using dynamic scaling parameter update and backtracking line search algorithm we obtain a globally convergent method with 1.5-Q-superlinear convergence rate (see [10]) called PDNRD method. In every outer iteration, system (3.5) is solved. Next, it is tested whether the value of the merit function was decreased superlineary. If not, the primal Newton direction Δx is used to minimize $\mathcal{L}$ in primal variable. Therefore, it can happen that several Newton steps (inner iterations) are made during one outer iteration.

Algorithm 3.2 (Globally convergent PDNRD method)
An initial approximation $x^0 \in \mathbb{R}^n$ is given. An accuracy parameter $\varepsilon > 0$ and initial scaling parameter $k \in \mathbb{R}_{++}$ are given. Parameters $q \in (0; 1)$, $\eta \in (0; 0.5)$, $\omega > 1$, $\sigma > 0$ and $\theta > 0$ are also given. Set $x := x^0$, $\lambda := (1, 1, \ldots, 1) \in \mathbb{R}^r$ and $H := \nu(x, \lambda)$.

(1°) **If** $H \le \varepsilon$, **then** stop, **output** (x, λ).

(2°) Find $\bar{\lambda}$ and $(\Delta x, \Delta\lambda)$ from primal-dual system (3.5) and set

$$\hat{x} := x + \Delta x, \quad \hat{\lambda} := \bar{\lambda} + \Delta\lambda, \quad \hat{H} := \nu(\hat{x}, \hat{\lambda}).$$

(3°) **If** $\hat{H} \leq \min\left\{H^{3/2-\theta}, 1-\theta\right\}$, **then** set

$$x := \hat{x},\ \lambda := \hat{\lambda},\ H := \hat{H},\ k := \max\left\{\frac{1}{\sqrt{H}}, k\right\}$$

and go to step (1°).

(4°) Find $\alpha \in (0;1\rangle$ so that it holds

$$\mathcal{L}(x+\alpha\Delta x;\lambda,k) - \mathcal{L}(x;\lambda,k) \leq \eta\alpha\Delta x^{\mathrm{T}}\nabla_x\mathcal{L}(x;\lambda,k),$$

using the backtracking line search algorithm.

(5°) Set

$$x := x + \alpha\Delta x,\ \hat{\lambda} := \Psi'(kc(x))\lambda.$$

(6°) **If**

$$\|\nabla_x\mathcal{L}(x;\lambda,k)\| \leq \frac{\sigma}{k}\|\hat{\lambda}-\lambda\|,$$

then go to step (8°).

(7°) Find $(\Delta x, \Delta\lambda)$ from primal-dual system (3.5) and go to step (4°).

(8°) **If** $\nu(x,\hat{\lambda}) \leq qH$, **then** set

$$\lambda := \hat{\lambda},\ H := \nu(x,\hat{\lambda}),\ k := \max\left\{\frac{1}{\sqrt{H}}, k\right\}$$

and go to step (1°).

(9°) Set $k := \omega k$ and go to step (7°).

3.4 Parameters

When using numerical method the "right" setting of the parameters is very important. Experience with numerical experiments helps us to find this optimal setting. Based on testing PDNRD method on examples from section 5, we made considerations about suitable setting of PDNRD method parameters.

Factor ω

The factor ω affects the rate of scaling parametr increase. For $\omega \in \langle 5;20\rangle$ we obtain almost the same results. So we set $\omega = 10$. However, even for the other choices $\omega > 1$ there are not any significant changes. At most, it may happen that it takes a few extra steps of the method.

Parameter σ

How we choose parameter σ is less important then the choice of the ratio between σ and the initial choice of scaling parameter k_{init}. It is fraction $\frac{\sigma}{k}$ which decides about the number of inner iterations (and thus about the number of Newton steps in damped fase of Newton's method). It is clear that for $\sigma \gg k$ there are too little inner steps. On the other hand, for $\sigma \ll k$ there are too many of them. According to this the choice $\sigma = \frac{1}{2}k_{\text{init}}$ or $\sigma = \frac{1}{5}k_{\text{init}}$ is suitable.

Parameter θ

The parameter θ affects whether the inner solver runs or not. If we set $\theta = 0$, the inner solver runs whenever the 1.5-superlinear decrease of the merit function was not achieved. Choosing $\theta = 0.5$ we are saying that we are satisfied with only linear decrease of the merit function. Therefore, it is wise to set $\theta \in \langle 0; 0.5\rangle$. It appears (see Tab. 3 and 6) that the method does not depend on this parameter too much.

In condition, which is related to the parameter θ, the term $\min\left\{H^{3/2-\theta}, 1-\theta\right\}$ is calculated. We can ask why do not simply use the term $H^{3/2-\theta}$ instead of the previous one. As we know, the classical Newton method is effective only in neighbourhood of the solution, so if we are "far" from the solution the damped Newton method is better suited to use. The expression "far" means that the merit function is greater than $1-\theta$.

Factor q

The number of inner steps is influenced by fraction $\frac{\sigma}{k}$ together with the factor q. This factor also affects how often is the scaling parameter increased. Due to the way in which is this parameter used in PDNRD method, it is needed to set $q \in (0; 1)$. Moreover, we must choose the factor q so that the method do not use too many inner steps, because of long-lasting cumputation (see Tab. 3). Based on data in Tab. 3 and 6 we recommend not to set the factor q too small. The values from interval $\langle 0.5; 1)$ is a suitable choice and gives us comparable results.

Parameter η

The backtracking line search parameter η is usually chosen in the range from 0.01 to 0.3 (see [1, p. 466]). According to data in Tab. 1, 2 and 3, the choice of the parameter $\eta \in \langle 0.01; 0.3\rangle$ is arbitrary when solving the chord problem. But when the steel brick problem is solved it is better to set $\eta = 0.01$ (see Tab. 4, 5 and 6).

Initial value of scaling parameter

Based on the results shown in Tab. 4 and 5 it might seem that with increasing number of variables we should increase also the initial value of the scaling parameter. However, this hypothesis was not confirmed when solving the steel brick problem, because for $n = 648$ is the suitable setting $k_{\text{init}} = 2 \cdot 10^2$ (see Tab. 5). Anyway, for $k_{\text{init}} \in \left\langle 10^2; 10^5\right\rangle$ the solution is reached usually after an acceptable number of steps.

Initial approximation

The computation also depends on the initial approximation. The number of steps can be decreased, if a "lucky" initial approximation is chosen. In problems like beam deflection or contact problems we choose an initial state of the system as initial approximation, because the shape changes of body are usually very small. Thus we set $x^0 = (0, 0, \ldots, 0)$.

4 Connection between nonlinear rescaling functions and self-concordant functions

Self-concordance is affine invariant property (see Theorem 4.2) and convergence analysis of Newton's method for self-concordant functions does not depend on any unknown constants. In addition, self-concordant functions include many logarithmic functions (e.g. $-\psi_2$ and $-\psi_{q_2}$).

4.1 Self-concordant functions

We begin with definition of a self-concordant function.

Definition 4.1 Three times continuously differentiable convex function $f\colon \mathcal{D}(f) \to \mathbb{R}$, where $\mathcal{D}(f) \subseteq \mathbb{R}$, is *self-concordant* on $\mathcal{A} \subseteq \mathcal{D}(f)$, if the following inequality holds

$$|f'''(x)| \leq 2[f''(x)]^{\frac{3}{2}}, \quad \forall x \in \mathcal{A}. \tag{4.1}$$

Definition 4.2 The function $f\colon \mathcal{D}(f) \to \mathbb{R}$, where $\mathcal{D}(f) \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$, $n > 1$, is *self-concordant*, if the function $\hat{f}(t) = f(x_0 + t\alpha)$ is self-concordant for any $\alpha \in \mathbb{R}^n$ such that x_0, $x_0 + t\alpha \in \mathcal{D}(f)$.

Remark 4.1 Obviously, linear functions and convex quadratic functions are self-concordant, because its second derivative is non-negative and its third derivative is identically equal to zero.

Self-concordant functions have several important properties, which can be used to decide whether a given function is self-concordant or not. We use these properties in section 4.3.

The symbol $\mathcal{S}^n$, $n \in \mathbb{N}$, denotes the set containing self-concordant functions $f\colon \mathcal{D}(f) \to \mathbb{R}$, where $\mathcal{D}(f) \subseteq \mathbb{R}^n$. It holds that

$$\mathcal{S}^n \subset \left\{ f \in \mathcal{C}^3(\mathcal{D}(f)),\ \mathcal{D}(f) \subseteq \mathbb{R}^n \right\}.$$

Theorem 4.1 *The set $\mathcal{S}^n$, $\forall n \in \mathbb{N}$, is closed under addition and under scaling by a factor exceeding one.*

Proof see [1, p. 499]. □

Corollary 4.1 *Let $f(x) \in \mathcal{S}^n$, $g(y) \in \mathcal{S}^m$ (m, $n \in \mathbb{N}$) with domains $\mathcal{D}(f)$, $\mathcal{D}(g)$. Then the function $h(x, y) = f(x) + g(y)$ is self-concordant on the set $\mathcal{D}(f) \times \mathcal{D}(g) \subseteq \mathbb{R}^{n+m}$.*

Proof The functions $f(x)$, $g(y)$ can be viewed as functions of $(n+m)$-variables. Hence the statement is direct consequence of Theorem 4.1. □

Theorem 4.2 *Let $f(x) \in \mathcal{S}^n$, where $n \in \mathbb{N}$. Suppose that $\mathbf{A} \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Then the function $f(\mathbf{A}x + b)$ is self-concordant. In other words, the property "self-concordance" is affine invariant.*

Proof It is sufficient to prove the statement only for $n = 1$ (see definition 4.2). Assume that the function $f\colon \mathcal{D}(f) \to \mathbb{R}$, where $\mathcal{D}(f) \subseteq \mathbb{R}$ is self-concordant. We show that also function $\hat{f}(x) = f(ax + b) \in \mathcal{S}^1$, where $a,\ b \in \mathbb{R},\ a \neq 0$. It holds that

$$\hat{f}''(x) = a^2 f''(ax + b), \qquad \hat{f}'''(x) = a^3 f'''(ax + b).$$

Moreover, the following inequality holds

$$\left|a^3 f'''(ax + b)\right| \leq 2\left[a^2 f''(ax + b)\right]^{\frac{3}{2}},$$

and thus the function $\hat{f}(x)$ is self-concordant. □

4.2 Newton's method for self-concordant functions

Assume that $f\colon \mathcal{D}(f) \to \mathbb{R}$, where $\mathcal{D}(f) \subseteq \mathbb{R}^n$ is a self-concordant function with minimizer $x^* \in \mathcal{D}(f)$. We analyze Newton's method for the function f. The upper bound on the number of Newton steps is already known, so it can be simply computed.

Theorem 4.3 *Let η and β be the line search parameters. An accuracy parameter $\varepsilon > 0$ is given. The total number of Newton steps is less than*

$$\frac{20 - 8\eta}{\eta\beta(1 - 2\eta)^2}(f(x_0) - f(x^*)) + \log_2(-\log_2 \varepsilon). \tag{4.2}$$

Proof see [1, p. 503]. □

The value of parameter $\beta \in (0; 1)$ determines how precise will be the estimate of the step length. In practice, values between 0.1 (crude) and 0.8 (more precise) are used (see [1]). The parameter $\alpha \in (0; 0.5)$ is usually ranges from 0.01 to 0.3 (see [1]). For example, in Section 4.3 we set $\eta = 0.01$, $\beta = 0.8$ and $\varepsilon = 10^{-6}$. With these values we obtain the upper bound

$$2592.7(f(x_0) - f(x^*)) + 4.3$$

This estimate shows us, what can happen in the worst case. The numerical experiments shows (see [1]) that the expression $f(x_0) - f(x^*) + \log_2(-\log_2 \varepsilon)$ is a good guess of the Number of Newton steps.

4.3 Application of self-concordant functions in NR theory

The functions ψ_2 and ψ_{q_2} are NR functions (see Section 3.1). From Definition 4.1, it follows that the negative of these two functions are self-concordant functions. We can use this property in NR theory. Suppose that we deal with a convex optimization problem using NR method. If we find out that the Lagrangian $\mathcal{L}$ for the equivalent problem, or its positive multiple, is self-concordant, we can use Theorem 4.3 for better analysis of our problem. We

show how to do it on the following quadratic programming problem with linear and quadratic constraints.

From the finite element approximation of contact problems of linear elasticity with friction in three space dimensions arise minimization problem

$$\begin{cases} \text{minimize } \frac{1}{2}x^{\mathrm{T}}\mathrm{A}x - x^{\mathrm{T}}b,\ x \in \mathbb{R}^n \\ \text{subject to } g_i^2 - x_{i+m}^2 - x_{i+2m}^2 \geq 0,\ i = 1, \ldots, m, \\ \qquad\qquad x_i - l_i \geq 0,\ i = 1, \ldots, m, \end{cases} \tag{4.3}$$

where $n = 3m$ is the number of variables, $\mathrm{A} \in \mathbb{R}^{n \times n}$ is symmetric and positive definite, $b \in \mathbb{R}^n$, $g \in \mathbb{R}^m_+$, $l \in \mathbb{R}^m$. This is a convex programming problem so we can use NR approach to solve it.

We transform the inequality constraints of the problem (4.3) into an equivalent set of constraints using the function ψ_2. The Lagrangian for the equivalent problem is defined by the following formula

$$\mathcal{L}(x; \lambda, k) = \frac{1}{2}x^{\mathrm{T}}\mathrm{A}x - x^{\mathrm{T}}b - k^{-1}\sum_{i=1}^{2m} \lambda_i \psi_2\left(kc_i(x)\right), \tag{4.4}$$

where

$$\begin{aligned} c_i(x) &= x_i - l_i,\ i = 1, \ldots, m, \\ c_{i+m}(x) &= g_i^2 - x_{i+m}^2 - x_{i+2m}^2,\ i = 1, \ldots, m. \end{aligned}$$

Suppose that arbitrary $\lambda \in \mathbb{R}^r_{++}$ and $k > 0$ are given. We prove that the Theorem 4.3 hold true when applied to the Lagrangian (4.4). First, we will show that the function

$$\hat{\mathcal{L}}(x; \lambda, k, r) := kr^{-1}\mathcal{L}(x; \lambda, k),$$

where $r := \min\{\lambda_i; \lambda_i \neq 0,\ i = 1, \ldots, 2m\}$, is self-concordant. The function $\hat{\mathcal{L}}$ has three parts, namely:

1. $kr^{-1}\left(\frac{1}{2}x^{\mathrm{T}}\mathrm{A}x - x^{\mathrm{T}}b\right)$,
2. $-r^{-1}\sum\limits_{i=1}^{m} \lambda_i \ln\left(k(x_i - l_i) + 1\right)$,
3. $-r^{-1}\sum\limits_{i=1}^{m} \lambda_{i+m} \ln\left(k(g_i^2 - x_{i+m}^2 - x_{i+2m}^2) + 1\right)$.

The first part is a quadratic function, therefore it is a self-concordant function (see Remark 4.1) for any $r > 0$.

From Definition 4.1, it follows that $-\ln t$ is self-concordant. Also the function $-\ln\left(k(x_i - l_i) + 1\right) \in \mathcal{S}^1$ (see Theorem 4.2). We scale this function by a factor $-r^{-1}\lambda_i \geq 1$ and the result $-r^{-1}\lambda_i \ln\left(k(x_i - l_i) + 1\right) \in \mathcal{S}^1$ (according to Theorem 4.1). This process can be done for all $i = 1, \ldots, m$. At last, the sum of self-concordant functions is again a self-concordant function (see Corollary 4.1), therefore the second part of $\hat{\mathcal{L}}$ belongs to the set $\mathcal{S}^m$.

To prove that the third part is also a self-concordant function, we use the following lemma and then analogical procedure as for the second part of the function $\hat{\mathcal{L}}$.

Lemma 4.1 *The function*

$$f(x) = -\ln\left(x^{\mathrm{T}}\mathrm{P}x + Q^{\mathrm{T}}x + R\right),$$

where $\mathrm{P} \in \mathbb{R}^{n\times n}$ *is negative definite matrix,* $Q \in \mathbb{R}^n$ *and* $R \in \mathbb{R}$, $n \in \mathbb{N}$, *is self-concordant on its domain.*

Proof see [1, p. 500]. □

Finally, $\hat{\mathcal{L}}$ is a self-concordant function according to Corollary 4.1 as the sum of three self-concordant function. Now, we can apply the Theorem 4.3 to the function $kl^{-1}\mathcal{L}(x;\lambda,k)$. According to the fact that Newton's minimization sequences are the same for a given function and its positive multiple, the same statement remain true for the original function $\mathcal{L}$.

Similar analysis can be made for many other problems (e.g. linear and quadratic programming), because the function $-\ln(kc(x)+1)$ is self-concordant for a wide enough class of functions $c(x)$.

5 Numerical experiments

PDNRD method was tested on two problems like (4.3)—the chord problem and the steel brick problem. All computations were performed in MATLAB on PC Intel Pentium (1.7 GHz) with 1 GB RAM. In tables below we report the number of iterations, the number of solutions of the primal-dual system and the solution time in seconds. If some data is missing, it means that the solution time was too long in comparison to other cases in table.

Function ψ_{q_2} was used for NR of the problems to show that ψ_{q_2} is appropriate NR function not only from theoretical point of view, but also from the practical point of view.

5.1 Chord problem

We consider a problem

$$\min_{u\in\mathcal{K}} \mathcal{J}(u), \tag{5.1}$$

where

$$\mathcal{J}(u) = \frac{1}{2}\int_0^1 \|u'(t)\|^2 \mathrm{d}t - \int_0^1 u(t)^{\mathrm{T}} f(t)\,\mathrm{d}t,$$

$$\mathcal{K} = \left\{u \in \left(H_0^1(0;1)\right)^2 : u_2(t) \geq 0, \forall t \in (0;0.5),\ \|u(t)\| \leq 1.4, \forall t \in (0.5;1)\right\},$$

$$f(t) = \left(36\pi^2 \sin 6\pi t, -4\pi^2 \sin 2\pi t\right)^{\mathrm{T}}.$$

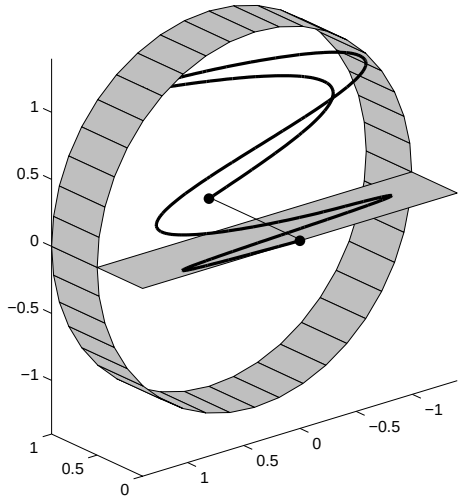

Figure 1: The chord deformation.

Minimization problem (5.1) describes loaded chord fixed at the endpoints that is partially above the plain and partially inside the cylindrical tube (see Fig. 1). Function $u(t)$ is the chord deflection.

The objective function $\frac{1}{2}x^{\mathrm{T}}\mathrm{A}x - x^{\mathrm{T}}b$, $x \in \mathbb{R}^n$ matches the convex quadratic functional $\mathcal{J}(u)$, linear constraints in problem (4.3) matches constraint $u_2(t) \geq 0, \forall t \in (0; 0.5)$ from the definition of the set $\mathcal{K}$ and quadratic constraints matches $\|u(t)\| \leq 1.4$, $\forall t \in (0.5; 1)$.

Solution: We solve the chord problem for different settings of parameters of PDNRD method. The main result from Tab. 1 and 2 is non-increasing (rather decreasing) number of iterations and number of solutions of primal-dual system while increasing the number of variables. The best choice for k_{init} is $2 \cdot 10^5$ in this case.

$\eta = 0.01$		k_{init}			
n	r	$2 \cdot 10^2$	$2 \cdot 10^3$	$2 \cdot 10^4$	$2 \cdot 10^5$
64	32	8/13/0.359	6/6/0.063	7/12/0.109	6/14/0.203
128	64	9/21/0.547	7/10/0.328	7/12/0.234	6/12/0.219
256	128	–	6/19/1.515	–	4/10/0.532
512	256	–	–	5/10/3.156	4/12/3.094
1024	512	–	–	5/13/19.890	3/6/9.578
2048	1024	–	–	–	4/7/62.375
4096	2048	–	–	–	4/9/553.481

Table 1: PDNRD method with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\mathrm{init}}$, $\theta = 0.4$, $q = 0.5$, $\eta = 0.01$, $\varepsilon = 10^{-6}$. The chord problem.

$\eta = 0.3$		k_{init}			
n	r	$2 \cdot 10^2$	$2 \cdot 10^3$	$2 \cdot 10^4$	$2 \cdot 10^5$
64	32	8/13/0.375	6/6/0.078	7/12/0.156	6/14/0.187
128	64	9/20/0.578	7/10/0.218	7/12/0.219	6/12/0.282
256	128	–	6/19/1.297	–	4/10/0.437
512	256	–	–	5/10/3.047	4/12/3.141
1024	512	–	–	5/13/20.469	3/6/9.063
2048	1024	–	–	–	4/7/63.484
4096	2048	–	–	–	4/9/551.516

Table 2: PDNRD method with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\text{init}}$, $\theta = 0.4$, $q = 0.5$, $\eta = 0.3$, $\varepsilon = 10^{-6}$. The chord problem.

	$\eta = 0.01$		$\eta = 0.3$	
q	$\theta = 0.1$	$\theta = 0.4$	$\theta = 0.1$	$\theta = 0.4$
0.1	–	–	–	–
0.5	3/6/9.094	3/6/9.078	3/6/9.109	3/6/9.078
0.9	3/6/9.094	3/6/9.079	3/6/9.125	3/6/9.109

Table 3: PDNRD with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\text{init}}$, $k_{\text{init}} = 2 \cdot 10^5$, $\varepsilon = 10^{-6}$. The chord problem ($n = 1024$).

From Tab. 3 it is obvious that choice $q = 0.1$ is the worst one. Parameters η and θ have no significant impact on the computation in this case.

5.2 Steel brick problem

The steel brick problem is also the problem of type (4.3). Let us consider a steel brick lying on the rigid obstacle. The brick occupies the domain $\mathcal{S} = (0;3) \times (0;1) \times (0;1)$. Boundary $\partial\mathcal{S}$ is divided into three parts

$$\Gamma_u = \{0\} \times (0;1) \times (0;1), \quad \Gamma_c = (0;3) \times (0;1) \times \{0\}, \quad \Gamma_p = \partial S \backslash \left(\bar{\Gamma}_u \cup \bar{\Gamma}_c\right),$$

on which are different boundary conditions (see Fig. 2). The problem is described in detail in [3].

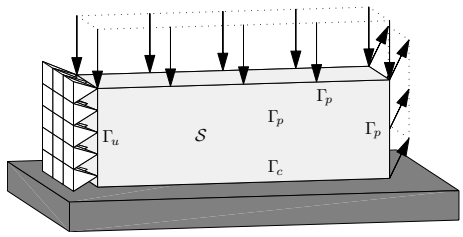

Figure 2: The steel brick.

Solution: The progress is similar to previos one in Section 5.1. We solve the steel brick problem for different choices of n. According to data in Tab. 4 and 5, the choice $\eta = 0.01$ is better then $\eta = 0.3$. The best setting for initial values of scaling parameter is $2 \cdot 10^3$ and $2 \cdot 10^4$ in this problem.

$\eta = 0.01$		k_{init}			
n	r	$2 \cdot 10^2$	$2 \cdot 10^3$	$2 \cdot 10^4$	$2 \cdot 10^5$
54	36	–	5/85/1.578	9/959/9.391	–
90	60	–	6/47/1.031	6/329/5.047	–
180	120	–	5/31/2.328	8/641/27.718	–
324	216	–	7/58/15.469	5/105/21.234	–
648	432	8/32/50.000	6/30/53.563	6/74/113.454	8/501/800.297

Table 4: PDNRD method with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\text{init}}$, $\theta = 0.4$, $q = 0.5$, $\eta = 0.01$, $\varepsilon = 10^{-6}$. The steel brick problem.

$\eta = 0.3$		k_{init}			
n	r	$2 \cdot 10^2$	$2 \cdot 10^3$	$2 \cdot 10^4$	$2 \cdot 10^5$
54	36	7/33/0.531	5/82/1.328	11/1167/9.578	–
90	60	–	6/39/0.969	6/340/5.640	–
180	120	–	7/48/3.328	10/717/39.594	–
324	216	–	–	5/116/25.922	–
648	432	8/33/59.484	–	6/106/145.313	9/609/891.375

Table 5: PDNRD method with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\text{init}}$, $\theta = 0.4$, $q = 0.5$, $\eta = 0.3$, $\varepsilon = 10^{-6}$. The steel brick problem.

	$\eta = 0.01$		$\eta = 0.3$	
q	$\theta = 0.1$	$\theta = 0.4$	$\theta = 0.1$	$\theta = 0.4$
0.1	5/105/23.328	5/105/23.312	5/116/25.812	5/116/25.735
0.5	5/105/24.141	5/105/23.984	5/116/25.610	5/116/25.984
0.9	5/105/23.203	5/105/23.437	5/116/25.734	5/116/25.781

Table 6: PDNRD method with parameters $\omega = 10$, $\sigma = \frac{1}{2}k_{\text{init}}$, $k_{\text{init}} = 2 \cdot 10^4$, $\varepsilon = 10^{-6}$. The steel brick problem ($n = 324$).

Results from Tab. 6 supports that the choice $\eta = 0.01$ is better then $\eta = 0.3$. Parameter θ has no significant impact on the computation.

6 Conclusion

In both examples, the outstanding step length shortening occured from certain moment for some parameter choices. As a consequence, this led to very high number of primal-dual system solutions, which slows down the method. It remains for further research, whether is this effect predictable or not.

MATLAB function `mldivide` was used to solve primal-dual systems. These systems were solved in two ways (see [9]) using sparse matrices (the chord problem) or positive definite matrices (the steel brick problem).

The connection between NR functions and self-concordant functions was studied. We considered a quadratic programming problem with linear and quadratic constraints and the respective equivalent problem. It was shown that there exists a positive multiple of Lagrangian of the equivalent problem such that it has the self-concordant property.

The parameters of PDNRD method were described and the recommendations about setting these parameters were made.

The solution of the primal-dual system is the most expensive operation, thus the number of solutions of the primal-dual system determines the total complexity of the computation. It was found out that the increasing number of variables in both presented problems has not a consequence in the increasing number of solutions of the primal-dual system. This fact supports the applicability of PDNRD method on problems of arbitrary size.

References

[1] Boyd, S., Vandenberghe, L.: Convex Optimization. *Cambridge University Press*, Cambridge, 2004.

[2] Griva, I., Nash, S. G., Sofer, A.: Linear and Nonlinear Optimization. Second edition, *SIAM*, Philadelphia, 2009.

[3] Kučera, R., Machalová, J., Netuka, H., Ženčák, P.: *An interior-point algorithm for the minimization arising from 3D contact problems with friction*. Optimization Methods and Software, (2013), in press.

[4] Nocedal, J., Wright, S. J.: Numerical Optimization. Second edition, *Springer*, New York, 2006.

[5] Polyak, R.: *Modified barrier functions (theory and methods)*. Mathematical Programming **54** (1992), 177–222.

[6] Polyak, R.: *Log-Sigmoid Multipliers Method in Constrained Optimization*. Annals of Operations Research **101** (2001), 427–460.

[7] Polyak, R.: *Nonlinear rescaling vs. Smoothing Technique in Convex Optimization*. Mathematical Programming **92A** (2002), 197–235.

[8] Polyak, R.: *Nonlinear Rescaling as Interior Quadratic Prox Method in Convex Optimization*. Computational Optimization and Applications **35** (2006), 347–373.

[9] Polyak, R., Griva I.: *Primal-Dual Nonlinear Rescaling Method for Convex Optimization*. JOTA **122**, 1 (2004), 111–156.

[10] Polyak, R., Griva, I.: *Primal-Dual Nonlinear Rescaling Method with Dynamic Scaling Parameter Update*. Mathematical Programming **106A** (2006), 237–259.

[11] Polyak, R., Griva, I.: *1.5-Q-superlinear convergence of an exterior-point method for constrained optimization*. Journal of Global Optimization **40**, 4 (2008), 679–695.

[12] Polyak, R., Griva, I.: *Proximal Point Nonlinear Rescaling Method for Convex Optimization*. Numerical Algebra, Control and Optimization **1**, 2 (2011), 283–299.

[13] Polyak, R., Teboulle, M.: *Nonlinear Rescaling and Proximal-Like Methods in Convex Optimization*. Mathematical programming **76** (1997), 265–284.