VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Development of Algorithms for Solving Minimizing Problems with Convex Quadratic Function on Special Convex Sets and Applications

DISSERTATION THESIS

Author: Ing. Lukáš Pospíšil
Supervisor: Prof. RNDr. Zdeněk Dostál, DSc.

2015

I declare I elaborated this thesis by myself. All literary sources and publications I have used had been cited.

Ostrava, June 18, 2015 .....

Acknowledgements

First of all I would like to thank my supervisor Prof. RNDr. Zdeněk Dostál, DSc. for his patience, help, support, and leadership during my doctoral studies, also many thanks to Ing. David Horák, PhD. for helping me with the beginning of study of optimization methods and for many very interesting discussions not only about mathematics. The final form of the thesis was consulted with Ing. Marta Jarošová, PhD. and Prof. Ing. Tomáš Kozubek, PhD.

A lot of ideas in this thesis came during the preparation of my lectures of Linear Algebra, Mathematical Analysis, and Numerical Methods. I want to thank to doc. RNDr. Jiří Bouchala, PhD. and the Department of Applied Mathematics at VŠB-Technical University of Ostrava for the opportunity to participate on teaching and taking lectures. Furthermore, I want to thank to doc. Mgr. Vít Vondrák, PhD. and all my students for inspiration and motivation to deeper study of the problematics.

My research was financed from many sources. Nowadays, I participate on the project IT4Innovations Centre of Excellence. The main merit on my employment has Prof. Ing. Tomáš Kozubek, PhD., the head of our Research Team. I want to thank all team members for keeping right team atmosphere. I really love my work, but this work started to be real fun right after I have started to cooperate with the Permon team. We are implementing and developing a new algorithms together and helping to each other. I want to thank to Ing. Václav Hapla (PhD.), (doc.) Ing. David Horák, PhD., Ing. Martin Čermák, PhD., Ing. Alexandros Markopoulos, PhD., and Ing. Alena Vašatová (PhD.). I will be a happy researcher if I have an opportunity to continue this cooperation.

I have started to be interested in particle dynamics, one of the main topics of the thesis, during my internship in SBEL University of Wisconsin-Madison. I want to gratefully thank to Prof. Dan Negrut, PhD. and his team. I hope that our cooperation will continue in the future.

I was always mostly inspired by the nearest collegues/friends from *my* office. They helped me a lot with the motivation to finish the thesis. They also tried to help me with my attempt to understand the academical world around me. Namely, I want to thank to Ing. Petr Kotas (PhD.), Ing. Pavla Hrušková, PhD., Ing. Jan Kracík, PhD., doc. Ing. Dalibor Lukáš, PhD., Ing. Kateřina Janurová, PhD., and Ing. Martin Hasal (PhD.).

During my studies, I lost many of my friends, mostly because of the time deficit to stay in touch with them. Nevertheless, the best ones kept. I gratefully want to thank Ing. Michal Prílepok, Ing. Jana Blinkalová, Ing. Lukáš Halagačka, PhD., and Ing. Lenka Šrubařová.

My family, the dearest ones, always stands behind my success. Nothing of my life can be possible without their support. They always know how to encourage me to work, to study, to live. My family is large and they know that I love them. At least I want to thank to my brothers - RNDr. Michal Pospíšil, PhD. and Matýsek.

And, of course, the greatest thanks belongs to the most important person in my life - my mum.

My research has been supported by

- the grant GA CR 201/07/0294,
- the grant GD103/09/H078 of Grant Agency of the Czech Republic,
- the Ministry of Education of the Czech Republic No. MSM6198910027,
- the VŠB-TUO SGS projects SP2011/183, SP2012/187, SP2013/191, SP2013/191, SP2014/204,
- IT4Innovations Center of Excellence, reg. no. CZ.1.05/1.1.00/02.0070.

Abstract

The thesis focuses on solving the optimization problems of a minimization a convex quadratic function on a special convex set. Such problems appear in many engineering applications, e.g., in the solution of contact problems of elasticity or in particle dynamics simulations. The number of unknows in these practical problems usually exceeds the potential of sequential algorithms. In the thesis, we present iterative methods which can be easily parallelizable.
The text is divided into three parts; the review of the basic quadratic programming theory, the algorithms for solving the problem, and the numerical experiments. We demonstrate and compare the efficiency of the algorithms on the solution of benchmarks with millions unknowns solved at the Anselm supercomputer of the IT4Innovations National Supercomputing Center.

Keywords: optimization, quadratic programming, contact problems, many-body simulations

Abstrakt

Disertační práce se zaměřuje na řešení optimalizačních úloh minimalizace konvexní kvadratické funkce na speciálních konvexních množinách. Tyto úlohy se vyskytují v mnoha technických aplikacích jako například řešení kontaktních úloh lineární elasticity a řešení úloh částicové dynamiky. Počet neznámých takovýchto praktických úloh obvykle překračuje možnosti sekvenčních algoritmů. Zde prezentované algoritmy jsou však snadno paralelizovatelné.
Text je rozdělen do tří částí - krátké shrnutí teorie kvadratického programování, algoritmy pro řešení úlohy a numerické experimenty. Efektivita prezentovaných algoritmů je demonstrována a porovnána na řešení úloh s miliony neznámých na superpočítači Anselm Národního Superpočítačového Centra IT4Innovations.

Klíčová slova: optimalizace, kvadratické programování, kontaktní úlohy, simulace více těles

Contents

Acknowledgements **1**

Abstract **5**

Contents **5**

List of figures **9**

Abbreviations **11**

Introduction **13**

Notations and preliminarities **17**

1 Fundamental Concepts **23**
1.1 Quadratic function 23
1.2 Convex sets and projections 28
1.3 Solution 31
1.3.1 Kernel and Image 31
1.3.2 Solvability and uniqueness 32
1.3.3 Relation to variational inequalities 35
1.3.4 Lagrange function and KKT conditions 36
1.4 Gradients 40
1.4.1 Projected gradient 40
1.4.2 Reduced gradient 42
1.4.3 Reduced projected gradient 43
1.4.4 Error measurement 44
1.5 Descent along the projected gradient path 45
1.6 Special cases in applications 46
1.6.1 General separable inequality constraints 46
1.6.2 Bound constraints 46
1.6.3 Box constraints 47
1.6.4 Spherical constraints 48
1.6.5 Elliptic constraints 49
1.6.6 Conical constraints 50
1.7 Additional linear equality constraints 50

2 Algorithms **57**
2.1 Spectral projected gradient method (SPG) . . . 58
2.1.1 Barzilai-Borwein method . . . 58
2.1.2 Grippo-Lampariello-Lucidi line-search technique . . . 59
2.1.3 Spectral projected gradient method . . . 61
2.1.4 Modification for QP with one matrix-vector multiplication (SPG-QP) . . . 62
2.2 Projected Barzilai-Borwein method (PBB) . . . 67
2.3 Accelerated projected gradient descent method (APGD) . . . 69
2.4 Active-set methods (MPGP,MPGPS) . . . 79
2.4.1 Modified Proportioning with Gradient projection (MPGP) . . . 81
2.4.2 Modified Proportioning with Barzilai-Borwein gradient projections (MPGP-BB) . . . 81
2.4.3 Modified Weak Proportioning with Gradient projections (Mw-PGP) . . . 81
2.4.4 Modified Proportioning with Reduced Gradient projections (MPRGP) . . . 83
2.4.5 Modified Proportioning with Reduced Gradient Projections for the SPS Hessian (MPRGPS) . . . 83
2.5 Augmented Lagrangian method (SMALSE-M) . . . 84

3 Applications **87**
3.1 Geometric optimization . . . 90
3.1.1 Polytope distance . . . 91
Numerical experiments . . . 93
3.1.2 Smallest enclosing ball . . . 97
Numerical experiments . . . 98
3.1.3 Projection onto intersection of hypercube and hyperplanes . . . 101
3.2 Contact problems of mechanics . . . 107
3.2.1 Membrane . . . 107
3.2.2 Semicoercive problem . . . 112
3.2.3 Problem with friction . . . 115
3.3 Granular dynamics . . . 122
3.3.1 Time-stepping scheme . . . 123
3.3.2 Contacts . . . 125
3.3.3 Friction . . . 130
3.3.4 Properties and solvability . . . 132
The problem without friction . . . 133
The problem with friction . . . 136
Independence of solution . . . 139
3.3.5 Numerical experiments . . . 140

Conclusions **147**

Author's bibliography **149**

References **151**

List of Figures

1 Subsymmetic set . . . 30
2 Projected and reduced gradient . . . 43
3 Feasible step-size in Armijo rule. . . . 60
4 Simplification of SPG algorithm . . . 64
5 SPG-QP: Generalized Armijo condition and possible situations of step-size. . . . 65
6 Performance profile: testing benchmark . . . 88
7 Performance profile: number of iterations of CG, BB, SD . . . 89
8 Polytope distance: testing benchmark . . . 93
9 Polytope distance: number of iterations . . . 95
10 Polytope distance: number of Hessian multiplications . . . 96
11 Polytope distance: performance profiles . . . 96
12 Polytope distance: norm of reduced gradient . . . 97
13 Enclosing ball: the solution of benchmark . . . 98
14 Enclosing ball: norm of reduced gradient . . . 99
15 Enclosing ball: number of iterations . . . 99
16 Enclosing ball: number of Hessian multiplications . . . 100
17 Enclosing ball: performance profiles . . . 100
18 Projection onto intersection of hypercube and hyperplanes: number of iterations . . . 101
19 Projection onto intersection of hypercube and hyperplanes: number of Hessian multiplications . . . 102
20 Projection onto intersection of hypercube and hyperplanes: performance profiles . . . 102
21 Projection onto intersection of sphere and hyperplanes: number of iterations . . . 104
22 Projection onto intersection of sphere and hyperplanes: number of Hessian multiplications . . . 105
23 Projection onto intersection of sphere and hyperplanes: performance profiles . . . 105
24 Membrane: contact problem definition . . . 107
25 Membrane: solution . . . 108
26 Membrane: active constraints in the solution . . . 109
27 Membrane: largest eigenvalue of stiffness matrix . . . 109
28 Membrane: number of iterations for p . . . 109
29 Membrane: performance profiles for p . . . 110
30 Membrane: number of iterations for n . . . 110

31 Membrane: performance profiles for n . . . 111
32 Membrane: norm of reduced gradient . . . 111
33 Cylinder: contact problem definition . . . 112
34 Cylinder: solution . . . 112
35 Cylinder: spectrum distribution . . . 113
36 Cylinder: decrease of projected gradient . . . 113
37 Contact problem with rigid obstacle and given friction. . . . 115
38 Normal and tangential vectors on Γ_C. . . . 116
39 Friction: TFETI example . . . 118
40 Friction: solution (displacement) . . . 119
41 Friction: solution (friction) . . . 120
42 Multibody: contact between two bodies . . . 125
43 Multibody: example of the contact of more bodies . . . 127
44 Multibody: tangent space in problems with friction . . . 137
45 Granular dynamics: solution of non-friction benchmark (1) . . . 141
46 Granular dynamics: solution of non-friction benchmark (2) . . . 141
47 Granular dynamics: number of bodies and contacts in non-friction benchmark . . . 141
48 Granular dynamics: number of iterations in non-friction benchmark . 142
49 Granular dynamics: performance profiles in non-friction benchmark . 142
50 Granular dynamics: number of bodies and contacts in friction benchmark . . . 143
51 Granular dynamics: number of iterations in friction benchmark . . . 143
52 Granular dynamics: performance profiles in friction benchmark 144
53 Granular dynamics: solution of friction benchmark (1) . . . 145
54 Granular dynamics: solution of friction benchmark (2) . . . 145
55 Granular dynamics: solution of friction benchmark (3) . . . 145

Abbreviations

SPD	–	Symmetric positive definite matrix
SPS	–	Symmetric positive semidefinite matrix
QP	–	Quadratic Programming problem
SPG	–	Spectral Projected Gradient method
SPG-QP	–	Spectral Projected Gradient method for Quadratic Programming problem
BB	–	Barzilai-Borwein method
CG	–	Conjugate Gradient method
SD	–	Steepest Descent method
PBB	–	Projected Barzilai-Borwein method
PBBf	–	Projected Barzilai-Borwein method with fallback
GLL	–	Grippo-Lampariello-Lucidi line-search technique
APGD	–	Accelerated Projected Gradient Descent method
MPGP	–	Modified Proportioning with Gradient Projection
MPRGP	–	Modified Proportioning with Reduced Gradient Projection
MPRGPS	–	Modified Proportioning with Reduced Gradient Projection for Quadratic Programming problem with SPS Hessian matrix
MPGP-BB	–	Modified Proportioning with Barzilai-Borwein Gradient Projection
MwPGP	–	Modified Weak Proportioning with Gradient Projection
SMALSE-M	–	Semimonotonic Augumented Lagrangian method for separable and equality constrained quadratic programming problem
FEM	–	Finite Element Method
FETI	–	Finite Element Tearing and Interconnecting method

Introduction

The numerical solution of many engineering problems leads to the problem of minimization of a convex quadratic function subject to a given set of equality and/or inequality constraints. The applications that will benefit from the developement of optimal algorithms for solving such optimization problem are, for instance, the linear elasticity contact problems or the simulation of granular dynamics.
The problem is given by

$$\bar{x} := \arg\min_{x \in \Omega} f(x) \ , \tag{1}$$

where the cost function $f : \mathbb{R}^n \to \mathbb{R}$ is a convex quadratic function and the feasible set $\Omega \subset \mathbb{R}^n$ is a special closed convex set. This problem will be reffered as *Quadratic Programming* (QP) problem. We are mostly interested in the development of algorithms for the solution of large-scale problem with large number of unknowns given by the problem dimension $n \in \mathbb{N}$.
The QP term usually denotes the problems with a feasible set described by linear inequality constraints. The problems of more general inequalities are much more difficult to solve. However, the basic principles of the solution, e.g., the solution conditions, still remain applicable in modified form. Therefore, we decided to generalize this term to the optimization problems of convex quadratic function minimization on any closed convex set described by the separable inequalities with differentiable constraint functions.
For instance, if we consider a linear elasticity contact problem with friction, the feasible set Ω is described not only by linear inequalities, which represent the non-penetration conditions, but also by an additional separable spherical or elliptical inequality constraints representing the friction conditions.
Thus we are also interested in the problems where the feasible set is described not only by inequality constraints, but also by additional equality constraints. These constraints can, for instance, describe gluing conditions in non-overlapping domain decomposition methods, such as Finite Element Tearing and Interconnection methods (FETI). In this case, we use our algorithms for inequality constraints as an inner loop of Augmented Lagrangian method.
The thesis is organized as follows. In Section 1, we review shortly the problem properties and discuss a solvability and an uniqueness of the solution. We recall the theory necessary for the developement of algorithms. These algorithms are presented in Section 2. We are mostly interested in the active-set methods, but for comparison, we choose also other types of methods suggested and sucessfully tested on QP problems by other authors. The other algoritms are usually designed to solve more general problems and they are not using all properties of cost quadratic func-

tion and/or feasible set. Therefore, we suggest our modifications of these methods for solving our QP problems. Moreover, we also briefly review a modification of Augmented Lagrangian method for solving problems with additional linear equality constraints. This method is used as an outer loop of the algorithm for solving QP problems with the combination of equality and inequality constraints and takes care of equality constraints. The problem in inner loop is QP with only inequality constraints. Therefore, we can focus only on the problems with inequalities in the thesis, but practically we are able to solve problems with both types of constraints. Finally, in Section 3 we compare the efficiency of algorithms. We solve the problems arising in practical applications, such as linear elasticity contact problems with friction or multibody dynamics problems.
Our algorithms were firstly implemented in Matlab and tested on reasonable large problems. Afterwards, we implemented the algorithms in C programming language to PERMON library, which is more suitable for high performace computing. The algorithms for solving linear elasticity contact problems were tested at the Anselm supercomputer of the IT4Innovations National Supercomputing Center. The numerical experiments of particle dynamics were implemented in C programming language with CUDA toolkit and tested on GPU card.

Contribution of the thesis

The thesis includes fundamental concepts, algorithms, and applications adopted from several sources and several authors. The source of these foreign ideas is always quoted. For the sake of simplicity, the original author's ideas are not explicitly determined in the text. However, in this short section we introduce a short list of author's contributions.

The thesis are based on papers published in journals and conference proceedings. Author of the thesis is a main author and/or co-author of these contributions. Moreover, these results have been presented on several international and domestic conferences. For full list see section *Author's bibliography*. Author has been cooperating on the paper about short review of QP [34], and developement of new algorithms PBBf (Section 2.2, [60]), MwPGP (Section 2.4.3, [15]), and MPRGPS (Section 2.4.5, [35]). Author published the idea of the extension of active-set algorithm by projected Barzilai-Borwein method instead of constant step-length MPGP-BB in [58]. He is also a main author of the publication focused on granular dynamics [59].

Moreover, this text includes the ideas and the results, which have not been published yet and they are fully auctorial. Author presents his original simplification of Spectral Projected Gradient method for solving QP. This algorithm was called SPG-QP and it performes only one matrix-vector multiplication during the iteration (Section 2.1.4). Author proved equivalency between reduced and mapping gradient (Lemma 2.3.1) and he presented the review of APGD method with projected gradient instead of mapping gradient (Section 2.3). Author suggested the combination of outer SMALSE-M with QP

algorithms different than MPGP or MPRGP; the combinations with SPG-QP, MPGP-BB, APGD are original. Author presents the generalization of the projected gradient definition with general separable index sets; these definitions in other sources are usually presented for problems with specific feasible sets (Section 1.5.1). Author extended the theorem about condition number of penalized matrices and presents the proof (Theorem 1.8.2). The thesis includes new results in the solvability of inner optimization problems in multi-body simulations, see Lemma 3.3.3 (property of QP in problems without friction), Theorem 3.3.1 (about solvability of problems without friction), Lemma 3.3.4 (property of QP in problems with friction), Theorem 3.3.2 (about solvability of problems with friction), and Lemma 3.3.5 (independence of solution and velocity).

Author of the thesis has implemented algorithms in Matlab, CUDA and PETSc. The implementation of algorithms in PETSc is a part of PERMON toolbox, see Hapla [43]. Nowadays, this open-source code is developed at IT4Innovations by several researchers. Author of the thesis has been participating on the implementation of generalized constraints and he has implemented QP algorithms for solving the problems with generalized separable inequalities, such as MPGP-BB, PBBf, SPG-QP, and APGD. The idea of CUDA implementation has been inspired by the Chrono::GPU software developed in SBEL University of Wisconsin-Madison [3]. However, author of the thesis has created his own code from the scratch and during this time he gained a lot of experience in CUDA programming and multi-body dynamics problems.

All bechmarks in the thesis are fully original, except the linear elasticity contact problem with friction in Section 3.2.3. This experiment has been proposed by Dostál and published in collective paper, see Bouchala et al. [15].

Notations and preliminarities

In this section, we shortly present the notations used throughout the thesis. For exact declarations, definitions, and basic properties of these objects, we refer to any basic course of linear algebra and numerical optimization, for instance Laub [51], Nocedal and Wright [54], Boyd and Vandenberghe [17], Golub and Van Loan [41], or Dostál [26].

$\mathbb{R}^n$	standard n-dimensional real vector space, we denote the number of vector components by $n \in \mathbb{N}$.
$v \in \mathbb{R}^n$	real n-dimensional vector. In whole thesis we consider only column (vertical) vectors. The null vector is denoted by 0.
$0 \in \mathbb{R}, 0 \in \mathbb{R}^n, 0 \in \mathbb{R}^{m,n}$	zero or null vector or matrix; the meaning is clear from the context.
$v_j, j = 1, \ldots, n$	components of n-dimensional real vector.
$A \in \mathbb{R}^{m,n}$	real matrix of dimension m, n, where m denotes the number of rows and n denotes the number of columns. The identity matrix is denoted by I.
$A_{i,j}, \quad i = 1, \ldots, m, \ j = 1, \ldots, n$	components of m, n-dimensional matrix, where i denotes the index of the row and j denotes the index of the column.
A^T	transpose of a matrix.

A^{-1}, A^{+}	matrix inverse, matrix pseudoinverse; If A is a nonsingular matrix, then there exists matrix A^{-1} such that $$A^{-1}A = AA^{-1} = I \ .$$ Nevertheless, even if A is a singular matrix, there exists matrix A^{+} such that weaker condition is fulfilled $$AA^{+}A = A \ .$$
diag $(\alpha_1, \ldots, \alpha_n)$	diagonal matrix of dimension n defined by diagonal entries. $$D_{i,j} = \begin{cases} \alpha_i & \text{if } i = j, \\ 0 & \text{elsewhere.} \end{cases}$$
$A = UDU^T$	spectral decomposition of square symmetric matrix $A \in \mathbb{R}^{n,n}$; $U = [v_1, \ldots, v_n] \in \mathbb{R}^{n,n}$ is orthogonal matrix whose columns are the eigenvectors, $D = \text{diag } (\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix whose diagonal entries are eigenvalues corresponding to eigenvectors. Each pair $\lambda_i, v_i, i = 1, \ldots, n$ satisfies $$Av_i = \lambda_i v_i \ .$$
$\sigma(A), \lambda^A_{\max}, \lambda^A_{\min}$	the spectrum of matrix A (the set of eigenvalues), the largest and the smallest eigenvalue. If the superscript is not present, the object matrix is clear from the context.

$\kappa(A), \hat{\kappa}(A)$	the condition number of SPD matrix A; the ratio between the largest and the smallest eigenvalue of A $$\kappa(A) := \frac{\lambda^A_{\max}}{\lambda^A_{\min}} \geq 1.$$ If the matrix A is only SPS, we can define regular condition number; the ratio between the largest and the smallest nonzero eigenvalue of A $$\hat{\kappa}(A) := \frac{\lambda^A_{\max}}{\min\{\lambda \in \sigma(A) : \lambda > 0\}} \geq 1.$$
SPS matrix	the symmetric positive semidefinite matrix; the square symmetric matrix $A \in \mathbb{R}^{n,n}$ which satisfies $$\forall x \in \mathbb{R}^n : x^T A x \geq 0 \ .$$ SPS matrices have real nonnegative eigenvalues. The number of zero eigenvalues is equal to $\dim \operatorname{Ker} A$.
$\operatorname{Ker} A$	the kernel (or the null space) of matrix $A \in \mathbb{R}^{m,n}$; the subspace of $\mathbb{R}^n$ defined by $$\operatorname{Ker} A := \{x \in \mathbb{R}^n : Ax = 0\} \ .$$
$\operatorname{Im} A$	the image of matrix $A \in \mathbb{R}^{m,n}$; the subspace of $\mathbb{R}^m$ defined by $$\operatorname{Im} A := \{Ax \in \mathbb{R}^m | x \in \mathbb{R}^n\} \ .$$

span $\{v_1, \ldots, v_m\} \subset \mathbb{R}^n$	span of the vector set; the subspace of $\mathbb{R}^n$ which consists of all linear combinations of vectors $$v_1, \ldots, v_m \in \mathbb{R}^n \ .$$
$\dim \mathcal{V}$	the dimension of vector space $\mathcal{V}$; the number of basis vectors.
$f : \mathbb{R}^n \to \mathbb{R}$	real function of n real variables.
$\nabla f : \mathbb{R}^n \to \mathbb{R}^n$	the gradient (vector of partial derivatives) of function $f : \mathbb{R}^n \to \mathbb{R}$. If the function f is a function of more vector variables, for example $f(x, y) : \mathbb{R}^{n+m} \to \mathbb{R}$, then the part of the gradient corresponding to partial derivatives of given vector variable is denoted using lower index, for example $\nabla_x f(x, y)$ is a vector of partial derivatives corresponding to the components of x.
$\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n,n}$	the Hessian matrix (matrix of second partial derivatives) of function $f : \mathbb{R}^n \to \mathbb{R}$.

$\langle .,. \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$	standard Euclidean real scalar product (dot product) defined for all $x, y \in \mathbb{R}^n$ by $$\langle x, y \rangle := x^T y = \sum_{i=1}^{n} x_i y_i \ .$$ The dot product is a mapping with these properties: • $\forall x, y \in \mathbb{R}^n : \langle x, y \rangle = \langle y, x \rangle$, • $\forall x \in \mathbb{R}^n : \langle x, x \rangle \geq 0, (\langle x, x \rangle = 0 \Leftrightarrow x = 0)$, • $\forall x, y, z \in \mathbb{R}^n \ \forall \alpha \in \mathbb{R} : \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle, \langle \alpha x, y \rangle = \alpha \langle x, y \rangle$.
$\|.\| : \mathbb{R}^n \to \mathbb{R}_0^+$	standard Euclidean norm defined for any $x \in \mathbb{R}^n$ by $$\|x\| := \sqrt{\langle x, x \rangle} \ .$$ The norm is a mapping with these properties: • $\forall x \in \mathbb{R}^n : \|x\| \geq 0, (\|x\| = 0 \Leftrightarrow x = 0)$, • $\forall x \in \mathbb{R}^n \ \forall \alpha \in \mathbb{R} : \|\alpha x\| = \|\alpha\|.\|x\|$, • $\forall x, y \in \mathbb{R}^n : \|x + y\| \leq \|x\| + \|y\|$.
$\|.\|_A : \mathbb{R}^n \to \mathbb{R}_0^+$	energy norm defined for any $x \in \mathbb{R}^n$ by $$\|x\|_A := \sqrt{\langle Ax, x \rangle} \ ,$$ where $A \in \mathbb{R}^{n,n}$ is given SPD matrix.

$\partial\Omega$	the boundary of closed set $\Omega \subset \mathbb{R}^n$.
$\bar{x} = \arg\min_{x\in\Omega} f(x)$	the optimization problem defined by cost function $f : \mathbb{R}^n \to \mathbb{R}$ and feasible set $\Omega \subset \mathbb{R}^n$. We are searching for $\bar{x} \in \Omega$ with the smallest function value, i.e. $$\forall x \in \Omega : f(x) \geq f(\bar{x}) \ .$$ The existence and uniqueness of the solution depend on the properties of f and Ω and will be discussed in the text.
$\bar{x} = \arg\min f(x)$	the optimization problem defined above with $\Omega := \mathbb{R}^n$.
$x^k \in \mathbb{R}^n, k = 0, 1, \ldots$	the approximation of the solution of given problem in the k-th iteration. The convergent algorithm generates the sequence $\{x^k\}$ such that $$\lim x^k = \bar{x},$$ where $\bar{x}$ is the solution of the problem.

1 Fundamental Concepts

In this section, we present the basic properties of QP problems. We are interested in the properties of the cost function (Section 1.1) and the feasible set (Section 1.2). These properties define the solvability of the optimization problem (Section 1.3). In our algorithms, we use different types of the projected gradients to measure the feasibility of the approximation progress. These types are presented in Section 1.4. The descent of quadratic function may be estimated by the norm of projected gradient. We review the fundamental theory in Section 1.5.
Finally, we apply the presented theory to selected special cases of feasible set in Section 1.6. We are mostly interested in the set types arising in applications presented in Section 3.
The problem with additional linear equalities is presented in Section 1.7.

1.1 Quadratic function

At first, let us define a quadratic function in the most general form.

Definition 1.1.1
(Quadratic function.)

Quadratic function *is a function* $f : \mathbb{R}^n \to \mathbb{R}$ *defined by a symmetric matrix* $A \in \mathbb{R}^{n,n}$*, a vector* $b \in \mathbb{R}^n$*, and prescription*

$$f(x) := \frac{1}{2} x^T A x - b^T x \ . \tag{1.1}$$

In our applications, we always consider symmetric matrix A. Such a quadratic function has suitable form of the gradient and Hessian matrix. But at first, let us introduce probably the most important equality of quadratic programming. Afterwards, using this property, we present the form of the gradient, Hessian matrix, and other suitable properties of quadratic functions.

Lemma 1.1.1
(Increment of the quadratic function value.)

Let f *be a quadratic function defined by* (1.1)*. Then for any* $x, d \in \mathbb{R}^n$ *and* $\alpha \in \mathbb{R}$

$$f(x + \alpha d) = f(x) + \alpha \langle Ax - b, d \rangle + \frac{1}{2} \alpha^2 \langle Ad, d \rangle \ . \tag{1.2}$$

Proof: The given equality is in fact the Taylor expansion of quadratic function. It can be obtained using the basic properties of matrix-vector multiplication and some manipulations. In the thesis, we present the form with scalar product. The quadratic function (1.1) can be written in form

$$f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \ . \tag{1.3}$$

Afterwards, we can substitute and use the basic properties of scalar product. We obtain

$$\begin{aligned} f(x+\alpha d) &= \tfrac{1}{2}\langle A(x+\alpha d), x+\alpha d\rangle - \langle b, x+\alpha d\rangle \\ &= \tfrac{1}{2}\langle Ax, x\rangle + \tfrac{1}{2}\alpha\langle Ad, x\rangle + \tfrac{1}{2}\alpha\langle Ax, d\rangle + \tfrac{1}{2}\alpha^2\langle Ad, d\rangle - \langle b, x\rangle - \alpha\langle b, d\rangle \end{aligned}$$

Since the matrix A is symmetric, we can write

$$\forall x, y \in \mathbb{R}^n : \langle Ax, y\rangle = (Ax)^T y = x^T A^T y = x^T Ay = \langle x, Ay\rangle \ .$$

The rest of proof of (1.2) may be completed by direct computation. □

Lemma 1.1.2
(Gradient and Hessian matrix of quadratic function.)
Let f be a quadratic function defined by (1.1). *Then*

$$\begin{aligned} \nabla f(x) &= Ax - b \ , \\ \nabla^2 f(x) &= A \ . \end{aligned} \tag{1.4}$$

Proof: In (1.2), we set $\alpha := 1$. Since the quadratic function is a polynomial function of the degree 2, the obtained equation can be considered as (exact) Taylor expansion of function f

$$f(x+d) = f(x) + d^T \nabla f(x) + \frac{1}{2} d^T \nabla^2 f(x) d \ . \tag{1.5}$$

From this formula, we can see the values of the gradient and the Hessian matrix (compare (1.2) with (1.5)). □

Remark: Let us notice, the form of the gradient and the Hessian matrix (1.4) is based on the symmetry of matrix A.

The previous lemma brings the basic idea of the problem of minimizing quadratic function on $\mathbb{R}^n$. If we are solving the unconstrained optimization problem with quadratic cost function, we are searching for stationary points, i.e. $\bar{x}$ which solves the system of linear equations $\nabla f(x) = Ax - b = 0$. See next lemma.

Lemma 1.1.3
(Relation between QP and system of linear equations)

Let $A \in \mathbb{R}^{n,n}$ be a SPD matrix, $b \in \mathbb{R}^n$ be a vector. Then the solution $\bar{x} \in \mathbb{R}^n$ of the system of linear equations

$$Ax = b \tag{1.6}$$

is the same as the solution of optimization problem

$$\bar{x} = \arg\min f(x) \ , \tag{1.7}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a quadratic function defined by (1.1).

Proof: The implication (1.7) $\Rightarrow$ (1.6) results from the necessary condition of minimum. The oposite implication can be proven using (1.2) with $\alpha := 1$, SPD property of matrix A, and assumption $A\bar{x} - b = 0$. Then for any $d \in \mathbb{R}^n \setminus \{0\}$, we can write

$$f(\bar{x} + d) - f(\bar{x}) = \frac{1}{2} d^T A d > 0$$

This gives for any $x \neq \bar{x}$ the inequality $f(x) > f(\bar{x})$. □

Remark: The vector b in the definition of quadratic function (1.1) is usually reffered as *right-hand side vector*. The reason is the relation between the optimization problem and the solution of the system of linear equations. See Lemma 1.1.3.

Lemma 1.1.4
(Difference of quadratic function values)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a quadratic function with gradient $g(x) := \nabla f(x)$. Then for any $y, z \in \mathbb{R}^n$

$$f(y) - f(z) = \frac{\langle g(y) + g(z), y - z \rangle}{2}. \tag{1.8}$$

Proof: We can modify the right-hand side of the equality (1.8) into form

$$\begin{aligned}
\tfrac{1}{2}\left(\langle g(y), y - z\rangle + \langle g(z), y - z\rangle\right) &= \tfrac{1}{2}\left(\langle Ay - b, y - z\rangle + \langle Az - b, y - z\rangle\right) \\
&= \tfrac{1}{2}\left(\langle Ay, y\rangle - \langle Ay, z\rangle - \langle b, y\rangle + \langle b, z\rangle + \right. \\
&\quad \left. + \langle Az, y\rangle - \langle Az, z\rangle - \langle b, y\rangle + \langle b, z\rangle\right) \\
&= \tfrac{1}{2}\left(\langle Ay, y\rangle - 2\langle b, y\rangle + 2\langle b, z\rangle - \langle Az, z\rangle\right) \\
&= f(y) - f(z)
\end{aligned}$$

□

The properties of the Hessian matrix of the cost function is usually the most important ingredient in optimization theory. In our problems, this matrix is SPD or SPS. The next lemma presents the basic estimation property of scalar product (or induced norms) with these matrices. It is used in many proofs, for instance in the estimation of inverse Rayleigh quotient in gradient descent methods.

Lemma 1.1.5
(Basic estimations with SPD matrices.)

Let $A \in \mathbb{R}^{n,n}$ be SPD matrix. Then for any $x \in \mathbb{R}^n$ it holds

$$\lambda_{\min}\langle x, x\rangle \leq \langle Ax, x\rangle \leq \lambda_{\max}\langle x, x\rangle \ . \tag{1.9}$$

Proof: Let us consider a spectral decomposition $A = UDU^T$ and notation $z := U^T x$. Then for any $x \in \mathbb{R}^n$

$$\langle Ax, x\rangle = x^T UDU^T x = z^T Dz = \sum_{i=1}^{n} \lambda_i z_i^2 \begin{cases} \leq & \lambda_{\max} z^T z = \lambda_{\max}\langle x, x\rangle, \\ \geq & \lambda_{\min} z^T z = \lambda_{\min}\langle x, x\rangle. \end{cases}$$

□

In optimization theory, the quality and properties of the cost function is crucial. In our problems, the cost function is quadratic. It is defined, continuous and continuously differentiable in any $x \in \mathbb{R}^n$. We can state the following lemma.

Lemma 1.1.6
(Convex quadratic function.)

The quadratic function f is strictly convex[a]*, i.e.*

$$\forall x, y \in \mathbb{R}^n, x \neq y, \forall \alpha \in (0,1) : f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha) f(y)$$

if and only if A is SPD.
The quadratic function f is convex, i.e.

$$\forall x, y \in \mathbb{R}^n, \forall \alpha \in [0,1] : f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

if and only if A is SPS.

[a] Some authors use the term *strong* instead of *strict*. In fact, these terms are equivalent and have the same meaning.

Proof: The lemma can be easily proven using (1.2) and/or from the properties of Hessian matrix A. □

There exists the relation between continuously differentiable functions and strict convexity. See next lemma.

Lemma 1.1.7
(Equivalent property of strictly convex function.)

The continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if and only if there exists constant $\mu > 0$ such that

$$\forall x, y \in \mathbb{R}^n : f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \ . \tag{1.10}$$

Any such μ is reffered to a constant of strong convexity *of the function f.*

Proof: Based on simple manipulations, see Nesterov [53]. □

Remark: Now we are ready to state the constant of strict convexity for quadratic function with SPD Hessian matrix. If we take equality (1.2) and set $\alpha := 1, y := x + d$, we obtain

$$f(y) = f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\|y - x\|_A^2 \ .$$

Using Lemma 1.1.5 we can estimate

$$\|y - x\|_A^2 \geq \lambda_{\min}\|y - x\|^2.$$

It is easy to see that $\mu = \lambda_{\min}$.

In our problems, we are trying to find the minimum of quadratic function. In such a problem, it is always valuable to have the function values bounded from below. In that case, we are trying to find the point in which this minimal value is obtained.

Definition 1.1.2
(Bounded function.)

Let f be a real function defined on non-empty set Ω. Then this function is called bounded from below *on Ω, if there exists a constant $M \in \mathbb{R}$ such that*

$$\forall x \in \Omega : f(x) \geq M \ .$$

Lemma 1.1.8
(Bounded quadratic function.)

The quadratic function (1.1) *is bounded from below on* $\mathbb{R}^n$*, if the Hessian matrix* A *is SPS and* $b \in \operatorname{Im} A$*.*

Proof: This property results from the existence of minimum of quadratic function on $\mathbb{R}^n$. The solvability of such an optimization problem will be discused in Lemma 1.3.2. □

Definition 1.1.3
(Lipschitz continuous function)

A function $f : \mathbb{R}^n \to \mathbb{R}$ *is called* Lipschitz continuous *if there exists constant* $L > 0$ *such that*

$$\forall x, y \in \mathbb{R}^n : \quad |f(x) - f(y)| \leq L\|x - y\| \ . \tag{1.11}$$

Any such L *is reffered to as a* Lipschitz constant *for the function* f*.*

1.2 Convex sets and projections

In this section, we shortly review the basic properties of convex sets. These sets appear in our optimization problems as a feasible sets.

Definition 1.2.1
(Convex set.)

The set $\Omega \subset \mathbb{R}^n$ *is* convex *if for each pair of points within* Ω*, every point on the straight line segment that joins the pair of these points is also within* Ω*, i.e.*

$$\forall x, y \in \Omega \ \forall \alpha \in [0, 1] : \alpha x + (1 - \alpha) y \in \Omega \ .$$

Later in the thesis, we will discuss the projection onto feasible set. In this case, it is necessary to have closed set to be sure that the projection exists.

Definition 1.2.2
(Closed set.)

The set Ω *is closed if for any sequence of points* $\{x^k\}$ *in* Ω*, all limit points of this sequence belong to* Ω*.*

Sometimes the feasible set has additional properties, which bring new properties to the solvability of optimization problem.

Definition 1.2.3
(Bounded set.)

The set Ω is bounded, if there exists a real constant $M > 0$ such that

$$\forall x \in \Omega : \|x\| \leq M \ .$$

The sets, which are bounded and closed are reffered to be compact.

Now we are ready to present the properties of the projection mapping, which assigns to each point $x \in \mathbb{R}^n$ the nearest point from the set. If the set is closed, then this nearest point always exists and it is unique. See next lemma.

Lemma 1.2.1
(Existence and uniqueness of projection.)

Let $\Omega \subset \mathbb{R}^n$ be a non-empty closed convex set. Then for every $x \in \mathbb{R}^n$ there exists a unique projection $P_\Omega(x) \in \Omega$ defined by

$$P_\Omega(x) := \arg\min_{y \in \Omega} \|x - y\| \ . \tag{1.12}$$

Furthermore, $P_\Omega(x)$ is the only point from $\mathbb{R}^n$ such that

$$\langle x - P_\Omega(x), y - P_\Omega(x) \rangle \leq 0 \qquad \forall y \in \Omega.$$

Proof: Since we use the projections in our algoritms very frequently, the proof is important. The projection is an optimization problem itself, the existence and uniqueness are based on the solvability of QP problems presented later in Section 1.3.
At first, let us notice, that the problem (1.12) can be written in the form

$$\arg\min_{y \in \Omega} \|x - y\| = \arg\min_{y \in \Omega} \frac{1}{2}\|x - y\|^2. \tag{1.13}$$

Let us denote the cost function of this optimization problem by $\varphi(y) := \frac{1}{2}\|y - x\|^2, \varphi : \mathbb{R}^n \to \mathbb{R}$ and introduce the non-empty set

$$\Theta := \Omega \cap \{x \in \mathbb{R}^n : \|y - x\| \leq r\} \ ,$$

where $r \geq 0$ is sufficiently large number.
Obviously, Θ is compact [1] and by Weierstrass theorem 1.3.2, there exists a minimum of $\varphi(y)$ on Θ. It is easy to see, that this minimum is equal to the minimum of $\varphi(y)$ on Ω, i.e.

$$\arg\min_{y \in \Theta} \varphi(y) = \arg\min_{y \in \Omega} \varphi(y) \ .$$

[1] The intersection of two bounded closed convex sets is bounded closed convex set.

Moreover, $\varphi(y)$ is a quadratic function with the additional constant term

$$\varphi(y) = \frac{1}{2}\langle y - x, y - x\rangle = \frac{1}{2}y^T I y - y^T x + \frac{1}{2}x^T x \ .$$

It has SPD Hessian matrix $\nabla^2\varphi(y) = I$, therefore φ is strictly convex function. In Section 1.3 we will show, that such a optimization problem with strictly convex cost function on convex feasible set has unique solution, see Lemma 1.3.4.

Let us suppose by contradiction that there exist two different projections of one point $p, \hat{p} \in \Omega, p = P_\Omega(x), \hat{p} = P_\Omega(x), p \neq \hat{p}$ such that

$$\begin{aligned} \langle x - p, y - p\rangle &\leq 0 \qquad \forall y \in \Omega, \\ \langle x - \hat{p}, y - \hat{p}\rangle &\leq 0 \qquad \forall y \in \Omega. \end{aligned}$$

In the first inequality, we choose $y = \hat{p}$ and in the second $y = p$. Summing these two inequalities we obtain

$$\langle p - \hat{p}, p - \hat{p}\rangle \leq 0,$$

which is satisfied if and only if $p = \hat{p}$. □

In [30], Dostál presented a new kind of convex sets. Such sets have additional special property, which is the key ingredient in the theory of the decrease of quadratic function along the projected path, see Theorem 1.5.2.

Definition 1.2.4
(Subsymmetric convex set)

A nonempty closed convex set $\Omega \subset \mathbb{R}^n$ *is* subsymmetric *if for any* $x \in \Omega$, $y \in \mathbb{R}^n$, $s = x - y$, *and* $\tau \in [0, 1]$

$$\|P_\Omega(y + \tau s) - y\| \geq \|P_\Omega(y - \tau s) - y\| \ . \tag{1.14}$$

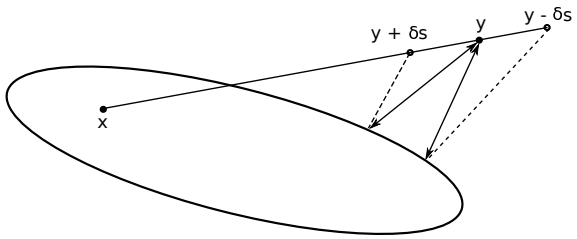


Figure 1: The subsymmetric convex set.

The example of the subsymmetric convex set is shown in Fig. 1. It was already proven that half-intervals, spheres, halfspaces, and their products are subsymmetric. But not all convex sets are subsymmetric, see Bouchala et al. [16]. Elliptic constraints are also subsymmetric, see Bouchala et al. [15].

1.3 Solution

Let us present the basic conditions for the solvability of QP problems. In this section, we briefly discuss the existence and uniqueness of the solution and also we present the optimality conditions.

1.3.1 Kernel and Image

The solvability of unconstrained QP problems depends on the solvability of equivalent system of linear equations, see Lemma 1.1.3. At first, let us remind the basic subspaces used for the discussion of solvability of such systems.

Definition 1.3.1
(Kernel and image)
Let $B \in \mathbb{R}^{n,m}$. We define

$$\begin{aligned} \operatorname{Ker} B &:= \{x \in \mathbb{R}^m : Bx = 0\} \subset \mathbb{R}^m, \\ \operatorname{Im} B &:= \{Bx \in \mathbb{R}^n, x \in \mathbb{R}^m\} \subset \mathbb{R}^n \end{aligned}$$

as a kernel *and an* image *of matrix B, respectively (or equivalently as a kernel and image of linear mapping defined by matrix B).*

The kernel consists of all vectors which are mapped by B onto null vector and the image is the space of all right-hand side vectors, for which the system of linear equations $Bx = b$ has solution.
The next theorem gives the basic relation between image and kernel.

Theorem 1.3.1
(Relation between image and kernel.)
Let $B \in \mathbb{R}^{n,m}$. Then

$$\begin{aligned} \operatorname{Ker} B &\perp \operatorname{Im} B^T , \\ \operatorname{Ker} B^T &\perp \operatorname{Im} B . \end{aligned}$$

Proof: See Laub [51]. □

In this case, the orthogonality between two vector spaces $\mathcal{U} \perp \mathcal{V}$ is defined as orthogonality between all vectors from each space, i.e.

$$\forall u \in \mathcal{U}\ \forall v \in \mathcal{V} : \langle u, v \rangle = 0 .$$

In next, we present the basic theory about simplification of kernel and image of Hessian matrix with special structure that arises in application of granular dynamics, see problems (3.29) and (3.35) in Section 3.

Lemma 1.3.1
(Kernel and image in generalized matrix congruence.)

Let $A \in \mathbb{R}^{n,n}$ be SPD and let $B \in \mathbb{R}^{n,m}$ be a rectangular matrix. Then

$$\operatorname{Ker} B^T AB = \operatorname{Ker} B, \tag{1.15a}$$

$$\operatorname{Im} B^T AB = \operatorname{Im} B^T. \tag{1.15b}$$

Proof: To prove (1.15a), it is necessary to show, that

$$\forall x \in \mathbb{R}^m : \quad B^T ABx = 0 \;\Leftrightarrow\; Bx = 0 \; .$$

($\Leftarrow$) If $Bx = 0$, then $B^T ABx = B^T A0 = 0$.
($\Rightarrow$) Let us consider $x \in \mathbb{R}^m$ such that $B^T ABx = 0$. We can write

$$0 = x^T 0 = x^T B^T ABx = \|Bx\|_A^2 \; .$$

From the property of the norm, we can see that if $\|Bx\|_A = 0$, then $Bx = 0$.
Now we are ready to prove (1.15b).

($\Rightarrow$) Let us consider $y \in \operatorname{Im} B^T AB$, i.e. there exists $x \in \mathbb{R}^m$ such that $B^T ABx = y$. If we denote $z := ABx$, we can see that there exists vector z such that $B^T z = y$. Thus $y \in \operatorname{Im} B^T$.
($\Leftarrow$) We suppose, that $y \in \operatorname{Im} B^T$, i.e. there exists $x \in \mathbb{R}^m$ such that $B^T x = y$.
Using Theorem 1.3.1, we can state that

$$y \in \operatorname{Im} B^T AB \quad \Leftrightarrow \quad \forall z \in \operatorname{Ker}(B^T AB)^T : \langle z, y \rangle = 0 \; .$$

Now we prove that the right-hand side of this equivalency comes true.
Let us consider $z \in \operatorname{Ker}(B^T AB)^T = \operatorname{Ker} B^T AB = \operatorname{Ker} B^T$. We can write

$$\langle z, y \rangle = \langle z, B^T x \rangle = \langle Bz, x \rangle = \langle 0, x \rangle = 0 \; .$$

□

1.3.2 Solvability and uniqueness

In this section, we discuss the solvability of QP (1) with quadratic cost function $f : \mathbb{R}^n \to \mathbb{R}$ defined in Section 1.1 and feasible set $\Omega \subset \mathbb{R}^n$ defined in Section 1.2. The solvability and uniqueness depend on the properties of both objects, i.e. the properties of quadratic function and the properties of feasible set. We are interested in QP with SPD or SPS Hessian matrices.
At first, we focus on the solvability of the problem without constraints, i.e. the problem with $\Omega := \mathbb{R}^n$.

Lemma 1.3.2
(QP solvability without constraints.)

If f is a quadratic function with SPS Hessian matrix A and right-hand side vector $b \in \operatorname{Im} A$, then the problem (1) with $\Omega := \mathbb{R}^n$ has a solution. Moreover, if A is SPD, then this solution is unique.

Proof: The unique solution of problem with SPD matrix is given by Lemma 1.1.3. In this case, the solution of the optimization problem (1) is equivalent to the solution of the linear system

$$Ax = b \ . \tag{1.16}$$

Since the matrix A is nonsingular, all solutions (i.e. the only one) are given by

$$\bar{x} = A^{-1}b \ .$$

In the case of SPS Hessian matrix, $\operatorname{Ker} A$ is non-trivial, so the system (1.16) is not solvable for all right-hand side vectors b. The system is solvable if and only if $b \in \operatorname{Im} A$. Moreover, the solutions of this system form the set of all stationary points, so the solvability of the system defines also the solvability of the optimization problem. Furthermore, the quadratic function with SPS matrix is convex (the function is continuous and Hessian matrix is SPS, see Lemma 1.1.2), therefore all stationary points are minimizers. If matrix A is SPS and $b \in \operatorname{Im} A$, then all stationary points are given by

$$\bar{x} := A^{+}b + R\alpha, \quad \alpha \in \mathbb{R}^r \ , \tag{1.17}$$

where A^{+} is a pseudoinverse of the matrix A, $R \in \mathbb{R}^{n,r}$ is a full rank matrix such that $\operatorname{Im} R = \operatorname{Ker} A$, and $r \leq n$ is a dimension of $\operatorname{Ker} A$. Suppose, that $b \in \operatorname{Im} A$, then there exists $y \in \mathbb{R}^n$ such that $b = Ay$. Using Lemma 1.1.1, we can express

$$\begin{aligned} \forall d \in \mathbb{R}^n : f(A^{+}b + R\alpha + d) - f(A^{+}b) &= \langle \nabla f(A^{+}b), R\alpha + d\rangle + \tfrac{1}{2}\langle A(R\alpha + d), R\alpha + d\rangle \\ &= \langle AA^{+}b - b, R\alpha + d\rangle + \tfrac{1}{2}\langle Ad, d\rangle \\ &= \langle AA^{+}Ay - Ay, R\alpha + d\rangle + \tfrac{1}{2}\langle Ad, d\rangle = \tfrac{1}{2}\langle Ad, d\rangle \geq 0 \ . \end{aligned}$$

Then for all $d \notin \operatorname{Ker} A$

$$f(\bar{x} + d) - f(\bar{x}) = \frac{1}{2}d^T Ad > 0 \ .$$

We proved that if $d \in \operatorname{Ker} A$, then $\bar{x} + d$ is still the solution of system (1.16), i.e. the minimum of f.
Therefore, for any $x \neq \bar{x}$ it holds $f(x) \geq f(\bar{x})$. □

Remark: If $b \notin \operatorname{Im} A$, then system (1.16) has no solution, the cost function has no stationary point. Therefore, the optimization problem (1) with $\Omega := \mathbb{R}^n$ has no solution.

Now let us consider nontrivial QP problems. Let $\Omega \neq \mathbb{R}^n$ be the feasible set. The basic theorem of solvability of such optimization problems is Weierstrass theorem.

Theorem 1.3.2
(Weierstrass extreme value theorem.)

If f is a real-valued continuous function on a non-empty compact (i.e. bounded and closed) domain Ω, then there exists $x \in \Omega$ such that $f(x) \geq f(y)$ for all $y \in \Omega$.

Using this theorem, we are ready to set the basic lemma of solvability of QP problems on compact sets.

Lemma 1.3.3
(QP solvability on compact set)

If $\Omega \subset \mathbb{R}^n$ is non-empty compact (i.e. bounded and closed), then QP (1) has always solution.

Proof: Directly from Weierstrass theorem 1.3.2. □

Remark: Therefore, the QP problem (1) with SPD or SPS matrix with compact feasible set has always solution.

In our optimization problems, the feasible set Ω is not always compact, for example the feasible set described by bound constraints. Nevertheless, all of our feasible sets are closed convex. In this case, the solvability is given by following lemma.

Lemma 1.3.4
(QP solvability on closed convex set)

If f is a quadratic function with SPD Hessian matrix and $\Omega \subset \mathbb{R}^n$ is closed convex set, then the QP problem (1) has an unique solution.
If f is a quadratic function with SPS Hessian matrix A, $b \in \operatorname{Im} A$, and $\Omega \subset \mathbb{R}^n$ is closed convex set, then the QP problem (1) has always a solution.

Proof: The quadratic function with SPD or SPS Hessian and $b \in \operatorname{Im} A$ is bounded from below on $\mathbb{R}^n$, see Lemma 1.1.8. If we consider only closed subset $\Omega \subset \mathbb{R}^n$, then this function is still bounded from below. This function is also continuous, therefore there exists a minimum. See Dostál [26]. □

Lemma 1.3.5
(DIFFERENCE OF QP SOLUTIONS.)

Let $\bar{x}_1, \bar{x}_2$ denote two solutions of QP problem (1). Then

$$\bar{x}_1 - \bar{x}_2 \in \operatorname{Ker} A \ .$$

Proof: See Dostál [26]. □

1.3.3 Relation to variational inequalities

The QP problem can arise as a numerical solution of variational inequalities. The next lemma presents this connection.

Lemma 1.3.6
(VARIATIONAL INEQUALITY EQUIVALENCY)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable strictly convex function and let $\Omega \subset \mathbb{R}^n$ be closed convex set.
Then $\bar{x} \in \Omega$ is a solution of optimization problem

$$\bar{x} : = \arg\min_{x \in \Omega} f(x) \tag{1.18}$$

if and only if

$$\forall x \in \Omega : \langle \nabla f(\bar{x}), x - \bar{x} \rangle \geq 0 \ . \tag{1.19}$$

Proof: Let us prove implication from bottom (1.19) to top (1.18), i.e. we suppose, that (1.19) holds. The function f is strictly convex, therefore using the property of strictly convex functions (1.10) with $x := \bar{x}$ and $y := x$ we get $\forall x \in \Omega$:

$$\begin{aligned} f(x) &\geq f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \tfrac{\mu}{2} \|\bar{x} - x\|^2 \\ &\quad \text{the norm is non-negative} \\ &\geq f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle \\ &\quad \text{use (1.19)} \\ &\geq f(\bar{x}). \end{aligned}$$

Therefore $\bar{x}$ is the minimizer of f.

The second implication can be proven by contradiction. Let $\bar{x}$ be a solution of the problem (1.18). Assume (by contradiction) that there exists $\hat{x} \in \Omega$ such that

$$\langle \nabla f(\bar{x}), \hat{x} - \bar{x} \rangle < 0 \ . \tag{1.20}$$

Let us consider a function $\phi(\alpha) := f(\bar{x} + \alpha(\hat{x} - \bar{x})), \alpha \in [0, 1]$. For this function, it holds

$$\begin{aligned} \phi(0) &= f(\bar{x}) \\ \phi'(0) &= \langle \nabla f(\bar{x}), \hat{x} - \bar{x} \rangle < 0 \quad \text{(using (1.20))} \end{aligned}$$

Therefore, for small enough α we have

$$f(\bar{x} + \alpha(\hat{x} - \bar{x})) = \phi(\alpha) < \phi(0) = f(\bar{x}).$$

And this is contradiction. □

Remark: Using previous lemma, we can see that the QP problem (1) with symmetric matrix A is equivalent to the solution of variational inequality

$$\forall y \in \Omega : \langle A\bar{x}, y - \bar{x} \rangle \geq \langle b, y - \bar{x} \rangle.$$

1.3.4 Lagrange function and KKT conditions

One of the most classical way how to solve optimization problem with differentiable objects is to set up Lagrange function and Karush-Kuhn-Tucker optimality conditions (KKT). Afterwards, the solution of these equations can be considered as a problem of linear programming. In the thesis, we rather do not solve QP problem using this technique, but the KKT conditions are still important in the developement of optimal QP solvers. They define the optimality conditions. The basic lemma modified for QP follows.

Lemma 1.3.7
(About Lagrange function and KKT conditions.)

Let us consider the QP problem (1)*, where the feasible set is convex, described by equality and inequality constraints*

$$\Omega := \left\{ x \in \mathbb{R}^n : \begin{array}{lcll} h_{Ei}(x) & = & 0 & i = 1, \ldots, m_E \\ h_{Ij}(x) & \leq & 0 & j = 1, \ldots, m_I \end{array} \right\} \neq \emptyset,$$

where

- $h_{Ei} : \mathbb{R}^n \to \mathbb{R}$ *are linear functions describing equality constraints,*
- $h_{Ij} : \mathbb{R}^n \to \mathbb{R}$ *are convex functions describing inequality constraints.*

Suppose $\bar{x}$ *solves this problem. Then there are vectors* $\lambda_E \in \mathbb{R}^{m_E}$ *and* $\lambda_I \in \mathbb{R}^{m_I}, \lambda_I \geq 0$ *such that* $\bar{x}$ *solves the Lagrangian problem*

$$\bar{x} := \arg\min_x L(x, \lambda_E, \lambda_I) \ ,$$

where $L : \mathbb{R}^n \times \mathbb{R}^{m_E} \times \mathbb{R}^{m_I} \to \mathbb{R}$ *is* Lagrange function *defined by*

$$L(x, \lambda_E, \lambda_I) := f(x) + \sum_{i=1}^{m_E} \lambda_{Ei} h_{Ei}(x) + \sum_{j=1}^{m_I} \lambda_{Ij} h_{Ij}(x) \ . \tag{1.21}$$

The appropriate optimality conditions of this problem, so-called Karush-Kuhn-Tucker *conditions (KKT), are given by*

$$\begin{aligned}
\nabla_x L(x, \lambda_E, \lambda_I) &= \nabla f(x) + \sum_{i=1}^{m_E} \lambda_{Ei} \nabla h_{Ei}(x) + \sum_{j=1}^{m_I} \lambda_{Ij} \nabla h_{Ij}(x) &= 0 \\
\nabla_{\lambda_E} L(x, \lambda_E, \lambda_I) &= [h_{E1}(x), \ldots, h_{Em_E}(x)]^T &= 0 \\
\nabla_{\lambda_I} L(x, \lambda_E, \lambda_I) &= [h_{I1}(x), \ldots, h_{Im_I}(x)]^T &\leq 0 \\
& \lambda_I &\geq 0 \\
& \lambda_{Ij} h_{Ij} &= 0, \quad j = 1, \ldots, m_I
\end{aligned} \tag{1.22}$$

Proof: See Luenberger [52], Bertsekas [12], Nocedal and Wright [54], Boyd and Vandenberghe [17], or Dostál [26]. □

Examples for the particular feasible sets can be found in Section 1.6. In the end of this section, let us present a simple examples to demonstrate the solvability of QP problem.

Example 1.3.1
Let us consider a QP problem (1) with

$$A := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \Omega := \left\{ x \in \mathbb{R}^2 : x_1 \leq 0 \wedge x_2 \leq 0 \right\}.$$

It is easy to check that A is SPS and the kernel and the image of A are given by

$$\operatorname{Im} A = \operatorname{span} \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}, \quad \operatorname{Ker} A = \operatorname{span} \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\},$$

and $b \in \operatorname{Im} A$. The appropriate Lagrange function $L : \mathbb{R}^4 \to \mathbb{R}$ is given by

$$L(x_1, x_2, \lambda_1, \lambda_2) = f(x) + \lambda_1 x_1 + \lambda_2 x_2.$$

In this case, the quadratic cost function can be written in simple form

$$f(x) = \frac{1}{2}x^T Ax - b^T x = \frac{1}{2}\,[x_1, x_2] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - [1, 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2}x_1^2 - x_1.$$

Using this, it is easy to check that the KKT conditions (1.22) for this problem are given by

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_1} = x_1 - 1 + \lambda_1 \qquad = 0 \tag{1.23a}$$

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_2} = \lambda_2 \qquad = 0 \tag{1.23b}$$

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial \lambda_1} = x_1 \qquad \leq 0 \tag{1.23c}$$

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial \lambda_2} = x_2 \qquad \leq 0 \tag{1.23d}$$

$$\lambda_1 \qquad \geq 0 \tag{1.23e}$$

$$\lambda_2 \qquad \geq 0 \tag{1.23f}$$

$$\lambda_1 x_1 \qquad = 0 \tag{1.23g}$$

$$\lambda_2 x_2 \qquad = 0 \tag{1.23h}$$

We can take equation (1.23a) and express

$$x_1 = 1 - \lambda_1 \tag{1.24}$$

and afterwards, we substitute this into condition (1.23g). We obtain

$$\lambda_1(1 - \lambda_1) = 0$$

This quadratic equation has two roots $\bar{\lambda}_1 = 1, \hat{\lambda}_1 = 0$. which satisfy condition (1.23e). Using (1.24) and $\bar{\lambda}_1 = 1$, we obtain $\bar{x}_1 = 0$, which satisfies also condition (1.23c). If we use $\hat{\lambda}_1 = 0$ in (1.24), we obtain $\hat{x}_1 = 1$, which violates condition (1.23c). Therefore, there exists only one solution $\bar{x}_1 = 0$.
Since from (1.23d) and (1.23f) we get $\bar{\lambda}_2 = 0$, the conditions (1.23d) and (1.23h) are satisfied for any $\bar{x}_2 \geq 0$.
Therefore, all solutions of optimization problem is given by

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ t \end{bmatrix}, \quad t \geq 0,$$

or equivalently

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} \in \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} + d, d \in \operatorname{Ker} A \right\} \cap \Omega.$$

■

Example 1.3.2
Let us consider QP with similar data as in Example 1.3.1, but in this case we consider right-hand side vector

$$b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

It is easy to check that $b \notin \operatorname{Im} A$. In this case, the Lagrange function is given by

$$L(x_1, x_2, \lambda_1, \lambda_2) = \frac{1}{2}x_1^2 - x_1 - x_2 + \lambda_1 x_1 + \lambda_2 x_2.$$

The KKT conditions are given by

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_1} = x_1 - 1 + \lambda_1 \qquad = 0 \tag{1.25a}$$

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_2} = -1 + \lambda_2 \qquad = 0 \tag{1.25b}$$

with the rest of conditions same as in previous example, i.e. (1.23c) - (1.23h).
The only λ_2, which satisfies condition (1.25b) is given by $\bar{\lambda}_2 = 1$. From (1.23h) we get $\bar{x}_2 = 0$. The first component of the solution $\bar{x}_1 = 0$ can be obtained using the same metology as in Example 1.3.1. Therefore, the problem has unique solution $[\bar{x}_1, \bar{x}_2]^T = 0$. ■

Example 1.3.3
Let us consider QP with similar data as in Example 1.3.2. In this case we consider right-hand side vector

$$b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

It is easy to check that $b \notin \operatorname{Im} A$. The KKT conditions are given by

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_1} = x_1 - 1 + \lambda_1 \qquad = 0 \tag{1.26a}$$

$$\frac{\partial L(x_1, x_2, \lambda_1, \lambda_2)}{\partial x_2} = 1 + \lambda_2 \qquad = 0 \tag{1.26b}$$

with the rest of conditions same as in previous examples, i.e. (1.23c) - (1.23h). From (1.26b), we get $\hat{\lambda}_2 = -1$. Obviously, this solution violates condition (1.23f). Therefore, the problem with this right-hand side vector has not solution. ■

1.4 Gradients

In this section, we define several types of projected gradients. The norm of these gradients can be used as a measurement of the solution accuracy.

1.4.1 Projected gradient

At first, we denote

$$\begin{aligned} \mathcal{N} &:= \{1, \ldots, n\}, \\ \mathcal{M} &:= \{1, \ldots, m\} \end{aligned}$$

the set of indices of unknown vector $x \in \mathbb{R}^n$ and the set of indices of constraints, respectively. We consider a feasible *separable* set $\Omega \subset \mathbb{R}^n$ composed from the sets of smaller dimension $\Omega_j, j \in \mathcal{M}$

$$\Omega := \Omega_1 \times \cdots \times \Omega_m, \tag{1.27}$$

where each $\Omega_j \subset \mathbb{R}^{|\mathcal{I}_j|}$ is described by one constraint function $h_j : \mathbb{R}^{|\mathcal{I}_j|} \to \mathbb{R}$

$$\Omega_j := \{x \in \mathbb{R}^{|\mathcal{I}_j|} : h_j(x) \leq 0\}. \tag{1.28}$$

Here $\mathcal{I}_j \subset \mathcal{N}$ is the index set with the indexes of components of x constrained by function h_j. The number of constrained components is denoted by $|\mathcal{I}_j|$, i.e. the dimension of the set $\mathcal{I}_j$. Furthermore, we consider a *separable* feasible set, thus

$$\forall i, j \in \mathcal{M}, i \neq j : \mathcal{I}_i \cap \mathcal{I}_j = \emptyset.$$

If all components of x are not constrained, we denote by $\mathcal{I}_{uncon}$ the index set with unconstrained indices. Afterwards, we can define the constraint function which is always satisfied

$$\begin{aligned} h_{uncon}(x) &:= -\infty, \quad h_{uncon} : \mathbb{R}^{|\mathcal{I}_{uncon}|} \to \mathbb{R}, \\ \Omega_{uncon} &:= \{x \in \mathbb{R}^{|\mathcal{I}_{uncon}|} : h_{uncon}(x) \leq 0\}. \end{aligned}$$

From this point of view, all components of x are constrained and we can write

$$\begin{aligned} \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_m &= \mathcal{N}, \\ |\mathcal{I}_1| + \cdots + |\mathcal{I}_m| &= |\mathcal{N}| = n. \end{aligned} \tag{1.29}$$

For every $x \in \mathbb{R}^n$, we can decompose the set of all constraints indexes into *free* and *active* set defined by

$$\begin{aligned} \mathcal{F}(x) &:= \{j \in \mathcal{M} : h_j(x_{\mathcal{I}_j}) \neq 0\} \ , \\ \mathcal{A}(x) &:= \{j \in \mathcal{M} : h_j(x_{\mathcal{I}_j}) = 0\} \ . \end{aligned} \tag{1.30}$$

Using this decomposition, we are able to define *free* gradient $\varphi(x) \in \mathbb{R}$ and *chopped* gradient with components given by

$$\begin{aligned} \varphi_{\mathcal{I}_j}(x) &:= \begin{cases} g_{\mathcal{I}_j}(x) & \text{if } j \in \mathcal{F}(x), \\ 0 & \text{if } j \in \mathcal{A}(x), \end{cases} \\ \beta_{\mathcal{I}_j}(x) &:= \begin{cases} 0 & \text{if } j \in \mathcal{F}(x), \\ g_{\mathcal{I}_j}(x) - \min\{n_j^T(x_{\mathcal{I}_j}) g_{\mathcal{I}_j}(x), 0\} n_j(x_{\mathcal{I}_j}) & \text{if } j \in \mathcal{A}(x), \end{cases} \end{aligned} \tag{1.31}$$

where $g(x) := \nabla f(x) = Ax - b \in \mathbb{R}^n$ denotes the gradient of cost quadratic function and $n_j(x) \in \mathbb{R}^{|\mathcal{I}_j|}$ is an unit outward normal to Ω_j defined for any $x \in \partial\Omega_j$ by

$$n_j(x) := \frac{1}{\|\nabla h_j(x)\|} \nabla h_j(x). \tag{1.32}$$

Afterwards, we define the *projected* gradient as a sum of free and chopped gradient

$$g^P(x) := \varphi(x) + \beta(x) \ .$$

Lemma 1.4.1
(About projected gradient and solution.)

Vector $\bar{x}$ is a solution of optimization problem (1) *if and only if*

$$g^P(\bar{x}) = 0 \ .$$

Proof: Can be proven directly from KKT conditions (1.22). See Bertsekas [12] and Dostál [26]. □

The previous lemma shows that the norm of the projected gradient can be considered as a natural measurement of the KKT violation, i.e. natural measurement of the optimality.

1.4.2 Reduced gradient

Other methods (mostly not active-set methods) use more general criterium of optimality. This condition can be also found in the solution of differential inequalities. We define *reduced* gradient by prescription

$$\widetilde{g}_\alpha(x) := \frac{1}{\alpha}(x - P_\Omega(x - \alpha\nabla f(x))) \ . \tag{1.33}$$

In our algorithms, we usually use $\alpha \in (0, 1/\lambda^A_{\max}\rangle$. This coefficient appears in projected gradient path theory given by Schöberl and Dostál [36].
The easiest way how to understand the meaning of reduced gradient is to take a look into projected gradient descend methods, i.e. the methods with iteration prescription

$$x_{k+1} := P_\Omega(x_k - \alpha_k\nabla f(x_k)) \ ,$$

which differs from classical gradient descend methods (such as Steepest Descent method) only with additional projection onto feasible set. Using this projection we are sure, that every iteration lies in the feasible set.
Then reduced gradient reflects the distance between *old* iteration and new iteration scaled by the inverse value of the step-size.

Lemma 1.4.2
(About reduced gradient and solution.)

Vector $\bar{x}$ is a solution of optimization problem (1) *if and only if*

$$\widetilde{g}_\alpha(\bar{x}) = 0 \ .$$

Proof: The optimization problem can be rewritten using Lemma 1.3.6 to

$$\text{find } \bar{x} \text{ such that } \ \forall x \in \Omega : \langle \nabla f(\bar{x}), x - \bar{x}\rangle \geq 0.$$

This equation can be easily transformed to form

$$\text{find } \bar{x} \text{ such that } \ \forall x \in \Omega : \langle \bar{x} - \alpha\nabla f(\bar{x}) - \bar{x}, x - \bar{x}\rangle \leq 0,$$

where we choose $\alpha > 0$. Afterwards, using Lemma 1.3.6 and the definition of projection in Lemma 1.2.1 we obtain equivalent condition

$$\bar{x} = P_\Omega(\bar{x} - \alpha\nabla f(\bar{x})).$$

The rest of the proof is straightforward. □

Remark: The previous lemma shows that the norm of the reduced gradient can be also considered as a natural measurement of the KKT violation, i.e. natural measurement of the optimality.

1.4.3 Reduced projected gradient

The projected and reduced gradient can be combined together into new type of gradient. This new type uses advantages of both, i.e. the decomposition of stopping criterion into free and active part and non-linear behaviour of constraints described by projections. The *reduced projected* gradient was succesfully used for solving the QP problems on convex sets with strong curvature, see Bouchala, et al. [15].

Example 1.4.1
Let us consider a problem with elliptic constraint. The projected and reduced gradient have each its own difficulty when it is used as a stopping criterion, see Fig. 2. Even if the size of the absolute error $\|x_k - \bar{x}\|$ is small, the size of used gradient can be large. These difficulties can be eliminated using the combination of these gradient types. ■

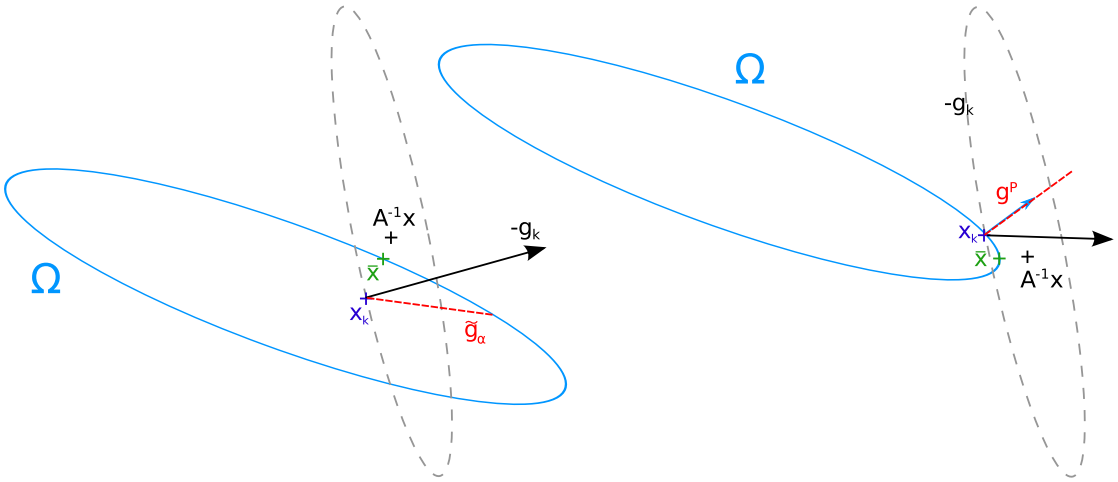


Figure 2: Example - reduced gradient (left) and projected gradient (right); even if the absolute error is small, the norm of the projected or reduced gradient could be large.

At first, we define *reduced chopped* gradient by prescription

$$\widetilde{\beta}^{\alpha}_{\mathcal{I}_j}(x) := \begin{cases} 0 & \text{if } j \in \mathcal{F}(x)\ , \\ g_{\mathcal{I}_j}(x) & \text{if } j \in \mathcal{A}(x) \text{ and } n_j^T(x_{\mathcal{I}_j}) g_{\mathcal{I}_j}(x) > 0\ , \\ \frac{1}{\alpha}(x_{\mathcal{I}_j} - P_{\Omega_j}(x_{\mathcal{I}_j} - \alpha g_{\mathcal{I}_j}(x))) & \text{if } j \in \mathcal{A}(x) \text{ and } n_j^T(x_{\mathcal{I}_j}) g_{\mathcal{I}_j}(x) \leq 0 \end{cases}$$

and afterwards, the *reduced projected* gradient is given by

$$\widetilde{g}_\alpha^P(x) := \varphi(x) + \widetilde{\beta}^\alpha(x) \ .$$

Lemma 1.4.3
(About reduced projected gradient and solution.)

Vector $\bar{x}$ is a solution of optimization problem (1) if and only if

$$\widetilde{g}_\alpha^P(\bar{x}) = 0 \ .$$

Proof: The property is a combination of Lemma 1.4.1 and 1.4.2. See Bouchala, et al. [15]. □

1.4.4 Error measurement

The next lemma brings together relations between the norm of absolute error and norm of gradient types defined above.

Lemma 1.4.4
(Absolute error and gradient types.)

Let $\bar{x}$ be the solution of (1). Then for any $x \in \Omega$ it holds

$$\|x - \bar{x}\|_A^2 \leq 2(f(x) - f(\bar{x})) \leq \|g^P(x)\|_{A^{-1}}^2 \leq \frac{1}{\lambda_{\min}^A} \|g^P(x)\|^2 \ , \tag{1.34a}$$

$$\|x - \bar{x}\| \leq \nu(\alpha) \|\widetilde{g}_\alpha(x)\| \ , \tag{1.34b}$$

$$\|x - \bar{x}\| \leq \nu(\alpha) \|\widetilde{g}_\alpha^P(x)\| \ , \tag{1.34c}$$

where

$$\nu(\alpha) := \begin{cases} 1/\lambda_{\min}^A & \text{for} \quad 0 < \alpha \leq 2/(\lambda_{\min}^A + \lambda_{\max}^A) \ , \\ \alpha / \left(2 - \alpha \lambda_{\max}^A\right) & \text{for} \quad 2/(\lambda_{\min}^A + \lambda_{\max}^A) \leq \alpha < 2/\lambda_{\max}^A \ . \end{cases}$$

Proof: Proof of (1.34a) can be found in Dostál and Kozubek [30], Lemma 2. For (1.34b) and (1.34c) see Bouchala, et al. [15]. □

Next lemma gives us relations between the gradient types.

Lemma 1.4.5
(Relations between gradient types.)

For $x \in \Omega$ it holds

$$\|\widetilde{g}_\alpha\|^2 \le g^T \widetilde{g}_\alpha \le g^T g^P = \|g^P\|^2, \tag{1.35a}$$
$$\|\widetilde{g}_\alpha\| \le \|\widetilde{g}^P_\alpha\| \le \|g^P\|. \tag{1.35b}$$

Moreover, there exists a constant $C \ge 1$ such that for each $x \in \Omega$ and $\alpha \in (0, 2/\lambda^A_{\max})$

$$\|\widetilde{g}^P_\alpha(x)\| \le \|g^P(x)\| \le C\|\widetilde{g}^P_\alpha(x)\| \ .$$

Proof: See Dostál and Kozubek [30], Bouchala, et al. [15]. □

1.5 Descent along the projected gradient path

In Dostál and Schöberl [36], authors give an estimation of descent of quadratic function along the projected path. They show, that if we use constant step-size in projected gradient descent method, we obtain a decrease of the cost function. See next theorem.

Theorem 1.5.1
(Constant step-length.)

Let Ω be a closed convex set, let $\bar{x}$ denote the unique solution of (1), and $\bar{\alpha} \in (0, \|A\|^{-1})$.
Then for all $x \in \Omega$

$$f\left(P_\Omega\left(x - \bar{\alpha} g(x)\right)\right) - f(\bar{x}) \le \rho\left(f(x) - f(\bar{x})\right),$$

where

$$\rho = (1 - \bar{\alpha}\lambda_{\min}(A)) < 1 \ .$$

Proof: See Dostál and Schöberl [36]. □

Furthermore, in next paper [30], Dostál and Kozubek show, that this interval can be extended, if the feasible set has special property. Such sets were called subsymmetric. It has been already proven that half-intervals, spheres, halfspaces, and their products are subsymmetric, but not all convex sets are subsymmetric, see Bouchala, et al. [16]. Elliptic constraints are subsymmetric, see Bouchala, et al. [15].

Theorem 1.5.2
(Constant step-length on subsymmetric set.)

Let Ω be a closed convex subsymmetric set, let $\bar{x}$ denote the unique solution of (1), $\overline{\mu} = 2\|A\|^{-1}$, and $\bar{\alpha} \in (0, \overline{\mu})$.
Then for all $x \in \Omega$

$$f\left(P_\Omega\left(x - \bar{\alpha} g(x)\right)\right) - f(\bar{x}) \leq \eta(\bar{\alpha})\left(f(x) - f(\bar{x})\right),$$

where

$$\eta(\alpha) := \max\left\{1 - \alpha\lambda_{\min}(A), 1 - (\overline{\mu} - \alpha)\lambda_{\min}(A)\right\} \ .$$

Proof: See Dostál and Kozubek [30]. □

1.6 Special cases in applications

Now we are ready to present KKT for QP on particular types of sets.

1.6.1 General separable inequality constraints

We have already presented a way how to define separable inequality constraints, mostly in the case of the combination of different types of constraints for different components of x, in Section 1.4.1. Generally, each constraint function h_j defines the part of the total feasible set $\Omega_j, j = 1, \ldots, m$, see (1.28). The constraint function constraints the components of the unknown vector x. These components are denoted by $\mathcal{I}_j$, see (1.29). Moreover, we suppose that the constraint functions $h_j : \mathbb{R}^{|\mathcal{I}_j|} \to \mathbb{R}$ are convex and differentiable, so we are able to compute unit outward normal in every boundary point $x \in \partial\Omega_j$, see (1.32).

1.6.2 Bound constraints

The feasible set defined by *bound constraints* is the closed convex set defined by formula

$$\Omega := \{x \in \mathbb{R}^n : x \geq l\} \ ,$$

where $l \in \mathbb{R}^n$ is the vector of lower bound components. To include the possibility that not all the components are constrained, we admit $l_i = -\infty$. In this case, we can use the index set to define constrained components, but if we consider problems with only bound constraints, we obtain only complicated notations. Since this, we define a feasible set using the whole vector of bound constraints.
The components of projection $P_\Omega(x)$ have for any $x \in \mathbb{R}^n$ simple form

$$[P_\Omega(x)]_i := \max\{l_i, x_i\}, \quad i = 1, \ldots, n \ .$$

We set in notations (1.27), (1.28), (1.29), (1.32)

$$\begin{aligned} \mathcal{I}_j &:= \{j\}, \quad j = 1, \ldots, n, \\ h_j(x) &:= l_j - x, \quad h_j : \mathbb{R} \to \mathbb{R}, \\ n_j(x) &:= -1 \end{aligned}$$

and KKT conditions (1.22) have form

$$\begin{aligned} Ax - b - \lambda &= 0, \\ l - x &\leq 0, \\ \lambda &\geq 0, \\ \forall i = 1, \ldots, n : \quad \lambda_i(l_i - x_i) &= 0. \end{aligned}$$

These conditions can be reformulated into the form ($\bar{x} \in \Omega$ is a solution if and only if)

$$A\bar{x} - b \geq 0 \quad \text{and} \quad (A\bar{x} - b)^T(l - \bar{x}) = 0 \ ,$$

see Dostál [26].

Bound constraints appear, for example, in contact mechanic problems as an enforcement of non-penetration condition, see Section 3.2.

1.6.3 Box constraints

If we add to the set described by lower bounds also upper bounds, we obtain a set described by box constraints

$$\Omega := \{x \in \mathbb{R}^n : l \leq x \leq u\rangle \ ,$$

where $u \in \mathbb{R}^n : u \geq l$ is a vector of upper bounds. Similarly, to include the possibility that not all the components are constrained by upper bound, we admit $u_i = \infty$. The components of the projection are given by

$$[P_\Omega(x)]_i := \min\{\max\{l_i, x_i\}, u_i\}, \quad i = 1, \ldots, n \ .$$

The most important difference between bound constraints and box constraints is non-separability of the box constraints. In notations (1.27), (1.28), (1.29), (1.32),

we can see two sets of constraint functions

$$\begin{aligned} \mathcal{I}_j &:= \{j\}, \quad j = 1, \ldots, n, \\ h_j^l(x) &:= l_j - x, \quad h_j^l : \mathbb{R} \to \mathbb{R}, \\ h_j^u(x) &:= x - u_j, \quad h_j^u : \mathbb{R} \to \mathbb{R}, \\ n_j^l(x) &:= -1, \\ n_j^u(x) &:= 1. \end{aligned}$$

In this case, the KKT conditions (1.22) have form

$$\begin{aligned} Ax - b - \lambda^l + \lambda^u &= 0, \\ l - x &\leq 0, \\ x - u &\leq 0, \\ \lambda^l, \lambda^u &\geq 0, \\ \forall i = 1, \ldots, n: \quad \lambda_i^l (l_i - x_i) &= 0, \\ \lambda_i^u (x_i - u_i) &= 0. \end{aligned}$$

Since $l < u$, we can decompose the active set into two disjont subsets

$$\begin{aligned} \mathcal{A}^l(x) &:= \{j \in \mathcal{N} : h_j^l(x_{\mathcal{I}_j}) = 0\} , \\ \mathcal{A}^u(x) &:= \{j \in \mathcal{N} : h_j^u(x_{\mathcal{I}_j}) = 0\} \end{aligned}$$

and then define free and active set by

$$\begin{aligned} \mathcal{F}(x) &:= \{j \in \mathcal{N} : h_j^l(x_{\mathcal{I}_j}) < 0 \ \wedge \ h_j^u(x_{\mathcal{I}_j}) < 0\} , \\ \mathcal{A}(x) &:= \mathcal{A}^l(x) \cup \mathcal{A}^u(x) . \end{aligned}$$

Box constraints appear, for example, in 2D contact problems with friction.

1.6.4 Spherical constraints

Spherical constraints define the relation between pairs of unknowns

$$\Omega_j := \{[x_1, x_2] \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq r_j^2\}, \quad j = 1, \ldots, m .$$

The vector of radii is represented by $r \in \mathbb{R}^m, r_j > 0$.
We set in notations (1.28), (1.29), (1.32)

$$\begin{aligned} \mathcal{N} &:= \{1, \ldots, m\}, \\ \mathcal{I}_j &:= \{2j-1, 2j\}, \quad j = 1, \ldots, m, \\ h_j(x) &:= x_1^2 + x_2^2 - r_j^2, \quad h_j : \mathbb{R}^2 \to \mathbb{R}, \\ n_j(x) &:= (1/\|x\|)\, x \end{aligned}$$

and KKT conditions (1.22) have form

$$\begin{aligned} Ax - b + 2\hat{\lambda}x &= 0, \\ \|x_{\mathcal{I}_j}\|^2 - r_j^2 &\le 0, \\ \lambda &\ge 0, \\ \forall j = 1, \ldots, m: \quad \lambda_j(\|x_{\mathcal{I}_j}\|^2 - r_j^2) &= 0, \end{aligned}$$

where $\hat{\lambda} \in \mathbb{R}^{n,n}$ is a diagonal matrix with Lagrange multipliers defined by

$$\hat{\lambda}_{kl} := \begin{cases} \lambda_j & \text{if } k = l \in \mathcal{I}_j, \\ 0 & \text{elsewhere.} \end{cases}$$

In this case, the projection onto the boudary of circle with radius r_j is realized by simple formula

$$P_{\Omega j}(x_{\mathcal{I}_j}) := \frac{r_j}{\|x_{\mathcal{I}_j}\|} x_{\mathcal{I}_j} \ .$$

Spherical constraints are important in 3D contact problems with friction, see Section 3.2.

1.6.5 Elliptic constraints

These constraints are the generalization of spherical constraints and have the form

$$\Omega_j := \{x \in \mathbb{R}^2 : x^T B_j x \le 1\}, \quad j = 1, \ldots, m \ .$$

In fact, the boundary of ellipse can be represented by quadratic function defined by SPD matrix $B_j \in \mathbb{R}^{2,2}$.
The projection is non-trivial and can be considered as optimization problem itself, see Haslinger et al. [44].

For any $x \in \partial\Omega_j$, the outward unit normal is given by

$$n_j := \frac{1}{\|B_j x\|} B_j x \ .$$

Elliptic constraints are important in 3D contact problems with anisotropic friction, see Bouchala, et al. [15].

1.6.6 Conical constraints

The conical constraint represents the cone in 3 dimensions

$$\Omega_j := \{x \in \mathbb{R}^3 : x_2^2 + x_3^2 \leq \mu_j^2 x_1^2 \quad \wedge \quad x_1 \geq 0\}, \quad j = 1, \ldots, m$$

or equivalently

$$\Omega_j := \{x \in \mathbb{R}^3 : \sqrt{x_2^2 + x_3^2} \leq \mu_j x_1\}, \quad j = 1, \ldots, m \ .$$

The vector $\mu \in \mathbb{R}^m, \mu_j \in [0, 1]$ represents the properties of cones.
The problem of projection can be solved analytically (see Heyn [13])

$$P_{\Omega_j}(x) := \begin{cases} x & \text{if } \mu_j x_1 - \sqrt{x_2^2 + x_3^2} \geq 0 \ , \\ 0 & \text{if } -\frac{x_1}{\mu_j} - \sqrt{x_2^2 + x_3^2} \geq 0 \ , \\ \frac{x_1 + \mu_j \sqrt{x_2^2 + x_3^2}}{\mu_j^2 + 1} \left[1, x_2 \frac{\mu_j}{\sqrt{x_2^2 + x_3^2}}, x_3 \frac{\mu_j}{\sqrt{x_2^2 + x_3^2}}\right]^T & \text{elsewhere} \end{cases}$$

and outward unit normal is given by

$$n_j(x) := \frac{\hat{n}_j(x)}{\|\hat{n}_j(x)\|}, \quad \hat{n}_j(x) := \begin{cases} [-1, 0, 0]^T & \text{if } x = 0, \\ \left[-\mu_j, \frac{x_2}{\sqrt{x_2^2 + x_3^2}}, \frac{x_3}{\sqrt{x_2^2 + x_3^2}},\right] & \text{elsewhere.} \end{cases}$$

Conical constraints appear in granular dynamics problems with friction, see Section 3.3.

1.7 Additional linear equality constraints

In this section, we consider QP problems with additional linear equality constraints, i.e.

$$\bar{x} := \arg\min_{x \in \Omega} f(x) \ , \tag{1.36}$$

where the feasible set $\Omega \subset \mathbb{R}^n$ is described by the combination of the feasible set presented Section 1.2 and additional linear equality constraints

$$\begin{aligned} \Omega &:= \Omega_I \cap \Omega_E, \\ \Omega_E &:= \{x \in \mathbb{R}^n : Bx = 0\}. \end{aligned} \tag{1.37}$$

We require neither that $B \in \mathbb{R}^{m,n}$ is a full row rank matrix nor $m \leq n$, but we assume that $\operatorname{Ker} B$ is non-trivial vector space to guarantee that Ω_E is not empty. We also assume that B is sparse or $m \ll n$. Let us point out that confining ourselves to the homogeneous equality constraints does not mean any loss of generality, as we can use a simple transform to reduce any non-homogeneous equality constraints to our case.
However, we start our exposition with the problem to find the minimizer of the quadratic function $f(x)$ subject to linear equality constraints, that is $\Omega_I := \mathbb{R}^n$. The appropriate Lagrange function (see Section 1.3) has form

$$L(x, \lambda) = f(x) + \lambda^T B x, \qquad L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R} \tag{1.38}$$

and KKT conditions are given by

$$\begin{array}{rcccl} \nabla_x L(x,\lambda) & = & Ax - b + B^T\lambda & = & 0, \\ \nabla_\lambda L(x,\lambda) & = & Bx & = & 0. \end{array} \tag{1.39}$$

The *saddle point* system of linear equations (1.39) is usually written in form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

and it can be solved simply directly as a system of linear equations with symmetric matrix. Furthermore, the preconditioning techniques can be also used, see Benzi et al. [11].
Moreover, if A is SPD and B is a full row rank matrix, the system (1.39) can be reduced eliminating primal variables x to the form of *Schur complement system*

$$BA^{-1}B^T\lambda = BA^{-1}b,$$

see e.g. Dostál [26]. Since the matrix of this system is SPD, we can write using Lemma 1.1.3 the problem in equivalent form

$$\bar{\lambda} := \arg\min_{\lambda \in \mathbb{R}^m} \frac{1}{2}\lambda^T BA^{-1}B^T\lambda - \lambda^T BA^{-1}b. \tag{1.40}$$

The technique of eliminating the primal variables to the problem with dual variables is called *dualization* and the problem (1.40) is called *dual problem*.

Another way how to enforce the equality constraints is to use the *penalty method*. It is well known (e.g. Bertsekas [12], Nocedal and Wright [54], Boyd and Vandenberghe [17], or Dostál [26]) that the problem with equality constrained problem can

be aproximated by the solution of unconstrained problem with additional penalty term in the cost function

$$\bar{x} := \arg\min_{x \in \Omega_E} f(x) \quad \approx \quad \hat{x} := \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2}\|Bx\|^2,$$

where $\rho > 0$ is sufficiently large penalty parameter. The relation between the solution of the constrained problem and unconstrained problem with penalty term is given by the next theorem presented by Dostál [26]. The theorem gives the estimation of the feasibility error of the solution of unconstrained penalized function.

Theorem 1.7.1
(About feasibility error of penalized function)

Let $A \in \mathbb{R}^{n,n}$ be SPD matrix, $B \in \mathbb{R}^{m,n}$ be nonzero matrix, and $b \in \mathbb{R}^n$. We assume that B is not necessarily a full rank matrix. Let $\varepsilon \geq 0$ and $\rho > 0$.
Let $\hat{x}$ is an approximate solution of unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \qquad f_\rho(x) := f(x) + \frac{\rho}{2}\|Bx\|^2 \tag{1.41}$$

such that the necessary optimality condition $\nabla_x f_\rho(x) = 0$ is satisfied approximately with respect to relative precision $\varepsilon\|b\|$, i.e.

$$\|\nabla_x f_\rho(x)\| \leq \varepsilon\|b\|,$$

then

$$\|Bx\| \leq \frac{1+\varepsilon}{\sqrt{\lambda^A_{\min}\rho}}\|b\|.$$

Proof: See Dostál [26] and Dostál and Horák [28]. □

Let us take a better look into the necessary optimality condition of the solution of penalized problem (1.41)

$$\nabla_x f_\rho(x) = Ax - b + \rho B^T Bx = 0.$$

This expression can be modified into

$$(A + \rho B^T B)x - b = 0.$$

The next theorem presents the properties of the system matrix in this equation.

Theorem 1.7.2
(About condition number of penalized matrix)

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite matrix, let $B \in \mathbb{R}^{m\times n}$, $\rho > 0$, and let $\operatorname{Ker} A \cap \operatorname{Ker} B = \{0\}$. *Then matrix*

$$A_\rho = A + \rho B^T B$$

is symmetric positive definite and

$$\kappa(A_\rho) \geq \hat{\kappa}(A) \ . \tag{1.42}$$

Moreover,

- *if* $\hat{\kappa}(B^T B) \leq \hat{\kappa}(A)$ *and*
$$\rho \in \left[\frac{\hat{\lambda}^A_{\min}}{\hat{\lambda}^{B^TB}_{\min}}, \frac{\lambda^A_{\max}}{\lambda^{B^TB}_{\max}} \right]$$
then $\kappa(A_\rho) = \hat{\kappa}(A)$,
- *if* $\hat{\kappa}(B^T B) > \hat{\kappa}(A)$ *then* $\forall \rho > 0 : \kappa(A_\rho) > \hat{\kappa}(A)$.

Proof: Dostál proved the first part of the theorem in [26], Lemma 1.2. This proof follows.
If $x \in \mathbb{R}^n \setminus \{0\}$ and $\operatorname{Ker} A \cap \operatorname{Ker} B = \{0\}$, then either $Ax \neq 0$ or $Bx \neq 0$. Since $Ax \neq 0$ is equivalent to $A^{\frac{1}{2}}x \neq 0$, we get for $\rho > 0$

$$\langle A_\rho x, x\rangle = \langle (A + \rho B^T B)x, x\rangle = \|A^{\frac{1}{2}}x\|^2 + \rho\|Bx\|^2 > 0 \ .$$

Thus A_ρ is positive definite.

Let us consider a spectral decomposition of symmetric matrix A in form

$$A = U_A^T \Sigma_A U_A = \begin{bmatrix} U^T_{\mathrm{Im}} U^T_{\mathrm{Ker}} \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_A & \\ & 0 \end{bmatrix} \begin{bmatrix} U_{\mathrm{Im}} \\ U_{\mathrm{Ker}} \end{bmatrix}, \tag{1.43}$$

where $\operatorname{Im} U^T_{\mathrm{Im}} = \operatorname{Im} A$, $\operatorname{Im} U^T_{\mathrm{Ker}} = \operatorname{Ker} A$, $r = \operatorname{rank} A$, and $\hat{\Sigma}_A \in \mathbb{R}^{r,r}$ is a diagonal matrix with positive eigenvalues of matrix A.
Matrix $B^T B \in \mathbb{R}^{n,n}$ is also symmetric positive semidefite. If we denote $M = U_{\mathrm{Ker}} B^T B U_{\mathrm{Ker}} \in \mathrm{R}^{n-r\times n-r}$, we can write

$$B^T B = U^T_{\mathrm{Ker}} M U_{\mathrm{Ker}} = \begin{bmatrix} U^T_{\mathrm{Im}} U^T_{\mathrm{Ker}} \end{bmatrix} \begin{bmatrix} 0 & \\ & M \end{bmatrix} \begin{bmatrix} U_{\mathrm{Im}} \\ U_{\mathrm{Ker}} \end{bmatrix}. \tag{1.44}$$

Furthermore, we can also consider a spectral decomposition of symmetric matrix M in form $M = V^T \hat{\Sigma}_M V$. Using this, (1.43), and (1.44) we obtain

$$\begin{aligned} A_\rho &= \left[U_{\mathrm{Im}}^T U_{\mathrm{Ker}}^T\right] \begin{bmatrix} \hat{\Sigma}_A & \\ & \rho M \end{bmatrix} \begin{bmatrix} U_{\mathrm{Im}} \\ U_{\mathrm{Ker}} \end{bmatrix} \\ &= \left[U_{\mathrm{Im}}^T \;\; U_{\mathrm{Ker}}^T V^T\right] \begin{bmatrix} \hat{\Sigma}_A & \\ & \rho\hat{\Sigma}_M \end{bmatrix} \begin{bmatrix} U_{\mathrm{Im}} \\ V U_{\mathrm{Ker}} \end{bmatrix} = U^T \begin{bmatrix} \hat{\Sigma}_A & \\ & \rho\hat{\Sigma}_M \end{bmatrix} U \ . \end{aligned}$$

Let us remark that matrix U is orthogonal, so eigenvalues of SPD matrix A_ρ are given by

$$\sigma(A_\rho) = \sigma(\hat{\Sigma}_A) \cup \sigma(\rho\hat{\Sigma}_M) \subset \mathbb{R}^+ \ . \tag{1.45}$$

Let us denote the largest eigenvalue of symmetric positive definite matrix M by $\lambda_{\max}^M$ and smallest eigenvalue by $\lambda_{\min}^M$. For symmetric positive semidefinite matrix A we can denote the smallest non-zero eigenvalue by $\hat{\lambda}_{\min}^A$. Then the condition number and the regular condition number are defined by

$$\kappa(M) = \frac{\lambda_{\max}^M}{\lambda_{\min}^M}, \quad \hat{\kappa}(A) = \frac{\lambda_{\max}^A}{\hat{\lambda}_{\min}^A}.$$

Using the definitions and the relations between the objects one can directly proof that

$$\kappa(\hat{\Sigma}_M) = \kappa(M) = \hat{\kappa}(B^T B) = \kappa(\rho M), \quad \kappa(\hat{\Sigma}_A) = \hat{\kappa}(A).$$

Now, we are interested in the relation between $\hat{\kappa}(A)$ and $\kappa(A_\rho)$. Obviously, the ratio between the largest and the smallest components of set $\sigma(\hat{\Sigma}_A) = \sigma(A) \setminus \{0\}$ in (1.45) can not be reduced by the union with $\sigma(\rho\hat{\Sigma}_M) = \sigma(\rho B^T B) \setminus \{0\}$. This implies

$$\kappa(A_\rho) \geq \hat{\kappa}(A).$$

Now we focus on the situation when $\kappa(A_\rho) = \hat{\kappa}(A)$. In this case, the eigenvalues can be ordered

$$\hat{\lambda}_{\min}^A \leq \rho\lambda_{\min}^M \leq \rho\lambda_{\max}^M \leq \lambda_{\max}^A. \tag{1.46}$$

Such an inequalities are fulfilled if and only if

$$\frac{\hat{\lambda}_{\min}^A}{\lambda_{\min}^M} \leq \rho \leq \frac{\lambda_{\max}^A}{\lambda_{\max}^M} \quad \text{and} \quad \kappa(M) \leq \hat{\kappa}(A). \tag{1.47}$$

Furthermore, if $\kappa(M) > \hat{\kappa}(A)$ or equivalently

$$\frac{\hat{\lambda}_{\min}^A}{\lambda_{\max}^A} > \frac{\lambda_{\min}^M}{\lambda_{\max}^M}$$

then there is not such a $\rho > 0$ that inequalities (1.46) are fulfilled. □

The similar technique can be used also for the optimization problem constrained by the combination of linear equalities and inequalities

$$\bar{x} := \arg \min_{x \in \Omega_I \cap \Omega_E} f(x) \quad \approx \quad \hat{x} := \arg \min_{x \in \Omega_I} \frac{1}{2} x^T (A + \rho B^T B) x - b^T x. \tag{1.48}$$

In this case, the necessary condition of the optimality of the problem constrained only by inequalities is given by the gradient types presented in Section 1.4.

Combining the Lagrangian for equality constrained QP problem (1.38) and penalty method (1.48), we obtain so-called *Augumented Lagrangian*

$$L(x, \lambda_E, \rho) = \frac{1}{2} x^T (A + \rho B^T B) x - (b - B^T \lambda)^T x. \tag{1.49}$$

2 Algorithms

In this section, we present the algorithms, which can be used to solve QP problem on selected feasible set presented in Section 1.6.

- *Spectral Projected Gradient method* (SPG, SPG-QP)
  This method was firstly presented by Martinéz et al. [13]. In the thesis, we shortly review the algorithm. The SPG method combines Barzilai-Borwein (BB) steps with projections and additional Grippo-Lampariello-Lucidi (GLL) line-search step to enforce the convergence based on the generalized Armijo condition. Here, we present our own modification of SPG for QP problem. The presented theory shows that if the cost function is quadratic, we can omit the iterative line-search. The appropriate step-size is given by simple formula.

- *Projected Barzilai-Borwein method* (PBBf)
  PBBf is a simple modification of SPG. We try to omit the line-search step and instead of using the generalized Armijo condition to enforce the convergence, we present an alternative descend control. This backtracing technique is based on the theory of projected gradient path presented in Section 1.5.

- *Accelerated Projected Gradient Descent method* (APGD)
  Nesterov [53] presented an algorithm for solving the optimization problems with Lipschitz continuous convex cost function on closed convex feasible sets. We shortly review the algorithm and present our own modification for solving QP problems. The modification is based on the properties of the quadratic cost function presented in Section 1.1.

- *Active-set methods* (MPRGP, MPRGPS, MPGP, MwPGP, MPGP-BB)
  Dostál [26] presented optimal active-set methods for solving QP problems. In this text, we shortly review the algorithms and present the new modification for the solution of QP with SPS Hessian. We also present the modification for separable constraints with strong curvature or modification with the projected Barzilai-Borwein method. Such steps speed up the convergence in final stage of the solution.

- *Augmented Lagrangian method* (SMALBE-M)
  We shortly review the algorithm for solving the problems with additional linear equality constraints. This modification of the classical Augmented Lagrangian algorithm by Hestenes and Stiefel [46] was presented by Dostál [26]. The idea is a combination of algorithms for solving the problem with inequalities and Augmented Lagrangian method with adaptive precision control of inner

optimization problem solution to enforce equality constraints. Dostál usually uses an active-set methods as an inner solver. In the thesis, we extend this study by tests with our alternative algorithms mentioned above.

2.1 Spectral projected gradient method (SPG)

At first, we shortly review the Barzilai-Borwein method (BB, see Barzilai and Borwein [10]). This algorithm will be extended by projections of approximations to feasible set. Furthermore, we use the Grippo-Lampariello-Lucidi line-search technique (GLL) to enforce the convergence. Using this combination, we obtain the Spectral projected gradient method (SPG, see [13]).

2.1.1 Barzilai-Borwein method

Gradient descent methods for solving unconstrained quadratic programming problem, i.e. (1) with $\Omega = \mathbb{R}^n$, are based on the construction of a sequence of the solution aproximations using recursive formula

$$x^{k+1} = x^k - \alpha_k g^k, \qquad k = 0, 1, \ldots \tag{2.1}$$

with the step size $\alpha_k \in \mathbb{R}^+$ and the vector of the steepest descent $-g^k := -\nabla f(x^k)$. The most popular gradient descent method is the Steepest Descent method (SD, firstly presented by Cauchy [19]). This method uses the step-length, which minimizes function $f(x^{k+1})$ using locally optimal step-size

$$x^{k+1} = \arg\min_{\alpha \in \mathbb{R}} \{x^k - \alpha \nabla f(x^k)\}, \qquad \Rightarrow \qquad \alpha_k = \frac{\langle g^k, g^k \rangle}{\langle Ag^k, g^k \rangle}.$$

The step-size of BB method is based on the different idea. To briefly review the relation of the BB method for the solution of unconstrained problems to the Newton method for solving a scalar non-linear equation $g(x) = 0$, let us follow Raydan [61] and replace the derivative $g'(x^k)$ in the Newton method by its secant approximation to get

$$x^{k+1} = x^k - \frac{1}{g'(x^k)} g(x^k) \;\approx\; x^k - \frac{x^k - x^{k-1}}{g(x^k) - g(x^{k-1})} g(x^k) \; . \tag{2.2}$$

Denoting $g^k = g(x^k) = f'(x^k) = \nabla f(x^k)$ and

$$\alpha_k = \frac{x^k - x^{k-1}}{g^k - g^{k-1}} \; , \tag{2.3}$$

we can see that the secant method (2.2) can be considered as a gradient descend method (2.1). If $g(x) : \mathbb{R}^n \to \mathbb{R}^n$, then we cannot evaluate α_k by (2.3), but we can

assemble the *secant equation*

$$\frac{1}{\alpha_k}(x^k - x^{k-1}) = g^k - g^{k-1} \tag{2.4}$$

and solve it in the least-squares sense

$$\alpha_k = 1/\arg\min_{\beta\in\mathbb{R}} \|(x^k - x^{k-1})\beta - (g^k - g^{k-1})\|^2 \ .$$

After denoting

$$s^k = x^k - x^{k-1} \ , \qquad g^k - g^{k-1} = As^k \ ,$$

and some simplifications, we get the strictly convex minimizing problem

$$\alpha_k = 1/\arg\min_{\beta\in\mathbb{R}} \left(\langle s^k, s^k\rangle\beta^2 - 2\langle As^k, s^k\rangle\beta + \langle As^k, As^k\rangle\right)$$

with the solution

$$\alpha_k = \frac{\langle s^k, s^k\rangle}{\langle As^k, s^k\rangle} \ . \tag{2.5}$$

This is the step-size of the BB method. This observation was firstly presented by Raydan [61]. The proof of convergence with estimations was presented by Dai and Liao [22].

2.1.2 Grippo-Lampariello-Lucidi line-search technique

In this section, we shortly review the method for finding the step-size $\beta_k \in \mathbb{R}^+$ in the second part of one SPG step given by Grippo, Lampariello, and Lucidi [42]. We consider more general descent method in the form

$$x^{k+1} = x^k + \beta_k d^k \ , \tag{2.6}$$

where $d^k \in \mathbb{R}^n$ is descent direction, i.e. we require

$$\langle d^k, g^k\rangle < 0 \ . \tag{2.7}$$

The problem is to find step-size $\beta_k \in \mathbb{R}^n$ to fulfill appropriate descent criterium to obtain the convergence of the algorithm to minimal value.

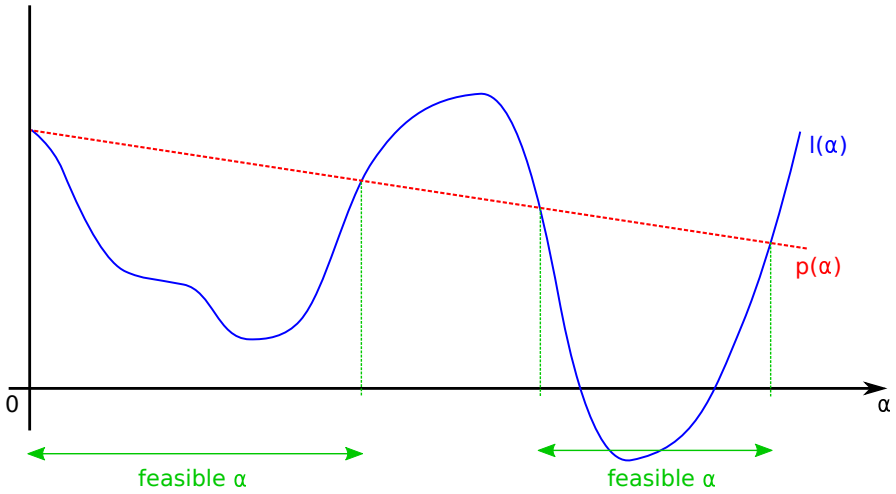


Figure 3: Feasible step-size in Armijo rule.

Definition 2.1.1
(Armijo condition)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a cost function of minimizing problem, let $x, d \in \mathbb{R}^n$ be an approximation of the solution and the descent direction, respectively, and let $\gamma \in (0, 1)$ be a constant. We refer the inequality

$$f(x + \alpha d) \leq f(x) + \gamma\alpha\langle g, d\rangle \tag{2.8}$$

as Armijo condition.

Remark: If we denote the left side of inequality (2.8) as $l(\alpha) : \mathbb{R} \to \mathbb{R}$ and the right side as $p(\alpha) : \mathbb{R} \to \mathbb{R}$, then $p(\alpha)$ is affine function, see Fig. 3. We say that α satisfies Armijo condition for given x, d, if (2.8) holds.

Algorithm 2 is the GLL algorithm to find β_k which satisfied the modification of (2.8). This modification is called *generalized Armijo condition*. Due to this condition, the algorithm is convergent. See theorem below.

Theorem 2.1.1
(About generalized Armijo condition)

Let $\{x^k\} \subset \mathbb{R}^n$ be a sequence generated by prescription

$$x^{k+1} = x^k + \beta_k d^k, \quad d^k \in \mathbb{R}^n \setminus \{0\} \ .$$

Let $\beta > 0, \sigma \in (0, 1), \gamma \in (0, 1)$ and M be a nonnegative integer.
Assume that

1. *the set*
$$\Omega_0 := \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$$
is compact.

2. *there exist constants* $c_1, c_2 > 0$ *such that*
$$\begin{aligned} \langle g^k, d^k \rangle &\leq -c_1 \|g^k\|^2 , \\ \|d^k\| &\leq c_2 \|g^k\| . \end{aligned}$$

3. $\beta_k = \sigma^{h_k}\beta$, *where* h_k *is the first nonnegative integer* h *for which a special variant of* Armijo-condition *holds*
$$f(x^k + \sigma^h \beta d^k) \leq \max_{0 \leq j \leq m(k)} \{f(x^{k-j})\} + \gamma \sigma^h \beta \langle g^k, d^k \rangle ,$$
where
$$\begin{aligned} m(0) &= 0 , \\ m(k) &: \quad 0 \leq m(k) \leq \min\{m(k-1)+1, M\}, \quad k \geq 1 . \end{aligned}$$

then

1. *the sequence* $\{x^k\}$ *remains in* Ω_0 *and every limit point* $\bar{x}$ *satisfies* $\nabla f(\bar{x}) = 0$,
2. *no limit point of* $\{x^k\}$ *is a local maximum of* f,
3. *if the number of the stationary points of* f *in* Ω_0 *is finite, the sequence* $\{x^k\}$ *converges.*

Proof: See Grippo, Lampariello, and Lucidi [42]. □

2.1.3 Spectral projected gradient method

The Spectral Projected Gradient method (SPG) is based on the combination of the projected BB method with the GLL line-search technique. In Birgin, Martinéz, and Raydan [14], it can be found Algorithm 3.

From the schema, we can see that every iteration consists of two steps. At first step, we use the projected gradient method using the BB step-length computed in previous iteration. Using GLL we find a new approximation between obtained

Algorithm 2: **GLL line-search.**

Given cost function $f : \mathbb{R}^n \to \mathbb{R}$*, parameter* $m \in \mathbb{N}$*, approximation and direction* $x^k, d^k \in \mathbb{R}^n$*, parameter* $\gamma \in (0, 1)$*, safeguarding parameters* $\sigma_1, \sigma_2 \in \mathbb{R} : 0 < \sigma_1 < \sigma_2 < 1$.

$f_{\max} := \max\{f(x^{k-j}) : 0 \leq j \leq \min\{k, m-1\}\}$

$x_{temp} := x^k + d^k$

$\delta := \langle \nabla f(x^k), d^k \rangle$

$\beta := 1$

while $f(x_{temp}) > f_{\max} + \gamma\beta\delta$

$\beta_{temp} := -\frac{1}{2}\beta^2\delta / (f(x_{temp}) - f(x^k) - \beta\delta)$

if $\beta_{temp} \in \langle \sigma_1, \sigma_2\beta \rangle$

$\beta := \beta_{temp}$

else

$\beta := \beta/2$

endif

$x_{temp} := x^k + \beta d^k$

endwhile

Return step-size β.

projection and previous approximation. Then, the new BB step-length is computed. See Fig. 4.
The proof of convergence is based on fulfillment of the generalized Armijo condition in every step, see Birgin, Martinéz, and Raydan [13].

2.1.4 Modification for QP with one matrix-vector multiplication (SPG-QP)

The SPG was developed to solve more general optimization problems on convex sets. In our problems, the cost function is quadratic function. We can use the prescription and the properties to simplify the GLL algorithm and we obtain an algorithm with less cost function evaluations, i.e. with the smaller number of the most time-consuming operation - multiplication by Hessian matrix.

Algorithm 3: **Spectral projected gradient method (SPG).**

Given cost function $f : \mathbb{R}^n \to \mathbb{R}$, *initial approximation* $x^0 \in \Omega$, *projection onto feasible set* $P_\Omega(x)$, *safeguarding parameters* $0 < \alpha_{\min} \ll \alpha_{\max}$, *precision* $\varepsilon > 0$, *and initial step-size* $\alpha_0 > 0$.

$k := 0$
while $\|P(x^k - \nabla f(x^k)) - x^k\| > \varepsilon$
  $d^k := P(x^k - \alpha_k \nabla f(x^k)) - x^k$
  compute step-size β_k *using GLL*
  $x^{k+1} := x^k + \beta_k d^k$
  $s^k := x^{k+1} - x^k$
  $y^k := \nabla f(x^{k+1}) - \nabla f(x^k)$
  if $\langle s^k, y^k \rangle \leq 0$
    $\alpha_{k+1} := \alpha_{\max}$
  else
    $\alpha_{k+1} := \min\{\alpha_{\max}, \max\{\alpha_{\min}, \langle s^k, s^k \rangle / \langle s^k, y^k \rangle\}\}$
  endif
  $k := k + 1$
endwhile

Return approximation of solution x^k.

We start with the most obvious simplifications. Notice that

$$y^k := \nabla f(x^{k+1}) - \nabla f(x^k) = (Ax^{k+1} - b) - (Ax^k - b) = A(x^{k+1} - x^k) = As^k.$$

Since matrix A is symmetric positive definite, we can write for any $s^k \in \mathbb{R}^n \setminus \{0\}$

$$\langle s^k, y^k \rangle = \langle s^k, As^k \rangle = (s^k)^T As^k > 0$$

and the condition in SPG algorithm is always fulfiled. Moreover, the BB step-length $\alpha_{k+1} = \langle s^k, s^k \rangle / \langle As^k, s^k \rangle$ is the inverse Rayleigh quotient and it can be bounded by

$$\frac{1}{\lambda_{\max}} \leq \alpha_{k+1} \leq \frac{1}{\lambda_{\min}},$$

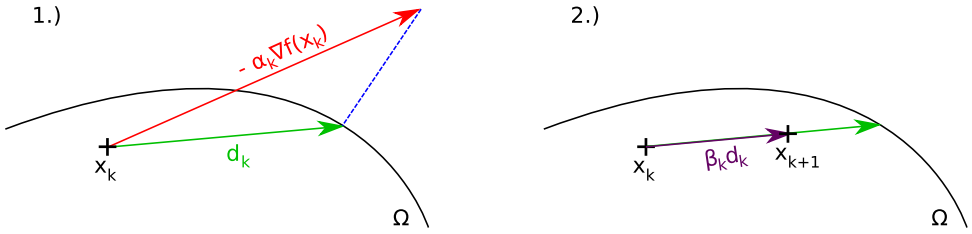


Figure 4: SPG iteration in two steps.

where $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and the largest eigenvalues of matrix A. Therefore, we can omit safeguarding parameters $\alpha_{\min}$ and $\alpha_{\max}$.

Let us take a better look into GLL line-search Algorithm 2. The computation of β_{temp} can be simplified using the Lemma 1.1.1. We obtain

$$\begin{aligned}\beta_{\text{temp}} &:= -\frac{\beta^2\delta}{2(f(x^k+\beta d^k)-f(x^k)-\beta\delta)} = -\frac{\beta^2\delta}{2\beta\delta+\beta^2\langle Ad^k,d^k\rangle-2\beta\delta}\\ &= -\frac{\langle\nabla f(x^k),d^k\rangle}{\langle Ad^k,d^k\rangle} := \bar{\beta}\ .\end{aligned}$$

This is a simple Cauchy step-size. Since the vector d^k is the descent direction (2.7) and A is SPS, our optimal $\bar{\beta}$ is positive.
Obviously, the computation of new β_{temp} is independent of the previous value and original GLL method performs solely the bisection method, i.e. it tries to half the coefficient β and verify generalized Armijo condition. Furthermore, the value of step-size β has to be from interval $[\sigma_1,\sigma_2]\subseteq[0,1]$, since the smaller or larger value may cause the leaving the feasible set.

The division of step-size β by two now modifies only the generalized Armijo condition and the algorithm stops when the condition

$$f(x_{temp})\leq f_{\max}+\gamma\beta\delta$$

is fulfilled. This condition can be also simplified

$$\begin{aligned}0 &\geq f(x^k+\beta d^k)-f_{\max}-\gamma\beta\delta\\ &= f(x^k)+\beta\langle\nabla f(x^k),d^k\rangle+\tfrac{1}{2}\beta^2\langle Ad^k,d^k\rangle-f_{\max}-\gamma\beta\langle\nabla f(x^k),d^k\rangle\\ &= \tfrac{1}{2}\beta^2\langle Ad^k,d^k\rangle+(1-\gamma)\beta\langle\nabla f(x^k),d^k\rangle+f(x^k)-f_{\max}\\ 0 &\geq \tfrac{1}{2}\beta^2+(1-\gamma)\beta\tfrac{\langle\nabla f(x^k),d^k\rangle}{\langle Ad^k,d^k\rangle}+\tfrac{1}{\langle Ad^k,d^k\rangle}(f(x^k)-f_{\max})\ .\end{aligned}$$

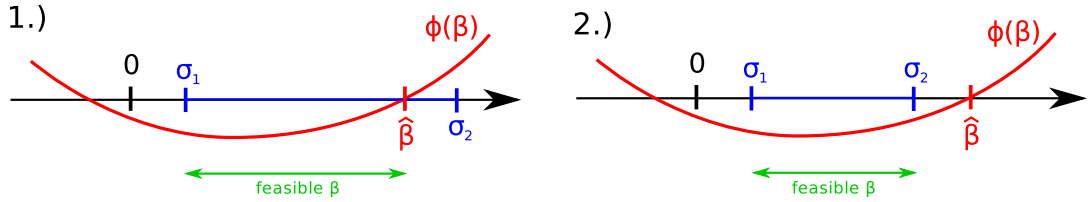


Figure 5: SPG-QP: Generalized Armijo condition and possible situations of step-size.

Afterwards, we denote the function on the right hand-side and the constant term by

$$\Phi(\beta) := \frac{1}{2}\beta^2 - (1-\gamma)\bar{\beta}\beta - \xi, \quad \xi := \frac{1}{\langle Ad^k, d^k\rangle}(f_{\max} - f(x^k)).$$

We are interested in β such that generalized Armijo condition in form

$$\Phi(\beta) \leq 0 \tag{2.9}$$

is fulfilled. The positive root of $\Phi(\beta)$ is given by

$$\hat{\beta} := (1-\gamma)\bar{\beta} + \sqrt{(1-\gamma)^2\bar{\beta}^2 + 2\xi}\ .$$

There exist only two possible situation, see Fig. 5.

Therefore, we can conclude that feasible step-size in the second step of SPG

$$\beta_k \in [\sigma_1, \min\{\sigma_2, \hat{\beta}\}].$$

This simple interval can replace GLL, i.e. any β_k from this interval fulfills the generalized Armijo condition.

The computation of the function values can be also simplified

$$f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle = \frac{1}{2}\langle g - b, x\rangle,$$

where $g := \nabla f(x) = Ax - b$. Finally, we can simplify the computation of BB step-length using $x^{k+1} = x^k + \beta_k d^k$

$$\alpha_{k+1} = \frac{\langle s^k, s^k\rangle}{\langle s^k, y^k\rangle} = \frac{\langle s^k, s^k\rangle}{\langle s^k, As^k\rangle} = \frac{\langle \beta_k d^k, \beta_k d^k\rangle}{\langle \beta_k d^k, \beta_k Ad^k\rangle} = \frac{\langle d^k, d^k\rangle}{\langle d^k, Ad^k\rangle}$$

and the recursive formula for the computation of new gradient

$$g^{k+1} := Ax^{k+1} - b = A(x^k + \beta_k d^k) - b = g^k + \beta_k Ad^k.$$

We use all the simplification to design the Algorithm 4. For the sake of simplicity, we relabel the coefficient $\gamma := 1 - \gamma \in (0, 1)$.

Notice that the most time-consuming operation - the multiplication by Hessian matrix A - is performed only once per iteration.

Algorithm 4: **Spectral projected gradient method for QP (SPG-QP).**

Given initial approximation $x^0 \in \Omega$, *parameters* $m \in \mathbb{N}, \gamma \in (0,1)$, *safeguarding parameter* $\sigma_2 \in (0,1)$, *and initial step-size* $\alpha_0 > 0$.

$k := 0$

$g^0 := Ax^0 - b$

$f^0 := 1/2\langle g^0 - b, x^0 \rangle$

while $\|\widetilde{g}_\alpha(x)\|$ *is not sufficiently small*

$d^k := P(x^k - \alpha_k g^k) - x^k$

$f_{\max} := \max\{f(x^{k-j}) : 0 \le j \le \min\{k, m-1\}\}$

$\xi := (f_{\max} - f^k)/\langle Ad^k, d^k \rangle$

$\bar{\beta} := -\langle g^k, d^k \rangle / \langle Ad^k, d^k \rangle$

$\hat{\beta} := \gamma\bar{\beta} + \sqrt{\gamma^2\bar{\beta}^2 + 2\xi}$

choose $\beta_k \in [\sigma_1, \min\{\sigma_2, \hat{\beta}\}]$

$x^{k+1} := x^k + \beta_k d^k$

$g^{k+1} := g^k + \beta_k Ad^k$

$f^{k+1} := 1/2\langle g^{k+1} - b, x^{k+1} \rangle$

$\alpha_{k+1} := \langle d^k, d^k \rangle / \langle Ad^k, d^k \rangle$

$k := k + 1$

endwhile

Return approximation of solution x^k.

2.2 Projected Barzilai-Borwein method (PBB)

The easiest way how to modify gradient descent method to solve constrained problem is to project every iteration onto the feasible set. We obtain

$$x^{k+1} = P_\Omega \left(x^k - \alpha_k \nabla f(x^k) \right) \ . \tag{2.10}$$

Using this simple technique, we obtain the sequence of approximations which lies in feasible set. If the sequence is converging to the point with the lowest function value, then this point will be also in feasible set, i.e. we obtain a solution of minimization problem with constraints. However, the basic property of BB method is non-monotonicity and this property is also typical for projected version.
The idea of the projected BB can be also seen from other point of view. The projection in prescription (2.10) can be formulated as optimization problem, see (1.12). We obtain

$$x^{k+1} = P_\Omega(x^k - \alpha_k g^k) = \arg\min_{x \in \Omega} \|(x^k - \alpha_k g^k) - x\| \ .$$

Using simple manipulations, this formula can be modified to the form

$$x^{k+1} = \arg\min_{x \in \Omega} f(x) - \frac{1}{2} \left(\frac{1}{\alpha_k} \|x - x^k\|^2 - \|x - x^k\|_A^2 \right) \ . \tag{2.11}$$

If we choose optimal $\alpha_k = \|x^{k+1} - x^k\|^2 / \|x^{k+1} - x^k\|_A^2$, we get

$$x^{k+1} = \arg\min_{x \in \Omega} f(x) \ ,$$

which is in fact the solution of the original QP problem (1). Unfortunatelly, value of x^{k+1} in the definition of optimal α_k is unknown. Probably, we can use the optimal value from the previous step. Afterwards, we obtain the prescription of PBB method.

PBB method is not convergent, there exist cases when the algorithm is cycling, i.e. algorithm generates the sequence of iterations which ends in the starting approximation, see Dai and Fletcher [23]. Thus we decided to enforce the convergence by simple fall-back strategy, see Algorithm 5.

The algorithm generates the PBB iterations until there is either the improvement of the cost function or there are K consecutive iterations without improvement. In the first case, an additional fixed step is carried out to achieve a sufficient decrease of the cost function, otherwise the next iteration is defined by the fixed step-length gradient projection step from the best of the last K iterations.
This algorithm was proposed by Pospíšil and Dostál [60].

Algorithm 5: **Projected Barzilai–Borwein algorithm with fall-back (PBBf).**

Choose $x^0, x^1 \in \Omega,\ \overline{\alpha} \in (0, 2/\lambda^A_{\max}\rangle, K \in \mathbb{N},$
set $k := 1, \hat{k} := 0, x^{\min} := x^1$
while $\|g^P_{\overline{\alpha}}(x^k)\|$ *is not small*
  $s := x^k - x^{k-1},\ \alpha_{bb} = s^T s / s^T A s$
  $x^{k+1} = P_\Omega\left(x^k - \alpha_{bb} g^k\right)$
  { Fall-back update}
  if $f(x^{k+1}) < f(x^{\min})$
    $x^{\min} := P_\Omega(x^{k+1} - \overline{\alpha} g^{k+1}),\ \hat{k} := 0$
  else
    $\hat{k} := \hat{k} + 1$
  endif
  {Fall-back application}
  if $\hat{k} \geq K$
    $x^{k+1} := P_\Omega(x^{\min} - \overline{\alpha} g^{\min})$
    $x^{\min} := x^{k+1},\ \hat{k} := 0$
  endif
  $k := k + 1$
end while
$\widehat{x} \approx x^k$

2.3 Accelerated projected gradient descent method (APGD)

Nesterov [53] developed black-box algorithm for solving more general optimization problems. This algorithm was called Accelerated Projected Gradient Descent method (APGD). It is based on two basic properties of the cost function - the convexity (1.10) and Lipschitz continuity (1.11). In algorithm, author uses the *mapping* gradient instead of *reduced* gradient (1.33), which is more common in optimization theory. Nevertheless, we show that these two gradient types are equivalent.

Definition 2.3.1
(Reduced and mapping gradient)

Let $\Omega \subset \mathbb{R}^n$ be a non-empty convex set, $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable function, and $\gamma, \alpha \in \mathbb{R}$ be positive non-zero constants.
Then for every $x \in \mathbb{R}^n$, we define

- reduced gradient *by prescription*

$$\widetilde{g}_\alpha(x) := \frac{1}{\alpha}(x - x_\alpha^P) \ ,$$

where

$$x_\alpha^P := P_\Omega(x - \alpha \nabla f(x))$$

and $P_\Omega : \mathbb{R}^n \to \Omega$ is a projection onto Ω.

- mapping gradient *by prescription*

$$g_{\Omega,\gamma}(x) := \gamma(x - x_{\Omega,\gamma}) \ ,$$

where

$$x_{\Omega,\gamma} := \arg\min_{y \in \Omega} \left[f(x) + \langle \nabla f(x), y - x \rangle + \frac{\gamma}{2}\|y - x\|^2 \right] \ . \tag{2.12}$$

In the next lemma, we present the equivalency of these two types of gradient in QP.

Lemma 2.3.1
(Equivalency of reduced and mapping gradient)

$$\forall x \in \mathbb{R}^n : \quad \widetilde{g}_\alpha(x) = g_{\Omega,\frac{1}{\alpha}}(x).$$

Proof: Let $\widetilde{g}_\alpha(x)$ and $g_{\Omega,\gamma}(x)$ be a mapping and projected gradient defined above. We should remind, that the argument of minima is independent of adding constants or multiplying whole cost function by positive constant. For the sake of simplicity, we denote by symbol $\propto$ the equality of minimizers of two functions, i.e.

$$f(x) \propto g(x) \quad \Leftrightarrow \quad \arg\min_{x\in\Omega} f(x) = \arg\min_{x\in\Omega} g(x).$$

Using this, we can write for any $y \in \mathbb{R}^n$

$$\begin{aligned}
f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{\gamma}{2}\|y-x\|^2 &\propto \langle \nabla f(x), y - x\rangle + \tfrac{\gamma}{2}\|y-x\|^2 \\
&\propto \langle \nabla f(x), y\rangle + \tfrac{\gamma}{2}\|y-x\|^2 \\
&\propto \langle \nabla f(x), y\rangle + \tfrac{\gamma}{2}\left(\langle y, y\rangle - 2\langle y, x\rangle + \langle x, x\rangle\right) \\
&\propto \langle \nabla f(x), y\rangle + \tfrac{\gamma}{2}\langle y, y\rangle - \gamma\langle y, x\rangle \\
&\propto \langle y, y\rangle + \tfrac{2}{\gamma}\langle \nabla f(x), y\rangle - 2\langle y, x\rangle \\
&\propto \langle y, y\rangle - 2\langle y, x - \tfrac{1}{\gamma}\nabla f(x)\rangle \\
&\propto \langle y, y\rangle - 2\langle y, x - \tfrac{1}{\gamma}\nabla f(x)\rangle + \langle x - \tfrac{1}{\gamma}\nabla f(x), x - \tfrac{1}{\gamma}\nabla f(x)\rangle \\
&\propto \|y - (x - \tfrac{1}{\gamma}\nabla f(x))\|^2 \\
&\propto \|y - (x - \tfrac{1}{\gamma}\nabla f(x))\|.
\end{aligned}$$

Afterwards, we can write

$$\begin{aligned}
\arg\min_{y\in\Omega}\left[f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{\gamma}{2}\|y-x\|^2\right] &= \arg\min_{y\in\Omega}\|y - (x - \tfrac{1}{\gamma}\nabla f(x))\| \\
&= P_\Omega(x - \tfrac{1}{\gamma}\nabla f(x)).
\end{aligned}$$

We can see the equality between mapping and projected gradient with $\gamma = \frac{1}{\alpha}$. □

The main idea of the method is the relation between function values, the norm of reduced gradient (Definition 2.3.1), Lipchitz constant (Definition 1.1.3), and the constant of strong convexity (Lemma 1.1.7). The next lemma presents the basic estimation.

Lemma 2.3.2
(Relation between L, μ, and $\widetilde{g}_\alpha$)

Let f be a Lipschitz continuous function and $\alpha \geq 1/L$. Then

$$\forall y \in \mathbb{R}^n \ \forall x \in \Omega : \quad f(x) \geq f(y^P) + \langle \widetilde{g}_\alpha(y), x - y \rangle + \frac{1}{2\gamma}\|\widetilde{g}_\alpha(y)\|^2 + \frac{\mu}{2}\|x - y\|^2 \ , \tag{2.13}$$

where

$$\begin{aligned} \widetilde{g}_\alpha(x) &:= \tfrac{1}{\alpha}(x - x_\alpha^P) \ , \\ x_\alpha^P &:= P_\Omega(x - \alpha \nabla f(x)) \ . \end{aligned} \tag{2.14}$$

Proof: See Nesterov [53]. The proof is the key idea, we decided to include it in the thesis.
At first, let us make some preparations. We define auxiliary function $\Phi : \mathbb{R}^n \to \mathbb{R}$ by

$$\Phi(x) := f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2\alpha}\|x - y\|^2 \ , \tag{2.15}$$

where $y \in \mathbb{R}^n$ is arbitrary, but fixed.
The gradient of the (2.15) is given by

$$\nabla \Phi(x) \colon = \nabla f(y) + \frac{1}{\alpha}(x - y) \ . \tag{2.16}$$

Furthermore, we can also write

$$\langle \nabla f(y) - \widetilde{g}_\alpha(y), x - y_\alpha^P \rangle = \langle \nabla \Phi(y_\alpha^P) - \frac{1}{\alpha}(y_\alpha^P - y) - \frac{1}{\alpha}(y - y_\alpha^P), x - y_\alpha^P \rangle = \langle \nabla \Phi(y_\alpha^P), x - y_\alpha^P \rangle \geq 0, \tag{2.17}$$

where we used (2.16) in form

$$\nabla \Phi(y_\alpha^P) = \nabla f(y) + \frac{1}{\alpha}(y_\alpha^P - y) \ .$$

The last inequality in (2.17) holds, because y^P is the minimizer of $\phi(x)$ (see Lemma 1.3.6 and Nesterov's definition of projection (2.12)).
From (2.17) we can easily get

$$\langle \nabla f(y), x - y_\alpha^P \rangle \geq \langle \widetilde{g}_\alpha(y), x - y_\alpha^P \rangle \ . \tag{2.18}$$

Using the definition of projected gradient in (2.14) we can write

$$\|\widetilde{g}_\alpha(y)\|^2 = \langle \widetilde{g}_\alpha(y), \widetilde{g}_\alpha(y) \rangle = \alpha^2 \|y - y_\alpha^P\|^2 \ \Rightarrow \ \|y - y_\alpha^P\|^2 = \alpha^2 \|\widetilde{g}_\alpha(y)\|^2. \tag{2.19}$$

Furthermore, using the definition of projected gradient in (2.14) we can write

$$\widetilde{g}_\alpha(y) = \frac{1}{\alpha}(y - y_\alpha^P) \Rightarrow y_\alpha^P = y - \alpha \widetilde{g}_\alpha(y). \tag{2.20}$$

Since $\frac{1}{\alpha} \geq L$, we can write

$$\forall y \in \mathbb{R}^n : \quad \Phi(y_\alpha^P) \geq f(y_\alpha^P) \ . \tag{2.21}$$

Now, we are ready to start with the main estimation. Function f is strongly convex, thus $\forall x, y \in \mathbb{R}^n$ holds

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|^2 \ ,$$

which can be modified to form

$$\begin{aligned}
f(x) - \tfrac{\mu}{2}\|x-y\|^2 \quad &\geq \quad f(y) + \langle \nabla f(y), x - y \rangle \\
&\quad \text{add smart zero} \\
&\quad = f(y) + \langle \nabla f(y), x - y \rangle + \langle \nabla f(y), y_\alpha^P \rangle - \langle \nabla f(y), y_\alpha^P \rangle \\
&\quad = f(y) + \langle \nabla f(y), y_\alpha^P - y \rangle + \langle \nabla f(y), x - y_\alpha^P \rangle \\
&\quad \text{estimate using (2.18)} \\
&\quad \geq f(y) + \langle \nabla f(y), y_\alpha^P - y \rangle + \langle \widetilde{g}_\alpha(y), x - y_\alpha^P \rangle \\
&\quad \text{use definition (2.15)} \\
&\quad = \Phi(y_\alpha^P) - \tfrac{1}{2\alpha}\|y_\alpha^P - y\|^2 + \langle \widetilde{g}_\alpha(y), x - y_\alpha^P \rangle \\
&\quad \text{use equation (2.19)} \\
&\quad = \Phi(y_\alpha^P) - \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y)\|^2 + \langle \widetilde{g}_\alpha(y), x - y_\alpha^P \rangle \\
&\quad \text{use equation (2.20)} \\
&\quad = \Phi(y_\alpha^P) - \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y)\|^2 + \langle \widetilde{g}_\alpha(y), x - y + \alpha\widetilde{g}_\alpha(y) \rangle \\
&\quad = \Phi(y_\alpha^P) + \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y)\|^2 + \langle \widetilde{g}_\alpha(y), x - y \rangle \\
&\quad \text{use equation (2.21)} \\
&\geq \quad f(y_\alpha^P) + \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y)\|^2 + \langle \widetilde{g}_\alpha(y), x - y \rangle.
\end{aligned}$$

□

In Nesterov [53], it can be found the whole derivation of the algorithm. Main idea of the algorithm, i.e. connection between the mapping gradient and the properties of cost function, is given by the next theorem.

Theorem 2.3.1
(Lipschitz continuous strongly convex function and projected gradient)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant L and strongly convex with constant of strong convexity μ.

Let $\{x_k\}$ be a sequence generated by prescription

$$\begin{aligned} x_0 &\in \Omega \ , \\ x_{k+1} &:= x_k - h\widetilde{g}_\alpha(x_k) \ , \end{aligned}$$

where the step-size of projected gradient $\alpha = \frac{1}{L}$ and $h \leq \frac{1}{L}$.
Then for any given k, we have

$$\|x_k - \bar{x}\|^2 \leq (1 - \mu h)^k \, \|x_0 - \bar{x}\|^2 \ ,$$

where $\bar{x} \in \Omega$ is a solution of the optimization problem

$$\bar{x} = \arg\min_{x \in \Omega} f(x) \ .$$

Proof: See Nesterov [53]. □

Instead of using constant step-size, Nesterov suggests to use estimate functions sequences.

Definition 2.3.2
(Estimate sequence)

Let $\mathcal{F}$ be a space of estimate functions.
A pair of sequences $\{\phi_k(x)\}_{k=0}^{\infty}, \phi_k \in \mathcal{F}$ and $\{\sigma_k\}_{k=0}^{\infty}, \sigma_k \in \mathbb{R}^+$ is an estimate sequence *of the cost function f if*

$$\begin{aligned} \sigma_k &\to 0, \\ \forall k \geq 0, \forall x \in \Omega: \quad \phi_k(x) &\leq (1 - \sigma_k) f(x) + \sigma_k \phi_0(x) \ . \end{aligned}$$

The next lemma demonstrates the effect of the estimate functions. It gives a condition of convergence of function values of approximations $\{x_k\}$ to the function value in the solution using estimate sequence.

Lemma 2.3.3
(Estimate sequence and the solution)

Let us suppose that for a sequence of approximations $\{x_k\}$

$$f(x_k) \leq \phi_k^{\min}, \qquad \phi_k^{\min} := \min_{x \in \mathbb{R}^n} \phi_k(x). \tag{2.22}$$

Then

$$f(x_k) - f(\bar{x}) \leq \sigma_k(\phi_0(\bar{x}) - f(\bar{x})) \to 0 \ .$$

Proof: See Nesterov [53]. The lemma is a key ingredient of the algorithm, thus we decided to introduce it in the thesis.
Assume that proposition (2.22) holds. We can write

$$\begin{aligned} f(x_k) \quad &\leq \quad \min_{x \in \Omega} \phi_k(x) \\ &\quad \text{using the definition of estimate sequence (2.3.2)} \\ &\leq \min_{x \in \Omega} [(1 - \sigma_k) f(x) + \sigma_k \phi_0(x)] \\ &\quad \bar{x} \text{ is solution, so } \forall x \in \Omega : f(x) \geq f(\bar{x}) \\ &\leq (1 - \sigma_k) f(\bar{x}) + \sigma_k \phi_0(\bar{x}) \\ &= f(\bar{x}) + \sigma_k(\phi_0(\bar{x}) - f(\bar{x})). \end{aligned}$$

Notice that $(\phi_0(\bar{x}) - f(\bar{x}))$ is independent of k, i.e. constant. If $\sigma_k \to 0$, then

$$\sigma_k(\phi_0(\bar{x}) - f(\bar{x})) \quad \to \quad 0 \ .$$

□

We can define estimate sequence in a special form based on the estimation term. See the lemma below.

Lemma 2.3.4
(Special choice of estimate sequence)

Let

- $\phi_0(x) \in \mathcal{F}$ *is arbitrary function,*
- $\{y_k\}_{k=0}^{\infty}$ *is arbitrary sequence in* Ω*,*
- $\{\alpha_k\}_{k=0}^{\infty} : \quad \alpha_k \in (0,1), \sum_{k=0}^{\infty} \alpha_k = \infty,$
- $\sigma_0 = 1.$

Then the pair of sequences $\{\phi_k(x)\}_{k=0}^{\infty}, \{\sigma_k\}_{k=0}^{\infty}$ *defined by*

$$\begin{aligned} \sigma_{k+1} &:= (1-\alpha_k)\sigma_k \\ \phi_{k+1}(x) &:= (1-\alpha_k)\phi_k(x) \\ &\quad + \alpha_k \left[f(y_{k,\alpha}^P) + \langle \widetilde{g}_\alpha(y_k), x - y_k \rangle + \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y_k)\|^2 + \tfrac{\mu}{2}\|x - y_k\|^2 \right] \end{aligned} \tag{2.23}$$

is estimate sequence.

Proof: See Nesterov [53]. The lemma shows how the basic estimation of cost function based on the norm of projected gradient, Lipschitz constant, and constant of strong convexity, influence the form of estimate function. We decided to include it in the thesis. At first, we show that for $k = 0$ is ϕ_k, σ_k holds the inequality (2.3.2), i.e.

$$\phi_0(x) \leq (1-\sigma_0)f(x) + \sigma_0\phi_0(x) \ .$$

This inequality in fact holds, because left side is equal to ϕ_0 after setting $\sigma_0 := 1$.

Let us presume, that (2.3.2) holds for $\phi_k, \sigma_k, k \geq 0$, i.e.

$$\phi_k(x) \leq (1-\sigma_k)f(x) + \sigma_k\phi_0(x) \ . \tag{2.24}$$

And now we try to estimate $\phi_{k+1}(x)$ and prove also the same inequality for ϕ_{k+1}, σ_{k+1}. At first, from (2.24) we can easily get

$$\phi_k(x) - (1-\sigma_k)f(x) \leq \sigma_k\phi_0(x) \ . \tag{2.25}$$

And now, we start with formulae (2.23)

$$\begin{aligned} \phi_{k+1}(x) &= (1-\alpha_k)\phi_k(x) + \alpha_k \left[f(y_k^P) + \langle \widetilde{g}_\alpha(y_k), x - y_k \rangle + \tfrac{\alpha}{2}\|\widetilde{g}_\alpha(y_k)\|^2 + \tfrac{\mu}{2}\|x - y_k\|^2 \right] \\ &\quad \text{we can use (2.13)} \\ &\quad \leq (1-\alpha_k)\phi_k(x) + \alpha_k f(x) \\ &\quad \text{add smart zero} \\ &\quad = (1-\alpha_k)\phi_k(x) + \alpha_k f(x) + (1 - 1 + \sigma_k - \sigma_k + \alpha_k\sigma_k - \alpha_k\sigma_k)f(x) \\ &\quad = (1 - (1-\alpha_k)\sigma_k)f(x) + (1-\alpha_k)(\phi_k(x) - (1-\sigma_k)f(x)) \\ &\quad \text{using (2.25) and } \alpha_k \in (0,1) \\ &\quad \leq (1 - (1-\alpha_k)\sigma_k)f(x) + (1-\alpha_k)\sigma_k\phi_k(x) \\ &\quad \text{using the recursive formulae for } \sigma_{k+1} \\ &\quad = (1-\sigma_{k+1})f(x) + \sigma_{k+1}\phi_k(x). \end{aligned}$$

□

Nesterov uses tricky simplifications and substitutions to obtain as simple algorithm as possible. At first, he defines the estimate sequence which uses the minimizers.

Lemma 2.3.5
(Substitution and simplification of APGD algorithm)

Let the first estimate function have a form

$$\phi_0(x) \colon = \phi_0^{\min} + \frac{\gamma_0}{2}\|x - v_0\|^2 \ ,$$

where $\phi_0^{\min} \in \mathbb{R}, v_0 \in \mathbb{R}^n$.
Then the process (2.23) *is equivalent to*

$$\phi_k(x) \colon = \phi_k^{\min} + \frac{\gamma_k}{2}\|x - v_k\|^2 \ ,$$

where

$$\begin{aligned}
\gamma_{k+1} &: = (1-\alpha_k)\gamma_k + \alpha_k\mu \ , \\
v_{k+1} &: = \tfrac{1}{\gamma_{k+1}}\left[(1-\alpha_k)\gamma_k v_k + \alpha_k\mu y_k - \alpha_k \widetilde{g}_\alpha(y_k)\right] \quad = \arg\min_x \phi_{k+1}(x) \ , \\
\phi_{k+1}^{\min} &: = (1-\alpha_k)\phi_k^{\min} + \alpha_k f(y_k^P) + \left(\tfrac{\alpha_k}{2L} - \tfrac{\alpha_k^2}{2\gamma_{k+1}}\right)\|\widetilde{g}_\alpha(y_k)\|^2 \\
&\quad + \tfrac{\alpha_k(1-\alpha_k)\gamma_k}{\gamma_{k+1}}\left(\tfrac{\mu}{2}\|y_k - v_k\|^2 + \langle \widetilde{g}_\alpha(y_k), v_k - y_k\rangle\right) \ .
\end{aligned}$$

Finally, if we choose

$$\begin{aligned}
\gamma_{k+1} &: = (1-\alpha_k)\gamma_k + \alpha_k\mu \ , \\
v_{k+1} &: = \tfrac{1}{\gamma_{k+1}}\left[(1-\alpha_k)\gamma_k v_k + \alpha_k\mu y_k - \alpha_k \widetilde{g}_{\alpha_k}(y_k)\right] \quad = \arg\min_x \phi_{k+1}(x) \ , \\
\phi_{k+1}^{\min} &: = (1-\alpha_k)\phi_k^{\min} + \alpha_k f(y_k^P) + \left(\tfrac{\alpha_k}{2L} - \tfrac{\alpha_k^2}{2\gamma_{k+1}}\right)\|\widetilde{g}_{\alpha_k}(y_k)\|^2 \\
&\quad + \tfrac{\alpha_k(1-\alpha_k)\gamma_k}{\gamma_{k+1}}\left(\tfrac{\mu}{2}\|y_k - v_k\|^2 + \langle \widetilde{g}_{\alpha_k}(y_k), v_k - y_k\rangle\right)
\end{aligned}$$

and make some other tricky simplifications, we obtain Algorithm 6.

Unfortunately, the main disadvantages of the algorithm are the estimations of Lipchitz constant and the constant of convexity. If the quadratic cost function has SPD Hessian, then the constant of convexity is given by the smallest eigenvalue, see the remark below Lemma 1.1.7. If the Hessian matrix is SPS, then the smallest eigenvalue is equal to zero, thus we choose in both cases $\mu = 0$. The solution of quadratic equation in Algorithm 6 is given by

$$\Theta_{k+1} = \frac{1}{2}(\Theta_k + \sqrt{\Theta_k^2 + 4\Theta_k}).$$

Algorithm 6: **Accelerated projected gradient descent method (APGD).**

Given cost function f, initial approximation x_0, projection onto feasible set $P_\Omega(x)$, accuracy $\varepsilon > 0$.

$k := 0$

$\Theta_0 := 1$

$y^0 := x^0$

while $\|\widetilde{g}_{1/L}(x^k)\|_2 > \varepsilon$

$x^{k+1} := P_\Omega(y^k - 1/L\nabla f(y^k))$

Θ_{k+1} *solves* $\Theta_{k+1}^2 = (1 + \Theta_{k+1})\Theta_k + \Theta_k \frac{\mu}{L}$

$\beta_k := \frac{\Theta_k(1-\Theta_k)}{\Theta_k^2+\Theta_{k+1}}$

$y^{k+1} := x^{k+1} + \beta_{k+1}(x^{k+1} - x^k)$

$k := k + 1$

endwhile

Return approximation of solution x_k.

To find appropriate L, Heyn [47] proposes the line-search algorithm similar to GLL algorithm. He sets initial

$$L_0 = \frac{\|\nabla f(x^0) - \nabla f(x^1)\|^2}{\|x^0 - x^1\|^2} = \frac{\|x^0 - x^1\|_A^2}{\|x^0 - x^1\|^2}$$

with arbitrary $x^0, x^1 \in \mathbb{R}^n, x^0 \neq x^1$. In every iteration of APDG, Heyn runs the backtracking algorithm to find L_k such that

$$\begin{aligned} f(x^{k+1}) &\leq f(y^k) + \langle \nabla f(y^k), x^{k+1} - y^k \rangle + \tfrac{L_k}{2}\|x^{k+1} - y^k\|^2, \\ x^{k+1} &= y^P_{k,1/L_k} = P_\Omega(y^k - 1/L_k \nabla f(y^k)) \end{aligned} \tag{2.26}$$

and using this, he preserves the convergence of the algorithm. In this thesis, we simplify this backtracing algorithm.
In (1.8) we set $y := x^{k+1}, z := y^k$ and write the inequality (2.26) in form

$$\frac{\left\langle \nabla f(x^{k+1}) + \nabla f(y^k), x^{k+1} - y^k \right\rangle}{2} \leq \langle \nabla f(y^k), x^{k+1} - y^k \rangle + \frac{L_k}{2}\|x^{k+1} - y^k\|_2^2$$

and after straightforward manipulations, we obtain the estimation

$$\frac{\|x^{k+1} - y^k\|_A^2}{\|x^{k+1} - y^k\|^2} \leq L_k.$$

Using Lemma 1.1.5 we can simply set

$$L_k = \lambda_{\max}^A.$$

In our numerical experiments with SPS Hessian matrix, we set $\mu = 0$ and $L = \lambda_{\max}$. Unfortunately, this choice is suitable only for some cases, see for instance the discussion in the end of Section 3.3.5.

2.4 Active-set methods (MPGP,MPGPS)

In this section, we present a basic schema of all active-set algorithms proposed by Dostál et al. [26, 34, 36, 32, 33, 31, 30, 16, 15, 29]. The basic version was proposed independently by Dostál [25] and Friedlander and Martínez [40] and can be considered as a modification of the Polyak algorithm. Dostál and Schöberl in [36] combine the proportioning algorithm with the gradient projections [65], they use the constant $\Gamma > 0$, the test to decide about leaving the face, and three types of steps to generate the sequence of iterates x^k that approximate the solution.

Algorithm 7: **Modified Proportioning with Gradient Projection (MPGP).**

Choose $x^0 \in \Omega$
for $k = 0, 1, 2, \ldots$ *(while a stopping criterion is not achieved)*
  if proportioning condition *is satisfied*
    CG step *or* CG halfstep
      make one CG step to solve problem on free set using free gradient
      if this step means leaving Ω*, do only a half-step and restart CG*
  else
    gradient projection step
      make one gradient projection step and restart CG on free set
  endif
  $k := k + 1$
endfor

The algorithm is based on using the free, chopped, and projected gradients to minimize the cost function on the free set and afterwards on the active set. The switching between these processes is realized by the proportioning condition.

The CG step of algorithm presents the original conjugate gradient method step (CG method was firstly presented by Hestenes and Stiefel [46]). The algorithm constructs the Krylov subspace and conjugate directions p^k from the free gradients

$\varphi(x^k)$ using prescriptions

CG step	
step-length	$\alpha_{cg} = \varphi(x^k)^T p^k / (p^k)^T A p^k$
new approximation	$x^{k+1} = x^k - \alpha_{cg} p^k$
new gradient	$g^{k+1} = g^k - \alpha_{cg} A p^k$
orthogonalization coefficient	$\beta_{cg} = \varphi(x^{k+1})^T A p^k / (p^k)^T A p^k$
new conjugate direction	$p^{k+1} = \varphi(x^{k+1}) - \beta_{cg} p^k$

In the case that the free set is changed, it is neccessary to start building the Krylov space from the begining, e.g. restart CG method on free set with

$$p^{k+1} = \varphi(x^{k+1}) \ .$$

The performance of MPGP is improved by enhancing the feasible half-step introduced in Dostál and Schöberl [36]. This modification of MPGP algorithm is based on a simple observation that the gradient can be updated at any point on the conjugate gradient path without a matrix–vector multiplication, so that the gradient projection can be carried out at nearly the same cost from the nearest point of the boundary of feasible set to the current iteration in the conjugate gradient direction rather than from the current iteration.

CG halfstep	
step-length	$\alpha_f = \max\{\alpha \in \mathbb{R} : x^k - \alpha p^k \in \Omega\}$
new approximation	$x^{k+1/2} = x^k - \alpha_f p^k$
new gradient	$g^{k+1/2} = g^k - \alpha_f A p^k$
projection step	$x^{k+1} = P_\Omega(x^{k+1/2} - \bar{\alpha} g^{k+1/2})$
new gradient	$g^{k+1} = A x^{k+1} - b$

In the algorithm, we use a projection step with the constant step-length $\bar{\alpha} \in (0, \|A\|^{-1})$. The reason is given by Theorem 1.5.1. Moreover, if the feasible set is subsymmetric, we can use Theorem 1.5.2 and the constant step-length $\bar{\alpha} \in (0, 2\|A\|^{-1})$. Since the analysis is based on the worst case analysis, the implementation of the feasible half-step does not result in improving the error bounds, but it improves the performance of MPGP due to the additional decrease of the cost function obtained just for a few scalar products.

Now we are ready to present the basic and modified variants of Algorithm 7.

2.4.1 Modified Proportioning with Gradient projection (MPGP)

This algorithm was developed to solve QP problems on any closed convex feasible set. It was proposed by Dostál et al. [26, 30] and it is based on Algorithm 7 with the *stopping criterion* based on the norm of projected gradient (see Lemma 1.4.1)

$$\|g^P(\tilde{x})\| \leq \varepsilon\|b\| \ , \tag{2.27}$$

the *proportioning condition*

$$\|\beta(x^k)\| \leq \Gamma\|\varphi(x^k)\| \ , \tag{2.28}$$

where $\Gamma > 0$ is a constant parameter of algorithm. The *gradient projection step* is based on the constant step-length (see Theorem 1.5.1 and Theorem 1.5.2)

$$x^{k+1} = P_\Omega(x^k - \bar{\alpha}g^k) \ . \tag{2.29}$$

2.4.2 Modified Proportioning with Barzilai-Borwein gradient projections (MPGP-BB)

Constant step-length always guarantees the descent of cost function, see Theorem 1.5.1 and Theorem 1.5.2. However, the numerical experiments show that nonmonotone stategies could decrease the number of *projection steps*. For instance, the modification with PBB step given by

$$x^{k+1} := P_\Omega(x^k - \alpha_k^{BB}\nabla f(x^k)), \quad \alpha_k^{BB} := \frac{\langle s^k, s^k\rangle}{\langle As^k, s^k\rangle}, \quad s^k := x^k - x^{k-1} \tag{2.30}$$

was presented by Pospíšil [58]. This algorithm was inspired by the SPG, which uses the similar type of steps, see Section 2.1. The main drawback of the presented MPGP-BB algorithm is the absence of the proof of convergence. The PBB method is non-monotone and difficult to analyze. Therefore, the SPG method is using an additional line-search method to control the descent of the cost function to achieve the global convergence. In our algorithm, we tried to omit this line-search. However, this control can be realized using the fallback strategy presented in Section 2.2. In Pospíšil and Dostál [59], we present numerical experiments of MPGP-BB for solving QP problem with separable conical constraints in particle dynamics.

2.4.3 Modified Weak Proportioning with Gradient projections (MwPGP)

In Bouchala et al. [15], we present a modification of the original MPGP for solving the QP problems on feasible sets with strong curvature, such as ill-conditioned ellipses. Instead of using the projected gradient, we propose to use reduced projected gradient. We use the *stopping criterion* given by

$$\|\widetilde{g}_{\bar{\alpha}}^P(\tilde{x})\| \leq \varepsilon\|b\| \ , \tag{2.31}$$

the *proportioning condition*

$$2\delta\|\widetilde{g}^P_{\bar{\alpha}}(x^k)\|^2 \leq \|\varphi(x^k)\|^2 \ , \tag{2.32}$$

where $\delta \in (0, 1/2\rangle$ is a constant parameter of algorithm. The *gradient projection step* based on the constant step-length (2.29) $\bar{\alpha} \in (0, 2\|A\|^{-1})$.

The main theoretical results concerning the MwPGP algorithm are the subject of the following theorem.

Theorem 2.4.1
(Convergence of MwPGP)

Let Ω be a closed convex set, let $\widehat{x}$ denote the unique solution of (1) *with subsymmetric feasible set, let λ_1 denote the smallest eigenvalue of A, and let x^k be generated by Algorithm MwPGP with $x^0 \in \Omega$, $\bar{\alpha} \in (0, 2\|A\|^{-1})$, and $\delta \in (0, 1/2\rangle$. Let $C \geq 0$ denote the constant introduced in Lemma 1.4.5. Then the following statements hold:*
(i) If $0 < \alpha \leq \|A\|^{-1}$, then for any $k \geq 0$

$$f\Big(x^{k+1}\Big) - f(\widehat{x}) \leq \eta(\alpha)\Big(f(x^k) - f(\widehat{x})\Big), \tag{2.33}$$

and

$$\|g^P(x^k)\|^2 \leq \frac{2\Big(1+\eta\Big)}{\widehat{\alpha}\Big(1-\eta\Big)}\eta^k\Big(f(x^0) - f(\widehat{x})\Big), \tag{2.34}$$

where

$$\eta = \eta(\alpha) = 1 - \delta C^{-2}\alpha\lambda_1, \quad \widehat{\alpha} = 2\|A\|^{-1} - \alpha.$$

(ii) If Ω is subsymmetric and $\|A\|^{-1} \leq \alpha \leq 2\|A\|^{-1}$, then (2.33) *and* (2.34) *hold with*

$$\eta = \eta(\alpha) = 1 - \frac{1}{2}\delta C^{-2}\widehat{\alpha}\lambda_1.$$

Proof: See Bouchala et al. [15]. □

2.4.4 Modified Proportioning with Reduced Gradient projections (MPRGP)

If the feasible set is described only using bound constraints, the *projection step* can be simplified to the form of Steepest Descend method with chopped gradient, e.g.

$$\begin{array}{ll} \textit{MPRGP proportioning step} & \\ \textbf{step-length} & \alpha_{SD} = (g^k)^T\beta(x^k)/\beta(x^k)^T A\beta(x^k) \\ \textbf{new approximation} & x^{k+1} = x^k - \alpha_{SD}\beta(x^k) \\ \textbf{new gradient} & g^{k+1} = g^k - \alpha_{SD}A\beta(x^k) \end{array} \tag{2.35}$$

This algorithm was presented by Dostál et al. [26]. Furthermore, the theory can be extended to QP with box constraints, see Dostál [25].

2.4.5 Modified Proportioning with Reduced Gradient Projections for the SPS Hessian (MPRGPS)

MPRGP algorithm was developed to solve the problems with a SPD Hessian matrix. The recent generalization to the problems with symmetric positive semidefinite Hessian suggests only one difference from the original algorithm, specifically a test of the problem solvability

Control the solvability

if $\min\{\alpha_f, \alpha_{cg}\} = \infty$, then the problem has no solution.

The coefficient α_f is the maximal feasible step-size and α_{cg} is a coefficient of the conjugate gradient computed from the free gradient. If both coefficients are equal to infinity, then the problem has no solution. The theory and numerical experiments will be published in Dostál and Pospíšil [35].

2.5 Augmented Lagrangian method (SMALSE-M)

The algorithms for solving the minimization problems with separable inequalities described in previous sections can be plugged into SMALSE-M (Semimonotonic Augmented Lagrangian for separable and equality constrained QP) algorithm for the solution of the problem to find the minimizer of a convex quadratic function subject to separable convex inequality and linear equality constraints (1.36).
Let us denote the Lagrange multipliers corresponding to inequality constraints and inequality constraints by λ_I and λ_E, respectively (see Lemma 1.3.7). The SMALSE-M is a Uzawa-type algorithm which generates the approximations for λ_E in the outer loop and solves auxiliary problems with inequality constraints in the inner loop.
Let us remind the augmented Lagrangian (1.49) for problem (1.36)

$$L(x, \lambda_E, \rho) = \frac{1}{2} x^T (A + \rho B^T B) x - (b - B^T \lambda)^T x.$$

SMALSE-M is closely related to the earlier work of Friedlander and Santos with the present author [27]. Application of the update rule of SMALSE for the penalty parameter again results in convergence of the feasibility error that is independent of the conditioning of the equality constraints. Let us recall that the basic scheme that we use was proposed by Conn, Gould and Toint [20], who adapted the augmented Lagrangian method to the solution of problems with a general cost function subject to general equality constraints and simple bounds.

Algorithm has been proved to be well defined, that is, any convergent algorithm for the solution of the auxiliary problem required in Step 1 will generate either x^k that satisfies (2.36) in a finite number of steps or a sequence of approximations that converges to the solution of QP problems with inequalities and linear equalities (1.36). The basic theoretical results concerning this algorithm are very similar to those for SMALE, it is just enough to replace the gradient g by the projected gradient g^P.

The optimality of the algorithm was shown by Dostál [26] and Dostál and Kozubek [30].

The algorithm was a key ingredient in the development of of scalable algorithms for contact problems by Dostál et al. [32], [33], Sadowská et al.[63], and contact shape optimization problems by Vondrák et al. [68]. More information about this algorithm can be found also in Dostál [26] and Horák [50]. The review can be found also in Dostál and Pospíšil [34].

Algorithm 8: **SMALSE–M**

Given $\eta > 0$, $\beta \in (0,1)$, $M_0 > 0$, $\rho > 0$, *and* $\lambda_E^0 \in \mathbb{R}^m$.

for $k = 0, 1, 2, \ldots$

Step 1. Inner iteration with adaptive precision control.

Find $x^k \in \Omega_I$ *such that*

$$\|g^P(x^k, \lambda_E^k, \rho)\| \leq \min\{M_k \|Bx^k\|, \eta\} \tag{2.36}$$

Step 2. Update Lagrange multipliers.

$$\lambda_E^{k+1} = \lambda_E^k + \rho B x^k$$

Step 3. Update M *provided the increase of the Lagrangian is not sufficient.*

if $k > 0$ *and*

$$L(x^k, \lambda_E^k, \rho) < L(x^{k-1}, \lambda_E^{k-1}, \rho) + \frac{\rho}{2}\|Cx^k\|^2$$

then

$$M_{k+1} = \beta M_k,$$

else

$$M_{k+1} = M_k.$$

endfor

$\widehat{x} \approx x^k$

3 Applications

We present the results of solving practical applications using algorithms presented in Section 2. In the first part of this section, we study the efficiency on solution of geometric problems, which leads to QP problems. Since these optimization problems has simple structure, our results are easy to reproduce. The second part of the section is dedicted to the solution of linear elasticity contact problems. The algorithms for solving such a problems have long history in the Department of Applied Mathematics at VŠB-Technical University of Ostrava and they are based on the work of Dostál. In the thesis, we present a numerical results performed on Anselm supercomputer. In the third part, we are interested in the solution of multi-body dynamics system. This part was motivated by results obtained by Negrut and his team from SBEL University of Wisconsin-Madison.

During the comparison of algorithms, we are interested mostly in

- *the descend of stopping criterion* - the norm of the projected gradient reflects the convergence of approximations to the solution of the problem,
- *computing time* - better algorithm solves the problem faster; this time depends on the hardware specification, we compare computing time of the algorithms on the same machine and in the same programming environment, so this is not important for the comparision,
- *number of iterations* - faster algorithm performs the lower number of iterations; nevertheless, the resulting time depends also on the complexity of one iteration,
- *number of Hessian multiplications* - the multiplication by Hessian matrix is the most time-consuming operation in QP; in some algorithms, this operation can be performed different times during different types of iterations, it better reflexes the computing complexity of the algorithm than the number of iterations,

with respect to problem properties such as

- *problem size* - the number of unkowns defines the basic complexity of the problem,
- *spectral properties of Hessian matrix* - the speed of convergence of gradient methods usually depends on the condition number and distribution of eigenvalues,

- *norm of the right-hand side vector* - our stopping criteria reflects the norm of this vector.

Short notice about performance profiling

To compare the efficiency of algorithms, we study and use the performance profiling introduced by Dolan and Moré [24]. Authors present *performance profile* for the solver as cumulative distribution function for a performance metric. Let us denote the set of all problems and the set of all solvers by $\mathcal{P}$ and by $\mathcal{S}$, respectively. For each problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$, we define

$$t_{p,s} := \text{ value of performance measure required to solve problem } p \text{ by solver } s$$

and *performance ratio*

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

If and only if the solver s does not solve the problem p, then we set the maximum value $r_{p,s} = r_{\max} < \infty$. Accordingly, we obtain $r_{p,s} \leq r_{\max}$ for all problems and solvers. Furthermore, we define $\rho_s(\tau)$ as the probability for solver s that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio, i.e.

$$\rho_s(\tau) := \frac{1}{|\mathcal{P}|} \, |\{p \in \mathcal{P} : r_{p,s} \leq \tau\}|.$$

The function $\rho_s : \mathbb{R} \to [0, 1]$ is the cumulative distribution function for the performance ratio. The solvers with large probability $\rho_s(\tau)$ is to be preferred.

Example 3.0.1

To demostrate the algorithm profiling, we solve a simple benchmark of the system of linear equations resulting from the Finite Element method (FEM) discretization of the 1D string deflection. We consider a string on interval $\langle 0, 1\rangle$ with homogeneous Dirichlet boundary conditions, see Fig. 6. The problem in continuous form is

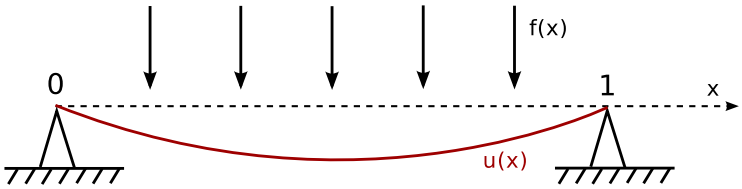


Figure 6: Benchmark - the deflection of the string.

described by Cauchy problem with 1D Laplace equation

$$\begin{aligned} -u''(x) &= f(x), \quad x \in [0,1], \\ u(0) = u(1) &= 0, \end{aligned}$$

where u is unknown displacement (the deflection of the string) and $f(x) := 1$ is a density of applied force. Using Galerkin method and FEM for discretization, we obtain a system of linear equations with the dimension based on the discretization parameter. The set of the problems $\mathcal{P}$ is defined by several choices of this parameter $n = 5, 10, 15, \ldots 100$. We compared Conjugate Gradient method (CG), Barzilai-Borwain method (BB), and Steepest Descent (SD) method to solve the problems with initial approximation $x_0 := 0$, relative precision $\varepsilon = 10^{-4}$, and maximum iterations 2000. The numbers of iterations can be found in Table 9.

n	5	10	15	20	25	30	35	40	45	50	55	...
CG	2	4	7	9	12	14	17	19	22	24	27	...
BB	6	28	51	88	108	99	181	171	109	329	207	...
SD	9	130	358	656	1076	1566	2000	2000	2000	2000	2000	...

Table 9: The number of iterations for solving system of linear equations in string benchmark.

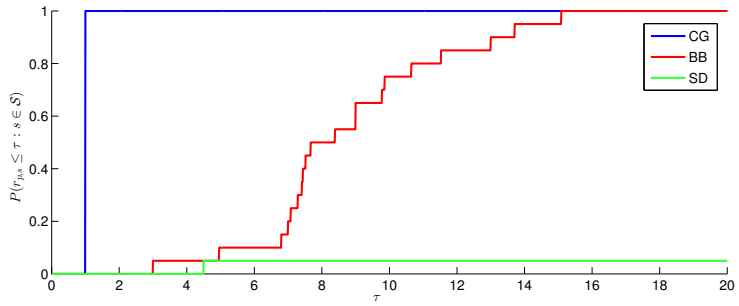


Figure 7: The number of iterations for solving system of linear equations in string benchmark - performance profile on $[0, 20]$.

We utilize the results from the table to performance profiles, see Fig. 7. The results show nothing revolutionary - the best algorithm in this case is CG. Nevertheless, the profiles can reveal the additional information. At first, we should remark

that the profile ignores the dependency of the number of iterations on dimension of the problem, the objective of profiling is *only* the ratio between the number of iterations of given solver and the best solver for each problem. It is constructed in a such way, that each problem has the same weight in the comparison. For instance, if we choose $\hat{\tau} = 9$, then the function values of cumulative distribution functions $\rho_{CG}(\hat{\tau}) = 1.0, \rho_{BB}(\hat{\tau}) = 0.65, \rho_{SD}(\hat{\tau}) = 0.05$ testify e.g. that

- the best algorithm (subject to the number of iterations) is CG. The ratio between the CG iterations and the best algorithm for given problem is equal to 1 for all problems, which means that also the cumulative distribution function ρ_{CG} is equal to one.
- there is 65% of all problems, which are solved by BB using the number of iterations $\hat{k}$, such that the ratio between $\hat{k}$ to solve problem p and the number of iterations necessary to solve p by the best algorithm (CG), is lower than $\hat{\tau} = 9$. Using the other words, 65% of the problems is solved using maximally 9-times larger number of iterations then the best algorithm (CG).
- $1-\rho_s(\tau)$ is the fraction of problems that the solver cannot solve within a factor $\hat{\tau}$ of the best solver, including problems for which the solver fails. This means that there exists 95% problems, which were solved at least 9-times slower by SD then by the best algorithm (CG).
- for our comparison, it is not important that SD method was stopped after 2000 iterations.

■

3.1 Geometric optimization

A lot of QP problems naturally arises in geometric problems. The reason is that every problem of minimization of a norm can be equivalently formulated as QP problem, i.e.

$$\arg\min_{x\in\Omega} \|x-a\| = \arg\min_{x\in\Omega} \langle x-a, x-a\rangle = \arg\min_{x\in\Omega} \frac{1}{2}x^T I x - a^T x \ .$$

Usually, such problems are solved by combinatorial algorithms. In this section, we present how to solve selected problems using iterative QP algorithms presented in Section 2. We are motivated by recent dissertation thesis by Schönherr [66]. The author compares several methods, but none of them has such optimal properties like the methods presented here. The efficiency of the algorithms is presented and demostrated on the numerical experiments.

3.1.1 Polytope distance

Definition 3.1.1
(Convex polytope.)

Convex polytope $\mathbf{S}$ *is the set of all convex combinations of a finite point set* $\mathcal{S}$*, i.e.*

$$\mathbf{S} := \left\{ v \in \mathbb{R}^n : v := \sum_{i=1}^{|\mathcal{S}|} \alpha_i v_i; \forall i = 1, \ldots, |\mathcal{S}| : \alpha_i \geq 0, v_i \in \mathcal{S}, \sum_{i=1}^{|\mathcal{S}|} \alpha_i = 1 \right\}.$$

Let us consider two convex polytopes $\mathbf{P}, \mathbf{Q}$ in $\mathbb{R}^d$ desribed by the boundary point sets

$$\begin{aligned} \mathcal{P} &:= \{p_1, \ldots, p_{n_p}\} \subset \mathbb{R}^d \ , \\ \mathcal{Q} &:= \{q_1, \ldots, q_{n_q}\} \subset \mathbb{R}^d \ . \end{aligned}$$

The problem is to find the shortest distance between these two objects

$$\min_{p \in \mathcal{P}, q \in \mathcal{Q}} \|p - q\| \ . \tag{3.1}$$

Every interior point of the convex polytope can be expressed as a convex linear combination of given points in sets $\mathcal{P}$ and $\mathcal{Q}$

$$\begin{aligned} \forall p \in \mathbf{P} \ \exists \alpha_1, \ldots, \alpha_{n_p} \in \mathbb{R} : p = \sum_{i=1}^{n_p} \alpha_i p_i \ , \\ \text{where } \sum_{i=1}^{n_p} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq 1 \ \forall i = 1, \ldots, n_p \ , \\ \forall q \in \mathbf{Q} \ \exists \beta_1, \ldots, \beta_{n_q} \in \mathbb{R} : q = \sum_{i=1}^{n_q} \beta_i q_i \ , \\ \text{where } \sum_{i=1}^{n_q} \beta_i = 1 \text{ and } 0 \leq \beta_i \leq 1 \ \forall i = 1, \ldots, n_q \ . \end{aligned} \tag{3.2}$$

We denote

$$\begin{aligned} y &:= [\alpha_1, \ldots, \alpha_{n_p}, \beta_1, \ldots, \beta_{n_q}]^T \in \mathbb{R}^{n_p + n_q} \ , \\ C &:= [p_1, \ldots, p_{n_p}, -q_1, \ldots, -q_{n_q}] \in \mathbb{R}^{d, n_p + n_q} \ , \\ B &:= \begin{bmatrix} 1 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & \ldots & 1 \end{bmatrix} \in \mathbb{R}^{2, n_p + n_q} \ , \\ c &:= [1, 1]^T \in \mathbb{R}^2. \end{aligned}$$

Afterwards, the cost function can be reformulated

$$\|p - q\| = \left\| \sum_{i=1}^{n_p} \alpha_i p_i - \sum_{i=1}^{n_q} \beta_i q_i \right\| = \|Cy\|$$

and feasible set conditions have the form

$$By = c \quad \wedge \quad y \geq 0 \ .$$

After these notations, the problem (3.1) can be reformulated as

$$\begin{aligned} \bar{y} &:= \arg \min_{y \in \Omega_E \cap \Omega_I} y^T C^T C y \ , \\ \Omega_E &:= \{y \in \mathbb{R}^{n_p+n_q} : By = c\} \ , \\ \Omega_I &:= \{y \in \mathbb{R}^{n_p+n_q} : y \geq 0\} \ . \end{aligned} \tag{3.3}$$

The next step consists of homogenization and orthogonalization. We introduce a substitution

$$x := y - y_{\text{in}} \quad \Rightarrow \quad y = x + y_{\text{in}} \ , \tag{3.4}$$

where y_{in} is arbitrary point from Ω_E. We can choose

$$y_{\text{in}} := \left[\frac{1}{n_p}, \ldots, \frac{1}{n_p}, \frac{1}{n_q}, \ldots, \frac{1}{n_q}\right] \in \mathbb{R}^{n_p+n_q} \ .$$

Afterwards, the cost function and conditions have the form

$$\begin{aligned} &f(x) := \tfrac{1}{2}\|C(x + y_{\text{in}})\|^2 = \tfrac{1}{2}x^T \overbrace{C^T C}^{=:A} x + x^T \overbrace{C^T C y_{\text{in}}}^{=:-b} + c, \quad c =: \tfrac{1}{2} y_{\text{in}}^T C^T C y_{\text{in}} = \text{ const.} \ , \\ &B(x + y_{\text{in}}) = Bx + By_{\text{in}} = Bx + c \quad \Rightarrow \quad (By = c \ \Leftrightarrow \ Bx = 0) \ , \\ &y \geq 0 \quad \Leftrightarrow \quad x \geq -y_{\text{in}} \ . \end{aligned}$$

Moreover, the matrix B can be orthonormalized using simple process

$$\hat{B} := \begin{bmatrix} \frac{1}{\sqrt{n_p}} & 0 \\ 0 & \frac{1}{\sqrt{n_q}} \end{bmatrix} B \ .$$

We obtained QP with homogeneous orthogonal linear equality constraints and bound inequality constraints

$$\begin{aligned} \bar{x} &:= \arg \min_{x \in \Omega_E \cap \Omega_I} \tfrac{1}{2} x^T A x - b^T x \ , \\ \Omega_E &:= \left\{x \in \mathbb{R}^{n_p+n_q} : \hat{B}x = 0\right\} \ , \\ \Omega_I &:= \{x \in \mathbb{R}^{n_p+n_q} : x \geq -y_{\text{in}}\} \ . \end{aligned} \tag{3.5}$$

After solving this problem, the original solution can be obtained using back substitution (3.4) to obtain y, i.e. the coeficients of linear combinations (3.2) of the nearest points from each polytope.

Numerical experiments

We consider two circles discretized by parameter $m \geq 3$, whose boundary points are defined by $P, Q \in \mathbb{R}^{2,m}$ with columns

$$P_{*,i} = \begin{bmatrix} \cos(2i\pi/m) - 2 \\ \sin(2i\pi/m) \end{bmatrix}, \quad Q_{*,i} = \begin{bmatrix} \cos(\pi - 2i\pi/m) + 2 \\ \sin(\pi - 2i\pi/m) \end{bmatrix}, \quad i = 0, \ldots, m-1 \ .$$

Examples for $m = 5$ and $m = 7$ can be found in Fig. 8.

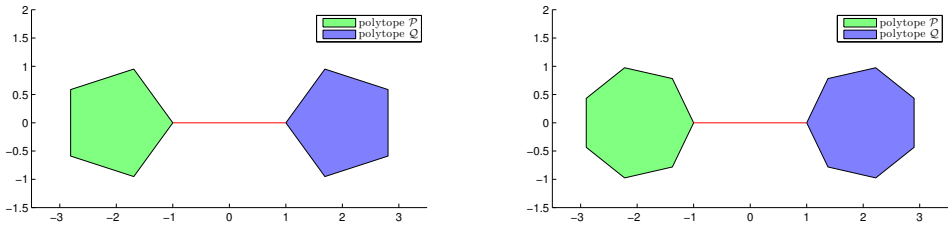


Figure 8: Testing benchmark for polytopes distance with discretization parameter $m = 5$ *(left)* and $m = 7$ *(right)*.

The solution of the problem for any m is given by

$$\bar{y} = [\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m]^T = [\underbrace{1, 0, \ldots, 0}_{\in \mathbb{R}^m}, \underbrace{1, 0, \ldots, 0}_{\in \mathbb{R}^m}]^T \ .$$

In this problem, we can directly compute the regular condition number of Hessian matrix

$$\kappa(A) = \kappa(C^T C) = \kappa(C C^T) = \kappa\left(\begin{bmatrix} \sum_{i=1}^{n_p} P_{i,1}^2 + \sum_{i=1}^{n_q} Q_{i,1}^2 & \sum_{i=1}^{n_p} P_{i,1}P_{i,2} + \sum_{i=1}^{n_q} Q_{i,1}Q_{i,2} \\ \sum_{i=1}^{n_p} P_{i,2}P_{i,1} + \sum_{i=1}^{n_q} Q_{i,2}Q_{i,1} & \sum_{i=1}^{n_p} P_{i,2}^2 + \sum_{i=1}^{n_q} Q_{i,2}^2 \end{bmatrix}\right) .$$

Moreover, it holds

$$\begin{aligned} \sum_{i=1}^{n_p} P_{i,2}P_{i,1} + \sum_{i=1}^{n_q} Q_{i,2}Q_{i,1} \ &= \ \sum_{i=0}^{m-1} \left(\cos\left(\tfrac{2i\pi}{m}\right) - 2\right) \sin\left(\tfrac{2i\pi}{m}\right) \\ &\quad + \sum_{i=0}^{m-1} \left(\cos\left(\pi - \tfrac{2i\pi}{m}\right) + 2\right) \sin\left(\pi - \tfrac{2i\pi}{m}\right) = 0 \ , \end{aligned}$$

so

$$\kappa(A) = \kappa\left(\begin{bmatrix} \sum_{i=1}^{n_p} P_{i,1}^2 + \sum_{i=1}^{n_q} Q_{i,1}^2 & 0 \\ 0 & \sum_{i=1}^{n_p} P_{i,2}^2 + \sum_{i=1}^{n_q} Q_{i,2}^2 \end{bmatrix}\right) .$$

Afterwards, the condition number can be expressed

$$\kappa(A) = \frac{\sum_{i=1}^{n_p} P_{i,1}^2 + \sum_{i=1}^{n_q} Q_{i,1}^2}{\sum_{i=1}^{n_p} P_{i,2}^2 + \sum_{i=1}^{n_q} Q_{i,2}^2} = \frac{\sum_{i=0}^{m-1} \left(\cos\left(\frac{2i\pi}{m}\right) - 2\right)^2}{\sum_{i=0}^{m-1} \sin^2\left(\frac{2i\pi}{m}\right)} \qquad \forall m \geq 3 .$$

Let us consider $\alpha \in \mathbb{R}$ and let us present a complex number $z \in \mathbb{C}$ by prescription

$$z := \cos\alpha + \mathbf{i}\sin\alpha ,$$

where $\mathbf{i}$ is imaginary unit. Then by De Moivre's formula we can write

$$\sum_{i=0}^{m-1} (\cos(i\alpha) + \mathbf{i}\sin(i\alpha)) = \sum_{i=0}^{m-1} z^i = \frac{z^m - 1}{z - 1} = \frac{\cos(m\alpha) + \mathbf{i}\sin(m\alpha) - 1}{z - 1} .$$

If we choose specific α in previous equality, we obtain next

$$\begin{aligned} \alpha := \tfrac{2\pi}{m} \quad &\Rightarrow \quad \sum_{i=0}^{m-1} \left(\cos\left(\tfrac{2i\pi}{m}\right) + \mathbf{i}\sin\left(\tfrac{2i\pi}{m}\right)\right) = 0 \quad \Rightarrow \quad \sum_{i=0}^{m-1} \cos\left(\tfrac{2i\pi}{m}\right) = 0, \\ \alpha := \tfrac{4\pi}{m} \quad &\Rightarrow \quad \sum_{i=0}^{m-1} \left(\cos\left(\tfrac{4i\pi}{m}\right) + \mathbf{i}\sin\left(\tfrac{4i\pi}{m}\right)\right) = 0 \quad \Rightarrow \quad \sum_{i=0}^{m-1} \cos\left(\tfrac{4i\pi}{m}\right) = 0. \end{aligned} \tag{3.6}$$

Now, we return back to regular condition number and $\forall m \geq 3$ we can write (using (3.6))

$$\begin{aligned} \kappa(A) = \frac{\sum_{i=0}^{m-1} \left(\cos\left(\frac{2i\pi}{m}\right)-2\right)^2}{\sum_{i=0}^{m-1} \sin^2\left(\frac{2i\pi}{m}\right)} &= \frac{\sum_{i=0}^{m-1} \cos^2\left(\frac{2i\pi}{m}\right) - 4\cos\left(\frac{2i\pi}{m}\right) + 4}{\sum_{i=0}^{m-1} \sin^2\left(\frac{2i\pi}{m}\right)} \\ = \frac{\sum_{i=0}^{m-1} \left(\frac{1+\cos\left(\frac{4i\pi}{m}\right)}{2} - 4\cos\left(\frac{2i\pi}{m}\right) + 4\right)}{\sum_{i=0}^{m-1} \frac{1-\cos\left(\frac{4i\pi}{m}\right)}{2}} &= \frac{\frac{1}{2}\sum_{i=0}^{m-1} \cos\left(\frac{4i\pi}{m}\right) - 4\sum_{i=0}^{m-1} \cos\left(\frac{2i\pi}{m}\right) + \sum_{i=0}^{m-1} \frac{9}{2}}{-\frac{1}{2}\sum_{i=0}^{m-1} \cos\left(\frac{4i\pi}{m}\right) + \sum_{i=0}^{m-1} \frac{1}{2}} = 9 . \end{aligned}$$

We implemented the algorithms in MATLAB environment and solved the problem with several values of the discretization parameter m. The dimension of optimization problem is given by $n = 2m$. We solve each problem using SMALSE-M with MPRGPS, PBBF, SPG-QP, and APGD. We demand relative precision $\varepsilon = 10^{-4} \cdot \|b\|$. As an initial approximation of inner solver, we used a previous iteration of outer solver. All other parameters of algorithms can be found in Table 10.

SMALSE-M	$\rho = \tilde{\lambda}^A_{\max}, \eta = 1, \beta = 2, M_0 = 1, x^0 = 0$
all inner solvers	$\bar{\alpha} = 1.95/\tilde{\lambda}^A_{\max}, x^0_{in} = P_\Omega(x^{k-1}_{out}), \varepsilon_{in} = 10^{-6}$
MPRGPS	$\Gamma = 1$
PBBf	$K = 10, x^1 = P_\Omega(x^0 - \bar{\alpha} g^0)$
SPG-QP	$m = 10, \gamma = 0.1, \sigma_2 = 0.9999, \alpha_0 = \bar{\alpha}$
APGD	$L = \tilde{\lambda}^A_{\max}$

Table 10: Algorithms settings.

Number of outer iterations of SMALSE-M is 6 independently on the inner solver and on dimension of the problem $2m$. We plot the sum of the numbers of inner solvers iterations in Fig. 9 and the number of all performed Hessian multiplications in Fig. 10. The performace profiles with number of iterations and number of Hessian multiplication can be found in Fig. 11. In this case $\|b\| = 120$.

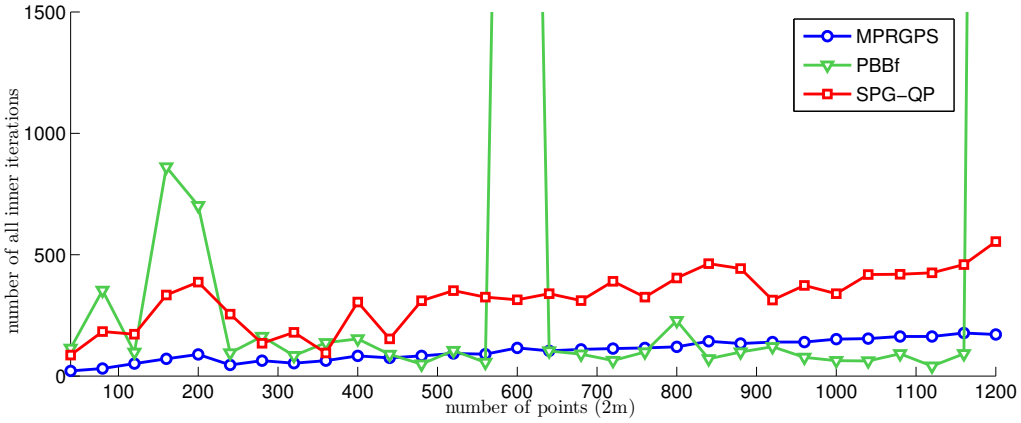


Figure 9: Polytope distance: number of inner iterations.

To study the behaviour of algorithms during the solution precess, we decided to choose $m = 100$ and depict the Euclidean norm of the stopping criteria - reduced gradient, see Fig. 12.

It is necessary to notice that the Hessian matrix of solved optimization problem is SPS. The theory of outer loop performed by SMALSE-M algorithm was presented by Dostál [26] only for SPD matrices and/or for the problems where the SPS Hessian matrix is in the inner loop regularized by the penalty, see Theorem 1.7.2. This

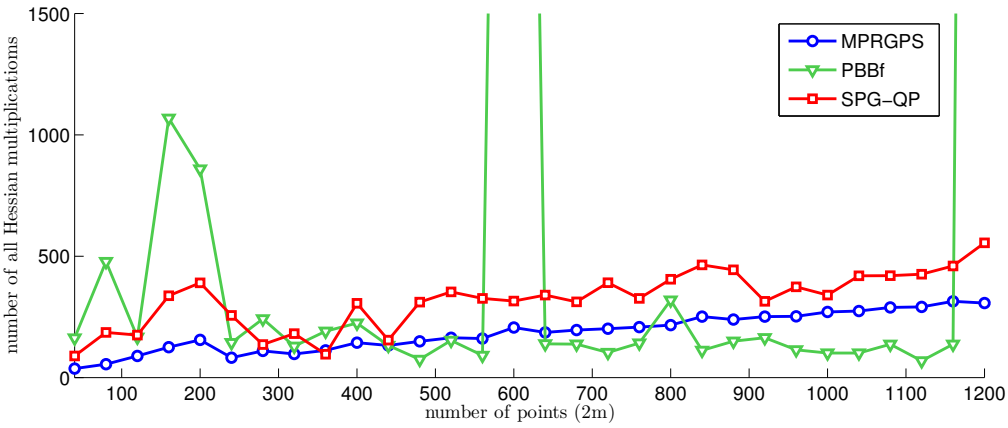


Figure 10: Polytope distance: number of Hessian multiplications.

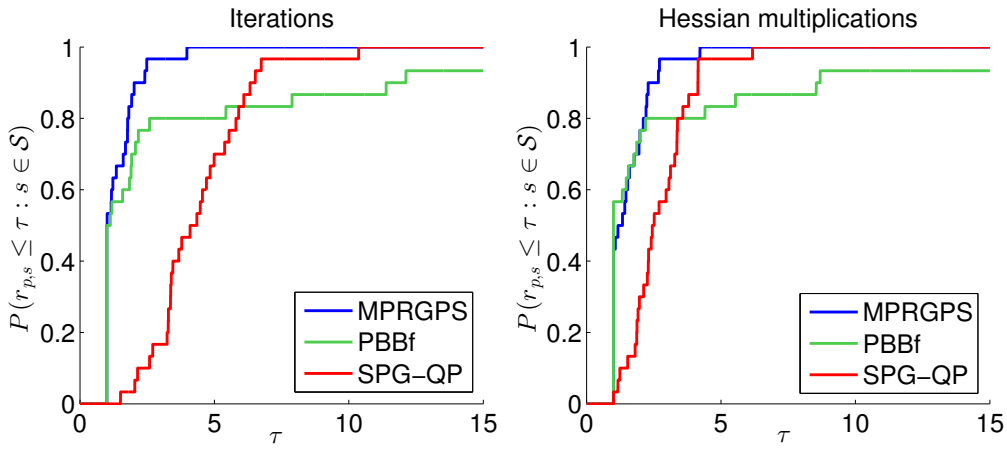


Figure 11: Polytope distance: performance profiles - number of iterations and number of Hessian multiplications.

property is based on

$$\operatorname{Ker} A \cap \operatorname{Ker} B = \{0\}.$$

If this condition is fulfilled, then the inner algorithm minimizes the problem with SPD matrix. However, the presented problem of polytope distance is not problem with such a property and the inner solver still works with only SPS Hessian matrix. Despite this fact, we are still able to solve the problem and our numerical results indicate the efficiency of MPRGPS and SPG-QP algorithms.

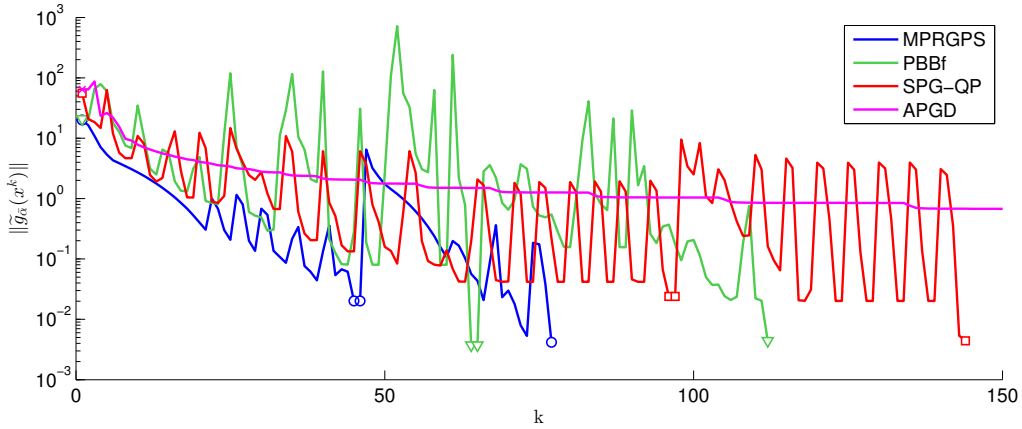


Figure 12: Polytope distance: the descent of the norm of reduced gradient.

3.1.2 Smallest enclosing ball

Let us consider the set of m points

$$\mathcal{P} = \{p_1, \ldots, p_m\} \subset \mathbb{R}^d \ .$$

The problem is to find a center of ball such that the maximum distance between this center and points in $\mathcal{P}$ is minimal, i.e.

$$\bar{p} := \arg\min_{p \in \mathbb{R}^d} \{\max_{p_i \in \mathcal{P}} \|p_i - p\|\} \ , \tag{3.7}$$

and the radius of this ball is given by the value of maximum distance between the center and points from $\mathcal{P}$, i.e.

$$r := \max_{p_i \in \mathcal{P}} \|p_i - p\| \ .$$

Similarly to Section 3.1.1, we will search for a coeficient vector of the convex linear combination

$$p = \sum_{i=1}^{m} \alpha_i p_i, \ \text{where} \ \sum_{i=1}^{m} \alpha_i = 1 \ \text{and} \ 0 \leq \alpha_i \leq 1 \ \forall i = 1, \ldots, m \ .$$

The reason is that the center of enclosing ball lies in the convex hull of $\mathcal{P}$ (see Schönherr [66]).
Afterwards, we denote

$$\begin{aligned} y &:= [\alpha_1, \ldots, \alpha_m]^T \in \mathbb{R}^m \ , \\ C &:= [p_1, \ldots, p_m] \in \mathbb{R}^{d,m} \ , \\ b &:= [p_1^T p_1, \ldots, p_m^T p_m] \in \mathbb{R}^m \ . \end{aligned}$$

Then the problem (3.7) can be reformulated as

$$\min_{p\in\mathbb{R}^d}\{\max_{p_i\in\mathcal{P}}\|p_i-p\|\} = \min_{y\in\Omega_E\cap\Omega_I}\{\max_{p_i\in\mathcal{P}}\frac{1}{2}y^TC^TCy-(Cp_i)^Ty\} = \min_{y\in\Omega_E\cap\Omega_I}\frac{1}{2}y^T\underbrace{C^TC}_{=:A}y-b^Ty\,,$$

where

$$\begin{aligned}\Omega_E &:= \{y\in\mathbb{R}^m : Bx=1\}\ ,\\ \Omega_I &:= \{y\in\mathbb{R}^m : x\geq 0\}\ ,\\ B &:= [1,\ldots,1]\in\mathbb{R}^{1,m}\ .\end{aligned}$$

Moreover, the problem can be homogenized and solved using the same methodology as in Section 3.1.1. We solve the problem to obtain the center of ball $\bar{p}$. Afterwards, the radius can be computed by

$$r=\sqrt{-\bar{p}^TA\bar{p}+2b^T\bar{p}}\ .$$

Numerical experiments

We work with random data in our benchmark. We generate $n = 100$ random points from circle

$$\{p\in\mathbb{R}^2 : \|p-[1,1]^T\|^2\leq 1\}.$$

Afterwards, we discard the information about the circle and try to find the enclosing ball using the process described above. As an initial approximation of inner solver, we used a previous iteration of outer solver. All other parameters of algorithms can be found in Table 10. We track the descend of stopping criteria during the solution of the numerical problem, see Fig. 14. In this case $\hat{\kappa} = 12.2912, \lambda_{\max} = 552.5665, \|b\| = 20.5792$. The solution can be found in Fig. 13.

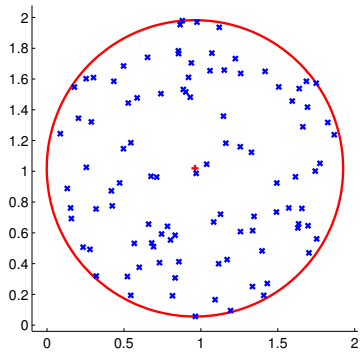


Figure 13: Enclosing ball: the solution of first benchmark.

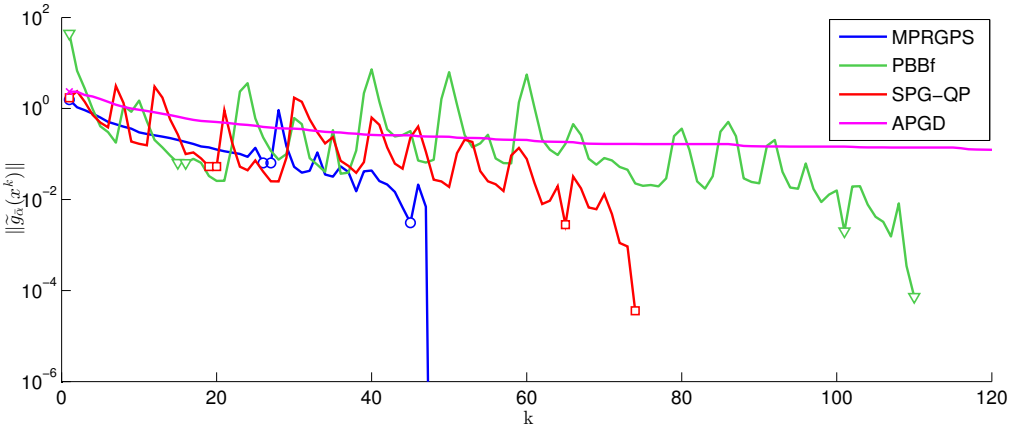


Figure 14: Enclosing ball: the descent of the norm of reduced gradient.

The second benchmark consists of the problems with various dimensions n. For each dimension n, we generate 100 sets of random points and after solving the problem for each set of n points, we take an average number of iterations for each algorithm. These average numbers can be found in Fig. 15. During the solution, we also track the number of Hessian multiplications using the same averaging process, see Fig. 16.

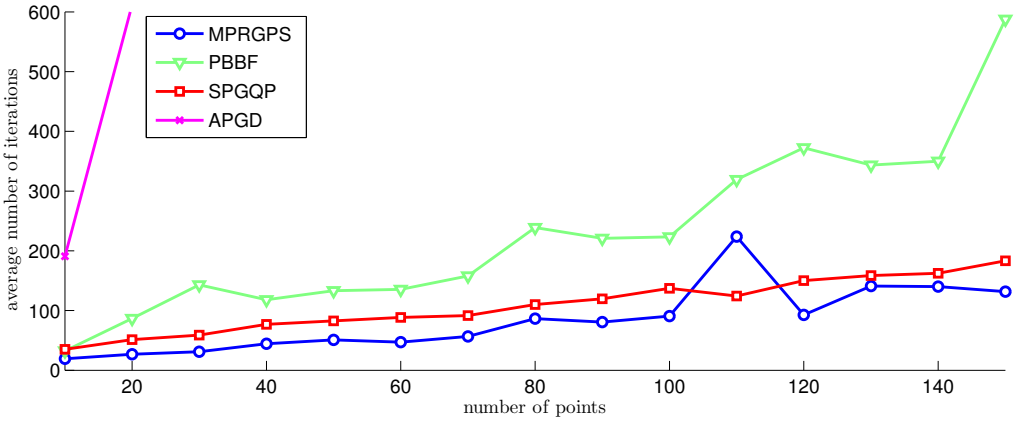


Figure 15: Enclosing ball: the average number of iterations depending on number of points.

To construct the performace profiles, we take numbers of iterations and Hessian multiplications from all problems generated in previous benchmark. The results can

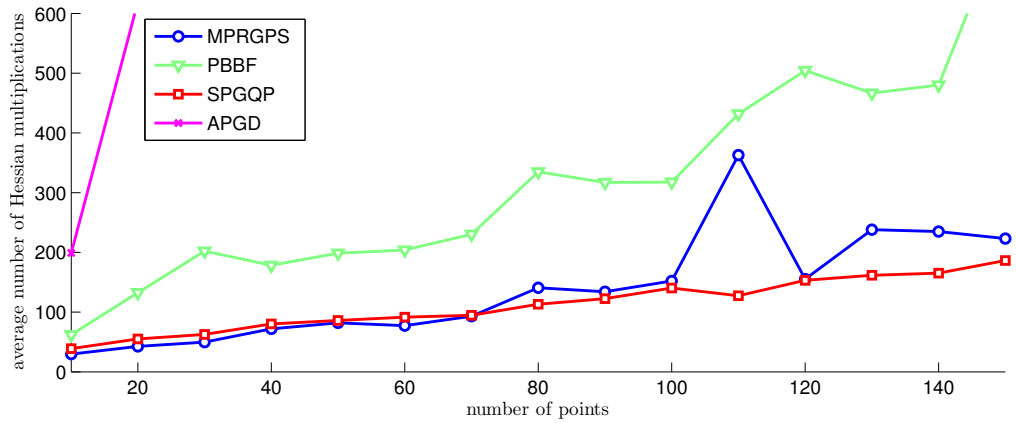


Figure 16: Enclosing ball: the average number of Hessian multiplications depending on number of points.

be found in Fig. 17.

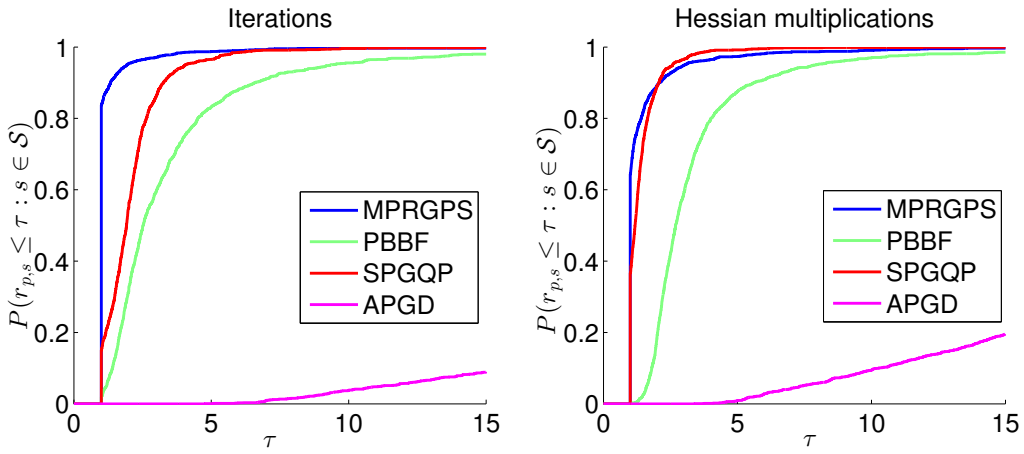


Figure 17: Enclosing ball: performance profiles - number of iterations and number of Hessian multiplications.

The problem of the smallest enclosing ball generates the objects with the similar properties as in polytope distance problems. The Hessian matrix of the system is SPS with large dimension of the kernel. This matrix is not regularized during outer loop and the inner solver has to deal with the problem with SPS matrix. Despite this fact, we are still able to solve the problem and our numerical results indicate the efficiency of MPRGPS and SPG-QP algorithms.

3.1.3 Projection onto intersection of hypercube and hyperplanes

In Dostál and Pospíšil [34], we demostrate the efficiency of our active-set algorithms by the evaluation of the projection of a point to the intersection of the unit cube and unit sphere with hyperplanes. We generate random parameters of a point and the hyperplanes for various dimensions n ranging from 5 to $5 \cdot 10^6$. Afterwards, we found the projection of the point to the intersection of the unit cube and the hyperplanes by solving the problem

$$\text{minimize} \quad q(x) = \frac{1}{2}\|x - a\|^2 \quad \text{subject to} \quad \|x\|_\infty \leq 1 \quad \text{and} \quad Bx = c \ ,$$

where $a \in \mathbb{R}^n$ is a random point, $B \in \mathbb{R}^{m,n}$ is a full row rank random matrix, $c \in \mathbb{R}^m$ is a random vector, and $\|x\|_\infty$ is the maximum norm defined for any $x \in \mathbb{R}^n$ by

$$\|x\|_\infty := \max\{|x_1|, \ldots, |x_n|\}.$$

As a stopping criterion, we use

$$\|\widetilde{g}_{\bar{\alpha}}(\bar{x})\| \leq \varepsilon \|a\| \tag{3.8}$$

with $\varepsilon = 10^{-4}$. In the thesis, we extend the results with other algorithms presented in Section 2.

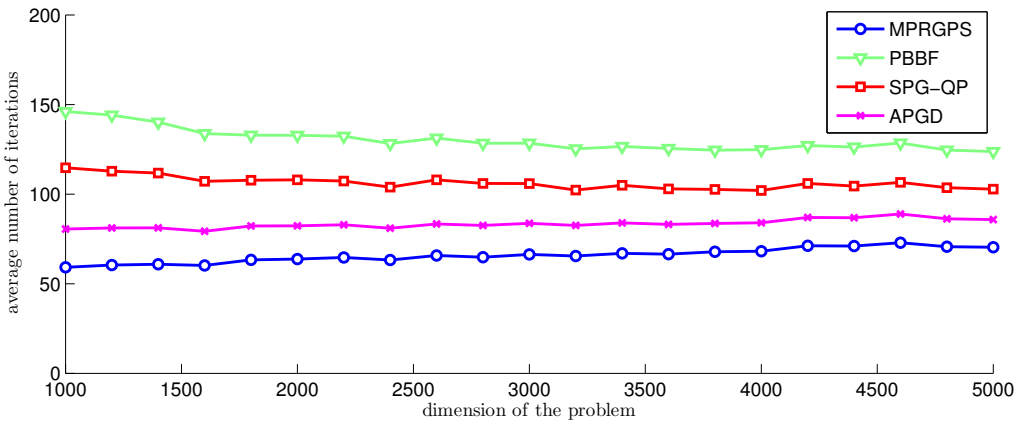


Figure 18: Projection onto intersection of hypercube and 10 hyperplanes: the average number of iterations for varying dimension of the problem.

However, the solution process requires knowledge of a feasible point, which is easy for one hyperplane or some special problems, but not in general, see Nocedal and Wright [54]. The results in Fig. 18 and Fig. 19 demonstrate the complexity of

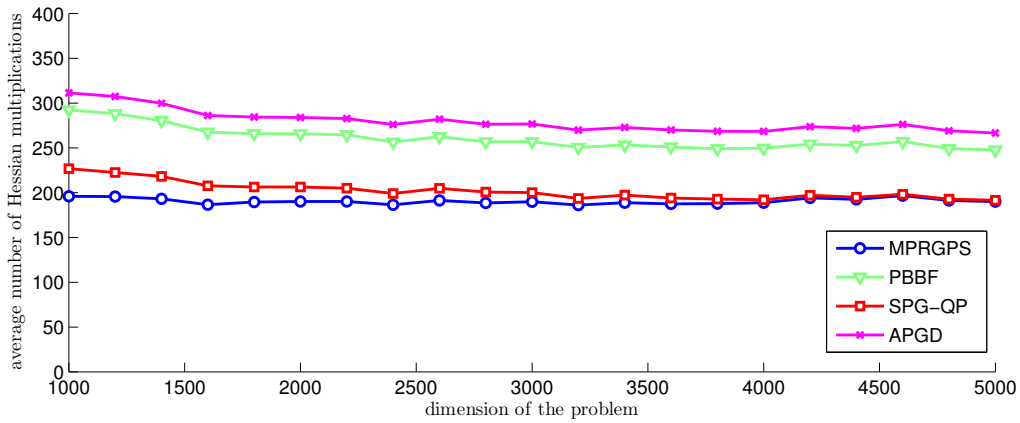


Figure 19: Projection onto intersection of hypercube and 10 hyperplanes: the average number of Hessian multiplications for varying dimension of the problem.

our algorithms for solving the problems with $m = 10$ hyperplanes. The algorithms perform almost constant number of iterations to solve the problem independently on the dimension. To generate performace profiles, we take number of iterations and Hessian multiplications from all problems generated in previous benchmark. The results can be found in Fig. 20.

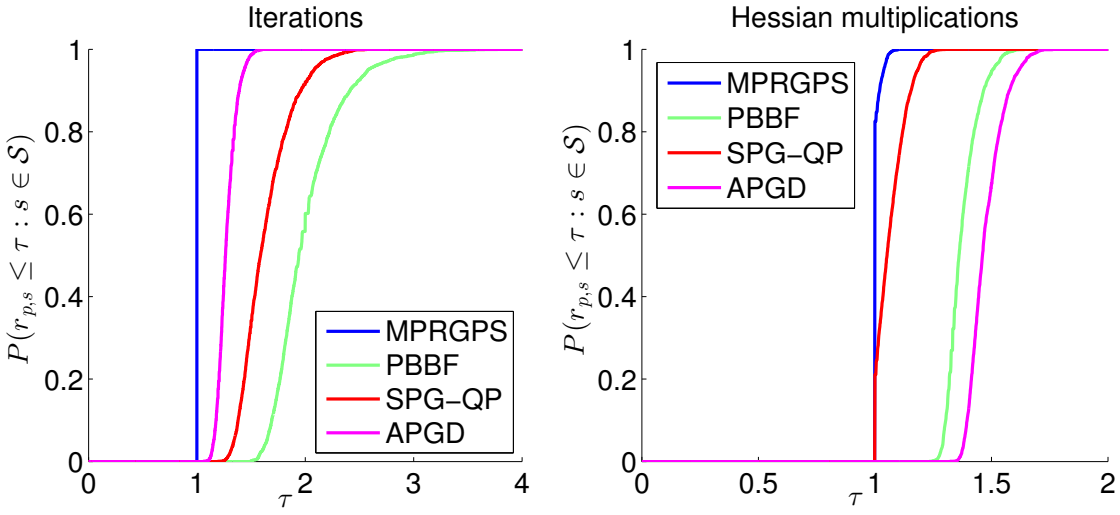


Figure 20: Projection onto intersection of hypercube and 10 hyperplanes: performance profiles - number of iterations and number of Hessian multiplications.

Using the same methodology, we show linear complexity of our algorithm for solving problem of projection of the point to the intersection of the unit sphere and the hyperplanes

$$\text{minimize} \;\; q(x) = \frac{1}{2}\|x-a\|^2 \;\; \text{subject to} \;\; \|x\| \leq r^2 \;\; \text{and} \;\; Bx = c \; . \tag{3.9}$$

Again, we generated random data of the problem and the results can be seen in Fig. 21 and Fig. 22. The profiles are in Fig. 23. In this case, we can use a point of the projection of the centre of sphere into given hyperplanes as an feasible point in our equations. See next lemma.

Lemma 3.1.1
(Feasible point in example.)
Let

$$\begin{aligned} \Omega_H &:= \{x \in \mathbb{R}^n : Bx = c\}, \\ \Omega_S &:= \{x \in \mathbb{R}^n : x^T x \leq r^2\}, \end{aligned}$$

be sets from problem (3.9).
Then

$$x_{\text{in}} := \arg\min_{x \in \Omega_H} \|x - 0\|_2 = B^T(BB^T)^{-1}c \tag{3.10}$$

satisfies

$$x_{\text{in}} \in \Omega_H \cap \Omega_S \; .$$

Proof: Notice that

$$Bx_{\text{in}} = BB^T(BB^T)^{-1}c = c \; ,$$

which implies $x_{\text{in}} \in \Omega_H$.
We assume, that feasible set is non-empty, so we can choose x_{in2} such that $x_{\text{in2}}^T x_{\text{in2}} \leq r^2$ and $Bx_{\text{in2}} = c$. Furthermore, x_{in} solves optimization problem (3.10), so we can write

$$x_{\text{in}}^T x_{\text{in}} = \|x_{\text{in}} - 0\|^2 \leq \|x_{\text{in2}} - 0\|_2^2 = x_{\text{in2}}^T x_{\text{in2}} \leq r^2 \; .$$

□

In algorithms, we use projection onto sphere

$$\Omega_S := \{x \in \mathbb{R}^n : (\tilde{x} + x_{\text{in}})^T(\tilde{x} + x_{\text{in}}) \leq r^2\}$$

defined by

$$P_{\Omega_S}(x) := \arg\min_{y \in \Omega_S} \|x - y\| \; .$$

In this case, we compute each component of projection onto Ω_S by simple formula

$$P_{\Omega_S}(x) := \begin{cases} x, & \text{if } x \in \Omega_S \ , \\ \frac{r}{\|x+x_{\text{in}}\|}(x + x_{\text{in}}) - x_{\text{in}}, & \text{if } x \notin \Omega_S \ . \end{cases}$$

Furthermore, we need to compute α_f as the largest step-size in the proportional step in MPGP, which does not cause leaving the set Ω_S

$$\alpha_f := \max\{\alpha \in \mathbb{R}^+ : x + \alpha p \in \Omega\}.$$

In this case, this coefficient can be computed using prescription (positive root of quadratic equation which occurs in solving problem of intersection line and sphere)

$$\alpha_f := \frac{-p^T y + \sqrt{(p^T y)^2 - p^T p(y^T y - r^2)}}{p^T p}, \quad y := x + x_{\text{in}}.$$

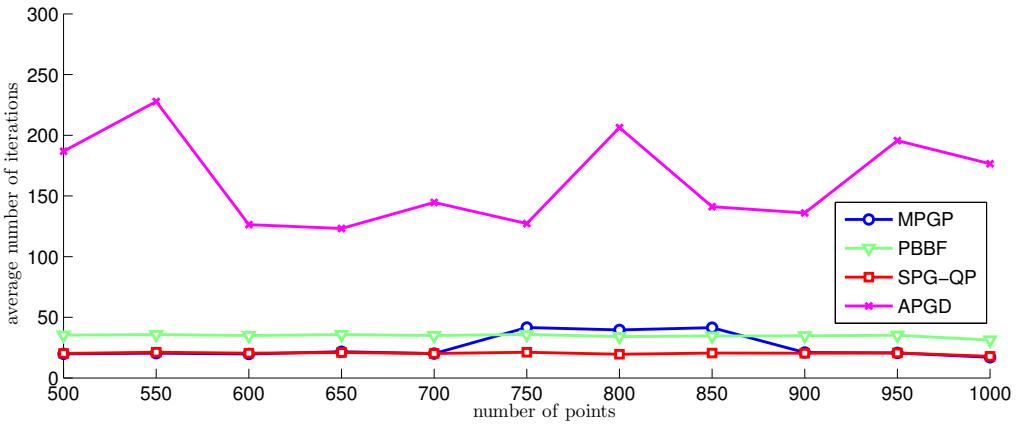


Figure 21: Projection onto intersection of sphere and 10 hyperplanes: the average number of iterations for varying dimension of the problem.

During our numerical experiments in problems with the intersection of hyperplane and sphere, we observe that the number of iterations of the algorithm decreases while we increase the problem dimension. The reason is that if we use the relative precision control based on the norm of right-hand side vector, then the problem is easier to solve. See next lemma. We prove, that if we choose sufficiently large ε, then the problem is solved in one iteration independently of the choice of initial approximation.

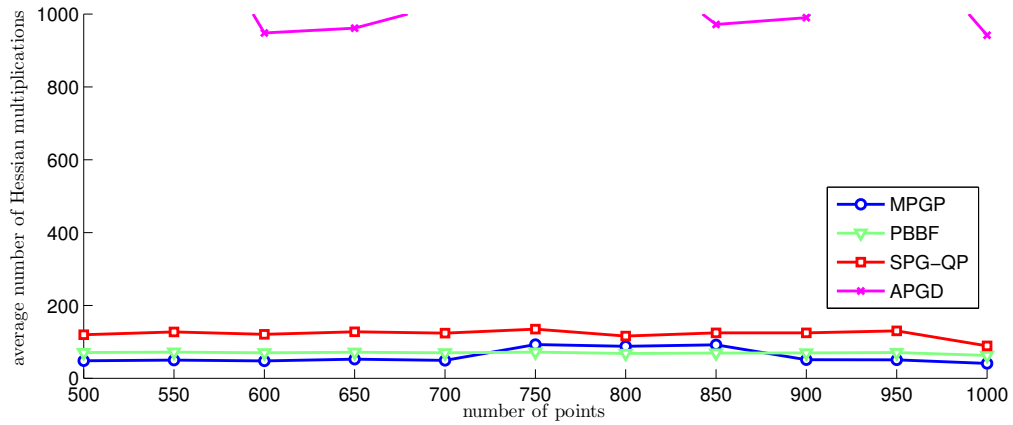


Figure 22: Projection onto intersection of sphere and 10 hyperplanes: the average number of Hessian multiplications for varying dimension of the problem.

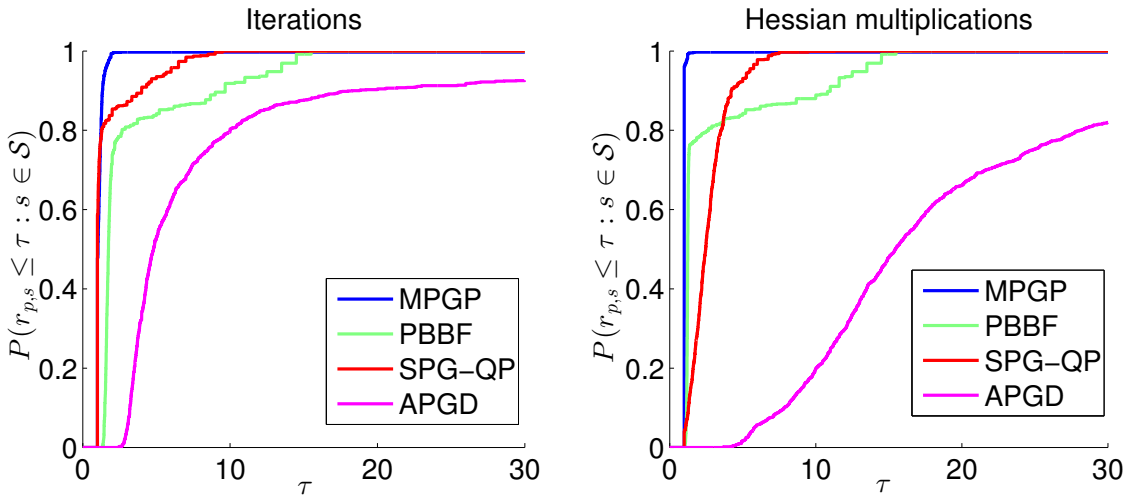


Figure 23: Projection onto intersection of sphere and 10 hyperplanes: performance profiles - number of iterations and number of Hessian multiplications.

Lemma 3.1.2
(Precision in problem with hyperplane and sphere.)
Let

$$\frac{2r\sqrt{\|B^TB\|}}{\|b\|} \leq \varepsilon \tag{3.11}$$

then

$$\|Bx^0 - c\| < \varepsilon\|b\|$$

for any $x^0 \in \Omega_S$.

Proof: We suppose that $\Omega_H \cap \Omega_S \neq \emptyset$, therefore we can choose $\bar{x} \in \Omega_H \cap \Omega_S$ (for instance the solution of the given problem (3.9)). Obviously

$$\begin{aligned} x^0, \bar{x} \in \Omega_S \quad &\Rightarrow \quad \|x^0 - \bar{x}\| \leq 2r, \\ \bar{x} \in \Omega_H \quad &\Rightarrow \quad B\bar{x} = c. \end{aligned}$$

Using this and the proposition of the lemma, we can estimate

$$\begin{aligned} \|Bx^0 - c\|^2 &= \|Bx^0 - B\bar{x}\|^2 = \langle B(x^0 - \bar{x}), B(x^0 - \bar{x})\rangle \\ &= (x^0 - \bar{x})^T B^T B(x^0 - \bar{x}) = \langle B^TB(x^0 - \bar{x}), x^0 - \bar{x}\rangle \\ &\leq \|B^TB\|.\|x^0 - \bar{x}\|^2 \leq \|B^TB\|.4r^2 \\ &\leq \|B^TB\|.4\frac{\varepsilon^2\|b\|^2}{4\|B^TB\|} = \varepsilon^2\|b\|^2. \end{aligned}$$

□

The numerical results show the efficiency of the algorithms. We show that the projection problems can be solved by optimal QP algorithms very effectively. In this case, the Hessian matrix of both outer and inner optimization problems is SPD and the number of iterations is independent on the dimesion of the problem. These results are mostly based on the property of outer SMALSE-M algorithm.

3.2 Contact problems of mechanics

3.2.1 Membrane

In this section, we demonstrate the behaviour of the algorithms on the numerical solution of simple two dimensional membrane deflection with contact. We consider a square-shaped ideal membrane on $\Omega_P := \langle 0,1\rangle \times \langle 0,1\rangle$ from homogeneous material with fixed boundary $\Gamma_N := \partial\Omega_P$ by homogeneous Dirichlet boundary condition, see Fig. 24. We are interested in the deflection of the membrane desribed by function

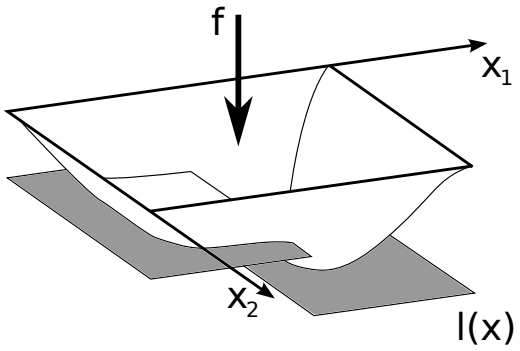


Figure 24: The contact problem of membrane and obstacle.

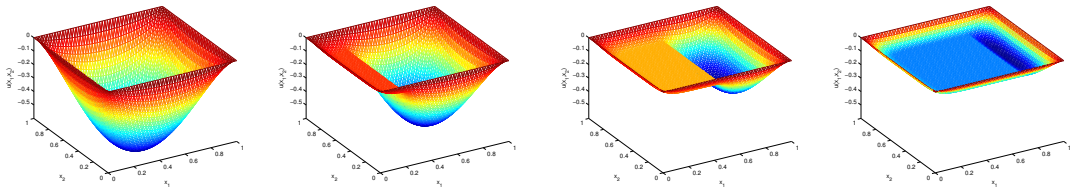
$u : \Omega_P \to \mathbb{R}$, which is caused by an acting force of the density $f(x_1, x_2) := -1$. Furthermore, we also consider a rigid obstacle defined by function

$$l(x_1, x_2) := \begin{cases} -1/10 & \text{for } (x_1, x_2) \in \langle 0, p\rangle \times \langle 0, 1\rangle, \\ -1 & \text{for } (x_1, x_2) \in (p, 1\rangle \times \langle 0, 1\rangle, \end{cases}$$

where $p \in \langle 0, 1\rangle$ is a parameter of the problem. We consider non-penetration condition of the membrane and above-mentioned obstacle.
The problem in continuous form is given by

$$\begin{aligned} -\Delta u(x) &= f(x) && \text{for } x \in \Omega_P, \\ u(x) &= 0 && \text{for } x \in \Gamma_N, \\ u(x) &\geq l(x) && \text{for } x \in \Omega_P. \end{aligned}$$

For the construction of discrete problem, we use popular Finite Element Method (FEM, see Brenner and Scott [18] or Alberty et al. [5]) on regular square grid. We denote the number of the division of a squere side by $h \in \mathbb{N}$. We obtain an



(a) $p = 0$ (b) $p = 0.25$ (c) $p = 0.5$ (d) $p = 0.75$

Figure 25: Solution of contact benchmark with the number of variables $n = 2500$ and several values of constraint parameter.

optimization problem

$$\begin{aligned} \bar{u} &:= \arg\min_{u \in \Omega} \tfrac{1}{2} u^T K u - f^T u, \\ \Omega &:= \{u \in \mathbb{R}^n : u \geq l\}, \end{aligned} \tag{3.12}$$

where $n \in \mathbb{N}$ denotes the number of FEM nodes and without any confusion, we denoted $u, l, f \in \mathbb{R}^n$ the discretized forms of functions $u, l, f : \Omega \to \mathbb{R}$. The stiffness matrix $K \in \mathbb{R}^{n,n}$ is SPD because of given Dirichlet boundary condition. The problem (3.12) is QP with bound constraints.

The solution of the problem (3.12) for several parameters p can be found in Fig. 25. The parameter p influences the number of active constraints in the solution of the problem Fig. 26 , e.g. the size of set $\mathcal{A}(\bar{x})$ given by (1.30), and also the number of iterations of the algorithms, see Fig. 28. In this case, we choose discretization parameter $h = 50$ and we obtain a problem with $n = 2500$ variables. Performance profiles can be found in Fig. 29.
In the second benchmark, we fix $p = 0.5$ and change the discretization parameter, i.e. the dimension of the problem. The number of iterations can be found in Fig. 30 and the performance profiles in Fig. 31.

These results show the basic advantage of active-set methods. If the algorithm combines the efficient solver for unconstrained problem on free set and an appropriate gradient projection method, it is able to solve both the unconstrained and constrained problems with a small number of constraints very efficiently.

In the last benchmark, we fix $p = 0.5$ and choose the dimension $n = 2500$. In this case $\lambda^A_{\max} = 7.9871, \kappa(A) = 615.8811, \|b\| = 0.2498$. The descent of the reduced gradient can be found in Fig. 32.

The algorithm MPRGP combines the most effective algorithm for solving unconstrained QP problems, e.g. CG method, and projected SD method for solving the problem on active set. Moreover, the number of iterations of MPRGP is theoretically independent of the problem dimension, and depends only on the largest eigenvalue of stiffness matrix. In FEM, this number is bounded independently of

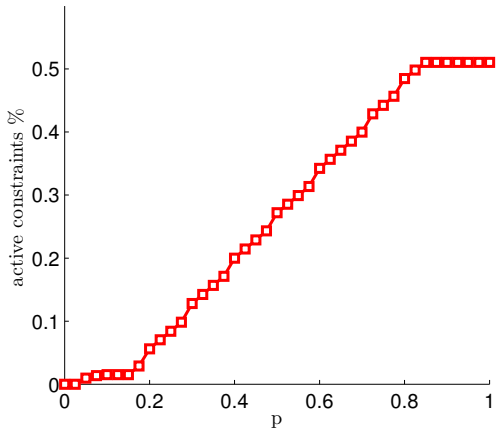


Figure 26: Percentage of active constraints depending on obstacle parameter p.

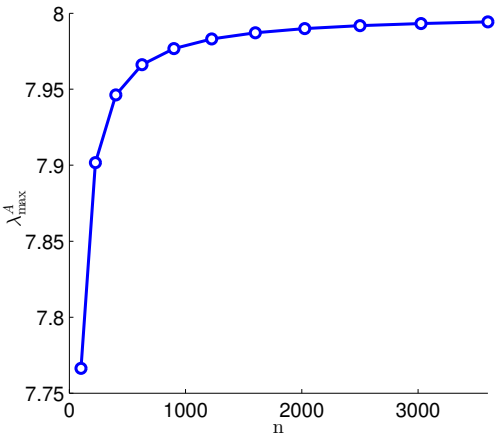


Figure 27: Largest eigenvalue for increasing problem size n.

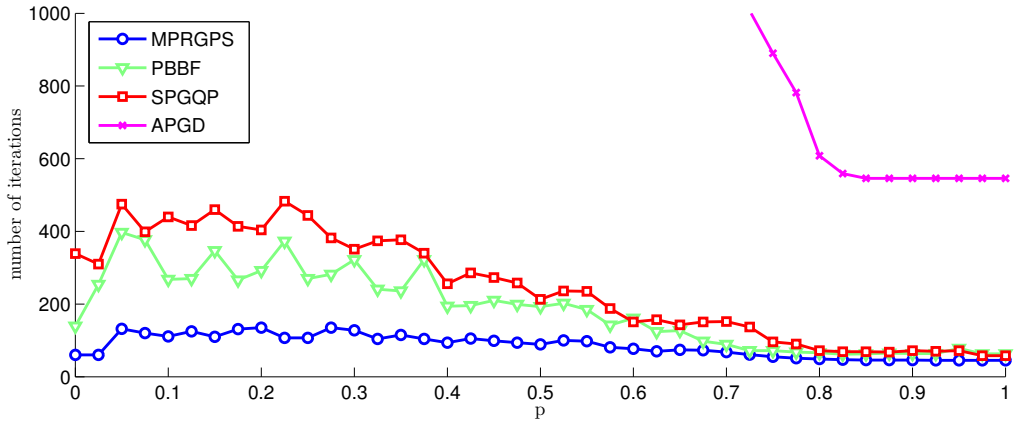


Figure 28: Number of iterations for the problem size $n = 2500$ and several choices of constraint parameter p.

the discretization. Practically, this effect can be seen on the results presented in Fig. 30, where the number of iterations for varying dimension of the problem can be found. This property of MPRGP is the result of the adaptive proportioning, e.g., the switching between solving the problem on free and active set.

Applications to contact problems of elasticity, including the transient contact problems, problems with friction, and contact shape optimization problems dis-

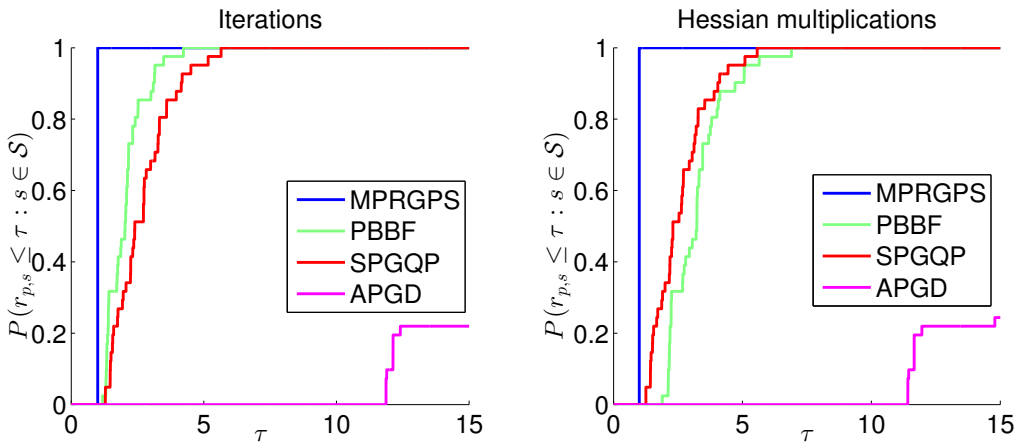


Figure 29: Performance profiles for the membrane problem with $n = 2500$ and several values of obstacle parameter p.

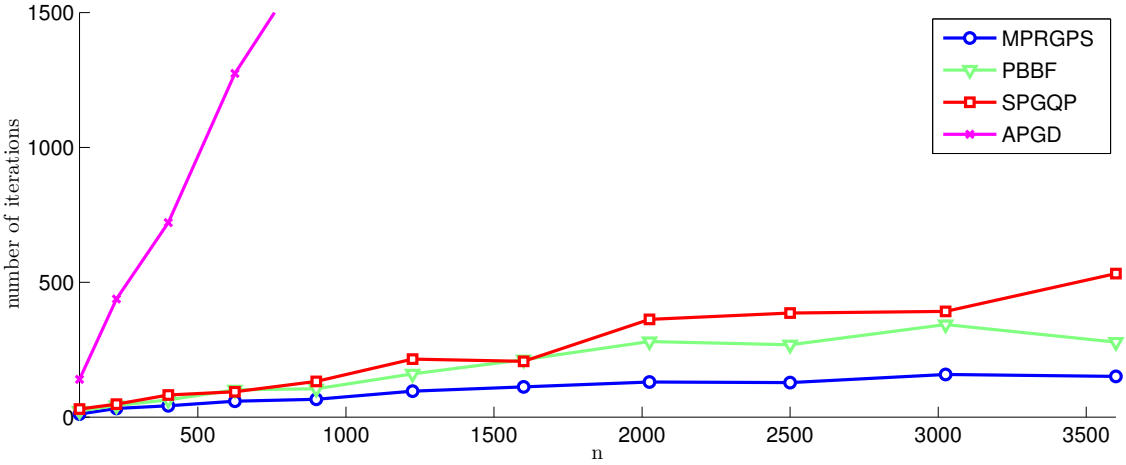


Figure 30: Number of iterations of algorithms for $p = 0.5$ and increasing problem size.

cretized up to more than $4 \cdot 10^7$ nodal variables can be found in Dostál et al. [32, 33, 31, 68, 63].

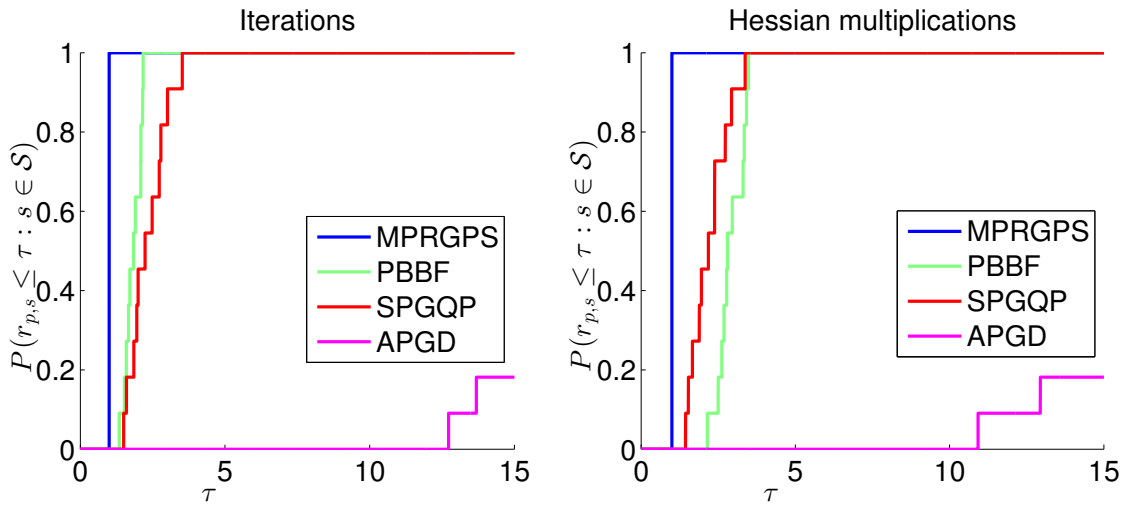


Figure 31: Performance profiles for the membrane problem with $p = 0.5$ and several values of problem dimension n.

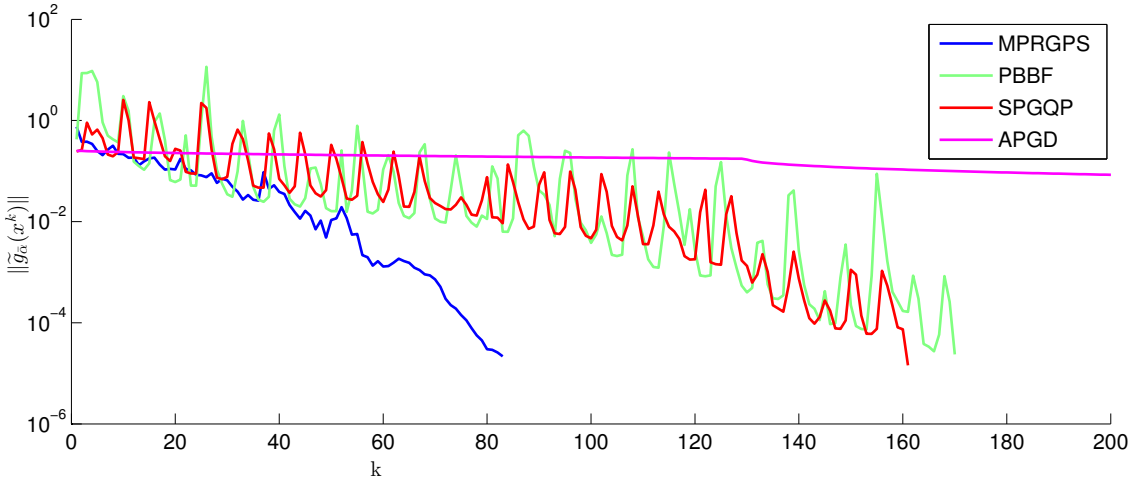


Figure 32: Membrane: the descent of the norm of reduced gradient on the problem with $p = 0.5, n = 2500$.

3.2.2 Semicoercive problem

In Dostál and Pospíšil [35], we test a variant of MPRGP algorithm on the solution of a 2D semi-coercive contact problem of elasticity with a floating body pressed against the rigid obstacle. This problem has SPS Hessian matrix A and $\operatorname{Ker} A$ with small dimension. Matrix A has uniformly distributed spectrum, see Fig. 35. The goal is to find the displacement of a homogeneous cylinder which is pressed into the rigid corner by the gravity force as in Fig. 33. The cylinder is made of the material with the density $\rho = 7.82\text{g/cm}^3$, Young's modulus $E = 10$ MPa, and Poisson's ratio $\nu = 0.3$. The problem was discretized by the linear elements using the MESH2D [37] library. The resulting mesh with the von Mises stress distribution is depicted in Fig. 34. To describe the non-penetration conditions by the bound constraints, the problem was rotated by $-\pi/4$.

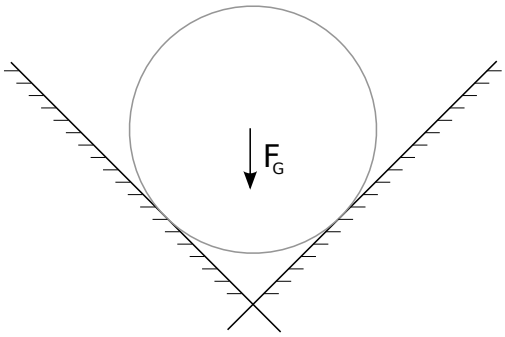


Figure 33: Pressed elastic cylinder into the rigid corner by the gravity force.

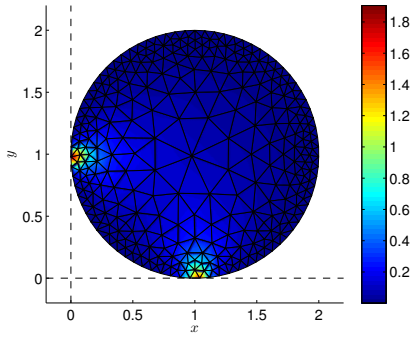


Figure 34: The resulting mesh with the von Mises stress distribution.

After the discretization, we obtain the vector of nodal forces b of the dimension $n = 660$ and the norm $\|b\| = 0.066588$. The stiffness matrix A has three zero eigenvalues, $\hat{\lambda}^A_{\min} = 0.16576$, $\lambda^A_{\max} = 72$, and the regular condition number $\hat{\kappa}(A) = 434.31$. The distribution of the spectrum of A is depicted in Fig. 35.

The displacement u minimizes the energy function $f(u) = \frac{1}{2}u^T Au - b^T u$ subject to the non-penetration constraints.

We resolved also the class of problems obtained by the parametrization of our benchmarks with $E \in \{1, 10, 100, 10^4, 10^6, 10^8\}$. The choice of this parameter influences the larges and the smallest eigenvalue of the Hessian matrix. However, the condition number remains the same. We solve the problems with the relative precision

$$\|g^P(x^k)\| \le \varepsilon\|b\|, \quad \varepsilon = 10^{-4} \ .$$

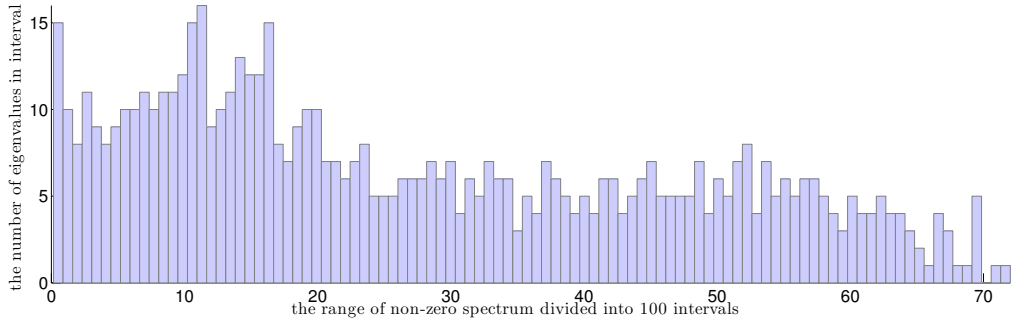


Figure 35: Cylinder: Histogram of the Hessian spectrum of problem with $E = 10$.

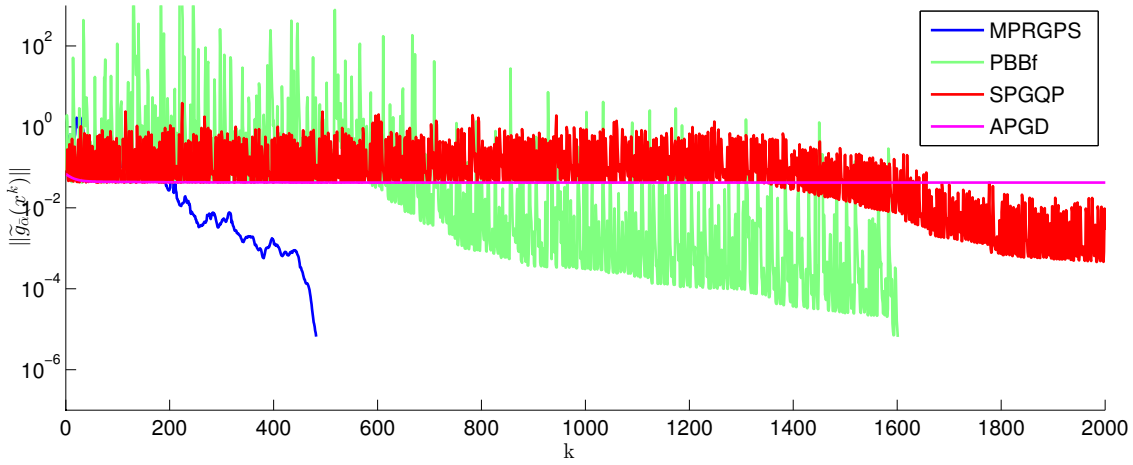


Figure 36: The decrease of $\|g^P\|$, $(E = 10)$.

The results are in Table 11. We conclude that MPRGPS minimizes the cost function the most efficiently and performs the best in the later stage of computations, see Fig. 36.

	$E = 1$	$E = 10$	$E = 10^2$	$E = 10^4$	$E = 10^6$
$\hat{\lambda}^A_{\min}$	7.2	72	720	$720 \cdot 10^2$	$720 \cdot 10^4$
$\lambda^A_{\max}$	0.016	0.166	1.66	166	$166 \cdot 10^2$
$\hat{\kappa}(A)$	434.41	434.41	434.41	434.41	434.41
active	33	13	6	3	3
MPRGPS	251 / 284	482 / 498	424 / 435	806 / 813	856 / 862
PBBf	381 / 637	1601 / 2610	4831 / 7897	$*$ /$*$	$*$ /$*$
SPG-QP	955 / 956	3569 / 3570	$*$ /$*$	$*$ /$*$	$*$ /$*$
APGD	$*$ /$*$	$*$ /$*$	$*$ /$*$	$*$ /$*$	$*$ /$*$

Table 11: $*$ – stopped after 10000 iterations. The numbers of iterations/matrix–vector products to solve the floating body benchmark with varying E and relative precision $\varepsilon = 10^{-4}$. Number of active constraints in the solutions also provided.

3.2.3 Problem with friction

In Pospíšil [58] and Bouchala et al. [15], we presented variants of MPGP for solving QP with separable spherical and elliptic constraints. Such problems arrise in linear elasticity contact problems with isotropic and anisotropic friction, respectively.
In this section, we shortly review how the friction problem with non-quadratic term can be transformed into QP with separable quadratic constraints. Afterwards, we present results of numerical experiments performed on Anselm supercomputer.
Let us consider simple contact problem with given friction. The block of homogeneous material has prescribed zero displacements on boundary Γ_D and imposed traction F on Γ_F. The part Γ_C denotes the part of boundary that may get into contact with rigid obstacle. The block is attracted to obstacle by incidence of gravity force F_G, see Fig. 37.

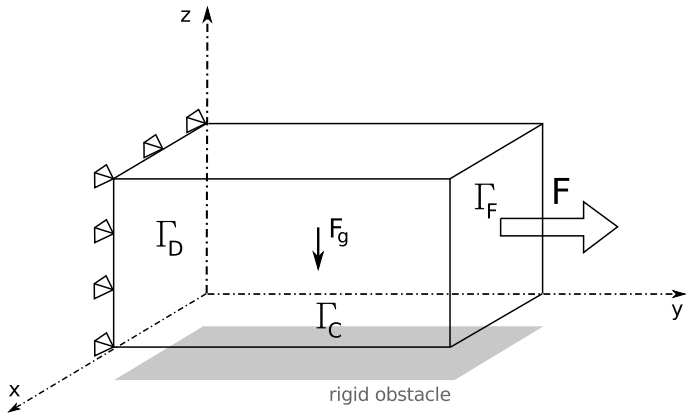


Figure 37: Contact problem with rigid obstacle and given friction.

We solve discretized form of the problem using Finite Element Method (FEM, see e.g. Brenner and Scott [18] or Alberty et al. [5]). This technique evolves optimization problem

$$\bar{u} := \min_{u \in \Omega_C} f(u) + j_h(u), \quad f(u) := \frac{1}{2} u^T K u - f^T u, \quad j_h(u) := \sum_{i=1}^{m_c} \psi_i \|T_i u\|, \quad (3.13)$$

where $N \in \mathbb{N}$ is number of used nodes and $n = 3N$ is number of variables, $u \in \mathbb{R}^n$ is a vector of unknown displacements, $\Omega_C := \{u \in \Gamma_C : u_z \geq -d_C\}$ is set of feasible u, $d_C \in \mathbb{R}$ is a distance between body and rigid obstacle, $f : \mathbb{R}^n \to \mathbb{R}$ denotes function of total potential energy, $K \in \mathbb{R}^{n,n}$ is a symmetric-positive definite stiffness matrix, $f \in \mathbb{R}^n$ is vector of internal forces resulting from the stresses imposed on the structure during a displacement, $j_h : \mathbb{R}^n \to \mathbb{R}$ is numerical integration of functional describing the friction forces in the weak formulation of the problem, $T_i \in \mathbb{R}^{m_c,2}$ are

formed by appropriately placed multiples of the unit tangential vectors in such way that the jump of tangential displacement due to displacement u is given by $T_i u$, and $\psi_i \in \mathbb{R}$ is slip bound associated with T_i.

At first, we denote $m_c \leq N$ as number of FEM nodes in Γ_C.
Since our problem has simple geometry, see Fig. 38, we can simply choose $n := [0, 0, -1]$ as normal vector and $t_1 := [1, 0, 0], t_2 := [0, 1, 0]$ as tangential vectors for every FEM node from Γ_C.

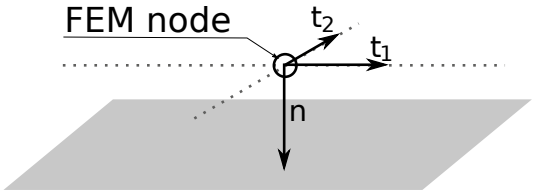


Figure 38: Normal and tangential vectors on Γ_C.

For every contact node (i-th node from Γ_C) is $T_i \in \mathbb{R}^{2,n}$ given by zero matrix with 1 in first row on appropriate x-coordinate of i-th node and in second row on appropriate y-coordinate of i-th node. Then we assume that $T := \left[T_1^T, \ldots, T_{m_c}^T\right]^T$ is the full rank matrix.

We can modify the non-differentiable term j_h in (3.13) into (see Hlaváček et al. [49])

$$j_h(u) = \sum_{i=1}^{m_c} \max_{\|\tau_i\| \leq \psi_i} \tau_i^T T_i u, \tag{3.14}$$

where $\tau_i \in \mathbb{R}^2$ are regulation variables. Moreover, we denote function and vector

$$L(u, \lambda) := f(u) + \tau^T T u, \quad \tau := [\tau_1^T, \ldots, \tau_{m_c}^T]^T, \tag{3.15}$$

then conditions $\|\tau_i\| \leq \psi_i$ can by written in the form

$$\sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, \quad i = 1, \ldots, m_c, \tag{3.16}$$

where τ_j is j-th component of τ.
We denote set of feasible τ as

$$\Lambda_\tau := \left\{ \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \ldots, m_c \right\}. \tag{3.17}$$

After substitution (3.14) into (3.13) and using (3.15),(3.16) we get

$$\bar{u} := \min_{u \in \Omega_C} (f(u) + j_h(u)) = \min_{u \in \Omega_C} \left(f(u) + \sum_{i=1}^{m_c} \max_{\|\tau_i\| \leq \psi_i} \tau_i^T T_i u \right) = \min_{u \in \Omega_C} \sup_{\tau \in \Lambda_\tau} L(u, \tau). \tag{3.18}$$

If we consider $L(u,\tau)$ as Lagrange function and τ as vector of Lagrange multipliers (in notation (3.15)), we can use the duality theorem (see Dostál [26]) to reformulate problem (3.18) and get

$$\min_{u\in\Omega_C}\sup_{\tau\in\Lambda_\tau} L(u,\tau) = \max_{\tau\in\Lambda_\tau}\min_{u\in\Omega_C} L(u,\tau). \tag{3.19}$$

We include condition $u \in \Omega_C$ by creating new Lagrange multipliers

$$\max_{\tau\in\Lambda_\tau}\min_{u\in\Omega_C} L(u,\tau) = \max_{\tau\in\Lambda_\tau,\lambda_C\geq 0}\min_{u\in\mathbb{R}^n} L(u,\tau) + \lambda_C^T(Bu-c), \tag{3.20}$$

where matrix $B \in \mathbb{R}^{m_c,n}$ and vector $c \in \mathbb{R}^{m_c}$ are constructed in such way, that

$$\{u \in \mathbb{R}^n : Bu \leq c\} = \Omega_C.$$

Let us suppose that in our problem with Dirichlet boundary conditions is f strictly convex quadratic function. In next, we can use standard inversion K^{-1}.
Due to geometry in our problem we can construct B very simply. B is zero matrix with -1 in every i-th row (which is corresponding to i-th node in Γ_C) on appropriate z-coordinate of i-th node (see choice of normal vectors for nodes in Γ_C).
Problem (3.13) is equivalent to the saddle point problem

$$(\bar{u},\bar{\lambda}) := \arg\max_{\lambda\in\Lambda}\min_{u\in\mathbb{R}^n} f(u) + \lambda^T(\tilde{B}u-\tilde{c}), \tag{3.21}$$

where

$$\lambda := \begin{bmatrix} \tau \\ \lambda_C \end{bmatrix}, \quad \tilde{B} := \begin{bmatrix} T \\ B \end{bmatrix}, \quad \tilde{c} := \begin{bmatrix} o \\ c \end{bmatrix}$$

and

$$\Lambda := \{[\tau,\lambda_C] \in \mathbb{R}^{3m_c} : \sqrt{\tau_{2i-1}^2+\tau_{2i}^2} \leq \psi_i, i = 1,\ldots,m_c, \lambda_C \geq o\}.$$

We solve problem (3.21) using dual formulation, dual function and KKT conditions, see Lemma 1.3.7.
At first, we introduce the first KKT condition

$$Ku - f + \tilde{B}^T\lambda = o \quad \Rightarrow \quad u = K^{-1}\left(f - \tilde{B}^T\lambda\right) \tag{3.22}$$

and substitute this into Lagrange function (3.15) and make some simplifications. We get

$$L(u,\lambda) = L\left(K^{-1}(f-\tilde{B}^T\lambda),\lambda\right) = -\frac{1}{2}\lambda^T\tilde{B}K^{-1}T^T\lambda + \lambda^T\tilde{B}K^{-1}f - \frac{1}{2}f^TK^{-1}f.$$

We obtain function of only one variable λ. Since we want to find maximizer (see saddle-point problem (3.21)), we omit the constant term and change signs. Then $\bar{\lambda}$ solves minimization problem

$$\bar{\lambda} = \min_{\lambda \in \Lambda} \Theta(\lambda), \quad \Theta(\lambda) := \frac{1}{2}\lambda^T F \lambda - \lambda^T d, \tag{3.23}$$

where we denoted

$$F := \tilde{B} K^{-1} \tilde{B}^T, \quad d := \tilde{B} K^{-1} f.$$

After solving minimization problem (3.23), the corresponding solution $\bar{u}$ of primal problem (3.13) can be evaluated using (3.22).
Obviously $F \in \mathbb{R}^{3m_c, 3m_c}$ is SPD and problem (3.23) is QP problem with separable quadratic constraints combinated with bound constraints.

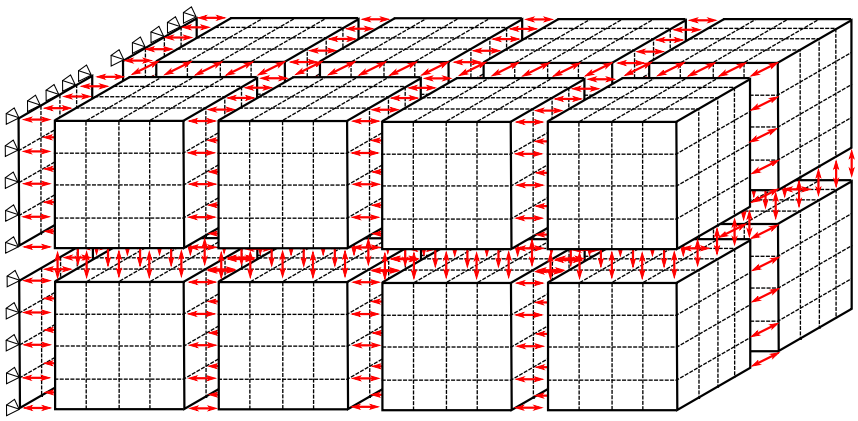


Figure 39: TFETI domain decomposition method with gluing conditions.

For the parallelization of the problem, we combine previous approach with TFETI (Total Finite Element Tearing and Interconnection method, see Dostál et al. [33]), which is a variant of the classical FETI (Finite Element Tearing and Interconnection method, see Farhat and Roux [38]). In this non-overlaping domain decomposition, we split the discretized problem into smaller parts, i.e. *subdomains*. The continuity of the global solution throughout subdomains is enforced by the *gluing* conditions. These conditions are represented by additional equality constraints, which express the equality of the solution in corresponding nodes in neighbouring subdomains. In dual formulation, these additional equality constraints entrain new set of Lagrange multipliers, which are not constrained. The TFETI method differes from the original FETI method in the way which is used to implement the Dirichlet boundary conditions. TFETI uses the additional equality constraints to *glue* the subdomains to the boundary whenever the Dirichlet boundary conditions are prescribed, see Fig.

39.
Moreover, we use additional ideas of optimal solution of the problem, e.g. homogenization, preconditioning by the projector, etc. The details can be found in Dostál [26].
In our elementary numerical benchmark, we use regular FEM grid for both of the discretization and the domain partitioning. The number of the domains in each direction is denoted by $[N_x, N_y, N_z]$ and number of elements in each domain is denoted by $[n_x, n_y, n_z]$, respectively.

In our numerical experiment we consider steel brick ($E = 2.10^5, \mu = 0.33, \rho = 7.85.10^{-2}$) of size $2\times1\times1$[m] in mutual contact with rigid obstacle. The displacement and the friction are caused by the force $F = [300, 600, 0]$. We consider given (Tresca) orthotropic friction between the brick and the obstacle. Let us recall that the Tresca friction is a simple friction law which violates some natural physical principles, but it can be used to define a mapping whose fixed point is a solution to the problem with the Coulomb friction, see Panagiotopoulos [55]. To discretize the problem, we choose $N_X = N_Z = 2, N_Y = 4$ and $n_x = n_y = n_z = 20$. We obtain a primal problem of dimension 444528 and dual problem of 54972 unknows. The feasible set of dual problem is described by 3528 bound constraints and 3528 spherical constraints. Our problem is a variant of the problem that is described in more detail together with its discretization in Haslinger et al. [44].

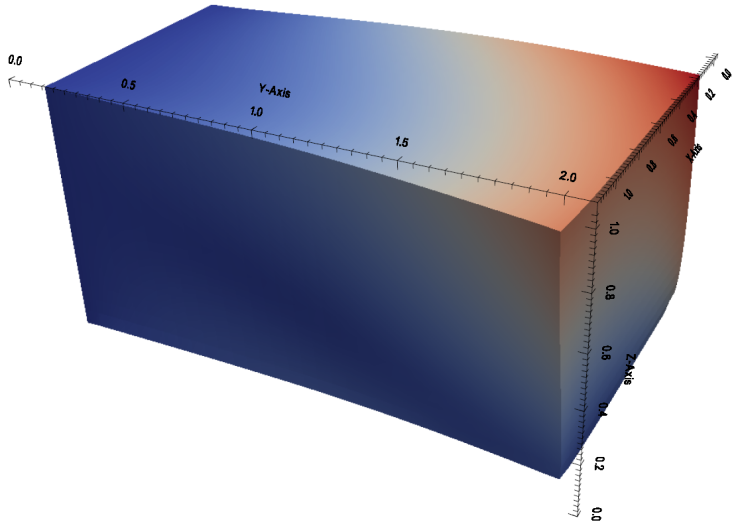


Figure 40: The primal solution of the problem of linear elasticity with friction - displacement.

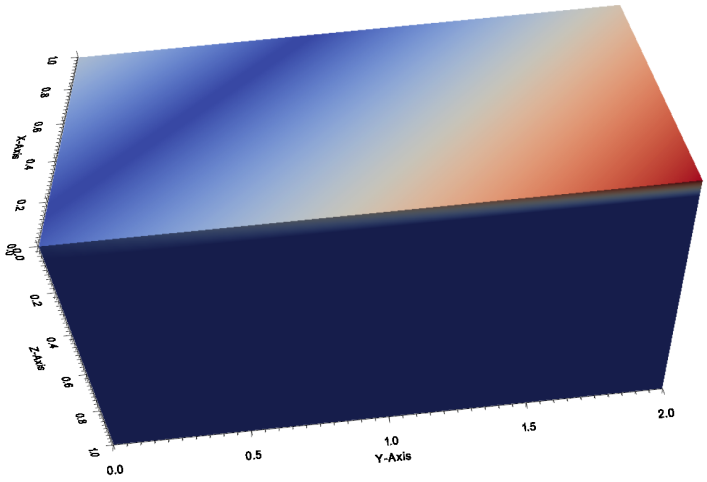


Figure 41: The dual solution of the problem of linear elasticity with friction. The magnitude of Lagrange multipliers corresponding to friction conditions. The figure shows the bottom side of original problem composition.

We have implemented MPGP, MPGP-BB, and SPG-QP in our PERMON toolbox. PERMON is our newly emerging set of tools for Parallel, Efficient, Robust, Modular, Object-oriented, Numerical simulations. It makes use of theoretical results in advanced quadratic programming algorithms, discretization and domain decomposition methods. The core solver layer of PERMON depends on PETSc and uses its coding style. More informations about the implementation can be found in Hapla [43]. The final solution of the problem was depicted using ParaView software [1], see Fig. 40 and 41.

The numerical results can be found in Table 12. We solve the problem with relative precision $\epsilon = 10^{-2}\|b\|$. In this table, we demonstrate the dependency of the number of active constraints in the solution and the value of friction coefficient. If the friction coefficient is small, then the number of active constraints is large. The PBB algorithm, which was plugged into original MPGP algorithm, decreases the number of projection steps. However, the PBB method is not convergent, see Section 2.4.2. The same observation was proposed by Pospíšil [58]. The SPG algorithm was not able to solve the problem in less then 50000 iterations. From these results, we suggest that the active-set algorithms are suitable for these problems.

ϕ		10	30	100	500
const.	active bound	98%	98%	98%	99%
	active quadratic	97%	89%	59%	48%
MPGP	SMALBE-M	3	3	4	4
	inner	15087	32877	35384	6148
	hess. mult.	26575	34674	35999	7157
	CG	11245	11830	8005	5145
	CG-half	2818	1793	610	1003
	projection	9690	19254	26769	1
MPGP-BB	SMALBE-M	4	3	4	4
	inner	15389	17417	12747	6148
	hess. mult.	18747	20207	14322	7157
	CG	11489	13258	9401	5145
	CG-half	3353	2786	1570	1003
	PBB	547	1373	1776	1
SPG-QP	SMALBE-M	3	3	4	2
	inner	50000*	50000*	50000*	50000*
	hess. mult.	-	-	-	-

Table 12: Number of constraints, and active constraints in the solution depends on the friction coefficient.

3.3 Granular dynamics

In this section, we will solve multibody dynamics problems. These problems may contain from hundreds to billions of discrete rigid bodies interacting through contact, impact, or mutual constraints, such as simulation of the movement of granular matter, are one of the most challenging issues in computer-aided kinematics and dynamics of mechanical systems. Many real-world systems contain or interact with granular material, as granular material belongs among the most manipulated materials. For instance, such a material is utilized in a variety of fields, from sand, gravel, or nanoscale powders to large boulders. Devices consisted of rigid bodies interacting through frictional contacts and mechanical joints pose numerical solution challenges because of the discontinuous nature of their motion, see Pfeiffer and Glocker [57].

Usually, the simulations are performed using discrete element method (DEM, see for instance Cundall [21], Avci and Wriggers [9]; a penalty method where the computation of interaction force is based on the kinematics of the interaction, some representative parameters, and an empirical force law.

From our point of view, an other method is more interesting. It is more similar to the solution of linear elasticity contact problems, because the problem is described by a differential variational inequality (DVI, see Pang and Stewart [56], Renouf and Alart [62], Heyn [47]). The method is sometimes referred to the Lagrange multiplier approach. It enforces the non-penetration of rigid bodies via a constraint-based approach. In the DVI method, a linear inequality constrained quadratic optimization problem with symmetric positive semidefinite Hessian matrix must be solved at each time step of the simulation. The unknowns in the problem are the normal contact forces between interacting bodies. Moreover, if we consider a problem with Coulomb friction, we obtain a quadratic programming problem with separable conical constraints and the additional unknowns represent frictional contact forces, see Anitescu and Tasora [8].

The efficient solution of inner optimization problem in DVI brings us to the development of Quadratic programming (QP) algorithms. Our research in the solution of particle dynamics simulations is motivated by the results achieved by Heyn et al. [48]. Authors used the MPRGP algorithm to solve DVI efficiently in spite of the fact that all theoretical results supporting the convergence of MPRGP were valid only for the strictly convex cost functions. Only recently, we successfully extended the theory and explained the convergence of the MPRGP for the problems with more general convex quadratic cost function, see Dostál and Pospíšil [35].

We are interested in simple simulations with sphere and box particles, but our algorithms can be easily generalized to particle problems with general geometry.

The first section consists of short review of the numerical solution concept and time-stepping scheme. In this thesis, we do not develop new simulations techniques or modify the mathematical modelling process. However, we are interested in numerical aspect of the inner QP optimization problem and our proofs of the solvability are based on the object structures. Therefore, we decided to present short review. The formulation of the problem and derivation of optimization problem can be found in the second subsection. The problems with friction are introduced in the third subsection. The presented theory and ideas in these subsections can be considered as a short review of Heyn [47]. Other details of computational dynamics and computer-aided kinematics can be studied, e.g., from Shabana [67] and Haug [45]. Section 3.3.4 introduces the mathematical aspect of the optimization problem solvability and other theoretical aspects.
The last subsection includes the numerical experiments and results. We have implemented algorithms in C programming language with CUDA library, and afterwards we perform simulations on GPU card. The problem of granular dynamics is ideal for solving on such a massively parallel architectures. However, the aim of the thesis is not to develop optimal implementation, but the development of algorithms. Therefore, we manage the problems constisting from small number of partlicles. Much more efficient implementation was presented by the team from Simulation-Based Engineering Lab (SBEL) University of Wisconsin-Madison in Chrono::Engine software [3].

3.3.1 Time-stepping scheme

In this section, we follow Heyn [47] and review the basic notations and ideas.
Let us consider the system of $n_b \in \mathbb{N}$ rigid bodies (*particles*) in the vector space $\{(x, y, z) \in \mathbb{R}^3\}$. Each particle has 6 degrees of freedom; location of the centre of gravity $[r_x, r_y, r_z]^T \in \mathbb{R}^3$ and the unit quaternion of rotation $[e_0, e_1, e_2, e_3]^T \in \mathbb{R}^4$. In this case, we can use also Euler angles to describe the rotation of the bodies, but such an approach evokes the evaluations of goniometric functions, which have to be performed numerically. This is the reason why quaternions are much more suitable; the rotation matrix can be created using simple operations such as multiplications and additions. Moreover, there exists a simple formula of the mapping matrix between position and velocity. More informations about computational dynamics can be found in Haug [45] and Heyn [47].

Let us denote

$$q_{(i)} := \begin{bmatrix} r_x \\ r_y \\ r_z \\ e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} \in \mathbb{R}^{3+4}, \quad v_{(i)} := \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \dot{\omega}_\phi \\ \dot{\omega}_\theta \\ \dot{\omega}_\psi \end{bmatrix} \in \mathbb{R}^{3+3}, \quad i = 1, \ldots, n_b$$

as a *vector of generalized position* and *generalized velocities* of body $T_{(i)}$ in given time t. Here, $\dot{\omega} \in \mathbb{R}^3$ denotes angular velocity in Euler angles.
We compose the vector of generalized positions and velocities from the vectors of positions and velocites from all bodies in the system and define

$$\begin{aligned} q &:= [q_{(1)}^T, \ldots, q_{(n_b)}^T]^T \in \mathbb{R}^{7n_b} , \\ v &:= [v_{(1)}^T, \ldots, v_{(n_b)}^T]^T \in \mathbb{R}^{6n_b} . \end{aligned}$$

There exists a linear mapping between derivative of position vector and vector of velocities $G_i(q_{(i)}) : \mathbb{R}^6 \to \mathbb{R}^7$ defined by prescription

$$G_i(q_{(i)}) := \left[I, \frac{1}{2} E^T \right], \quad E := \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix}, \quad \dot{q}_{(i)} = G_i(q_{(i)}) v_{(i)} .$$

The basic time-stepping schema (can be regarded as one step of implicit Euler method for the discretized First Newton law) has form

$$q^{(t+h)} = q^{(t)} + h.\dot{q}^{(t)} ,$$

where h is a *sufficiently small* time step and $q^{(t)}$ is a value of q in time t.

Let us suppose that we have already computed the position in time-step t and now, the problem is to find the vector of velocities $v^{(t+h)}$, which increment subject to $v^{(t)}$ depends on

- the mass of each body,
- external forces $F_{ext}(t, q, v)$,

- contacts and other limiting conditions.

This situation is described by the second Newton law, i.e.

$$M\dot{v} = F_C + F_{ext} \qquad M(v^{(t+h)} - v^{(t)}) = h(F_{ext} + F_C) \ , \tag{3.24}$$

where

- $M \in \mathbb{R}^{7n_b,7n_b}$ is generalized mass matrix,
- $F_C \in \mathbb{R}^{7n_b}$ is a vector of forces induced by contact constraints,
- $F_{ext} \in \mathbb{R}^{7n_b}$ is a vector of external forces.

3.3.2 Contacts

In this section, we review the determination of the contact force, which influences the change of the velocity.
Let us consider the contact between the bodies T_A and T_B (that means the body T_A is in the contact with the body T_B, the body T_A has effect to the body T_B - the order is important), see Fig. 42.
Let us denote $n_A(C)$ the unit outward normal of the body T_A in the contact point

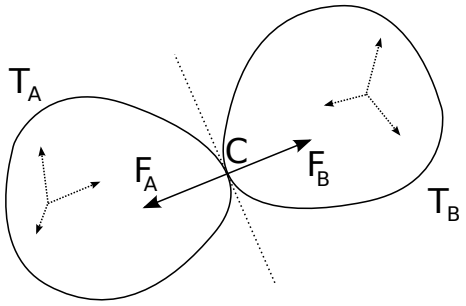


Figure 42: Contact between two bodies.

C in global coordinate system. Then the body T_A effects on the body T_B by force

$$F_B := \hat{\gamma} n_A(C) \ ,$$

where $\hat{\gamma} \geq 0$ is unknown size of the force. This force evocates the change of the position of the body T_B (the components of the generalized velocity vector corresponding to the position of gravity centre).
Analogously, the force

$$F_A := -\hat{\gamma} n_A(C)$$

effect to the body T_A.
Accordingly, the change of rotation of the body T_B (the components of the generalized velocity vector corresponding to the rotation) is effected by associated torque

$$M_B := C^B \times F_B \ ,$$

where C^B is the contact point C in local coordinate system of the body T_B (this coordinate system has the centre in T_B and coordinate axis correspond to coordinate system of this body with zero rotation). Using some manipulations, we obtain

$$M_B = C^B \times F_B = \hat{\gamma}(C^B \times n_A(C)) = \hat{\gamma}\widehat{C}^B n_A(C) \ ,$$

where

$$\widehat{C}^B := \begin{bmatrix} 0 & -C_z^B & C_y^B \\ C_z^B & 0 & -C_x^B \\ -C_y^B & C_x^B & 0 \end{bmatrix} \in \mathbb{R}^{3,3}$$

is a *vector product matrix*.
Analogously, the torque

$$M_A := C^A \times F_A = -\hat{\gamma}\widehat{C}^A n_A(C)$$

is acting to the body T_A.
For the sake of simplicity, let us define the vector

$$d^{AB} := \begin{bmatrix} -n_A(C) \\ -\widehat{C}^A n_A(C) \\ n_A(C) \\ \widehat{C}^B n_A(C) \end{bmatrix} \in \mathbb{R}^{4\cdot 3} \ . \tag{3.25}$$

Sometimes this vector is reffered as the *tangent space generator*. It is used to transform the contact forces from local to global frame.
If the problem consists of $n_c \in \mathbb{N}$ contacts (it will always consist of more contact - even if there are only two bodies, the number of contacts is two - contact of T_A with T_B and contact of T_B with T_A), the column vectors $d_i, i = 1, \ldots, n_c$ can be formed into the block matrix $D \in \mathbb{R}^{6n_b, n_c}$ and the vector of the unknown sizes of contact forces is given by $\hat{\gamma} = [\hat{\gamma}_1, \ldots, \hat{\gamma}_{n_c}]^T \in \mathbb{R}^{n_c}$. See following Example 3.3.1.

If the size of the force $\hat{\gamma}$ is known, then the resulting change of the velocity can be computed by (by inducting previous observations to (3.24))

$$M(v^{(t+h)} - v^{(t)}) = hF_{ext} + D\gamma \ ,$$

where $\gamma := h\tilde{\gamma}$.

Example 3.3.1
Let us consider the system of three bodies T_A, T_B, T_C. The vector of generalized positions is given by $q = [q_A^T, q_B^T, q_C^T]^T \in \mathbb{R}^{3\cdot 7}$ and the vector of generalized velocities by $v = [v_A^T, v_B^T, v_C^T]^T \in \mathbb{R}^{3\cdot 6}$. Let positions of the bodies be the same as in Fig. 43.

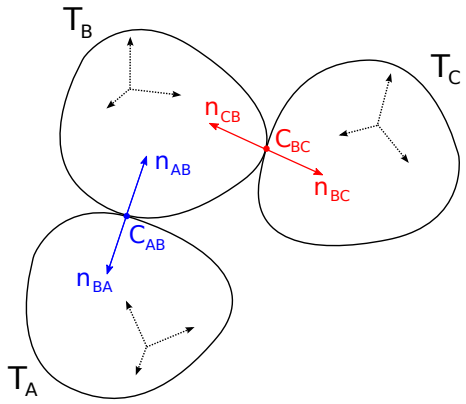


Figure 43: Example of the contact of more bodies.

In figure, we denoted these objects:

- T_A, T_B, T_C - the objects (bodies, particles),
- $n_{AB}, n_{BA}, n_{BC}, n_{CB}$ - the unit outward normals; for example n_{AB} denotes the unit outward normal of the body T_A in the contact point C_{AB},
- C_{AB}, C_{BC} - contact points in global coordinates (obviously $C_{AB} = C_{BA}, C_{BC} = C_{CB}$).

Moreover, we denote by C_{AB}^A the coordinates of the contact point C_{AB} in local coordinate system of the body T_A and similarly we denote $C_{AB}^B, C_{BC}^B, C_{BC}^C$.

In this case, we are searching for the sizes of four unknown forces:

- the force, which the body T_A applies to the body T_B resulting from the contact between T_A and T_B,
- the force, which the body T_B applies to the body T_A resulting from the contact between T_B and T_A,
- the force, which the body T_B applies to the body T_C resulting from the contact between T_B and T_C,
- the force, which the body T_C applies to the body T_B resulting from the contact between T_C and T_B.

Thus we are searching for $\gamma = [\gamma_{AB}, \gamma_{BA}, \gamma_{BC}, \gamma_{CB}]^T \in \mathbb{R}^4$.

The matrix $D \in \mathbb{R}^{3\cdot 6,4}$ has a block structure

$$D = \begin{bmatrix} -n_{AB} & n_{BA} & 0 & 0 \\ -\tilde{C}^A_{AB} n_{AB} & -\tilde{C}^A_{AB} n_{BA} & 0 & 0 \\ n_{AB} & -n_{BA} & -n_{BC} & n_{CB} \\ \tilde{C}^A_{AB} n_{AB} & -\tilde{C}^B_{AB} n_{AB} & -\tilde{C}^B_{BC} n_{BC} & \tilde{C}^B_{BC} n_{CB} \\ 0 & 0 & n_{BC} & -n_{CB} \\ 0 & 0 & \tilde{C}^C_{BC} n_{BC} & -\tilde{C}^C_{BC} n_{CB} \end{bmatrix} . \tag{3.26}$$

■

The non-penetration condition of the bodies T_A and T_B can be described by *gap function* $\Phi : \mathbb{R}^{7+7} \to \mathbb{R}$. In the simplest case, it can be considered as the "distance between bodies" (the minimum distance between the points of the boundary of the first and the second body).
It holds

- $\Phi([q_A, q_B]) = 0$ if the bodies are in contact,
- $\Phi([q_A, q_B]) > 0$ if the bodies are not in contact,
- $\Phi([q_A, q_B]) < 0$ if the bodies penetrate each other.

For the size of the affecting force γ, it holds analogically

- $\gamma \geq 0$ if the bodies are in contact (the force exists),
- $\gamma = 0$ if the bodies are not in contact (the force does not exist).

Merging two last observations, we obtain the *complementarity condition*

$$0 \leq \Phi(q) \perp \gamma \geq 0 \, . \tag{3.27}$$

Instead of these conditions, we can consider more numerically stable conditions (see Anitescu [6])

$$0 \leq \frac{1}{h}\Phi(q) + D^T v^{(t+h)} \perp \gamma \geq 0 \, . \tag{3.28}$$

The next theorem shows how to reformulate the problem with these conditions to the quadratic problem with bound constraints.

Lemma 3.3.1
(ABOUT REFORMULATION OF THE PROBLEM)

The solution of the optimization problem

$$\min_{\gamma \geq 0} \frac{1}{2}\gamma^T N\gamma + r^T\gamma \ , \tag{3.29}$$

where

$$N := D^T M^{-1} D, \tag{3.30a}$$
$$r := \frac{1}{h}\Phi + D^T M^{-1} k, \tag{3.30b}$$
$$k := Mv^{(t)} + h.F_{ext}, \tag{3.30c}$$

is equivalent to the solution of original problem

$$M(v^{(t+h)} - v^{(t)}) = hF_{ext} + D^T\gamma, \tag{3.31a}$$
$$\frac{1}{h}\Phi(q) + D^T v^{(t+h)} \geq 0, \tag{3.31b}$$
$$\frac{1}{h}\Phi(q) + D^T v^{(t+h)} \perp \gamma, \tag{3.31c}$$
$$\gamma \geq 0. \tag{3.31d}$$

Proof: The Lagrange function of the optimization problem (3.29) is given by

$$L(\gamma, \lambda) = \frac{1}{2}\gamma^T N\gamma + r^T\gamma - \lambda^T\gamma,$$

so the appropriate Karush-Kuhn-Tucker conditions have form

$$N\gamma + r - \lambda = 0, \tag{3.32a}$$
$$\gamma \geq 0, \tag{3.32b}$$
$$\lambda \geq 0, \tag{3.32c}$$
$$\gamma^T\lambda = 0. \tag{3.32d}$$

Conditions (3.32b) and (3.31d) are the same. From condition (3.32a), we can directly derive

$$\lambda = N\gamma + r \tag{3.33}$$

and together with (3.30) and adaption to (3.32d), we obtain

$$\begin{aligned} 0 &= \gamma^T \lambda = \gamma^T (N\gamma + r) \\ &= \gamma^T (D^T M^{-1} D\gamma + \tfrac{1}{h}\Phi + D^T M^{-1} k) \\ &= \gamma^T (D^T M^{-1} D\gamma + \tfrac{1}{h}\Phi + D^T M^{-1} (M v^{(t)} + h.F_{ext})) \\ &= \gamma^T (\tfrac{1}{h}\Phi + D^T (v^{(t)} + M^{-1} D\gamma + h.M^{-1} F_{ext})). \end{aligned}$$

Furthermore, using (3.31a) we get

$$0 = \gamma^T (\frac{1}{h}\Phi + D^T (v^{(t+h)} - v^{(t)}),$$

which is in fact condition (3.31c).
Finally, we can adapt (3.33) into (3.32c) and we get

$$0 \leq \lambda = N\gamma + r = \frac{1}{h}\Phi + D^T v^{(t+h)} \ ,$$

which is condition (3.31b). □

So, the algorithm in every time-step has the form of Algorithm 13.

3.3.3 Friction

In problems with friction, the situation is quite similar. Let us consider two bodies T_A and T_B in the contact. Then, we can denote

- $n \in \mathbb{R}^3$ unit outward normal in global coordinates,
- $u, w \in \mathbb{R}^3$ unit orthogonal direction vectors of contact plane in global coordinates.

Obviously, $\{n, u, w\}$ is a set of orthonormal vectors and $\{u, w\}$ forms the basis of *tangent space*.

Friction force affected in contact point to the body T_A can be expressed by

$$F = F_n + F_T = \gamma_n n + \gamma_u u + \gamma_w w = S\gamma \ ,$$

where

- $F_n = \gamma_n n \ \in \mathbb{R}^3$ is normal component of the contact force,
- $F_T = \gamma_u u + \gamma_w w \ \in \mathbb{R}^3$ is tangent component of the contact force,
- $\gamma_n > 0$ is a size of normal component of the contact force,

Algorithm 13: **Time-stepping schema.**

Given $t, h, q^{(t)}, v^{(t)}$.

find contacts

if *there is a contact*

set up N, r *from contacts*

solve the problem

$$\gamma := \arg\min_{\gamma \geq 0} \frac{1}{2}\gamma^T N\gamma + r^T\gamma$$

$v^{(t+h)} := v^{(t)} + M^{-1}(hF_{ext} + D\gamma)$

else

$v^{(t+h)} := v^{(t)} + hM^{-1}F_{ext}$

endif

$q^{(t+h)} := q^{(t)} + h.G(q^{(t)})v^{(t)}$

Return $q^{(t+h)}, v^{(t+h)}$.

- $\gamma_u, \gamma_w \in \mathbb{R}$ are sizes of tangent components of the contact force,
- $\gamma := [\gamma_n, \gamma_u, \gamma_w]^T \in \mathbb{R}^3$ is a vector of unknown component sizes,
- $S := [n, u, w] \in \mathbb{R}^{3,3}$ is orthogonal matrix.

Relation between components of the friction force can be described by *Coulomb friction model* (3.34), see Anitescu and Potra [7],

$$\gamma_n \geq 0, \quad \Phi(q) \geq 0, \quad \Phi(q)\gamma_n = 0 \ , \tag{3.34a}$$

$$\sqrt{\gamma_u^2 + \gamma_w^2} \leq \mu\gamma_n \ , \tag{3.34b}$$

$$\|v_T\| \left(\mu\gamma_n - \sqrt{\gamma_u^2 + \gamma_w^2}\right) = 0 \ , \tag{3.34c}$$

$$\langle F_T, v_T\rangle = -\|F_T\|.\|v_T\| \ . \tag{3.34d}$$

Lemma 3.3.2
(ABOUT REFORMULATION OF THE PROBLEM WITH FRICTION)

The solution of the optimization problem

$$\min_{\gamma\in\Omega} \frac{1}{2}\gamma^T N\gamma + r^T\gamma \ , \tag{3.35}$$

where

$$N := D^T M^{-1} D,$$
$$r := [\tfrac{1}{h}\Phi, 0, 0]^T + D^T M^{-1} k,$$
$$k := Mv^{(t)} + h.F_{ext},$$
$$\Omega := \Omega_1 \times \cdots \times \Omega_{nc}, \quad \Omega_j := \{[x, y, z]^T \in \mathbb{R}^3 : \sqrt{y^2 + z^2} \leq \mu_i x\}$$

is equivalent to the solution of original problem

$$M(v^{(t+h)} - v^{(t)}) = hF_{ext} + D^T\gamma$$

with conditions (3.34).

Proof: See Heyn [47]. □

3.3.4 Properties and solvability

Because we are mostly interested in the solution of inner optimization problems (3.29) and (3.35), we should discuss the solvability of these problems. In Heyn [47], one can find this physical explanation:

> *For example, consider a rigid symmetric four-legged table resting on a perfectly flat rigid plane. The vertical reaction forces at the four legs are non-unique. For example, if the weight of the table is* 100N*, it is possible that the vertical reaction force at each leg is* 25N*. However, it is equally possible that two diagonally opposite legs have reaction forces of* 30N *each, while the other pair of diagonally opposite legs have reaction forces of* 20N *each. In fact, there are infinitely many sets of reaction forces which satisfy the equations corresponding to the case of the rigid table resting on a rigid plane.*

This is the reason, why the inner optimization problem has always solution (from physical point of view) and in some cases, it can have infinite number of solutions. In the thesis, we focus on the mathematical aspects of the solvability.

The problem without friction

In this section, we prove that the optimization problem in particle dynamics without friction is QP problem with right hand-side vector from image of the Hessian matrix N.

Lemma 3.3.3
(PROPERTY OF QP IN PROBLEMS WITHOUT FRICTION)

The optimization problem (3.29) *is a QP problem with SPS Hessian and right-hand side vector from the image of Hessian, i.e.*

$$\forall x \in \mathbb{R}^{n_c} : \langle Nx, x\rangle \geq 0, \tag{3.36a}$$

$$r \in \operatorname{Im} N. \tag{3.36b}$$

Proof: At first, we prove (3.36a). It is necessary to show that $\forall x \in \mathbb{R}^{n_c} : \langle Nx, x\rangle \geq 0$. We use that $M \in \mathbb{R}^{6n_b,6n_b}$ is SPD. Therefore, the inverse is also SPD and it induces the norm in $\mathbb{R}^{6n_b}$. Furthermore, we can write

$$\langle Nx, x\rangle = \langle D^T M^{-1} Dx, x\rangle = \langle M^{-1} Dx, Dx\rangle = \|Dx\|^2_{M^{-1}} \geq 0 \ .$$

If we take a look at the prescription of right-hand side vector

$$r := \frac{1}{h}\Phi + D^T M^{-1} k \ ,$$

we can use Lemma 1.3.1 and simplify the proof of (3.36b) into the proof of

$$\Phi \in \operatorname{Im} D^T \ . \tag{3.37}$$

At first, let us define the vector spaces $\mathcal{V}, \mathcal{W} \subset \mathbb{R}^{nc}$

$$\mathcal{V} := \operatorname{span}\left\{\begin{bmatrix}1\\-1\\0\\0\\\vdots\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\\\vdots\\0\\0\end{bmatrix}, \ldots \begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\-1\end{bmatrix}\right\}, \quad \mathcal{W} := \operatorname{span}\left\{\begin{bmatrix}1\\1\\0\\0\\\vdots\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\1\\\vdots\\0\\0\end{bmatrix}, \ldots \begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\1\end{bmatrix}\right\} .$$

We take a better look into the structure of matrix $D \in \mathbb{R}^{6n_b,n_c}$ whose blocks are given by (3.25). This matrix always consists of the pairs of contacts - the contact between

the body T_A and the body T_B (denoted by contact AB) as well as the contact between body B and A (denoted by contact BA), see Example 3.3.1. For the sake of simplicity we consider the construction of matrix $D \in \mathbb{R}^{6n_b,n_c}$ with consecutive collocation of the contact pairs. Then each pair of columns corresponds to the pair of contacts and the nonzero rows corresponding to the indexes of the bodies in this contact is given by prescription (3.25), i.e. the filled submatrix for one pair of contacts AB and BA has structure

$$\begin{aligned} D_{\mathcal{R},\mathcal{C}} &= \begin{bmatrix} -n_A(C_{AB}) & n_B(C_{BA}) \\ -\widetilde{C}^A_{AB} n_A(C_{AB}) & \widetilde{C}^A_{AB} n_B(C_{BA}) \\ n_A(C_{AB}) & -n_B(C_{BA}) \\ \widetilde{C}^B_{AB} n_A(C_{AB}) & \widetilde{C}^B_{BA} n_B(C_{BA}) \end{bmatrix}, \\ \mathcal{R} &:= \{\text{indexes of } v_A, \text{indexes of } v_B\}, \\ \mathcal{C} &:= \{\text{index of contact } AB, \text{index of contact } BA\} \ . \end{aligned} \tag{3.38}$$

The key ingredient of the proof is a small observation; the contact forces in the contact point $C_{AB} = C_{BA}$ have opposite direction and the unit outward normals of the bodies in the contact are also opposite. This observation implies $n_A = -n_B$.
Therefore, the submatrix (3.38) can be written in simplier form

$$D_{\mathcal{R},\mathcal{C}} = \begin{bmatrix} -n_A & -n_A \\ -\widetilde{C}^A n_A & -\widetilde{C}^A n_A \\ n_A & n_A \\ \widetilde{C}^B n_A & \widetilde{C}^B n_A \end{bmatrix},$$

where we used notations $n_A := n_A(C_{AB}) = -n_B(C_{BA})$ and $C^A := C^A_{AB} = C^A_{BA}, C^B := C^B_{AB} = C^B_{BA}$.
Using this simple structure, it is easy to check that

$$\begin{aligned} \forall v \in \mathcal{V} : &\quad Dv = 0, \\ \forall w \in \mathcal{W} \setminus \{0\} : &\quad Dw \neq 0. \end{aligned}$$

Furthermore, we proved that $\mathcal{V} \subset \operatorname{Ker} D$ and $\mathcal{W} \cap \operatorname{Ker} D = \{0\}$. Using this and Theorem 1.3.1, we can write

$$\left.\begin{array}{l} \operatorname{Ker} D \supset \mathcal{V} \perp \mathcal{W} \\ \mathcal{W} \cap \operatorname{Ker} D = \{0\} \end{array}\right\} \quad \Rightarrow \quad \mathcal{W} \perp \operatorname{Ker} D \quad \Rightarrow \quad \mathcal{W} \subset \operatorname{Im} D^T \ .$$

To prove (3.37), notice that $\Phi \in \mathcal{W}$ because the gap function has the same value for both contacts AB and BA (the distance between bodies T_A and T_B is the same as distance between bodies T_B and T_A).

□

Example 3.3.2
Let us demonstrate the basic ideas from the previous proof on the matrix from Example 3.3.1. From the Fig. 43, we can suggest

$$n_{AB} = -n_{BA} \quad \text{and} \quad n_{BC} = -n_{CB} \ ,$$

so the matrix (3.26) can be written in the form

$$D = \begin{bmatrix} -n_{AB} & -n_{AB} & 0 & 0 \\ -\widetilde{C}^A_{AB} n_{AB} & -\widetilde{C}^A_{AB} n_{AB} & 0 & 0 \\ n_{AB} & n_{AB} & -n_{BC} & -n_{BC} \\ \widetilde{C}^A_{AB} n_{AB} & \widetilde{C}^A_{AB} n_{AB} & -\widetilde{C}^B_{BC} n_{BC} & -\widetilde{C}^B_{BC} n_{BC} \\ 0 & 0 & n_{BC} & n_{BC} \\ 0 & 0 & \widetilde{C}^C_{BC} n_{BC} & \widetilde{C}^C_{BC} n_{BC} \end{bmatrix} .$$

We choose arbitrary two nonzero vectors

$$\begin{aligned} v &:= [\alpha, -\alpha, \beta, -\beta]^T \in \mathcal{V}, \qquad \alpha, \beta \in \mathbb{R}, \\ w &:= [\gamma, \gamma, \delta, \delta]^T \in \mathcal{W}, \qquad \gamma, \delta \in \mathbb{R}. \end{aligned}$$

These vectors are orthogonal

$$\langle v, w \rangle = \alpha.\gamma + (-\alpha).\gamma + \beta.\delta + (-\beta).\delta = 0.$$

Moreover, multiplying these vectors by matrix D we get

$$Dv = \begin{bmatrix} -n_{AB} & -n_{AB} & 0 & 0 \\ -\widetilde{C}^A_{AB} n_{AB} & -\widetilde{C}^A_{AB} n_{AB} & 0 & 0 \\ n_{AB} & n_{AB} & -n_{BC} & -n_{BC} \\ \widetilde{C}^A_{AB} n_{AB} & \widetilde{C}^A_{AB} n_{AB} & -\widetilde{C}^B_{BC} n_{BC} & -\widetilde{C}^B_{BC} n_{BC} \\ 0 & 0 & n_{BC} & n_{BC} \\ 0 & 0 & \widetilde{C}^C_{BC} n_{BC} & \widetilde{C}^C_{BC} n_{BC} \end{bmatrix} \begin{bmatrix} \alpha \\ -\alpha \\ \beta \\ -\beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} ,$$

$$Dw = D \begin{bmatrix} \gamma \\ \gamma \\ \delta \\ \delta \end{bmatrix} = \begin{bmatrix} -2\gamma n_{AB} \\ -2\gamma \widetilde{C}^A_{AB} n_{AB} \\ 2\gamma n_{AB} - 2\delta n_{BC} \\ 2\gamma \widetilde{C}^A_{AB} n_{AB} - 2\delta \widetilde{C}^B_{BC} n_{BC} \\ 2\delta n_{BC} \\ 2\delta \widetilde{C}^C_{BC} n_{BC} \end{bmatrix} .$$

Since $n_{AB} \neq o$, $n_{BC} \neq o$ and γ, δ are not both equal to zero, we can suggest that $Dw \neq 0$.

■

Theorem 3.3.1
(ABOUT SOLVABILITY OF PROBLEMS WITHOUT FRICTION)

The optimization problem (3.29) *has always a solution.*

Proof: The statement is easy corollary of Lemma 3.3.3, Lemma 1.3.4 and classical results given by Frank and Wolfe [39]. Since the Hessian matrix is SPS, there exists a possibility of infinite number of solutions. □

The problem with friction

In this section, we generalize our results from previous section to problems with friction.

Lemma 3.3.4
(PROPERTY OF QP IN PROBLEMS WITH FRICTION)

The optimization problem (3.35) *is a QP problem with SPS Hessian and right-hand side vector from the image of Hessian, i.e.*

$$\forall x \in \mathbb{R}^{n_c} : \langle Nx, x \rangle \geq 0, \tag{3.39a}$$

$$r \in \operatorname{Im} N. \tag{3.39b}$$

Proof: The first part of the proof is the same as in non-friction case, see proof of Lemma 3.3.4. To prove (3.39b) it is necessary to show that

$$\phi \in \operatorname{Im} D^T. \tag{3.40}$$

The following ideas are similar to the proof on non-friction case. We consider bodies T_A, T_B in contact. In fact, there exists two contacts; the contact of body T_A with T_B and contact between T_B and T_A. In this case, the submatrix of matrix D which corresponds

to these contacts is given by

$$
\begin{aligned}
D_{\mathcal{R},\mathcal{C}} &= \begin{bmatrix} -S_{AB} & S_{BA} \\ -\widetilde{C}^A_{AB} S_{AB} & \widetilde{C}^A_{AB} S_{BA} \\ S_{AB} & -S_{BA} \\ \widetilde{C}^B_{AB} S_{AB} & \widetilde{C}^B_{BA} S_{BA} \end{bmatrix}, \\
\mathcal{R} &:= \{\text{indexes of } v_A, \text{indexes of } v_B\}, \\
\mathcal{C} &:= \{\text{indexes of } \{\gamma_n^{AB}, \gamma_u^{AB}, \gamma_w^{AB}, \}, \text{indexes of } \{\gamma_n^{BA}, \gamma_u^{BA}, \gamma_w^{BA}, \}\}, \\
S_{AB} &:= [n_A(C_{AB}), u_A(C_{AB}), w_A(C_{AB})] \\
S_{AB} &:= [n_B(C_{BA}), u_B(C_{BA}), w_B(C_{BA})] \ .
\end{aligned}
\tag{3.41}
$$

Similarly to non-friction case $C_{AB} = C_{BA}$ and $n_A = -n_B$, see Fig. 44. Moreover, the

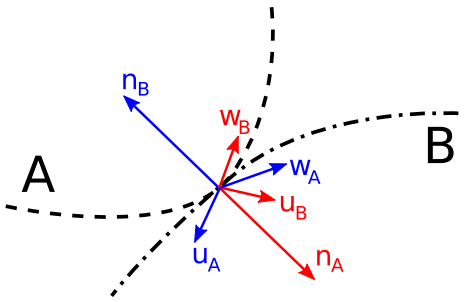


Figure 44: Tangent space in problems with friction.

tangent plane of both contacts is the same. This implies that the orthogonal tangent space generators $\{u_A, w_A\}$ and $\{u_B, w_B\}$ represent orthogonal basis of the same subspace of $\mathbb{R}^3$. Therefore, there exists a matrix $K \in \mathbb{R}^{3,3}$ such that

$$
S_{AB} = -S_{BA} K. \tag{3.42}
$$

This matrix represents the rotation or the flip of the orthogonal basis $\{n_A, u_A, w_A\}$ to the basis $\{n_B, u_B, w_B\}$. Since $S_{AB} = [n_A, u_A, w_A]$, and $S_{BA} = [n_B, u_B, w_B]$ are orthogonal, we can compute K from (3.42)

$$
K = -S_{BA}^T S_{AB} = -\begin{bmatrix} -n_A^T \\ u_B^T \\ w_B^T \end{bmatrix} [n_A \ u_A \ w_A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\langle u_B, u_A \rangle & -\langle u_B, w_A \rangle \\ 0 & -\langle w_B, u_A \rangle & -\langle w_B, w_A \rangle \end{bmatrix}. \tag{3.43}
$$

Using these observations, we can write (3.41) in form

$$D_{\mathcal{R},\mathcal{C}} = \begin{bmatrix} -S_A & -S_A K \\ -\widetilde{C}^A S_A & -\widetilde{C}^A S_A K \\ S_A & S_A K \\ \widetilde{C}^A S_A & \widetilde{C}^B S_A K \end{bmatrix}, \tag{3.44}$$

where we used notations $S_A := S_{AB} = -S_{BA}K, C^A := C^A_{AB} = C^A_{BA}$, and $C^B := C^B_{AB} = C^B_{BA}$.
Now we are ready to introduce vector spaces

$$\begin{aligned}
\mathcal{V} \;:=\; \text{span}\,\{\; & [1,0,0,-1,0,0 \quad 0,0,0,0,0,0, \quad \dots \quad 0,0,0,0,0,0]^T, \\
& [0,0,0,0,0,0 \quad 1,0,0,-1,0,0, \quad \dots \quad 0,0,0,0,0,0]^T, \\
& \qquad \dots \\
& [0,0,0,0,0,0 \quad 0,0,0,0,0,0, \quad \dots \quad 1,0,0,-1,0,0]^T \;\} \\
\mathcal{W} \;:=\; \text{span}\,\{\; & [1,0,0,1,0,0 \quad 0,0,0,0,0,0, \quad \dots \quad 0,0,0,0,0,0]^T, \\
& [0,0,0,0,0,0 \quad 1,0,0,1,0,0, \quad \dots \quad 0,0,0,0,0,0]^T, \\
& \qquad \dots \\
& [0,0,0,0,0,0 \quad 0,0,0,0,0,0, \quad \dots \quad 1,0,0,1,0,0]^T \;\}.
\end{aligned}$$

Notice that for any $\alpha \in \mathbb{R}$ it holds

$$K \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\langle u_B, u_A \rangle & -\langle u_B, w_A \rangle \\ 0 & -\langle w_B, u_A \rangle & -\langle w_B, w_A \rangle \end{bmatrix} \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$$

and afterwards, using (3.44) it is easy to check that

$$\begin{aligned} \forall v \in \mathcal{V}: \quad & Dv = 0, \\ \forall w \in \mathcal{W} \setminus \{0\}: \quad & Dw \neq 0. \end{aligned}$$

Finally, we can use Theorem 1.3.1 and the same arguments as in the non-friction case

$$\left.\begin{array}{l} \operatorname{Ker} D \supset \mathcal{V} \perp \mathcal{W} \\ \mathcal{W} \cap \operatorname{Ker} D = \{0\} \end{array}\right\} \;\Rightarrow\; \mathcal{W} \perp \operatorname{Ker} D \;\Rightarrow\; \mathcal{W} \subset \operatorname{Im} D^T .$$

To prove (3.40), notice that $\Phi \in \mathcal{W}$ because the gap function has the same value for both contacts AB and BA (the distance between bodies T_A and T_B is the same as distance between bodies T_B and T_A). □

Theorem 3.3.2
(ABOUT SOLVABILITY OF PROBLEMS WITH FRICTION)

The optimization problem (3.35) *has always a solution.*

Proof: The statement is easy corollary of Lemma 3.3.4, Lemma 1.3.4, and classical results given by Frank and Wolfe [39]. Since the Hessian matrix is SPS, there exists a possibility of infinite number of solutions. □

Independence of solution

Optimization problems (3.29) and (3.35) can have infinite number of solutions. The next lemma shows, that it is not important which solution we choose to update velocity. Therefore, the aim of the solver is to find arbitrary solution of the problem.

Lemma 3.3.5
(INDEPENDENCE OF SOLUTION AND VELOCITY.)

The velocity in the next time-step $v^{(t+h)}$ *given by Algorithm 13 is independent of the choice of the solution of the optimization problem* (3.29) *in non-friction case, or* (3.35) *in friction case.*

Proof: Let $\bar{\gamma}_1, \bar{\gamma}_2$ denote different solutions of optimization problem (3.29) or (3.35). Then appropriate velocity in the next time-step given by Algorithm 13 depending on these solutions is given by

$$\begin{aligned} v^{(t+h)}(\bar{\gamma}_1) &= v^{(t)} + M^{-1}(hF_{ext} + D\bar{\gamma}_1), \\ v^{(t+h)}(\bar{\gamma}_2) &= v^{(t)} + M^{-1}(hF_{ext} + D\bar{\gamma}_2). \end{aligned}$$

We examine the difference of these velocities

$$v^{(t+h)}(\bar{\gamma}_1) - v^{(t+h)}(\bar{\gamma}_2) = M^{-1}D(\bar{\gamma}_1 - \bar{\gamma}_2) \ .$$

Since the difference of the solutions lies in the Ker N (see Lemma 1.3.5) and Ker N = Ker D (see Lemma 1.3.1), we can write

$$v^{(t+h)}(\bar{\gamma}_1) - v^{(t+h)}(\bar{\gamma}_2) = M^{-1}0 = 0 \ .$$

□

3.3.5 Numerical experiments

In the thesis, we present the numerical results on two simple benchmarks. The first one is without friction and the second one is with the particles with various friction coefficients.
Algorithm was implemented in C programming language with CUDA environment [4]. For contact detection, we are using our own implementation of the Moving Bounding-Box algorithm [64] and for generating the images, we are using POV-Ray Software [2].
In our experiments, we demand the relative stopping tolerance

$$\|g^P(\gamma)\| \leq 10^{-4} \cdot \|r\|.$$

We compare MPRGPS, PBBf, SPG-QP and APGD. Algorithms settings can be found in Table 14.

all solvers	$\bar{\alpha} = 1.95/\tilde{\lambda}^A_{\max}, x^0 = 0, \varepsilon_{in} = 10^{-4}, it_{\max} = 10^4$
MPRGPS	$\Gamma = 1$
PBBf	$K = 10, x^1 = P_\Omega(x^0 - \bar{\alpha} g^0)$
SPG-QP	$m = 10, \gamma = 0.1, \sigma_2 = 0.9999, \alpha_0 = \bar{\alpha}$
APGD	$L = \tilde{\lambda}^A_{\max}$

Table 14: Algorithms settings in granular dynamics benchmarks.

In the first benchmark without friction, we consider a system of 32810 spherical particles. During the first stage of the simulation, small particles are scattered into simple box represented by five walls. The initial position of the particles and final position can be found in Fig. 45 and Fig. 46. Afterwards in $t = 0.3$ s, we add large spherical particle to study the behaviour of the impact. The material of the bodies is represented by density $\rho = 2800$ kg $\cdot$ m^{-3}. Small particles have radius $r = 0.011$ m and the large one $r_2 = 0.15$ m. The stepsize of the time-stepping scheme is $h = 8 \cdot 10^{-4}$ s.

The number of bodies in the system and the number of contacts can be found in Fig. 47. The number of iterations depends on the dimension of the inner problem, see Fig. 48. The dimension of inner minimization problem with bound constraints is equal to the number of contacts. During our simulation, not all algorithms were able to solve the problem using 15000 iterations, so we decided to stop them. The next step of the simulation was computed by SPG-QP, which has the best efficiency. The perfomance profiles were constructed from all optimization problems during the simulation and can be found in Fig. 49.

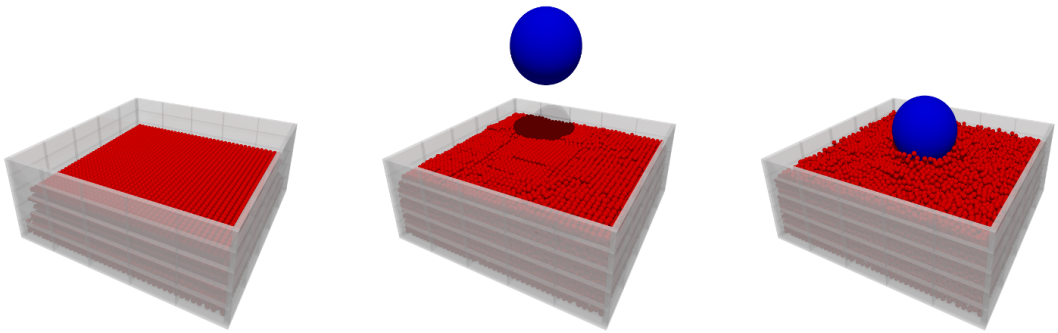


Figure 45: The state of the first benchmark in $t = 0.136$ s, 0.4 s, 0.56 s.

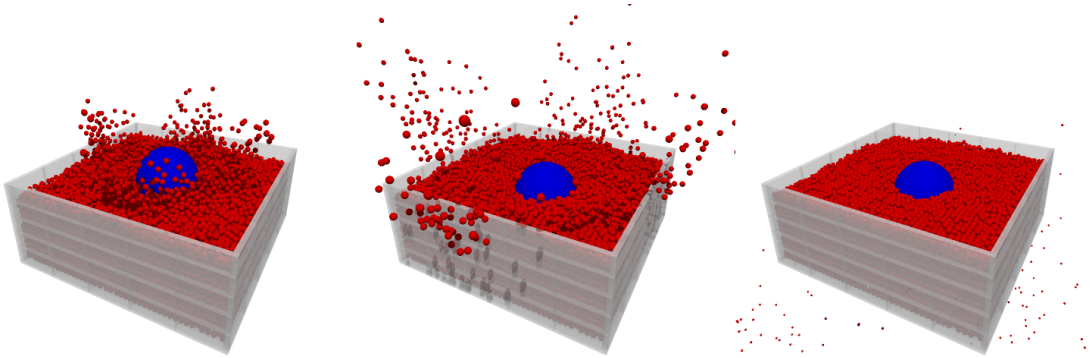


Figure 46: The state of the first benchmark in $t = 0.64$ s, 0.8 s, 1.44 s.

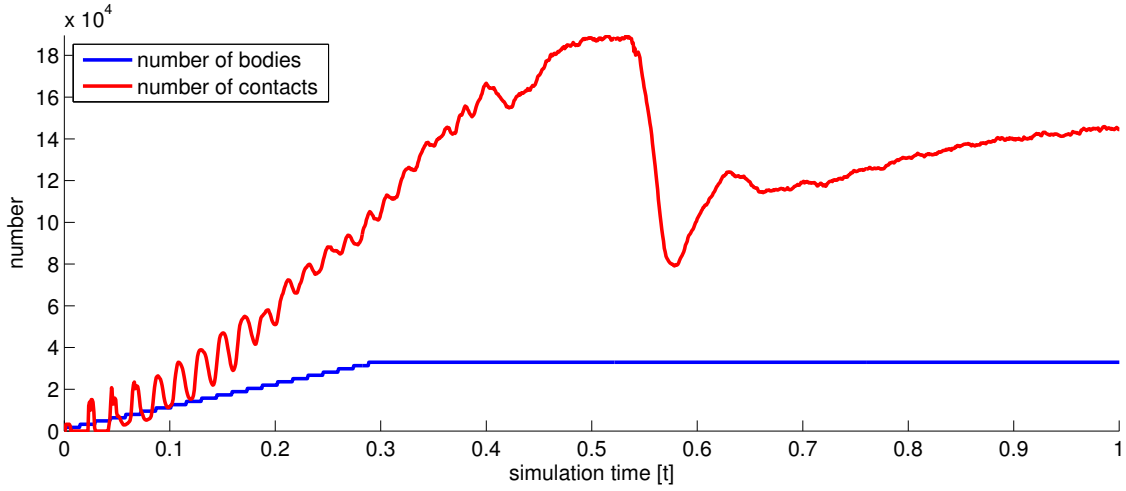


Figure 47: The number of bodies and contacts in the system during the first simulation without friction.

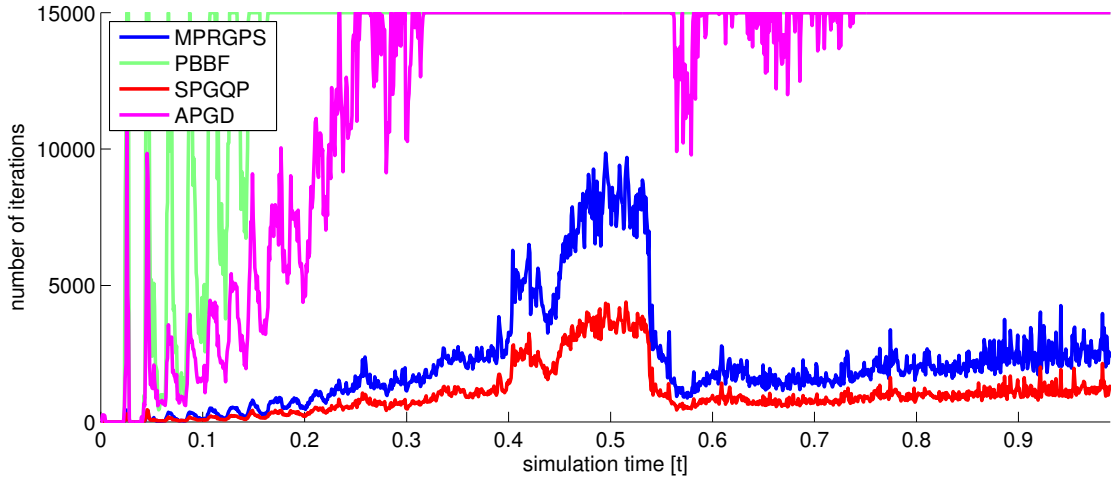


Figure 48: The number of iterations during the first simulation without friction.

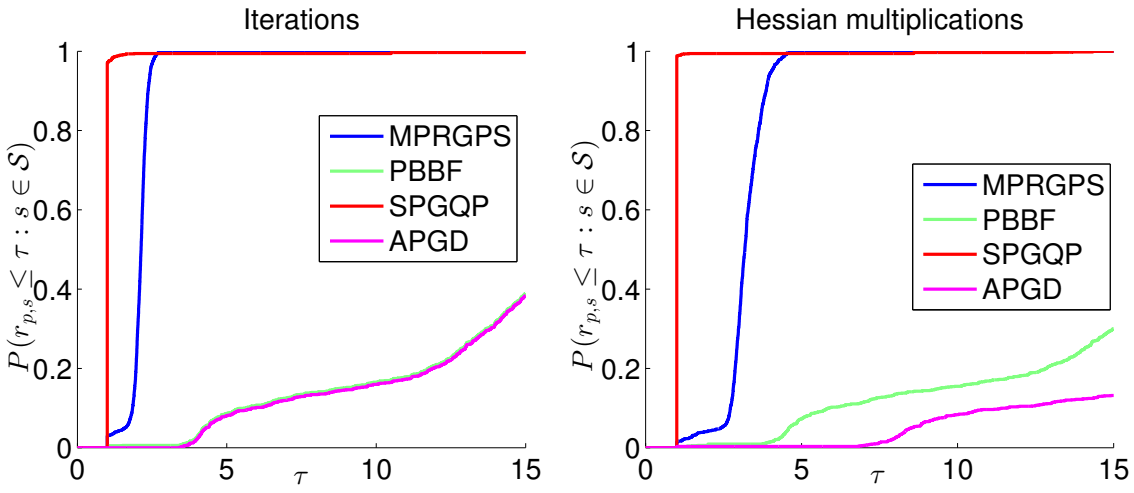


Figure 49: Performance profiles - number of iterations and number of Hessian multiplications.

In the second benchmark, we sprinkle 24435 spherical particles into one fixed obstacle. These particles are divided into tree groups, see Fig. 53, Fig. 54, and Fig. 55. Particles in each group have the different friction coefficient $\mu_1 = 0.1$ (red), $\mu_2 = 0.3$ (blue), and $\mu_3 = 0.5$ (green). The material of the bodies is represented by density $\rho = 2800\ \mathrm{kg} \cdot \mathrm{m}^{-3}$ and the particles have radius $r = 0.01$ m. The stepsize of the time-stepping scheme is $h = 8 \cdot 10^{-4}$ s.

The number of bodies in the system and the number of contacts can be found in Fig. 50. In this case, the dimension of inner minimization problem with separable

conical constraints is three times larger then the number of contacts. In fact, the number of constraints is equal to the number of contacts. Number of performed iterations can be found in Fig. 51 and performance profiles in Fig. 52. In this case, we set the maximum iterations to 10000.

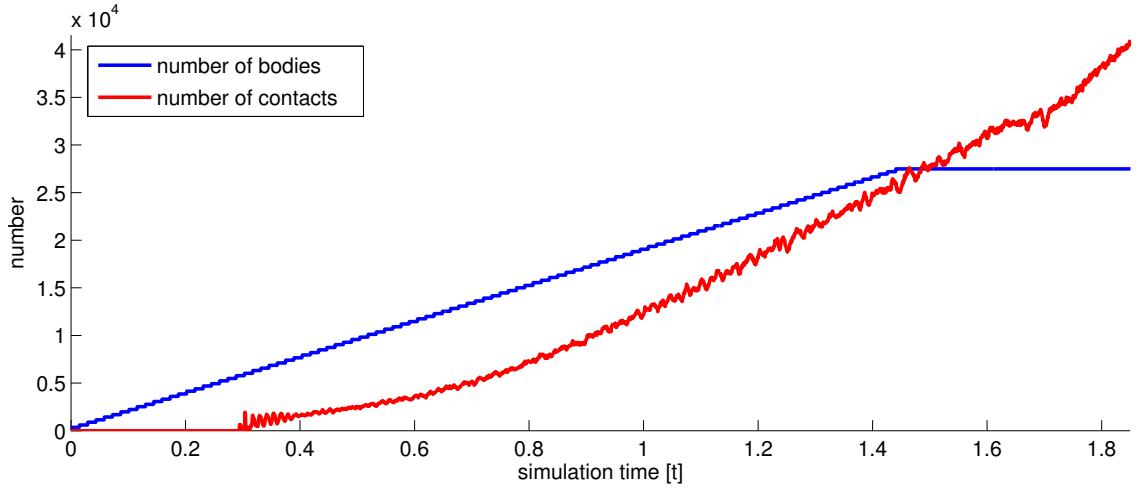


Figure 50: The number of bodies and contacts in the system during the second simulation with friction.

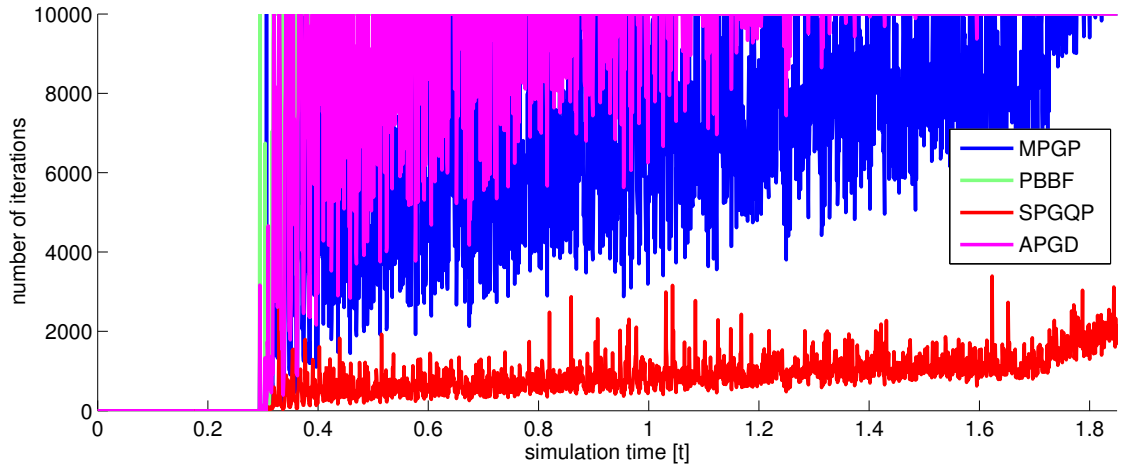


Figure 51: The number of iterations and Hessian multiplications during the second simulation with friction.

The number of iterations depends on the dimension of the problem. In the non-friction case, the constraints of QP problem consist of bound constraints and the

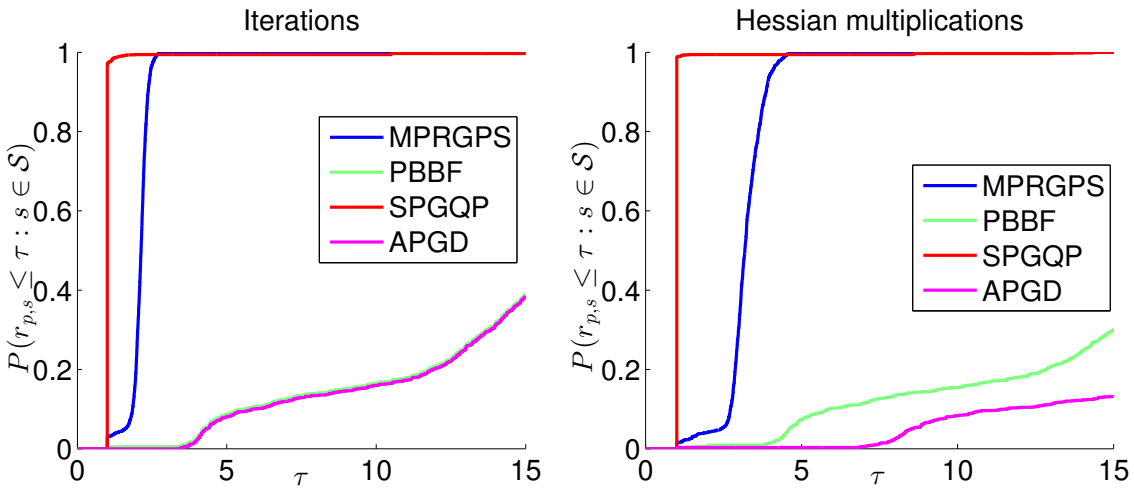


Figure 52: Performance profiles - number of iterations and number of Hessian multiplications.

best algorithms (subject to the number of iterations and the number of Hessian matrix multiplications) are MPRGP and SPG-QP. These algorithms are able to solve all given optimization problems from simulation. In friction case, the active-set algorithm MPGP fails to solve the problem because of the large number of projection steps with constant step-lenght. Furthermore, let us notice, that in APGD algorithm, we are using the constant Lipschitz constant, i.e. the largest eigenvalue. For this estimation, we are using power method. Unfortunatelly, our implementation on GPU is not able to find the largest eigenvalue with high precision, which is important for APGD. Heyn et al. [48] are much more successfull with adaptive/heuristic line-search methods for Lipschitz constant estimation.

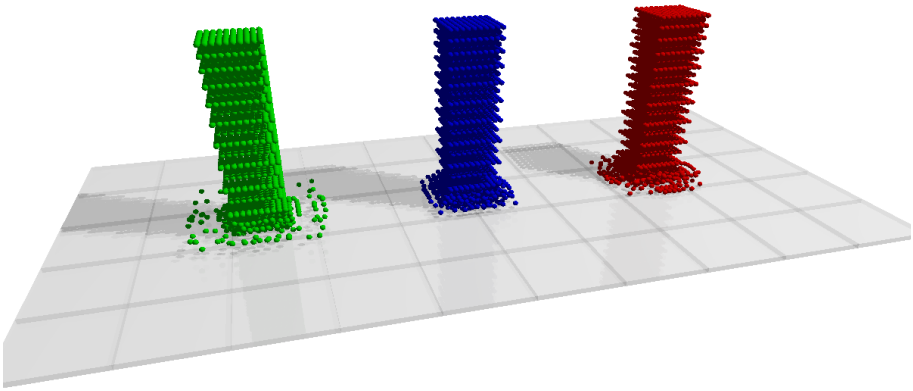


Figure 53: The state of the second benchmark in $t = 0.11$ s.

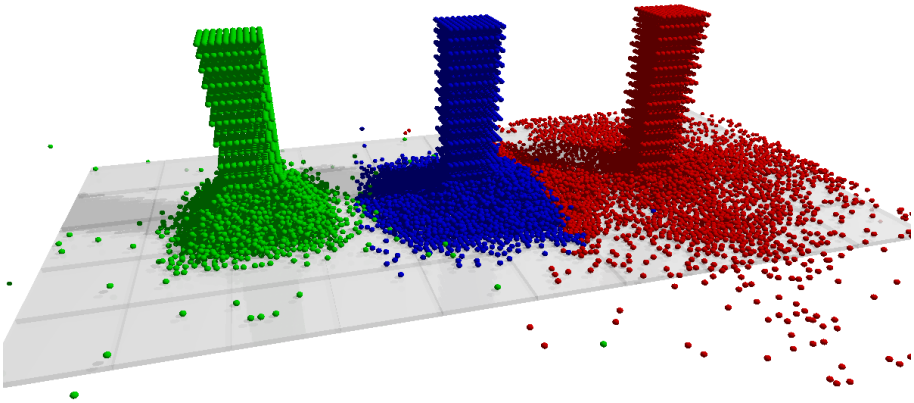


Figure 54: The state of the second benchmark in $t = 0.85$ s.

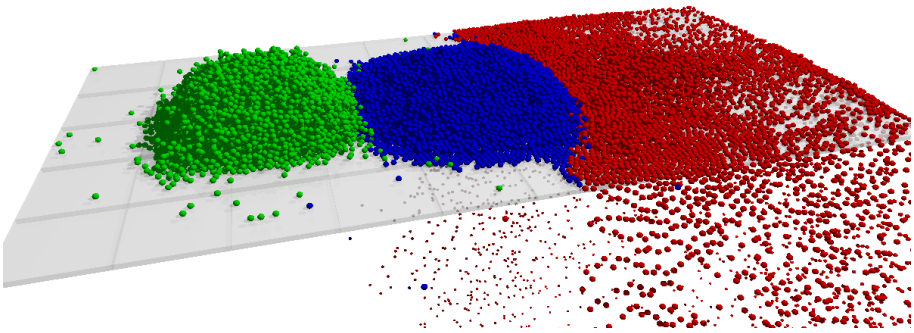


Figure 55: The state of the second benchmark in $t = 1.76$ s.

Conclusions

In the thesis, we have reviewed the fundamental concepts of quadratic programming and we have presented algorithms for solving the minimization problem with convex quadratic function and closed convex feasible set. These optimization problems appear in practical applications. We performed numerical experiments on solving geometrical problems, linear elasticity contact problems, and multi-body dynamics problems. We have implemented algorithms in Matlab environment, C programming language with Cuda toolkit and PETSc toolkit. We have designed practical benchmarks and we have solved these benchmarks on personal computer, GPU card, and Anselm supercomputer.

We have focused on active set methods and projected gradient methods. Our results suggest that each type of presented algorithms has both advantages and disadvantages and the performance depends on the problem properties. If the number of unconstrained components of solution vector is dominating over the constrained part, then the active set methods are able to deal with this part very efficiently. On the other hand, projected gradient methods are effective on the problems with feasible set described by non-linear constraints, such as separable conical constraints. The combination of these two approaches suggests new type of methods.

The properties of the optimization problem are crucial. In quadratic programming, the Hessian matrix, the right-hand side vector, and the constraint functions determine the problem solvability and defines the conditioning of the problem. In the thesis, we have reviewed the basic theory and we have presented our own theoretical results. For instance, we show that the inner optimization problem of multi-body dynamics has always solution in both of non-friction and friction case.

Our results show that the black-box convex programming algorithms can be also used for solving quadratic programs. Furthermore, the efficiency of these methods can be extended using the results from classical quadratic programming algorithms. This issue is the aim of the further development.

Author's bibliography

Journal

- Bouchala J., Dostál Z., Kozubek T., Pospíšil L., Vodstrčil P.: *On the solution of convex QPQC problems with elliptic and other separable constraints with strong curvature*, Applied Mathematics and Computation (IF 1.6), 247:848–864, 2014.

- Dostál Z., Pospíšil L.: *Optimal iterative QP and QPQC algorithms*, Annals of Operations Research (IF 1.103), pages 1–14, 2013.

- Pospíšil L., Dostál Z.: *The projected Barzilai-Borwein method with fall-back for separable strictly convex quadratically constrained quadratic programs*, accepted in Mathematics and Computers in Simulation (IF 0.856), 2014.

- Dostál Z., Pospíšil L.: *Minimization of the quadratic function with semidefinite Hessian subject to the bound constraints*, accepted in Computers and Mathematics with Applications (IF 1.996), 2014.

- Dostál Z., Pospíšil L.: *Fast conjugate gradients for symmetric positive semidefinite least square problems*, submitted to IF, 2014.

Proceedings

- Pospíšil L., Dostál Z.: *Minimization of a convex quadratic function subject to separable conical constraints in granular dynamics.* In J. Chleboun, K. Segeth, J. Šístek, and T. Vejchodský, editors, Proceedings of Programs and Algorithms of Numerical Mathematics 17, Prague IM ASCR, 2014.

- Pospíšil L.: *An optimal algorithm with Barzilai-Borwein steplength and super-relaxation for QPQC problem*, In Chleboun J., Segeth K., Šístek J., Vejchodský T., editors, Proceedings of Programs and Algorithms of Numerical Mathematics 16, pages 155–161, WOS:000317994100024, Prague IM ASCR, 2012.

- Pospíšil L., Dostál Z., Horák D.: *Active-set based quadratic programming algorithm for solving inner optimization problems with inequalities in granular dynamics simulations*, Proceedings of PARTICLES 2015, Barcelona, will be submitted for indexation in the Conference Proceedings Citation Index – ISI Web of Knowledge and in SCOPUS database, accepted, 2015.

Conferences

- Pospíšil L., Dostál Z.: *Active-set based quadratic programming algorithm for solving optimization problems arising in granular dynamics simulations*, ICCCM15, Hannover, 2015.
- Dostál Z., Pospíšil L.: *Minimization of quadratic function with semidefinite Hessian subject to bound constraints*, PMAA 14, Lugano, 2014.
- Pospíšil L.: *Active-set algorithm with spectral steps for semicoercive quadratic problems in particle dynamics*, Wofex 2014, FEECS Dean's award for the results in doctoral studies, 2014.
- Pospíšil L., Dostál Z.: *The solution of quadratic programming problem with separable conical constraints in granular dynamics*, PANM 17, Dolní Maxov, 2014.
- Dostál Z., Pospíšil L.: *The Projected Barzilai-Borwein Method for Solving Quadratic Programming Problems with Separable Elliptic Constraints*, MODELLING 2014, Rožnov pod Radhoštěm, 2014.
- Pospíšil L., Dostál Z., Kozubek T.: *Optimal active-set and spectral algorithms for the solution of 3D contact problems with anisotropic friction*, MAFELAP 2013, London, 2013
- Pospíšil L.: *An optimal algorithm with Barzilai-Borwein steplength and super-relaxation for QPQC problem*, PANM 16, Dolní Maxov, 2012.
- Pospíšil L.: *The Polynomials in Barzilai-Borwein Algorithm*, Wofex 2012, Ostrava, 2012.
- Pospíšil L., Menšík M.: *Faster Gradient Descent Methods*, SNA12, Liberec, 2012.
- Pospíšil L.: *The Improvement of Projected SD for Quadratic Optimization with Separable Elliptic Constraints*, Wofex 2011, Ostrava, 2011.

References

[1] Paraview - an open-source, multi-platform data analysis and visualization application. http://www.paraview.org/.

[2] Pov-ray - the persistence of vision raytracer - a high-quality, free software tool for creating stunning three-dimensional graphics. http://povray.org/.

[3] Projectchrono - a set of open-source tools for mechanical simulations. http://chronoengine.info/.

[4] Cuda programming guide. http://docs.nvidia.com/cuda/index.html, 2015.

[5] J. Alberty, C. Carstensen, and S. A. Funken. Remarks around 50 lline of matlab: short finite element implementation. *Numerical Algorithms*, (20):117–137, 1999.

[6] M. Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming*, 105(1):113–143, 2006.

[7] M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14:231–247, 1997.

[8] M. Anitescu and A. Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, 47(2):207–235, 2010.

[9] B. Avci and P. Wriggers. A dem-fem coupling approach for the direct numerical simulation of 3d particulate flows. *Journal of Applied Mechanics*, 79(1), 2011.

[10] J. Barzilai and J. M. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[11] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[12] D. P. Bertsekas. *Nonlinear Optimization*. Belmont, second edition, April 2004.

[13] E. G. Birgin, J. M. Martínez, and M. M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

[14] E. G. Birgin, J. M. Martínez, and M. M. Raydan. Spg: Software for convex-constrained optimization. *ACM Transactions on Mathematical Software*, 27:340–349, 2001.

[15] J. Bouchala, Z. Dostál, T. Kozubek, L. Pospíšil, and P. Vodstrčil. On the solution of convex qpqc problems with elliptic and other separable constraints with strong curvature. *Applied Mathematics and Computation*, 247:848–864, 2014.

[16] J. Bouchala, Z. Dostál, and P. Vodstrčil. Separable spherical constraints and the decrease of a quadratic function in the gradient projection. *Journal of Optimization Theory and Applications*, 157:132–140, April 2013.

[17] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 1st edition, 2004.

[18] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3rd edition, 2008.

[19] A. L. Cauchy. Compte rendu des s'eances de l'acad'emie des sciences. *Comptes Rendus Hebd. Seances Acad. Sci.*, 21(25):536–538, 1847.

[20] A. R. Conn, N. I. M. Gould, and P. L. Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28:545–572, 1991.

[21] P. A. Cundall. A computer model for simulating progressive large scale movements in blocky rock systems. *Proceedings of the Symposium of the International Society of Rock Mechanics*, 2(8), 1971.

[22] Y. Dai and L. Liao. R-linear convergence of the barzilai and borwein gradient method. *The Institute of Mathematics and its Applications*, 2002.

[23] Y.-H. Dai and R. Fletcher. Projected barzilai-borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100:21–47, 2005.

[24] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[25] Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization*, 7:871–887, 1997.

[26] Z. Dostál. *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, volume 23. SOIA, Springer, New York, US, 2009.

[27] Z. Dostál, A. Friedlander, and S. A. Santos. Augmented lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints. *SIAM Journal on Optimization*, 13:1120–1140, 2003.

[28] Z. Dostál and D. Horák. Scalable feti with optimal dual penalty for a variational inequality. *Numerical Linear Algebra with Applications*, 11:455–472, 2004.

[29] Z. Dostál, D. Horák, R. Kučera, V. Vondrák, J. Haslinger, J. Dobiáš, and S. Pták. Feti based algorithms for contact problems: scalability, large displacements and 3d coulomb friction. *Computer Methods in Applied Mechanics and Engineering*, 194(2–5):395–409, 2005.

[30] Z. Dostál and T. Kozubek. An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. *Mathematical programming, Ser. A*, (135):195–220, 2012.

[31] Z. Dostál, T. Kozubek, T. Brzobohatý, A. Markopoulos, and O. Vlach. Scalable tfeti with optional preconditioning by conjugate projector for transient contact problems of elasticity. *Computer methods in applied mechanics and engineering*, 247–248:37–50, 2012.

[32] Z. Dostál, T. Kozubek, P. Horyl, T. Brzobohatý, and A. Markopoulos. Scalable tfeti algorithm for two dimensional multibody contact problems with friction. *Journal of Computational and Applied Mathematics*, 235:403–418, 2010.

[33] Z. Dostál, T. Kozubek, A. Markopoulos, T. Brzobohatý, V. Vondrák, and P. Horyl. A theoretically supported scalable tfeti algorithm for the solution of multibody 3d contact problems with friction. *Computer methods in applied mechanics and engineering*, 205-208:110–120, 2012.

[34] Z. Dostál and L. Pospíšil. Optimal iterative qp and qpqc algorithms. *Annals of Operations Research*, pages 1–14, 2013.

[35] Z. Dostál and L. Pospíšil. Minimization of the quadratic function with semidefinite hessian subject to the bound constraints. *Computers and Mathematics with Applications*, accepted, 2014.

[36] Z. Dostál and J. Schöberl. Minimizing quadratic functions subject to bound constraints. *Computational Optimization and Applications*, 30(1):23–43, January 2005.

[37] D. Engwirda. Mesh2d - automatic mesh generation. available online on MatlabCentral: http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation.

[38] C. Farhat and F.-X. Roux. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM Journal on Scientific and Statistical Computing*, (1), 1992.

[39] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[40] A. Friedlander and J. M. Martínez. On the maximization of a concave quadratic function with box constraints. *SIAM Journal on Optimization*, pages 177–192, 1994.

[41] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

[42] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[43] V. Hapla. *Massively parallel quadratic programming ssolver with applications in Mechanics*. PhD thesis, VŠB- Technical University of Ostrava, 2015.

[44] J. Haslinger, R. Kučera, and T. Kozubek. Convex programming with separable ellipsoidal constraints: application in contact problems with orthotropic friction. *Lecture Notes in Computational Science and Engineering*, 101:221–242, 2014.

[45] E. J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, volume I. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[46] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[47] T. D. Heyn. *On the Modeling, Simulation, and Visualization of Many-Body Dynamics Problems with Friction and Contact*. PhD thesis, University of Wisconsin–Madison, May 2013.

[48] T. D. Heyn, M. Anitescu, A. Tasora, and D. Negrut. Using krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. *International Journal for Numerical Methods in Engineering*, 95(7):541–561, 2013.

[49] I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek. *Solution of Variational Inequalities in Mechanics*. Springer Verlag, Berlin, 1988.

[50] D. Horák. *FETI based domain decomposition methods for variational inequalities*. PhD thesis, VŠB - Technical University of Ostrava, 2007.

[51] A. J. Laub. *Matrix Analysis For Scientists And Engineers*. Society for Industrial and Applied Mathematics, 2014.

[52] D. Luenberger and Y. G. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science. Springer, third edition, 2008.

[53] Y. Nesterov. *Introductory lectures on Convex Optimization, a Basic Course*. Boston, 2004.

[54] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2003.

[55] V. P. Panagiotopoulos. A non-linear programming approach to the unilateral contact- and friction boundary value problem. *Ingenieur-Archiv*, 44(6):421–432, 1975.

[56] J.-S. Pang and D. E. Stewart. Differential variational inequalities. *Mathematical Programming*, 113(2):345–424, 2008.

[57] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts*. John Wiley & Sons, Inc., 2008.

[58] L. Pospíšil. An optimal algorithm with barzilai-borwein steplength and superrelaxation for qpqc problem. In J. Chleboun, K. Segeth, J. Šístek, and T. Vejchodský, editors, *Proceedings of Programs and Algorithms of Numerical Mathematics 16*, pages 155–161, Prague, 2012. IM ASCR.

[59] L. Pospíšil and Z. Dostál. Minimization of a convex quadratic function subject to separable conical constraints in granular dynamics. In J. Chleboun, K. Segeth, J. Šístek, and T. Vejchodský, editors, *Proceedings of Programs and Algorithms of Numerical Mathematics 17*, Prague, 2014. IM ASCR.

[60] L. Pospíšil and Z. Dostál. The projected barzilai-borwein method with fall-back for separable strictly convex quadratically constrained quadratic programs. *Mathematics and Computers in Simulation*, accepted, 2015.

[61] M. M. Raydan. *Convergence properties of the Barzilai and Borwein gradient method*. PhD thesis, Rice University, 1991.

[62] M. Renouf and P. Alart. Conjugate gradient type algorithms for frictional multi-contact problems: applications to granular materials. *Computer Methods in Applied Mechanics and Engineering*, 194(18–20):2019–2041, 2005.

[63] M. Sadowská, Z. Dostál, T. Kozubek, J. Bouchala, and A. Markopoulos. Scalable total beti based solver for 3d multibody frictionless contact problems in mechanical engineering. *Engineering Analysis with Boundary Elements*, 35:330–341, 2011.

[64] A. Schinner. Fast algorithms for the simulation of polygonal particles. *Granular Matter*, 2:35–43, 1999.

[65] J. Schöberl. Solving the signorini problem on the basis of domain decomposition techniques. *Computing,*, 60:323–344, 1998.

[66] S. Schönherr. *Quadratic Programming in Geometric Optimization: Theory, Implementation, and Applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 2002.

[67] A. A. Shabana. *Computational Dynamics*. John Wiley & Sons, Inc., third edition, 2010.

[68] V. Vondrák, T. Kozubek, and Z. Dostál. Parallel solution of contact shape optimization problems based on total feti domain decomposition method. *Structural and Multidisciplinary Optimization*, 42:955–964, 2010.