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

Minimizing quadratic functions with separable quadratic constraints

master thesis

2010 Lukáš Pospíšil

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

Ostrava, May 5, 2010 ....

Rád bych na tomto místě poděkoval především Prof. RNDr. Zdeňku Dostálovi, DrSc. za pomoc a vynikající motivující vedení mé práce, zejména za myšlenku nového algoritmu. Poděkování si zaslouží i Doc. RNDr. Radek Kučera, Ph.D. za poskytnutí zdrojových kódů "konkurenčního" algoritmu.

Dále děkuji mému bratru Mgr. Michalovi Pospíšilovi. Jeho pomoc s anglickým jazykem dala většině vět v této práci smysl.

Za morální a finanční pomoc a podporu děkuji přátelům a rodině, zejména mamince - díky ní jsem, vím a chci vědět.

Speciální poděkování patří mé milované Veronice za nový smysl.

Abstract

This thesis deals with the application of Dual problem in quadratic programming and introduces algorithms for solving minimizing problem of quadratic function subject to set prescribed by quadratic constraint functions. We proceed from simple observations to a new algorithm which was never presented before. Quadratic constraints are characteristic for contact problems with Coulomb friction.

Keywords: Dual problem, Inverse dual problem, quadratic function, PDP

Abstrakt

Tato práce popisuje využití Duální úlohy v kvadratickém programování a představuje algoritmy pro minimalizaci kvadratické funkce vzhledem k množině popsané vazebními kvadratickými funkcemi. Od pozorování jednoduchých algoritmů přechází k algoritmu novému, který zatím nebyl nikde publikován. Kvadratické vazby jsou charakteristické pro kontaktní úlohy s Coulombovským třením.

Klíčová slova: Duální úloha, Inverzní duální úloha, kvadratická funkce, PDP

Contents

1 Introduction

In my master thesis I try to show application of Dual problem in minimizing quadratic functions with separable quadratic constraints solutions. This problem arises in problems with Coulomb friction. The motivation example is presented in Chapter 2.

Formulation of minimizing problem can be found in Chapter 3. In this chapter are given also graphs of quadratic function and quadratic constraint set of one constraint problem.

I introduce Lagrange function in Chapter 4 and also its utilization in analytical solution of a simle two dimensional problem. From presented example one can see the point of using KKT conditions. In this chapter, I introduce two numerical algorithms used later - Conjugate gradient method and Modified proportioning with reduced gradient projections. These algorithms are introduced without detail analysis. Further implementation can be found in Appendix.

In Chapter 5, I examine KKT conditions for more dimensional problems. Using simple modifications we can infer Dual problem and Inverse dual problem - two key components of new algorithm.

In Chapter 6, I try to find meaning of Lagrange multiplier. Due to my observations, it can be regarded as linear penalty, increasing of which we can attract aproximations to boundary of constraint set.

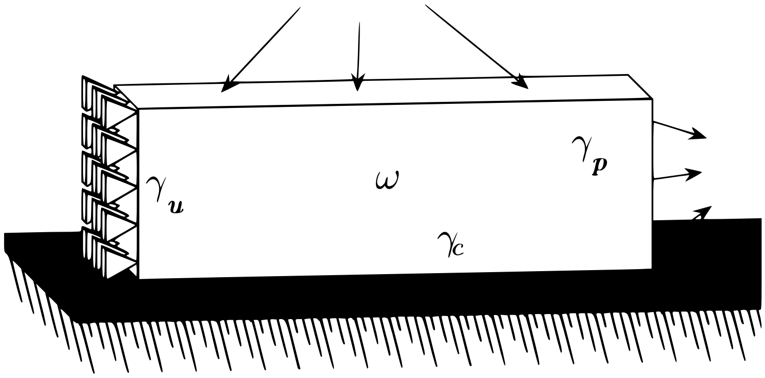
Simple algorithms, which use first KKT condition and linear update, are introduced in Chapter 7. Their convergence depends on the choice of input data. These algorithms are helpful in costruction of a main algorithm.

Finally, I used all previous observations to introduce the new pretentious algorithm in Chapter 8 - Projected Dual Problem method (PDP). This algorithm uses both of KKT conditions and update Lagrange multipliers in the best way - it uses projection to boundary of quadratic constraint set. Numerical tests are also presented.

2 Motivation

We start with motivation example. This problem consists of solving minimizing problem of quadratic function with linear inequalities and quadratic inequality constraints. But in this thesis, I try to solve simpler problem only with quadratic constraints.

Example 2.0.1
Let us consider the steel brick lying on a rigid foundation as it is shown in figure.[1]



The brick occupies in the reference configuration the domain $\omega \subset \mathbb{R}^3$, whose boundary $\partial\omega$ is split into three nonempty disjoint parts γ_u, γ_p, and γ_c with different boundary conditions: zero displacements γ_u, surface tractions γ_p and contact conditions γ_c (i.e., the nonpenetration and the effect of friction).

The elastic behavior of the brick is described by Lamé equations that, after finite element discretization, lead to a symmetric positive definite stiffness matrix $K \in \mathbb{R}^{3n_c \times 3n_c}$ and to a load vector $f \in \mathbb{R}^{3n_c}$. Moreover, we introduce full rank matrices $N, T1, T2 \in \mathbb{R}^{m_c \times 3n_c}$ projecting displacements at contact nodes to normal and tangential directions, respectively, and we denote $B = \left(N^T, T_1^T, T_2^T\right)^T \in \mathbb{R}^{3n_c \times 3n_c}$. Here, we shall use the dual formulation in terms of contact stresses.
We start with the contact problem with *Tresca friction* that reads as

$$\begin{cases} \text{minimize} & \frac{1}{2}\lambda^T Q \lambda - \lambda^T h, \\ \text{subject to} & \lambda_{\nu,i} \geq 0, \lambda_{t_1,i}^2 + \lambda_{t_2,i}^2 \leq r_i^2, i = 1, \ldots, m_c \\ & \lambda = \left(\lambda_\nu^T, \lambda_{t_1}^T, \lambda_{t_2}^T\right)^T, \lambda_\nu, \lambda_{t_1}, \lambda_{t_2} \in \mathbb{R}^{m_c}, \end{cases}$$

where $Q = BK^{-1}B^T, h = BK^{-1}f$, and $r_i \geq 0$ are given slip bound values at contact nodes. Let us point out that λ_ν and λ_{t_1}, λ_{t_2} represent normal and tangential contact stresses, respectively.

■

[1]This example was introduced and numericaly solved in [5]. More details about model problem can be found in [6].

3 Problem definition

In this chapter I formulate minimizing problem and show how quadratic function and set prescribed by quadratic function looks.

3.1 Quadratic function

Definition 3.1.1 (Quadratic function definition)
The quadratic function has prescription

$$f(x) \overset{\text{def}}{=} \frac{1}{2}x^T Ax - b^T x \tag{1}$$

where

- $n \in \mathbb{N}$ is problem dimension
- $f : \mathbb{R}^{2n} \to \mathbb{R}$
- $A \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}$ is symetric positive definite matrix
- $b \in \mathbb{R}^{2n}$ is vector of *right sides*

Theorem 3.1.1 (Quadratic function gradient)
Gradient of function defined by equation (1) is

$$\triangledown f = Ax - b$$

Remark: Minimum of 3.1.1 without constraints is equal to solution of system $\triangledown f = o$, respectively $Ax = b$. That is the reason, why we called b the vector of *right sides*. $\approx$

Proof: Let us consider improvement $x + \alpha v$ of point x, where $x, v \in \mathbb{R}^n, \alpha \in \mathbb{R}$
Then

$$f(x + \alpha v) - f(x) = \left(\frac{1}{2}(x + \alpha v)^T A(x + \alpha v) - b^T(x + \alpha v)\right) - \left(\frac{1}{2}x^T Ax - b^T x\right) =$$

$$= \alpha x^T Av - \alpha b^T v + \frac{1}{2}\alpha^2 v^T Av = \frac{1}{2}\alpha^2 v^T Av + \alpha(Ax - b)^T v$$

- $\bar{x} = \min f(x) \Rightarrow A\bar{x} = b$
  Necessary condition of $\min f(x)$ is $\bigtriangledown f = o$
  $\bigtriangledown f(x) = Ax - b \Rightarrow Ax - b = o \Rightarrow Ax = b$

- $\bar{x} = \min f(x) \Leftarrow A\bar{x} = b$
  $A\bar{x} = b \Rightarrow A\bar{x} - b = o$
  $f(\bar{x} + \alpha v) - f(\bar{x}) = \frac{1}{2}\alpha^2 v^T A v \geq 0, \forall \alpha \in \mathbb{R} \forall v \in \mathbb{R}^n$
  (A is positive definite)
  $\Rightarrow f(\bar{x} + \alpha v) \geq f(\bar{x}), \forall \alpha \in \mathbb{R} \forall v \in \mathbb{R}^n$

□

Theorem 3.1.2 (Existence of minimum)

Function $f(x)$ given by equation (1) has one minimum.
System $\bigtriangledown f = o$ has only one solution.

3.2 Separated quadratic constraints

Definition 3.2.1 (Constraint function)

Let us define n quadratic constraint functions

$$g_i(x) \stackrel{\text{def}}{=} x_{2i-1}^2 + x_{2i}^2 - r_i^2, i = 1, 2, \ldots n \tag{2}$$

where

- $n \in \mathbb{N}$ is number of constraint functions,
- $g_i : \mathbb{R}^{2n} \to \mathbb{R}$,
- $r \in \mathbb{R}^n$ is vector of *radii*.

If we choose firm $g(x) = 0$ then the geometric representation of set described by quadratic function is circle with radius r.

Definition 3.2.2 (Constraint set)
Quadratic constraint functions define *equality* constraint set

$$\Omega_E \stackrel{\text{def}}{=} \{x \in \mathbb{R}^{2n} : g_i(x) = 0, i = 1, 2, \ldots n\}. \tag{3}$$

We can also define *inequality* constraint set

$$\Omega_I \stackrel{\text{def}}{=} \{x \in \mathbb{R}^{2n} : g_i(x) \leq 0, i = 1, 2, \ldots n\}. \tag{4}$$

3.3 Minimizing function subject to constraint set

Definition 3.3.1 (Minimizing problem)
Unconstrained problem: Find

$$\bar{x} \stackrel{def}{=} \min_{x \in \mathbb{R}^{2n}} f(x) \tag{5}$$

Equality problem: Find

$$\bar{x}_E \stackrel{def}{=} \min_{x \in \Omega_E} f(x) \tag{6}$$

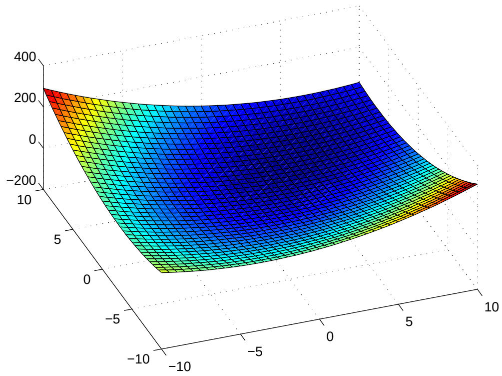
Inequality problem: Find

$$\bar{x}_I \stackrel{def}{=} \min_{x \in \Omega_I} f(x) \tag{7}$$

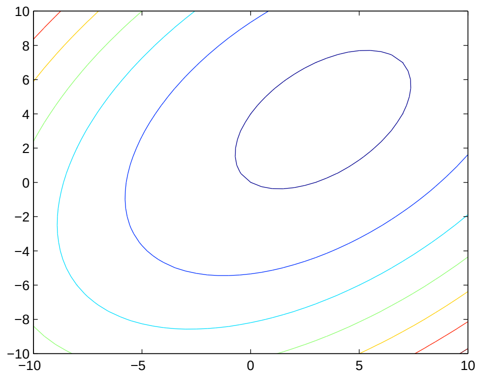
Example 3.3.1
Find solution of Equality problem defined by equation 6 with

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, r = 1$$

Geometrically, quadratic function $f(x)$ is a modified elliptic paraboloid.

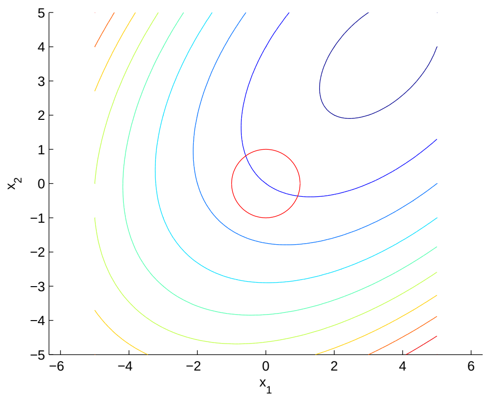


Isolines of this function (curves with same function value) are depicted in the following figure.



The geometric representation of equality constraint set Ω_E of (6) is a circle with centre at $[0, 0]$ and radius $r = 1$.

If we combine isolines and constraint we can estimate the probable location of minimum, as plotted in the next figure:



■

4 Behind the new algorithm

4.1 Lagrange function

Definition 4.1.1 (Lagrange function)
Lagrange function has prescription

$$L(x, \lambda) \stackrel{\text{def}}{=} f(x) + \lambda g(x)$$

where

- $f(x) : \mathbb{R}^n \to \mathbb{R}$ is *cost function*
- $g(x) : \mathbb{R}^n \to \mathbb{R}$ is *constraint function*
- $\lambda \in \mathbb{R}$ is *Lagrange multiplier*

Theorem 4.1.1 (bounded local extremes subject to equality constraint set)
Let

- $f, g : \mathbb{R}^n \to \mathbb{R}$ be C^1 in open set $\Omega \subset \mathbb{R}^n, n > 1$
- grad $g(x) \neq (0, \dots, 0)$ for each $x \in \Omega$
- $\Omega_E \stackrel{\text{def}}{=} \{x \in \Omega : g(x) = 0\}$.

Then

1. (Necessary condition of existence of local bounded extreme)
If f has in $c \in \Omega$ local extreme subject to set Ω_E, there exists $\lambda \in \mathbb{R}$ such that c is a stationary point of $L(x) = f(x) + \lambda g(x), x \in \Omega$.
2. (Sufficient condition of existence of local bounded extreme)
Let $c \in \Omega$ be a stationary point of function $L(x) = f(x) + \lambda g(x)$ for some $\lambda \in \mathbb{R}$,
let f and g have in c continuous second partial derivatives
and let $d^2 L_c$ (for given λ) be positive definite quadratic form.
Then f has in c local minimum subject to Ω_E.

Theorem 4.1.2 (Sufficient condition subject to inequality constraint set)
Let f, g, L be same functions as in Definition 4.1.1.
Let $\Omega_I \overset{\text{def}}{=} \{x \in \Omega : g(x) \leq 0\}$.
If

- $c \in \Omega$ is a stationary point of function $L(x) = f(x) + \lambda g(x)$ for some $\lambda \geq 0$,
- f and g have in c continuous second partial derivatives,
- $d^2 L_c$ (for given λ) is positive definite quadratic form

then f has in c local minimum subject to Ω_I.

These theorems (i.e., sufficient conditions) gives us manual how to find bounded local extremes subject to equality and inequality constraint set.

Definition 4.1.2 (Lagrange function for more constraints)
Lagrange function for problems with m constraints has prescription

$$L(x, \lambda) \overset{\text{def}}{=} f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

where

- $f(x) : \mathbb{R}^n \to \mathbb{R}$ is *cost function*
- $m \geq 1$ is number of constraints
- $g_i(x) : \mathbb{R}^n \to \mathbb{R}$ is one of m *constraint functions*
- $\lambda \in \mathbb{R}^m$ is vector of *Lagrange multipliers*

4.2 Analytical solution

In analytical solution we will proceed with standart "bounded local extremes" search algorithm. At first we consider Equality problem (see Definition 3.3.1). We assume a simple problem with one quadratic constraint.

Assume Lagrange function for one condition

$$L(x,\lambda) = f(x) + \lambda . g(x).$$

For saddle point of this Lagrange function applies

$$\min_{x\in\Omega} f(x) = \min_{x\in\mathbb{R}^2} L(x,\lambda) = \bar{x}.$$

In saddle point also holds constraint condition

$$g(\bar{x}) = 0$$

Derivative of Lagrange function in saddle point has zero value (it is stacionary point of Lagrange function), so our task is to compute derivative of $L(x,\lambda)$ and set it equal to zero. Since Lagrange function is a function of two variables, we have to compute partial derivatives and solve system of two equations.

$$I.)\ \ \nabla_x L(x,\lambda) = o$$

$$II.)\ \ \nabla_\lambda L(x,\lambda) = o$$

These conditions are also called "Karush-Kuhr-Tucker conditions" (alias "KKT system", see Chapter 5).

So

$$I.)\ \ \nabla_x L(x,\lambda) = \nabla_x f(x) + \lambda \nabla_x g(x) = Ax - b + 2\lambda x$$
$$II.)\ \ \nabla_\lambda L(x,\lambda) = \nabla_\lambda f(x) + \nabla_\lambda \lambda g(x) = g(x)$$

and derived KKT system is

$$I.)\ \ Ax - b + 2\lambda x = o \tag{8}$$

$$II.)\ \ g(x) = 0 \tag{9}$$

For Inequality problem, we simply modify second condition

$$II.)\ \ g(x) \leq 0$$

Example 4.2.1
Consider Inequality problem with input data

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, r = 1.$$

So our problem is to find

$$\bar{x} \overset{def}{=} \min_{x \in \Omega_I} f(x)$$

where

$$\Omega_I \overset{\text{def}}{=} \{x \in \mathbb{R}^2 : g(x) \leq 0\}$$

and one constraint is defined

$$g(x) \overset{\text{def}}{=} x_1^2 + x_2^2 - r^2$$

Left-hand side of first KKT equation (8) (we consider λ as parameter $\lambda \in \mathbb{R}, \lambda \geq 0$) has the form

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 2\lambda x_1 \\ 2\lambda x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 - 3 + 2\lambda x_1 \\ -x_1 + 2x_2 - 4 + 2\lambda x_2 \end{bmatrix}$$

Thus (8) is transformed to the next system

$$\begin{aligned} (2+2\lambda)x_1 - x_2 &= 3 \\ -x_1 + (2+2\lambda)x_2 &= 4. \end{aligned}$$

Using Kramer formulas we obtain

$$D = \begin{vmatrix} 2+2\lambda & -1 \\ -1 & 2+2\lambda \end{vmatrix} = (2+2\lambda)^2 - (-1)^2 = 3 + 8\lambda + 4\lambda^2$$

$$D_1 = \begin{vmatrix} 3 & -1 \\ 4 & 2+2\lambda \end{vmatrix} = 3(2+2\lambda) - (-1).4 = 10 + 6\lambda$$

$$D_2 = \begin{vmatrix} 2+2\lambda & 3 \\ -1 & 4 \end{vmatrix} = 4.(2+2\lambda) - 3.(-1) = 11 + 8\lambda$$

and parametric solution is (for common case refer to Dual task in Chapter 5)

$$x_1 = \frac{D_1}{D} = \frac{10+6\lambda}{3+8\lambda+4\lambda^2}$$

$$x_2 = \frac{D_2}{D} = \frac{11+8\lambda}{3+8\lambda+4\lambda^2}$$

Now consider the constraint function

$$g(x) = x_1^2 + x_2^2 - 1 = \left(\frac{10+6\lambda}{3+8\lambda+4\lambda^2}\right)^2 + \left(\frac{11+8\lambda}{3+8\lambda+4\lambda^2}\right)^2 - 1$$

and put it equal to zero

$$\left(\frac{10+6\lambda}{3+8\lambda+4\lambda^2}\right)^2 + \left(\frac{11+8\lambda}{3+8\lambda+4\lambda^2}\right)^2 = 1$$

$$\frac{(10+6\lambda)^2 + (11+8\lambda)^2}{(3+8\lambda+4\lambda^2)^2} = 1$$

$$(10 + 6\lambda)^2 + (11 + 8\lambda)^2 = (3 + 8\lambda + 4\lambda^2)^2$$
$$(100 + 120\lambda + 36\lambda^2) + (121 + 176\lambda + 64\lambda^2) = (3 + 8\lambda + 4\lambda^2)(3 + 8\lambda + 4\lambda^2)$$
$$221 + 296\lambda + 100\lambda^2 = 9 + 48\lambda + 88\lambda^2 + 64\lambda^3 + 16\lambda^4$$
$$0 = 16\lambda^4 + 64\lambda^3 - 12\lambda^2 - 248\lambda - 212$$

This polynom has 4 roots - two of them are real, one is positive. Value of it is approximately

$$\lambda = 1.9877,$$

so the solution after substitution is

$$\bar{x} = \begin{bmatrix} 0.6318 \\ 0.7751 \end{bmatrix}.$$

■

Example 4.2.2
Consider Inequality problem with input data

$$A = \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, r = 1.$$

So our problem is to find

$$\bar{x} \overset{def}{=} \min_{x \in \Omega_I} f(x)$$

where

$$\Omega_I \overset{\text{def}}{=} \{x \in \mathbb{R}^2 : g(x) \leq 0\}$$

and one constraint is defined

$$g(x) \overset{\text{def}}{=} x_1^2 + x_2^2 - r^2$$

First, we refer (8):

$$\begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 2\lambda x_1 \\ 2\lambda x_2 \end{bmatrix} = \begin{bmatrix} 4x_1 - x_2 - 1 + 2\lambda x_1 \\ -x_1 + 2x_2 - 1 + 2\lambda x_2 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Now solve system

$$\begin{aligned} (4 + 2\lambda)x_1 - x_2 = 1 \\ -x_1 + (2 + 2\lambda)x_2 = 1 \end{aligned}$$

using Kramer formulas

$$D = \begin{vmatrix} 4 + 2\lambda & -1 \\ -1 & 2 + 2\lambda \end{vmatrix} = (4 + 2\lambda)(2 + 2\lambda) - (-1)^2 = 7 + 12\lambda + 4\lambda^2$$

$$D_1 = \begin{vmatrix} 1 & -1 \\ 1 & 2 + 2\lambda \end{vmatrix} = (2 + 2\lambda) - (-1) = 3 + 2\lambda$$

$$D_2 = \begin{vmatrix} 4 + 2\lambda & 1 \\ -1 & 1 \end{vmatrix} = (4 + 2\lambda) - (-1) = 5 + 2.\lambda$$

Parametric solution is

$$x_1 = \frac{D_1}{D} = \frac{3+2\lambda}{7+12\lambda+4\lambda^2}$$
$$x_2 = \frac{D_2}{D} = \frac{5+2\lambda}{7+12\lambda+4\lambda^2}.$$

Constraint function

$$g(x) = x_1^2 + x_2^2 - 1 = \left(\frac{3+2\lambda}{7+12\lambda+4\lambda^2}\right)^2 + \left(\frac{5+2\lambda}{7+12\lambda+4\lambda^2}\right)^2 - 1$$

is set to zero and solved

$$(3+2\lambda)^2 + (5+2\lambda)^2 = (7+12\lambda+4\lambda^2)^2$$
$$(9+12\lambda+4\lambda^2) + (25+20\lambda+4\lambda^2) = 16\lambda^4 + 96\lambda^3 + 200\lambda^2 + 168\lambda + 49$$
$$0 = 16\lambda^4 + 96\lambda^3 + 192\lambda^2 + 136\lambda + 15$$

This polynom has two real roots, but all of them are negative, because minimum of original problem naturaly satisfies quadratic inequality constraint.
Thus we search for $\lambda \geq 0$ and because founded $\lambda < 0$, we simply choose $\lambda = 0$ and get

$$L(x,0) = f(x) + 0.g(x) = f(x) \;\Rightarrow\; \min L(x,\lambda) = \min f(x)$$

We can find minimum of this Inequality problem using simple minimalization algorithm without constraints.

$$\bar{x} = \min_{x\in\Omega_I} f(x) = \min_{x\in\mathbb{R}^2} f(x) \Leftrightarrow \nabla_x f(\bar{x}) = 0$$

We solve equation

$$A\bar{x} - b = o$$
$$A\bar{x} = b$$

using Gauss-Jordan elimination method we have

$$\left[\begin{array}{cc|c} 4 & -1 & 1 \\ -1 & 2 & 1 \end{array}\right] \sim \left[\begin{array}{cc|c} 4 & -1 & 1 \\ -4 & 8 & 4 \end{array}\right] \sim \left[\begin{array}{cc|c} 4 & -1 & 1 \\ 0 & 7 & 5 \end{array}\right] \sim \left[\begin{array}{cc|c} 28 & -7 & 7 \\ 0 & 7 & 5 \end{array}\right] \sim$$
$$\sim \left[\begin{array}{cc|c} 28 & 0 & 12 \\ 0 & 7 & 5 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 0 & \frac{3}{7} \\ 0 & 1 & \frac{5}{7} \end{array}\right]$$

We obtain the solution

$$\bar{x} = \frac{1}{7}\left[\begin{array}{c} 3 \\ 5 \end{array}\right]$$

which really satisfies constraint

$$g(\bar{x}) = \bar{x}_1^2 + \bar{x}_2^2 - r^2 = \left(\frac{3}{7}\right)^2 + \left(\frac{5}{7}\right)^2 - 1 = \frac{9+25-49}{49} = -\frac{15}{49} < 0 \Rightarrow \bar{x} \in \Omega_I$$

■

4.3 Conjugate gradient method

The Conjugate gradient method is iterative method for solving system

$$Ax = b$$

where A is symmetric positive definite matrix and b is the vector of right sides.

Remark: The CG method is also used to find minimum of quadratic function with SPD matrix. (see Remark after Theorem 3.1.1) $\approx$

More information about Conjugate Gradient method can be found in [1].

4.4 MPRGP

Modified proportioning with reduced gradient projections (MPRGP) is iterative method for minimizing quadratic cost function

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

subject to linear inequalities $l \in \mathbb{R}^m$

$$\forall i = 1, \dots m : x_i \geq l_i$$

More information about MPRGP can be found in [2].

5 KKT system and dual problem

During analytical solution of minimizing problem in Section 4.2 we deduce that in minimum of Lagrange function are accomplished two equations implied from partial derivatives of this function. In this section we try to generalize these equations and then we make some observations which we use into Dual problem definition.

In whole section we consider minimizing problem with quadratic function $f : \mathbb{R}^{2n} \to \mathbb{R}$ and n quadratic constraints which bind together succesively pairs of components of vector of variables x.

5.1 KKT system

5.1.1 Minimum of Lagrange function

We consider Lagrange function (see Definition 4.1.1)

$$L(x,\lambda) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x)$$
$$\lambda \in \mathbb{R}^n, L : \mathbb{R}^{2n+n} \to \mathbb{R}$$

and express KKT conditions in saddle point of $L(x,\lambda)$ using Theorem **??**

$$\bigtriangledown_x L(x,\lambda) = \bigtriangledown f(x) + \sum_{i=1}^{n} \lambda_i \bigtriangledown g_i(x) = o_{2n} \tag{10}$$

$$\bigtriangledown_\lambda L(x,\lambda) = g(x) = o_n \tag{11}$$

Remark: o_n denote zero vector of n components. $\approx$

5.1.2 Duplication of Lagrange multipliers

Now consider first KKT condition (10)

$$\bigtriangledown f(x) + \sum_{i=1}^{n} \lambda_i \bigtriangledown g_i(x) = o_{2n}.$$

At first we express gradient of quadratic function

$$\bigtriangledown f(x) = Ax - b \tag{12}$$

and gradient of separable quadratic constraints

$$\bigtriangledown g_i(x) = \begin{pmatrix} \frac{\partial g_i}{\partial x_1} \\ \vdots \\ \frac{\partial g_i}{\partial x_{2i-1}} \\ \frac{\partial g_i}{\partial x_{2i}} \\ \vdots \\ \frac{\partial g_i}{\partial x_{2n}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 2x_{2i-1} \\ 2x_{2i} \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{2n} \tag{13}$$

Then we substitute (12) and (13) into (10). We obtain

$$\triangledown f(x) + \sum_{i=1}^{n} \lambda_i \triangledown g_i(x) = Ax - b + \sum_{i=1}^{n} \left(\lambda_i (0, \dots, 2x_{2i-1}, 2x_{2i}, \dots, 0)^T\right) =$$

$$= Ax - b + 2\left(\sum_{i=1}^{2n} \lambda_{\lceil \frac{i}{2} \rceil} x_i\right) = Ax - b + 2\mathrm{diag}(\tilde{\lambda})x$$

where

$$\tilde{\lambda} \stackrel{\mathrm{def}}{=} (\lambda_1, \lambda_1, \lambda_2, \lambda_2, \dots \lambda_n, \lambda_n)^T \in \mathbb{R}^{2n}. \tag{14}$$

Hence

$$Ax - b + 2\ \mathrm{diag}(\tilde{\lambda})x = o_{2n}. \tag{15}$$

5.2 Dual problem

Let us assume modified first KKT condition (15) and express variable x:

$$Ax - b + 2\ \mathrm{diag}(\tilde{\lambda})x = o_{2n}$$

$$Ax + 2\ \mathrm{diag}(\tilde{\lambda})x = b$$

$$(A + 2\ \mathrm{diag}(\tilde{\lambda}))x = b$$

$$x = (A + 2\ \mathrm{diag}(\tilde{\lambda}))^{-1} b \tag{16}$$

We call equation (16) Dual problem. It represents relation between variable x and corresponding Lagrange multipliers λ (supposing the first KKT condition to be accomplished). If we have λ, we can simply solve (16) with Conjugate gradient method (see Section 4.3) to get solution x.

Definition 5.2.1 (Dual problem solution)

We say that a pair (x, λ) solve *Dual problem*, if equation

$$x = (A + 2\ \mathrm{diag}(\dots, \lambda_i, \lambda_i, \dots))^{-1} b \tag{17}$$

is fulfilled.

5.3 Inverse dual problem

5.3.1 Inverse problem

Now we consider situation, when we have approximation x and our task is to find corresponding Lagrange multipliers λ that (x, λ) solve Dual problem (17) *as good as possible*.

Remark: In Dual problem dimension of vector of Langrange multipliers λ is half of dimension of variable x. That is reason why not for all $x \in \mathbb{R}^{2n}$ exists corresponding $\tilde{\lambda}$. $\approx$

From λ we require that $\tilde{\lambda}$ given by (14) satisfies (16) *as good as possible*:

- $\forall i = 1, \dots n : \tilde{\lambda}_{2i-1} = \tilde{\lambda}_{2i} = \lambda_i$
- equation (17) from Dual problem is accomplished *as good as possible*

5.3.2 Error function

At first we express $\tilde{\lambda}$ from Dual problem (16)

$$Ax + 2.\, \mathrm{diag}(\tilde{\lambda})x = b$$

$$2.\, \mathrm{diag}(\tilde{\lambda})x = (b - Ax)$$

$$2.\mathrm{diag}(x)\tilde{\lambda} = (b - Ax) \tag{18}$$

From equation (18) we can derive error function, which describes distance of approximate solution $\tilde{\lambda}$ to exact solution of dual problem equation (16).

$$\mathrm{err} \stackrel{\mathrm{def}}{=} 2.\mathrm{diag}(x)\tilde{\lambda} - (b - Ax) \tag{19}$$

Our aim is to have err *as small as possible*

$$\|\mathrm{err}\|^2 = \mathrm{err}^T\mathrm{err}$$

Substitute and compose

$$\mathrm{err}^T\mathrm{err} = \Big(2.\mathrm{diag}(x)\tilde{\lambda} - (b - Ax)\Big)^T \Big(2.\mathrm{diag}(x)\tilde{\lambda} - (b - Ax)\Big) =$$

$$= 4.\tilde{\lambda}^T \mathrm{diag}(x)^2\tilde{\lambda} - 4.\tilde{\lambda}^T \mathrm{diag}(x)(b - Ax) + (b - Ax)^T(b - Ax) =$$

$$= 4.\sum_{i=1}^{2n} \Big(\tilde{\lambda}_i^2 x_i^2\Big) - 4.\sum_{i=1}^{2n} \Big(\tilde{\lambda}_i x_i [b - Ax]_i\Big) + (b - Ax)^T(b - Ax) \tag{20}$$

But we know $\tilde{\lambda}_{2i-1} = \tilde{\lambda}_{2i} = \lambda_i$, so we can write (20) as follows

$$4.\sum_{i=1}^{n} \big(\lambda_i^2(x_{2i-1}^2 + x_{2i}^2)\big) - 4.\sum_{i=1}^{n} (\lambda_i(x_{2i-1}[b - Ax]_{2i-1} + x_{2i}[b - Ax]_{2i})) + (b - Ax)^T(b - Ax) =$$

$$= 4.\lambda^T \mathrm{diag}(\dots, x_{2i-1}^2 + x_{2i}^2, \dots)\lambda - 4.(\dots, x_{2i-1}[b - Ax]_{2i-1} + x_{2i}[b - Ax]_{2i}, \dots)\lambda + (b - Ax)^T(b - Ax)$$

5.3.3 Final simplification

If we denote

$$Q \stackrel{\text{def}}{=} 8.\ \mathrm{diag}(\dots, x_{2i-1}^2 + x_{2i}^2, \dots) \tag{21}$$

$$q \stackrel{\text{def}}{=} 4.(\dots, x_{2i-1}[b - Ax]_{2i-1} + x_{2i}[b - Ax]_{2i}, \dots)^T \tag{22}$$

our next task is to minimize

$$\|\mathrm{err}\|^2 = \frac{1}{2}\lambda^T Q\lambda - q^T\lambda + (b - Ax)^T(b - Ax). \tag{23}$$

Further work depends on original problem formulation (due to Definition 3.3.1)

- *Equality problem*

$$\lambda = \min_{\lambda \in \mathbb{R}} \|\mathrm{err}\|^2$$

- *Inequality problem*

$$\lambda = \min_{\lambda \leq \mathbb{R}} \|\mathrm{err}\|^2$$

5.3.4 Minimum of error function without constraints

Let us consider Equality problem from Definition 3.3.1.
We look for $\lambda \in \mathbb{R}^n$ minimizing error function (23). So we have to find roots of first derivative.
At first we compute first derivative

$$\frac{\partial\left[\|\mathrm{err}\|^2\right]}{\partial \lambda_i} = Q\lambda - q$$

Remark: We used remark after Theorem (3.1.1) for minimizing quadratic function.
Q is symmetric positive definite matrix and

$$\frac{\partial (b - Ax)^T(b - Ax)}{\partial \lambda} = 0.$$

$\approx$

Now we put first derivative of error function equal to zero

$$\lambda = Q^{-1}q$$

and substitute (21) and (22)

$$\lambda = \frac{1}{8}.\begin{bmatrix} \ddots & & 0 \\ & \frac{1}{x_{2i-1}^2 + x_{2i}^2} & \\ 0 & & \ddots \end{bmatrix}.4.\begin{bmatrix} \vdots \\ x_{2i-1}[b - Ax]_{2i-1} + x_{2i}[b - Ax]_{2i} \\ \vdots \end{bmatrix}.$$

We express prescription for i-th element of λ

$$\lambda_i = \frac{1}{2(x_{2i-1}^2 + x_{2i}^2)}\left(x_{2i-1}[b - Ax]_{2i-1} + x_{2i}[b - Ax]_{2i}\right). \tag{24}$$

Thence Inverse Dual problem for Equality problem (see Definition 3.3.1) can be solved using equation (24). But for Inequality problem we use MPRGP algorithm (see Section 4.4).

6 Lagrange multipliers

In this chapter we describe Lagrange multipliers and their relation to a proper parameter. We consider Equality problem (see Definition 3.3.1). In our exploration figures will be very useful.

6.1 Lagrange multiplier as linear penalty

6.1.1 Linear penalty introduction

Let us consider function $\tilde{f}_v(x)$ given by

$$\tilde{f}_v(x) \stackrel{\text{def}}{=} f(x) + v^T g(x) \tag{25}$$

where $v \in \mathbb{R}^n$ is *appropriately* chosen constant vector with positive components.
We refer to v as the *linear penalty parameter*.
We can derive these properties:

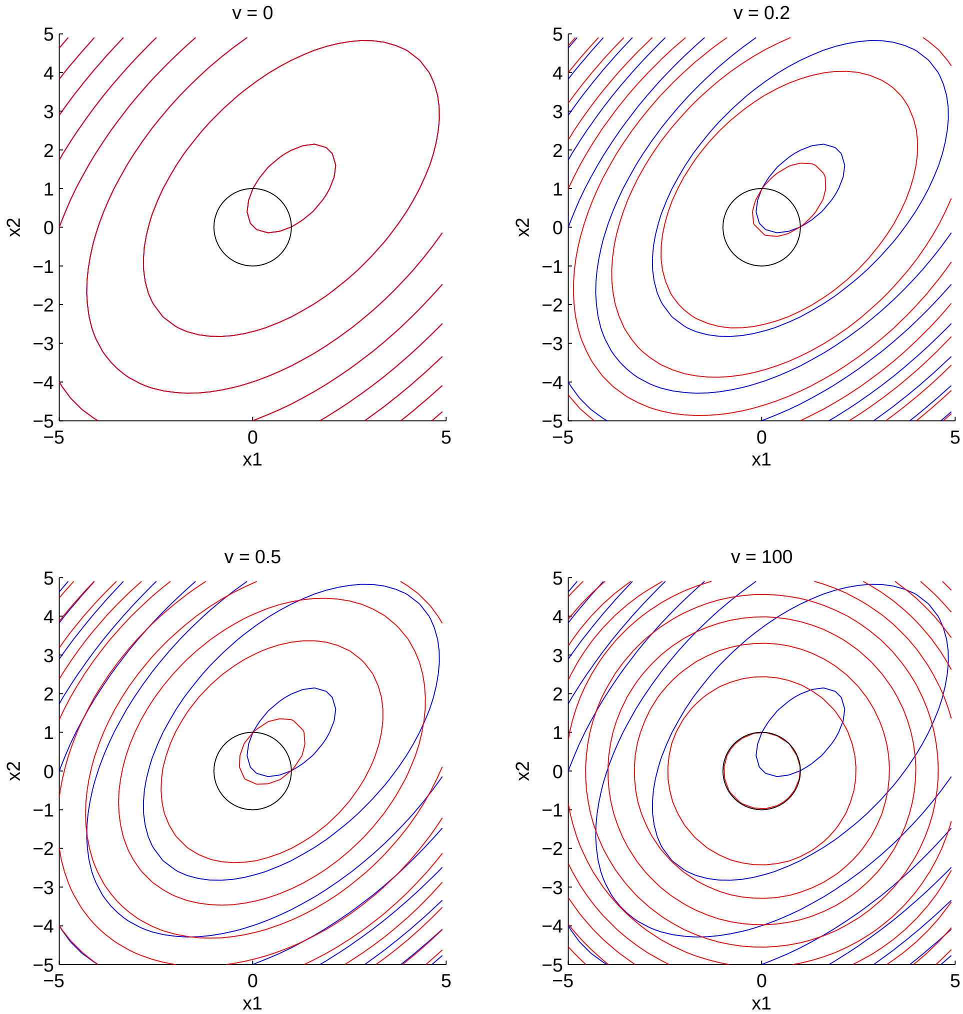
- if $x \in \partial\Omega_I$, then $g(x) = o$,
  thus $\tilde{f}_v(x) = f(x) + v^T g(x) = f(x) + v^T o = f(x)$,
- if $x \in \Omega_I \setminus \partial\Omega_I$, then $-c \leq g(x) < o$ (meaning $-c_i \leq g_i(x) < 0, \forall i = 1 \dots n$),
  hence $\tilde{f}_v(x) = f(x) + v^T g(x) < f(x)$
- if $x \in \mathbb{R}^{2n} \setminus \Omega_I$ then $g(x) > o$ (meaning $g_i(x) > 0, \forall i = 1 \dots n$),
  therefore $\tilde{f}_v(x) = f(x) + v^T g(x) > f(x)$.

Due to the observations we can say that *linear penalty modifies value subject to constraints - it increases values in $\mathbb{R}^{2n} \setminus \Omega_I$ and decreases in $\Omega_I \setminus \partial\Omega_I$.*

Example 6.1.1
Let us have specific values of Equality problem with one constraint

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, r = 1$$

and draw isolines of original function $f(x)$ and function with linear penalty $\tilde{f}_v(x)$ subject to one quadratic constraint. We try some different values of linear penalty parameter v.

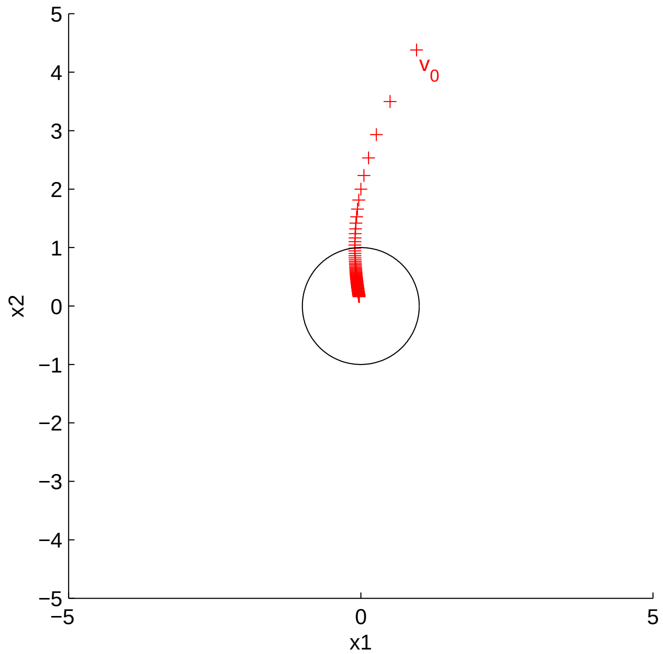


■

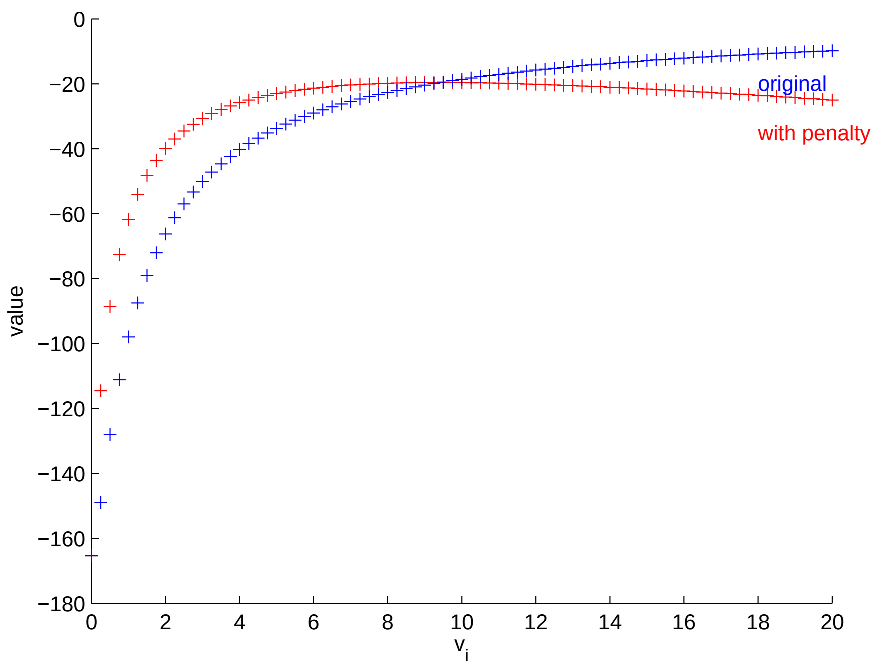
Example 6.1.2
Another values can be

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 4 \\ 20 \end{bmatrix}, r = 1$$

But now, for more illustrative example, we put a cross into figure on coordinates where, for concrete value of v, the real minimum of $\tilde{f}_v(x)$ is.
We try to set $v_0 = 0$, $v_{i+1} = v_i + 0.25, i = 0, 1..20$.



We can also demostrate difference between value in minimum of original function f and function $\tilde{f}$ using specific v_i.



■

6.1.2 Lagrangie multipliers as linear penalty parameter

For next consideration we will need prescription of common Lagrange function

$$L(x, \lambda) \stackrel{\text{def}}{=} f(x) + \lambda^T g(x)$$

Linear penalty has the same rules as Lagrange function, so logically, we can consider the vector of Lagrange multipliers as linear penalty parameter.

From Example 6.1.2 we can note that if we increase Lagrange multiplier, the minimum of $L(x, \lambda)$ will be more closer to the center of the circle defined by constraint function $g(x)$.

Written in limit form

$$\lim_{\lambda \to \infty} (\arg\min L(x, \lambda)) \to o$$

where convergence $\lambda \to \infty$ means

$$\forall i = 1 \ldots n : \lambda_i \to \infty.$$

6.2 Lagrange multipliers sequence

Let us consider a simple two-dimensional problem.
That means, we have only one constraint and also only one Langrange multiplier.
We already tried to find minimum of quadratic function, but this minimum is not from Ω_E (see Definition 3.2.2). There exists $\bar{\lambda}$ which is efficient to construct function $L(x, y)$ which minimum is in this set. At this point x, the first KKT condition is accomplished (it is the minimum of Lagrange function at all) also the second (this x is from Ω_E). We refer to this point as the $\bar{x}$.
Let us get back to Example 6.1.2. In fact, we construct a sequence of Lagrange multipliers

$$\lambda_1 < \lambda_2 < \ldots < \bar{\lambda} < \ldots < \infty$$

and we stepwise by substitute members of this sequence to Lagrange function. Minimum of this function started to move towards Ω_E, but it didn't stop in Ω_E, but it continues to zero point o (to the centre of the circle described by quadratic constraint). Now our task is to find $\bar{\lambda}$ corresponding to minimum $\bar{x}$ of Lagrange function in Ω_E.

6.3 Constant Update of Lagrange multipliers

We simply try to put some values of Langrange multipliers into Dual problem. Since we want to show how Dual problem works, we choose simple equidistant arithmetic progression with convenient $\epsilon \in \mathbb{R}$ difference.

$$\lambda_{k+1} = \lambda_k + \epsilon \tag{26}$$

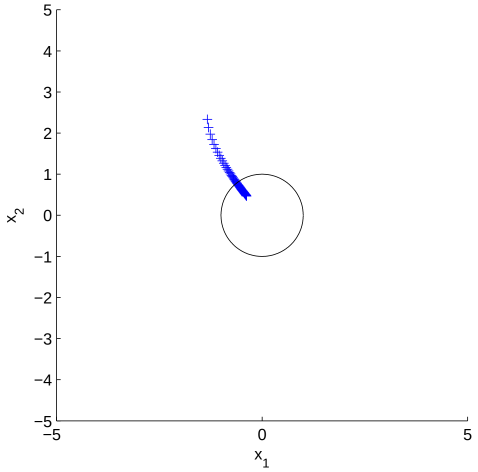
Listing 1: Constant update Lagrangian method

```
lambdas = 0:epsilon:lambda_max;

for i=1:(length(lambdas))
        xi(i,:) = cg(A + 2 * diag([lambdas(i),lambdas(i)]),b,x0,e);
end
```

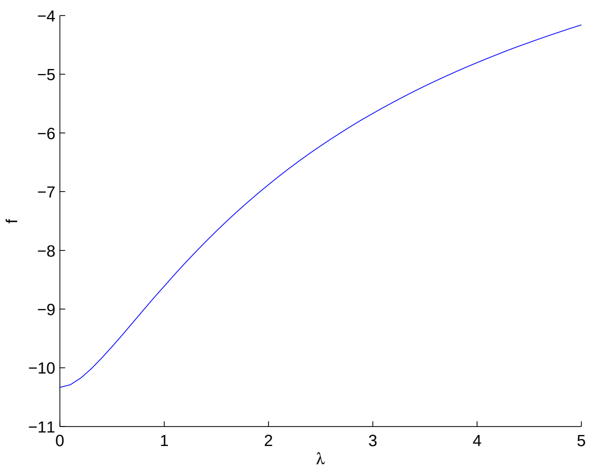
Example 6.3.1
Consider input data

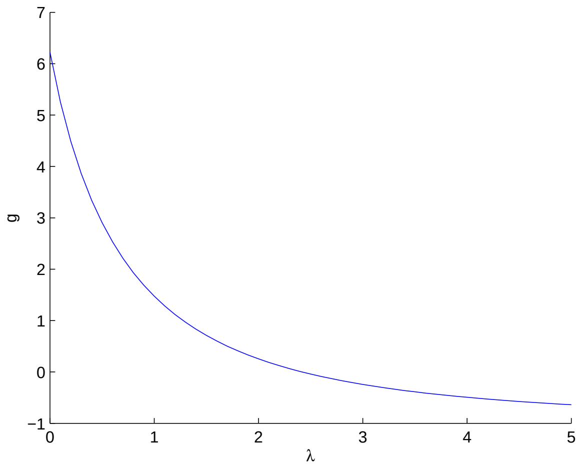
$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}, b = \begin{pmatrix} -5 \\ 6 \end{pmatrix}, r = 1, \epsilon = 0.1, \lambda_{\max} = 5$$

If we try to plot aproximations x_k, we get something like this:



and in case that we evaluate quadratic function and quadratic constraint:





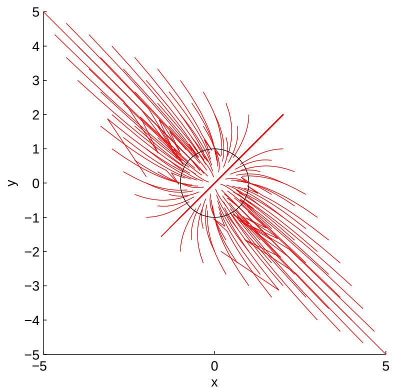
■

6.3.1 Sequence of Update Lagrange algorithm aproximation using different input data

Example 6.3.2
Let us consider testing data

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, r = 1, \epsilon = 0.1, \lambda_{\max} = 2$$

and let us try to plot sequence of minima for different right side vectors. We choose $b \in \{-5, \dots, 5\} \times \{-5, \dots, 5\}$. Output:



■

7 Simple update Lagrange methods

We consider Equality problem (see Definition 3.3.1) with one quadratic constraint. From previous observations in Chapter 6 we know how to *move aproximations towards* equality constraint set Ω_E. But we do not know how to stop this progression. In this chapter we try some simple algorithms which solve this problem.

7.1 Linear constraint update

Let us consider prescription

$$\lambda_{k+1} = \lambda_k + \rho.g(x_k)$$

where ρ is sufficiently small real constant.
This prescription tries to update Lagrange multiplier using sofisticated method - size of update is adequate to distance of actual aproximation from Ω_E.

Using this prescription we construct algorithm:

- input
  - $A \in \mathbb{R}^2 \times \mathbb{R}^2$ - SPD matrix
  - $b \in \mathbb{R}^2$ - right side vector
  - $r \in \mathbb{R}$ - radius of boundary
  - $e \in \mathbb{R}$ - precision of algorithm
  - $x_0 \in \mathbb{R}^2$ - initial approximation
  - $\lambda_0 = 0$ - initial approximation of Lagrange multiplier
  - $k = 0$ - iterator
- while $x_k^T.x - r > e$ do
  - $x_{k+1} = cg(A + 2 * diag([\lambda_k, \lambda_k]), b, x_k, e)$
  - $\lambda_{k+1} = \lambda_k + \rho.(x_{k+1}^T.x_{k+1} - r)$
  - $k = k + 1$

where $cg(A, b, x_0, e)$ is implemented algorithm of Conjugated gradient method, see Section 4.3.

Example 7.1.1
Let us choose the following input data

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 4 \\ 20 \end{bmatrix}, x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, e = 10^{-4}, r = 1.$$

Using different constant coeficients ρ, algorithm find solution subject to precision using different number of iterations:

ρ	# of iterations
0.020	2779
0.021	2340
0.022	5000+
0.023	1791
0.023	1791
0.024	1872
0.025	1602
0.026	1542
0.027	1503
0.028	1843
0.029	1600
0.03	1232
0.031	1528
0.0311	922
0.0312	623
0.03121	2295
0.03122	1105
0.03123	735
0.03124	653
0.03125	541
0.03126	407
0.03127	200
0.03128	-
0.0313	-
0.032	-
0.04+	-

■

7.2 Adaptive linear constraint update

We modify previous algorithm - we find adequate coeficient ρ by testing and making shorter in every iteration.

- input
  - $A \in \mathbb{R}^2 \times \mathbb{R}^2$ - SPD matrix
  - $b \in \mathbb{R}^2$ - right side vector
  - $r \in \mathbb{R}$ - radius of boundary

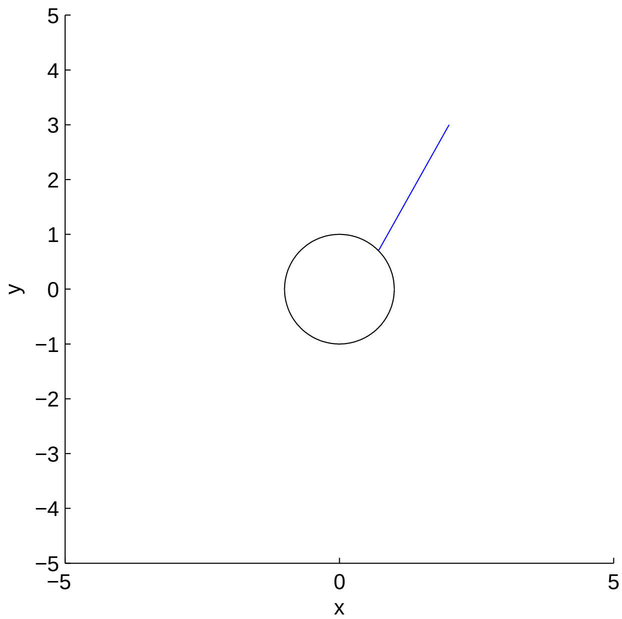
- $e \in \mathbb{R}$ - precision of algorithm
- $x_0 \in \mathbb{R}^2$ - initial approximation
- $\lambda_0 = 0$ - initial approximation of Lagrange multiplier
- $\rho = 1$ - initial update coefficient
- $k = 0$ - iterator

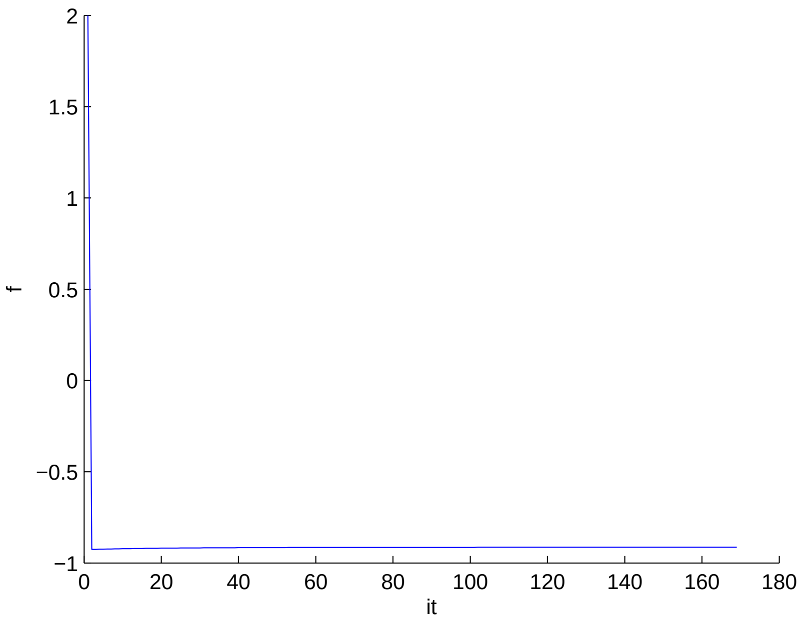
- while $x_k^T.x_k - r > e$ do
  - try to update: $\lambda_{test} = \lambda_k + \rho.(x_{k+1}^T.x_{k+1} - r)$
  - compute testing aproximation: $x_{k+1} = cg(A + 2 * diag([\lambda_{test}, \lambda_{test}]), b, x_k, e)$
  - while $x_{test}^T.x_{test} - r < e$
    * $\rho = \frac{\rho}{2}$
    * try to update: $\lambda_{test} = \lambda_k + \rho.(x_{k+1}^T.x_{k+1} - r)$
    * compute testing aproximation: $x_{test} = cg(A + 2 * diag([\lambda_{test}, \lambda_{test}]), b, x_k, e)$
  - $\lambda_{k+1} = \lambda_{test}$
  - $x_{k+1} = x_{test}$

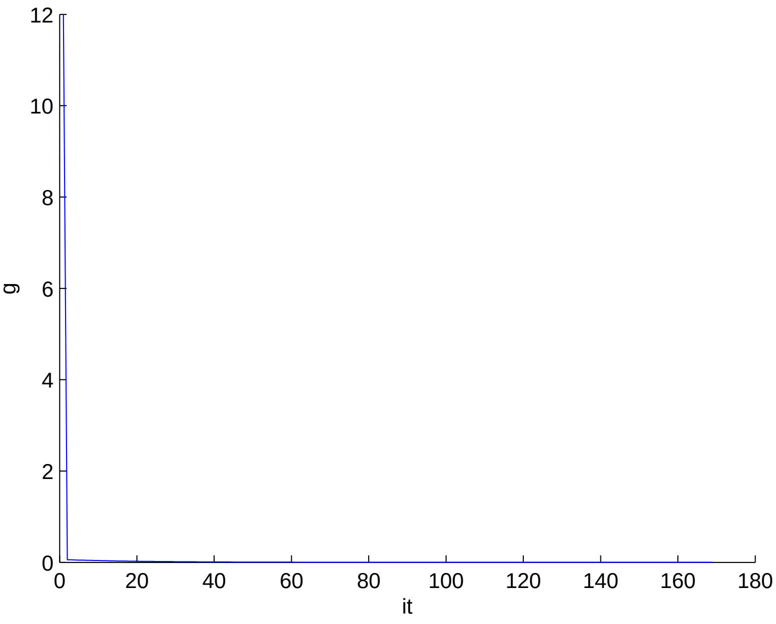
Example 7.2.1
Consider input data

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, r = 1, x_0 = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, eps = 10^{-4}$$

Output of this algorithm:







■

7.3 Bisection method

In this algorithm we try to find λ_{max} using bisection method.
There exists sufficient by large λ_{max} such that

$$g(x(\lambda_{max})) < 0$$

Then our solution $\bar{x} = x(\bar{\lambda})$ with $\bar{\lambda} \in (0, \lambda_{max})$. We search this $\bar{\lambda}$ using Bisection method with stop condition

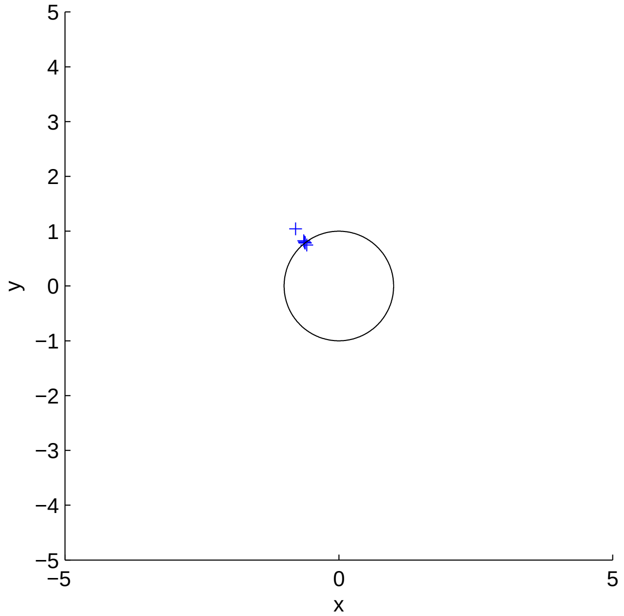
$$|g(x(\bar{\lambda}))| < \epsilon$$

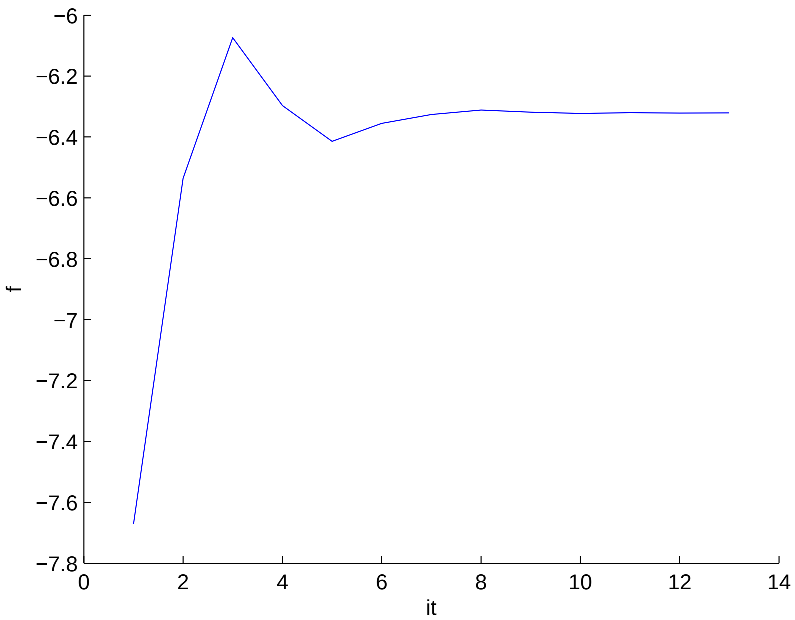
where $\epsilon > 0$ is required precision.

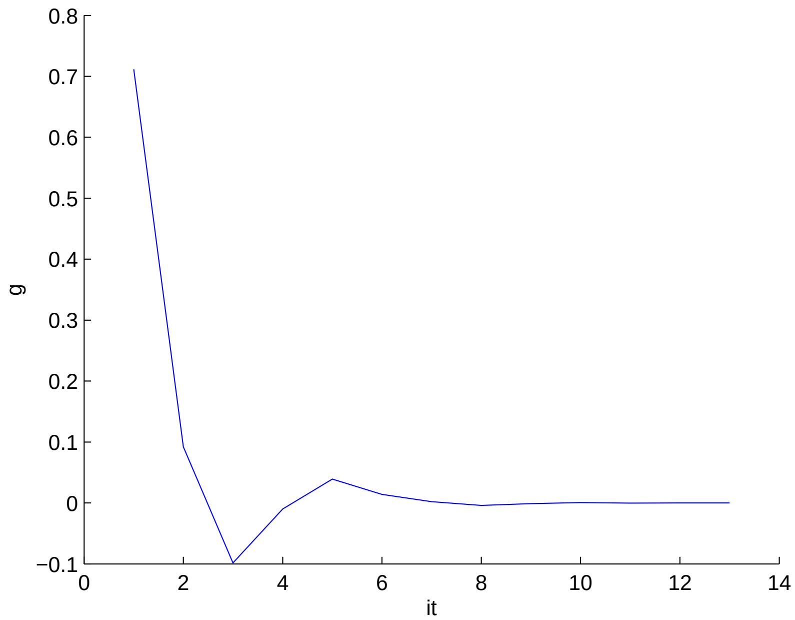
Listing 2: bisect

```
% find any lambda_max
lambda_max = 0;
x_max = cg(A + 2 * diag([lambda_max,lambda_max]),b,x_00,e);
while (x_max'*x_max - c) > e
    lambda_max = lambda_max + 1; % try to increase
    x_max = cg(A + 2 * diag([lambda_max,lambda_max]),b,x_00,e);
end

% initialization
a_bisect = 0; % lower estimation
b_bisect = lambda_max; % upper estimation
s_bisect = (a_bisect+b_bisect)/2; % pivot
x = cg(A + 2 * diag([s_bisect,s_bisect]),b,x_00,e);

% main iterations
while abs(x'*x - r) > e
    % compute new interval
    if x'*x - r > 0
        a_bisect = s_bisect;
    else
        b_bisect = s_bisect;
    end

    % compute new pivot
    s_bisect = (a_bisect + b_bisect)/2;
    x = cg(A + 2 * diag([s_bisect,s_bisect]),b,x_00,e);
end
```

Output of this algorithm:







7.4 Numerical tests

$$A_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, A_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, A_3 = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}, A_4 = \begin{bmatrix} 2 & 2 \\ 2 & 3 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -5 \\ 6 \end{bmatrix}, b_2 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, b_3 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, b_4 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

input		adaptive ULM				bisection method			
A	**b**	x_1	x_2	λ	**it**	x_1	x_2	λ	**it**
A_1	b_1	-0.654301	0.756363	3.399414	523	-0.654208	0.756261	3.399414	12
A_1	b_2	0.585708	0.810689	1.723145	52	0.585623	0.810529	1.723145	12
A_1	b_3	-1.000000	-1.000000	0.621338	8	-0.707101	-0.707101	0.621338	12
A_1	b_4	-0.224782	0.974464	1.167847	70	-0.224741	0.974410	1.167847	14
A_2	b_1	-0.616196	0.787549	2.418091	73	-0.616200	0.787545	2.418091	13
A_2	b_2	0.645064	0.764311	2.693481	170	0.644944	0.764177	2.693481	13
A_2	b_3	-0.039247	-0.039247	1.621338	6	-0.707101	-0.707101	1.621338	13
A_2	b_4	0.224782	0.974464	1.167847	70	0.224741	0.974410	1.167847	14
A_3	b_1	-0.578076	0.816043	3.530762	926	-0.578042	0.815980	3.530762	13
A_3	b_2	0.438290	0.898793	2.037598	192	0.438317	0.898819	2.037598	11
A_3	b_3	-0.432857	-0.901445	0.923828	121	-0.432894	-0.901480	0.923828	9
A_3	b_4	-0.158926	0.987283	1.606201	252	-0.158925	0.987305	1.606201	13
A_4	b_1	-0.697204	0.721933	3.635254	1000	-0.694557	0.719450	3.635254	13
A_4	b_2	0.641243	0.767328	0.922363	31	0.641220	0.767341	0.922363	11
A_4	b_3	-0.948984	-0.315123	0.248535	34	-0.949028	-0.315105	0.248535	11
A_4	b_4	-0.419434	0.907905	1.164917	183	-0.419347	0.907851	1.164917	14

8 Projected Dual problem (PDP) algorithm

In the most important chapter of this thesis we introduce new algorithm for solving Equality and Inequality problems. At first we introduce projection to boundary of set and then use observations in previous chapters to construct PDP algorithm.

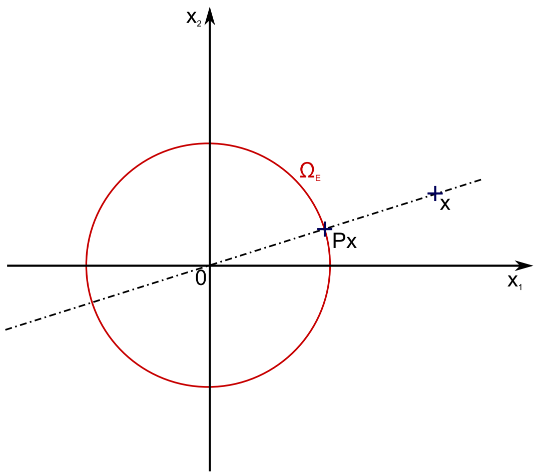
8.1 Projection

Our following problem is to find the nearest $Px \in \mathbb{R}^2$ to x, which satisfy II. KKT condition (11) in the best way.

Definition 8.1.1 (Projection)

$$\forall x \in \mathbb{R}^2 \setminus \{o\} : Px \stackrel{\text{def}}{=} \frac{r}{\|x\|_2} x$$

Remark: We simply normalize vector of actual iteration x and then extend it to r, thus $g(Px) = 0 \Leftrightarrow Px \in \Omega_E, \partial\Omega_I$.



≈

Theorem 8.1.1
For every iteration $x_k \in \mathbb{R}^2$ is Px_k from (8.1.1) the nearest point accomplishing II. KKT condition (11).

$$\forall x_k \in \mathbb{R}^2 \forall y \in \mathbb{R}^2 : (g(y) = 0 \wedge y \neq Px_k) \Rightarrow (\|x_k - Px_k\|_2 < \|x_k - y\|_2)$$

Proof: For projection holds

$$\|x_k\|_2 = \|Px_k\|_2 + \|x_k - Px_k\|_2$$

so

$$\|x_k - Px_k\|_2 = \|x_k\|_2 - \|Px_k\|_2 = \|x_k\|_2 - r$$

and because $g(y) = 0 \Rightarrow \|y\|_2 = r$, we have

$$\|x_k\|_2 - r = \|x_k\|_2 - \|y\|_2 = \|(x_k - y) + y\|_2 - \|y\|_2$$

For every norm $\|x + y\| \leq \|x\| + \|y\|$, so we can write

$$\|(x_k - y) + y\|_2 - \|y\|_2 \leq \|x_k - y\|_2 + \|y\|_2 - \|y\|_2 = \|x_k - y\|$$

Equality is possible, only if $x_k - y = -y \Rightarrow x_k = 0$. For this point, projection (8.1.1) is not defined.
So we can say

$$\|x - Px_k\|_2 < \|x_k - y\|_2$$

□

Definition 8.1.2 (Projection in more dimensions)
For every iteration

$$x \in \mathbb{R}^{2n} \setminus \{x \in \mathbb{R}^{2n} : \|(x_{2i-1}, x_{2i})\|_2 \neq 0, i = 1, 2, \dots n\}$$

we define projection

$$Px = (P(x_1, x_2), \dots P(x_{2i-1}, x_{2i}), \dots P(x_{2n-1}, x_{2n}))^T$$

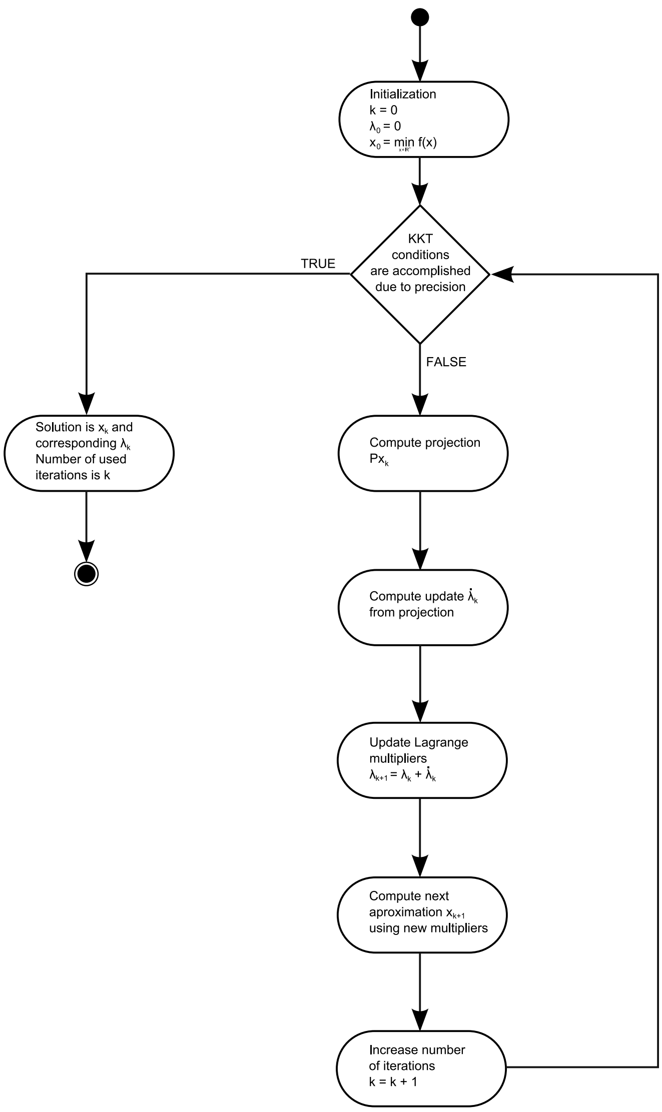
8.2 Idea of PDP

Previous algorithms in Chapter 7 (except Bisection method) update Lagrange multipliers from previous iteration by multiple of value of quadratic contraint in this iteration. Now we try to compute this update using more sofisticated process - we use Lagrange multiplier corresponding to *projection* of previous iteration *to boundary of constraint set*.
We will use update prescription

$$\lambda_{k+1} = \lambda_k + \dot{\lambda}_k$$

where

- λ_k is Lagrange multiplier from previous iteration
- λ_{k+1} is Lagrange multiplier corresponding to next iteration
- $\dot{\lambda}_k$ is update

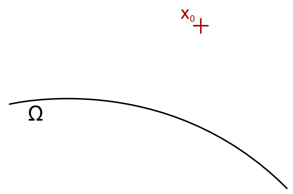


The algorithm consists of these steps:

- *Initialization*
  Find minimum of quadratic function without constraints

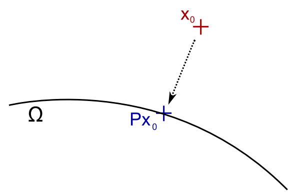
$$x_0 = \min_{x \in \mathbb{R}^{2n}} f(x)$$

  using CG method. Set $\lambda_0 = 0$.



- *KKT conditions accomplishment*
  The algorithm is over, if both of KKT conditions are accomplished due to precision. Because first KKT condition is accomplished in every iteration (every next iteration is computed using dual problem solver), we simply test accomplishment of second KKT condition.

- *Projection computation*
  Compute projection of actual iteration using Definition 8.1.2.

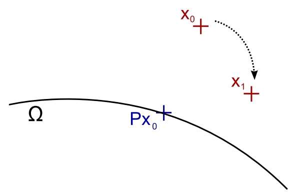


- *Update computation*
  Minimize inverse dual problem function - find Lagrange multiplier corresponding to projection using:
  - CG algorithm for finding $\dot{\lambda}_k \in \mathbb{R}$ without confidement - if original problem is with *equality constraints* (or use prescription from Section 5.3.4),
  - MPGRP algorithm for finding $\dot{\lambda}_k > 0$ - if original problem is with *inequality constraints*.

- *Lagrange multipliers update*
  Compute next Lagrange multiplier by updating

$$\lambda_{k+1} = \lambda_k + \dot{\lambda}_k$$

- *Next aproximation computation*
  Find next aproximation x_{k+1} corresponding to λ_{k+1}, using Dual problem definition 5.2.1 - use CG algorithm.



8.3 Inequality constraints

8.3.1 Characterization

Denote (x_k, λ_k) k-th iteration (solution aproximation and corresponding vector of Langrange multipliers in k-th iteration, this pair solve *Dual problem*, see Definition 5.2.1). Next iteration (x_{k+1}, λ_{k+1}) can be expressed from previous one.
We compute first iteration (x_0, λ_0):

- $\lambda_0 = 0$
- $x_0 = \min_{x \in \mathbb{R}^{2n}} f(x)$ is minimum without constraints
  (can be computed using CG method, see Section 4.3)

In each iteration we compute pair (x_{k+1}, λ_{k+1}) using this method:
(We denote for simplicity $(x_k, \lambda_k) = (x, \lambda)$)

- projection of previous iteration

$$Px_k = [\dots \frac{r_i}{\|(x_{2i-1}, x_{2i})\|_2}(x_{2i-1}, x_{2i}) \dots]^T$$

  (projection in more dimensions see Definition 8.1.2)

- Langrange multiplier from projection

$$Q = 8.\ \mathrm{diag}(\dots, (Px)_{2i-1}^2 + (Px)_{2i}^2, \dots)$$

$$q = 4.(\dots, (Px)_{2i-1}[b - A(Px)]_{2i-1} + (Px)_{2i}[b - A(Px)]_{2i}, \dots)^T$$

$$\dot{\lambda}_k = \min_{\lambda \geq 0} \frac{1}{2}\lambda^T Q \lambda - q^T \lambda$$

  (minimum of Invert Dual problem error function with constraint $\lambda \geq 0$, see equation (23) in Section 5.3.3) for solving this problem, we use minimalization algorithm MPRGP, see Chapter 4.4.

- update Lagrange multipliers

$$\lambda_{k+1} = \lambda_k + \dot{\lambda}_k$$

- compute next iteration using new multipliers

$$x_{k+1} = (A + 2\mathrm{diag}(\dots, [\lambda_{k+1}]_i, [\lambda_{k+1}]_i, \dots))^{-1} b$$

  (for solving this system can be used CG method see Chapter 4.3)

8.3.2 Algorithm in Matlab

Main algorithm:

Listing 3: pdp ineq

```
% initialization
k = 0;
x_k = cg(A,b,x_00,eps);
lambda_k = zeros(length(x_k),1);

% main iterations
while  ~is_in_omega(x_k,r,eps)
    % projection
    Px_k = projection(x_k,r,eps);

    % find update
    lambda_dot_k = get_lambda(A + 2 * diag(lambda_k),b,r,Px_k,eps\);

    % update lagrange multipliers
    lambda_k = lambda_k + lambda_dot_k;

    % find next aproximation using Dual problem
    x_k = cg(A + 2*diag(lambda_k),b,x_k,eps);

    % increase iteration counter
    k = k + 1;
end
```

Stop condition:

Listing 4: is in omega

```
function [return_value] = is_in_omega(x,r,eps)
  return_value=true;
  for i=1:(length(x)/2)
    if(~satisfy_quadratic_constrain(x((2*i-1):(2*i)),r(i),eps))
        x(2*i-1)^2 + x(2*i)^2 - r(i)^2
        return_value = false;
    end
  end
end
```

Verify condition:

Listing 5: satisfy condition

```
function [return_value] = satisfy_quadratic_constrain(x,r,eps)
      if x(1)^2 + x(2)^2 - r^2 <= eps
          return_value = true;
      else
          return_value = false;
      end
end
```

Projection:

Listing 6: projection

```
function [x] = projection(x,r,eps)
     for i = 1:(length(x)/2) % for all constraints
         x_couple = x((2*i-1):(2*i));
        % compute projection to actual boudary
         x((2*i-1):(2*i)) = (r(i))/(sqrt(x_couple(1)^2+x_couple(2)^2)) ...
                                    * x_couple;
     end
end
```

Update computation:

Listing 7: compute update

```
function [lambda_out] = get_lambda(A,b,r,x,eps)
       reziduum = b-A*x;
       Q = zeros(length(x)/2,length(x)/2);
       q = zeros(length(x)/2,1);
       for i = 1:length(x)/2
          Q(i,i) = 8*(x(2*i-1)^2 + x(2*i)^2);
          q(i) = 4*(x(2*i-1)*reziduum(2*i-1) + x(2*i)*reziduum(2*i));
       end
       % compute solution using MPRGP
       lambda = mprgp(Q,q,zeros(length(x)/2,1),eps);

       lambda_out = zeros(length(lambda)*2,1);
       for i = 1:length(lambda)
         lambda_out(2*i-1) = lambda(i);
         lambda_out(2*i) = lambda(i);
       end
end
```

8.4 Radius scaling

We shall remind minimizing problem with separable inequality quadratic constraints (**??**):
Find $\bar{x} \in \mathbb{R}^{2n}$ such that

$$\begin{array}{l} \bar{x} \stackrel{\text{def}}{=} \min_{x \in \Omega} f(x) \\ f(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T A x - b^T x \\ \Omega \stackrel{\text{def}}{=} \{x \in \mathbb{R}^{2n} : g_i(x) \leq 0, i = 1, 2, \ldots n\} \\ g_i(x) \stackrel{\text{def}}{=} x_{2i-1}^2 + x_{2i}^2 - r_i^2 \end{array} \tag{27}$$

where

- $n \in \mathbb{N}$ is problem dimension, resp. number of constraint functions
- $f : \mathbb{R}^{2n} \to \mathbb{R}$ is quadratic function
- $A \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}$ is symetric positive definite matrix
- $b \in \mathbb{R}^{2n}$ is vector of right-hand sides
- $r \in \mathbb{R}^n$ is vector of radii

Definition 8.4.1 (Identity of radius)

We say that problem 27 has identical radius $r = \rho \in \mathbb{R}$ if

$$\forall i = 1, \ldots n : r_i = \rho$$

Let us consider constraint function $g_i(x) \leq 0$. We try to *identity* its radius

$$g_i(x) \leq 0$$

$$x_{2i-1}^2 + x_{2i}^2 - r_i^2 \leq 0$$

$$x_{2i-1}^2 + x_{2i}^2 \leq r_i^2$$

$$\rho^2 x_{2i-1}^2 + \rho^2 x_{2i}^2 \leq \rho^2 r_i^2$$

$$\frac{\rho^2}{r_i^2} x_{2i-1}^2 + \frac{\rho^2}{r_i^2} x_{2i}^2 \leq \rho^2$$

$$\tilde{x}_{2i-1}^2 + \tilde{x}_{2i}^2 \leq \rho^2$$

where we used subsitution

$$\begin{array}{l} \tilde{x}_{2i-1} = \frac{\rho}{r_i} x_{2i-1} \\ \tilde{x}_{2i} = \frac{\rho}{r_i} x_{2i} \end{array} \tag{28}$$

We can use this substitution to whole vector x:

$$\tilde{x} = Rx, \quad R \stackrel{\text{def}}{=} \operatorname{diag}(\frac{\rho}{r_1}, \frac{\rho}{r_1}, \dots \frac{\rho}{r_n}, \frac{\rho}{r_n}) \tag{29}$$

Then $x \in \Omega$ is equivalent to $Rx \in \{x \in \mathbb{R}^{2n} : \tilde{g}_i(x) \leq 0, i = 1, 2, \dots n\}$, where

$$\tilde{g}_i(x) \stackrel{\text{def}}{=} x_{2i-1}^2 + x_{2i}^2 - \rho^2 \tag{30}$$

Now we can express $f(\tilde{x})$ using the previous substitution

$$f(\tilde{x}) = f(Rx) = \frac{1}{2}(Rx)^T A(Rx) - b^T(Rx) = \frac{1}{2}x^T RARx - (Rb)^T x = \frac{1}{2}x^T \tilde{A}x - \tilde{b}^T x$$

where

$$\begin{aligned} \tilde{A} &= RAR \\ \tilde{b} &= Rb \end{aligned} \tag{31}$$

Theorem 8.4.1 (Problems equivalency)

Solution of problem 27 denoted as $\bar{x}$ is equivalent (after substitution $\bar{x} = R^{-1}\bar{\tilde{x}}$) to solution of problem with identical radius:
Find $\bar{\tilde{x}} \in \mathbb{R}^{2n}$ such that

$$\begin{aligned} &\bar{\tilde{x}} \stackrel{\text{def}}{=} \min_{x \in \Omega} \tilde{f}(x) \\ &\tilde{f}(x) \stackrel{\text{def}}{=} \tfrac{1}{2}x^T \tilde{A}x - \tilde{b}^T x \\ &\Omega \stackrel{\text{def}}{=} \{x \in \mathbb{R}^{2n} : \tilde{g}_i(x) \leq 0, i = 1, 2, \dots n\} \\ &\tilde{g}_i(x) \stackrel{\text{def}}{=} x_{2i-1}^2 + x_{2i}^2 - \rho^2 \end{aligned}$$

8.5 Numerical tests

Example 8.5.1
Let us consider input data:

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, r = \begin{pmatrix} 1 \end{pmatrix}, eps = 0.0001$$

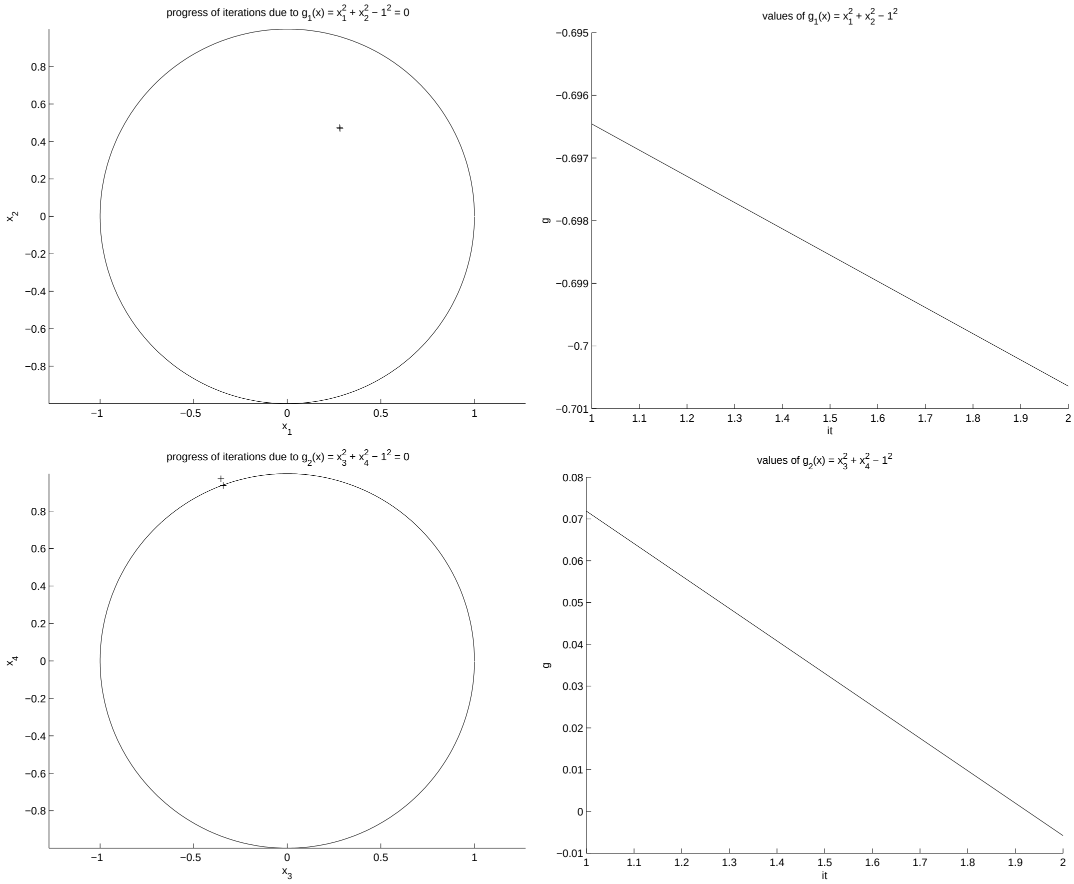
Algorithm is over in one iteration.

■

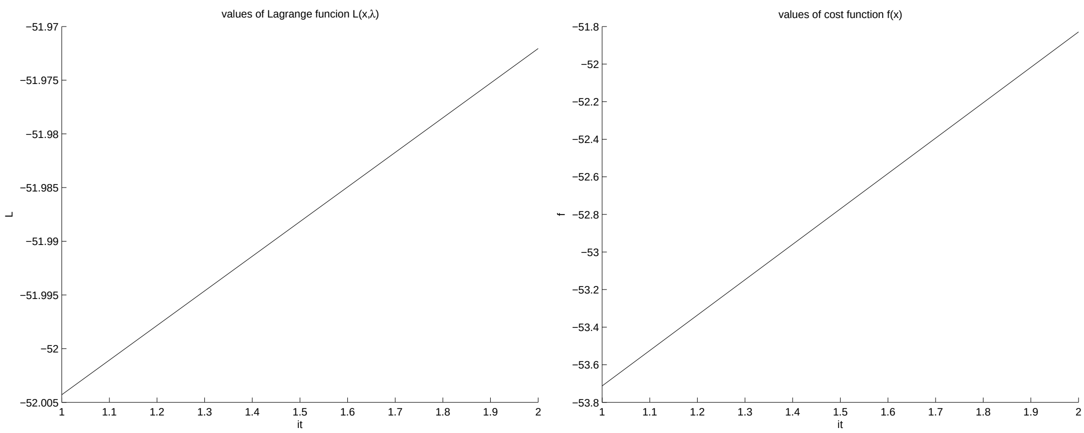
Example 8.5.2
Let us consider input data:

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}, b = (1, 1, -20, 50)^T, r = (1, 1)^T, eps = 0.0001$$

Algorithm is over in two iterations.
Iteration progress and progress of constraint functions values:



Function value progress and progress of Lagrange function values:



We can verify our solution - induct solution into KKT conditions:

$$\text{KKT1}_{err} = Ax - b + 2\,\text{diag}(\tilde{\lambda})x = 10^{-13}. \begin{pmatrix} 0 \\ -0.0033 \\ -0.0711 \\ 0.1421 \end{pmatrix}$$

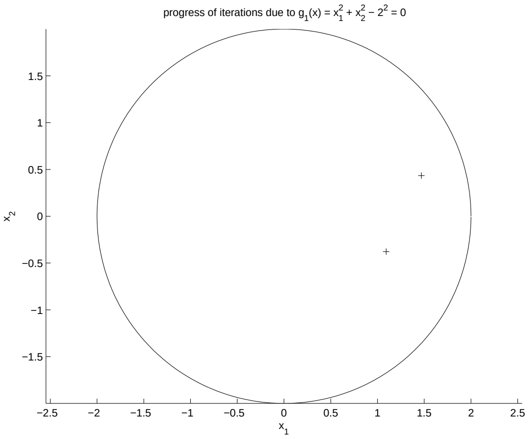
$$g(x) = \begin{pmatrix} -0.7006 \\ -0.0058 \end{pmatrix}$$

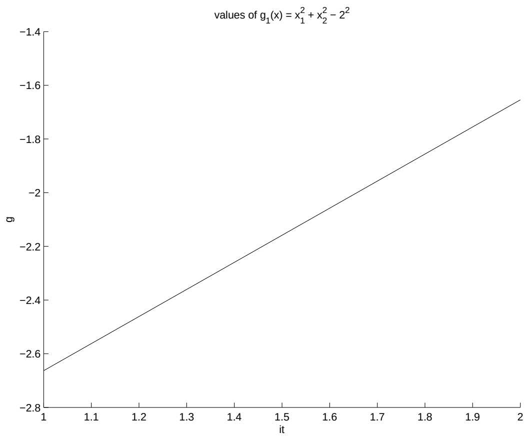
■

Example 8.5.3
(with large variability of radius)[2]
Let us consider input data

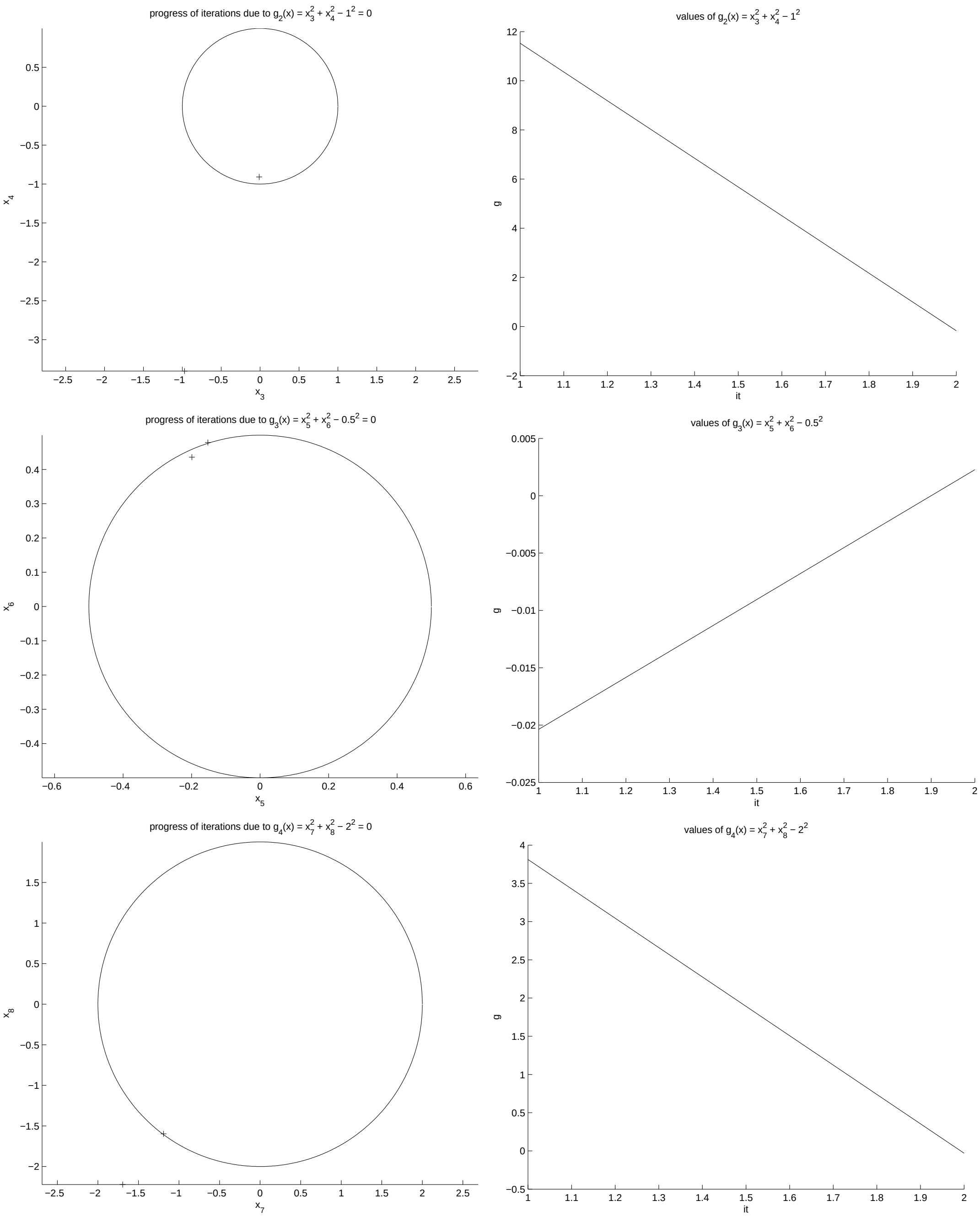
$$\begin{aligned} &A = \text{ fivediag}(-1,-1,4,-1,-1) \in \mathbb{R}^{12\times 12} \\ &b = Ay \\ &y = (2,1,0.5,0,0,11,10^{-5},-1,\sqrt{2},-0.1,4.1*10^{-4},143)^T \\ &r = (2,1,0.5,2,10^{-3},154)^T \\ &eps = 10^{-4} \end{aligned}$$

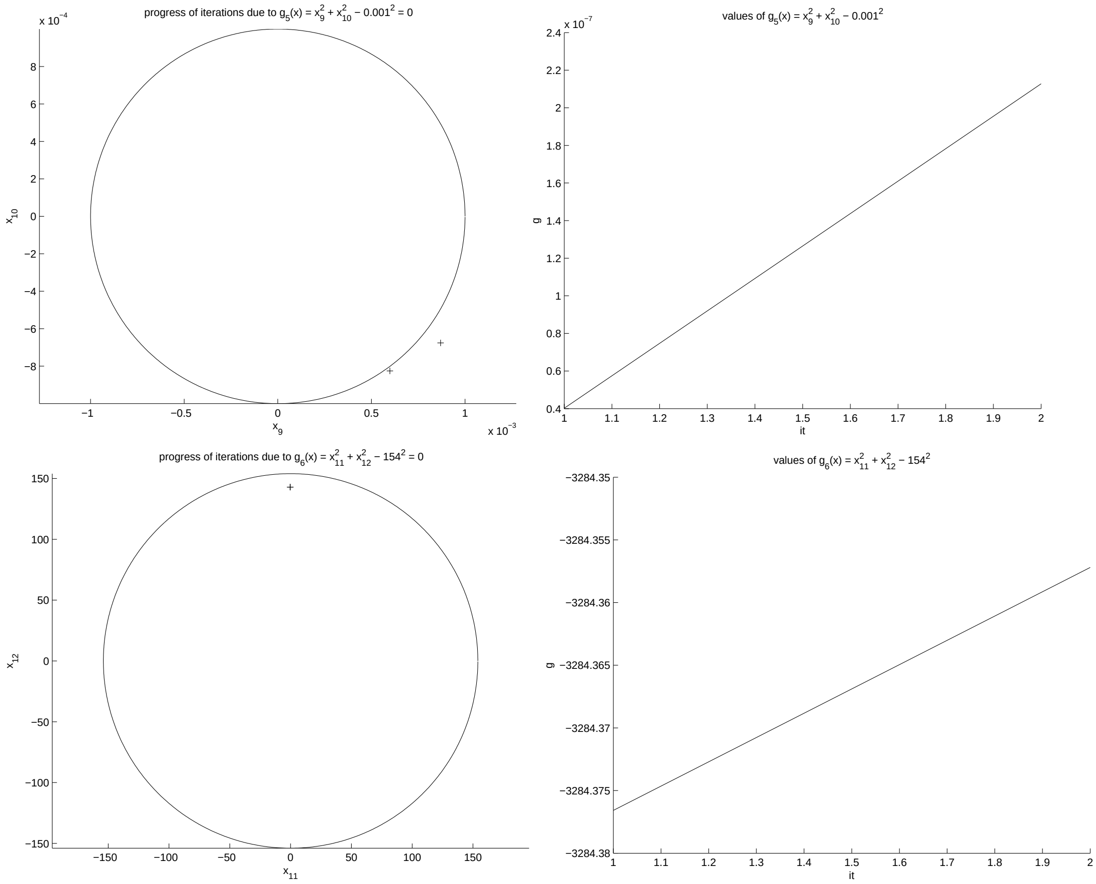
Algorithm is over in 2 outer iterations.
Iteration progress and progress of constraint functions values:



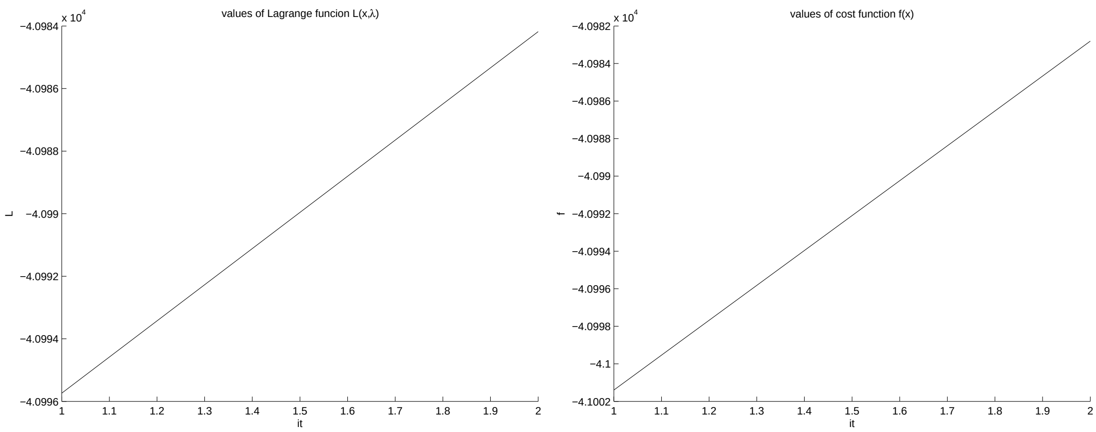


[2]This example was introduced and solved in [4]


progress of iterations due to g2(x) = x3^2 + x4^2 − 1^2 = 0
x3
x4
values of g2(x) = x3^2 + x4^2 − 1^2
it
g
progress of iterations due to g3(x) = x5^2 + x6^2 − 0.5^2 = 0
x5
x6
values of g3(x) = x5^2 + x6^2 − 0.5^2
it
g
progress of iterations due to g4(x) = x7^2 + x8^2 − 2^2 = 0
x7
x8
values of g4(x) = x7^2 + x8^2 − 2^2
it
g



Function value progress and progress of Lagrange function values:



We can verify our solution - induct solution into KKT conditions:

$$\mathrm{KKT1}_{err} = Ax - b + 2\,\mathrm{diag}(\tilde{\lambda})x = 10^{-12}. \begin{pmatrix} -0.0013 \\ 0.0007 \\ 0.0066 \\ -0.0036 \\ -0.4334 \\ 0.4405 \\ 0.0027 \\ -0.0089 \\ -0.0120 \\ -0.0089 \\ 0 \\ 0 \end{pmatrix}$$

$$g(x) = 10^3. \begin{pmatrix} -0.0017 \\ -0.0002 \\ 0.0000 \\ -0.0000 \\ 0.0000 \\ -3.2844 \end{pmatrix}$$

Inner iterations:

- *Initialization*
  - number of CG iterations: 13
- *1. iteration*
  - number of MPRGP iterations: 22
  - number of CG iterations: 18
- *2. iteration*
  - number of MPRGP iterations: 9
  - number of CG iterations: 11

■

9 Conclusion

In this thesis, we used observations from simple algorithms to construct new very effective algorithm for solving problem of minimizing of quadratic function with separated quadratic constraints. We call it PDP. It represent a new way how to use Dual problem and projection to boundary of a set - it uses Inverse Dual problem to find corresponding update of Lagrange multipliers. It is probably the best update of Lagrange multiplier of previous iteration.

First numerical tests imply good convergence, but proof was not constructed yet. Also precondition can improve number of inner CG and MPRGP iterations.

In fact, contact problems imply minimizing problems with quadratic constraints, moreover linear equalities and inequalities. That is the reason, why PDP algorithm is useless in these cases. It has to be modified, probably using classic MPRGP algorithm.

10 References

[1] V. Vondrák *Numerical analysis I* Syllabus VSB-TU Ostrava

[2] Z. Dostál *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities* Springer, 2009.

[3] Z. Dostál, R. Kučera *An optimal algorithm for minimization of quadratic functions with bounded spectrum subject to separable convex inequality and linear equality constraints* MSM6198910027.

[4] R. Kučera *Minimizing quadratic functions with separable quadratic constraints* Optimization methods and software. 2007, vol. 22, issue 3, p. 453-467.

[5] R. Kučera *Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints* SIAM J. Optim., Vol. 19, No. 2, pp. 846-862

[6] J. Haslinger, R. Kučera, and Z. Dostál *An algorithm for the numerical realization of 3D contact problems with Coulomb friction* J. Comput. Appl. Math., 164-165 (2004), pp. 387–408.

Appended CD includes these folders with matlab functions:

- `Chapter 3`
  - algorithm for generating figures in Chapter 3
- `Chapter 4`
  - `chapter4/cg` - implementation of CG algorithm
  - `chapter4/mprgp` - implementation of MPRGP algorithm
- `Chapter 5`
  - `chapter5/dual problem` - figures in Section 5.2
  - `chapter5/invert dual problem` - usage of Invert Dual problem
- `Chapter 6`
  - `chapter6/constant update lagrange` - constant update of Lagrange multipliers
  - `chapter6/sequence` - sequence of Lagrange multipliers
- `Chapter 7`
  - `chapter7/linear update` - Linear update of Lagrange multipliers
  - `chapter7/adaptive linear update` - Adaptive linear update of Lagrange multipliers
  - `chapter7/bisection` - Bisection algorithm
- `Chapter 8`
  - `chapter8/pdp_eq` - implementation of PDP algorithm for Equality problem
  - `chapter8/pdp_ineq` - implementation of PDP algorithm for Inequality problem