

# Interior Point Methods applied to Quadratic Programming\*

Daniel Bergmann<sup>1</sup>

**Abstract**—Describe in a few sentences what the paper is about and why it is interesting to read it.

## I. INTRODUCTION

**TODO!**Some general introducing sentences about the topic, motivation and relevance of problem/algorithm.

In this paper we give an introduction to the results presented in paper(s) [?].

We present the problem statement (optimization problem) the main results/algorithms, discuss the underlying ideas and illustrate the results by numerical simulations.

Notation. Define notation.

## II. PROBLEM STATEMENT AND BACKGROUND

For theoretical discussions, we consider the convex constrained optimization problem

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m. \\ & && A_{\text{eq}}x = b_{\text{eq}}. \end{aligned} \quad (1)$$

with  $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$  convex and twice differentiable,  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $i = 1, \dots, m$  convex and twice **TODO!**differentiable,  $A_{\text{eq}} \in \mathbb{R}^{n \times p}$ ,  $b_{\text{eq}} \in \mathbb{R}^p$  with equality and inequality constraints. Moreover, we give a MATLAB-implementation of a primal-dual interiorpoint method for a convex quadratic optimization problem. Quadratic problems are a subclass of (1) and denote as

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) = \frac{1}{2}x^T Qx + c^T x \\ & \text{subject to} && A_{\leq}x - b_{\leq} \leq 0, \quad A_{\leq} \in \mathbb{R}^{m \times n}, b_{\leq} \in \mathbb{R}^m \\ & && A_{\text{eq}}x - b_{\text{eq}} = 0, \quad A_{\text{eq}} \in \mathbb{R}^{p \times n}, b_{\text{eq}} \in \mathbb{R}^p \end{aligned} \quad (2)$$

with matrices  $0 \prec Q \in \mathbb{R}^{n \times n}$ ,  $c \in \mathbb{R}^n$ .

## III. MAIN RESULTS

**TODO!**define dual problem

### A. Concept of Barrier Methods

Convex optimization Problems with no inequality constraints can be solved efficiently by using Newton's method. If inequality constraints are involved, Newton's method can not guarantee feasibility of a solution. It is hence desirable, to transform an inequality-constrained optimization problem into a only equality-constrained one. Therefore, we move the

inequality constraints implicitly to the objective function. **TODO!**improve sentence **TODO!**A simple and also precise way to do this, evaluate an indicator function

$$I_{-}(x) := \begin{cases} 0 & \text{for } u \neq 0 \\ \infty & \text{for } u > 0 \end{cases} \quad (3)$$

on the values of the inequality constraints  $f_i, i = 1, \dots, m$ . Then, the optimization Problem has the shape

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) + \sum_{i=1}^m I_{-}(f_i(x)) \\ & \text{subject to} && A_{\text{eq}}x - b_{\text{eq}} = 0. \end{aligned} \quad (4)$$

This problem is an equivalent to (1) and has no inequality constraints. However, it is clearly neither convex nor continuous (and hence not differentiable). Since we need these properties to solve the optimization problem computationally, we approximate the indicator function  $I_{-}$  by the function

$$\hat{I}_{-}(u) = \begin{cases} \frac{1}{t} \log(-u) & \text{for } u < 0, \\ \infty & \text{for } u \geq 0, \end{cases} \quad (5)$$

The parameter  $t > 0$  sets the approximation's accuracy. The higher  $t$  is, the better the indicator function is approximated. By replacing the Indicator functions by  $\hat{I}_{-}$ , we obtain an approximation

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) - \sum_{i=1}^m \frac{1}{t} \log(-f_i(x)) \\ & \text{subject to} && A_{\text{eq}}x - b_{\text{eq}} = 0 \end{aligned} \quad (6)$$

of problem (1).

Note, that  $\frac{1}{t} \log(-u)$  is convex, increasing in  $u$ , and differentiable on the feasible set. Hence the entire function  $\sum_{i=1}^m \hat{I}_{-}(f_i(x))$  is convex and (6) is a convex Problem with differentiable objective function. These properties allow us to solve (6) computationally. We call an optimal point  $x^*(t)$  of (6) with parameter  $t$  a central point and a solution to its dual problem  $(\lambda^*(t), \nu^*(t))$  a dual central point. The set of (dual) solutions of (6) for all  $t > 0$  we call the (dual) central path. One can show, that solutions  $(x^*(t), \lambda^*(t), \nu^*(t))$  of (6) converge to the solution  $(x^*, \lambda^*, \nu^*)$  of (1) for  $t \rightarrow 0$ . The proof is shown in [2].

### B. Measure for the Approximation's quality

An immediately arising question is, what conclusions about the solution  $(x^*, \lambda^*, \nu^*)$  of (1) can be drawn from a knowing a solution of (6) for a certain  $t > 0$  about the value  $f_0(x^*(t))$  of a central point  $x^*(t)$ , compared with the optimal value  $p^*$  of the original problem. By **TODO!**show

\*Project within the course Convex Optimization, University of Stuttgart, July 8, 2020.

<sup>1</sup>Daniel Bergmann is a student of the Bachelor study program Mechatronics, University of Stuttgart, st108500@stud.uni-stuttgart.de

arguments?

For compactness, we denote the barrier term of a the approximated problem as

$$\phi(x) = - \sum_{i=1}^m \log(-f_i(x)),$$

and keep in mind, that its Jacobian and Hessian are

$$\begin{aligned} \nabla \phi(x) &= \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x), \\ \nabla^2 \phi(x) &= \sum_{i=1}^m \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla^2 f_i(x). \end{aligned}$$

For the sake of simplifying notation, throughout this section we consider the problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & t f_0(x) + \phi(x) \\ \text{subject to} \quad & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned} \quad (7)$$

that is obtained by multiplying the objective in (6) with  $t > 0$ . The original and the obtained problem are equivalent. For an arbitrary central point  $x^*(t)$  we know, that  $x^*(t)$  is a strictly feasible point of (1). Since  $x^*(t)$  solves (7), there exists  $\hat{\nu} \in \mathbb{R}^p$ , such that **TODO!** define  $L_t$  as gradient for approx problem

$$\begin{aligned} \nabla L_t(x^*(t), \hat{\nu}) &= t \nabla f_0(x^*(t)) + \nabla \phi(x^*(t)) + A_{\text{eq}}^T \hat{\nu} \quad (8) \\ &= t \nabla f_0(x^*(t)) \\ &\quad + \sum_{i=1}^m \frac{1}{-f_i(x^*(t))} \nabla f_i(x^*(t)) + A_{\text{eq}}^T \hat{\nu}. \quad (9) \\ &= 0. \quad (10) \end{aligned}$$

holds. We keep in mind, that  $x^*(t)$  minim Using this insight, we know that there exists a dual feasible point  $(x^*(t), \lambda^*(t), \nu^*(t))$  of the original problem (1). In particular, we choose

$$\lambda^*(t) = - \frac{1}{t f_i(x^*(t))} \text{ for } i = 1, \dots, m, \quad \nu^*(t) = \frac{\hat{\nu}}{t}.$$

Here,  $\lambda^*(t) > 0$  follows from  $f_i(x^*) < 0$  for all  $i = 1, \dots, m$  since  $x^*$  is strictly feasible.

Note, that (10) is the derivative of the Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i^*(t) f_i(x) + \nu^*(t)^T (A_{\text{eq}} x^*(t) - b_{\text{eq}})$$

divided by  $t > 0$  of the original problem. The Lagrangian is convex in the first coordinate, hence we infer that  $x^*(t)$  minimizes the Lagrangian of the original problem for any fixed  $(\lambda, \nu)$ . For the dual function of the original Problem, we obtain

$$\begin{aligned} g(\lambda^*(t), \nu^*(t)) &= f_0(x^*(t)) + \sum_{i=1}^m \lambda_i^*(t) f_i(x^*(t)) + \nu^*(t)^T (A_{\text{eq}} x^*(t) - b_{\text{eq}}) \\ &= f_0(x^*(t)) - \frac{m}{t}. \end{aligned} \quad (11)$$

$$= f_0(x^*(t)) - \frac{m}{t}. \quad (12)$$

The second of the three summands adds up to  $m \cdot 1$ , because by the particular choice of  $\lambda^*(t)$  fractions cancel out. The last summand equals zero, since  $A_{\text{eq}} x^*(t) - b_{\text{eq}} = 0$ .

By weak duality, this means, that the optimum  $x^*(t)$  approximated problem (6) has an objective value  $f_0(x^*(t))$  that is maximally by  $\frac{m}{t}$  larger (and hence worse) than the real optimal value  $p^*$  of the original problem. Thus, one can theoretically force a desired bound on the suboptimality  $\epsilon > 0$  by just choosing  $t$  large enough, in particular  $t := \frac{m}{\epsilon}$ . However, just solving (6) with a large choice of  $t$  does not work out in general, since numerical issues can make convergence of Newton's Method dependent on the choice of the initial point  $x_0$ .

### C. Algorithmic Use of the Barrier Concept

As already mentioned in section ??, one can not in general solve (6) without a good guess at the initial value  $x_0$ . So how to make use of the barrier concept? The idea of interior methods is to find points along the problem's central path. Two methods are introduced in the following. Emphasis of the explanations as well as the implementation in MATLAB will be on the Primal-Dual Interior Point Method.

1) *Interior Point Method with Full Newton Search:* As mentioned before, for large  $t$  a good initial point  $x_0$ , meaning an initial point that is not far away from the actual minimum of (1) is crucial for avoiding large numerical errors. This can be achieved by starting with optimization of (6) for small  $t = t_1$ , which leads to a rather bad approximation of the original problem, but also to better numerical behavior. After finding  $x^*(t_1)$  via Newton's method,  $t$  is increased to  $t = t_2 > t_1$  by a certain rate and (6) is solved again with parameter  $t = t_2$ , with choice  $x_0 = x^*(t_1)$  for the initial point. For step  $n$  of the algorithm call finding  $x^*(t)$  the centering, and updating and updating  $t$  and  $x^*(t)$  an outer iteration or centering point and a iteration of the newton algorithm within the centering step an inner iteration. The whole procedure is written in Algorithm 1.

---

#### Algorithm 1: Interior Point Method with full Newton search

---

**Result:**  $x^*(t)$ , approximate solution of (1) with

$$f_0(x^*(t)) - p^* < \frac{m}{t}$$

initialization: Matrices  $0 \prec Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$ .

defining the objective function, matrices

$A_{\text{eq}}, b_{\text{eq}}, A_{\leq}, b_{\leq}$  defining constraints, initial point  $x$ ,

initial approximation parameter  $t > 0$ , rate for

increasing approx. param.  $\mu > 1$  tolerance  $\epsilon$

**TODO!** dimensions of constr. matrices

**while**  $\frac{m}{t} \geq \epsilon$  **do**

    Compute  $x^*(t)$  by solving (6) via Newton's Method, starting at  $x$ ;

    Update  $x := x^*(t)$ ;

    Increase  $t$  by  $t := \mu t$

**end**

---

2) *Primal-Dual Interior Point Method*: Like the previously introduced algorithm, the Primal-Dual Interior Point method uses the barrier concept to handle inequality constraints. It is motivated by the following idea. Since the points generated by each outer iteration converge to the desired optimum on the central path, one does not gain much advantage by computing the central points with a high level of accuracy. So many newton-steps are computed, without improving the convergence towards the optimum value of (1). Hence, it would be useful to reduce the accuracy of each outer iteration as much as possible, without losing convergence to the optimum. Therefore, in this method only one newton step will be computed for each parameter  $t$  in the approximated problem (6). Additionally, the Newton step is computed differently. While in the search directions are computing only considering the primal problem, in the Primal-Dual Method we also take the dual problem of

**TODO!** write dual problem

problem (6) into account. In particular Newton's method is applied to a system of residual terms, that have to equal all zero by the KKT-conditions, here presented like in [2].

*Theorem 1 (KKT-Conditions for convex Problems)*: For a convex Optimization Problem (1), the following conditions on a primal-dual point  $(x, \lambda, \nu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$  are necessary and sufficient for  $x$  being a solution to the primal problem and  $(\lambda, \nu)$  being a solution to the dual problem:

$$f_i(x) \leq 0, \quad \text{for } i = 1, \dots, m \quad (13a)$$

$$A_{\text{eq}}x - b_{\text{eq}} = 0 \quad (13b)$$

$$\lambda_i \geq 0, \quad \text{for } i = 1, \dots, m \quad (13c)$$

$$\lambda_i f_i(x) = 0, \quad \text{for } i = 1, \dots, m \quad (13d)$$

$$\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x) = 0. \quad (13e)$$

Stacked in one vector, this yields

$$\begin{aligned} F(x, \lambda, \nu) &:= r_\mu(x, \lambda, \nu) = \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \\ &= \begin{pmatrix} \nabla f_0(x) + Jf(x)^T \lambda + A_{\text{eq}}^T \nu \\ -\text{diag}(\lambda)f(x) - \mu \mathbf{1} \\ A_{\text{eq}}x - b_{\text{eq}} \end{pmatrix} = 0. \end{aligned} \quad (14)$$

to apply Newton on. For formulation of the linear Newton equality, we also compute the jacobian

$$\frac{d(r_\mu(x, \lambda, \nu))}{d(x, \lambda, \nu)^T} = \begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x) & A_{\text{eq}}^T \\ -\text{diag}(\lambda)Jf(x) & -\text{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{pmatrix} \quad (15)$$

of the residual. Consequently, the Newton equality for finding the search direction  $(\Delta x, \Delta \lambda, \Delta \nu)$  in each newton step

is exactly

$$\begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x) & A_{\text{eq}}^T \\ -\text{diag}(\lambda)Jf(x) & -\text{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \quad (16)$$

(**TODO!** equality should fit in col.)

Unfortunately, adding the obtained step direction  $(\Delta x, \Delta \lambda, \Delta \nu)$  to  $(x, \lambda, \nu)$ , does not in general yield a feasible point. Therefore we compute a suitable step-size  $s^*$  via a backtracking-linesearch, such that a certain decrease of the residual and feasibility is guaranteed for the next iteration point

$$\begin{pmatrix} x^+ \\ \lambda^+ \\ \nu^+ \end{pmatrix} = \begin{pmatrix} x \\ \lambda \\ \nu \end{pmatrix} + s^* \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta \nu \end{pmatrix}.$$

The detailed procedure of the backtracking linesearch is displayed in Algorithm 2.

---

**Algorithm 2:** Backtracking linesearch

---

**Result:** Stepsize  $s^*$ , s.t.  $\lambda^+ > 0$ ,  $f(x^+) < 0$  and  $r_\mu$  decreases by certain amount.

**Data:** Problem matrices, current  $x, \lambda, \nu$ , Newton direction  $\Delta x, \Delta \lambda, \Delta \nu$ , barrier parameter  $\mu$ , backtracking parameters  $\alpha \geq 0, \beta \in (0, 1)$ .

Initial step-size

$$s_{\text{max}} := \min\{1, \min_{i|\Delta \lambda_i < 0} -\lambda_i / \Delta \lambda_i\}$$

compute  $r_\mu(x, \lambda, \nu)$ ;

$s = s_{\text{max}}$ ;

$\text{found} = \text{false}$ ;

**while**  $\text{found} == \text{false}$  **do**

    set  $s = \beta s$ ;

    compute  $(x^+, \lambda^+, \nu^+)$ ;

    compute  $r_\mu(x^+, \lambda^+, \nu^+)$  and  $f(x^+)$ ;

**if**  $f(x^+) < 0$  **and**

$\|r_\mu(x^+, \lambda^+, \nu^+)\| \leq (1 - \alpha s) \|r_\mu(x, \lambda, \nu)\|$  **then**

        |  $\text{found} = \text{true}$

**end**

**end**

---

Finally, we can present the entire algorithm of the Primal-Dual Method.

*Remark 1:* If a strictly feasible primal variable  $x \in \mathbb{R}^n$  is known,  $\lambda = -1/f_i(x) \geq 0, \nu = 0$  is always a valid choice for the initial dual variables.

**TODO!** explain notation of  $f$  without index

D. How to find a feasible initial point

The Algorithms 1 and 3 both need a strictly feasible initial point to start. Since such a point is in general not trivial to find, one can formulate the search for the initial point as another convex optimization problem, that is easier to solve

---

**Algorithm 3:** Primal-Dual Interior Point Method

---

**Result:** approximate optimizer  $\hat{x}^*$ , approx. opt. value  $\hat{p}^*$ , approx. dual optimizer  $(\hat{\lambda}^*, \hat{\nu}^*)$ , surrogate duality gap  $\hat{\eta}^*$  as measure of optimality  
**Data:** Problem matrices, primal-dual initial point  $(x, \lambda, \nu)$  with  $f_i(x) < 0$  for all  $i = 1, \dots, m$ ,  $\lambda > 0, \nu \in \mathbb{R}^p$  (initial point strictly feasible), reduction factor  $\gamma \in (0, 1)$ , tolerances  $\epsilon_{\text{feas}}, \epsilon_{\text{opt}} > 0$

Initialization;

determine problem dimensions  $n, m, p$ ;

set  $\text{found} = \text{false}$ ;

**while**  $\text{found} == \text{false}$  **do**

    compute surrogate duality gap:  $\hat{\eta} = -f(x)^T \lambda$ ;  
    compute KKT residual vector  $r_\mu(x, \lambda, \nu)$  via (14);  
    compute search direction  $(\Delta x, \Delta \lambda, \Delta \nu)$  by solving (16);  
    determine suitable step size  $s$  via backtracking algorithm 2;  
    update current primal and dual points:  
     $(x, \lambda, \nu) = (x, \lambda, \nu) + (\Delta x, \Delta \lambda, \Delta \nu)$ ;

**end**

return  $\hat{x}^* = x, \hat{p}^* = f_0(\hat{x}^*), \hat{\lambda}^* = \lambda, \hat{\nu}^* = \nu, \hat{\eta}^* = \hat{\eta}$ ;

---

than the original one. For problem (1) one way to implement this, is solving

$$\begin{aligned} & \underset{x}{\text{minimize}} && s \\ & \text{subject to} && f_i(x) \leq s, \quad i = 1, \dots, m \\ & && A_{\text{eq}}x - b_{\text{eq}} = 0, \end{aligned} \quad (17)$$

via Newton's method. If a point with optimal value strictly smaller than zero for (17) is found, then this point is strictly feasible. Solving such a first, more simple problem is called a Phase I problem. More examples of such problems can be found in [2].

#### E. Complexity Analysis Barrier Method

In this section we discuss the time complexity of the barrier method, meaning the total number of newton steps needed to solve (1). An upper bound of these iterations can be proven for problems with objectives that are self-concordant. While Linear and quadratic functions satisfy selfconcordance in general, any other convex optimization problem can be rewritten as a self-concordant one, so this condition is not very restrictive. The upper bound

$$\frac{f(x) - p^*}{\gamma} + c \quad (18)$$

on the maximal number of newton iterations that is needed to obtain a certain accuracy, determined by the initial point  $x$ , the optimal value  $p^*$ ,  $\gamma$  a constant dependent on the backtracking parameters and  $c$  dependent on the accuracy tolerance. One can show via reasoning with the Lagrangian, **TODO!**detailed?

that this bound holds uniformly for any parameter  $t$  for all problems (6). Since there are exactly

$$\left\lceil \frac{\log(m/\epsilon t_0)}{\log \mu} \right\rceil$$

outer steps necessary to solve (6) with initial parameter  $t = t_0$  and accuracy tolerance  $\epsilon$ , the entire barrier method needs maximally

$$N = \dots$$

**TODO!**

iterations to yield a result with a suboptimality of  $\epsilon$  or smaller.

#### F. Newton's Method

**TODO!**move to beginning Newton's method is an iterative process to solve nonlinear equality systems

$$F(x) = 0 \quad (19)$$

for a differentiable map  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . The idea of this algorithm is as follows: At a given point  $x_k$ , the zero of the linear approximation of  $F$  around  $x_k$  is computed. This point is chosen as the next iterate  $x_{k+1}$ . In particular, a linear approximation of  $F$  in  $x_k$  is defined as

$$L(x) := F(x_k) + JF(x_k)(x - x_k) \text{ for } x \in \mathbb{R}^n, \quad (20)$$

where  $JF(x_k)$  is the Jacobian of  $F$  at the point  $x_k$ . If  $JF(x_k)$  invertible, the point  $\tilde{x}$  with  $L(\tilde{x}) = 0$  is exactly the solution of the linear equality  $JF(x_k)x = -F(x_k)$ . Technical conditions and proofs about convergence rates of Newton's method can be found in [1].

For the purpose of optimizing a convex, twice differentiable objective function  $f_0$  we want to find a zero of the gradient  $\nabla f_0$ . Therefore we can apply the Newton Method to solve the non-linear equation

$$F(x) := \begin{pmatrix} \nabla f_0(x) \\ g(x) \end{pmatrix} = 0 \quad \text{with } g(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_p(x) \end{pmatrix}$$

. By convexity, satisfying  $\nabla f_0(x^*) = 0$  is not only necessary, but also sufficient for  $x^*$  to be a global minimum of  $f_0$ .

## IV. EXAMPLES

**TODO!**Show and discuss simulation examples etc....

## V. CONCLUSIONS

**TODO!**

Summarize the main points (with more details than in the preceding introduction). The paper should not be between 4 and 8 pages.

## APPENDIX

Add for example your Matlab code here. (Code should be nicely formatted and documented).

Appendixes should appear before the acknowledgment.

## ACKNOWLEDGMENT

### REFERENCES

- [1] Carsten Scherer *Vorlesungsskript Einführung in die Optimierung* 2019: Lehrstuhl für Mathematische Systemtheorie, Universität Stuttgart.
- [2] Stephen Boyd, Lieven Vandenberghe *Convex Optimization* 2004: Cambridge University Press.