

Modified Augmented Lagrangian Method for the minimization of functions with quadratic penalty terms

Martin Neuenhofen

November 8, 2018

Abstract

The Augmented Lagrangian Method (ALM) is an iterative method for the solution of equality-constrained non-linear programming problems. In contrast to the quadratic penalty method, the ALM can satisfy equality constraints in an exact way. Further, ALM is claimed to converge in less iterations, indicating that it is superior in approach to a quadratic penalty method.

It is referred to as an advantage that the ALM solves equality constraints in an exact way, meaning that the penalty parameter does not need to go to infinity in order to yield accurate feasibility for the constraints. However, we sometimes actually want to minimize an unconstrained problem that has large quadratic penalty terms. In these situations it is unclear how the ALM could be utilized in the correct way. This paper presents the answer: We derive a modified version of the ALM that is also suitable for solving functions with large quadratic penalty terms.

1 Introduction

We consider numerical methods for the minimization of three types of optimization problems. These problems are stated below.

Merit problem We consider the unconstrained minimization of a function with quadratic penalty terms:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Phi(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2 \cdot \omega} \cdot \|c(\mathbf{x})\|_2^2, \quad (1)$$

where $m, n \in \mathbb{N}$, a very small number $\omega \in \mathbb{R}^+ \setminus \{0\}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are given. Notice that the statement of this problem makes sense regardless whether $m \leq n$ or $m > n$. This is an unconstrained optimization problem. Under suitable assumptions on Φ one can show well-posedness.

Penalty problem Though redundant, it helps the presentation to define also the following optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Psi(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x})\|_2^2 \quad (2)$$

where $\tilde{\omega} \in \mathbb{R}^+ \setminus \{0\}$ is a small number. We think of $\tilde{\omega}$ as a number that is small but that is still larger than ω . E.g., it could be $\omega = 10^{-10}$ and $\tilde{\omega} = 10^{-2}$. Given a suitable initial guess, penalty problems can be considered solvable with a conventional Newton-type method whereas merit problems can be considered intractable because ω is usually so small that their scaling is too bad.

Constrained problem We consider the following constrained problem.

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (3a)$$

$$\text{s.t.} \quad c(\mathbf{x}) = \mathbf{0} \quad (3b)$$

If a solution \mathbf{x} of $c(\mathbf{x}) = \mathbf{0}$ exists then again under suitable assumptions one can show well-posedness. Further, in this case solutions of problem (1) converge to solutions of (3) for decreasing positive values of ω . Since in practice we only solve optimization problems up to some tolerance $\text{tol} > 0$ for the residual of the Karush-Kuhn-Tucker (KKT) conditions, we can consider (3) as a sub-class of problem (1). We do so because when choosing $\omega \in \mathcal{O}(\text{tol})$, a solution of (1) is also a sufficiently accurate solution of (3), cf. [7].

The Lagrangian of (3) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) - \boldsymbol{\lambda}^\top \cdot c(\mathbf{x}). \quad (4)$$

Outline In this paper we consider three local solution methods:

1. The penalty method: It is an iterative method for solving (1) by solving a sequence of (2) with decreasing values for $\tilde{\omega}$. Since from a numerical perspective (3) is a special case of (1), the penalty method can also be used to solve (3).
2. The ALM: It is a method for solving (3) by solving a finite sequence of (2) with a value for $\tilde{\omega}$ that remains moderate, e.g. 10^{-2} , throughout the sequence.
3. A modified ALM: It is a method for solving (1) by solving a finite sequence of (2) with a value for $\tilde{\omega}$ that remains small, e.g. 10^{-2} , throughout the sequence.

The first and second method are well-known from the literature. The third approach is a novelty that we propose in this paper. The motivation for using this method is that it can solve (1) by solving a sequence of (2). But in contrast to the penalty method it uses moderate values of $\tilde{\omega}$, as is the case of the conventional ALM. This is beneficial, as it is claimed that ALM usually converges in less iterations than the penalty method [6].

2 Summary of penalty method and ALM

We review the two methods from the literature. We work out their computational framework and give pseudo-code that can be used for a practical implementation.

2.1 Penalty method

The penalty method solves a sequence of (2), where $\tilde{\omega}$ iteratively decreases until it finally reaches ω . The solution of each problem (2) is used as initial guess to solve the subsequent problem (2) with a decreased value of $\tilde{\omega}$. While decreasing values of $\tilde{\omega}$ make the problem (2) more difficult to solve, the initial guesses obtained from the former solution yield a compensating effect. In particular, the found minimizers become more and more accurate initial guesses for the subsequent penalty problem, altogether resulting in a potentially efficient approach for solving (1).

The method can be described with the following pseudocode.

```

1: Given is:  $f, c, \omega > 0, 0 < \theta_\omega < 1, \tilde{\omega}_1 > \omega, \mathbf{x}_0 \in \mathbb{R}^n$ 
2: for  $k = 1, 2, 3, \dots$  do
3:   Solve (2) for  $\tilde{\omega} = \tilde{\omega}_k$  with initial guess  $\mathbf{x}_{k-1}$ .
4:   Write the solution of (2) into the vector  $\mathbf{x}_k \in \mathbb{R}^n$ .
5:   if  $\tilde{\omega}_k == \omega$  then
6:     return  $\mathbf{x}_k$ 
7:   else
8:     Update  $\tilde{\omega}_{k+1} := \max\{\theta_\omega \cdot \tilde{\omega}_k, \omega\}$ .
9:   end if
10: end for

```

The Newton iteration within the penalty method In order to solve (2), typically a Newton-type method is used. In the simplest approach, the Newton type method needs two things:

- Root function: This is a multivariate function that has as many output as input dimensions. Its root, computed by Newton's method, determines a local minimizer.
- A merit-function: This is a function for which the Newton step (subject to suitable modifications) is a descent direction. It is needed within the line search, which in turn is needed for globalization of the Newton method.

Problem (2) has the Karush-Kuhn Tucker (KKT) [5] condition

$$\nabla \Psi(\mathbf{x}) \equiv \nabla f(\mathbf{x}) - \nabla c(\mathbf{x}) \cdot \left(\frac{-1}{\tilde{\omega}} \cdot c(\mathbf{x}) \right) = \mathbf{0}.$$

As discussed in [1], for sake of well-posedness it is advantageous to rather consider the following equivalent set of equations, where $\tilde{\lambda} = -1/\tilde{\omega} \cdot c(\mathbf{x})$ is a

substituted auxiliary variable.

$$F(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) := \begin{pmatrix} \nabla f(\mathbf{x}) - \nabla c(\mathbf{x}) \cdot \tilde{\boldsymbol{\lambda}} \\ c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}} \end{pmatrix} = \mathbf{0}$$

The Jacobian is

$$DF(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) = \begin{bmatrix} \nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) & -\nabla c(\mathbf{x}) \\ \nabla c(\mathbf{x})^\top & \tilde{\omega} \cdot \mathbf{I} \end{bmatrix}.$$

A symmetric indefinite system is obtained by multiplying the second column with -1 . Further, replacing $\nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}, \tilde{\boldsymbol{\lambda}})$ with a symmetric matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ we obtain the symmetric Newton system

$$\underbrace{\begin{bmatrix} \mathbf{B} & \nabla c(\mathbf{x}) \\ \nabla c(\mathbf{x})^\top & -\tilde{\omega} \cdot \mathbf{I} \end{bmatrix}}_{=: \mathbf{K}} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ -\Delta \tilde{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) \\ c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}} \end{pmatrix}. \quad (5)$$

We use the short-hands $\mathbf{w} := (\mathbf{x}^\top, \tilde{\boldsymbol{\lambda}}^\top)^\top$, $\Delta \mathbf{w} := (\Delta \mathbf{x}^\top, \Delta \tilde{\boldsymbol{\lambda}}^\top)^\top \in \mathbb{R}^{n+m}$. The symmetric Newton matrix we call \mathbf{K} .

We say that \mathbf{B} is chosen *suitable* when \mathbf{K} has n strictly positive and m strictly negative eigenvalues. For instance, \mathbf{B} is suitable when it is positive definite. This can be achieved in the simplest way by adding a multiple of the identity to $\nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}, \tilde{\boldsymbol{\lambda}})$. More sophisticated is the use of a symmetric indefinite LDL-factorization with a subsequent modification of the diagonal matrix. For details we refer to [4, Sec. 4.1]. We assume in the following that \mathbf{B} is suitable.

Finally, we need the merit function. As shown in [4], $\Delta \mathbf{w}$ is a descent direction for the following merit function for arbitrary values of $\nu \in \mathbb{R}^+ \setminus \{0\}$:

$$M(\mathbf{w}) := f(\mathbf{x}) + \frac{\tilde{\omega}}{2} \cdot \|c(\mathbf{x})\|_2^2 + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}}\|_2^2. \quad (6)$$

For the purpose of an overview and a concrete method, below we state the solution procedure for (2), inherited in line 2 of the penalty method, below.

- 1: Given is: $f, c, \tilde{\omega} > 0$, initial guess \mathbf{x} , $\text{tol} > 0$.
- 2: $\tilde{\boldsymbol{\lambda}} := -1/\tilde{\omega} \cdot c(\mathbf{x})$
- 3: **while** $\|F(\mathbf{w})\|_2 > \text{tol}$ **do**
- 4: Choose \mathbf{B} suitable; e.g. $\mathbf{B} := \nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) + \xi \cdot \mathbf{I} \succ \mathbf{0}$ with $\xi \geq 0$ sufficiently large.
- 5: Solve the linear system (5).
- 6: Find $\alpha \in (0, 1]$ that satisfies a sufficient descent condition for the merit function in (6).
- 7: Update $\mathbf{w} := \mathbf{w} + \alpha \cdot \Delta \mathbf{w}$.
- 8: **end while**

2.2 ALM

The ALM solves a sequence of (2), where usually $\tilde{\omega}$ remains bounded below by some moderate number. The actual requirements on the minimum value for $\tilde{\omega}$ remain technical and are discussed to in [6] and the references therein.

In order for minimizers of (2) to solve (3), the function f in Ψ is replaced by $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$, where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is computed suitable through an outer iteration scheme. For better reference we introduce an outer iteration index $k \in \mathbb{N}_0$ that we add as footnote to $\boldsymbol{\lambda}_k$.

For the k -th outer iteration we write out the particular problem of kind (2) that is solved within ALM:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Psi_k(\mathbf{x}) := \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}_k) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x})\|_2^2 \quad (7)$$

In this problem, $\boldsymbol{\lambda}_k$ is a fixed given vector from the outer iteration k . $\tilde{\omega}$ is assumed to be sufficiently small so that the problem is bounded.

As we did for the penalty problem solved within the penalty method, from the solution of (7) we obtain a tuple $(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) \in \mathbb{R}^n \times \mathbb{R}^m$, where $\tilde{\boldsymbol{\lambda}} \approx -1/\tilde{\omega} \cdot c(\mathbf{x})$, where \approx accounts for the inaccuracy in terms of the tolerance $\text{tol} > 0$.

The algorithm of ALM goes as follows.

- 1: Given is: $f, c, 0 < \theta_{\tilde{\omega}} < 1, 0 < \theta_{\lambda} < 1, \tilde{\omega}_1 > 0, \mathbf{x}_0 \in \mathbb{R}^n$
- 2: $\tilde{\boldsymbol{\lambda}}_0 := -1/\tilde{\omega}_1 \cdot c(\mathbf{x}_0), \boldsymbol{\lambda}_0 := \mathbf{0}$
- 3: **for** $k = 1, 2, 3, \dots$ **do**
- 4: Solve (7) for $\tilde{\omega} = \tilde{\omega}_k$ with initial guess $(\mathbf{x}_{k-1}, \tilde{\boldsymbol{\lambda}}_{k-1})$.
- 5: Write the solution of (7) into the tuple $(\mathbf{x}_k, \boldsymbol{\lambda}_k) \in \mathbb{R}^n \times \mathbb{R}^m$.
- 6: **if** $\|\nabla_{\mathbf{x}\lambda} \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k)\|_2 \leq \text{tol}$ **then**
- 7: **return** \mathbf{x}_k
- 8: **else**
- 9: // Update $\boldsymbol{\lambda}$ and $\tilde{\omega}$
- 10: **if** $\|c(\mathbf{x}_k)\|_2 \leq \theta_{\lambda} \cdot \min_{j=0, \dots, k-1} \{\|c(\mathbf{x}_j)\|_2\}$ **then**
- 11: $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}_k, \tilde{\omega}_{k+1} := \tilde{\omega}_k / \sqrt{\theta_{\omega}}$
- 12: $\tilde{\boldsymbol{\lambda}}_k := \mathbf{0}$
- 13: **else**
- 14: $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k, \tilde{\omega}_{k+1} = \theta_{\omega} \cdot \tilde{\omega}_k$
- 15: **end if**
- 16: **end if**
- 17: **end for**

This algorithm, apart from constant parameters $\theta_{\omega}, \theta_{\lambda}$, e.g. $\theta_{\omega} = \theta_{\lambda} = 0.1$, is equivalent to the ALM as presented in [6]. The increase of $\tilde{\omega}$ in the update in line 11 is optional.

We describe the steps of the algorithm. We start by solving problem (7). For moderate but sufficiently small values of $\tilde{\omega}$ the solution \mathbf{x}_k of this problem will be bounded and yields a small norm for $\|c(\mathbf{x}_k)\|_2$. In line 6 the KKT conditions are checked. If they are satisfied up to a tolerance $\text{tol} > 0$ then the problem is considered solved and the solution is returned. Otherwise, $\boldsymbol{\lambda}$ and $\tilde{\omega}$ are updated. This can happen in either of two ways, of which we first consider the update in line 11. ALM considers $\tilde{\boldsymbol{\lambda}}_k$ as a guess for an update of the Lagrange multiplier $\boldsymbol{\lambda}$. This can be motivated from similarities of the KKT conditions for the four problems (6),(2),(3) and (7). The update for $\boldsymbol{\lambda}$ in line 11 however requires that $\tilde{\boldsymbol{\lambda}}$ converges to zero for increasing indices k ; clearly, otherwise $\boldsymbol{\lambda}$ could never converge. For sufficiently small (but bounded

below) values of $\tilde{\omega}$ this will eventually happen under mild assumptions, cf. [6] and references therein. However, during earlier iterations the value of $\tilde{\omega}$ may be too small. Thus there is a conditional statement. If the norm of $\|c(\mathbf{x}_k)\|_2$ decreases monotonously then the whole iteration is considered convergent, i.e. $\tilde{\omega}$ is assumed to be sufficiently large. In this case the ALM update is chosen. Otherwise the iteration resorts to a penalty method (update in line 14), which converges under milder assumptions – until $\|c(\mathbf{x}_k)\|_2$ decreases again.

The reason why $\tilde{\lambda}_k$ will eventually converge to zero is due to the values of λ_k . In ALM λ_k is an iteratively improving guess for the solution of the local Lagrange-dual problem [2]. λ_k balances the constraints against the objective. Therefore, eventually for large indices of the outer iteration k the local minimizer \mathbf{x}_k of

$$f(\mathbf{x}_k) - \lambda_k^\top \cdot c(\mathbf{x}_k)$$

will just satisfy $c(\mathbf{x}_k) = \mathbf{0}$. It is identical to a local minimizer of (7), regardless whether $\tilde{\omega}$ is small or large (which is why one can even increase it). Hence, $\tilde{\lambda}_k = -1/\tilde{\omega} \cdot c(\mathbf{x}_k)$ converges to zero.

Newton system and merit function in the ALM As discussed formerly for the penalty method, in order to solve the penalty problem (7), we typically use a Newton method. This requires the following two items:

- Root function
- Merit-function

Substituting again $\tilde{\lambda} = -\tilde{\omega} \cdot c(\mathbf{x})$, the KKT conditions of (7) can be expressed equivalently as

$$F(\mathbf{x}, \tilde{\lambda}) := \begin{pmatrix} \nabla f(\mathbf{x}) - \nabla c(\mathbf{x}) \cdot (\lambda_k + \tilde{\lambda}) \\ c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\lambda} \end{pmatrix} = \mathbf{0}. \quad (8)$$

Notice that λ_k , the given fixed Lagrange multiplier estimate from the outer iteration k , appears as a constant in the root function. Using again a shorthand \mathbf{w} for the argument of F , a well-posed Newton system for the update $\Delta \mathbf{w} = (\Delta \mathbf{x}^\top, \Delta \tilde{\lambda}^\top)^\top$ is given by

$$\underbrace{\begin{bmatrix} \mathbf{B} & \nabla c(\mathbf{x}) \\ \nabla c(\mathbf{x})^\top & -\tilde{\omega} \cdot \mathbf{I} \end{bmatrix}}_{=: \mathbf{K}} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ -\Delta \tilde{\lambda} \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda_k + \tilde{\lambda}) \\ c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\lambda} \end{pmatrix},$$

where this time \mathbf{B} must be a suitable approximation of $\nabla_{\mathbf{x}\mathbf{x}}^2 \mathcal{L}(\mathbf{x}, \lambda_k + \tilde{\lambda})$. Again, with suitable we mean that \mathbf{K} must have n positive and m negative eigenvalues.

The step $\Delta \mathbf{w}$ is a descent direction for the merit function

$$M_k(\mathbf{w}) = f(\mathbf{x}) - \lambda_k^\top \cdot c(\mathbf{x}) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x})\|_2^2 + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \tilde{\omega} \cdot \tilde{\lambda}\|_2^2$$

for a fixed arbitrary $\nu > 0$ [4, Sec. 3.1]. The algorithm for the Newton iteration within the ALM is then analogous to the formerly discussed Newton iteration within the penalty method: Given \mathbf{w} , we first solve the Newton system for $\Delta \mathbf{w}$ and then perform a line search on M_k . We repeat these two steps until $\|F(\mathbf{w})\|_2 \leq \text{tol}$ is satisfied for F from (8).

3 Modified Augmented Lagrangian Method for the penalty problem

Motivation Both the penalty method and the ALM can be used to solve the constrained problem (3). In reaching for this purpose, the ALM is claimed to be more efficient than the penalty method [6]. This is for several arguments:

- In the penalty method the parameter $\tilde{\omega}$ must be chosen very small whereas in the ALM it often remains at a moderate size.
- A very small value of $\tilde{\omega}$ can result in numerical problems when solving the Newton system. Thus, penalty methods are considered less numerically robust than ALM.
- A very small value of $\tilde{\omega}$ can also result in steep valleys in the surface of the merit function. Especially for non-linear functions c this valley is curved, like the Rosenbrock "banana" function. This leads to the problem that the line search will make slower progress in the penalty method.

On the other hand, the penalty method can solve the more general problem (1). We would like to adapt ALM such that it can be used for solving this problem, too. To this end we derive a modified ALM that is suitable for solving both problems (1) and (3).

Our modified ALM works with a method parameter $\omega \in \mathbb{R}^+ \cup \{0\}$. It is a true generalization of the conventional ALM as presented above. This is meant in the following sense: For $\omega > 0$ the modified ALM solves the problem (1). If $\omega = 0$ then the modified ALM becomes equivalent to the conventional ALM and solves (3). Further, the modification is only a small change that can be easily incorporated into existing implementations of ALM.

3.1 Derivation of modified ALM

To derive the modified ALM, we proceed in two steps:

1. We write the conventional ALM as a template.
2. We then fit the merit problem into this template and simplify the resulting expressions.

The template of ALM The ALM can be applied to programming problems that can be expressed as the constrained problem

$$\min_{\mathbf{y} \in \mathbb{R}^N} \quad \tilde{f}(\mathbf{y}) \quad (9a)$$

$$\text{s.t.} \quad \tilde{c}(\mathbf{y}) = \mathbf{0}. \quad (9b)$$

Then the root-function of ALM is

$$F(\mathbf{y}, \tilde{\boldsymbol{\lambda}}) := \begin{pmatrix} \nabla_{\mathbf{y}} \tilde{f}(\mathbf{y}) - \nabla_{\mathbf{y}} \tilde{c}(\mathbf{y}) \cdot (\boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}) \\ \tilde{c}(\mathbf{y}) + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}} \end{pmatrix} = \mathbf{0}.$$

We use the iterate $\mathbf{w} := (\mathbf{y}^\top, \tilde{\boldsymbol{\lambda}}^\top)^\top$. The merit function is

$$M_k(\mathbf{w}) = \tilde{f}(\mathbf{y}) - \boldsymbol{\lambda}_k^\top \cdot \tilde{c}(\mathbf{y}) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|\tilde{c}(\mathbf{y})\|_2^2 + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|\tilde{c}(\mathbf{y}) + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}}\|_2^2$$

for $\mathbf{w} := (\mathbf{y}, \tilde{\boldsymbol{\lambda}})$. And the update formulas, as in the ALM algorithm lines 11 and 14, are:

$$\begin{aligned} \boldsymbol{\lambda}_{k+1} &:= \boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}_k, & \tilde{\omega}_{k+1} &:= \tilde{\omega}_k / \sqrt{\theta_\omega} & \text{if } \|\tilde{c}(\mathbf{y}_k)\|_2 \leq \theta_\lambda \cdot \min_{0 \leq j < k} \{\|\tilde{c}(\mathbf{y}_j)\|_2\} \\ \boldsymbol{\lambda}_{k+1} &:= \boldsymbol{\lambda}_k, & \tilde{\omega}_{k+1} &:= \theta_\omega \cdot \tilde{\omega}_k & \text{otherwise} \end{aligned}$$

Expressing Merit-Problem as Constrained Problem We can express (1) equivalently as a constrained problem by using auxiliary variables $\boldsymbol{\xi} = 1/\omega \cdot c(\mathbf{x})$:

$$\min_{\mathbf{x} \in \mathbb{R}^n, \boldsymbol{\xi} \in \mathbb{R}^m} \quad f(\mathbf{x}) + \frac{\omega}{2} \cdot \|\boldsymbol{\xi}\|_2^2 \quad (10a)$$

$$\text{s.t.} \quad c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi} = \mathbf{0} \quad (10b)$$

This problem in turn fits into the template (9) by using the following definitions:

$$\begin{aligned} \mathbf{y} &:= (\mathbf{x}^\top, \boldsymbol{\xi}^\top)^\top \in \mathbb{R}^N, & N &:= n + m \\ \tilde{f}(\mathbf{y}) &:= f(\mathbf{x}) + \frac{\omega}{2} \cdot \|\boldsymbol{\xi}\|_2^2 \\ \tilde{c}(\mathbf{y}) &:= c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi} \end{aligned}$$

From the template we obtain the root function

$$F(\mathbf{x}, \boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}}) := \begin{pmatrix} \begin{pmatrix} \nabla f(\mathbf{x}) \\ \omega \cdot \boldsymbol{\xi} \end{pmatrix} - \begin{pmatrix} \nabla c(\mathbf{x}) \\ \omega \cdot \mathbf{I} \end{pmatrix} \cdot (\boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}) \\ c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi} + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}} \end{pmatrix} = \mathbf{0} \quad (11)$$

and the merit function

$$\begin{aligned} M_k(\mathbf{x}, \boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}}) &= f(\mathbf{x}) + \frac{\omega}{2} \cdot \|\boldsymbol{\xi}\|_2^2 - \boldsymbol{\lambda}_k^\top \cdot (c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi}) \\ &+ \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi}\|_2^2 + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \boldsymbol{\xi} + \tilde{\omega} \cdot \tilde{\boldsymbol{\lambda}}\|_2^2. \quad (12) \end{aligned}$$

Elimination of ξ We propose to eliminate ξ . From the second row in F in (11) we find $\xi = \lambda_k + \tilde{\lambda}$. Inserting this for (11), we arrive at

$$F(\mathbf{x}, \tilde{\lambda}) := \begin{pmatrix} \nabla f(\mathbf{x}) - \nabla c(\mathbf{x}) \cdot (\lambda_k + \tilde{\lambda}) \\ c(\mathbf{x}) + \omega \cdot \lambda_k + (\omega + \tilde{\omega}) \cdot \tilde{\lambda} \end{pmatrix} = \mathbf{0}. \quad (13)$$

The modified symmetric Newton system is then

$$\underbrace{\begin{bmatrix} \mathbf{B} & \nabla c(\mathbf{x}) \\ \nabla c(\mathbf{x})^\top & -(\omega + \tilde{\omega}) \cdot \mathbf{I} \end{bmatrix}}_{=: \mathbf{K}} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ -\Delta \tilde{\lambda} \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda_k + \tilde{\lambda}) \\ c(\mathbf{x}) + \omega \cdot \lambda_k + (\omega + \tilde{\omega}) \cdot \tilde{\lambda} \end{pmatrix}, \quad (14)$$

where \mathbf{B} is a suitable approximation of $\nabla_{\mathbf{x}\mathbf{x}}^2 \mathcal{L}(\mathbf{x}, \lambda_k + \tilde{\lambda})$, i.e. so that \mathbf{K} has n positive and m negative eigenvalues.

We also insert the elimination of ξ into the merit function (12):

$$\begin{aligned} M_k(\mathbf{x}, \tilde{\lambda}) &= f(\mathbf{x}) + \frac{\omega}{2} \cdot \|\lambda_k + \tilde{\lambda}\|_2^2 - \lambda_k^\top \cdot (c(\mathbf{x}) + \omega \cdot \lambda_k + \omega \cdot \tilde{\lambda}) \\ &\quad + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \lambda_k + \omega \cdot \tilde{\lambda}\|_2^2 \\ &\quad + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \lambda_k + (\omega + \tilde{\omega}) \cdot \tilde{\lambda}\|_2^2 \end{aligned}$$

We can use the following identity:

$$\begin{aligned} &\frac{\omega}{2} \cdot \|\lambda_k + \tilde{\lambda}\|_2^2 - \lambda_k^\top \cdot (c(\mathbf{x}) + \underbrace{\omega \cdot \lambda_k}_{constant} + \omega \cdot \tilde{\lambda}) \\ &= \frac{\omega}{2} \cdot \|\tilde{\lambda}\|_2^2 + \omega \cdot \lambda_k^\top \cdot \tilde{\lambda} - \lambda_k^\top \cdot c(\mathbf{x}) - \omega \cdot \lambda_k^\top \cdot \tilde{\lambda} + const \\ &= \frac{\omega}{2} \cdot \|\tilde{\lambda}\|_2^2 - \lambda_k^\top \cdot c(\mathbf{x}) + const \end{aligned}$$

Using the identity, the merit function can be simplified to

$$\begin{aligned} M_k(\mathbf{x}, \tilde{\lambda}) &:= f(\mathbf{x}) + \frac{\omega}{2} \cdot \|\tilde{\lambda}\|_2^2 - \lambda_k^\top \cdot c(\mathbf{x}) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \lambda_k + \omega \cdot \tilde{\lambda}\|_2^2 \\ &\quad + \frac{\nu}{2 \cdot \tilde{\omega}} \cdot \|c(\mathbf{x}) + \omega \cdot \lambda_k + (\omega + \tilde{\omega}) \cdot \tilde{\lambda}\|_2^2. \end{aligned} \quad (15)$$

We observe the desirable property that $\omega > 0$ forces boundedness for $\tilde{\lambda}$.

Convergence of the modified ALM The modified ALM inherits all convergence properties of ALM. This is because the modified ALM is algebraically equivalent to applying the ALM to problem (10).

Modified ALM can be considered a true generalization of ALM because modified ALM solves (13) where λ_k is chosen such that in the limit $\tilde{\lambda} = \mathbf{0}$. ALM on the other hand solves (8) where λ_k is chosen such that in the limit $\tilde{\lambda} = \mathbf{0}$. For $\omega = 0$ the modified ALM becomes equivalent to ALM. For $\omega > 0$ it can also solve the problem (1), which as formerly discussed is a generalization of (3).

3.2 Algorithm of modified ALM

In summary we state the algorithm of our proposed modified ALM:

```

1: Given is:  $f, c, \omega > 0, 0 < \theta_\omega < 1, 0 < \theta_\lambda < 1, \tilde{\omega}_1 > \omega, \mathbf{x}_0 \in \mathbb{R}^n$ 
2:  $\tilde{\boldsymbol{\lambda}}_0 := -1/\tilde{\omega}_1 \cdot c(\mathbf{x}_0), \boldsymbol{\lambda}_0 := \mathbf{0}$ 
3: for  $k = 1, 2, 3, \dots$  do
4:   Solve (13) for  $\tilde{\omega} = \tilde{\omega}_k$  with initial guess  $(\mathbf{x}_{k-1}, \tilde{\boldsymbol{\lambda}}_{k-1})$ , using the Newton
   iteration defined by system (14) and merit (15).
5:   Write the solution of (13) into the tuple  $(\mathbf{x}_k, \tilde{\boldsymbol{\lambda}}_k) \in \mathbb{R}^n \times \mathbb{R}^m$ .
6:   if  $\max\{\omega + \tilde{\omega}_k, \|\nabla c(\mathbf{x}_k)\|_2\} \cdot \|\tilde{\boldsymbol{\lambda}}\|_2 \leq \text{tol}$  then
7:     return  $\mathbf{x}_k$ 
8:   else
9:     // Update  $\boldsymbol{\lambda}$  and  $\tilde{\omega}$ 
10:    if  $\|c(\mathbf{x}_k) + \omega \cdot \boldsymbol{\lambda}_k + \omega \cdot \tilde{\boldsymbol{\lambda}}_k\|_2 \leq \theta_\lambda \cdot \min_{0 \leq j < k} \{ \|c(\mathbf{x}_j) + \omega \cdot \boldsymbol{\lambda}_j + \omega \cdot \tilde{\boldsymbol{\lambda}}_j\|_2 \}$ 
    then
11:       $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}_k, \tilde{\omega}_{k+1} := \tilde{\omega}_k / \sqrt{\theta_\omega}$ 
12:       $\tilde{\boldsymbol{\lambda}}_k := \mathbf{0}$ 
13:    else
14:       $\boldsymbol{\lambda}_{k+1} := \boldsymbol{\lambda}_k, \tilde{\omega}_{k+1} = \theta_\omega \cdot \tilde{\omega}_k$ 
15:    end if
16:  end if
17: end for

```

The steps of the algorithm are clear since it is – with the only exception in line 4 – fully identical to the algorithm of ALM discussed in Section 2.2.

We briefly discuss the termination condition in line 6. We find that (13) coincides with the KKT condition of (2) for $\omega > 0$ and with the KKT conditions of (3) when $\omega = 0$. The only differences appear in terms of $\tilde{\boldsymbol{\lambda}}$. Thus, when $\tilde{\boldsymbol{\lambda}}$ has converged to zero, which it does as formerly discussed, then the KKT system for the respective value of ω is solved up to the residual tolerance $\text{tol} > 0$.

4 Outlook: Sophisticated iterations for the penalty program

We discussed three methods: Penalty method, ALM, and modified ALM. All these methods employ a Newton iteration to solve penalty problems of some form

$$\min_{\mathbf{y} \in \mathbb{R}^p} g(\mathbf{y}) + \frac{1}{2 \cdot \tilde{\omega}} \cdot \|d(\mathbf{y})\|_2^2. \quad (16)$$

The exact functions for g, d and the exact iterates for \mathbf{y} are method-dependent and shall not be further specified for the purpose of this section.

For sake of simplicity, in the former sections we always used a line-search Newton method for solving the penalty program. However, practical methods may use enhanced iterations for solving the penalty problem. These sophistica-

tions are: An adaptive choice of the penalty parameter $\tilde{\omega}$; the use of second-order corrections within the line search.

Adaptive choice of the penalty parameter The value of $\tilde{\omega}$ must be just of the right order of magnitude in order to achieve the following two antagonistic goals:

- $\tilde{\omega}$ must be sufficiently large so that (16) is well-scaled, in order to admit large steps within the Newton iteration.
- $\tilde{\omega}$ must be sufficiently small to yield boundedness of (16), otherwise a local minimizer may not exist.

In order for the outer iteration k to converge, there is usually an upper bound for $\tilde{\omega}$. Let us call this bound $\hat{\omega}_k$. Choosing $\tilde{\omega} := \hat{\omega}_k$ may not be sufficient to guarantee boundedness of (16). Thus, it is of benefit for a practical implementation that $\tilde{\omega}$ can be decreased within the inner iteration. Also, when $\tilde{\omega} < \hat{\omega}_k$ and the line search makes slow progress, we may think of increasing $\tilde{\omega}$ by using a suitable adaptive scheme, e.g., as presented in [1]. While it is of interest for our paper that we pointed out there is freedom in adapting $\tilde{\omega}$, a further discussion of how this adaption is performed is beyond the scope of this paper.

Second-order corrections within the line search Second-order corrections (SOC) are often motivated as a way for avoiding the Maratos effect [5]. Throughout this paper we did only consider unconstrained minimization techniques [3] that do not suffer from the Maratos effect by construction. However, it is acknowledged that in practice the line search benefits from second-order corrections [5].

We spend care that the SOC is formulated in a correct way with respect to our problem. Usually, consistency is achieved by the fact that the Newton iteration drives $c(\mathbf{x}) = \mathbf{0}$. But in our case this equation will never be satisfied, making it sophisticated to construct a consistent SOC. Below we give a consistent SOC for the modified ALM.

For the modified ALM we applied a quadratic penalty to the constraint

$$\tilde{c}(\mathbf{y}) \equiv c(\mathbf{x}) + \omega \cdot (\boldsymbol{\lambda}_k + \tilde{\boldsymbol{\lambda}}) = \mathbf{0}.$$

This equation has a solution but the penalty method will not converge to it. Instead, as we find from (13), the iteration will eventually solve

$$c(\mathbf{x}) + \omega \cdot \boldsymbol{\lambda}_k + (\omega + \tilde{\omega}_k) \cdot \tilde{\boldsymbol{\lambda}} = \mathbf{0}. \quad (17)$$

Therefore, at given a trial point $\mathbf{w}(\alpha) := (\mathbf{x} + \alpha \cdot \Delta \mathbf{x}, \tilde{\boldsymbol{\lambda}} + \alpha \cdot \Delta \tilde{\boldsymbol{\lambda}})$, it makes sense to consider its second-order corrected trial point

$$\mathbf{w}(\alpha)^{\text{SOC}} := \mathbf{w}(\alpha) + (\Delta \mathbf{x}^{\text{SOC}}, \Delta \tilde{\boldsymbol{\lambda}}^{\text{SOC}}),$$

that we construct from the solution of the following linear system:

$$\begin{bmatrix} \mathbf{S} & \nabla c(\mathbf{x}) \\ \nabla c(\mathbf{x})^\top & -(\omega + \tilde{\omega}) \cdot \mathbf{I} \end{bmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x}^{\text{SOC}} \\ -\Delta \tilde{\boldsymbol{\lambda}}^{\text{SOC}} \end{pmatrix} = - \begin{pmatrix} \mathbf{0} \\ c(\mathbf{x}(\alpha)) + \omega \cdot \boldsymbol{\lambda}_k + (\omega + \tilde{\omega}) \cdot \tilde{\boldsymbol{\lambda}}(\alpha) \end{pmatrix}$$

In [8] the authors propose to use multiple SOC steps. That is, a point \mathbf{w}^{SOC} is considered as trial point $\mathbf{w}(\alpha)$, and is again incremented with a correction step following the above construction.

The matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ must be symmetric positive definite and should represent the problem's scaling. The matrix determines the induced \mathbf{S} -norm in which $\Delta \mathbf{x}^{\text{SOC}}$ is minimized subject to solving a linearization of (17).

5 Conclusions

The modified ALM is an iterative method that can solve programming problems with equality constraints and problems with large quadratic penalty terms. The modified ALM is a generalization of the ALM and inherits all convergence properties of the ALM. Like the ALM, the modified ALM solves a sequence of subproblems whose penalty-parameters remain of moderate size under the same suitable conditions as for the ALM. This is advantageous for the numerical solution of the Newton system and for achieving large step sizes in the line search.

Future work is related to incorporating this method into SQP methods and Interior-Point methods for the general non-linear programming problem.

References

- [1] Paul Armand, Joël Benoist, Riadh Omhenni, and Vincent Pateloup. Study of a primal-dual algorithm for equality constrained minimization. *Comput. Optim. Appl.*, 59(3):405–433, December 2014.
- [2] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [3] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential unconstrained minimization techniques*. John Wiley & Sons.
- [4] Anders Forsgren and Philip E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–21, 11 1998. Copyright - Copyright] © 1998 Society for Industrial and Applied Mathematics; Last updated - 2012-07-02.
- [5] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.
- [6] M. Saunders. <https://web.stanford.edu/class/msande318/notes/notes09-bcl.pdf>.

- [7] Robert J. Vanderbei. Loqo:an interior point code for quadratic programming. *Optimization Methods and Software*, 11(1-4):451–484, 1999.
- [8] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.