On an inexact trust-region SQP-filter method for constrained nonlinear optimization

Andrea Walther* and Lorenz Biegler†

October 14, 2014

Abstract

A class of trust-region algorithms is developed and analyzed for the solution of optimization problems with nonlinear equality and inequality constraints. Based on composite-step trust region methods and a filter approach, the resulting algorithm also does not require the computation of exact Jacobians; only Jacobian vector products are used along with approximate Jacobian matrices. As demonstrated on numerical examples, this feature has significant potential benefits for problems where Jacobian calculations are expensive.

1 Introduction

We consider nonlinear optimization problems (NLPs) of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^N} f(x) \quad & \text{subject to} \\ c_{\mathcal{E}}(x) &= 0, \\ c_{\mathcal{I}}(x) &\le 0, \end{aligned} \tag{1}$$

where the target function $f : \mathbb{R}^N \to \mathbb{R}$, the equality constraints $c_{\mathcal{E}} : \mathbb{R}^N \to \mathbb{R}^M$ as well as the inequality constraints $c_{\mathcal{I}} : \mathbb{R}^N \to \mathbb{R}^P$ are sufficiently smooth functions.

*Institut für Mathematik, Universität Paderborn, Germany andrea.walther@uni-paderborn.de

†Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, USA lb01@andrew.cmu.edu

To apply a trust-region filter algorithm, we consider at the kth iteration for the given iterate x_k the quadratic subproblem QP(x_k)

$$\begin{aligned} \min_{s\in\mathbb{R}^N} \; & f_k + g_k^\top s + \frac{1}{2} s^\top H_k s \quad \text{subject to} \\ & c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)s = 0, \\ & c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)s \leq 0, \end{aligned} \tag{2}$$

where $f_k := f(x_k)$, $g_k := \nabla f(x_k)$, $A^{\mathcal{E}}(x_k) := \nabla c_{\mathcal{E}}(x_k)^T$, $A^{\mathcal{I}}(x_k) := \nabla c_{\mathcal{I}}(x_k)^T$, and H_k is the Hessian (or its approximation) of the Lagrange function

$$\mathcal{L}(x, \lambda_{\mathcal{E}}, \lambda_{\mathcal{I}}) = f(x) + \lambda_{\mathcal{E}}^\top c_{\mathcal{E}}(x) + \lambda_{\mathcal{I}}^\top c_{\mathcal{I}}(x)$$

with the Lagrange multipliers $\lambda_{\mathcal{E}} \in \mathbb{R}^M$ and $\lambda_{\mathcal{I}} \in \mathbb{R}^P_+$

For the step computation, we consider a restricted version, i.e., the subproblem TRQP(x_k, Δ_k)

$$\begin{aligned} \min_{s\in\mathbb{R}^N} \; & m_k(x_k + s) \quad \text{subject to} \\ c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)s &= 0, \\ c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)s &\leq 0, \\ \|s\| &\leq \Delta_k \end{aligned}$$

where

$$m_k(x_k + s) := f_k + g_k^\top s + \frac{1}{2} s^\top H_k s$$

and Δ_k is a positive trust-region radius.

For a considerable field of applications the evaluation of the Jacobians $A^{\mathcal{E}}(x_k)$ and $A^{\mathcal{I}}(x_k)$ is very expensive and may easily dominate the computing time required for solving the optimization problem. Examples for such a setting are Periodic Adsorption Processes (PAPs) that consist of vessels or beds packed with solid sorbent. The sorbent is contacted with a multi-component fluid stream to preferentially absorb one of the chemical components onto the solid. PAPs are typically operated in a cyclic manner with each bed repeatedly undergoing a sequence of steps. These cycle models consist of the bed models, PDAEs in time and space, solved for each step. After a relatively brief start-up period, the adsorption beds run in a cyclic steady state. That is, the bed conditions at the beginning of each cycle match those at the end of the cycle. This fact yields dense constraint

Jacobians, where the time required for the computation of the Jacobian dominates the overall optimization process see, e.g., [10].

Therefore, the algorithm proposed in this paper allows the inexact computation of steps. Scenarios that fit into this setting comprise the approximation of the Jacobians using low-rank updates as proposed in [8] in combination with algorithmic differentiation, reduced order models in combination with the possible evaluation of high accuracy models, or inexact system solves. However, we assume that the gradient information g_k can be evaluated exactly.

The inexact trust-region algorithm presented here extends directly from the filter TR approach proposed in [6], but allows for an inexact step computation. The central feature of the new algorithm is the ability to generate an exact local model within a bounded number of steps. This may be required for the progress of the algorithm when an exact step computation is needed. This assumption is similar to the one introduced already in [4]. The method presented here can be viewed also as a continuation of the study [14] that handles only inequality constraints.

For purely equality constrained optimization problems, an alternative approach to handle inexactness was proposed by Byrd, Curtis, and Nocedal [2] using a line search method, where inexact solves of the KKT system are used to compute the next optimization step. In the context of inexact trust-region methods, Ziems and Ulbrich proposed an approach for PDE-constrained optimization [15], where adaptive PDE discretizations were exploited. An inexact trust-region full-space approach for equality constrained optimization was presented by Heinkenschloss and Ridzal in [9]. An interior point algorithm using a line-search technique in combination with inexact step computation for qualities and inequalities was proposed by Curtis, Schenk, and Wächter [5].

The paper has the following structure; in Sect. 2 we introduce our notion of inexactness and a corresponding accuracy requirement that is necessary to obtain global convergence of the proposed algorithm. In Sect. 3, the concept of the filter approach is first revisited to introduce the features required by the inexact trust region method. The overall algorithm is presented next. The global convergence of this algorithm is proved in Sect. 4. Sect. 5 details numerical results obtained for several test problems. Finally, we give conclusions in Sect. 6 as well as an outlook for future work.

2 The inexact setting

Following the approach of Byrd [1] and Omojokun [11], we apply a composite-step trust region method, where we consider a decomposition of the full step

$$s_k = n_k + t_k$$

into a normal step n_k towards feasibility and a tangential step t_k towards optimality. In the exact setting, the normal step has to satisfy the equations

$$c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)n_k = 0, \qquad c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)n_k \leq 0.$$

Assuming existence of the normal step, one may compute n_k as orthogonal projection onto the feasible set

$$\mathcal{C}_k := \left\{x_k + s \,|\, c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)s = 0,\ c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)s \leq 0\right\}$$

of the convex QP(x_k) stated in (2). Hence, one possible normal step can be obtained as solution of the convex QP

$$\begin{aligned} &\min_p \frac{1}{2}\|n\|^2 \\ c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)n = 0 \ &\Leftrightarrow A^{\mathcal{E}}(x_k)n = -c_{\mathcal{E}}(x_k) \\ c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)n \leq 0 \ &\Leftrightarrow A^{\mathcal{I}}(x_k)n \leq -c_{\mathcal{I}}(x_k). \end{aligned}$$

However, since we do not want to build the exact Jacobians, we will work throughout with an inexact normal step. That is, we assume that the normal step n_k satisfies only

$$\|c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)n\| \leq \mathrm{err}_k^{\mathcal{E},n} \quad \text{and} \quad c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)n \leq (\mathrm{err}_k^{\mathcal{I},n})e \tag{3}$$

where $e = [1, 1, \ldots 1]$, for suitable bounded errors $\mathrm{err}_k^{\mathcal{E},n}, \mathrm{err}_k^{\mathcal{I},n} \geq 0$ as described below in more detail. Such an inexact step calculation might be due to an inexact solve of QP(x_k). Alternatively, one may use the inexact representation of the feasible set of QP(x_k) given by

$$c_{\mathcal{E}}(x_k) + A_k^{\mathcal{E}}n_k = 0, \qquad c_{\mathcal{I}}(x_k) + A_k^{\mathcal{I}}n_k \leq 0, \tag{4}$$

which is based on inexact Jacobians $A_k^{\mathcal{E}}$ and $A_k^{\mathcal{I}}$. In this case, the normal step n_k could be computed as orthogonal projection on the approximation

$$\tilde{\mathcal{C}}_k := \left\{x_k + s \,|\, c_{\mathcal{E}}(x_k) + A_k^{\mathcal{E}}s = 0,\ c_{\mathcal{I}}(x_k) + A_k^{\mathcal{I}}s \leq 0\right\}$$

of the corresponding set $\mathcal{C}_k$ in the exact setting.

We will use this approach also for our numerical tests presented in Sect. 5, but other alternatives are also possible. In fact, as noted in [6] there are a number of ways to calculate n_k that are compatible with the analysis of trust region filter methods. Examples include Byrd-Omojukun normal steps and non-orthogonal projections. In analogy to [6], the only requirement we need is that $n_k \in \tilde{\mathcal{C}}_k$ and that $\|n_k\|$ is bounded above as discussed below. Based on $\tilde{\mathcal{C}}_k$ we also define the following Inexact Trust Region QP, ITRQP(x_k, Δ_k):

$$\begin{aligned} \min_{s \in \mathbb{R}^N} \ m_k(x_k+s) \quad & \text{subject to} \\ c_{\mathcal{E}}(x_k) + A_k^{\mathcal{E}} s &= 0, \\ c_{\mathcal{I}}(x_k) + A_k^{\mathcal{I}} s &\le 0, \\ \|s\| &\le \Delta_k \end{aligned}$$

Note that our analysis is based only on the inexactness given in (3), allowing for a wide range of inexact step computations.

As in [6], we will use the concept of a compatible trust-region subproblem, i.e., the normal step must not yield a point outside the trust region, and not too close to the trust-region boundary. In the context of ITRQP(x_k, Δ_k) this corresponds to the following stronger condition.

Definition 2.1 (Compatibility)**.** *We will call the subproblem ITRQP(x_k, Δ_k) compatible if the corresponding normal step n_k satisfies*

$$\|n_k\| \le \kappa_\Delta \Delta_k \min\{1, \kappa_\mu \Delta_k^\mu\} \tag{5}$$

for constants $\kappa_\Delta \in (0,1]$, $\kappa_\mu > 0$, and $\mu \in (0,1)$.

If the trust-region subproblem ITRQP(x_k, Δ_k) is compatible then

$$x_k^n = x_k + n_k$$

represents a step towards feasibility in the inexact setting. For the analysis of our algorithm, we relate the normal step n_k with the maximum violation of the nonlinear constraints at the same iterate $\theta_k := \theta(x_k)$ given by

$$\theta(x) := \max\left\{0,\ \max_{i\in\mathcal{E}} |c_i(x)|,\ \max_{i\in\mathcal{I}} c_i(x)\right\} \tag{6}$$

and we require:

Assumption 2.2 (Existence and Boundedness of Normal Step)**.** *Throughout we assume that the normal step*

$$n_k \text{ exists} \quad \text{and} \quad \|n_k\| \le \kappa_{ubn}\theta_k \;\text{ if }\; \theta_k \le \delta_n \tag{7}$$

for constants $\kappa_{ubn} > 0$ and $\delta_n > 0$.

If ITRQP(x_k, Δ_k) is not compatible, i.e., there exists no normal step or condition (5) is violated, a restoration phase is used to compute a step r_k such that ITRQP$(x_k + r_k, \Delta_{k+1})$ is compatible for some $\Delta_{k+1} > 0$. This can be achieved by ensuring that $\theta(x_k + r_k)$ is sufficiently reduced from $\theta(x_k)$, for example by solving

$$\min_{x \in \mathbb{R}^N} \theta(x) \tag{8}$$

or an appropriate smoothed version of this nonsmooth problem.

In our algorithm, we will also allow an inexact computation of the tangential step, i.e., t_k is computed as solution of the inexact tangential problem

$$\min_{t \in \mathbb{R}^N} \; (g_k + H_k n_k)^\top t + 1/2 t^\top H_k t \quad \text{subject to} \tag{9}$$

$$\|A^{\mathcal{E}}(x_k) t\| \le \mathrm{err}_k^{\mathcal{E},t}, \tag{10}$$

$$c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)(n_k + t) \le \mathrm{err}_k^{\mathcal{I}}, \tag{11}$$

$$\|n_k + t\| \le \Delta_k \tag{12}$$

for suitable bounded errors $\mathrm{err}_k^{\mathcal{E},t}, \mathrm{err}_k^{\mathcal{I}} \ge 0$ as described below in more detail.

The tangential step is used to judge the sufficient decrease condition to obtain convergence. For the corresponding analysis, we will require:

Assumption 2.3 (Sufficient Decrease Condition)**.** *If ITRQP(x_k, Δ_k) is compatible, then*

$$m_k(x_k^n) - m_k(x_k^n + t_k) \ge \kappa_{sdt} \hat{\chi}_k \min\left\{ \frac{\hat{\chi}_k}{\beta_k}, \Delta_k \right\} \tag{13}$$

holds for a fixed constant $\kappa_{sdt} > 0$ and $\beta_k := 1 + \|H_k\|$.

Note, that for the full step $s_k = n_k + t_k$ one has:

$$\|c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k) s\| \le \mathrm{err}_k^{\mathcal{E},n} + \mathrm{err}_k^{\mathcal{E},t}, \quad c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k) s \le \mathrm{err}_k^{\mathcal{I}}.$$

Similar to the proof in [3, Theo. 12.2.2], one can show that (13) holds for the inexact setting considered here, when the model reduction exceeds what would be obtained at the generalized Cauchy point for ITRQP.

As in [6], one may use the following criticality measure to deduce first-order optimality

$$\chi_k := \left| \min \left\{ (g_k + H_k n_k)^\top t \,\middle|\, A^{\mathcal{E}}(x_k)t = 0, \right.\right.$$
$$\left.\left. c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)(n_k + t) \leq 0, \|t\| = 1 \right\} \right|. \tag{14}$$

Since the feasible set $\mathcal{C}$ of ITRQP(x_k) is convex, one may alternatively employ also

$$\chi_k := \|x_k - P_{\mathcal{C}_k}(x_k - \nabla f(x_k))\| \tag{15}$$

as criticality measure, see [3, Section 12.1.4].

In this work we define an inexact criticality measure since we do not have the exact Jacobians at hand. The inexact version of the criticality measure (14) is given by

$$\hat{\chi}_k := \left| \min \left\{ (g_k + H_k n_k)^\top t \,\middle|\, \|A^{\mathcal{E}}(x_k)t\| \leq \mathrm{err}_k^{\mathcal{E},t}, \right.\right.$$
$$\left.\left. c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)(n_k + t) \leq (\mathrm{err}_k^{\mathcal{I}})e, \|t\| = 1 \right\} \right|,$$

again for suitable bounded errors $\mathrm{err}_k^{\mathcal{E},t}, \mathrm{err}_k^{\mathcal{I}} \geq 0$ as described below in more detail. Hence, if $t_k = 0 \in \mathbb{R}^n$ solves the tangential problem (9), then this inexact criticality measure is zero indicating that $x_* = x_k + n_k$ is a first-order critical point for the inexact setting.

The inexact version

$$\hat{\chi}_k := \|x_k - P_{\hat{\mathcal{C}}_k}(x_k - \nabla f(x_k))\|$$

for the alternative (15) projects once more onto the feasible set of ITRQP(x_k), i.e.,

$$\hat{\mathcal{C}}_k := \left\{ x_k^n + t \,|\, \|A^{\mathcal{E}}(x_k)t\| \leq \mathrm{err}_k^{\mathcal{E},t},\ c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)(n_k + t) \leq (\mathrm{err}_k^{\mathcal{I}})e \right\}.$$

The analysis of the algorithm presented in the next section is independent of the choice of the criticality measure and builds on upper bounds for the errors $\mathrm{err}_k^{\mathcal{E},t}, \mathrm{err}_k^{\mathcal{I}} \geq 0$

For the proof of convergence in the inexact setting, we need a certain quality of the approximate solutions as stated next.

Assumption 2.4 (Accuracy Requirement)**.** *The step $s_k = n_k + t_k$ fulfills the accuracy requirement if*

$$\max \left\{ \|c_{\mathcal{E}}(x_k) + A^{\mathcal{E}}(x_k)n_k\|,\ c_{\mathcal{I}}(x_k) + A^{\mathcal{I}}(x_k)n_k \right\} \leq \kappa_{en}\Delta_k^{1+\mu} \tag{16}$$

as well as

$$\max\left\{\|c_{\mathcal{E}}(x_k)+A^{\mathcal{E}}(x_k)s_k\|, c_{\mathcal{I}}(x_k)+A^{\mathcal{I}}(x_k)s_k\right\} \le \kappa_{es} \min\left\{\Delta_k^{1+\mu}, \Delta_k^{1+\sigma}\right\} \quad (17)$$

hold for constants $\sigma > \mu > 0$ and $\kappa_{en}, \kappa_{es} > 0$ independent of k.

We note that the requirement (16) is the same as presented in [6], while (17) is a relaxation from [6] to reflect the effect of the inexactness in determining t_k. These requirements can be verified easily if one works with the inexact Jacobians $A_k^{\mathcal{E}}$ and $A_k^{\mathcal{I}}$, and assumes that Jacobian-vector products can be efficiently evaluated. Using (4), the requirements (16) and (17), respectively, can be tested using

$$\max\left\{\|A_k^{\mathcal{E}} n_k - A^{\mathcal{E}}(x_k)n_k\|, A_k^{\mathcal{I}} n_k - A^{\mathcal{I}}(x_k)n_k\right\} \le \kappa_{en}\Delta_k^{1+\mu}$$

and

$$\max\left\{\|A_k^{\mathcal{E}} s_k - A^{\mathcal{E}}(x_k)s_k\|, \|A_k^{\mathcal{I}} s - A^{\mathcal{I}}(x_k)s_k\|\right\} \le \kappa_{es} \min\left\{\Delta_k^{1+\mu}, \Delta_k^{1+\sigma}\right\}.$$

3 An Inexact Trust-Region SQP-Filter Algorithm

For the globalization of our inexact approach, we apply a filter technique to decide whether a point $x_k + s_k$ (or $x_k + r_k$ in the restoration phase) is in some sense better than the current iterate x_k. For this purpose, we build a collection $\mathcal{F}$ of pairs (θ_j, f_j) known as filter. More details about the idea and the background of the filter method can be found in [3, Chapter 15.5] and [6]. For a given filter $\mathcal{F}$, the following criterion is used to judge the progress provided by the next iterate:

Definition 3.1 (Acceptance for the filter)**.** *A point x is acceptable for the filter $\mathcal{F}$ if and only if*

$$\theta(x) \le (1-\gamma_\theta)\theta_j \quad \textit{or} \quad f(x) \le f_j - \gamma_\theta\theta_j \qquad \textit{for all } (\theta_j, f_j) \in \mathcal{F}$$

for a constant $\gamma_\theta \in (0,1)$. Furthermore, a point x is acceptable for the filter $\mathcal{F}$ and x_k if and only if

$$\theta(x) \le (1-\gamma_\theta)\theta_j \quad \textit{or} \quad f(x) \le f_j - \gamma_\theta\theta_j \qquad \textit{for all } (\theta_j, f_j) \in \mathcal{F} \cup \{(\theta_k, f_k)\}$$

with the same constant $\gamma_\theta \in (0,1)$.

During the optimization process, we will use the term that *a point x_k is added to the filter*, although one adds the pair (θ_k, f_k) to the set $\mathcal{F}$. In this case, one also removes all pairs (θ_j, f_j) with

$$\theta_j \geq \theta_k \quad \text{and} \quad f_j - \gamma_\theta \theta_j \geq f_k - \gamma_\theta \theta_k$$

from the filter.

The inexact trust-region algorithm below builds heavily on the ability to generate an exact model m_k within a bounded number of steps. For instance, specialized quasi-Newton updates for inexact Jacobians were developed in [14]. This ability is required by the so-called *Improvement Algorithm* in the optimization process, so that one can compute exact steps such that

$$\text{err}_k^{\mathcal{E},n} = \text{err}_k^{\mathcal{I},n} = \text{err}_k^{\mathcal{E},t} = \text{err}_k^{\mathcal{I}} = 0 \tag{18}$$

and the inequalities (16) and (17) are valid. The model m_k, where (18) is satisfied is called *a good model*. A discussion of the improvement algorithm will be given below when the calling structure from the trust-region algorithm is known.

Algorithm I: Trust-Region SQP-Filter Algorithm

0. **Initialization.** Choose an initial point x_0, initial values $\Delta_0 > 0$, $A_0^{\mathcal{E}}$, $A_0^{\mathcal{I}}$, H_0, and constants $0 < \gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$; $0 < \eta_1 \leq \eta_2 < 1$; $\varepsilon_c > 0$; $\gamma_\theta, \kappa_\theta \in (0,1)$; $\kappa_\Delta, \kappa_{sdt} \in (0,1]$; $\kappa_\mu, \kappa_{ej} > 0$; $\sigma, \mu \in (0,1)$, $\sigma > \mu$; $\psi > 1/(1+\mu)$, $\kappa_{en}, \kappa_{es} > 0$. Compute $f(x_0)$, $c_{\mathcal{E}}(x_0)$, $c_{\mathcal{I}}(x_0)$. Set $\mathcal{F} = \emptyset$ and $k = 0$.

1. **Model quality.** If $\hat{\chi}_k < \varepsilon_c$ then apply Improvement Algorithm and update $\hat{\chi}_k$.

2. **Optimality test.** If $\theta_k = 0$ and $\hat{\chi}_k = 0$: STOP

3. **Compatibility test.** Attempt to compute n_k
   If n_k exists and ITRQP(x_k, Δ_k) is compatible go to 5.

4. **Model Improvement/Restoration Check.** If m_k is not a good model apply Improvement Algorithm, go to 3.
   Else, if m_k is a good model then

   add x_k to the filter and attempt to compute a restoration step r_k for which ITRQP(x_k+r_k, Δ_{k+1}) is compatible for some $\Delta_{k+1} > 0$, and $x_k + r_k$ is acceptable for the filter.
   If this is not possible: STOP.
   Otherwise, set $x_{k+1} = x_k + r_k$ and go to 9.

5. **Step computation.** Compute t_k and set $s_k := n_k + t_k$

6. **Accuracy test.** If (16) and (17) do not hold then
   apply Improvement Algorithm and go to 3.

7. **Acceptance test.** Compute $f(x_k + s_k)$ and $c(x_k + s_k)$.
   If $x_k + s_k$ is not acceptable for the filter and x_k then
   set $x_{k+1} = x_k$, $n_{k+1} = n_k$, choose $\Delta_{k+1} \in [\gamma_0\Delta_k, \gamma_1\Delta_k]$, set $k = k+1$, go to 3.
   If

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta \theta_k^\psi \tag{19}$$

$$\text{and} \qquad \rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} < \eta_1$$

   then set $x_{k+1} = x_k$, $n_{k+1} = n_k$, choose $\Delta_{k+1} \in [\gamma_0\Delta_k, \gamma_1\Delta_k]$, set $k = k + 1$, go to 3.

8. **Filter update:** If (19) does not hold add x_k to the filter $\mathcal{F}$.

9. **Next iterate:** Set $x_{k+1} = x_k + s_k$. If (19) holds choose

$$\Delta_{k+1} \in \begin{cases} [\gamma_1\Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \\ [\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k \geq \eta_2 \end{cases}$$

10. **Updates:** Determine H_{k+1}, $A_{k+1}^{\mathcal{E}}$, $A_{k+1}^{\mathcal{I}}$, set $k = k + 1$, go to Step 1.

We will call an iterate x_k together with the step s_k *successful*, if $x_k + s_k$ passes all tests to reach step 8 of Algorithm I, i.e., x_k is not added to the filter.

This enforced accuracy given by (18) is required for the calls of the Improvement Algorithm in Step 1, i.e., near a possible first-order critical point, cf. the proof of Theorem 4.15. This is similar to the exactness requirements used in [4]. Furthermore, there is an obvious relation to inexact Newton methods where exact solutions are enforced and required only near the solution point.

For all other calls of the improvement algorithm, one may for example increase the accuracy repeatedly as long as (18) is valid after a certain fixed number of calls to the improvement algorithm at the same point x_k. In the setting of the inexact Jacobians the enforced accuracy (18) could be achieved by computing the exact Jacobians each time when the accuracy tests (16) and (17) fail. Alternatively, one may employ a suitable update strategy in the steps 3 and 5 to improve the inexact Jacobians as long as (18) is valid after a certain fixed number of calls to the improvement algorithm at the same point x_k.

For these relaxed updating strategies, it is important to note that the iteration point x_k itself is not updated if the accuracy test fails. Hence, we apply the *Improvement Algorithm* to ensure that (18) holds after a fixed number of updates. For instance, specialized quasi-Newton updates for inexact Jacobians from [14] could be applied here. The modified updating in Steps 4 and 6 would allow faster progress of the algorithm, but each updating step is then more expensive due to the computation of intermediate results. These might become useless when repeated failure of the tests (16) and (17) occurs. It follows that there is a wide range of possibilities to explore. For this reason, we leave the details of Algorithm II open. These can be adapted based on where the model improvement algorithm is invoked.

Algorithm II: Improvement Algorithm

Improve the model m_k such that after a finite number K of iterations the good model m_k allows the computation of exact steps fulfilling (18).

4 Convergence analysis

Definition 4.1 (Sets of Iterates)**.** *We define*

$$\mathcal{S} := \{k \,|\, x_{k+1} = x_k + s_k\}$$

as the set of indices of successful iterations and

$$\mathcal{R} := \left\{k \,|\, n_k \text{ does not satisfy } (7) \quad \text{or} \quad \|n_k\| > \kappa_\Delta \Delta \min\{1, \kappa_\mu \Delta_k^\mu\}\right\}$$

as the set of indices of restoration iterations. Furthermore let

$$\mathcal{Z} := \{k \,|\, x_k \text{ is added to the filter }\}$$

be the set of the iterates where the filter is modified.

For the proof of the global convergence result, we will need the following properties as in [6]:

Assumption 4.2.

AS1 The target function f as well as the constraint functions $c_\mathcal{E}$ and $c_\mathcal{I}$ are twice continuously differentiable.

AS2 The Hessian approximations H_k of all iterates are bounded from above, i.e., there exists a constant $\kappa_{ubh} > 1$ such that

$$\|H_k\| \leq \kappa_{ubh} - 1 \qquad \text{for all } k.$$

AS3 The iterates $\{x_k\}$ remain in a closed, bounded domain X.

From these assumptions, one can directly derive that there exists a constant κ_{ef} bounding the error in the model from above in the following way:

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{ef}\Delta_k^2.$$

Furthermore, one obtains from the smoothness of the functions and the boundedness of X that there exist constants $f_{\min} \in \mathbb{R}$ and $\theta_{\max} \geq 0$ with

$$f_{\min} \leq f(x_k) \quad \text{and} \quad 0 \leq \theta_k \leq \theta_{\max} \qquad \text{for all } k.$$

Hence the (θ, f)-space containing all filter iterates can be restricted to the rectangle

$$\mathcal{A} := [0, \theta_{\max}] \times [f_{\min}, \infty].$$

Lemma 4.3 (First upper bound on θ)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1 and AS3 hold and that*

$$\theta_k \leq \delta_n$$

is valid. Then, the normal step n_k exists. Furthermore, there is a constant $\kappa_{ubj} > 0$ independent of k such that one has

$$\theta_k \leq \kappa_{en}\Delta_k^{1+\mu} + \kappa_{ubj}\|n_k\|. \tag{20}$$

Proof: The existence of the normal step follows from Assumption 2.2 since $\theta_k \leq \delta_n$. Defining

$$\mathcal{V}_k := \{j \in \mathcal{E} \,|\, \theta_k = |c_j(x_k)|\} \bigcup \{j \in \mathcal{I} \,|\, \theta_k = -c_j(x_k)\}$$

as the set of indices of the most violated constraints, one obtains from the definition of θ_k in (6), the normal step n_k in (3), the accuracy requirement (16) and the Cauchy-Schwarz inequality that

$$\begin{aligned}\theta_k = |c_j(x_k)| &\leq \|c_j(x_k) + (A^{\mathcal{E}}(x_k))_j n_k\| + \|(A^{\mathcal{E}}(x_k))_j n_k\| \\ &\leq \kappa_{en}\Delta_k^{1+\mu} + \|(A^{\mathcal{E}}(x_k))_j\| \, \|n_k\|\end{aligned}$$

for all $j \in \mathcal{E} \cap \mathcal{V}_k$, where B_j denotes the row j of the matrix B. Then Assumption AS3 yields the existence of a constant $\kappa_{ubj1} > 0$ such that

$$\theta_k \leq \kappa_{en}\Delta_k^{1+\mu} + \kappa_{ubj1}\|n_k\|$$

for all $j \in \mathcal{E} \cap \mathcal{V}_k$. For $j \in \mathcal{I} \cap \mathcal{V}_k$, one obtains in a similar way that

$$\begin{aligned}\theta_k = c_j(x_k) &\leq \|c_j(x_k) + (A^{\mathcal{I}}(x_k))_j n_k\| + \|(A^{\mathcal{I}}(x_k))_j n_k\| \\ &\leq \kappa_{en}\Delta_k^{1+\mu} + \kappa_{ubj2}\|n_k\|.\end{aligned}$$

With $\kappa_{ubj} := \max\{\kappa_{ubj1}, \kappa_{ubj2}\}$, one obtains the assertion. □

Lemma 4.4 (Limit of θ_k for Filter Iterates)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1 and AS3 hold and that $|\mathcal{Z}| = \infty$. Then*

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{Z}}} \theta_k = 0.$$

Proof: This lemma can be shown along the lines of [6, Lemma 3.3] or [7, Lemma 1 and Corollary 1] since the proofs in these papers only rely on the filter mechanism, which is here exactly the same as in these papers. □

Lemma 4.5 (Further upper bound on θ)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1 and AS3 hold, that $k \notin \mathcal{R}$, and that n_k satisfies* (20)*. Then there exists a constant $\kappa_{ubt} > 0$ such that*

$$\theta_k \leq \kappa_{ubt}\Delta_k^{1+\mu} \tag{21}$$

and

$$\theta(x_k + s_k) \leq \kappa_{ubt} \max\{\Delta_k^{1+\sigma}, \Delta_k^{1+\mu}, \Delta_k^2\}.$$

Proof: Since $k \notin \mathcal{R}$, one obtains from (5) and (20) that

$$\theta_k \leq \kappa_{en}\Delta_k^{1+\mu} + \kappa_{ubj}\|n_k\| \leq (\kappa_{en} + \kappa_{ubj}\kappa_\Delta\kappa_\mu)\Delta_k^{1+\mu}.$$

The jth constraint function at $x_k + s_k$ can be expressed as

$$c_j(x_k + s_k) = c_j(x_k) + \nabla_x c_j(x_k)^\top s_k + \frac{1}{2}s_k^\top \nabla_{xx} c_j(\xi_k) s_k$$

for $j \in \mathcal{E} \cup \mathcal{I}$ applying AS1 and the mean-value theorem for $\xi_k \in \{x \,|\, x = x_k + \alpha s_k, \alpha \in [0,1]\}$. One obtains from the accuracy requirement (17) in step 5 of Algorithm I that

$$|c_j(x_k) + (A^{\mathcal{E}}(x_k))_j s_k| \leq \kappa_{es} \min\{\Delta_k^{1+\mu}, \Delta_k^{1+\sigma}\}$$

for all $j \in \mathcal{E}$. Furthermore, AS3 gives a bound κ_1 for the Hessian of the constraint functions. Therefore, one obtains from $\|s_k\| \leq \Delta_k$ and the Cauchy-Schwarz inequality that

$$|\frac{1}{2} s_k^\top \nabla_{xx} c_j(\xi_k) s_k| \leq \kappa_1 \Delta_k^2.$$

Combining both upper bounds, it follows that

$$|c_j(x_k + s_k)| \leq (\kappa_{es} + \kappa_1) \max\left\{\Delta_k^{1+\sigma}, \Delta_k^{1+\sigma}, \Delta_k^2\right\}.$$

For $j \in \mathcal{I}$, it follows from the accuracy requirement (17) in step 5 of Algorithm I and AS3 that

$$\begin{aligned} c_j(x_k + s_k) &= c_j(x_k) + \nabla_x c_j(x_k)^\top s_k + \frac{1}{2} s_k^\top \nabla_{xx} c_j(\xi_k) s_k \\ &\leq \kappa_{es} \Delta_k^{1+\sigma} + \|\frac{1}{2} s_k^\top \nabla_{xx} c_j(\xi_k) s_k\| \\ &\leq (\kappa_{es} + \kappa_2) \max\left\{\Delta_k^{1+\sigma}, \Delta_k^{1+\sigma}, \Delta_k^2\right\} \end{aligned}$$

for a constant $\kappa_2 > 0$ independent of k. This gives the required bound with

$$\kappa_{ubt} = \max\{\kappa_{en} + \kappa_{ubj}\kappa_\Delta\kappa_\mu, \kappa_{es} + \kappa_1, \kappa_{es} + \kappa_2\}.$$

□

Lemma 4.6 (Lower Bound on Model Improvement I)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3,* (5)*, and* (13) *hold, that* $k \notin \mathcal{R}$*, that*

$$\hat{\chi}_k \geq \varepsilon$$

for a constant $\varepsilon > 0$*, and that*

$$\Delta_k \leq \min\left\{\frac{\varepsilon}{\kappa_{ubh}}, \left(2\frac{\kappa_{ubg}}{\kappa_{ubh}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{1+\mu}}, \left(\frac{\kappa_{sdt}\varepsilon}{4\kappa_{ubg}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{\mu}}\right\} =: \delta_m,$$

where $\kappa_{ubg} := \max_{x\in X} \|\nabla_x f(x)\|$*. Then*

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}\kappa_{sdt}\varepsilon\Delta_k.$$

Proof: This is the same proof as in [6, Lemma 3.5]. One only has to substitute the exact criticality measure χ_k used in [6] with the inexact one $\hat{\chi}_k$ considered here. □

Lemma 4.7 (Very Successful Steps)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and* (13) *hold, that $\hat{\chi}_k \geq \varepsilon$ for a constant $\varepsilon > 0$, $k \notin \mathcal{R}$, and that*

$$\Delta_k \leq \min\left\{\delta_m, \frac{(1-\eta_2)\kappa_{sdt}\varepsilon}{2\kappa_{ef}}, 1\right\} =: \delta_\rho,$$

where δ_m was defined in Lemma 4.6. Then

$$\rho_k \geq \eta_2$$

Proof: One can use exactly the same argument as in [6, Lemma 3.6]. □

Lemma 4.8 (Lower Bound on Model Improvement II)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3,* (5)*, and* (13) *hold, that $\hat{\chi}_k \geq \varepsilon$ for a constant $\varepsilon > 0$, that n_k satisfies* (20)*, that $k \notin \mathcal{R}$, and that*

$$\Delta_k \leq \min\left\{\delta_m, \left(\frac{\kappa_{sdt}\varepsilon}{2\kappa_\theta\kappa_{ubt}^\psi}\right)^{\frac{1}{\psi(1+\mu)-1}}\right\} =: \delta_f.$$

where δ_m was defined in Lemma 4.6. Then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta\theta_k^\psi.$$

Proof: One can use exactly the same argument as in [6, Lemma 3.7]. □

Lemma 4.9 (Decrease in Target Function)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3,* (13) *hold, that $\hat{\chi}_k \geq \varepsilon$ for a constant $\varepsilon > 0$, that n_k satisfies* (20)*, that $k \notin \mathcal{R}$, $\Delta_k \leq \delta_\rho$ where δ_ρ is defined in Lemma 4.7 and that*

$$\theta_k \leq \kappa_{ubt}^{-\frac{1}{\mu}}\left(\frac{\eta_2\kappa_{sdt}\varepsilon}{2\gamma_\theta}\right)^{\frac{1+\mu}{\mu}} =: \delta_\theta.$$

Then

$$f_k(x_k + s_k) \leq f(x_k) - \gamma_\theta\theta_k.$$

Proof: One can use exactly the same argument as in [6, Lemma 3.8]. □

Lemma 4.10 (Compatibility)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and* (7) *hold, that* $\hat{\chi}_k \geq \varepsilon$ *for a constant* $\varepsilon > 0$*, that* (13) *holds for* $k \notin \mathcal{R}$*, and that*

$$\Delta_k \leq \min\left\{\gamma_0\delta_\rho, (1/\kappa_\mu)^{\frac{1}{\mu}}, \hat{\Delta}, \bar{\Delta}\right\} =: \delta_{\mathcal{R}}. \tag{22}$$

$$\textit{with} \quad \hat{\Delta} := \left(\frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{ubn}\kappa_{ubt}}\right)^{\frac{1}{\sigma-\mu}}, \; \bar{\Delta} = \left(\frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{ubn}\kappa_{ubt}}\right)^{\frac{1}{1-\mu}}$$

and δ_ρ *as defined in Lemma 4.7. Furthermore, suppose that* $k > 0$ *and that*

$$\theta_k \leq \min\{\delta_\theta, \delta_n\}.$$

Then $k \notin \mathcal{R}$*.*

Proof: This is an adapted version of the proof for [6, Lemma 3.9] that takes the inexactness into account.

The existence of the normal step follows from Assumption 2.2 since $\theta_k \leq \delta_n$. Now assume for deriving a contradiction that $k \in \mathcal{R}$, i.e.,

$$\|n_k\| > \kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} \tag{23}$$

using $\kappa_\mu\Delta_k^\mu \leq 1$ due to the bound on Δ_k. Then, Algorithm I ensures that $k-1 \notin \mathcal{R}$. Suppose that iteration $k-1$ is not successful. From Lemma 4.7 we obtain that $\rho_{k-1} \geq \eta_2$. Furthermore, Lemma 4.9 holds at iteration $k-1$ due to (7), the bound (22) on the trust-region radius, the fact that $\theta_k = \theta_{k-1}$, $\theta_k \leq \delta_\theta$, and because $k-1 \notin \mathcal{R}$. This yields

$$f(x_{k-1} + s_{k-1}) \leq f(x_{k-1}) - \gamma_\theta\theta_{k-1}.$$

Since x_{k-1} is acceptable for the filter at the beginning of iteration $k-1$, this iteration can only be not successful if $x_{k-1} + s_{k-1}$ is not acceptable for the filter and x_{k-1}. Due to the last estimate for the reduction in the objective function, the only way that this point can be unacceptable to the filter is if

$$\theta(x_{k-1} + s_{k-1}) > (1-\gamma_\theta)\theta_{k-1} = (1-\gamma_\theta)\theta_k.$$

However using $\gamma_0 \in (0,1)$, Lemma 4.5, $\Delta_{k-1} \leq 1$ and the mechanism of Algorithm I imply that

$$\begin{aligned}(1-\gamma_\theta)\theta_k \leq \kappa_{ubt}\max\left\{\Delta_{k-1}^{1+\sigma}, \Delta_{k-1}^2\right\} &\leq \kappa_{ubt}\max\left\{\frac{\Delta_k^{1+\sigma}}{\gamma_0^{1+\sigma}}, \frac{\Delta_k^2}{\gamma_0^2}\right\} \\ &\leq \frac{\kappa_{ubt}}{\gamma_0^2}\max\left\{\Delta_k^{1+\sigma}, \Delta_k^2\right\}.\end{aligned}$$

Combining this inequality with (23) and (7), one can conclude that

$$\kappa_\Delta \kappa_\mu \Delta_k^{1+\mu} < \|n_k\| \leq \kappa_{ubn}\theta_k \leq \frac{\kappa_{ubn}\kappa_{ubt}}{\gamma_0^2(1-\gamma_\theta)} \max\left\{\Delta_k^{1+\sigma}, \Delta_k^2\right\} \tag{24}$$

and therefore that

$$\max\left\{\Delta_k^{1-\mu}, \Delta_k^{\sigma-\mu}\right\} > \frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{ubn}\kappa_{ubt}}.$$

Hence, at least one of the inequalities

$$\Delta_k > \left(\frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{ubn}\kappa_{ubt}}\right)^{\frac{1}{\sigma-\mu}} \quad \text{and} \quad \Delta_k > \left(\frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{ubn}\kappa_{ubt}}\right)^{\frac{1}{1-\mu}}.$$

must hold. However, both inequalities contradict (22), the assumed bound on Δ_k. Hence, the assumption that iteration $k-1$ is not successful cannot be true. Therefore, iteration $k-1$ must be successful yielding

$$\begin{aligned}\kappa_\Delta \kappa_\mu \Delta_k^{1+\mu} < \|n_k\| &\leq \kappa_{ubn}\kappa_{ubt} \max\left\{\Delta_{k-1}^{1+\sigma}, \Delta_{k-1}^2\right\} \\ &\leq \frac{\kappa_{ubn}\kappa_{ubt}}{\gamma_0^2} \max\left\{\Delta_k^{1+\sigma}, \Delta_k^2\right\}.\end{aligned}$$

Since $1-\gamma_\theta < 1$ this contradicts once more the assumed bound on Δ_k yielding that the assumption (23) must be false and, the original assertion is proved. □

Lemma 4.11 (Convergence of θ and $\hat{\chi}$ for Infinite Filter)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and* (7) *hold, and that* (13) *holds for* $k \notin \mathcal{R}$. *Suppose that* $|\mathcal{Z}| = \infty$. *Then there exists a subsequence* $\{k_i\} \subset \mathcal{Z}$ *such that*

$$\lim_{i\to\infty} \theta_{k_i} = 0$$

and

$$\lim_{i\to\infty} \hat{\chi}_{k_i} = 0.$$

Proof: This is the same proof as in [6, Lemma 3.10], which builds on Lemma 4.4, Lemma 4.8, and Lemma 4.10. One only has to substitute the exact criticality measure χ_k used in [6] with the inexact one $\hat{\chi}_k$ considered here. □

From now on, let k_0 denote the last iteration for which x_{k_0-1} is added to the filter.

Lemma 4.12 (Convergence of θ_k for Finite Filter)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and* (7) *hold, and that* (13) *holds for* $k \notin \mathcal{R}$*. Suppose that* $|\mathcal{Z}| < \infty$*. Then one has*

$$\lim_{k \to \infty} \theta_k = 0.$$

Proof: One can use exactly the same argument as in [6, Lemma 3.11]. □

Lemma 4.13 (Bounded Trust-region Radius)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and that* (13) *holds for* $k \notin \mathcal{R}$*. Suppose that* $|\mathcal{Z}| < \infty$ *and that* $\hat{\chi}_k \geq \varepsilon$ *hold for a constant* $\varepsilon > 0$ *and all* $k \geq k_0$*. Then there exists a* $\Delta_{\min} > 0$ *such that*

$$\Delta_k \geq \Delta_{\min}$$

Proof: This is an adapted version of the proof for [6, Lemma 3.12] to take the inexactness into account.

Using Lemma 4.12, we know that there exists an index $k_1 \geq k_0$ such that $\theta_k \leq \min\{\delta_n, \delta_\theta\}$ for all $k \geq k_1$. Then, (7) holds for all $k \geq k_1$. Using (24), we now now assume, for the purpose of deriving a contradiction, that iteration j is the first iteration following k_1 for which

$$\Delta_j \leq \gamma_0 \min \left\{ \delta_\rho, \left(\frac{(1-\gamma_\theta)\theta^F}{\kappa_{ubt}} \right)^{\frac{1}{2}}, \left(\frac{(1-\gamma_\theta)\theta^F}{\kappa_{ubt}} \right)^{\frac{1}{1+\sigma}}, \Delta_{k_1} \right\} =: \gamma_0 \delta_s \quad (25)$$

and $\theta^F := \min_{i \in \mathcal{Z}} \theta_i$ denotes the smallest constraint violation in the filter. Since we have $\Delta_j \leq \gamma_0 \Delta_{k_1}$ it follows that $j \geq k_1 + 1$ and therefore $j - 1 \notin \mathcal{R}$. Then, Algorithm I and the bound on Δ_j ensure that

$$\Delta_{j-1} \leq \frac{1}{\gamma_0} \Delta_j \leq \delta_s.$$

Due to the bounds on Δ_j and Δ_{j-1} we can apply Lemma 4.7 yielding $\rho_{j-1} \geq \eta_2$. Since (7) holds also for $j-1$, Lemma 4.3 ensures that we use Lemma 4.5 to obtain with the bound (25)

$$\theta(x_{j-1} + s_{j-1}) \leq \kappa_{ubt} \max \left\{ \Delta_{j-1}^{1+\sigma}, \Delta_{j-1}^2 \right\} \leq (1 - \gamma_\theta)\theta^F.$$

Due to the bound on Δ_j we can also apply Lemma 4.9 such that we get

$$f(x_{j-1} + s_{j-1}) \leq f(x_{j-1}) - \gamma_\theta \theta_{j-1}.$$

The last two inequalities yield that $x_{j-1}+s_{j-1}$ is acceptable for the filter and x_{j-1}. Combining this with $\rho_{j-1} \geq \eta_2$ and the mechanism of Algorithm I, we obtain $\Delta_j \geq \Delta_{j-1}$. However, then iteration j cannot be the first iteration following k_1 such that (25) holds. This contradiction yields $\Delta_k \geq \gamma_0 \delta_s$ for all $k > k_1$ and the assertion follows with

$$\Delta_{\min} = \min\{\Delta_0, \ldots, \Delta_{k_1}, \gamma_0 \delta_s\}.$$

□

Lemma 4.14 (Limit of $\hat{\chi}_k$)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3 and* (13) *holds for $k \notin \mathcal{R}$. Suppose that $|\mathcal{Z}| < \infty$. Then*

$$\liminf_{k \to \infty} \hat{\chi}_k = 0$$

Proof: This is the same proof as in [6, Lemma 3.13], which is based also on Lemma 4.13. One only has to substitute the exact criticality measure χ_k used in [6] with the inexact one $\hat{\chi}_k$ considered here. □

Theorem 4.15 (Convergence to First-order Critical Point)**.** *Assume that Algorithm I is applied to the NLP* (1) *and that finite termination does not occur. Suppose that AS1–AS3, and that* (13) *holds for $k \notin \mathcal{R}$. Suppose that $|\mathcal{Z}| < \infty$. Then either the restoration procedure terminates unsuccessfully by converging to an infeasible first-order critical point of the subproblem* (8) *or there exists a subsequence $\{k_i\}$ for which*

$$\lim_{i \to \infty} x_{k_i} = x_*$$

and x_* is a first-order critical point for (1)*.*

Proof: Suppose that the restoration iteration always terminates successfully since otherwise nothing is to show. Now, consider a subsequence $\{k_i\}$ such that x_* is the limit of the subsequence x_{k_i}. Then, Lemmas 4.11, 4.12, and 4.14 ensure that there exists a subsequence $\{k_j\} \subset \{k_i\}$ such that

$$\lim_{j \to \infty} \hat{\chi}_{k_j} = 0 \tag{26}$$

and

$$\lim_{j\to\infty} x_{k_j} = x_*$$

Step 2 of Algorithm I ensures that m_k is a good model if $\hat{\chi}_{k_j} < \varepsilon_c$. Therefore, one obtains from (26) for j_0 chosen such that $\hat{\chi}_{k_j} < \varepsilon_c$ hold for all $j_0 \geq j$ that

$$\lim_{\substack{j\to\infty \\ j\geq j_0}} \chi_{k_j} = \lim_{\substack{j\to\infty \\ j\geq j_0}} \hat{\chi}_{k_j} = 0.$$

It follows that $\chi(x_*) = 0$. Hence, x_* is first-order critical. □

5 Numerical Results

To assess the performance of the inexact algorithm, we implemented an initial, simple version in C++ and present some preliminary results. We compare the inexact trust region aproach with an exact version as well as with IPOPT. Default parameters were specified following mostly the suggestion in [6], that is,

$$\begin{array}{lllll} \gamma_0 = 0.1, & \gamma_1 = 0.2, & \gamma_2 = 2, & \eta_1 = 0.01, & \eta_2 = 0.9 \\ \gamma_\theta = 10^{-4}, & \kappa_\Delta = 0.7, & \kappa_\mu = 100, & \mu = 0.01, & \kappa_\theta = 10^{-4}, \\ \kappa_{tmd} = 0.01, & \Delta_0 = 0.5 & & & \end{array}$$

together with

$$\sigma = 0.02, \quad \psi = 2.0, \quad \kappa_{en} = 1, \quad \kappa_{es} = 2 \quad \text{and} \quad \varepsilon = 0.01$$

but no parameter tuning was performed to improve performance. We also used the AD tool ADOL-C [13] to provide exact Jacobian-vector and vector-Jacobian products. Since only Hessian-vector products are needed during the step computation we also provide this second-order information exactly. Furthermore, we use the exact derivative matrices as initializations for $A_0^{\mathcal{E}}$ and $A_0^{\mathcal{I}}$. To approximate the Jacobian matrices during the optimization process, we use the TR1 quasi-Newton update as proposed in [8].

In our implementation, if the accuracy test (Step 6 of Algorithm I) fails we just compute exact Jacobians for the current iterate for the Improvement Algorithm. This is a simple but expensive approach. A more sophisticated implementation should provide a less expensive algorithm in this case. This is the subject of ongoing research and will be considered in a future study. Instead, these preliminary numerical results illustrate the viability and potential of the new inexact approach rather than a comprehensive evaluation and comparison.

					IPOPT			
		N	M	P	opt. value	it	$\#A^{\mathcal{E}}$	$\#A^{\mathcal{I}}$
1	hs014	2	1	1	1.3935	7	8	8
2	hs032	3	1	4	1.0000	16	17	17
3	hs035mod	3	1	4	0.2500	15	0	16
4	hs041	4	1	4	1.9257	9	10	0
5	hs053	5	3	10	4.0930	6	7	0
6	hs054	6	1	6	0.1929	7	8	0
7	hs060	3	1	6	0.0326	7	8	0
8	hs062	3	1	6	-26272.5145	7	8	0
9	hs073	4	1	6	29.8944	8	9	9
10	hs074	4	3	10	5126.4980	9	10	10
11	hs075	4	3	10	5174.4127	9	10	10
12	hs080	5	3	10	0.0539	6	7	0
13	hs081	5	3	10	0.0539	7	8	0
14	hs107	9	6	7	5055.0118	10	11	0
15	hs112	10	3	10	-47.7611	17	18	0
16	hs119	16	8	32	244.8997	14	15	0

Table 1: Data for CUTEr Test Problems

5.1 CUTEr Test Problems

We tested 20 CUTEr test problems that have both equalities and inequalities as constraints. Here, we report the results for 16 problems, where the variants of our trust region algorithm and IPOPT yield the same optimal values as shown in Tab. 1. We also include the IPOPT results as a baseline comparison. On two of the remaining four problems, the filter trust-region approach terminates in the restoration phase; options to improve the restoration phase were not explored in this work. The third problem terminated with an error in the solution of the QP subproblem, and the fourth problem converged to a different solution.

Tab. 1 shows the IPOPT results as a baseline. It includes the dimension of the test problems, the optimal values obtained, the iteration count (it), the number of times the exact Jacobian of the equality constraints was evaluated and the number of times the exact Jacobian of the inequality constraints was evaluated. Since we used AMPL models as interfaces to IPOPT, a presolve phase could be used to identify that some Jacobians need not be evaluated at all, as can be observed from the iteration counts in the table. Table 2 shows the iteration counts and evaluation counts of the Jacobians for the exact

	ex. TR			inex. TR V1			inex. TR V2			inex. TR V3		
	it	$\#A^{\mathcal{E}}$	$\#A^{\mathcal{I}}$	it	$\#A^{\mathcal{E}}$	$\#A^{\mathcal{I}}$	it	$\#A^{\mathcal{E}}$	$\#A^{\mathcal{I}}$	it	$\#A^{\mathcal{E}}$	$\#A^{\mathcal{I}}$
1	7	8	8	8	6	6	8	9	6	7	6	8
2	3	4	4	3	2	2	3	4	2	3	2	4
3	3	4	4	3	2	2	3	4	2	3	2	4
4	15	12	12	15	10	10	15	12	10	15	10	12
5	14	15	15	18	13	13	18	15	13	18	13	15
6	4	5	5	5	3	3	5	6	3	5	3	6
7	11	12	12	21	8	8	11	12	6	21	8	13
8	10	9	9	10	5	5	10	9	5	10	5	9
9	7	8	8	7	5	5	7	7	5	7	5	8
10	20	21	21	77	28	28	20	21	21	77	28	34
11	14	15	15	15	13	13	14	15	12	15	13	16
12	21	14	14	50	26	26	21	14	9	50	26	41
13	24	17	17	29	14	14	24	17	14	29	14	21
14	8	9	9	20	5	5	10	11	6	20	5	18
15	12	13	13	12	7	7	12	13	7	12	7	13
16	41	17	17	21	13	13	16	17	13	21	13	17

Table 2: Iteration Counts for CUTEr Test Problems

trust-region filter approach as proposed in [6] and the inexact trust-region filter algorithm as proposed in this paper. We consider three variants of this inexact approach. First, we approximate both the Jacobian of the equality constraints and inequality constraints. Second, only the approximation of the Jacobian of the equality constraints is considered together with an exact evaluation of the Jacobian of the inequalities. Finally we approximate the Jacobian of the inequality constraints together with an exact evaluation of the Jacobian of the equalities.

For all test problems we use the same parameters to handle the inexactness. Also all other parameters remain the same, except in hs074 and hs075 we set the $\Delta_0 = 5.0$ and $\gamma_2 = 10.0$ to avoid the restoration phase. As can be observed for most of the problems the iteration count does not increase much. On the other hand, the number of Jacobian evaluation is reduced and is quite pronounced on some of the problems. For a few test problems (e.g., hs074 and hs080) the iteration count increased but the number of Jacobian evaluation is still reduced. Since reducing Jacobian calculations is the focus of our problem class, this situation is still acceptable as long as the increase in the iteration count is not too large.

5.2 Binary Simulated Moving Bed Problem

We now consider the optimization of a Periodic Adsorption Process (PAP) through a simplified model for a Simulated Moving Bed (SMB) process, which separates components A and B from a liquid feed stream. Optimization of PAPs is particularly challenging because the Jacobian evaluations can require over 90% of the total cost of the optimization process [12].

As shown in Fig. 1 the process consists of six chromatographic columns, packed with solid adsorbent and arranged in four zones. The four zones are delimited by two input streams, feed Fe and desorbent De, and two exit streams, raffinate Ra (which has a high concentration of A) and extract Ex (which has a high concentration of B). Note that the first zone has one column, with flow Q_1, the second zone has two columns, with flows Q_2 and Q_3, the third zone has two columns, with flows Q_4 and Q_5 and the fourth zone has a single column, with flow Q_6. To simulate the countercurrent flow of liquid and solid in the columns, these streams are switched in a counterclockwise direction at regular time intervals of length T; this constitutes a step in the operation of the SMB. As a result, the process is never at steady state and requires periodic boundary conditions to reflect *cyclic steady state* (CSS) isotherm [12].

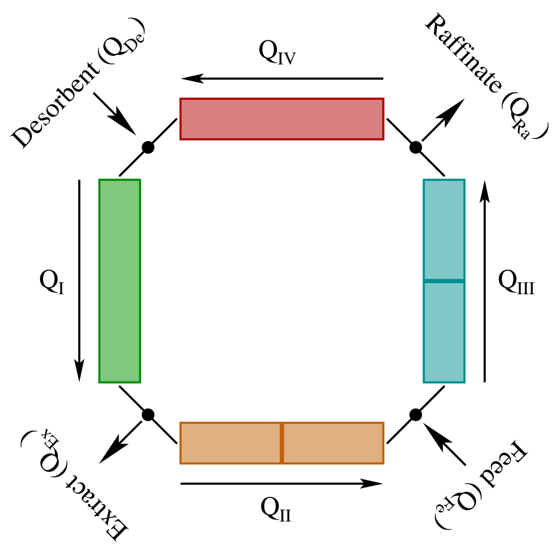

Figure 1: Simplified model of an SMB unit

The isotherm of the chromatographic columns, identified by $i = 1, \ldots, Q_{col}$, is described through an equilibrium assumption between solid and liquid phases along with a simple spatial discretization. Here, the mass balance in the liquid phase is given by:

$$\epsilon_b \frac{\partial c_i^\ell(x,t)}{\partial t} + (1-\epsilon_b)\frac{\partial q_i^\ell(x,t)}{\partial t} + (Q_i(t)/S_i)\frac{\partial c_i^\ell(x,t)}{\partial x} = 0 \tag{27}$$

with equilibrium between the liquid and solid phases given by the following isotherm:

$$q_i^\ell(x,t) = K_\ell(c_i^\ell)c_i^\ell(x,t) \tag{28}$$

Here ϵ_b is the void fraction, $c_i^\ell(x,t)$ is the concentration in the liquid phase of component $\ell \in \{A, B\}$ in column $i = 1, \ldots 6$, q_i^ℓ is the concentration in the solid phase, $K_\ell(c_i^\ell)$ is the equilibrium constant, $Q_i(t)$ and S_i are the flowrate and cross-sectional area in the ith column. We can combine (27) and (28) and rewrite the model as:

$$\frac{\partial c_i^\ell(x,t)}{\partial t} = -(Q_i(t)/S_i\bar{K}_\ell)\frac{\partial c_i^\ell(x,t)}{\partial x} \tag{29}$$

where $\bar{K}_\ell = \epsilon_b + (1-\epsilon_b)(K_\ell + \frac{dK_\ell}{dc_i^\ell}c_i^\ell)$. Dividing the column into $N_{\rm dis}$ compartments and applying a simple backward difference with $\Delta x = L/N_{\rm dis}$ leads to:

$$\frac{dc_{i,j}^\ell}{dt} = \frac{Q_i(t)N_{\rm dis}}{S_i\bar{K}_\ell L}[c_{i,j-1}^\ell(t) - c_{i,j}^\ell(t)] = k^\ell(c_{i,j}^\ell)Q_{dis}[c_{i,j-1}^\ell(t) - c_{i,j}^\ell(t)] \tag{30}$$

for $j = 1, \ldots, N_{\rm dis}$ with $c_{i,0}^\ell(t) = c_i^\ell(0,t)$ and $c_i^\ell(L,t) = c_{i,N_{\rm dis}}^\ell(t)$. Also, we initially choose a linear isotherm with $k^A = 2, k^B = 1$.

In addition, we define the state variables for this system, $c_{i,j}^\ell = x_m(t), m = 1, \ldots, 12N_{\rm dis}$, as the concentrations of A and B in the j^{th} compartment for the six columns where the index is ordered as: $m = j+(i-1)*N_{\rm dis}$ for A and $m = j+6*N_{\rm dis}+(i-1)*N_{\rm dis}$ for B. Referring to Fig. 1, one can choose the switching time T and the constant stream flows $u = [Q_1, Q_{De}, Q_{Ex}, Q_{Fe}]^T$, as independent decision variables, with the remaining flows determined from a linear mass balance: $Q_3 = Q_2 = Q_1 - Q_{Ex}$, $Q_5 = Q_4 = Q_1 - Q_{Ex} + Q_{Fe}$ and $Q_6 = Q_1 - (Q_{Ex} + Q_{Ra}) + Q_{Fe}$.

In addition, periodic boundary conditions are defined so that the column concentration profiles at the beginning of a step are equal to the profiles in the *next* column at the end of the step. The dynamic SMB model can be described by the following equations:

$$\begin{aligned} \frac{dx_m}{dt} &= g_m(x_m, u), \quad m = 1, \dots 6N_{\text{dis}} \\ x_m(0) &= x_{m+N_{\text{dis}}}(T), \quad m = 1, \dots 5N_{\text{dis}} \\ x_m(0) &= x_{m-5N_{\text{dis}}}(T), \quad m = 5N_{\text{dis}}+1, \dots 6N_{\text{dis}} \end{aligned} \tag{31}$$

Additional information on this optimization problem can be found in [12].

We study optimization problems with two different objective functions. For the first, we match a desired concentration profile in the system and for the second we maximize the feed throughput subject to the specifications on the purities of A and B in the raffinate and the extract streams, respectively. The corresponding optimization problems have the form

$$\begin{aligned} \max f(x_m(0), u, T) & \qquad \text{such that} \\ (31) \ \text{is satisfied} & \\ Purity_{Ra} \geq 0.95 & \\ Purity_{Ex} \geq 0.95 & \\ 0.01 \leq u \leq 2, & \\ 0.01 \leq T \leq 1.1, & \end{aligned}$$

where the inequalities have to be understood componentwise. For the first objective, i.e., the matching of a given profile, the optimizers found different local minima. Therefore, we modified the target function to include the required time T by using

$$f(x_m(0), u, T) = \min_{x_m(0),u,T} \sum_{m=1}^{N_{\text{dis}}} \|x_m(0) - x_m^{\text{tar}}(0)\|^2 + \alpha T^2,$$

where the penalty parameter α is chosen as 10^{-7}. The purity requirements are fulfilled by the target profile. For the second objective, we have

$$f(x_m(0), u, T) = Q_{Fe}.$$

Note that the equality constraint Jacobian from the periodic boundary conditions is dense and each function evaluation requires a forward simulation of an ODE with $12N_{\text{dis}}$ states. We have chosen a fourth order Runge-Kutta integrator with appropriate number of fixed time steps to simulate these states. This is exactly the type of simulation-based optimization problem,

				exact TR		inexact TR	
N_{dis}	n	m	# nnz($A^{\mathcal{E}}$)	iter	# $A^{\mathcal{E}}$	iter	# $A^{\mathcal{E}}$
10	125	120	15000	6	7	15	7
20	245	240	58800	6	7	12	5
30	365	360	131400	6	7	23	9
40	485	480	232800	6	7	37	14
50	605	600	363000	6	7	8	5

Table 3: Iteration Counts for Tracking-type Target, Linear Isotherm

			exact TR		inexact TR	
N_{dis}	n	m	iter	# $A^{\mathcal{E}}(x)$	iter	# $A^{\mathcal{E}}(x)$
10	125	120	120	107	166	112
20	245	240	98	88	82	59

Table 4: Iteration Counts for Maximizing the Throughput, Linear Isotherm

with many "hidden variables", expensive function evaluations and dense Jacobians, for which our novel inexact trust-region algorithm is designed.

For the optimization we set $\Delta_0 = 0.2$ and $\gamma_2 = 1.2$. All other parameter values were not changed. Table 3 considers the Tracking-type objective with a linear isotherm. It compares the exact trust-region approach described in [6] and the inexact trust region approach proposed in this paper with the iteration counts and the number of times the Jacobian $A^{\mathcal{E}}$ has to be evaluated. As can be seen, when using the inexact approach, the iteration count increases but the number of Jacobian evaluations is reduced in two cases and stays the same in another case. Only in one situation is a considerable increase in the number of Jacobian evaluations observed.

On the other hand, the maximization of the throughput is a much harder optimization problem, as can be observed from the iteration counts in Table 4. Here we considered only two discretizations. Otherwise, we would require a more sophisticated restoration routine, which is beyond the scope of this paper. For this optimization task, a considerable reduction of the Jacobian evaluations can be achieved for the finer discretization, whereas the times of Jacobian evaluation increases only slightly for the coarser discretization.

Finally, we consider a more complex SMB model with a more realistic adsorption isotherm with nonlinear functions for k^A and k^B. Here we

				exact TR		inexact TR	
N_{dis}	n	m	k	iter	# $A^{\mathcal{E}}(x)$	iter	# $A^{\mathcal{E}}(x)$
10	125	120	0.1	6	7	12	11
10	125	120	1	6	7	9	5
10	125	120	2	6	7	10	5
10	125	120	5	7	8	11	5
20	245	240	0.1	6	7	14	6
20	245	240	1	6	7	14	5
20	245	240	2	6	7	25	10
20	245	240	5	66	53	54	32
30	365	360	0.1	6	7	26	11
30	365	360	1	6	7	28	10

Table 5: Iteration Counts for Tracking-type Target, Nonlinear Isotherm

consider a Langmuir isotherm and set

$$k^A = 2\frac{2 + 2k(c^A)^2}{1 + (1 + kc^A)^2}, \qquad k^B = \frac{2 + 2k(c^B)^2}{1 + (1 + kc^B)^2}.$$

For this nonlinear isotherm with the tracking objective function, Table 5 presents the iteration counts and the number of times the Jacobian $A^{\mathcal{E}}$ has to be evaluated, both for the exact trust-region approach [6] and our inexact trust region approach. The iteration count for the inexact approach is once again higher than for the exact approach. However, we also achieve a decrease in the number of Jacobian $A^{\mathcal{E}}$ evaluations in more than half of the cases.

As a result of these three different optimization cases, we observe potential reduction of Jacobian evaluations with our inexact trust region algorithm, particularly for the more difficult nonlinear cases. These promising preliminary results motivate the implementation of a more sophisticated algorithm with more efficient and robust QP and restoration steps.

6 Conclusions and Outlook

We extend a trust-region approach for solving nonlinear optimization problems with equality and inequality constraints to the situation where only inexact information with respect to the Jacobian of the constraints is available. Assuming only that exact Jacobian-vector and vector-Jacobian products can be evaluated we provide accuracy measures that ensure global convergence to first-order optimal points, We include preliminary numerical results based on a basic implementation of the proposed algorithm. These numerical results are encouraging since the number of Jacobian evaluations can be reduced for several test cases. However, several improvements are possible and will be the subject of future work. First, an adaptive steering of the accuracy measure could enhance the efficiency of the proposed algorithm. Here, the strategy would be similar to the truncated Newton method where a higher accuracy is only needed near the solution. For the numerical results we computed a new exact Jacobian each time the accuracy test fails. Here, more sophiticated model improvement strategies would be advantageous. This issue will also be the focus of future research.

References

[1] R. Byrd. Robust trust region methods for constrained optimization, Houston, USA. Third SIAM Conference on Optimization, 1987.

[2] R.H. Byrd, F.E. Curtis, and J. Nocedal. An inexact Newton method for nonconvex equality constrained optimization. *Math. Program.*, 122(2 (A)):273–299, 2010.

[3] A. Conn, N. Gould, and Ph. Toint. *Trust-region methods.* SIAM, 2000.

[4] A. Conn, K. Scheinberg, and L. Vicente. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM J. Opt.*, 20, 2009.

[5] F.E. Curtis, O. Schenk, and A. Wächter. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. *SIAM J. Sci. Comput.*, 32(6):3447–3475, 2010.

[6] R. Fletcher, N. Gould, S. Leyffer, P. Toint, and A. Wächter. Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *SIAM J. Optim.*, 13(3):635–659, 2002.

[7] R. Fletcher, S. Leyffer, and P. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59, 2002.

[8] A. Griewank and A. Walther. On constrained optimization by adjoint-based quasi-Newton methods. *Optim. Methods Softw.*, 17:869–889, 2002.

[9] M. Heinkenschloss and D. Ridzal. A matrix-free trust-region sqp method for equality constrained optimization. Technical Report TR11-17, CAAM, Rice University, 2011.

[10] L. Jiang, L.T. Biegler, and G. Fox. Optimization of pressure swing adsorption systems for air separation. *AIChE Journal*, 49:1140–1157, 2003.

[11] E. Omojokun. *Trust region algorithms for optimization with nonlinear equality and inequality constraints.* PhD thesis, Dept. of Computer Science, University of Colorado, 1989.

[12] S. R. Vetukuri, L. T. Biegler, and A. Walther. An inexact trust-region algorithm for the optimization of periodic adsorption processes. *I & EC Research*, 49:12004 – 12013, 2010.

[13] A. Walther and A. Griewank. *Combinatorial Scientific Computing*, chapter Getting Started with ADOL-C, pages 181–202. Chapman-Hall CRC Computational Science, 2012.

[14] A. Walther, S.R.R. Vetukuri, and L.T. Biegler. A first-order convergence analysis of trust-region methods with inexact Jacobians and inequality constraints. *Optimization Methods and Software*, 27(2):373–389, 2012.

[15] J.C. Ziems and S. Ulbrich. Adaptive multilevel inexact SQP methods for PDE-constrained optimization. *SIAM J. Optim.*, 21(1):1–40, 2011.