Contents

1 Variable Metric Bundle Method

A way to extend the proximal bundle method is to use an arbitrary metric $\frac{1}{2}\langle d, W_k d\rangle$ with a symmetric and positive definite matrix W_k instead of the Euclidean metric for the stabilization term $\frac{1}{2t_k}\|d\|^2$. Methods doing so are called *variable metric bundle methods*. This section combines the method of Hare et al. presented in section **??** with the second order model function used by Noll in [10] to a metric bundle method suitable for nonconvex functions with noise.

The section starts by explaining the ideas from [10] used to extend the method presented above. It then gives an explicit strategy how to update the metric during the steps of the algorithm and concludes with a convergence proof for the developed method.

Throughout this section we still consider the optimization problem (**??**). We also keep the names and definitions of the objects used in section **??**.

1.1 Main Ingredients to the Method

As already mentioned in section **??** the stabilization term can be interpreted in many different ways. In the context of this section we can understand it as a pretty rough approximation of the curvature of the objective function. Of course bundle methods are designed to work with non differentiable objectives so it cannot be expected that the function provides any kind of curvature. However, if it has regions where there is curvature, this information can be used to speed up convergence.

1.1.1 Variable Metric Bundle Methods

Variable metric bundle methods use an approach that can be motivated by the thoughts stated above. Instead of using the Euclidean norm for the stabilization term $\frac{1}{2}\|d\|^2$ the metric is derived from a symmetric and positive definite matrix W_k. As the name of the method suggests, this matrix can vary over the iterations of the algorithm. The subproblem in the k'th iteration therefore reads

$$\min_{\hat{x}^k+d\in\mathbb{R}^n} M_k(\hat{x}^k+d) + \mathtt{i}_X(\hat{x}^k+d) + \frac{1}{2}\langle d, W_k d\rangle .$$

As explained in [4] like (**??**) this is a Moreau-Yosida regularization of the objective function (on the constraint set), so this subproblem is still strictly convex and has a unique

solution. It is however harder to solve especially if the matrices W_k are no diagonal matrices [7]. In the unconstrained case or for a very simple constraint set the subproblem can be solved by calculating a quasi Newton step. Such a method is presented by Lemaréchal and Sagastizábal in [5] for convex functions. Lukšan and Vlček use an algorithm in those lines in [13] which is adapted to a limited memory setting by Haarala et al. in [1].

A challenging question is how to update the matrices W_k. It is important that the updating strategy preserves positive definiteness of the matrices and that the matrices stay bounded. The updates that are used most often are the symmetric rank 1 formula (SR1 update) and the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update. These updates make it possible to assure the required conditions with only little extra effort even in the nonconvex case. Concrete instances of the updates are given in [13] and [4].

1.1.2 Noll's Second Order Model

In [11] Noll et al. present a proximal bundle method for nonconvex objective functions. An important ingredient to the method is that not the objective function itself is approximated in the subproblem but a quadratic model of it:

$$\Phi(x, \hat{x}) = \phi(x, \hat{x}) + \frac{1}{2}\langle x - \hat{x}, Q(\hat{x})(x - \hat{x})\rangle \tag{1.1}$$

The first order model $\phi(\cdot, \hat{x})$ is convex and possibly nonsmooth. The second order part $\frac{1}{2}\langle \cdot - \hat{x}, Q(\hat{x})(\cdot - \hat{x})\rangle$ is quadratic but not necessarily convex.

As the first order part of this model is convex it can be approximated by a cutting plane model just like the objective function in usual convex bundle methods. The subproblem emerging from this approach is

$$\min_{\hat{x}^k + d} m(\hat{x}^k + d) + \frac{1}{2}\left\langle d, Q(\hat{x}^k)d\right\rangle + \frac{1}{2t_k}\|d\|^2$$

where m_k is the cutting plane model (**??**) for the nonsmooth function ϕ.

The matrix $Q(\hat{x})$ itself does not have to be positive definite. In fact the only conditions put on this matrix are that it is symmetric and that all eigenvalues are bounded. We adopt the notation in [10] and write

$$Q(\hat{x}^k) := Q_k = Q_k^\top \quad \text{and} \quad -q\mathbb{I} \prec Q_k \prec q\mathbb{I} \text{ for } q > 0.$$

The notation $A \prec B$ with $A, B \in \mathbb{R}^{n\times n}$ means that the matrix $(B-A)$ is positive definite.

As the matrix Q_k is symmetric it can also be pulled into the stabilization term. The k'th bundle subproblem then is

$$\min_{\hat{x}^k+d\in X} M_k(\hat{x}^k+d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle. \tag{1.2}$$

If $W_k = Q_k + \frac{1}{t_k}\mathbb{I}$ is positive definite, this is a variable metric subproblem.

The decomposition of the stabilization term into a curvature approximation and a proximal term makes is easier to reach two goals at the same time:

One the one hand, curvature of the objective can be approximated only under the conditions of the boundedness and symmetry of Q_k. No positive definiteness has to be ensured for convergence. On the other hand the proximal term can be used in the trust region inspired way to make a line search obsolete. As already mentioned in section **??** this is an advantage especially when working with inexact functions where a line search is not useable.

comment on line search and curve search in [4, 5, 13]?

1.1.3 The Descent Measure

Due to the different formulation of the subproblem (1.2) the descent measure δ_k has to be adapted in the varible metric bundle method. In the same way as for (**??**) from the optimality condition

$$0 \in \partial M_k(x^{k+1}) + \partial \mathtt{i}_D(x^{k+1}) + \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k$$

follows that

$$S^k + \nu^k = -\left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k. \tag{1.3}$$

S^k and ν^k being the augmented aggregate subgradient and outer normal defined in (**??**) and (**??**) respectively.

From this the model decrease (**??**) can be recovered using (**??**), (**??**) and (1.3):

$$\begin{aligned}\delta_k &= \hat{f}_k - M_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\ &= \hat{f}_k - A_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\ &= C_k - \left\langle S^k + \nu^k, d^k \right\rangle \\ &= C_k + \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle .\end{aligned} \tag{1.4}$$

The new δ_k is used in the same way as in algorithm **??**.1 for the descent test and stopping conditions.

Because the changes in the algorithm concern only the stabilization and the decrease measure δ_k all other relations that were obtained for the different parts of the model M_k in section **??** are still valid.

1.2 The Variable Metric Bundle Algorithm

The variable b=metric bundle algorithm can now be stated as a varaiation of algorithm **??**.1.

same form as Hare algorithm (nullstep)
add Q calculation

Algorithm 1.1: Nonconvex Variable Metric Bundle Method with Inexact Information

---

Select parameters $m \in (0,1), \gamma > 0$ and a stopping tolerance $\texttt{tol} \geq 0$.
Choose a starting point $x^1 \in \mathbb{R}^n$ and compute f_1 and g^1. Set the initial metric matrix $Q = \mathbb{I}$, the initial index set $J_1 := \{1\}$ and the initial prox-center to $\hat{x}^1 := x^1$, $\hat{f}_1 = f_1$ and select $t_1 > 0$.
For $k = 1, 2, 3, \ldots$

1. Calculate

$$d^k = \arg\min_{d \in \mathbb{R}^n} \left\{ M_k(\hat{x}^k + d) + \mathbb{I}_X(\hat{x}^k + d) + \frac{1}{2} d^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d \right\}.$$

2. Set

$$G^k = \sum_{j \in J_k} \alpha_j^k s_j^k,$$

$$C_k = \sum_{j \in J_k} \alpha_j^k c_j^k,$$
$$\delta_k = C_k + (d^k)^\top \left(Q + \frac{1}{t_k} \mathbb{I} \right) d^k.$$

If $\delta_k \leq \texttt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.
4. Compute f^{k+1}, g^{k+1}.
   If
   $$f^{k+1} \leq \hat{f}^k - m\delta_k \quad \rightarrow \text{ serious step}$$
   Set $\hat{x}^{k+1} = x^{k+1}, \hat{f}^{k+1} = f^{k+1}$ and select $t_{k+1} > 0$.
   Calculate $Q(\hat{x}^k)$ ... Otherwise $\rightarrow$ nullstep
   Set $\hat{x}^{k+1} = \hat{x}^k, \hat{f}^{k+1} = f^{k+1}$ and choose $0 < t_{k+1} \leq t_k$.
5. Select new bundle index set J_{k+1}, keeping all active elements. Calculate
   $$\eta_k = \max \left\{ \max_{j \in J_{k+1}, x^j \neq \hat{x}^{k+1}} \frac{-2e_j^k}{|x^j - \hat{x}^{k+1}|^2}, 0 \right\} + \gamma$$
   and update the model M^k.

1.3 Convergence Analysis

ausfuhrungen im Beweis genauer

In this section the convergence properties of the new method are analyzed. We do this the same way it is done by Hare et al. in [3].

In the paper all convergence properties are first stated in [3, Lemma 5]. It is then shown that all sequences generated by the method meet the requirements of this lemma which we repeat here for convenience.

Lemma 1.1 ([3, Lemma 5]) *Suppose that the cardinality of the set* $\{j \in J^k | \alpha_j^k > 0\}$ *is uniformly bounded in* k.

(i) If $C^k \rightarrow 0$ *as* $k \rightarrow \infty$, *then*

$$\sum_{j \in J^k} \alpha_j^k \|x^j - \hat{x}^k\| \rightarrow 0 \text{ as } k \rightarrow \infty.$$

(ii) If additionally for some subset $K \subset \{1, 2, \dots\}$,

$$\hat{x}^k \to \bar{x}, S^k \to \bar{S} \text{ as } K \ni k \to \infty, \text{ with } \{\eta_k | k \in K\} \text{ bounded,}$$

then we also have

$$\bar{S} \in \partial f(\bar{x}) + B_{\bar{\theta}}(0).$$

(iii) If in addition $S^k + \nu^k \to 0$ *as* $K \in k \to \infty$, *then* $\bar{x}$ *satisfies the approximate stationarity condition*

$$0 \in (\partial f(\bar{x}) + \partial \mathtt{i}_X(\bar{x})) + B_{\bar{\theta}}(0). \tag{1.5}$$

(iv) Finally if f *is also lower-*$\mathcal{C}^1$, *then for each* $\varepsilon > 0$ *there exists* $\rho > 0$ *such that*

$$f(y) \geq f(\bar{x}) - (\bar{\theta} + \varepsilon)\|y - \bar{x}\| - 2\bar{\sigma}, \quad \text{for all } y \in X \cup B_\rho(\bar{x}). \tag{1.6}$$

As neither the stabilization nor the descent test is involved in the proof of Lemma 1.1 it is the same as in [3].

We prove now that also the variable metric version of the algorithm fulfills all requirements of Lemma 1.1. The proof is divided into two parts. The first case covers the case of infinitely many serious steps, the second one considers infinitely many null steps.

For both proofs the following lemma is needed:

Lemma 1.2 *For a symmetric matrix* $A \in \mathbb{R}^{n \times n}$, *a vector* $d \in \mathbb{R}^n$ *and* $\xi > 0$ *the following result holds:*

$$A \prec \xi \mathbb{I} \Rightarrow Ad < \xi d.$$

The second inequality is considered componentwise.

Proof: As the matrix A is real and symmetric it is orthogonally diagonalizeable. There exist eigenvalues $\lambda_i \in \mathbb{R}, i = \{1, ..., n\}$ and corresponding eigenvectors $v^i \in \mathbb{R}^n, i = \{1, ..., n\}$ that satisfy the equations

$$Av^i = \lambda_i v^i \quad i = \{1, ..., n\}.$$

The eigenvectors v^i generate a basis for $\mathbb{R}^n$ so any vector $d \in \mathbb{R}^n$ can be written as

$$d = \sum_i \alpha_i v^i$$

for $\alpha_i \in \mathbb{R}^n, i = \{1, ..., n\}$.

This yields

$$Ad = A \sum_i \alpha_i v^i = \sum_i \alpha_i \lambda_i v^i. \tag{1.7}$$

Plugging the assumption $A \prec \xi \mathbb{I}$ which is equivalent to $\max_i \lambda_i < \xi$ into (1.7) we get relation (1.3) by

$$Ad < \xi \sum_i \alpha_i v^i = \xi d.$$

□

Theorem 1.3 (c.f.[3, Theorem 6]) *Let the algorithm generate and infinite number of serious steps. Then $\delta_k \to 0$ as $k \to \infty$.*
Let the sequence $\{\eta_k\}$ be bounded. If $\liminf_{k\to\infty} t_k > 0$ then as $k \to \infty$ we have $C_k \to 0$, and for every accumulation point $\bar{x}$ of $\{\hat{x}^k\}$ there exists $\bar{S}$ such that $S^k \to \bar{S}$ and $S^k + \nu^k \to 0$.
In particular if the cardinality of $\{j \in J^k | \alpha_j^k > 0\}$ is uniformly bounded in k then the conclusions of Lemma 1.1 hold.

The proof is very similar to the one stated in [3] but minor changes have to be made due to the different formulation of the nominal decrease δ_k.

Proof: At each serious step we have

$$\hat{f}_{k+1} \leq \hat{f}_k - m\delta_k \tag{1.8}$$

where $m,\ \delta_k > 0$. From this follows that the sequence $\{\hat{f}_k\}$ is nonincreasing. Since $\{\hat{x}^k\} \subset X$ and f is continuous the sequence $f(\hat{x}^k)$ is bounded. With $|\sigma_k| < \bar{\sigma}$ the sequence $\{f(\hat{x}^k) + \sigma_k\} = \{\hat{f}_k\}$ is bounded below. Together with the fact that $\{\hat{f}_k\}$ is nonincreasing one can conclude that it converges.
Using (1.8), one obtains

$$0 \leq m \sum_{k=1}^{l} \delta_k \leq \sum_{k=1}^{l} \left(\hat{f}_k - \hat{f}_{k+1} \right),$$

so letting $l \to \infty$,

$$0 \leq m \sum_{k=1}^{\infty} \delta_k \leq \hat{f}_1 - \underbrace{\lim_{k\to\infty} \hat{f}_k}_{\neq \pm\infty}.$$

This yields

$$\sum_{k=1}^{\infty} \delta_k = \sum_{k=1}^{\infty} \left(C^k + (d^k)^\top \left(Q + \frac{1}{t_k}\mathbb{I} \right) d^k \right) < \infty.$$

Hence, $\delta_k \to 0$ as $k \to \infty$. All quantities above are nonnegative due to positive definiteness of $Q + \frac{1}{t_k}\mathbb{I}$, so it also holds that

$$C_k \to 0 \quad \text{and} \quad (d^k)^\top \left(Q + \frac{1}{t_k}\mathbb{I} \right) d^k \to 0.$$

For any accumulation point $\bar{x}$ of the sequence $\{\hat{x}^k\}$ the corresponding subsequence $d^k \to 0$ for $k \in K \subset \{1, 2, ...\}$. As $\liminf_{k\to\infty} t_k > 0$ and the eigenvalues of Q are bounded the whole expression

$$S^k + \nu^k = \left(Q + \frac{1}{t_k} I \right) d^k \to 0 \quad \text{for} \quad k \in K.$$

And from local Lipschitz continuity of f follows then that $S^k \to \bar{S}$ for $k \in K$.

□

For the case of infinitely many null steps we need result (31) from [3]. It only depends on the definitions of the augmented linearization error and subgradient.

Whenever x^{k+1} is as declared a null step, the relation

$$-c_{k+1}^{k+1} + \left\langle s_{k+1}^{k+1}, x^{k+1} - \hat{x}^k \right\rangle \geq -m\delta_k \tag{1.9}$$

holds.

Theorem 1.4 (c.f. [3, Theorem 7]) *Let a finite number of serious iterates be followed by infinite null steps. Let the sequence* $\{\eta_k\}$ *be bounded and* $\liminf_{k\to\infty} t_k > 0$.
Then $\{x^k\} \to \hat{x}$, $\delta_k \to 0$, $C_k \to 0$, $S^k + \nu^k \to 0$ *and there exist* $K \subset \{1, 2, ...\}$ *and* $\bar{S}$ *such that* $S^k \to \bar{S}^k$ *as* $K \ni k \to \infty$.
In particular if the cardinality of $\{j \in J^k | \alpha_j^k > 0\}$ *is uniformly bounded in* k *then the conclusions of Lemma 1.1 hold for* $\bar{x} = \hat{x}$.

Proof: Let k be large enough such that $k \geq \bar{k}$, where $\bar{k}$ is the iterate of the last serious

step. Let $\hat{x}^k = \hat{x}$ and $\hat{f}_k = \hat{f}$ be fixed. Define the optimal value of the subproblem (1.2) by

$$\Psi_k := M_k(x^{k+1}) + \frac{1}{2}\left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k. \tag{1.10}$$

It is first shown that the sequence $\{\Psi_k\}$ is bounded above. From definition (**??**) follows

$$A_k(\hat{x}) = M_k(x^{k+1}) - \langle S^k, d^k\rangle.$$

Using (1.3) for the second equality, the subgradient inequality for $\nu^k \in \partial \mathtt{i}_D$ in the first inequality and (**??**) for the second inequality one obtains

$$\begin{aligned}
\Psi^k + \frac{1}{2}\left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k &= A_k(\hat{x}) + \langle S^k, d^k\rangle + \left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \\
&= A_k(\hat{x}) - \langle \nu^k, k\rangle \\
&\leq A(\hat{x}) \\
&\leq M_k(\hat{x}) \\
&= \hat{f}.
\end{aligned}$$

By boundedness of d^k and $Q + \frac{1}{t_k}\mathbb{I}$ this yields that $\Psi_k \leq \hat{f}$, so the sequence $\{\Psi_k\}$ is bounded above. In the next step is shown that $\{\Psi_k\}$ is increasing. For this we obtain

$$\begin{aligned}
\Psi_{k+1} &= M_k(x^{k+2}) + \frac{1}{2}\left(d^{k+1}\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1} \\
&\geq A_k(x^{k+2}) + \frac{1}{2}\left(d^{k+1}\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1} \\
&= M_k(x^{k+1}) + \langle S^k, x^{k+2} - x^{k+1}\rangle + \frac{1}{2}\left(d^{k+1}\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1} \\
&= \Psi_k - \frac{1}{2}\left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k + \frac{1}{2}\left(d^{k+1}\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1} \\
&\quad - \left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right) - \langle \nu^k, x^{k+2} - x^{k+1}\rangle \\
&\geq \Psi_k + \frac{1}{2}\left(d^{k+1} - d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right) \\
&= \Psi_k + \frac{1}{2}\underbrace{\|d^{k+1} - d^k\|_{Q+\frac{1}{t_k}\mathbb{I}}}_{\geq 0},
\end{aligned}$$

where the first inequality comes from (**??**) and the fact that $t_{k+1} \leq t_k$ for null steps. The second equality follows from (**??**), the third equation by (1.3) and (1.10) and the last inequality holds by $\nu^k \in \partial \mathtt{i}_X(x^{k+1})$.

As Q is fixed in null steps and $\liminf_{k\to\infty} t_k > 0$ the sequence$\{\Psi_k\}_{k\in\mathbb{N}}$ is increasing and bounded from above. The sequence is therefore convergent. Taking into account that $1/t_k \geq 1/t_{\bar{k}}$, it therefore follows that

$$\|d^{k+1} - d^k\| \to 0, \quad k \to \infty. \tag{1.11}$$

By the first line in (1.4) and the fact that the augmented aggregate error can be expressed as

$$C_k = \hat{f} - M_k(x^{k+1}) + \left\langle S^k, d^k \right\rangle$$

by the KKT conditions follows

$$\begin{aligned}
\hat{f} &= \delta_k + M_k(\hat{x}) - C_k - \left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^k\right) \\
&= \delta_k + M_k(x^{k+1}) - \langle S^k, d^k\rangle - \left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^k\right) \\
&= \delta_k + M_k(\hat{x} + d^k) + \left\langle \nu^k, d^k \right\rangle \\
&\geq \delta_k + M_k(\hat{x} + d^k)
\end{aligned}$$

Where the last inequality is given by $\nu^k \in \partial \mathtt{i}_X(x^{k+1})$. Therefore

$$\delta^{k+1} \leq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}). \tag{1.12}$$

By assumption (**??**) on the model, written for $d = d^{k+1}$,

$$-\hat{f}_{k+1} + c_{k+1}^{k+1} - \left\langle s_{k+1}^{k+1}, d^{k+1} \right\rangle \geq -M_{k+1}(\hat{x} + d^{k+1}).$$

In the null step case it holds $\hat{f}_{k+1} = \hat{f}$ so adding condition (1.9) to the inequality above, one obtains that

$$m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \geq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}).$$

Combining this relation with (1.12) yields

$$0 \leq \delta_{k+1} \leq m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle.$$

Because $m \in (0,1)$ and $\left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \to 0$ as $k \to \infty$ due to (1.11) and the boundedness of $\{\eta_k\}$ using [12, Lemma 3, p.45] it follows from (1.3) that

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

From formulation (1.4) of the model decrease follows that $C_k \to 0$ as $k \to \infty$. Since $Q + \frac{1}{t_k}\mathbb{I} \succ \xi\mathbb{I}$ due to $\liminf_{k\to\infty} t_k > 0$ and the bounded eigenvalues of Q we have

$$\xi \left(d^k\right)^\top d^k \leq \left(d^k\right)^\top \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \to 0$$

This means that $d^k \to 0$ for $k \to \infty$ and therefore $\lim_{k\to\infty} x^k = \hat{x}$. It also follows that $\|S^k + \nu^k\| \to 0$ as $k \to \infty$. Passing to some subsequence if necessary we can conclude that S^k converges to some $\bar{S}$ and as $\hat{x}^k = \bar{x}$ for all k all requirements of Lemma 1.1 are fulfilled.

□

Remark: All results deduced in section **??** are still valid for this algorithm as they do not depend on the kind of stabilization used.

1.4 Updating the Metric

In [11] and [10] it is not specified how the matrix Q_k is to be chosen. For convergence it is necessary that the eigenvalues of Q_k are bounded. Additionally the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ has to be positive definite. Here we present two possible versions to update the metric matrix Q_k that fulfill both conditions.

Both updates are based on the usual BFGS-update formula (named after Broyden, Goldfarb, Fletcher and Shanno)

$$Q_{k+1} = Q_k + \frac{y^k {y^k}^\top}{\langle y^k, d^k \rangle} - \frac{Q_k d^k (Q_k d^k)^\top}{\langle d^k, Q_k d^k \rangle}. \tag{1.13}$$

Usually y^k is defined as the difference of the last two gradients of f. To adapt the formula

to the nondifferentiable case the difference $y^k := g^{k+1} - g^k$ of two subgradients of f is taken instead as proposed in [1]. The starting matrix $Q_1 = \mathbb{I}$.

By definition the BFGS update is symmetric. To assure boundedness of the matrix Q_{k+1} the updates can be manipulated in the following ways:

1.4.1 Scaling of the Whole Matrix

A simple way to keep the absolute value all of eigenvalues of the constructed matrix Q_k below some threshold $0 < q < \infty$ is to scale the whole matrix down as soon as the absolute value of one eigenvalue is larger than this number. To do this define $\lambda_{max} := \max\{|\lambda_i| \mid \lambda_i \text{ is eigenvalue of } Q_k|\}$. If $\lambda_{max} > q$, set $Q_k = \frac{q}{\lambda_{max}} Q_k$. This way the absolute value of all eigenvalues is always smaller or equal to q. An advantage of this method is besides its simplicity that by scaling the whole matrix the ratio of the eigenvalues of Q_k is preserved. Scaling of Q_k corresponds to shrinking the whole quadratic function and in this way also the 'ratio of curvature' at different points of the graph stays the same.

1.4.2 Adaptive Scaling of Single Eigenvalues

This second method is motivated by the following observation: The variable metric bundle algorithm is to be used for nonsmooth functions. This means that the objective function has some kinks. For locally Lipschitz functions the number of kinks is finite and in between the kinks the function is smooth. This means, that there is indeed curvature information present in the smooth parts of the functions. At the kinks however the curvature is not defined. Taking a look at the one-sided differential quotient for $x \in \mathbb{R}$ shows that it diverges at such points: Let a kink be at $x_{\text{kink}} \in \mathbb{R}$ and the left sided limiting value of the derivative be $f'(x_{\text{kink}}) = a$ the right-sided one $f'(x_{\text{kink}}) = b \neq a$. Because f was assumed to be locally Lipschitz both limits exist and are finite. The following quotient can then be stated:

$$\lim_{h \searrow 0} \frac{f'(x_{\text{kink}} + h) - f'(x_{\text{kink}})}{h} = \frac{\overbrace{f'(x_{\text{kink}} + h)}^{\to b} - a}{h} = \pm\infty,$$

the sign depending on if $a > b$ or vice versa.

In more dimensions this is the same for the components of the Hessian corresponding to the direction where the kink occurs. This supports also to the intuitive thought that at a kink the slope changes 'infinitly fast'. Numerically the BFGS-update (1.13) can result

in very large values for the entries of Q_k corresponding to points near the kink.

On the other hand due to the local Lipschitz property the slope of the objective function is always finite on closed sets. This means that there exists a neighborhood $B_\varepsilon(x_{\text{kink}})$ where the function f behaves similar to the scaled modulus $a|\cdot|$, $a \in \mathbb{R}$, in the direction perpendicular to the kink. Therefore in this neighborhood almost no curvature is present.

Summarized this means that on the one hand, the matrix Q_k should be close to zero in the components representing the directions perpendicular to the kink as soon as the iterates approach x_{kink}. But contrary to that the method that constructs Q_k can give very high values for those components.

The idea is now to scale only those eigenvalues of Q_k that are especially large. To lower the probability that eigenvalues are chosen which represent a really existing large curvature of of the objective function at the observed point, the change of the eigenvalues is viewed in relation to the step size d^k. As the eigenvalues of a matrix $A \in \mathbb{R}^n$ depend continuously on the entries of the matrix (see for example [14, Theorem 1.2]) the change in the eigenvalues could be calculated by ordering them by size and then calculating $\text{diff}_{\lambda_i} = |\lambda_i^{k-1} - \lambda_i^k|, i = 1, ..., n$, where λ^{k-1} and $\lambda^k \in \mathbb{R}^n$ are vectors with the sorted eigenvalues of Q_{k-1} and Q_k respectively as components. If $\frac{\text{diff}_{\lambda_i}}{\|d^k\|} > q$ for a threshold $q < \infty$ and any $i = 1, ..., n$, indicating that the i'th eigenvalue changes much while only a small step is done, the i'th eigenvalue is set to $\lambda_i^k = \frac{1}{2}\lambda_1^{k-1}$ (remember that we stated before that near a kink there is often a very small curvature, hence halving the eigenvalue of the matrix of the step before).

Because this method considers only the relative change of the eigenvalues, it is still possible that one or more of them is larger than the threshold q. If this is the case, the whole matrix can be scaled as described above.

This technique however lacks a practical method of implementation and has also some theoretical issues. The most important issue is that eigenvalues do not have any intrinsic order. Sorting them by size may work in most cases due to continuity but at least in the situation when an eigenvalue is scaled this is not a continuous operation any more. This can result in the eigenvalues being sorted differently in the next iteration such that all differences of eigenvalues that follow the one that contains the scaled one are not differences of corresponding eigenvalues.

write only the thing below; write things above as comment in third part of this section???

Another possibility is to not work with the change of the eigenvalues but with their absolute size like in the method presented before; advantage: no additional scaling necessary

any more

Remark: There appear many parameters to control the scaling of the metric update. Although these parameters were not especially tuned in this thesis they have a considerable impact on the convergence speed of the method also depending on the objective function used. This has to be kept in mind when implementing the method in practice.

1.4.3 Other Updating Possibilities

There are certainly many other possibilities to update the metric Q_k. A third variation based on BFGS-updates is the limited memory update suggested in [9]. If the update is skipped whenever $\|\rho s s^\top\| = \frac{\|d^k\|}{\|y^k\|} > q$ the matrix Q_k stays bounded. This strategy is also supported by the fact that if $\frac{\|d^k\|}{\|y^k\|} > q$ the change in the subgradient relative to the step size is rather small indicating that the current iterate lies within a region with only small changes in curvature. In such regions the update can be skipped.

Another updating method is using the symmetric-rank-1 (SR1) update

$$Q_k = Q_{k-1} + \frac{(y^k - Q_k d^k)(y^k - Q_k d^k)^\top}{\langle y^k - Q_k d^k, d^k \rangle}.$$

Boundedness can be assured in the same as for the normal BFGS update.

In [1] where the metric matrix is updated also in null steps a BFGS update is used in serious steps and an SR1 update in null steps.

As a last remark on this topic we want to say that although in this thesis the adaption of update strategies originally developed to be used with gradients seems to work in the presented setting also with subgradients this does not always have to be the case. Although it is argued for example in [6] that locally Lipschitz functions are differentiable almost everywhere and with an adequate linesearch it is improbable to arrive at an iterate that is a nondifferentiable point of the objective function this can still happen. Especially if such a linesearch is not used like in the algorithm presented here. So despite the promising practical behavior this area is still open to research.

- Q only updated in serious steps - why?
- comment in SR1 update? Null steps?

1.5 Numerical Testing

To compare the proximal bundle algorithm **??**.1 with its variable metric variant algorithm 1.1 it is tested on some academic test functions and on a set of lower-$\mathcal{C}^2$ functions in different dimensions. The tests are done with the following parameters given in [3]: $m = 0.05$, $\gamma = 2$ and $t_0 = 0.1$. The chosen stopping tolerance is $\texttt{tol} = 10^{-6}$. If the algorithms do not meet the stopping condition after $250n$ steps for $x \in \mathbb{R}^n$, they are terminated. Contrary to [3] the stopping test is taken as given in the algorithm and the tolerance not multiplied by $1 + \hat{f}_k$. The proximity control parameters κ_- and κ_+ from (**??**) and (**??**) respectively are chosen as $\kappa_- = 0.8$ and $\kappa_+ \in \{1.2, 2\}$.

-> check what is missing of introduchtion

To test the performance for inexact function and subgradient values different types of noise are introduced. This is done by adding randomly generated elements with norm less or equal to σ_k and θ^k to the exact values $f(x^{k+1})$ and $g(x^{k+1})$ respectively.

Five different forms of noise are tested:

- N_0: No noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta^k = 0$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{f,g}}$: Constant noise, $\bar{\sigma} = \sigma_k = 0.01$ and $\bar{\theta} = \theta^k = 0.01$ for all k,
- $\mathrm{N}_\mathrm{v}^{\mathrm{f,g}}$: Vanishing noise, $\bar{\sigma} = 0.01, \sigma_k = \min\{0.01, \|x^k\|/100\}$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{g}}$: Constant subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta_k = 0.01$ for all k and
- $\mathrm{N}_\mathrm{v}^{\mathrm{g}}$: Vanishing subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k.

The exact case is used for comparison. The constant noise forms represent cases where the inexactness is outside of the optimizer's control. The vanishing noise forms represent cases where the noise can be controlled but the mechanism is considered expensive, so it is only used when approaching the minimum. The two forms of subgradient noise represent the case where the subgradient is approximated numerically.

To address the randomness of the noise the tests for the last four noise forms were executed how many??? times and the results averaged. -> write this for everz experiment on its own.

To compare the performance of the different methods the accuracy is measured by

$$\text{accuracy} = |\log_{10}(\hat{f}_{\bar{k}})|.$$

$\hat{f}_{\bar{k}}$ being the current $\hat{f}_k$ when the algorithm stops.

1.5.1 Academic Test Examples

For the comparison in this section the two BFGS update rules for Q_k presented in section 1.4 are used. The algorithm using the scaled BFGS update is called "Variable Metric BFGS" and the algorithm that uses the adaptive method only scaling single eigenvalues is referred to as "Varible Metric BFGS Adaptive".

To explore the benefit of the matrix Q the algorithms **??**.1 and 1.1 are tested on a smooth and a nonsmooth version of a badly conditioned parabola. The smooth test function is

$$p(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \langle x, Ax \rangle ,$$

where the matrix is chosen as $A = \begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix}$. The condition number of this matrix is $\kappa_A = \frac{\lambda_{max}}{\lambda_{min}} = 50$, where $\lambda_{min}, \lambda_{max}$ are the smallest and largest eigenvalue of A respectively. From smooth optimization it is known that gradient descent methods have a rather poor convergence rate for such badly conditioned matrices (c.f. Chapter 7.4 in [8]). Figure 1 shows the sequences of serious iterates resulting from the two algorithms on the contour lines of the parabola. On the left the complete sequence is depicted. The plot on the right shows a detail of the left figure near the minimum of the objective. As the descent direction taken in algorithm **??**.1 is an aggregate subgradient and second order information is only provided by the stabilization term $\frac{1}{t_k}\|d\|^2$ we can see a zig-zagging behavior of the sequence for the parabola in figure 1. Contrary to that the sequence of serous iterates provided by algorithm 1.1 can take advantage of the second order information provided by Q. It approaches the minimum almost in a straight line. The difference in behavior of the two algorithms is especially visible on the detail plot of 1 that shows the situation near the minimum: The proximal bundle algorithm needs a lot of steps circling around the minimum whereas the variable metric algorithm approaches the minimum directly. The resulting advantage of this behavior is the smaller number of steps needed by the variable metric algorithm.

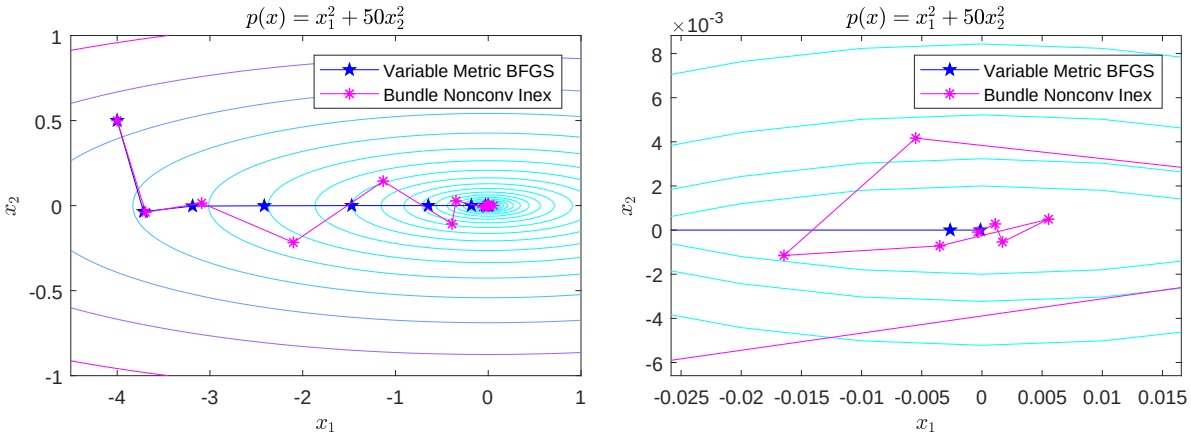

Figure 1: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of parabola p. The right image is a detail of the plot on the left.*

The second test function is a nonsmooth version of the above parabola. The function is given by

$$p_n(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \left\langle \frac{1}{2}x, Ax \right\rangle + \frac{1}{2}|x_1| + 25|x_2|.$$

Due to the kink along the x_1-axis the curvature information supplied by Q is less reliable than for the smooth parabola. Figure 2 shows the sequences constructed by the two algorithms. Still the sequence provided by the variable metric algorithm does less zig-zagging than the one coming from the proximal bundle algorithm. It is interesting to note, that the sequence provided by the proximal bundle algorithm is the same for both functions. This is not the case for the sequence generated by the metric bundle algorithm because the second order information of the two objective functions is different.

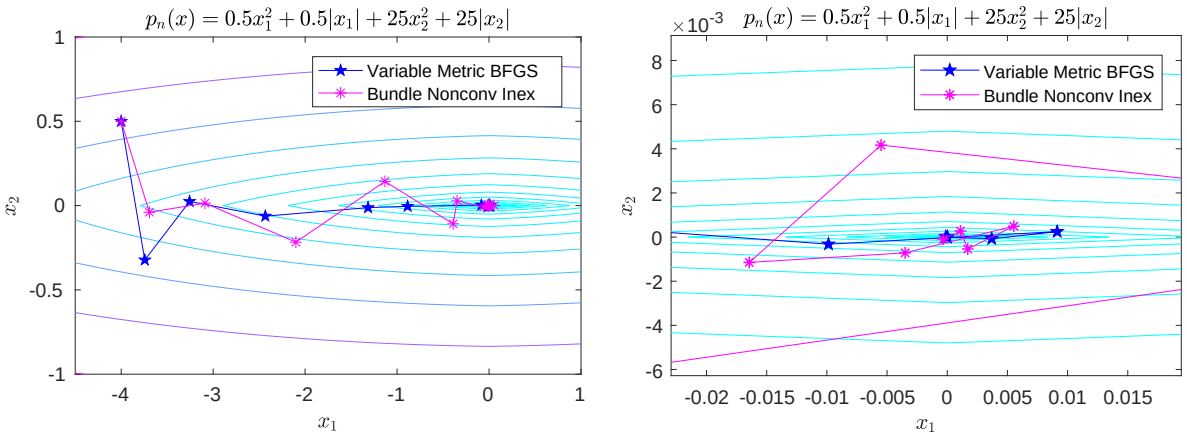

Figure 2: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of the nonsmooth quadratic function p_n. The right image is a detail of the plot on the left.*

The bar plots in figures 3 and 4 compare the accuracy of the solution and the number of steps that is needed by the different algorithms for the various noise forms. Here the nonconvex proximal bundle algorithm is compared to the

Noise form N_v^g is the one that is most similar to no noise; one can see: gradient noise has not as much impact as inexact function value (makes sense because only calculating with subgradients) -> in those cases variable metric algorithm more exact, but all over desired tolerance in terms of exactness proximal bundle algorithm slightly better, but not much; the two variants of the variable metric algorithm are more or less the same.
for step sizes> proximal bundle algorithm needs a lot more steps than variable metric especially in the case of constant function value and gradient noise

Figure 4: in nonsmooth case accuracy of proximal bundle algorithm overall slightly better; need generally more steps, numbers of steps more equal apart from constant noise part where for a minor gain of accuracy a really high numer of steps is done (plot is cropped, really: over 250 steps)
bigger difference if changing κ_+ -> add plot for that and comment???

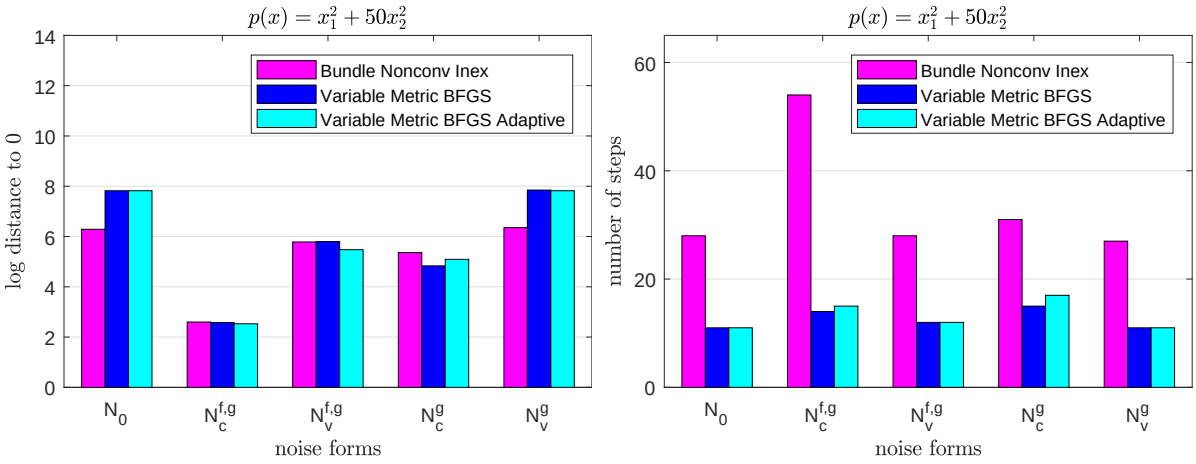

Figure 3: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the parabola p under different form of noise.*
Right: Comparison of the number of steps for the three algorithms.

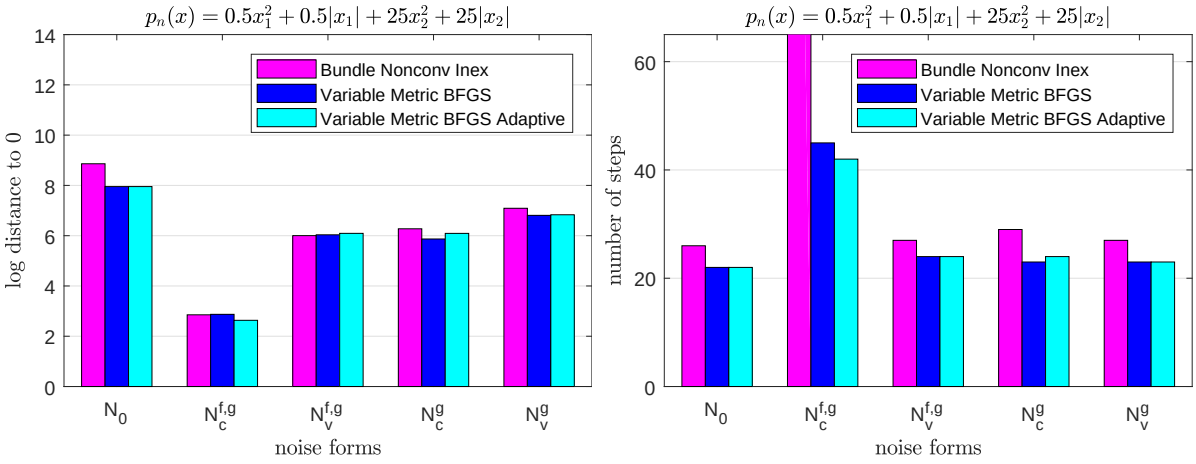

Figure 4: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the nonsmooth quadratic function p_n under different form of noise.*
Right: Comparison of the number of steps for the three algorithms. The bar for the number of steps in the case of constant noise is cropped.

second example: nonsmooth version of the parabola

$$p : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \left\langle x, \frac{1}{2}Ax \right\rangle + 0.5|x_1| + 25|x_2|$$

kink makes problem with second order information; different possibilities

1. change nothing, just make sure Q bounded
2. scale Q if it becomes too big
   advantage: not so much wrong information due to "krummungs-Paradoxon"
   drawback: everything scaled -> direction wrong if kink only in one direction; have to decide when Q too big, because exiist functions with high curvature
3. change only eigenvalue that got too high; check Change relative to last Q and stepsize
4. bigger stepsizes?

first and third variants make sense in example

recht gute Ergebnisse mit $\frac{1}{k}Q$
ohne $\frac{1}{k}$ eta extrem gross; mit: eta wird auch gross: 1e10

In this section the above variant of an inexact bundle algorithm is compared with the nonconvex inexact bundle algorithm by Hare et al. in [3].
For reasons of comparability the setting was chosen as in [3].
The parameter set was chosen as follows: $m = 0.05$, $\gamma = 2$, $t_1 = 0.1$ and the scaling parameters for the choice of t_k are $\kappa_+ = 1.2$ and $\kappa_- = 0.8$ with the minimum possible t being $t_{min} = 0.03$. For the choice of the new bundle the information of the current prox-center and all elements with $\alpha > 10^{-15}$ were kept in the bundle. As stopping condition

$$\delta < \texttt{tol}(1 + f_{hat})$$

was used. If this did not apply within $\max(300, 205n)$ steps the algorithm also stopped.

1.5.2 Test Examples in Higher Dimensions

For the second test, which involves testing the performance of the different algorithms in different dimensions of the minimizer, the Ferrier polynomials are chosen as objective functions. These nonsmooth and nonconvex functions have already been used in [2] and [3]. The polynomials are constructed in the following way:

For $i = 1, ..., n$ we define

$$h_i : \mathbb{R}^n \to \mathbb{R}, \quad h(x) = (ix_i^2 - 2x_i) + \sum_{j=1}^{n} x_j.$$

These functions are used to define

$$
\begin{aligned}
f_1(x) &:= \sum_{i=1}^{n} |h_i(x)|, \\
f_2(x) &:= \sum_{i=1}^{n} (h_i(x))^2, \\
f_3(x) &:= \max i \in [1, ..., n] |h_i(x)|, \\
f_4(x) &:= \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}|x|^2, \\
f_5(x) &:= \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}|x|.
\end{aligned}
$$

plots of graphs???

The graphs of the Ferrier polynomials for $x \in \mathbb{R}^2$ are shown in figure 5.

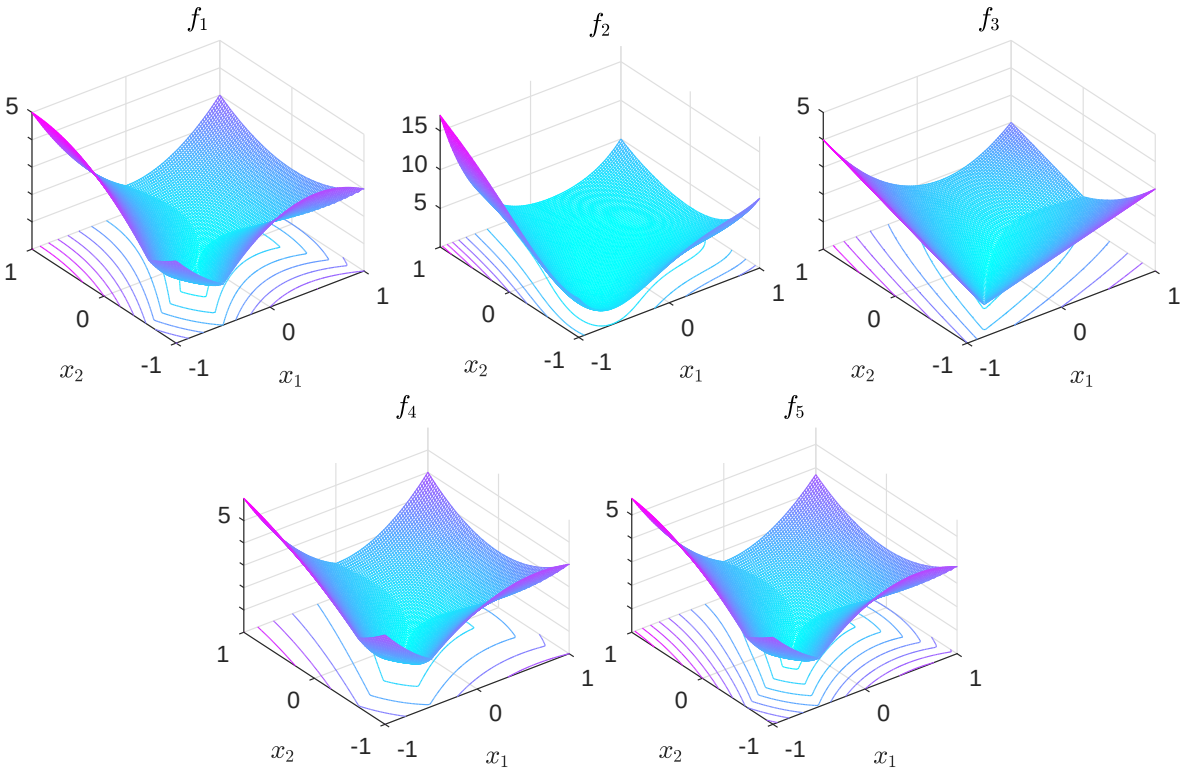

Figure 5: *Graphs of the testfunctions f_1 to f_5 for $x \in \mathbb{R}^2$*

Ferrier polynomials are nonconvex, except for f_2, nonsmooth and lower-$\mathcal{C}^2$. They all have 0 as a global minimizer [2]. The compact constraint set is $X = \{d \in \mathbb{R}^n | d_i + \hat{x}_i^k \leq 10, i =$

$1, ..., n\}$.

The five test functions $f_1, ...f_5$ were optimized for the dimensions $n = 1, 2, ..., 15, 20, 25, 30$. The starting value for each test problem being $x^1 = [1, \frac{1}{4}, \frac{1}{9}, ..., \frac{1}{n^2}]^\top$.

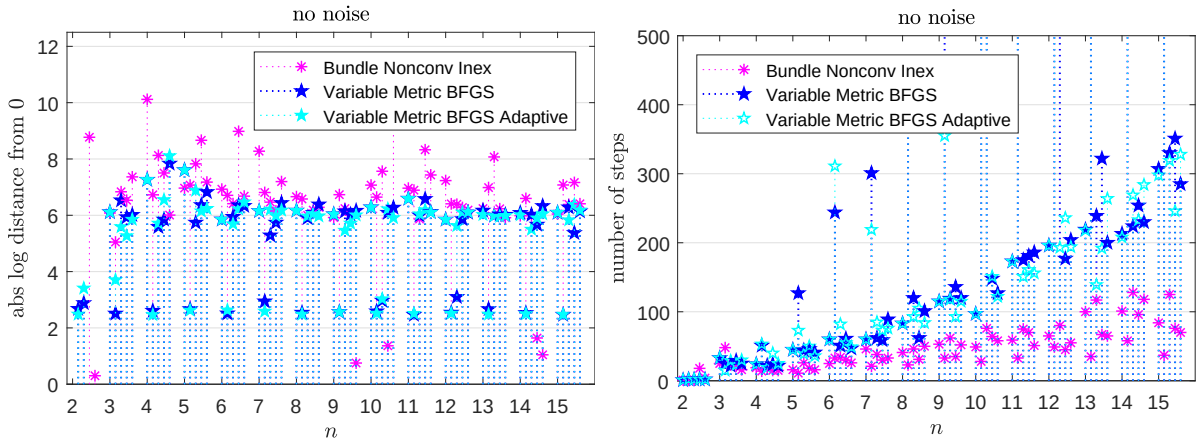

Figure 6: *No noise*

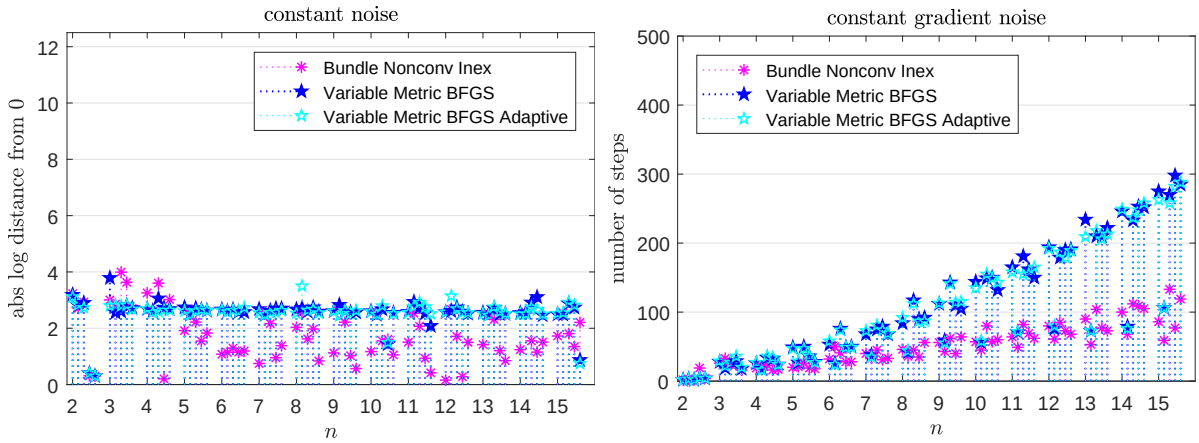

Figure 7: *Constant noise*

say where plots left and right...

- Hare Algo seems to be better in "global" optimization
  Noll seems to get stuck in local Minima more often
- different "Versions" of Noll: Q "ignored" near kinks because there no curvature, but seems be be "infinite"
- the t-update Parameter has A LOT of influence on the performance of the algorithm

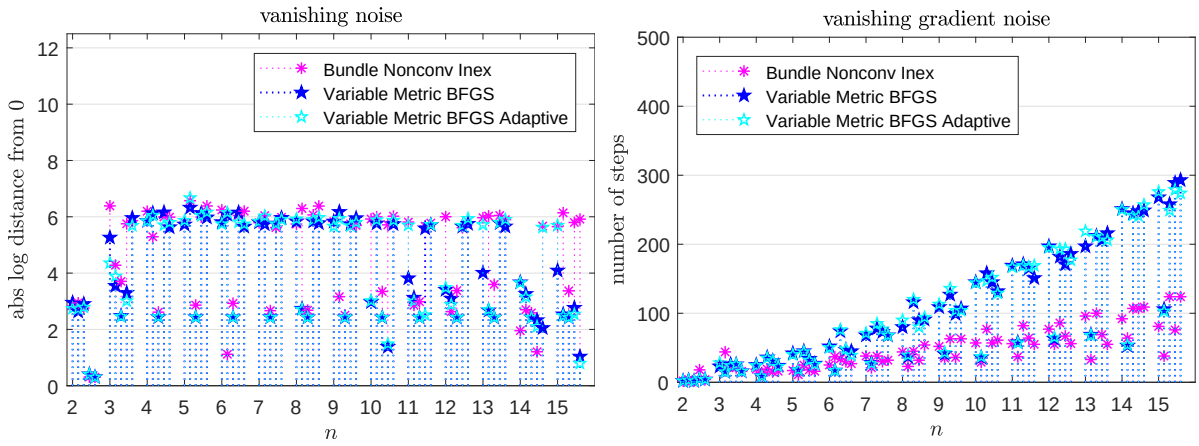

Figure 8: *Vanishing noise*

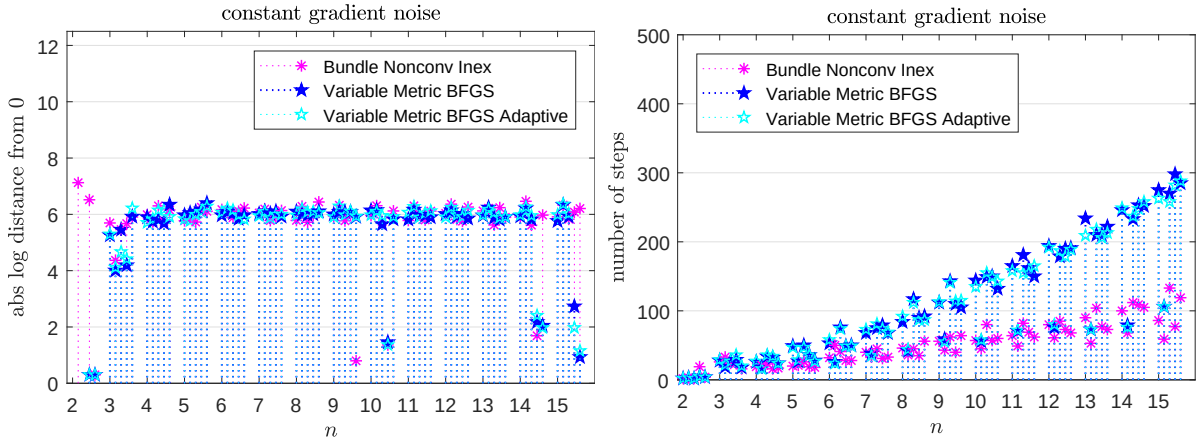

Figure 9: *Constant gradient noise*

Rechtschreibfehler, Namen, Stil überprüfen

References

[1] Napsu Haarala, Kaisa Miettinen, and Marko M. Mäkelä. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Mathematical Programming*, 109(1):181–205, 2007.

[2] Warren Hare and Claudia Sagastizàbal. A redistributed proximal bundle method for nonconvex optimization. *SIAM Journal on Optimization*, 20(5):2442–2473, 2010.

[3] Warren Hare, Claudia Sagastizàbal, and Mikhail Solodov. A proximal bundle method for nonsmooth nonconvex functions with inexact information. *Computational Optimization and Applications*, 63:1–28, 2016.

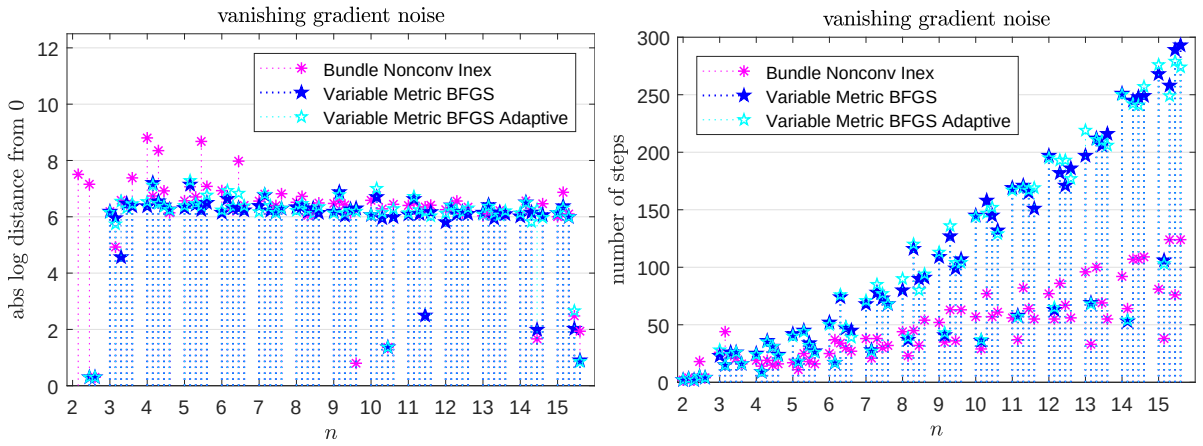

Figure 10: *Vanishing gradient noise*

[4] Claude Lemaréchal and Claudia Sagastizábal. *An approach to variable metric bundle methods*, pages 144–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[5] Claude Lemaréchal and Claudia Sagastizábal. Variable metric bundle methods: From conceptual to implementable forms. *Mathematical Programming*, 76(3):393–410, 1997.

[6] Adrian S Lewis and Michael L Overton. Nonsmooth optimization via bfgs. *submitted to SIAM Journal on Optimization*, pages 1–35, 2009.

[7] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *Journal of Optimization Theory and Applications*, 102(3):593–613, sep 1999.

[8] Stefan Ulbrich Michael Ulbrich. *Nichtlineare Optimierung*. Springer Basel AG, 2012.

[9] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[10] Dominikus Noll. Bundle method for non-convex minimization with inexact subgradients and function values. In *Computational and Analytical Mathematics*, pages 555–592. Springer Nature, 2013.

[11] Dominikus Noll, Olivier Prot, and Aude Rondepierre. A proximity control algorithm to minimize non-smooth and non-convex functions. *Pacific Journal of Optimization*, 4(3):571–604, 2012.

[12] Boris T. Polyak. *Introduction to Optimization*. Optimization Software , Inc., Publications Division, New York, 1987.

[13] J. Vlček and L. Lukšan. Globally convergent variable metric bundle method for nonconvex nondifferentiable unconstrained minimization. *Journal of Optimization Theory and Applications*, 111(2):407–430, 2001.

[14] Jochen Werner. *Numerische Mathematik 2*. Vieweg, Wiesbaden, 1992.