Journal of Computational Mathematics
Vol.xx, No.x, 2023, 1–32.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2306-m2022-0279

# ADAPTIVE REGULARIZED QUASI-NEWTON METHOD USING INEXACT FIRST-ORDER INFORMATION*

Hongzheng Ruan[1)]
*School of Mathematical Sciences, Fudan University, Shanghai 200433, P.R. China*
*Email: hzruan19@fudan.edu.cn*
Wei Hong Yang
*School of Mathematical Sciences, Fudan University, Shanghai 200433, P.R. China*
*Email: whyang@fudan.edu.cn*

**Abstract**

Classical quasi-Newton methods are widely used to solve nonlinear problems in which the first-order information is exact. In some practical problems, we can only obtain approximate values of the objective function and its gradient. It is necessary to design optimization algorithms that can utilize inexact first-order information. In this paper, we propose an adaptive regularized quasi-Newton method to solve such problems. Under some mild conditions, we prove the global convergence and establish the convergence rate of the adaptive regularized quasi-Newton method. Detailed implementations of our method, including the subspace technique to reduce the amount of computation, are presented. Encouraging numerical results demonstrate that the adaptive regularized quasi-Newton method is a promising method, which can utilize the inexact first-order information effectively.

*Mathematics subject classification:* 90C30, 68Q25.
*Key words:* Inexact first-order information, Regularization, Quasi-Newton method.

## 1. Introduction

In this paper, we consider the following problem:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where $f$ is a continuously differentiable function. In many applications, the first-order information can not be known exactly. There are some errors in computations of $f(x)$ and $\nabla f(x)$, which are due to stochastic noise [10], internal discretization [29,30], and gradient approximations based on finite differencing, interpolating, or smoothing [4,5,23,24,31,33]. Problems of this type arise in a variety of fields, such as multidimensional numerical integration optimization [12], derivative-free optimization (DFO) [15,28], and machine learning [16].

Devolder *et al.* [21] introduce the notion of first-order inexact oracle, and study the properties of several smooth and non-smooth convex optimization algorithms relying on such oracles. Inexact oracles have been widely studied for smoothing convex optimization problems. Readers are referred to [17,20,22,32].

Inexact oracles can also be applied to DFO problems. In [7], a derivative-free method based on inexact oracles is proposed. Recently there has been much interest in the proximal method for composite optimization with inexact oracles. Readers are referred to [18,35,36].

* Received December 12, 2022 / Revised version received March 27, 2023 / Accepted June 28, 2023 / Published online November 26, 2023 /
1) Corresponding author

Classical quasi-Newton algorithms have been extended to problems with inexact first-order information. An established BFGS algorithm with inexact gradient, named implicit filtering method, is proposed in [26]. It is designed for the case when errors are diminishing. Quasi-Newton method with non-diminishing and bounded errors (BFGSe) is analyzed in [40]. It describes a noise tolerant modification of the BFGS method. Berahas *et al.* [3] propose implementations of the BFGS method and L-BFGS method in which gradients are computed by an appropriate finite differencing technique. The study of BFGS methods with inaccurate gradients has attracted much interest in recent years. For details readers are referred to [6].

An interesting approach to problem (1.1) is the trust region method. Analysis for trust region methods with inexact gradients is presented in [11], which establishes strong global convergence results under the relative error condition. Convergence results for truncated trust region methods are established in [27]. Trust region methods can also be applied to problems where only stochastic gradient information can be used. Chen *et al.* [14] establish almost sure global convergence under the assumption that the first-order information is sufficiently accurate with high enough probability. In [2, 8], the authors analyze trust region methods with adaptive accuracy in function and gradient evaluation.

Inspired by the idea in [13, 25], we propose an adaptive regularized quasi-Newton method, named adaQN, to solve problem (1.1). Specifically, we approximate problem (1.1) and construct a subproblem by adding a regularization term to the quasi-Newton quadratic model. One major difference between adaQN and BFGSe is that adaQN does not use the line search and utilizes a trust-region-like framework to monitor the acceptance of trial steps. In many cases, this strategy can save some computational cost. Numerical experiments demonstrate the effectiveness of the adaQN method. For large scale problems, we incorporate subspace techniques [37, 41] into our adaQN method. Such techniques can reduce the amount of computation significantly in early iterations, especially when the dimension of the problem is very large. We also study the global convergence of the adaQN method. Under mild conditions, local convergence rates of adaQN are established.

This paper is organized into six sections. In Section 2, we describe the problem and propose the adaptive regularized quasi-Newton method. The main convergence results for different kinds of errors are presented in Section 3. In Section 4, we describe practical implementations of the proposed method. Numerical experiments, summarized in Section 5, indicate that the proposed method is more effective and robust. Some conclusions are drawn in Section 6.

## 2. Problem and Algorithm

Our goal in this section is to design an effective quasi-Newton method, which uses inexact first-order information to solve problem (1.1). We first introduce some notations that will be used throughout the paper for our descriptions. We use $g(x)$ to denote $\nabla f(x)$ in the paper. Given $x \in \mathbb{R}^n, \delta \geq 0$ and $\eta \geq 0$, we use $f_\delta(x)$ and $g_\eta(x)$ to denote the approximate values of $f(x)$ and $g(x)$ with errors controlled by $\delta$ and $\eta$, that is

$$|f(x) - f_\delta(x)| \leq \delta, \tag{2.1}$$

$$\|g(x) - g_\eta(x)\| \leq \eta. \tag{2.2}$$

Given a sequence $\{x_k\}$, $g(x_k)$ is denoted as $g_k$ for ease of notation. Let $\{\eta_k\}$ be a non-negative sequence. We use the notation $\tilde{g}_k := g_{\eta_k}(x_k)$, where $g_{\eta_k}(x_k)$ satisfies (2.2).

In the classical quasi-Newton method, the approximate Hessian matrix $B_k$ is updated by using $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$, and the search direction is computed by $d_k = -B_k^{-1} g_k$. Since we can only obtain approximate values of $g_k$ and $g_{k-1}$, we need an inexact version of the quasi-Newton method. We update $B_k$ by the BFGS update formula

$$B_k = B_{k-1} + \frac{\tilde{y}_{k-1}\tilde{y}_{k-1}^\top}{\tilde{y}_{k-1}^\top s_{k-1}} - \frac{B_{k-1}s_{k-1}s_{k-1}^\top B_{k-1}}{s_{k-1}^\top B_{k-1}s_{k-1}}, \tag{2.3}$$

where

$$s_{k-1} = x_k - x_{k-1}, \quad \tilde{y}_{k-1} = \tilde{g}_k - \tilde{g}_{k-1}.$$

Let $d_k = -B_k^{-1}\tilde{g}_k$ be the search direction. If $\eta_k > 0$, then $d_k$ may not be a descent direction in general.

To generate a descent direction at iterate $x_k$, we add a regularization term to the second-order Taylor model of $f$ to obtain the objective function of our subproblem

$$m_k(d) := f_{\delta_k}(x_k) + \tilde{g}_k^\top d + \frac{1}{2}d^\top B_k d + \frac{1}{2}\sigma_k \|d\|^2, \tag{2.4}$$

where $\{\delta_k\}_{k\geq 0}$ is a non-negative sequence, and $\sigma_k > 0$ is the regularization parameter. Let $d_k$ be the solution of (2.4). Then $d_k = -(B_k + \sigma_k I)^{-1}\tilde{g}_k$. In our algorithm, we adjust $\sigma_k$ adaptively to ensure that $d_k$ is a descent direction of $f$ at $x_k$. Such a strategy is used in adaptive regularized methods (see [25,38,39]). Numerical experiments in [39] demonstrate that the adaptive regularization strategy can improve the performance of the algorithm.

Now we give a brief description of our method. Our algorithm starts from a feasible initial point $x_0$, a positive definite matrix $B_0$, and an initial regularization parameter $\sigma_0$. At iteration $k$, our method computes $d_k$, the solution of (2.4). In our algorithm, we adjust $\sigma_k$ to guarantee the angle of $d_k$ and $-\tilde{g}_k$ is bounded by an acute angle. Thus, there exists $\tau > 0$ such that

$$-\tilde{g}_k^\top d_k \geq \tau \|\tilde{g}_k\| \, \|d_k\|.$$

Using $(B_k + \sigma_k I)d_k = -\tilde{g}_k$, we can deduce that

$$\begin{aligned} m_k(0) - m_k(d_k) &= f_{\delta_k}(x_k) - \left[ f_{\delta_k}(x_k) + \tilde{g}_k^\top d_k + \frac{1}{2}d_k^\top B_k d_k + \frac{1}{2}\sigma_k \|d_k\|^2 \right] \\ &= -\frac{1}{2}\tilde{g}_k^\top d_k \geq \frac{\tau}{2}\|\tilde{g}_k\| \, \|d_k\|. \end{aligned} \tag{2.5}$$

In order to decide whether $x_k + d_k$ should be accepted as the next iterate and whether the regularization parameter $\sigma_k$ should be updated, we calculate the following ratio:

$$\rho_k = \frac{f_{\delta_k}(x_k) - f_{\delta_k}(x_k + d_k) + t\zeta_k}{m_k(0) - m_k(d_k) + t\zeta_k}, \tag{2.6}$$

where $t > 0$ is a constant, $\{\zeta_k\}$ is a positive sequence. Let $c_1$ and $c_2$ be two parameters, which satisfy $0 < c_1 < c_2 < 1$. If $\rho_k \geq c_2$, we say that the iteration is very successful, and we set $x_{k+1} = x_k + d_k$, if $\rho_k \in [c_1, c_2)$, we say that the iteration is successful, $x_k$ is updated as well, otherwise, the iteration is not successful, and $x_k$ remains the same. Based on the value of $\rho_k$, our algorithm makes corresponding adjustment for $\sigma_k$.

Now we provide a description of our method in Algorithm 2.1.

**Algorithm 2.1:** Adaptive Regularized Quasi-Newton Method Using Inexact First-Order Information (adaQN).

**Require:** Starting point $x_0$, $\sigma_0 > 0$, $B_0 \succ 0$, $t > 0$, $\tau \in (0,1)$, $0 < c_1 < c_2 < 1$ and $0 < a_0 < a_1 < 1 < a_2 < a_3$, error sequences $\{\delta_k\}$ and $\{\eta_k\}$, and a positive sequence $\{\zeta_k\}$.

1 **for** $k = 0, 1, \dots$ until convergence **do**

2 Compute the search direction $d_k = -(B_k + \sigma_k I)^{-1} \tilde{g}_k$.

3 **if** $-\tilde{g}_k^\top d_k < \tau \|\tilde{g}_k\| \cdot \|d_k\|$ **then**

4 Update $\sigma_k = a_3 \sigma_k$, and go back to step 2.

5 **end**

6 Evaluate the ratio $\rho_k$ as in (2.6).

7 Set

$$x_{k+1} = \begin{cases} x_k + d_k, & \text{if} \quad \rho_k \geq c_1, \\ x_k, & \text{otherwise}. \end{cases} \tag{2.7}$$

8 Set

$$\sigma_{k+1} = \begin{cases} [a_0 \sigma_k, a_1 \sigma_k], & \text{if} \quad \rho_k \geq c_2, \\ \sigma_k, & \text{if} \quad c_1 \leq \rho_k < c_2, \\ [a_2 \sigma_k, a_3 \sigma_k], & \text{otherwise}. \end{cases}$$

9 **if** $\rho_k \geq c_1$ **then** update $B_{k+1}$ by (2.3) **else** $B_{k+1} = B_k$.

10 **end**

## 3. Convergence Analysis

In this section, we study the convergence properties of Algorithm 2.1. When the sequences $\{\delta_k\}$ and $\{\eta_k\}$ are diminishing, we prove the global convergence of our method under some mild conditions in Section 3.1, the local convergence rate of Algorithm 2.1 is analyzed in Section 3.2. When the sequences $\{\delta_k\}$ and $\{\eta_k\}$ are non-diminishing and bounded, we present the convergence results of our method in Section 3.3.

We present some assumptions which will be used throughout the paper.

**Assumption 3.1.** *$f(x)$ is bounded below and continuously differentiable. The optimal solution of* (1.1) *exists and is denoted by $x^*$.*

**Assumption 3.2.** *$g(x)$ is globally Lipschitz continuous with Lipschitz constant $L$, that is for any $x, y \in \mathbb{R}^n$,*

$$\|g(x) - g(y)\| \leq L \|x - y\|.$$

**Assumption 3.3.** *$B_k$ is a symmetric matrix and $B_k + \sigma_k I$ is nonsingular for all $k \geq 0$. There exists $\overline{\kappa} > 0$ such that $\|B_k\| \leq \overline{\kappa}$ for all $k \geq 0$. Here $\|\cdot\|$ denotes the 2-norm of a matrix.*

Let $\tau$ be the parameter in Algorithm 2.1. We will use the notation $\tau_0$ in the following results, which is defined as

$$\tau_0 := \sqrt{\frac{\tau^2}{1 - \tau^2}}. \tag{3.1}$$

**Lemma 3.1.** *Let $x, y$ be two vectors, and $\alpha$ be a scalar. Suppose $\|x\| = 1$. If*

$$\alpha \geq (1 + \tau_0)\|y\|, \tag{3.2}$$

*then $(\alpha x + y)^\top x \geq \tau\|\alpha x + y\|$, where $\tau$ and $\tau_0$ satisfy* (3.1).

*Proof.* Write $y$ as

$$y = \beta_1 x + \beta_2 z,$$

where $\beta_1, \beta_2 \in \mathbb{R}, z$ is a unit vector in $x^\perp = \{w : w^\top x = 0\}$. Since $x$ and $z$ are unit vectors, taking into account that $z \in x^\perp$, we have

$$\|y\| \geq \max\{|\beta_1|, |\beta_2|\}, \tag{3.3}$$

$$(\alpha x + y)^\top x = \alpha + \beta_1, \tag{3.4}$$

$$\|\alpha x + y\|^2 = (\alpha + \beta_1)^2 + \beta_2^2. \tag{3.5}$$

By (3.2) and (3.3), we can deduce that $\alpha + \beta_1 \geq \tau_0|\beta_2| \geq 0$, which together with (3.1) implies

$$(\alpha + \beta_1)^2 - \tau^2\big((\alpha + \beta_1)^2 + \beta_2^2\big) \geq 0.$$

Combining the above inequality with (3.4) and (3.5) yields the assertion. □

The following lemma guarantees that in steps 3-5 of Algorithm 2.1, we can find a $\sigma_k$ such that the search direction $d_k$ satisfies

$$-d_k^\top \tilde{g}_k \geq \tau\|\tilde{g}_k\| \, \|d_k\|. \tag{3.6}$$

**Lemma 3.2.** *Suppose Assumption* 3.3 *holds. If*

$$\sigma_k \geq (1 + \tau_0)\overline{\kappa}, \tag{3.7}$$

*then $d_k = -(B_k + \sigma_k I)^{-1}\tilde{g}_k$ satisfies* (3.6).

*Proof.* If $d_k = 0$, then (3.6) holds. We only need to consider the case $d_k \neq 0$. If (3.7) holds, by Assumption 3.3, we have

$$\sigma_k\|d_k\| \geq (1 + \tau_0)\|B_k\| \, \|d_k\| \geq (1 + \tau_0)\|B_k d_k\|.$$

Letting $\alpha = \sigma_k\|d_k\|$, $x = d_k/\|d_k\|$ and $y = B_k d_k$, by Lemma 3.1, we can deduce that

$$[(B_k + \sigma_k I)d_k]^T d_k \geq \tau\|(B_k + \sigma_k I)d_k\| \, \|d_k\|,$$

which together with $\tilde{g}_k = -(B_k + \sigma_k I)d_k$ implies the assertion. □

From (2.5), it follows that $f_{\delta_k}(x_k) = m_k(0) \geq m_k(d_k)$. Since $\{\zeta_k\}$ is a positive sequence and $t > 0$, the denominator of $\rho_k$ in (2.5) is greater than zero. Let

$$r_k := [f_{\delta_k}(x_k) - f_{\delta_k}(x_k + d_k) + t\zeta_k] - c_2[f_{\delta_k}(x_k) - m_k(d_k) + t\zeta_k], \tag{3.8}$$

where $c_2$ is the parameter in Algorithm 2.1. It is obvious that $r_k \geq 0$ if and only if $\rho_k \geq c_2$. In the rest of the paper, we assume that the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$ satisfy

$$t(1 - c_2) \geq 2, \tag{3.9}$$

$$\zeta_k \geq \delta_k, \quad \forall k. \tag{3.10}$$

In the following results, we use the notation

$$\bar{c} := \frac{1}{4}\left(1 - \frac{c_2}{2}\right). \tag{3.11}$$

**Lemma 3.3.** *Suppose Assumptions* 3.1-3.3 *hold, and for the parameter* $t$*, sequences* $\{\delta_k\}$ *and* $\{\zeta_k\}$*,* (3.9) *and* (3.10) *hold. Further suppose*

$$\|g_k\| \geq (1/\bar{c} + 1)\eta_k. \tag{3.12}$$

*If*

$$\sigma_k \geq \bar{\sigma} := \max\{(1+\tau_0)\overline{\kappa}, \overline{\kappa} + L/(2\bar{c}), 3\overline{\kappa}\}, \tag{3.13}$$

*then the* $k$*-th iteration is very successful.*

*Proof.* Let $\lambda_{\min}(B_k)$ and $\lambda_{\max}(B_k)$ be the minimum and maximum eigenvalues of $B_k$. It follows from Assumption 3.3 that

$$\overline{\kappa} \geq \|B_k\| = \max\big(|\lambda_{\min}(B_k)|, |\lambda_{\max}(B_k)|\big).$$

By (3.13), $B_k + \sigma_k I$ is positive definite. Using the above inequality, we can deduce that

$$\begin{aligned}\lambda_{\min}(B_k + \sigma_k I) = \sigma_k + \lambda_{\min}(B_k) &\geq \frac{1}{2}(\sigma_k + \sigma_k - 2\overline{\kappa}) \\ &\geq \frac{1}{2}(\sigma_k + \overline{\kappa}) \geq \frac{1}{2}\lambda_{\min}(B_k + \sigma_k I).\end{aligned}$$

By the above inequality, we have

$$\begin{aligned}\|\tilde{g}_k\| = \|-(B_k + \sigma_k I)d_k\| &\leq \lambda_{\max}(B_k + \sigma_k I)\|d_k\| \\ &\leq 2\lambda_{\min}(B_k + \sigma_k I)\|d_k\|.\end{aligned}$$

Using the Taylor expansion of $f(x_k + d_k)$ around $x_k$, we can deduce that

$$\begin{aligned}&m_k(d_k) - f_{\delta_k}(x_k + d_k) \\ &= \Big[f(x_k) + \tilde{g}_k^\top d_k + \frac{1}{2}d_k^\top B_k d_k + \frac{1}{2}\sigma_k\|d_k\|^2 - f(x_k + d_k)\Big] \\ &\quad + \big(f_{\delta_k}(x_k) - f(x_k)\big) + \big(f(x_k + d_k) - f_{\delta_k}(x_k + d_k)\big) \\ &\geq -2\delta_k + \Big[f(x_k) + \tilde{g}_k^\top d_k + \frac{1}{2}d_k^\top B_k d_k + \frac{1}{2}\sigma_k\|d_k\|^2 - f(x_k + d_k)\Big] \\ &= -2\delta_k + \big(\tilde{g}_k - g(\omega_k)\big)^\top d_k + \frac{1}{2}d_k^\top B_k d_k + \frac{1}{2}\sigma_k\|d_k\|^2,\end{aligned} \tag{3.14}$$

where $\omega_k$ lies in the line segment $(x_k, x_k + d_k)$, and the first inequality is due to the definition of $\delta_k$. From (3.9) and (3.10), it follows that

$$(1 - c_2)t\zeta_k \geq 2\zeta_k \geq 2\delta_k.$$

Using the above inequality with (3.14), we have

$$\begin{aligned}r_k &= [f_{\delta_k}(x_k) - f_{\delta_k}(x_k + d_k) + t\zeta_k] - c_2[f_{\delta_k}(x_k) - m_k(d_k) + t\zeta_k] \\ &= [m_k(d_k) - f_{\delta_k}(x_k + d_k)] + (1 - c_2)[f_{\delta_k}(x_k) - m_k(d_k)] + (1 - c_2)t\zeta_k \\ &\geq \big(\tilde{g}_k - g(\omega_k)\big)^\top d_k - (1 - c_2)\tilde{g}_k^\top d_k + \frac{c_2}{2}\big(d_k^\top B_k d_k + \sigma_k\|d_k\|^2\big) + (1 - c_2)t\zeta_k - 2\delta_k \\ &\geq (\tilde{g}_k - g_k)^\top d_k + \big(g_k - g(\omega_k)\big)^\top d_k + \Big(1 - \frac{c_2}{2}\Big) d_k^\top (B_k + \sigma_k I)d_k \\ &\geq \|d_k\|\big(-\eta_k - \|g_k - g(\omega_k)\| + 4\bar{c}\lambda_{\min}(B_k + \sigma_k I)\|d_k\|\big),\end{aligned}$$

where we use $\tilde{g}_k = -(B_k + \sigma_k I)d_k$ in the second inequality, and use $\|\tilde{g}_k - g_k\| \le \eta_k$ in the last inequality. From (3.12), it follows that

$$\|\tilde{g}_k\| \ge \|g_k\| - \|\tilde{g}_k - g_k\| \ge \frac{\eta_k}{\bar{c}}.$$

Combining the above inequality with (3.14), we have

$$\begin{aligned} 2\bar{c}\lambda_{\min}(B_k + \sigma_k I)\|d_k\| - \eta_k &\ge \bar{c}\lambda_{\max}(B_k + \sigma_k I)\|d_k\| - \eta_k \\ &\ge \bar{c}\|(B_k + \sigma_k I)d_k\| - \eta_k \\ &= \bar{c}\|\tilde{g}_k\| - \eta_k \ge 0. \end{aligned} \tag{3.15}$$

Using Assumption 3.2 and (3.13), we can deduce that

$$\begin{aligned} &2\bar{c}\lambda_{\min}(B_k + \sigma_k I)\|d_k\| - \|g_k - g(\omega_k)\| \\ \ge\ &2\bar{c}\big(\sigma_k - \overline{\kappa} + \lambda_{\min}(B_k + \overline{\kappa}I)\big)\|d_k\| - L\|\omega_k - x_k\| \\ \ge\ &2\bar{c}(\sigma_k - \overline{\kappa})\|d_k\| - L\|d_k\| \ge 0. \end{aligned} \tag{3.16}$$

Combining (3.15) and (3.16) yields $r_k \ge 0$, which is equivalent to $\rho_k \ge c_2$. Thus the assertion holds. $\square$

### 3.1. Global convergence when $\{\delta_k\}$ and $\{\eta_k\}$ are diminishing

In this subsection, we make the following assumption.

**Assumption 3.4.** *$\{\eta_k\}, \{\delta_k\}$ are two non-negative sequences, and $\{\zeta_k\}$ is a positive sequence. Moreover,*

$$\eta_k \ \to\ 0 \quad as \quad k \ \to\ \infty, \tag{3.17}$$

*and*

$$\sum_{k=1}^{\infty} \zeta_k < \infty. \tag{3.18}$$

By Assumption 3.4, we know that $\zeta_k \to 0$ as $k \to \infty$. From (3.18) and (3.10), it follows that

$$\sum_{k=1}^{\infty} \delta_k < +\infty.$$

We record the indices of successful and very successful iterations in the set

$$\mathbb{S} := \{k \ge 0 : \text{the } k\text{-th iteration is successful or very successful}\}. \tag{3.19}$$

By the procedures of Algorithm 2.1, we can see that

$$x_{k+1} = x_k, \quad \forall k \notin \mathbb{S}. \tag{3.20}$$

As usual, $|\mathbb{S}|$ is used to denote the cardinality of $\mathbb{S}$. The following result tells us that if $|\mathbb{S}|$ is finite, then $g_k = 0$ for all sufficiently large $k$.

**Theorem 3.1.** *Suppose Assumptions* 3.1-3.4 *hold,* (3.9) *and* (3.10) *hold for $t, \{\delta_k\}$ and $\{\zeta_k\}$. Further assume $|\mathbb{S}| < \infty$. Then there exists $k'$ such that $x_j = x_{k'+1}$ for all $j > k'$, and $g_{k'+1} = 0$.*

*Proof.* Let $x_{k'}$ be the last successful or very successful iterate, where $k' > 0$ is an integer. By (3.20), we have $x_j = x_{k'+1}$ for all $j > k'$. Now we prove $g_{k'+1} = 0$. We show it by contradiction. Assume that $\epsilon := \|g_{k'+1}\| > 0$. Let $\bar{c}$ be defined by (3.11). From Assumption 3.4, we know that $\eta_k \to 0$. Thus, there exists $k_1$ such that $\epsilon > (1/\bar{c}+1)\eta_k$ for all $k > k_1$. Note that the $k$-th iteration is unsuccessful for all $k > k'$. By the step 8 of Algorithm 2.1, $\sigma_{k+1} \geq a_2\sigma_k$ for all $k > k'$. Thus, there exists $k_2 > \max\{k', k_1\}$ such that $\sigma_{k_2} > \bar{\sigma}$, where $\bar{\sigma}$ is defined by (3.13). From $k_2 > k'$, it follows that $\|g_{k_2}\| > \epsilon$ and therefore $\|g_{k_2}\| > (1/\bar{c}+1)\eta_k$. Using this and $\sigma_{k_2} > \bar{\sigma}$, by Lemma 3.3, we obtain that the $k_2$-th iteration is very successful, giving a contradiction. Thus $g_{k'+1} = 0$. $\square$

By Theorem 3.1, we can assume that $|\mathbb{S}| = \infty$ in the rest of the paper.

Now we present our main results of this subsection. We separate our proof into two parts. First, we prove a weak convergence result $\liminf_{k\to\infty} \|g_k\| = 0$ in Theorem 3.2. Next, with the help of the result of Theorem 3.2, we establish the stronger convergence result in Theorem 3.3.

**Theorem 3.2.** *Suppose Assumptions* 3.1-3.4 *hold. Further assume that* (3.9) *and* (3.10) *hold for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. Then we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.21}$$

*Proof.* The proof is by contradiction. Assume that (3.21) does not hold. Then there exist $\epsilon > 0$ and $k_1$ such that $\|g_k\| \geq 2\epsilon$ for all $k > k_1$. Since $\{\eta_k\}$ is a diminishing sequence, without loss of generality, assume that $\eta_k < \epsilon$ for all $k > k_1$, which implies that

$$\|\tilde{g}_k\| \geq \|g_k\| - \|g_k - \tilde{g}_k\| \geq \epsilon, \quad \forall k > k_1. \tag{3.22}$$

Let us first prove

$$\sum_{k\in\mathbb{S}} \|d_k\| < \infty, \tag{3.23}$$

where $\mathbb{S}$ is defined by (3.19). From (2.1), (2.6) and (2.7), we know that for all $k \in \mathbb{S}$,

$$\begin{aligned}
f(x_k) - f(x_{k+1}) &\geq \big(f_{\delta_k}(x_k) - f_{\delta_k}(x_{k+1}) + t\zeta_k\big) - (2\delta_k + t\zeta_k)\\
&\geq c_1\big(m_k(0) - m_k(d_k) + t\zeta_k\big) - (2\delta_k + t\zeta_k)\\
&= -c_1\left(\tilde{g}_k^\top d_k + \frac{1}{2}d_k^\top B_k d_k + \frac{1}{2}\sigma_k\|d_k\|^2\right) - \big(2\delta_k + (1-c_1)t\zeta_k\big)\\
&= -\frac{c_1}{2}\tilde{g}_k^\top d_k - \big(2\delta_k + (1-c_1)t\zeta_k\big)\\
&\geq \frac{\tau c_1}{2}\|\tilde{g}_k\|\,\|d_k\| - \big(2\delta_k + (1-c_1)t\zeta_k\big),
\end{aligned} \tag{3.24}$$

where the second equality is due to $(B_k + \sigma_k I)d_k = -\tilde{g}_k$, and the last inequality follows from steps 3-5 of Algorithm 2.1.

Pick an integer $k_2$ in $\mathbb{S}$ such that $k_2 > k_1$. From (3.20), we know that for any $j \in \mathbb{S}$ with $j \geq k_2$, we have

$$\begin{aligned}
f(x_{k_2}) - f(x_{j+1}) &= \sum_{k=k_2}^{j}[f(x_k) - f(x_{k+1})] = \sum_{k_2\leq k\leq j, k\in\mathbb{S}}[f(x_k) - f(x_{k+1})]\\
&\geq \frac{\tau c_1}{2}\sum_{k_2\leq k\leq j, k\in\mathbb{S}}\|\tilde{g}_k\|\,\|d_k\| - \sum_{k_2\leq k\leq j, k\in\mathbb{S}}[2\delta_k + (1-c_1)t\zeta_k]\\
&\geq \frac{\tau c_1\epsilon}{2}\sum_{k_2\leq k\leq j, k\in\mathbb{S}}\|d_k\| - \sum_{k_2\leq k\leq j, k\in\mathbb{S}}[2\delta_k + (1-c_1)t\zeta_k],
\end{aligned} \tag{3.25}$$

where the first inequality follows from (3.24) and the last inequality is due to (3.22). By (3.10) and (3.18), there exists $M > 0$ such that

$$\sum_{k_2 \leq k \leq j, k \in \mathbb{S}} [2\delta_k + (1 - c_1)t\zeta_k] < M, \quad \forall j \geq k_2,$$

which together with (3.25) and Assumption 3.1 implies

$$\sum_{k=k_2, k \in \mathbb{S}}^{j} \|d_k\| < \frac{2}{\tau c_1 \epsilon}\big(M + f(x_{k_2}) - f(x^*)\big). \tag{3.26}$$

Letting $j \to \infty$ in (3.26) yields $\sum_{k=k_2,k\in\mathbb{S}}^{\infty} \|d_k\| < \infty$, and therefore (3.23) holds.

It follows from (3.23) that

$$\lim_{k \in \mathbb{S}, k \to \infty} \|d_k\| = 0.$$

Note that for $k \in \mathbb{S}$ with $k > k_1$, it holds that

$$\epsilon \leq \|\tilde{g}_k\| = \| - (B_k + \sigma_k)d_k\| \leq (\overline{\kappa} + \sigma_k)\|d_k\|,$$

which implies that

$$\sigma_k \quad \to \quad \infty \quad \text{as} \quad k \quad \to \quad \infty, \quad k \in \mathbb{S}. \tag{3.27}$$

Let $\bar{\sigma}$ be defined by (3.13). Since $\eta_k \to 0$, there exists $k_3$ such that (3.12) holds for all $k \geq k_3$. For any $k' \geq k_3$, by Lemma 3.3, we can see that

$$\text{if} \quad \sigma_{k'} \geq \bar{\sigma}, \quad \text{then} \quad k' \in \mathbb{S}. \tag{3.28}$$

For an integer $k \in \mathbb{S}$ with $k > k_3$, we consider two cases $k - 1 \in \mathbb{S}$ or not. If $k - 1 \notin \mathbb{S}$, by (3.28), we know that $\sigma_{k-1} < \bar{\sigma}$. From the procedures of Algorithm 2.1, it holds that $\sigma_k \leq a_3\sigma_{k-1} < a_3\bar{\sigma}$. If $k - 1 \in \mathbb{S}$, then $\sigma_k \leq \sigma_{k-1}$. Combining the two cases, we can deduce that $\sigma_k \leq \max\{\sigma_{k_3}, a_3\bar{\sigma}\}$ for all $k \in \mathbb{S}$ with $k > k_3$, a contradiction to (3.27). □

In Theorem 3.3, we show that $\lim_{k\to\infty} \|g_k\|$=0, which is equivalent to $\lim_{k\to\infty} \|\tilde{g}_k\|$=0. The equivalence can be deduced by using (3.17) and the fact that $\|g_k - \tilde{g}_k\| \leq \eta_k$.

**Theorem 3.3.** *Under the assumptions of Theorem* 3.2*, we have*

$$\lim_{k \to \infty} \|g_k\| = 0. \tag{3.29}$$

*Proof.* We prove the assertion by contradiction. If (3.29) does not hold, there exist a scalar $\epsilon > 0$ and a sequence $\{t_i\}$ such that

$$\|g_{t_i}\| \geq 4\epsilon, \quad \forall i. \tag{3.30}$$

Since $|\mathbb{S}| = \infty$, by (3.20), we can assume that $\{t_i\} \subseteq \mathbb{S}$. Since $\{\eta_k\}$ is diminishing, there exists $\bar{k}$ such that $|\eta_k| \leq \epsilon$ for all $k \geq \bar{k}$. Without loss of generality, assume that $t_i \geq \bar{k}$ for all $i$. By Theorem 3.2, for each $t_i$ there exists $l_i$ such that

$$l_i > t_i, \quad \|g_{l_i}\| < 3\epsilon. \tag{3.31}$$

From (3.30) and (3.31), it follows that

$$\|g_{l_i} - g_{t_i}\| \geq \|g_{t_i}\| - \|g_{l_i}\| \geq \epsilon. \tag{3.32}$$

By (3.20), we can assume that $l_i \in \mathbb{S}$. Further we assume that $l_i$ is the smallest integer in set $\mathbb{S}$ which satisfies (3.31). Then, for $k \in \mathbb{S}$ satisfying $t_i < k < l_i$, we have $\|g_k\| \geq 3\epsilon$. Let $\mathbb{K} := \{k \in \mathbb{S} : t_i \leq k < l_i\}$. Note that $k \geq \bar{k}$ for all $k \in \mathbb{K}$. Then we have $\eta_k \leq \epsilon$ for all $k \in \mathbb{K}$. Thus, we can deduce that

$$\|\tilde{g}_k\| \geq \|g_k\| - \eta_k \geq 2\epsilon, \quad \forall k \in \mathbb{K}. \tag{3.33}$$

Let

$$p_k := f(x_k) - \sum_{i=0}^{k} [2\delta_i + (1-c_1)t\zeta_i], \quad k \geq 0.$$

If $k \notin \mathbb{S}$, then $p_k \geq p_{k+1}$, otherwise, by (3.24), we have

$$p_k - p_{k+1} = f(x_k) - f(x_{k+1}) + [2\delta_k + (1-c_1)t\zeta_k] \geq \frac{\tau c_1}{2}\|\tilde{g}_k\|\,\|d_k\|. \tag{3.34}$$

Thus $\{p_k\}$ is a non-increasing sequence, and therefore $\lim_{k\to\infty} p_k$ exists. Further, we can deduce that

$$\begin{aligned} p_{t_i} - p_{l_i} &\geq \sum_{k\in\mathbb{K}} [f(x_k) - f(x_{k+1}) + 2\delta_k + (1-c_1)t\zeta_k] \\ &\geq \tau c_1 \epsilon \sum_{k=t_i, k\in\mathbb{S}}^{l_i-1} \|d_k\| = \tau c_1 \epsilon \sum_{k=t_i}^{l_i-1} \|x_{k+1} - x_k\| \geq \tau c_1 \epsilon \|x_{t_i} - x_{l_i}\|, \end{aligned} \tag{3.35}$$

where the second inequality is due to (3.33) and (3.34). Since $p_{t_i} - p_{l_i}$ converges to zero as $i \to \infty$, from (3.35) it follows that $\|x_{l_i} - x_{t_i}\| \to 0$. Thus, we have

$$\|g_{l_i} - g_{t_i}\| \leq L\|x_{l_i} - x_{t_i}\| \ \ \to \ \ 0,$$

yielding a contradiction to (3.32). □

### 3.2. Convergence rate analysis when $\{\delta_k\}$ and $\{\eta_k\}$ are diminishing

In this subsection, we establish the convergence rate of Algorithm 2.1. For the convenience of discussion, assume that the parameter $t$ and the sequence $\{\zeta_k\}$ satisfy

$$t = \frac{2}{1-c_2}, \quad \zeta_k = \delta_k, \quad \forall k. \tag{3.36}$$

Then (3.9) and (3.10) hold for $t$ and $\{\zeta_k\}$. We separate the contents of this subsection into three parts.

#### 3.2.1. $O(1/\epsilon^2)$ complexity

We now give some assumptions needed in this subsection.

**Assumption 3.5.** *The sequence* $\{\eta_k\}$ *satisfies*

$$\eta_k \leq (1/\bar{c} + 1)^{-1}\|g_k\|, \quad \forall k, \tag{3.37}$$

*where* $\bar{c}$ *is defined by* (3.11), *and the sequence* $\{\delta_k\}$ *satisfies*

$$\delta_k < \mathcal{A}\|g_k\|^2, \quad \forall k, \tag{3.38}$$

*where*

$$\mathcal{A} < \frac{8c_1\tau}{(\overline{\kappa} + a_3\overline{\sigma})(5 - c_2/2)^2\big(2 + t(1-c_1)\big)}. \tag{3.39}$$

It follows from (2.2) and (3.37) that

$$\frac{1+\bar{c}}{1+2\bar{c}}\|g_k\| \leq \|\tilde{g}_k\| \leq \frac{1+2\bar{c}}{1+\bar{c}}\|g_k\|,$$

that is, $\|g_k\| = O(\|\tilde{g}_k\|)$. Since we can only compute the value of $\tilde{g}_k$ in the numerical experiments, we set $\eta_k = \omega\|\tilde{g}_k\|$ in Section 5.3.

Under the previous assumptions, we can prove $\{\sigma_k\}$ is bounded in the following result. Let

$$\begin{aligned} N_0 &:= \left\lfloor \log_{a_1} \frac{\bar{\sigma}}{\sigma_0} \right\rfloor + 1, \\ N_1 &:= \max\{0, N_0\}. \end{aligned} \tag{3.40}$$

**Lemma 3.4.** *Suppose Assumptions* 3.1-3.3 *and* 3.5 *hold. Further suppose* (3.36) *holds for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. Then for all $k \geq N_1$, we have*

$$\sigma_k \leq a_3\bar{\sigma}. \tag{3.41}$$

*Proof.* Assume that $\sigma_0 \geq \bar{\sigma}$. Then $N_1 = N_0 \geq 1$. Now we prove

$$\text{there exists} \quad k_0 \in \{0, 1, \ldots, N_0\} \quad \text{such that} \quad \sigma_{k_0} < \bar{\sigma}. \tag{3.42}$$

By (3.37) and Lemma 3.3, we know that if $\sigma_k \geq \bar{\sigma}$, then the $k$-th iteration is very successful, which implies $\sigma_{k+1} \leq a_1\sigma_k$. Thus, if $\sigma_i \geq \bar{\sigma}$ for $i = 0, \ldots, N_0 - 1$, then $\sigma_{N_0} < \bar{\sigma}$. That is, (3.42) holds. Next, we prove (3.41) holds for all $k \geq k_0$. We prove it by induction. Obviously, it holds for $k = k_0$. Assume that (3.41) holds for some $k = j \geq k_0$. If $\sigma_j < \bar{\sigma}$, by the procedures of Algorithm 2.1 (see steps 4 and 8), we can deduce that $\sigma_{j+1} \leq a_3\bar{\sigma}$; if $\sigma_j \geq \bar{\sigma}$, from Lemma 3.3 it follows that $\sigma_{j+1} \leq a_1\sigma_j$, which together with the induction hypothesis implies $\sigma_{j+1} < a_3\bar{\sigma}$. Thus (3.41) holds for $k = j + 1$, and therefore the assertion holds.

If $\sigma_0 < \bar{\sigma}$, then $N_1 = 0$. Similarly, we can prove that (3.41) holds for all $k \geq 0$ by induction. We omit the detail. □

Recall $\mathbb{S}$ is defined by (3.19). We write it as $\mathbb{S} = \{k_1, k_2, \cdots\}$. From the procedures of Algorithm 2.1, we have

$$\frac{\sigma_{k+1}}{\sigma_k} \geq a_0, \quad \text{if} \quad k \in \mathbb{S}, \tag{3.43}$$

$$\frac{\sigma_{k+1}}{\sigma_k} \geq a_2, \quad \text{if} \quad k \notin \mathbb{S}. \tag{3.44}$$

Moreover, for the index $k$ satisfying $k_i < k \leq k_{i+1}$, by (3.20), we have $f(x_k) = f(x_{k_{i+1}})$. In the following result, we study the properties of the sequence $\{f(x_{k_i})\}$.

We use the notation

$$\phi := \frac{32c_1\tau}{(\overline{\kappa} + a_3\bar{\sigma})(10 - c_2)^2} - \big(2 + (1 - c_1)t\big)\mathcal{A}, \tag{3.45}$$

where $\mathcal{A}$ is defined by (3.39). From Assumption 3.5 it follows that $\phi > 0$. By (3.13), we have

$$\phi \leq \frac{32c_1\tau}{(\overline{\kappa} + a_3\bar{\sigma})(10 - c_2)^2} \leq \frac{32}{81\bar{\sigma}} \leq \frac{32}{81L/2\bar{c}} \leq \frac{16}{81L}. \tag{3.46}$$

**Lemma 3.5.** *Suppose Assumptions* 3.1-3.3 *and* 3.5 *hold. Further suppose* (3.36) *holds for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. Then for all $k_i \geq N_1$,*

$$f(x_{k_i}) - f(x_{k_{i+1}}) \geq \phi\|g_{k_i}\|^2, \quad \forall i \geq 1. \tag{3.47}$$

*Proof.* For all $k \geq N_1$, by Assumption 3.3 and (3.41), we can deduce that

$$\|\tilde{g}_k\| = \| - (B_k + \sigma_k I)d_k\| \leq (\|B_k\| + \sigma_k)\|d_k\| \leq (\overline{\kappa} + a_3\bar{\sigma})\|d_k\|. \tag{3.48}$$

Assume that $k_i \geq N_1$. By (3.24), (3.36) and (3.48), we have

$$\begin{aligned} f(x_{k_i}) - f(x_{k_{i+1}}) &\geq \frac{c_1\tau}{2}\|\tilde{g}_{k_i}\|\,\|d_{k_i}\| - \big(2\delta_{k_i} + (1-c_1)t\zeta_{k_i}\big) \\ &\geq \frac{c_1\tau}{2(\overline{\kappa} + a_3\bar{\sigma})}\|\tilde{g}_{k_i}\|^2 - \big(2 + (1-c_1)t\big)\delta_{k_i}. \end{aligned} \tag{3.49}$$

From (3.37), it follows that

$$\|\tilde{g}_{k_i}\| \geq \|g_{k_i}\| - \eta_{k_i} \geq \frac{1}{1+\bar{c}}\|g_{k_i}\|. \tag{3.50}$$

Combining (3.11), (3.38), (3.49) and (3.50) yields

$$\begin{aligned} f(x_{k_i}) - f(x_{k_{i+1}}) &\geq \frac{32c_1\tau}{(\overline{\kappa} + a_3\bar{\sigma})(10-c_2)^2}\|g_{k_i}\|^2 - \big(2 + (1-c_1)t\big)\delta_{k_i} \\ &\geq \left(\frac{32c_1\tau}{(\overline{\kappa} + a_3\bar{\sigma})(10-c_2)^2} - \big(2 + (1-c_1)t\big)\mathcal{A}\right)\|g_{k_i}\|^2. \end{aligned}$$

That is, (3.47) holds. □

It is easy to prove the following result and the proof is omitted.

**Corollary 3.1.** *Under the assumptions of Lemma* 3.5*, we have*

$$\lim_{k\to\infty}\|g_k\| = 0.$$

For a positive integer $k$, assume that $k_N < k \leq k_{N+1}$ for some $N \geq 1$. By (3.43) and (3.44), we have

$$\sigma_k = \sigma_0\prod_{j=0}^{k-1}\frac{\sigma_{j+1}}{\sigma_j} = \sigma_0\prod_{\substack{j\in\mathbb{S}\\ j<k}}\frac{\sigma_{j+1}}{\sigma_j}\prod_{\substack{j\notin\mathbb{S}\\ j<k}}\frac{\sigma_{j+1}}{\sigma_j} \geq \sigma_0 a_0^N a_2^{k-N},$$

which together with (3.41) implies (assume that $k \geq N_1$)

$$\ln(a_3\bar{\sigma}) \geq \ln\sigma_k \geq \ln\sigma_0 + N\ln a_0 + (k-N)\ln a_2. \tag{3.51}$$

Thus, we can deduce that

$$k \leq \log_{a_2}\left(\frac{a_2}{a_0}\right)N + \log_{a_2}\frac{a_3\bar{\sigma}}{\sigma_0}. \tag{3.52}$$

Next, we give the iteration complexity analysis of the Algorithm 2.1.

**Theorem 3.4.** *Suppose Assumptions* 3.1-3.3 *and* 3.5 *hold. Further suppose* (3.36) *holds for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. Let*

$$\gamma_1 = \log_{a_2}\left(\frac{a_2}{a_0}\right)\frac{1}{\phi}\left[\sum_{i=0}^{N_1}2(2-c_1)\delta_i + f(x_0) - f(x^*)\right], \tag{3.53}$$

$$\gamma_2 = \log_{a_2}\left(\frac{a_2}{a_0}\right)N_1 + \log_{a_2}\frac{a_3\bar{\sigma}}{\sigma_0}, \tag{3.54}$$

*where $N_1$ is defined by* (3.40)*. If $k \geq \gamma_1/\epsilon^2 + \gamma_2$, then*

$$\min_{j=0,\ldots,k}\|g_j\| < \epsilon.$$

*Proof.* Let $k_{i_0}$ be the smallest integer in $\mathbb{S}$ such that $k_{i_0} \geq N_1$. We consider two cases $\sigma_0 \geq \bar{\sigma}$ or $\sigma_0 < \bar{\sigma}$.

1) Assume that $\sigma_0 \geq \bar{\sigma}$. Then $N_1 = N_0 \geq 1$. From (3.40), it is easy to see that $i_0 \leq N_1 + 1$. For any integer $k > k_{i_0}$, assume that $k_N < k \leq k_{N+1}$ for some $N$. If $\|g_j\| \geq \epsilon$ for all $j \in \{0, \cdots, k\}$, by (3.47), we have

$$f(x_{k_{i_0}}) - f(x^*) \geq f(x_{k_{i_0}}) - f(x_{k_{N+1}}) \geq \phi \sum_{i=i_0}^{N} \|g_{k_i}\|^2 \geq \phi(N - i_0 + 1)\epsilon^2. \tag{3.55}$$

From $\sigma_0 \geq \bar{\sigma}$ and Lemma 3.3, it follows that $x_{k_1} = x_0$. From (3.24), we can deduce that

$$f(x_{k_{i_0}}) \leq f(x_0) + \sum_{i=0}^{N_1} \big(2\delta_i + (1 - c_1)t\zeta_i\big).$$

Combining this and above inequality, we have

$$N \leq \frac{1}{\phi\epsilon^2}\left(\sum_{i=0}^{N_1} 2(2 - c_1)\delta_i + f(x_0) - f(x^*)\right) + N_1,$$

which together with (3.52) implies $k < \gamma_1/\epsilon^2 + \gamma_2$, where $\gamma_1$ and $\gamma_2$ are defined by (3.53) and (3.54).

2) Assume that $\sigma_0 < \bar{\sigma}$. Then $N_1 = 0$, and therefore $i_0 = 1$. From (3.20), we can deduce that $x_0 = x_{k_1}$. Then the assertion follows directly from (3.55). □

### 3.2.2. Linear convergence

Under the Polyak-Lojasiewicz condition, we can establish the linear convergence rate of Algorithm 2.1.

**Assumption 3.6.** *Assume that $f$ satisfies the Polyak-Lojasiewicz inequality* [34]*, that is, for some $\mu > 0$, the following inequality holds:*

$$\|g(x)\|^2 \geq 2\mu\big(f(x) - f(x^*)\big), \quad \forall x \in \mathbb{R}^n. \tag{3.56}$$

If $g$ satisfies Assumption 3.2, then $\|g(x)\|^2 \leq 2L(f(x) - f(x^*))$, which together with (3.56) implies $L \geq \mu$. Recall that $\phi$ is defined by (3.45). By (3.46) and $L \geq \mu$, we have $2\mu\phi < 1$. Then we can prove the following result. We use the notation $N_1$, which is defined by (3.40).

**Theorem 3.5.** *Suppose Assumptions* 3.1-3.6 *except* 3.4 *hold. Further suppose* (3.36) *holds for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. Then for all $k \geq N_1$, we have*

$$f(x_k) - f(x^*) \leq \Theta\nu^k,$$

*where*

$$\nu := (1 - 2\mu\phi)^{1/\log_{a_2}(a_2/a_0)}, \quad \Theta = \nu^{-N_1 - \log_{a_2}(a_3\bar{\sigma}/\sigma_0 a_0^{N_1})}\big(f(x_{N_1}) - f(x^*)\big).$$

*Proof.* Pick any $k \geq N_1$. Assume that $k_N$ is the smallest integer in $\mathbb{S}$ such that $k \leq k_N$. Then $x_k = x_{k_N}$. Assume that $k_{\widetilde{N}}$ is the smallest integer in $\mathbb{S}$ such that $N_1 \leq k_{\widetilde{N}}$. Then $x_{N_1} = x_{k_{\widetilde{N}}}$. For any $i \geq \widetilde{N}$, by (3.47) and (3.56), we can deduce that

$$\begin{aligned} f(x_{k_{i+1}}) - f(x^*) &= f(x_{k_{i+1}}) - f(x_{k_i}) + f(x_{k_i}) - f(x^*) \\ &\leq \big(f(x_{k_i}) - f(x^*)\big) - \phi\|g_{k_i}\|^2 \\ &\leq (1 - 2\mu\phi)\big(f(x_{k_i}) - f(x^*)\big). \end{aligned}$$

Using the above inequalities recursively from $\widetilde{N}$ to $N$, we have

$$f(x_{k_N}) - f(x^*) \leq (1 - 2\mu\phi)^{N-\widetilde{N}}\big(f(x_{k_{\widetilde{N}}}) - f(x^*)\big). \tag{3.57}$$

Similar to the derivation of (3.52), we can deduce that

$$\begin{aligned} k - N_1 &\leq \log_{a_2}\left(\frac{a_2}{a_0}\right)(N - \widetilde{N}) + \log_{a_2}\frac{a_3\bar{\sigma}}{\sigma_{N_1}} \\ &\leq \log_{a_2}\left(\frac{a_2}{a_0}\right)(N - \widetilde{N}) + \log_{a_2}\frac{a_3\bar{\sigma}}{\sigma_0 a_0^{N_1}}, \end{aligned} \tag{3.58}$$

where the last inequality follows from $\sigma_{N_1} \geq \sigma_0 a_0^{N_1}$ (see (3.43) and (3.44)). Combining (3.57) and (3.58) yields the assertion. □

### 3.2.3. Superlinear convergence

In this subsection, we establish the superlinear convergence of Algorithm 2.1. Assume that $f(x)$ is twice continuously differentiable. We use $G(x)$ to denote $\nabla^2 f(x)$. The following assumptions arise commonly in the proof of superlinear convergence.

**Assumption 3.7.** *$G(x)$ is Lipschitz continuous near $x^*$, that is*

$$\|G(x) - G(y)\| \leq L_G\|x - y\| \tag{3.59}$$

*for all $x, y$ near $x^*$, where $L_G > 0$. Meanwhile, there exist positive constant $m$ and $M$ such that*

$$m\|z\|^2 \leq z^\top G(x)z \leq M\|z\|^2, \quad \forall z \in \mathbb{R}^n \tag{3.60}$$

*for all $x$ near $x^*$.*

**Assumption 3.8.** *There exists $\underline{\kappa} > 0$ such that $\underline{\kappa}I \preceq B_k \preceq \overline{\kappa}I$ for all $k \geq 0$.*

**Assumption 3.9.** *Assume that $x_k$ converges to $x^*$, and $B_k$ satisfies the following Dennis-Móre condition* [19]*:*

$$\frac{\big\|\big(B_k - G(x^*)\big)d_k\big\|}{\|d_k\|} \to 0, \quad \textit{whenever} \quad \|g_k\| \to 0. \tag{3.61}$$

**Assumption 3.10.** *The sequence $\{\delta_k\}$ satisfies* (3.38)*. Assume that $g_k \neq 0$ for all $k \geq 1$. The sequence $\{\eta_k\}$ satisfies $\eta_k \geq \eta_{k+1}$ and*

$$\lim_{k\to\infty}\frac{\eta_k}{\|g_k\|} = 0. \tag{3.62}$$

Inspired by [13, Theorem 4.3], we can prove the following result.

**Lemma 3.6.** *Suppose Assumptions* 3.1 *and* 3.7-3.10 *hold. Further suppose* (3.9) *and* (3.10) *hold, and* $\tau \le \underline{\kappa}/\overline{\kappa}$. *Then the iterations generated by Algorithm* 2.1 *are eventually very successful, and* $\sigma_k$ *converges to zero as* $k$ *goes to infinity.*

*Proof.* By Assumption 3.9, we know that $x_k \to x^*$ and therefore $\|g_k\| \to 0$. Under the conditions of Assumptions 3.1, 3.7 and 3.8, similar to the proof of Lemma 3.4, we can prove that $\sigma_k \le a_3\bar{\sigma}$ for all sufficiently large $k$. Since $d_k = -(B_k + \sigma_k I)^{-1}\tilde{g}_k$, by Assumption 3.8,

$$(\overline{\kappa} + a_3\bar{\sigma})^{-1}\|\tilde{g}_k\| \le \|d_k\| \le \underline{\kappa}^{-1}\|\tilde{g}_k\| \tag{3.63}$$

for $k$ large enough.

For any $\sigma > 0$, from Assumption 3.8 we can deduce that

$$\tilde{g}_k(B_k + \sigma I)^{-1}\tilde{g}_k \ge \frac{1}{\overline{\kappa} + \sigma}\|\tilde{g}_k\|^2 \ge \frac{\tau}{\underline{\kappa} + \sigma}\|\tilde{g}_k\|^2 \ge \tau\|\tilde{g}_k\|\big\|(B_k + \sigma)^{-1}\tilde{g}_k\big\|. \tag{3.64}$$

Thus, after steps 3-5 of Algorithm 2.1, the value of $\sigma_k$ remains the same.

By (3.8)-(3.10), taking into account $(B_k + \sigma_k I)d_k = -\tilde{g}_k$, we have

$$\begin{aligned}
r_k &= [f_{\delta_k}(x_k) - f_{\delta_k}(x_k + d_k) + t\zeta_k] - c_2[f_{\delta_k}(x_k) - m_k(d_k) + t\zeta_k] \\
&\ge [f(x_k) - f(x_k + d_k)] - c_2[f_{\delta_k}(x_k) - m_k(d_k)] \\
&= -\left(g_k^\top d_k + \frac{1}{2}d_k^\top G(\omega_k)d_k\right) + c_2\left(\tilde{g}_k^\top d_k + \frac{1}{2}d_k^\top(B_k + \sigma_k I)d_k\right) \\
&= (\tilde{g}_k - g_k)^\top d_k + \frac{1}{2}d_k^\top\big(B_k - G(\omega_k)\big)d_k - \frac{1-c_2}{2}\tilde{g}_k^\top d_k + \frac{1}{2}\sigma_k\|d_k\|^2 \\
&\ge \|d_k\|\left(-\eta_k - \frac{1}{2}\big\|\big(B_k - G(\omega_k)\big)d_k\big\| + \frac{\tau(1-c_2)}{2}\|\tilde{g}_k\| + \frac{1}{2}\sigma_k\|d_k\|\right),
\end{aligned} \tag{3.65}$$

where $\omega_k \in (x_k, x_k+d_k)$, and the last inequality is due to (3.64). It follows from Assumptions 3.7 and 3.9 that

$$\begin{aligned}
\big\|\big(B_k - G(\omega_k)\big)d_k\big\| &\le \big\|\big(B_k - G(x^*)\big)d_k\big\| + \big\|\big(G(x_k) - G(x^*)\big)d_k\big\| \\
&\quad + \big\|\big(G(x_k) - G(\omega_k)\big)d_k\big\| = o(\|d_k\|).
\end{aligned} \tag{3.66}$$

From (3.62) and (3.63), it follows that $\eta_k = o(\|g_k\|) = o(\|d_k\|)$. By (3.63), (3.65) and (3.66), we have $r_k \ge 0$ for all sufficiently large $k$, that is, all iterations are eventually very successful. By the first assertion, from (3.64) and step 8 of Algorithm 2.1, it follows that $\sigma_k \to 0$. □

Now we establish the superlinear convergence rate of our method.

**Theorem 3.6.** *Suppose the same assumptions hold as in Lemma* 3.6. *Then the sequence* $\{x_k\}$ *converges superlinearly to* $x^*$.

*Proof.* By (3.61), taking into account $x_k \to x^*$ and (3.59), we have

$$\big\|\big(G(x_k) - B_k\big)d_k\big\| = o(\|d_k\|).$$

It follows from Assumptions 3.7 and 3.8 that $\|G(x_k) - B_k\| \le M + \overline{\kappa}$ for sufficiently large $k$. By Lemma 3.6, we know that $\sigma_k \to 0$ and $\eta_k = o(\|d_k\|)$. Using this, we can deduce that

$$\|x_{k+1} - x^*\| = \|x_k + d_k - x^*\|$$

$$
\begin{aligned}
&= \big\|x_k - G(x_k)^{-1}g_k - x^* + G(x_k)^{-1}g_k - (B_k+\sigma_k I)^{-1}\tilde{g}_k\big\| \\
&\le \big\|x_k - G(x_k)^{-1}g_k - x^*\big\| + \big\|G(x_k)^{-1}(g_k - \tilde{g}_k)\big\| \\
&\quad + \big\|G(x_k)^{-1}\tilde{g}_k - B_k^{-1}\tilde{g}_k\big\| + \big\|B_k^{-1}\tilde{g}_k - (B_k+\sigma_k I)^{-1}\tilde{g}_k\big\| \\
&\le O\big(\|x_k - x^*\|^2\big) + \eta_k\big\|G(x_k)^{-1}\big\| \\
&\quad + \big\|G(x_k)^{-1}\big(G(x_k) - B_k\big)\big(I + \sigma_k B_k^{-1}\big)d_k\big\| + \big\|\sigma_k B_k^{-1} d_k\big\| \\
&\le O\big(\|x_k - x^*\|^2\big) + o(\|d_k\|) \\
&\quad + \big\|G(x_k)^{-1}\big(G(x_k) - B_k\big)\sigma_k B_k^{-1} d_k\big\| + \big\|G(x_k)^{-1}\big(G(x_k) - B_k\big)d_k\big\| \\
&= O\big(\|x_k - x^*\|^2\big) + o(\|d_k\|). \qquad (3.67)
\end{aligned}
$$

From the above inequality and using $d_k = x_{k+1} - x_k$, we have $\|d_k\| = O(\|x_k - x^*\|)$, which together with (3.67) implies

$$\|x_{k+1} - x^*\| = o\big(\|x_k - x^*\|\big).$$

The proof is complete. □

### 3.3. Convergence rate analysis when $\{\delta_k\}$ and $\{\eta_k\}$ are bounded

In this subsection, we discuss the case that $\{\delta_k\}$ and $\{\eta_k\}$ are uniformly bounded, i.e. there exist nonnegative constants $\epsilon_f, \epsilon_g$ such that

$$\delta_k \le \epsilon_f, \quad \eta_k \le \epsilon_g, \quad \forall k. \qquad (3.68)$$

Recall that $\bar{c}$ is defined by (3.11). The following result follows directly from Lemma 3.4 and (3.68).

**Lemma 3.7.** *Suppose Assumptions* 3.1-3.3 *hold. Further assume* (3.36) *holds for the parameter $t$, sequences $\{\delta_k\}$ and $\{\zeta_k\}$. If there exists $\bar{k}$ such that*

$$\|g_k\| > (1/\bar{c} + 1)\epsilon_g, \quad \forall k \ge \bar{k}, \qquad (3.69)$$

*then there exists $\check{k} \ge \bar{k}$ such that for all $k \ge \check{k}$,*

$$\sigma_k \le a_3\bar{\sigma},$$

*where $\bar{\sigma}$ is defined in* (3.13).

Before proceeding more, we introduce the notations

$$
\begin{aligned}
&\theta_1 := \frac{(\overline{\kappa} + a_3\bar{\sigma})}{c_1\tau}, \quad \theta_2 := \frac{2 - c_1 - c_2}{1 - c_2}, \\
&e_g := \max\Big\{(1/\bar{c} + 1)\epsilon_g, \sqrt{(1 + 4\theta_1\theta_2)\epsilon_f} + \epsilon_g\Big\}. \qquad (3.70)
\end{aligned}
$$

**Lemma 3.8.** *Suppose Assumptions* 3.1-3.3 *hold,* (3.36) *holds for $t, \{\delta_k\}$ and $\{\zeta_k\}$. Further suppose for some $k > 0$,*

$$\|g_k\| \ge e_g, \qquad (3.71)$$

*and $\sigma_k \le a_3\bar{\sigma}$. Then, if $k \in \mathbb{S}$, we have*

$$f(x_k) - f(x_k + d_k) \ge \frac{1}{2\theta_1}\epsilon_f.$$

*Proof.* By $\sigma_k \le a_3\bar{\sigma}$, we have

$$\|\tilde{g}_k\| = \|(B_k + \sigma_k I)d_k\| \le (\overline{\kappa} + a_3\bar{\sigma})\|d_k\|.$$

Since $k \in \mathbb{S}$, by (2.5), (3.36) and (3.68), we can deduce that

$$\begin{aligned} f(x_k) - f(x_k + d_k) &\ge f_{\delta_k}(x_k) - f_{\delta_k}(x_k + d_k) - 2\delta_k \\ &\ge c_1\big(m_k(0) - m_k(d_k)\big) - (1 - c_1)t\zeta_k - 2\delta_k \\ &\ge \frac{c_1\tau}{2}\|\tilde{g}_k\|\,\|d_k\| - \left(2 + 2\frac{1-c_1}{1-c_2}\right)\delta_k \\ &\ge \frac{c_1\tau}{2(\overline{\kappa} + a_3\bar{\sigma})}\|\tilde{g}_k\|^2 - \left(2 + 2\frac{1-c_1}{1-c_2}\right)\epsilon_f. \end{aligned} \tag{3.72}$$

From (3.68) and (3.71), it follows that

$$\|\tilde{g}_k\| \ge \|g_k\| - \epsilon_g \ge \sqrt{(1 + 4\theta_1\theta_2)\epsilon_f}. \tag{3.73}$$

Combining (3.72) and (3.73) yields the assertion. □

Next we establish the main result of this subsection.

**Theorem 3.7.** *Under the assumptions of Lemma* 3.8*, the sequence of iterates* $\{x_k\}$ *generated by Algorithm* 2.1 *infinitely visits the critical region* $C_1$*, defined as*

$$C_1 = \{x : \|g(x)\| \le e_g\}, \tag{3.74}$$

*where* $e_g$ *is defined by* (3.70).

*Proof.* We prove the assertion by contradiction. Assume that there exists $\bar{k}$ such that

$$\|g_k\| > e_g, \quad \forall k > \bar{k}.$$

First, we prove $|\mathbb{S}| = \infty$. If not, by the procedures of Algorithm 2.1, we can deduce that $\sigma_k \to \infty$ as $k \to \infty$. Let $\hat{k}$ be the largest integer in $\mathbb{S}$. Then there exists $\tilde{k} > \max\{\bar{k}, \hat{k}\}$ such that $\sigma_{\tilde{k}} \ge \bar{\sigma}$. Note that

$$\|g_{\tilde{k}}\| \ge e_g \ge (1/\bar{c} + 1)\eta_{\tilde{k}}.$$

By Lemma 3.3, we can see that $\tilde{k} \in \mathbb{S}$, giving a contradiction.

By Lemma 3.7, we know that there exists $\check{k} \ge \bar{k}$ such that

$$\sigma_k \le a_3\bar{\sigma}, \quad \forall k \ge \check{k}.$$

For any $k \in \mathbb{S}$ satisfying $k \ge \check{k}$, using Lemma 3.8, we can obtain

$$f(x_k) - f(x_k + d_k) \ge \frac{1}{2\theta_1}\epsilon_f.$$

Assume that $k_{i_0-1} < \check{k} \le k_{i_0}$ for some $i_0$. Then $x_{\check{k}} = x_{k_{i_0}}$. Using the above inequality, we have

$$\begin{aligned} f(x_{k_{i_0}}) - f(x^*) &\ge f(x_{k_{i_0}}) - f(x_{k_{i_0+j}}) \\ &= \sum_{t=0}^{j-1}\left[f(x_{k_{i_0+t}}) - f(x_{k_{i_0+t+1}})\right] \ge \frac{j}{2\theta_1}\epsilon_f. \end{aligned}$$

Letting $j \to \infty$ in the above inequality, we can derive a contradiction. □

## 4. Practical Implementation

In this section, we provide several techniques to make our method more efficient.

### 4.1. Subspace implementation

Inspired by [37], we use subspace techniques to reduce the amount of computation. Subspace techniques play an important role in the development of numerical methods for large-scale nonlinear optimization. We refer to [41] for a detailed discussion on these techniques.

From the Algorithm 2.1, we know that $B_k$ is updated only if $k \in \mathbb{S}$. Note that $x_{k_{i+1}} = x_{k_i+1}$ for all $i$. To obtain $B_{k_{i+1}}$, we compute $s_{k_i}$ and $\tilde{y}_{k_i}$ as follows:

$$s_{k_i} = x_{k_{i+1}} - x_{k_i}, \quad \tilde{y}_{k_i} = \tilde{g}_{k_{i+1}} - \tilde{g}_{k_i}. \tag{4.1}$$

Let $\mathcal{G}_i$ be the subspace generated by the following formula:

$$\mathcal{G}_i := \operatorname{span}\{\tilde{g}_{k_1}, \cdots, \tilde{g}_{k_i}\}. \tag{4.2}$$

Let $Z_i := [z_1, \cdots, z_{l_i}]$ be the orthonormal basis of $\mathcal{G}_i$, where $l_i$ is the dimension of $\mathcal{G}_i$. It is easy to see that $Z_i^T Z_i = I_{l_i}$. Let

$$\hat{g}_i := Z_i^T \tilde{g}_{k_i} \in \mathbb{R}^{l_i}, \quad \mathbb{B}_i := Z_i^T B_{k_i} Z_i \in \mathbb{R}^{l_i \times l_i}. \tag{4.3}$$

The proof of the following result is similar to that of [37], and therefore we omit it.

**Lemma 4.1.** *Suppose $B_0 = \delta I$ for some $\delta > 0$. Then problem (2.4) is equivalent to the following problem:*

$$\min_{\bar{d} \in \mathbb{R}^{l_i}} \bar{m}_i(\bar{d}) = f_{\delta_{k_i}}(x_{k_i}) + \hat{g}_i^\top \bar{d} + \frac{1}{2}\bar{d}^\top \mathbb{B}_i \bar{d} + \frac{1}{2}\sigma_{k_i}\|\bar{d}\|^2 \tag{4.4}$$

*in the sense that if $d_{k_i}$ is a solution of (2.4), then $\bar{d}_i = Z_i^\top d_{k_i}$ is a solution of (4.4), if $\bar{d}_i$ is a solution of (4.4), then $d_{k_i} = Z_i \bar{d}_i \in \mathcal{G}_i$ must be a solution of (2.4).*

The cost of updating $\mathbb{B}_i$ is significantly less than that of updating $B_{k_i}$ when $i$ is much smaller than $n$. Given $Z_i, \mathbb{B}_i$ and $\hat{g}_i$, after $\bar{d}_i$ is obtained, we have a convenient way of computing $Z_{i+1}$, $\mathbb{B}_{i+1}$ and $\hat{g}_{i+1}$. Now we give a detailed description. Let $z_{i+1}$ be a unit vector in $\mathcal{G}_i^\perp$, which satisfies

$$\phi_{i+1} z_{i+1} = \tilde{g}_{k_{i+1}} - Z_i u_i, \tag{4.5}$$

where

$$\phi_{i+1} = \left\| \left(I - Z_i Z_i^T\right) \tilde{g}_{k_{i+1}} \right\|, \quad u_i = Z_i^T \tilde{g}_{k_{i+1}}.$$

We consider the following two cases:

1. If $\phi_{i+1} > 0$, we set $Z_{i+1} = [Z_i, z_{i+1}]$. By (4.5), we have

$$
\begin{aligned}
\hat{g}_{i+1} &:= Z_{i+1}^T \tilde{g}_{k_{i+1}} = \begin{bmatrix} Z_i^T \tilde{g}_{k_{i+1}} \\ z_{i+1}^T \tilde{g}_{k_{i+1}} \end{bmatrix} = \begin{bmatrix} u_i \\ \phi_{i+1} \end{bmatrix}, \\
\hat{s}_i &:= Z_{i+1}^T s_{k_i} = \begin{bmatrix} Z_i^T s_{k_i} \\ z_{i+1}^T s_{k_i} \end{bmatrix} = \begin{bmatrix} \bar{d}_i \\ 0 \end{bmatrix}, \\
\hat{y}_i &:= Z_{i+1}^T \tilde{y}_{k_i} = \begin{bmatrix} c Z_i^T (\tilde{g}_{k_{i+1}} - \tilde{g}_{k_i}) \\ z_{i+1}^T (\tilde{g}_{k_{i+1}} - \tilde{g}_{k_i}) \end{bmatrix} = \begin{bmatrix} u_i - \hat{g}_i \\ \phi_{i+1} \end{bmatrix}, \\
\hat{B}_i &:= Z_{i+1}^T B_{k_i} Z_{i+1} = \begin{bmatrix} \mathbb{B}_i & 0 \\ 0 & \delta \end{bmatrix}.
\end{aligned} \tag{4.6}
$$

2. If $\phi_{i+1} = 0$, set $Z_{i+1} = Z_i$. Then we have

$$
\begin{aligned}
\hat{g}_{i+1} &:= Z_i^T \tilde{g}_{k_{i+1}} = u_i, \\
\hat{s}_i &:= Z_i^T s_{k_i} = \bar{d}_i, \\
\hat{y}_i &= Z_i^T (\tilde{g}_{k_{i+1}} - \tilde{g}_{k_i}) = u_i - \hat{g}_i, \\
\hat{B}_i &= Z_i^T B_{k_i} Z_i = \mathbb{B}_i.
\end{aligned} \tag{4.7}
$$

Since $s_{k_i} \in \mathcal{G}_i \subset \mathcal{G}_{i+1}$, it follows that $Z_{i+1} Z_{i+1}^\top s_{k_i} = s_{k_i}$. From (4.6) and (4.7), we can deduce that

$$
\mathbb{B}_{i+1} = Z_{i+1}^T B_{k_{i+1}} Z_{i+1} = \hat{B}_i + \frac{\hat{y}_i \hat{y}_i^\top}{\hat{y}_i^\top \hat{s}_i} - \frac{\hat{B}_i \hat{s}_i \hat{s}_i^\top \hat{B}_i}{\hat{s}_i^\top \hat{B}_i \hat{s}_i}.
$$

Note that the dimension of $\mathcal{G}_i$ will become very large during the process of iteration. To overcome this difficulty, we restart the updating procedure every $\varpi$ steps. That is, for all $j \geq 0$, we set

$$
Z_{j\varpi+1} = \begin{bmatrix} \dfrac{\tilde{g}_{k_{j\varpi+1}}}{\|\tilde{g}_{k_{j\varpi+1}}\|} \end{bmatrix}, \quad B_{j\varpi+1} = \delta > 0.
$$

The subspace implementation can reduce the amount of computation significantly in early iterations, especially when $n$ is very large. So we use this technique for some large scale problems.

### 4.2. Adjustment for $\sigma_k$ and termination criterion

To improve the performance of our algorithm, we use a strategy to adjust the regularized parameter $\sigma_k$. Now we describe it. When $\rho_k < c_1$ and the iteration $k$ is unsuccessful, we use the following formula to adjust $\sigma_k$:

$$
\sigma_{k+1} = \begin{cases} a_3 \sigma_k, & \rho_k \leq 0, \\ \dfrac{a_3(c_1 - \rho_k) + a_2 \rho_k}{c_1} \sigma_k, & 0 < \rho_k < c_1. \end{cases}
$$

When $\rho_k \geq c_2$ and the iteration $k$ is very successful, we adjust $\sigma_k$ by

$$
\sigma_{k+1} = \begin{cases} \dfrac{a_0(\rho_k - c_2) + a_1(1 - \rho_k)}{1 - c_2} \sigma_k, & c_2 \leq \rho_k < 1, \\ a_0 \sigma_k, & \rho_k \geq 1. \end{cases}
$$

We terminate the algorithm as soon as $\|\tilde{g}_k\| \leq tol$, where $tol$ is a user-specified parameter. When error sequences $\delta_k, \eta_k$ are uniformly bounded by $\epsilon_f$ and $\epsilon_g$, $tol$ can not be too small compared to $\epsilon_f, \epsilon_g$. For more discussions on termination criteria for optimization with inexact first-order information, readers are referred to [3, 15, 26].

## 5. Numerical Experiments

In this section, we present numerical results to illustrate the efficiency of methods proposed in this paper. In all of our tests, we set the parameters in Algorithm 2.1 as follows: $\sigma_0 = 1$, $\tau = 0.01, c_1 = 0.2, c_2 = 0.5, a_0 = 0.3, a_1 = 0.5, a_2 = 3, a_3 = 5$. All numerical experiments were implemented in MATLAB R2020b on a laptop running macOS Monterey with an Intel Core i5-4260U Processor. In what follows, $f_{\delta_k}(x_k)$ is denoted as $\tilde{f}_k$. In summary, the six tested methods are:

**IGD:** The inexact gradient descent method with the iterative scheme $x_{k+1} = x_k - \alpha_k \tilde{g}_k$. The stepsize $\alpha_k$ is chosen to satisfy the following approximate line search condition:

$$f_{\delta_k}(x_k + \alpha_k d_k) \leq f_{\delta_k}(x_k) + \alpha_k \tilde{g}_k^T d_k + \zeta_k,$$

where $\{\zeta_k\}$ is the positive sequence defined in (2.6) and $d_k$ is set as $d_k = -\tilde{g}_k$.

**CGM:** The gradient method proposed in [21].

**FGM:** The fast gradient method proposed in [21].

**BFGSe:** The BFGS method with errors proposed in [40], which incorporates the Armijo-Wolfe line search and lengthening procedure.

**adaQN:** The adaptive regularized quasi-Newton method given by Algorithm 2.1.

**adaQNsub:** The adaptive regularized quasi-Newton method with the subspace technique given by Algorithm 2.1.

In our numerical experiments, we consider three types of problems, which are described in detail later. For all numerical experiments, we randomly generate 50 instances and record the averaged numerical performance of these instances. For the stopping criterion, we terminate all algorithms when $\|\tilde{g}_k\| < tol$, or the algorithm reaches the maximum iteration number $max\_iter$. For each methods, we use the parameters that give the best performance of the algorithm. Numerical results are shown in several tables and figures.

### 5.1. The case of quadratic function

We first consider the simple quadratic function

$$f(x) = x^\top D x,$$

where $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix. Such a problem is consider in [40]. We inject uniformly distributed errors in the evaluations of function and gradient. Let

$$X_f \sim U(-\epsilon_f, \epsilon_f), \quad X_g \sim \mathbb{B}_n(0, \epsilon_g), \tag{5.1}$$

where $U(-a, a)$ denotes the uniform distribution from $-a$ to $a$, and $\mathbb{B}_n(0, b)$ denotes the uniform distribution on the $n$ dimensional ball centered at 0 with radius $b$. We consider four different types of errors to test the performance of the proposed methods

$$f_k - \tilde{f}_k = \frac{X_f}{k^2}, \qquad g_k - \tilde{g}_k = \frac{X_g}{k}, \qquad \zeta_k = \frac{\epsilon_f}{k^2}, \tag{5.2}$$

$$f_k - \tilde{f}_k = X_f \|g_k\|^2, \quad g_k - \tilde{g}_k = X_g g_k, \quad \zeta_k = \epsilon_f \|g_k\|^2, \tag{5.3}$$

$$f_k - \tilde{f}_k = \frac{X_f}{k^2} \|g_k\|^2, \quad g_k - \tilde{g}_k = \frac{X_g}{k} g_k, \quad \zeta_k = \frac{\epsilon_f}{k^2} \|g_k\|^2, \tag{5.4}$$

$$f_k - \tilde{f}_k = X_f, \qquad g_k - \tilde{g}_k = X_g, \qquad \zeta_k = \epsilon_f, \tag{5.5}$$

where $X_f, X_g$ are defined in (5.1). The parameters $\epsilon_f, \epsilon_g$ are separately set as $\epsilon_f = 10^{-5}$ and $\epsilon_g = 10^{-5}$. By generating errors and $\zeta_k$ in this way, we have $\zeta_k \geq \delta_k$.

Three examples are tested.

1) We set $n = 5$ as the dimension of the quadratic problem and the diagonal matrix is set as $D = \mathrm{diag}(0.001, 0.01, 0.1, 1, 10)$.

2) We set $n = 300$ to show the performance of our method with subspace techniques and the diagonal matrix is set as $D = \mathrm{diag}(0.01, 0.02, \ldots, 3)$.

3) To test the performance of these methods for large-scale problems, we set $n = 2000$ and the diagonal matrix is set as $D = \mathrm{diag}(1.001, 1.001^2, \ldots, 1.001^{2000})$.

We set the initial point $x_0 = (1, 1, \cdots, 1) \in \mathbb{R}^n$ and $max\_iter = 5000$ for all methods.

When errors and $\zeta_k$ satisfies (5.2) and (5.3), we set $tol = 10^{-4}$; when errors and $\zeta_k$ satisfies (5.4) and (5.5), we set $tol = 10^{-9}$ when $n = 5$ and $n = 300$, and set $tol = 10^{-6}$ when $n = 2000$. The results are presented in Table 5.1. In all tables, the terms "its", "time", "f" and "nrmG" denote the total number of iterations, the CPU time that the algorithms spent to reach the stopping criterions, the final objective value, and the final norm of the gradient, respectively.

From Table 5.1, we can see that the adaQNsub method performs best. When the dimension becomes large, the adaQN and BFGSe method require more running time. The reason for this is that the computation cost of updating the quasi-Newton matrix increases rapidly as $n$ increases.

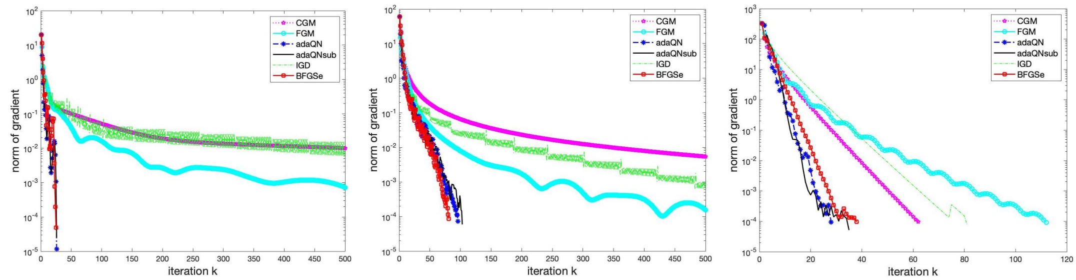
Table 5.1: Numerical results of quadratic function with different kinds of dimension numbers and errors.

| $n = 5$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Solver | Errors satisfying (5.2) | | | | Errors satisfying (5.3) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 5000 | 0.125 | 4.47E-05 | 1.22E-03 | 5000 | 0.147 | 4.47E-05 | 1.22E-03 |
| CGM | 5000 | 0.064 | 2.40E-04 | 9.79E-04 | 5000 | 0.051 | 2.40E-04 | 9.79E-04 |
| FGM | 1168 | 0.040 | 2.57E-07 | 9.80E-05 | 5000 | 0.053 | 2.49E-08 | 9.94E-06 |
| BFGSe | 25 | 0.015 | 8.74E-10 | 4.90E-05 | 35 | 0.011 | 2.41E-19 | 8.79E-10 |
| adaQN | 26 | 0.009 | 3.93E-08 | 1.15E-05 | 31 | 0.007 | 5.12E-21 | 2.63E-10 |
| adaQNsub | 26 | 0.008 | 3.76E-08 | 2.49E-05 | 50 | 0.007 | 2.35E-20 | 1.69E-10 |
| Solver | Errors satisfying (5.4) | | | | Errors satisfying (5.5) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 5000 | 0.160 | 4.47E-05 | 1.22E-03 | 5000 | 0.230 | 6.65E-05 | 4.07E-03 |
| CGM | 5000 | 0.064 | 2.40E-04 | 9.79E-04 | 5000 | 0.074 | 2.49E-04 | 9.83E-04 |
| FGM | 5000 | 0.064 | 2.49E-08 | 9.94E-06 | 1164 | 0.035 | -4.44E-06 | 9.95E-05 |
| BFGSe | 36 | 0.010 | 2.93E-20 | 6.43E-10 | 5001 | 0.557 | 6.96E-04 | 3.40E-03 |
| adaQN | 31 | 0.002 | 5.08E-21 | 2.62E-10 | 26 | 0.003 | -2.07E-06 | 3.25E-05 |
| adaQNsub | 50 | 0.003 | 2.32E-20 | 1.70E-10 | 26 | 0.003 | -1.87E-06 | 5.35E-05 |

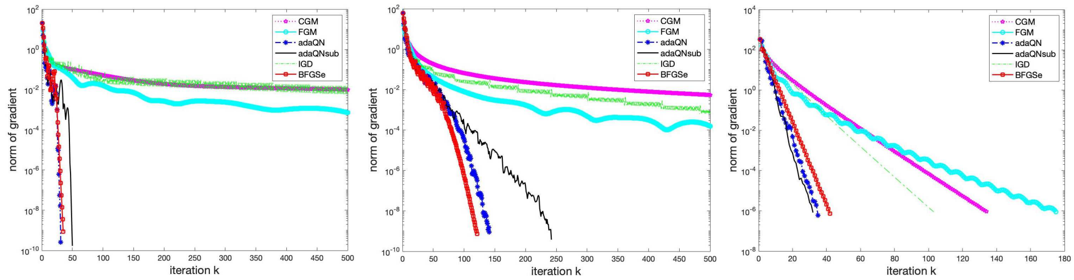
Table 5.1: Numerical results of quadratic function with different kinds of dimension numbers and errors (cont'd).

| $n = 300$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Solver | Errors satisfying (5.2) | | | | Errors satisfying (5.3) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 731 | 0.110 | 1.59E-07 | 9.61E-05 | 2273 | 0.849 | 1.01E-17 | 7.62E-10 |
| CGM | 1853 | 0.134 | 2.50E-07 | 9.99E-05 | 5000 | 0.778 | 3.75E-15 | 1.23E-08 |
| FGM | 510 | 0.058 | 2.01E-07 | 9.42E-05 | 2816 | 0.407 | 1.01E-17 | 7.39E-10 |
| BFGSe | 81 | 0.169 | 2.33E-09 | 8.65E-05 | 121 | 0.254 | 1.33E-19 | 7.28E-10 |
| adaQN | 96 | 0.171 | 1.65E-09 | 7.25E-05 | 141 | 0.226 | 1.49E-18 | 9.17E-10 |
| adaQNsub | 103 | 0.041 | 1.43E-08 | 5.92E-05 | 242 | 0.126 | 3.97E-19 | 3.80E-10 |
| Solver | Errors satisfying (5.4) | | | | Errors satisfying (5.5) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 2273 | 0.845 | 1.01E-17 | 7.62E-10 | 5000 | 0.492 | 6.37E-06 | 0.0047 |
| CGM | 5000 | 0.880 | 3.75E-15 | 1.23E-08 | 1858 | 0.129 | -4.43E-06 | 9.99E-05 |
| FGM | 2816 | 0.567 | 1.01E-17 | 7.39E-10 | 519 | 0.047 | -6.11E-06 | 6.19E-05 |
| BFGSe | 121 | 0.269 | 1.33E-19 | 7.28E-10 | 557 | 0.969 | 2.58E-06 | 9.92E-05 |
| adaQN | 149 | 0.263 | 9.77E-20 | 7.22E-10 | 120 | 0.187 | -8.75E-06 | 7.21E-05 |
| adaQNsub | 242 | 0.114 | 1.55E-18 | 6.26E-10 | 139 | 0.035 | 6.50E-06 | 9.85E-05 |
| $n = 2000$ | | | | | | | | |
| Solver | Errors satisfying (5.2) | | | | Errors satisfying (5.3) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 81 | 0.754 | 7.76E-10 | 8.15E-05 | 103 | 4.789 | 2.71E-14 | 8.94E-07 |
| CGM | 62 | 0.354 | -5.26E-11 | 9.72E-05 | 134 | 2.013 | 2.14E-13 | 9.38E-07 |
| FGM | 112 | 0.762 | 1.75E-09 | 9.18E-05 | 175 | 2.656 | 1.95E-13 | 9.02E-07 |
| BFGSe | 38 | 10.524 | 7.66E-09 | 9.63E-05 | 42 | 11.683 | 4.50E-14 | 7.35E-07 |
| adaQN | 28 | 3.894 | -3.77E-09 | 9.54E-05 | 35 | 5.946 | 1.99E-14 | 5.98E-07 |
| adaQNsub | 35 | 0.257 | -1.24E-09 | 5.18E-05 | 32 | 1.339 | 3.54E-14 | 8.26E-07 |
| Solver | Errors satisfying (5.4) | | | | Errors satisfying (5.5) | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 103 | 6.362 | 2.71E-14 | 8.94E-07 | 61 | 0.656 | 5.72E-08 | 5.49E-05 |
| CGM | 134 | 2.260 | 2.14E-13 | 9.38E-07 | 62 | 0.385 | 9.36E-07 | 9.70E-05 |
| FGM | 175 | 2.758 | 1.95E-13 | 9.02E-07 | 113 | 0.671 | 7.07E-07 | 8.77E-05 |
| BFGSe | 42 | 13.070 | 4.50E-14 | 7.35E-07 | 54 | 14.639 | -5.11E-07 | 8.82E-05 |
| adaQN | 35 | 6.219 | 1.98E-14 | 5.96E-07 | 30 | 4.225 | -2.49E-07 | 9.83E-05 |
| adaQNsub | 32 | 1.373 | 3.53E-14 | 8.25E-07 | 31 | 0.262 | 8.57E-07 | 5.83E-05 |

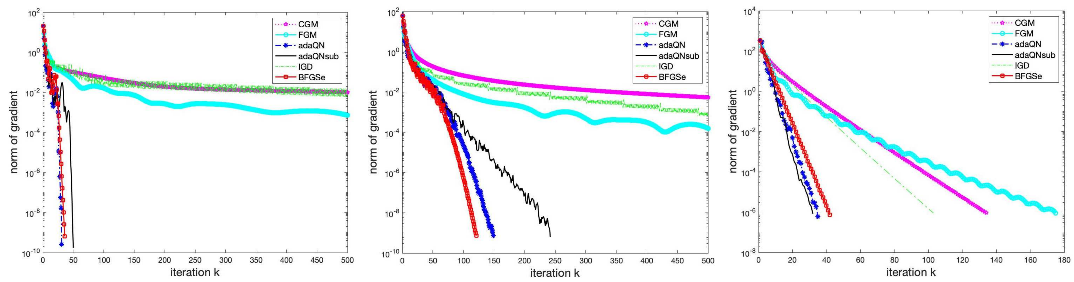
In Fig. 5.1, we plot the norm of gradients versus iteration numbers of all six methods with different dimensions and different errors. We can see that adaQN, adaQNsub and BFGSe usually converge linearly with a good rate. In some cases, we can observe the superlinear convergence rate of these three methods when errors are relatively small enough.



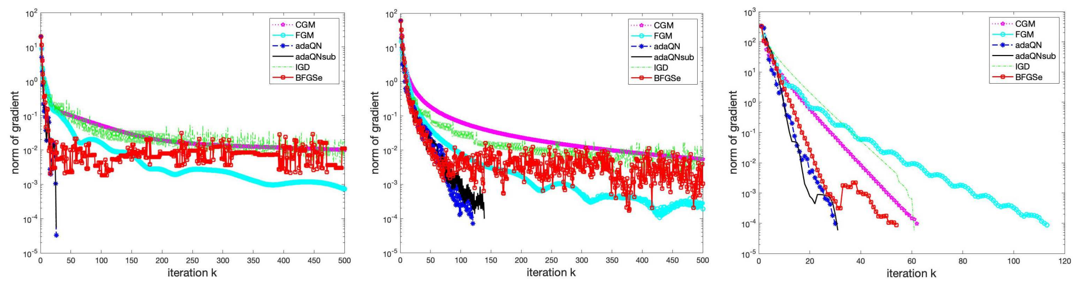

(a) $n = 5$, errors are (5.2) (b) $n = 300$, errors are (5.2) (c) $n = 2000$, errors are (5.2)




(d) $n = 5$, errors are (5.3) (e) $n = 300$, errors are (5.3) (f) $n = 2000$, errors are (5.3)




(g) $n = 5$, errors are (5.4) (h) $n = 300$, errors are (5.4) (i) $n = 2000$, errors are (5.4)




(j) $n = 5$, errors are (5.5) (k) $n = 300$, errors are (5.5) (l) $n = 2000$, errors are (5.5)

Fig. 5.1. Gradient norm of the proposed methods.

## 5.2. The case of unconstrained test problems

We select 18 unconstrained problems considered in [1], which are listed in Table 5.2. For these problems, we calculate the function values accurately and calculate the approximated gradients by finite differences

$$[\tilde{g}_k]_i = \frac{f(x + h_k e_i) - f(x)}{h_k}, \quad i = 1, \ldots, n,$$

where $h_k > 0$. When function $f$ is twice continuously differentiable and $\|\nabla^2 f(x)\|$ is bounded for all $x \in \mathbb{R}^n$, we have $\tilde{g}_k \to g_k$ as $h_k \to 0$. When $h_k$ is small enough, it holds that $\|\tilde{g}_k - g_k\| \sim O(h_k\|\nabla^2 f(x)\|)$. We set $\zeta_k = \max\{10^{-6}/k, 10^{-8}\}$ and $h_k = X_f$, where $X_f$ is defined in (5.1) and $\epsilon_f = 10^{-8}$. The initial point is given in [1]. We set $n = 150$, $tol = 10^{-6}$ and $max\_iter = 10000$ for all methods. The results are presented in Table 5.3. To test the performance of these

Table 5.2: Unconstrained optimization test functions.

| Problem | Problem | Problem |
|---|---|---|
| Extended Freudenstein and Roth | Extended Rosenbrock | BDEXP |
| Perturbed quadratic | Raydan 1 | NONDIA |
| Raydan 2 | ARWHEAD | NONSCOMP |
| DQDRTIC | EG2 | QUARTC |
| LIARWHD | POWER | COSINE |
| ENGVAL1 | EDENSCH | BDQRATIC |

Table 5.3: Numerical results of different kinds of unconstrained test functions when dimension is 150.

| Solver | Extended Freudenstein and Roth function | | | | Extended Rosenbrock function | | | |
|---|---|---|---|---|---|---|---|---|
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 2879 | 2.672 | 3673.82 | 0 | 10001 | 3.828 | 2.04E-10 | 4.60E-04 |
| CGM | 2849 | 2.600 | 3673.88 | 0 | 10001 | 3.609 | 8.13E-06 | 1.16E-01 |
| FGM | 2158 | 1.937 | 3673.82 | 0 | 10001 | 3.467 | 5.04E-07 | 6.63E-04 |
| BFGSe | 327 | 1.126 | 3673.82 | 0 | 8713 | 30.704 | 8.80E-11 | 8.11E-07 |
| adaQN | 380 | 0.166 | 3673.82 | 0 | 463 | 0.499 | 1.31E-12 | 3.51E-07 |
| adaQNsub | 171 | 0.574 | 3673.82 | 0 | 550 | 0.268 | 2.70E-11 | 7.52E-07 |
| Solver | Perturbed Quadratic function | | | | Raydan 1 function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 10001 | 2.477 | 1.23E-11 | 1.73E-04 | 2616 | 1.527 | 1132.50 | 0 |
| CGM | 3440 | 0.862 | 1.97E-13 | 9.01E-07 | 4160 | 2.340 | 1132.51 | 0 |
| FGM | 5052 | 1.420 | 9.81E-14 | 8.95E-07 | 986 | 0.556 | 1132.50 | 0 |
| BFGSe | 187 | 0.253 | 3.13E-16 | 8.89E-07 | 59 | 0.159 | 1132.50 | 0 |
| adaQN | 471 | 0.437 | 2.90E-14 | 6.71E-07 | 302 | 0.345 | 1132.50 | 0 |
| adaQNsub | 5699 | 1.589 | 6.72E-14 | 8.70E-07 | 110 | 0.088 | 1132.50 | 0 |
| Solver | Raydan 2 function | | | | ARWHEAD function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 855 | 0.374 | 150 | 0 | 402 | 0.147 | 2.31E-10 | 0 |
| CGM | 377 | 0.163 | 150 | 0 | 1345 | 0.424 | 3.09E-12 | 0 |
| FGM | 353 | 0.143 | 150 | 0 | 744 | 0.226 | 0 | 9.82E-07 |
| BFGSe | 29 | 0.166 | 150 | 0 | 46 | 0.109 | 4.93E-14 | 0 |
| adaQN | 15 | 0.013 | 150 | 0 | 19 | 0.022 | 1.97E-14 | 4.50E-07 |
| adaQNsub | 8 | 0.005 | 150 | 0 | 15 | 0.010 | 0 | 3.24E-07 |
| Solver | DQDRTIC function | | | | EG2 function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 8985 | 3.710 | 5.89E-13 | 7.19E-07 | 10001 | 7.055 | -148.92 | 0.02 |
| CGM | 10001 | 4.380 | 2.32E-11 | 1.11E-05 | 10001 | 6.893 | -148.60 | 0.17 |
| FGM | 9071 | 3.693 | 5.98E-14 | 4.78E-07 | 1298 | 0.901 | -148.93 | 0 |
| BFGSe | 61 | 0.240 | 3.67E-14 | 4.08E-07 | 1321 | 15.768 | -149.50 | 0 |
| adaQN | 77 | 0.094 | 2.95E-13 | 5.85E-07 | 48 | 0.093 | -149.00 | 0 |
| adaQNsub | 101 | 0.049 | 5.09E-14 | 9.82E-07 | 309 | 0.378 | -149.00 | 0 |

Table 5.3: Numerical results of different kinds of unconstrained test functions when dimension is 150 (cont'd).

| Solver | LIARWHD function | | | | POWER function | | | |
|---|---|---|---|---|---|---|---|---|
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 10001 | 1.443 | 3.85E-11 | 1.48E-04 | 10001 | 2.317 | 2.60E-11 | 2.46E-04 |
| CGM | 10001 | 1.396 | 4.49E-12 | 2.79E-06 | 10001 | 2.328 | 2.61E-12 | 5.08E-06 |
| FGM | 8317 | 1.146 | 3.50E-13 | 9.11E-07 | 10001 | 2.447 | 3.21E-12 | 4.65E-06 |
| BFGSe | 210 | 0.420 | 4.86E-13 | 6.95E-07 | 176 | 0.211 | 2.04E-13 | 3.04E-07 |
| adaQN | 54 | 0.040 | 3.49E-13 | 8.56E-07 | 471 | 0.371 | 5.95E-14 | 4.45E-07 |
| adaQNsub | 22 | 0.013 | 1.17E-12 | 3.89E-07 | 10001 | 1.570 | 1.11E-11 | 9.27E-05 |
| Solver | ENGVAL1 function | | | | EDENSCH function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 533 | 0.237 | 164.59 | 0 | 2070 | 7.637 | 298.26 | 0 |
| CGM | 653 | 0.276 | 164.59 | 0 | 2292 | 8.411 | 298.26 | 0 |
| FGM | 533 | 0.230 | 164.59 | 0 | 1376 | 5.139 | 298.26 | 0 |
| BFGSe | 165 | 0.311 | 164.59 | 0 | 273 | 4.012 | 298.26 | 0 |
| adaQN | 255 | 0.286 | 164.59 | 0 | 366 | 1.671 | 298.26 | 0 |
| adaQNsub | 124 | 0.061 | 164.59 | 0 | 40 | 0.169 | 298.26 | 0 |
| Solver | BDEXP function | | | | NONDIA function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 10001 | 11.069 | 2.33E-02 | 1.64E-02 | 10001 | 2.051 | 2.39E-03 | 4.47E-02 |
| CGM | 10001 | 11.013 | 3.73E-02 | 2.55E-02 | 10001 | 1.959 | 2.67E-03 | 1.63E-02 |
| FGM | 10001 | 11.240 | 8.76E-06 | 8.46E-06 | 10001 | 1.943 | 7.77E-08 | 2.10E-04 |
| BFGSe | 4 | 0.090 | 8.08E-19 | 6.28E-19 | 1271 | 2.128 | 7.25E-10 | 8.12E-07 |
| adaQN | 24 | 0.057 | 4.86E-07 | 5.09E-07 | 1589 | 2.458 | 9.25E-10 | 8.95E-07 |
| adaQNsub | 24 | 0.051 | 4.86E-07 | 5.09E-07 | 239 | 0.152 | 1.03E-09 | 9.68E-07 |
| Solver | NONSCOMP function | | | | QUARTC function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 10001 | 2.822 | 5.09 | 3.12E-03 | 10001 | 28.365 | 2.34E-04 | 2.16E-03 |
| CGM | 10001 | 2.707 | 7.59 | 8.02E-02 | 10001 | 28.249 | 5.25E-04 | 3.96E-03 |
| FGM | 10001 | 2.911 | 3.63E-06 | 1.27E-05 | 774 | 2.218 | 8.35E-09 | 9.98E-07 |
| BFGSe | 1118 | 4.744 | 7.07E-08 | 9.44E-07 | 2 | 0.021 | 2.27E-21 | 3.76E-16 |
| adaQN | 850 | 0.765 | 2.11E-07 | 9.44E-07 | 2 | 0.024 | 8.02E-26 | 1.88E-19 |
| adaQNsub | 10001 | 1.643 | 1.80E-06 | 8.91E-04 | 2 | 0.022 | 8.55E-25 | 1.05E-18 |
| Solver | COSINE function | | | | BDQRTIC function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 121 | 0.130 | -149 | 0 | 992 | 0.750 | 626.25 | 0 |
| CGM | 226 | 0.211 | -149 | 0 | 234 | 0.178 | 626.25 | 0 |
| FGM | 275 | 0.302 | -149 | 0 | 585 | 0.433 | 626.25 | 0 |
| BFGSe | 327 | 1.763 | -149 | 0 | 204 | 6.491 | 626.25 | 0 |
| adaQN | 389 | 0.683 | -149 | 0 | 603 | 0.799 | 626.25 | 0 |
| adaQNsub | 123 | 0.142 | -149 | 0 | 107 | 0.661 | 626.25 | 0 |

methods for large-scale problems, we choose 12 unconstrained problems from Table 5.2 and set $n = 1000$ for all methods. The results are presented in Table 5.4. Among all kinds of problems, we can see that the adaQN method performs best when $n$ is small, and the adaQNsub method performs best when $n$ is large. The FGM method performs better than CGM and IGD, which can be observed from Tables 5.3 and 5.4.

Table 5.4: Numerical results of different kinds of unconstrained test functions when dimension is 1000.

| Solver | Extended Freudenstein and Roth function | | | | Raydan 1 function | | | |
|---|---|---|---|---|---|---|---|---|
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 2227 | 24.264 | 24492.23 | 0 | 583 | 10.764 | 50050.02 | 0 |
| CGM | 1932 | 20.221 | 24499.66 | 0 | 497 | 9.013 | 50050.20 | 0 |
| FGM | 930 | 9.756 | 24492.72 | 0 | 335 | 6.492 | 50050.00 | 0 |
| BFGSe | 857 | 63.266 | 24492.13 | 0 | 223 | 22.116 | 50050.00 | 0 |
| adaQN | 396 | 15.488 | 24492.14 | 0 | 336 | 14.682 | 50050.00 | 0 |
| adaQNsub | 119 | 1.294 | 24492.22 | 0 | 152 | 2.882 | 50050.01 | 0 |
| Solver | Raydan 2 function | | | | ARWHEAD function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 833 | 11.317 | 1000.00 | 0 | 888 | 4.849 | 1.33E-10 | 0 |
| CGM | 1688 | 23.126 | 1000.00 | 0 | 3785 | 20.252 | 2.25E-11 | 0 |
| FGM | 324 | 4.369 | 1000.00 | 0 | 604 | 3.250 | 8.67E-09 | 0 |
| BFGSe | 18 | 2.893 | 1000.00 | 0 | 4431 | 274.041 | 1.76E-13 | 0 |
| adaQN | 8 | 0.226 | 1000.00 | 0 | 18 | 0.387 | 2.22E-13 | 0 |
| adaQNsub | 8 | 0.107 | 1000.00 | 0 | 19 | 0.111 | 0 | 0 |
| Solver | DQDRTIC function | | | | EG2 function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 3918 | 27.387 | 3.77E-12 | 5.99E-07 | 3378 | 66.839 | -998.28 | 0 |
| CGM | 7861 | 53.082 | 6.81E-13 | 9.92E-07 | 10001 | 198.785 | -998.28 | 0.11 |
| FGM | 10001 | 67.022 | 2.88E-11 | 5.24E-05 | 3126 | 61.713 | -998.93 | 0 |
| BFGSe | 2173 | 193.712 | 3.63E-14 | 7.47E-07 | 2162 | 137.327 | -999.00 | 0 |
| adaQN | 90 | 3.264 | 1.07E-13 | 4.96E-07 | 2474 | 126.345 | -998.94 | 0 |
| adaQNsub | 199 | 1.426 | 8.78E-13 | 9.04E-07 | 1848 | 37.508 | -998.95 | 0 |
| Solver | LIARWHD function | | | | POWER function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 10001 | 29.261 | 1.99E-04 | 3.53E-02 | 10001 | 45.842 | 1.21E-11 | 8.40E-04 |
| CGM | 10001 | 29.134 | 12.87 | 7.32 | 10001 | 44.031 | 5.05E-12 | 8.09E-05 |
| FGM | 10001 | 29.444 | 3.59E-05 | 1.20E-02 | 10001 | 42.873 | 6.15E-12 | 1.07E-04 |
| BFGSe | 10001 | 559.816 | 3.91E-04 | 5.35E-02 | 2352 | 139.066 | 1.14E-11 | 8.24E-08 |
| adaQN | 94 | 2.944 | 1.11E-12 | 6.47E-07 | 3571 | 89.343 | 1.80E-10 | 8.20E-08 |
| adaQNsub | 125 | 0.401 | 5.24E-14 | 7.10E-07 | 10001 | 25.043 | 3.65E-09 | 3.61E-05 |
| Solver | ENGVAL1 function | | | | EDENSCH function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 293 | 2.470 | 1108.19 | 0 | 5443 | 15.150 | 298.26 | 0 |
| CGM | 5386 | 38.998 | 1108.19 | 0 | 10001 | 35.427 | 298.27 | 3.99E-02 |
| FGM | 976 | 6.973 | 1108.19 | 0 | 4647 | 16.705 | 298.26 | 0 |
| BFGSe | 857 | 57.209 | 1108.19 | 0 | 1019 | 19.437 | 298.26 | 0 |
| adaQN | 1034 | 32.555 | 1108.19 | 0 | 374 | 1.568 | 298.26 | 0 |
| adaQNsub | 38 | 0.311 | 1108.19 | 0 | 155 | 0.554 | 298.26 | 0 |
| Solver | COSINE function | | | | QUARTC function | | | |
| | its | time | f | nrmG | its | time | f | nrmG |
| IGD | 138 | 3.352 | -999 | 0 | 10001 | 491.460 | 1.56E-03 | 5.58E-03 |
| CGM | 345 | 8.043 | -999 | 0 | 10001 | 502.839 | 9.70E-03 | 2.20E-02 |
| FGM | 127 | 3.224 | -999 | 0 | 2176 | 113.345 | 1.57E-08 | 1.00E-06 |
| BFGSe | 132 | 15.635 | -999 | 0 | 2 | 0.211 | 2.22E-20 | 1.30E-15 |
| adaQN | 453 | 21.709 | -999 | 0 | 2 | 0.084 | 2.70E-23 | 8.69E-18 |
| adaQNsub | 123 | 3.346 | -999 | 0 | 2 | 0.078 | 1.78E-24 | 1.03E-18 |

### 5.3. The case of numerical integration optimization

We use the optimization problem proposed in [12] as the test problem. Let $x$ and $t$ be vectors in $\mathbb{R}^n$. Define

$$h_1(t) := \exp\left(-\sum_{i=1}^{n}\frac{t_i}{i}\right), \quad h_2(t,x) := \cos\left(\sum_{i=1}^{n}\frac{t_i}{x_i}\right).$$

The objective function $f$ is given by

$$f(x) = \int_{t\in\Omega}\big(h_2(t,x) - h_1(t)\big)^2 dt, \tag{5.6}$$

where $\Omega = \{t : 0 \le t_i \le 1, i = 1, \dots n\}$. By (5.6), for $j = 1, \dots, n$, we have

$$\big(g(x)\big)_j = \left(\frac{\partial f}{\partial x}\right)_j = \int_{t\in\Omega} 2\big(h_2(t,x) - h_1(t)\big)\sin\left(\sum_{i=1}^{n}\frac{t_i}{x_i}\right)\frac{t_j}{x_j^2}dt. \tag{5.7}$$

Given $x \in \mathbb{R}^n$, we can calculate $f(x)$ and $g(x)$ separately by numerical integrations using the Simpsons rule (see [9]) with uniform mesh. In practice, $f(x)$ and $g(x)$ can be computed to any desired accuracy by successively decreasing the step size in Simpsons rule. Thus for each $k$, we can get an approximated pair $(\tilde{f}_k, \tilde{g}_k)$. For details of the computational procedure, readers are referred to [12].

For numerical comparison, we set $\eta_k = \omega\|\tilde{g}_k\|$, where $\omega > 0$ is independent of $k$. Then (2.1) and (2.2) can be rewritten as

$$|f_k - \tilde{f}_k| \le \delta_k, \tag{5.8}$$

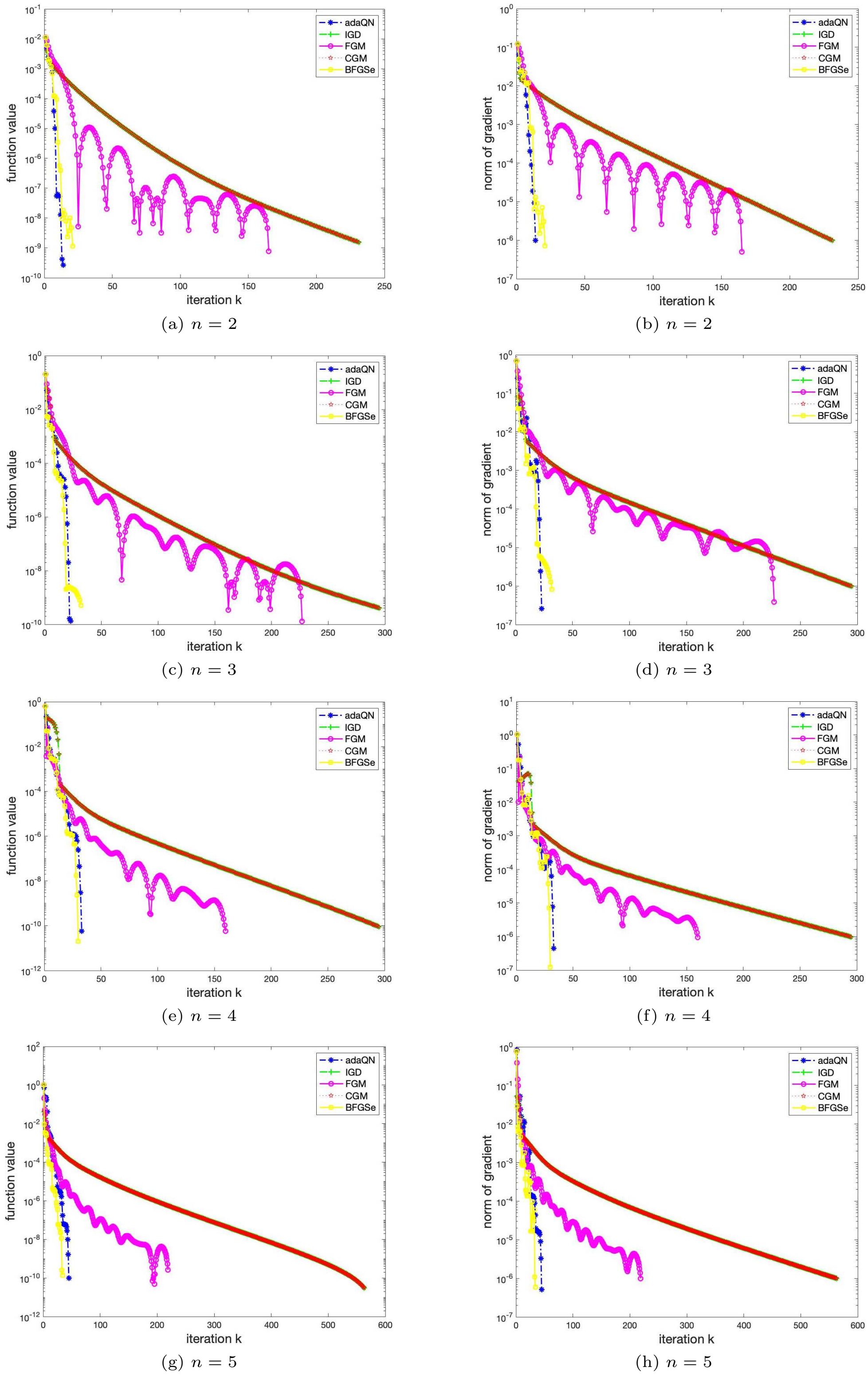
$$\|g_k - \tilde{g}_k\| \le \omega\|\tilde{g}_k\|, \quad \forall k. \tag{5.9}$$

At the $k$-th iteration, we compute a pair $(\tilde{f}_k, \tilde{g}_k)$ which satisfies (5.8) and (5.9), and use it as an approximation of $(f_k, g_k)$. In the implementation of our method, we set $\delta_k = \max\{10^{-7}, 10^{-4}/k^2\}$ and $\zeta_k = \delta_k$ for all $k \ge 1$.

Since we use the Simpsons rule to calculate the values in (5.6) and (5.7), the computational cost increases rapidly as the dimension $n$ increases. Thus, we only report the numerical results of all methods with $n$ varying from 2 to 5.

We do not take the adaQNsub method into consideration because the dimension of the numerical integration problem is too small. The initial point is $x_0 = (1, 1, \cdots, 1) \in \mathbb{R}^n$. We set $tol = 10^{-6}$ and $max\_iter = 5000$ for all methods. The results are presented in Table 5.5. We can see that the adaQN method performs best in all cases. From Table 5.5, we can see that the BFGSe method needs less iterations but takes more time to converge compared to first-order methods. The reason is that the Armijo-Wolfe line search in this method is time-consuming.

We further demonstrate the convergence behaviors of the five methods with different dimensions and different errors in Figs. 5.2 and 5.3. The function value is defined as $f(x_k) - f(x^*)$. From Figs. 5.2 and 5.3, we can see that BFGSe and adaQN methods require less iterations to converge compared to first-order methods.



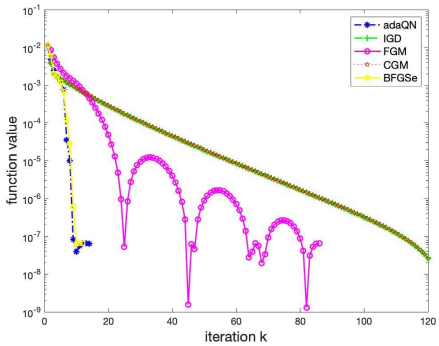

(a) $n = 2$ (b) $n = 2$

(c) $n = 3$ (d) $n = 3$

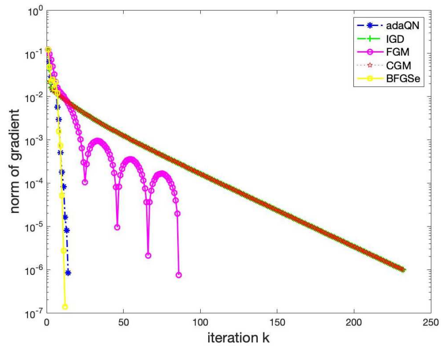
(e) $n = 4$ (f) $n = 4$

(g) $n = 5$ (h) $n = 5$

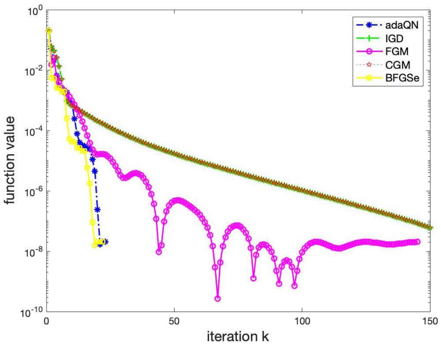
Fig. 5.2. Convergence results when $\omega = 10^{-3}$.



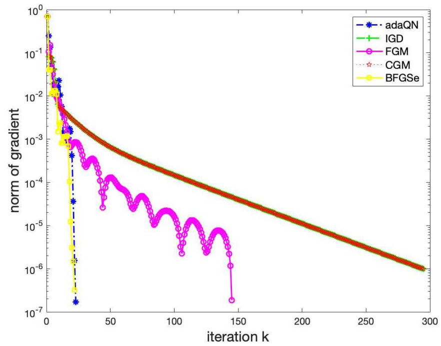
(a) $n = 2$



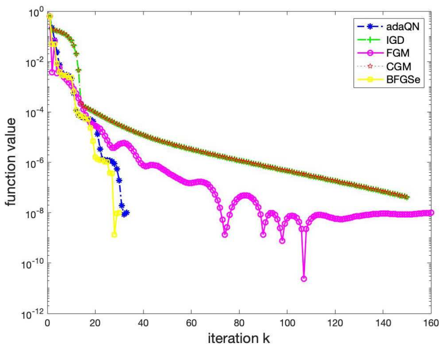
(b) $n = 2$



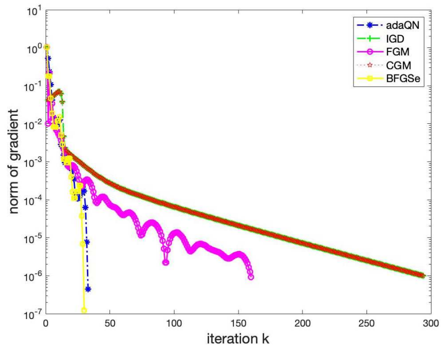
(c) $n = 3$



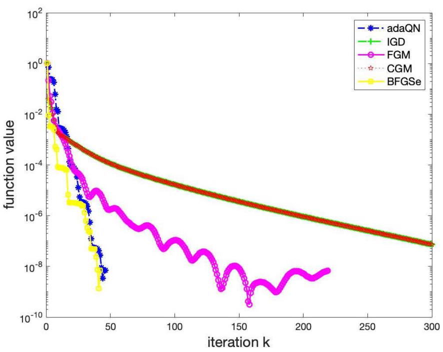
(d) $n = 3$



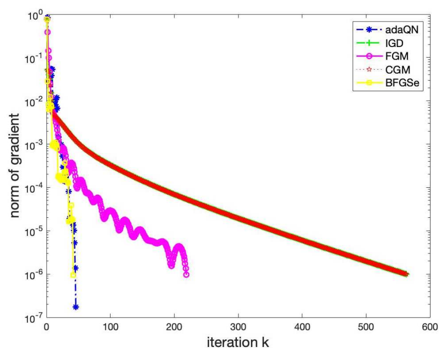
(e) $n = 4$



(f) $n = 4$



(g) $n = 5$



(h) $n = 5$

Fig. 5.3. Convergence results when $\omega = 10^{-4}$.

Table 5.5: Numerical results of numerical integration optimization.

| Solver | | $n=2$ | | $n=3$ | | $n=4$ | | $n=5$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | its | time | its | time | its | time | its | time |
| $\omega=10^{-3}$ | IGD | 231 | 0.0785 | 295 | 1.3354 | 295 | 15.8623 | 563 | 364.5628 |
| | CGM | 231 | 0.0410 | 295 | 0.7266 | 295 | 9.0680 | 563 | 194.4044 |
| | FGM | 165 | 0.0339 | 227 | 0.5431 | 160 | 4.7415 | 219 | 75.1692 |
| | BFGSe | 21 | 0.0697 | 32 | 2.0583 | 30 | 12.8350 | 34 | 93.1020 |
| | adaQN | 14 | 0.0193 | 23 | 0.0935 | 33 | 1.4617 | 45 | 15.8936 |
| $\omega=10^{-4}$ | IGD | 232 | 0.0726 | 384 | 2.2285 | 294 | 17.4680 | 563 | 467.1695 |
| | CGM | 232 | 0.0600 | 384 | 1.3092 | 294 | 10.3846 | 563 | 259.6112 |
| | FGM | 86 | 0.0211 | 265 | 0.8592 | 160 | 5.5432 | 219 | 109.5669 |
| | BFGSe | 12 | 0.0201 | 22 | 1.5277 | 30 | 14.7075 | 42 | 103.9503 |
| | adaQN | 14 | 0.0177 | 23 | 0.1303 | 33 | 1.3158 | 46 | 21.1601 |

## 6. Conclusions

In this paper, we propose an adaptive regularized quasi-Newton method for solving unconstrained problems under the condition that function and gradient evaluations are inexact. Our method uses a trust-region-like framework to monitor the acceptance of trial steps. The advantage of this strategy is that we can save the computational cost of the line search. Under some mild conditions, we prove the global convergence of our method and establish the convergence rate of our method. Numerical experiments demonstrate the efficiency of the method. The numerical comparisons illustrate that our proposed method is promising. The regularized quasi-Newton method is a suitable method which can handle inexact first-order information of the problem.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (Grant No. NSFC-11971118).

## References

[1] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.*, **10** (2008), 147–161.

[2] S. Bellavia, G. Gurioli, B. Morini, and P. Toint, Trust-region algorithms: Probabilistic complexity and intrinsic noise with applications to subsampling techniques, *EURO J. Comput. Optim.*, **10** (2022), 100043.

[3] A.S. Berahas, R.H. Byrd, and J. Nocedal, Derivative-free optimization of noisy functions via quasi-Newton methods, *SIAM J. Optim.*, **29** (2019), 965–993.

[4] A.S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, Linear interpolation gives better gradients than Gaussian smoothing in derivative-free optimization, *arXiv:1905.13043*, 2019.

[5] A.S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, A theoretical and empirical comparison of gradient approximations in derivative-free optimization, *Found. Comput. Math.*, **22** (2022), 507–560.

[6] A.S. Berahas, L. Cao, and K. Scheinberg, Global convergence rate analysis of a generic line search algorithm with noise, *SIAM J. Optim.*, **31** (2021), 1489–1518.

[7] L. Bogolubsky, P. Dvurechenskii, A. Gasnikov, G. Gusev, Y. Nesterov, A.M. Raigorodskii, A. Tikhonov, and M. Zhukovskii, Learning supervised pagerank with gradientbased and gradient-free optimization methods, *Adv. Neural Inf. Process. Syst.*, (2016), 4914–4922.
[8] R. Bollapragada, R. Byrd, and J. Nocedal, Adaptive sampling strategies for stochastic optimization, *SIAM J. Optim.*, **28** (2018), 3312–3343.
[9] R.L. Burden, J.D. Faires, and A.M. Burden, *Numerical Analysis*, Cengage learning, 2015.
[10] R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, *Acta Numer.*, **7** (1998), 1–49.
[11] R.G. Carter, On the global convergence of trust region algorithms using inexact gradient information, *SIAM J. Numer. Anal.*, **28** (1991), 251–265.
[12] R.G. Carter, Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information, *SIAM J. Sci. Comput.*, **14** (1993), 368–388.
[13] C. Cartis, N.I. Gould, and P.L. Toint, Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results, *Math. Program.*, **127** (2011), 245–295.
[14] R. Chen, M. Menickelly, and K. Scheinberg, Stochastic optimization using a trust-region method and random models, *Math. Program.*, **169** (2018), 447–487.
[15] A.R. Conn, K. Scheinberg, and L.N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, 2009.
[16] F.E. Curtis and K. Scheinberg, Adaptive stochastic optimization: A framework for analyzing stochastic optimization algorithms, *IEEE Signal Process. Mag.*, **37** (2020), 32–42.
[17] A. d'Aspremont, Smooth optimization with approximate gradient, *SIAM J. Optim.*, **19** (2008), 1171–1183.
[18] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal, Convex proximal bundle methods in depth: A unified analysis for inexact oracles, *Math. Program.*, **148** (2014), 241–277.
[19] J.E. Dennis and J.J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, *Math. Comp.*, **28** (1974), 549–560.
[20] O. Devolder, *Exactness, Inexactness and Stochasticity in First-Order Methods for Large-Scale Convex Optimization*, PhD Thesis, Université catholique de Louvain, Brussels, 2013.
[21] O. Devolder, F. Glineur, and Y. Nesterov, First-order methods of smooth convex optimization with inexact oracle, *Math. Program.*, **146** (2014), 37–75.
[22] P. Dvurechensky and A. Gasnikov, Stochastic intermediate gradient method for convex problems with stochastic inexact oracle, *J. Optim. Theory Appl.*, **171** (2016), 121–145.
[23] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright, Computing forward-difference intervals for numerical optimization, *SIAM J. Sci. Statist. Comput.*, **4** (1983), 310–321.
[24] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, SIAM, 2019.
[25] J. Hu, A. Milzarek, Z. Wen, and Y. Yuan, Adaptive quadratically regularized Newton method for Riemannian optimization, *SIAM J. Matrix Anal. Appl.*, **39** (2018), 1181–1207.
[26] C.T. Kelley, *Implicit Filtering*, SIAM, 2011.
[27] C. Kelley and E. Sachs, Truncated Newton methods for optimization with inaccurate functions and gradients, *J. Optim. Theory Appl.*, **116** (2003), 83–98.
[28] J. Larson, M. Menickelly, and S.M. Wild, Derivative-free optimization methods, *Acta Numer.*, **28** (2019), 287–404.
[29] J.J. Moré and S.M. Wild, Benchmarking derivative-free optimization algorithms, *SIAM J. Optim.*, **20** (2009), 172–191.
[30] J.J. Moré and S.M. Wild, Estimating computational noise, *SIAM J. Sci. Comput.*, **33** (2011), 1292–1314.
[31] J.J. Moré and S.M. Wild, Estimating derivatives of noisy simulations, *ACM Trans. Math. Softw.*, **38** (2012), 1–21.
[32] I. Necoara, A. Patrascu, and F. Glineur, Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming, *Optim. Methods Softw.*, **34** (2019), 305–335.

[33] Y. Nesterov and V. Spokoiny, Random gradient-free minimization of convex functions, *Found. Comput. Math.*, **17** (2017), 527–566.

[34] B. T. Polyak, Gradient methods for the minimisation of functionals, *USSR Computational Mathematics and Mathematical Physics*, **3** (1963), 864–878.

[35] J. Rasch and A. Chambolle, Inexact first-order primal-dual algorithms, *Comput. Optim. Appl.*, **76** (2020), 381–430.

[36] T. Sun, I. Necoara, and Q. Tran-Dinh, Composite convex optimization with global and local inexact oracles, *Comput. Optim. Appl.*, **76** (2020), 69–124.

[37] Z. Wang and Y. Yuan, A subspace implementation of quasi-Newton trust region methods for unconstrained optimization, *Numer. Math.*, **104** (2006), 241–269.

[38] Z. Wen, A. Milzarek, M. Ulbrich, and H. Zhang, Adaptive regularized self-consistent field iteration with exact Hessian for electronic structure calculation, *SIAM J. Sci. Comput.*, **35** (2013), A1299–A1324.

[39] X. Wu, Z. Wen, and W. Bao, A regularized Newton method for computing ground states of Bose-Einstein condensates, *J. Sci. Comput.*, **73** (2017), 303–329.

[40] Y. Xie, R.H. Byrd, and J. Nocedal, Analysis of the BFGS method with errors, *SIAM J. Optim.*, **30** (2020), 182-209.

[41] Y. Yuan, A review on subspace methods for nonlinear optimization, in: *Proceedings of the International Congress of Mathematics*, (2014), 807–827.