Newton-CG methods for nonconvex unconstrained optimization with Hölder continuous Hessian

Chuan He* Heng Huang† Zhaosong Lu‡

November 13, 2023 (Revised: December 10, 2024; April 13, 2025)

Abstract

In this paper we consider a nonconvex unconstrained optimization problem minimizing a twice differentiable objective function with Hölder continuous Hessian. Specifically, we first propose a Newton-conjugate gradient (Newton-CG) method for finding an approximate first- and second-order stationary point of this problem, assuming the associated the Hölder parameters are explicitly known. Then we develop a parameter-free Newton-CG method without requiring any prior knowledge of these parameters. To the best of our knowledge, this method is the first parameter-free second-order method achieving the best-known iteration and operation complexity for finding an approximate first- and second-order stationary point of this problem. Finally, we present preliminary numerical results to demonstrate the superior practical performance of our parameter-free Newton-CG method over a well-known regularized Newton method.

Keywords Nonconvex unconstrained optimization, Newton-conjugate gradient method, Hölder continuity, iteration complexity, operation complexity

Mathematics Subject Classification 49M15, 49M37, 58C15, 90C25, 90C30

1 Introduction

In this paper we consider the nonconvex unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is twice continuously differentiable and $\nabla^2 f$ is Hölder continuous in an open neighborhood of a level set of f (see Assumption 1 for details). Our goal is to propose *easily implementable* second-order methods with complexity guarantees, particularly, Newton-conjugate gradient (Newton-CG) methods for finding approximate first- and second-order stationary points of problem (1).

In recent years, there have been significant advancements in second-order methods with complexity guarantees for problem (1) when $\nabla^2 f$ is *Lipschitz continuous*. Notably, cubic regularized Newton methods [1, 6, 10, 30], trust-region methods [16, 17, 27], second-order line-search method [32], inexact regularized Newton method [18], quadratic regularization method [4], and Newton-CG method [31] were developed for finding an $(\epsilon,\sqrt{\epsilon})$-second-order stationary point (SOSP) x of problem (1) satisfying

$$\|\nabla f(x)\|\le\epsilon, \qquad \lambda_{\min}(\nabla^2 f(x))\ge-\sqrt{\epsilon},$$

where $\epsilon\in(0,1)$ is a tolerance parameter and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the associated matrix. Under suitable assumptions, it was shown that these second-order methods achieve an iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for finding an $(\epsilon,\sqrt{\epsilon})$-SOSP, which has been proved to be optimal in [9, 13]. In addition to iteration

*Department of Mathematics, Linköping University, Sweden (email: chuan.he@liu.se).
†Department of Computer Science, University of Maryland, USA (email: heng@umd.edu).
‡Department of Industrial and Systems Engineering, University of Minnesota, USA (email: zhaosong@umn.edu).

complexity, operation complexity of the methods in [1, 6, 16, 31, 32], measured by the number of their fundamental operations, was also studied. Under suitable assumptions, it was shown that these methods achieve an operation complexity of $\widetilde{\mathcal{O}}(\epsilon^{-7/4})$ for finding an $(\epsilon, \sqrt{\epsilon})$-SOSP of problem (1) with high probability.[1] Similar operation complexity bounds have also been achieved by gradient-based methods (e.g., see [2, 7, 8, 23, 26, 28, 33]).

Nonetheless, there has been limited study on second-order methods for problem (1) – a nonconvex unconstrained optimization problem with Hölder continuous Hessian. The regularized Newton methods proposed in [11, 14, 15, 20, 34] appear to be the only existing second-order methods for problem (1). Specifically, the cubic regularized Newton method in [20] tackles problem (1) by solving a sequence of cubic regularized Newton subproblems. It is a parameter-free second-order method and does not require any prior information on the modulus H_ν and exponent ν associated with the Hölder continuity (see Assumption 1). Under mild assumptions, it was shown in [20] that this method enjoys an iteration complexity of

$$\mathcal{O}(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)}) \tag{2}$$

for finding an ϵ_g-first-order stationary point (FOSP) x of problem (1) satisfying $\|\nabla f(x)\| \le \epsilon_g$. This iteration complexity matches the lower iteration complexity bound established in [12, 13]. In another early work [11], a regularized Newton method, which solves a sequence of $(2+\nu)$th-order regularized Newton subproblems, has been proposed for solving problem (1). Iteration complexity of this method for finding an ϵ_g-FOSP has been established, with the order dependence on ϵ_g matching the optimal one in (2). This method has been generalized in [14, 15] to regularized high-order methods for solving nonconvex problems with Hölder continuous high-order derivatives. It shall be noted that these methods [11, 14, 15, 20] require solving regularized Newton or high-order polynomial optimization problems *exactly* per iteration, which may be highly expensive to implement in general. Recently, in [34], two adaptive regularized Newton methods were proposed for finding an approximate SOSP of the problem minimizing a nonconvex function with Hölder continuous Hessian on a Riemannian manifold, which includes problem (1) as a special case. Specifically, when applied to problem (1), one method in [34] inexactly solves a sequence of $(2+\nu)$th-order regularized Newton subproblems, while another method in [34] inexactly solves a sequence of trust-region subproblems. It has been shown in [34] that their methods exhibit an iteration complexity of $\mathcal{O}(\epsilon_g^{-(2+\nu)/(1+\nu)})$ for finding an ϵ_g-FOSP of problem (1), which achieves the optimal order of dependence on ϵ_g as given in (2). However, these methods in [34] are *not fully parameter-free* since prior knowledge of the Hölder exponent is required in order to achieve the best-known complexity.

As discussed above, the existing second-order methods [11, 14, 15, 20, 34] for problem (1) require solving a sequence of sophisticated trust-region or regularized Newton subproblems. In this paper, we propose *easily implementable* second-order methods, particularly Newton-CG methods for (1), by applying the capped CG method [31, Algorithm 1] to solve a sequence of systems of linear equations with coefficient matrix resulting from a proper perturbation on the Hessian of f. Specifically, we first propose a Newton-CG method (Algorithm 1) to find an approximate FOSP and SOSP of (1), assuming the parameters associated with the Hölder continuity of $\nabla^2 f$ are explicitly known. Then we develop a *parameter-free* Newton-CG method (Algorithm 2) for finding an approximate FOSP and SOSP of (1) without requiring any prior knowledge of these parameters. We show that these methods achieve the best-known iteration and operation complexity for finding an approximate FOSP and/or SOSP of (1). Moreover, when $\nabla^2 f$ is Lipschitz continuous, our proposed methods achieve an improved iteration and operation complexity over the Newton-CG methods [21, 31] in terms of the dependence on the Lipschitz constant of $\nabla^2 f$. In addition, preliminary numerical results are presented, demonstrating the practical advantages of our parameter-free Newton-CG method over the cubic regularized Newton method [20].

The main contributions of this paper are summarized as follows.

- We propose a Newton-CG method (Algorithm 1) to find an approximate FOSP and SOSP of (1), assuming that the parameters associated with the Hölder continuity of $\nabla^2 f$ are explicitly known. In contrast with the regularized Newton methods [11, 20, 34], our method is *easily implementable* and solves much *simpler subproblems* by a capped CG method, while achieving the best-known iteration and operation complexity.
- We propose a *parameter-free* Newton-CG method (Algorithm 2) for finding an approximate FOSP and SOSP of (1) without requiring any prior knowledge of these parameters. To the best of our knowledge, this

[1] $\widetilde{\mathcal{O}}(\cdot)$ represents $\mathcal{O}(\cdot)$ with logarithmic terms omitted.

is *the first fully parameter-free method* for finding an approximate FOSP and SOSP of (1), while achieving the best-known iteration and operation complexity.

The rest of this paper is organized as follows. In Section 2, we introduce some notation and assumptions that will be used in the paper. In Section 3, we propose a Newton-CG method for problem (1) and study its complexity. In Section 4, we propose a parameter-free Newton-CG method for problem (1) and study its complexity. Section 5 presents preliminary numerical results. In Section 6, we present the proofs of the main results.

2 Notation and assumptions

Throughout this paper, we let $\mathbb{R}^n$ denote the n-dimensional Euclidean space. We use $\|\cdot\|$ to denote the Euclidean norm of a vector or the spectral norm of a matrix. For any $s \in \mathbb{R}$, we let s_+ and $\lceil s \rceil$ denote the nonnegative part of s and the least integer no less than s, respectively, and we let $\mathrm{sgn}(s)$ be 1 if $s \geq 0$ and -1 otherwise. For a real symmetric matrix H, we use $\lambda_{\min}(H)$ to denote its minimum eigenvalue. In addition, $\widetilde{\mathcal{O}}(\cdot)$ represents $\mathcal{O}(\cdot)$ with logarithmic terms omitted.

We make the following assumptions on problem (1) throughout this paper.

Assumption 1. (a) *The level set $\mathscr{L}_f(x^0) := \{x : f(x) \leq f(x^0)\}$ is compact for some $x^0 \in \mathbb{R}^n$.*

(b) *The function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, and $\nabla^2 f$ is Hölder continuous in a bounded convex open neighborhood, denoted by Ω, of $\mathscr{L}_f(x^0)$, i.e., there exist $\nu \in [0, 1]$ and a finite $H_\nu > 0$ such that*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq H_\nu \|x - y\|^\nu, \quad \forall x, y \in \Omega. \tag{3}$$

It follows from Assumption 1(a) that there exist $f_{\mathrm{low}} \in \mathbb{R}$, $U_g > 0$ and $U_H > 0$ such that

$$f(x) \geq f_{\mathrm{low}}, \quad \|\nabla f(x)\| \leq U_g, \quad \|\nabla^2 f(x)\| \leq U_H, \quad \forall x \in \mathscr{L}_f(x^0). \tag{4}$$

We now make some remarks on Assumption 1(b).

Remark 1. (i) *When $\nu = 1$, the condition (3) corresponds to the standard Lipschitz continuity of $\nabla^2 f$. When $\nu = 0$, the condition (3) means that the variation of $\nabla^2 f$ on Ω is bounded, which is equivalent to the boundedness of $\nabla^2 f$ on Ω.*

(ii) *As a consequence of Assumption 1(b), the following two inequalities hold for all $x, y \in \Omega$ (e.g., see equations (2.7) and (2.8) in [20]):*

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x)\| \leq \frac{H_\nu \|y - x\|^{1+\nu}}{1 + \nu}, \tag{5}$$

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x) + \frac{H_\nu \|y - x\|^{2+\nu}}{(1 + \nu)(2 + \nu)}. \tag{6}$$

3 A Newton-CG method for problem (1)

In this section, we propose a Newton-CG method in Algorithm 1 for finding an ϵ_g-FOSP and (ϵ_g, ϵ_H)-SOSP of problem (1), assuming the parameters H_ν and ν associated with the Hölder continuity of $\nabla^2 f$ in (3) are explicitly known, and then analyze its complexity results.

Our Newton-CG method uses two important subroutines, a capped CG method and a minimum eigenvalue oracle. Specifically, the capped CG method is a modified CG method proposed in [31, Algorithm 1] for solving a possibly indefinite linear system

$$(H + 2\varepsilon I)d = -g, \tag{7}$$

where $0 \neq g \in \mathbb{R}^n$, $\varepsilon > 0$, and $H \in \mathbb{R}^{n \times n}$ is a symmetric matrix. It terminates within a finite number of iterations and returns either an approximate solution d to (7) satisfying $\|(H + 2\varepsilon I)d + g\| \leq \hat{\zeta}\|g\|$ and $d^T H d \geq -\varepsilon \|d\|^2$

for some $\hat{\zeta} \in (0,1)$ or a sufficiently negative curvature direction d of H with $d^T H d < -\varepsilon\|d\|^2$. In addition, the minimum eigenvalue oracle was proposed in [31, Procedure 2] to check whether a sufficiently negative curvature direction exists for a symmetric matrix H. It either produces a sufficiently negative curvature direction v of H satisfying $\|v\| = 1$ and $v^T H v \le -\varepsilon/2$ or certifies that $\lambda_{\min}(H) \ge -\varepsilon$ holds with high probability. For ease of reference, we present the capped CG method and the minimum eigenvalue oracle in Algorithms 3 and 4 in Appendices A and B, respectively.

We are now ready to introduce our Newton-CG method (Algorithm 1) for solving problem (1). This algorithm has two options: (i) when the tolerance $\epsilon_H \in (0,1)$ for second-order stationarity is not provided, this algorithm can find an ϵ_g-FOSP x of (1) that satisfies $\|\nabla f(x)\| \le \epsilon_g$ for some $\epsilon_g \in (0,1)$; (ii) when such an ϵ_H is provided, this algorithm can find a stochastic (ϵ_g, ϵ_H)-SOSP x of (1), satisfying $\|\nabla f(x)\| \le \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x)) \ge -\epsilon_H$ with probability at least $1-\delta$.

Specifically, at each iteration k of Algorithm 1, if the current iterate x^k does not satisfy $\|\nabla f(x^k)\| \le \epsilon_g$, the capped CG method (Algorithm 3) is invoked to find either an inexact Newton direction or a negative curvature direction by solving the following damped Newton system:

$$\big(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I\big) d = -\nabla f(x^k), \tag{8}$$

where $\gamma_\nu(\epsilon_g)$ is an inexact Lipschitz constant[2] of $\nabla^2 f$ defined as

$$\gamma_\nu(\epsilon_g) := 4 H_\nu^{2/(1+\nu)} \epsilon_g^{-(1-\nu)/(1+\nu)}. \tag{9}$$

The next iterate x^{k+1} is generated by performing a line search along the descent direction obtained from solving (8). Otherwise, if $\|\nabla f(x^k)\| \le \epsilon_g$, this algorithm has two options. First, when the tolerance $\epsilon_H \in (0,1)$ for second-order stationarity is not provided, Algorithm 1 terminates with x^k as an ϵ_g-FOSP. Second, when such an ϵ_H is provided, a minimum eigenvalue oracle (Algorithm 4) is further invoked to either obtain a sufficiently negative curvature direction and generate the next iterate x^{k+1} via a line search, or certify that x^k is an (ϵ_g, ϵ_H)-SOSP with high probability and terminates the algorithm. The details of this algorithm are presented in Algorithm 1.

The following theorem states the iteration and operation complexity of Algorithm 1 for finding an ϵ_g-FOSP, whose proof is relegated to Section 6.1.

Theorem 1. *Suppose that Assumption 1 holds with some $H_\nu > 0$ and $\nu \in [0,1]$, and ϵ_H is not provided for Algorithm 1. Let $\epsilon_g \in (0,1)$ be given, f_{low} and U_H be given in (4), $\gamma_\nu(\epsilon_g)$ be given (9), ζ, η, and θ be given in Algorithm 1, and*

$$c_{\text{sol}} := \eta \min\left\{ \left(\frac{2}{4+\zeta+\sqrt{(4+\zeta)^2+1}}\right)^2, \frac{1}{6}\left(\frac{2(1-\eta)\theta}{3}\right)^2 \right\}, \qquad c_{\text{nc}} := \frac{\eta\theta^2}{4}, \tag{14}$$

$$K_1 := \left\lceil \frac{f(x^0) - f_{\text{low}}}{\min\{c_{\text{sol}}, c_{\text{nc}}\}} \gamma_\nu(\epsilon_g)^{1/2} \epsilon_g^{-3/2} \right\rceil + 1. \tag{15}$$

Then the following statements hold.

(i) **(iteration complexity)** *Algorithm 1 terminates in at most K_1 iterations with*

$$K_1 = \mathcal{O}\left(H_\nu^{1/(1+\nu)} \epsilon_g^{-(2+\nu)/(1+\nu)}\right). \tag{16}$$

Moreover, its output x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ for some $0 \le k \le K_1$.

(ii) **(operation complexity)** *The main operations of Algorithm 1 consist of*

$$\widetilde{\mathcal{O}}\big(H_\nu^{1/(1+\nu)} \epsilon_g^{-(2+\nu)/(1+\nu)} \min\{n, U_H^{1/2}/(H_\nu \epsilon_g^\nu)^{1/(2+2\nu)}\}\big)$$

gradient evaluations and Hessian-vector products of f.

[2] In the literature (e.g., [19, 22, 29]), inexact Lipschitz constant of ∇f has been used to design and analyze first-order methods for problem (1), where f has a Hölder continuous gradient.

Algorithm 1 A Newton-CG method for problem (1)

input: tolerance $\epsilon_g \in (0,1)$, starting point x^0, CG-accuracy parameter $\zeta \in (0,1)$, backtracking ratio $\theta \in (0,1)$, line-search parameter $\eta \in (0,1)$, probability parameter $\delta \in (0,1)$, $\gamma_\nu(\epsilon_g)$ given in (9); **optional input**: tolerance $\epsilon_H \in (0,1)$;
for $k = 0,1,2,\ldots$ **do**

▷ *This part aims to improve first-order stationarity by calling Algorithm 3.*

if $\|\nabla f(x^k)\| > \epsilon_g$ **then**
Call Algorithm 3 (Appendix A) with $H = \nabla^2 f(x^k)$, $\varepsilon = (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}$, $g = \nabla f(x^k)$, accuracy parameter ζ, and $U = 0$ to obtain outputs d, d_type;
if d_type=NC **then**
Set

$$d^k \leftarrow -\operatorname{sgn}(d^T \nabla f(x^k))\max\{1, 1/\gamma_\nu(\epsilon_g)\}\frac{|d^T\nabla^2 f(x^k)d|}{\|d\|^3}d;$$

else {d_type=SOL}
Set

$$d^k \leftarrow d;$$

end if
Go to **Line Search**;
else if $\|\nabla f(x^k)\| \le \epsilon_g$ and ϵ_H is not provided **then**
Output x^k and terminate;
end if

▷ *This part aims to improve second-order stationarity by calling Algorithm 4.*

if $\|\nabla f(x^k)\| \le \epsilon_g$ and ϵ_H is provided **then**
Call Algorithm 4 (Appendix B) with $H = \nabla^2 f(x^k)$, $\varepsilon = \epsilon_H$, and probability parameter δ;
if Algorithm 4 certifies that $\lambda_{\min}(\nabla^2 f(x^k)) \ge -\epsilon_H$ **then**
Output x^k and terminate;
else {Sufficiently negative curvature direction v returned by Algorithm 4}
Set d_type=MEO and

$$d^k \leftarrow -\operatorname{sgn}(v^T\nabla f(x^k))|v^T\nabla^2 f(x^k)v|v; \tag{10}$$

Go to **Line Search**;
end if
end if

▷ *This part provides line search procedures.*

Line Search:
if d_type=SOL **then**
if $f(x^k + d^k) \le f(x^k)$ and $\|\nabla f(x^k + d^k)\| \le \epsilon_g$ **then** set $\alpha_k = 1$;
else Find $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer j such that

$$f(x^k + \theta^j d^k) \le f(x^k) - \eta(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\theta^{2j}\|d^k\|^2; \tag{11}$$

end if
else if d_type=NC **then**
Find $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer j such that

$$f(x^k + \theta^j d^k) \le f(x^k) - \eta\min\{1, \gamma_\nu(\epsilon_g)\}\theta^{2j}\|d^k\|^3/4; \tag{12}$$

else if d_type=MEO **then**
Find $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer j such that

$$f(x^k + \theta^j d^k) \le f(x^k) - \eta\theta^{2j}\|d^k\|^3/2; \tag{13}$$

end if

▷ *This part updates the next iterate.*

Set $x^{k+1} = x^k + \alpha_k d^k$;
end for

Remark 2. (i) *The iteration complexity presented in Theorem 1(i) matches the lower iteration complexity bound stated in* (2) *(see also [12, 13]) for finding an ϵ_g-FOSP of* (1) *using a second-order method. Moreover, the operation complexity stated in Theorem 1(ii) is a novel contribution to the literature. While some operation complexity results have been established in [34, Corollaries 4 and 5] for adaptive regularized*

Newton methods, those results only guarantee x satisfying $\|\nabla f(x)\| \le \epsilon_g$ with high probability. In contrast, the operation complexity in Theorem 1(ii) is achieved by Algorithm 1 for deterministically finding an ϵ_g-FOSP.

(ii) *When $\nu = 1$, the iteration and operation complexity of Algorithm 1 for finding an ϵ_g-FOSP of (1) are given by*

$$\mathcal{O}\left(L_H^{1/2}\epsilon_g^{-3/2}\right) \quad \text{and} \quad \widetilde{\mathcal{O}}\big(L_H^{1/2}\epsilon_g^{-3/2}\min\left\{n, U_H^{1/2}/(L_H\epsilon_g)^{1/4}\right\}\big),$$

respectively, where L_H is the Lipschitz constant of $\nabla^2 f$. These results demonstrate improved dependence on L_H compared to the following iteration and operation complexity, respectively, achieved by the Newton-CG methods in [21, 31] for finding an ϵ_g-FOSP of (1):

$$\mathcal{O}\left(L_H^2\epsilon_g^{-3/2}\right) \quad \text{and} \quad \widetilde{\mathcal{O}}\big(L_H^2\epsilon_g^{-3/2}\min\left\{n, U_H^{1/2}/\epsilon_g^{1/4}\right\}\big).$$

The next theorem establishes iteration and operation complexity of Algorithm 1 for finding a stochastic (ϵ_g, ϵ_H)-SOSP. Its proof is deferred to Section 6.1.

Theorem 2. *Suppose that Assumption 1 holds with some $H_\nu > 0$ and $\nu \in (0,1]$, and $\epsilon_H \in (0,1)$ is provided for Algorithm 1. Let $\epsilon_g \in (0,1)$ be given, $f_{\rm low}$ and U_H be given in (4), K_1 be defined in (15), η and θ be given in Algorithm 1, and*

$$c_{\rm meo} := (\eta/2)\min\left\{1, \theta((1-\eta)/H_\nu)^{1/\nu}\right\}^2(1/2)^{(2+\nu)/\nu}, \tag{17}$$

$$K_2 := \left\lceil \frac{f(x^0) - f_{\rm low}}{c_{\rm meo}}\epsilon_H^{-(2+\nu)/\nu}\right\rceil + 1. \tag{18}$$

Then the following statements hold.

(i) **(iteration complexity)** *Algorithm 1 terminates in at most $K_1 + 2K_2 - 1$ iterations with*

$$K_1 + 2K_2 - 1 = \mathcal{O}\left(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\right). \tag{19}$$

Moreover, its output x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x^k)) \ge -\epsilon_H$ with probability at least $1-\delta$ for some $0 \le k \le K_1 + 2K_2 - 1$.

(ii) **(operation complexity)** *Algorithm 1 requires at most*

$$\widetilde{\mathcal{O}}\Big(\big(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\big)\min\left\{n, U_H^{1/2}/(H_\nu\epsilon_g^\nu)^{1/(2+2\nu)}\right\} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\min\left\{n, (U_H/\epsilon_H)^{1/2}\right\}\Big)$$

gradient evaluations and Hessian-vector products of f.

Remark 3. (i) *The operation complexity stated in Theorem 2(ii) is a novel contribution to the literature. While similar operation complexity results have been established in [34] for adaptive regularized Newton methods, those results only guarantee finding a point x satisfying $\|\nabla f(x)\| \le \epsilon_g$ and $\lambda_{\min}(\nabla^2 f(x)) \ge -\epsilon_H$, both with high probability. In contrast, the operation complexity in Theorem 2(ii) is achieved by Algorithm 1 for finding a point x satisfying $\|\nabla f(x)\| \le \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x)) \ge -\epsilon_H$ with high probability.*

(ii) *When $\nu = 1$, the iteration and operation complexity results of Algorithm 2 for finding a stochastic (ϵ_g, ϵ_H)-SOSP of (1) are given by $\mathcal{O}\left(L_H^{1/2}\epsilon_g^{-3/2} + L_H^2\epsilon_H^{-3}\right)$ and*

$$\widetilde{\mathcal{O}}\Big(\big(L_H^{1/2}\epsilon_g^{-3/2} + L_H^2\epsilon_H^{-3}\big)\min\left\{n, U_H^{1/2}/(L_H\epsilon_g)^{1/4}\right\} + L_H^2\epsilon_H^{-3}\min\left\{n, (U_H/\epsilon_H)^{1/2}\right\}\Big),$$

respectively, where L_H is the Lipschitz constant $\nabla^2 f$. When $\epsilon_H \ge (L_H\epsilon_g)^{1/2}$, these iteration and operation complexity results reduce to $\mathcal{O}\left(L_H^{1/2}\epsilon_g^{-3/2}\right)$ and $\widetilde{\mathcal{O}}\big(L_H^{1/2}\epsilon_g^{-3/2}\min\left\{n, U_H^{1/2}/(L_H\epsilon_g)^{1/4}\right\}\big)$, respectively. These bounds exhibit improved dependence on L_H compared to those achieved by the Newton-CG methods in [21, 31] for finding a stochastic (ϵ_g, ϵ_H)-SOSP of (1), which are $\mathcal{O}\left(L_H^2\epsilon_g^{-3/2}\right)$ and $\widetilde{\mathcal{O}}\big(L_H^2\epsilon_g^{-3/2}\min\left\{n, U_H^{1/2}/(L_H\epsilon_g)^{1/4}\right\}\big)$, respectively.

4 A parameter-free Newton-CG method for problem (1)

In Section 3, we proposed a Newton-CG method (Algorithm 1) for solving problem (1), assuming that the parameters ν and H_ν associated with the Hölder continuity of $\nabla^2 f$ are explicitly known. This method achieves the best-known iteration complexity for finding an ϵ_g-FOSP deterministically and an (ϵ_g, ϵ_H)-SOSP with high probability, and its fundamental operations rely only on gradient evaluations and Hessian-vector products of f. However, this method requires explicit knowledge of ν and H_ν to compute the quantity $\gamma_\nu(\epsilon_g)$, making it inapplicable to problem (1) when these parameters are unknown. In addition, even when ν and H_ν are known, they may be overly conservative since they must satisfy (3) globally. This conservativeness can result in an excessively large $\gamma_\nu(\epsilon_g)$, potentially leading to slower practical convergence for Algorithm 1. To address these challenges, we propose a parameter-free Newton-CG method (Algorithm 2), which incorporates an innovative backtracking scheme for locally estimating $\gamma_\nu(\epsilon_g)$. This method achieves a similar order of iteration and operation complexity as Algorithm 1, but without requiring any prior knowledge of ν and H_ν.

We now briefly describe the parameter-free Newton-CG method (Algorithm 2) for solving (1). At each outer iteration k, we perform the following operations.

(i) If x^k satisfies $\|\nabla f(x^k)\| > \epsilon_g$, we invoke the capped CG method (Algorithm 3) to solve a damped Newton system

$$\big(\nabla^2 f(x^k) + 2(\sigma_t \epsilon_g)^{1/2} I\big) d = -\nabla f(x^k), \tag{20}$$

where σ_t is a trial value replacing $\gamma_\nu(\epsilon_g)$ in (8). We then evaluate whether the current trial σ_t appropriately estimates $\gamma_\nu(\epsilon_g)$ by performing several checks on the output d_k^t of Algorithm 3 as follows.

- If both $f(x^k + d_k^t) \le f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \le \epsilon_g$ hold, the trial σ_t is deemed an appropriate estimate of $\gamma_\nu(\epsilon_g)$, and d_k^t is accepted as a suitable descent direction for generating the next iterate x^{k+1}.
- If $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$, the trial σ_t is considered an inappropriate estimate of $\gamma_\nu(\epsilon_g)$. In this case, σ_t is increased by a ratio r, and the process is repeated with the updated σ_t.
- If $6\|d_k^t\| \ge (\epsilon_g/\sigma_t)^{1/2}$, a line search is performed to determine whether a suitable step size exists for d_k^t to achieve sufficient reduction in f. If a suitable step size is found, the next iterate x^{k+1} is generated using this step size and the direction d_k^t. If no such step size exists, σ_t is increased by a ratio r and the process is repeated with the updated σ_t.

(ii) If x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$, similar to Algorithm 1, this algorithm offers two options.

- When the tolerance $\epsilon_H \in (0,1)$ for second-order stationarity is not provided, Algorithm 2 terminates with x^k as an ϵ_g-FOSP.
- When $\epsilon_H \in (0,1)$ is provided, a minimum eigenvalue oracle (Algorithm 4) is invoked to either obtain a sufficiently negative curvature direction and generate the next iterate x^{k+1} via a line search, or certify that x^k is an (ϵ_g, ϵ_H)-SOSP with high probability and terminate the algorithm.

In what follows, we present the complexity results for Algorithm 2. For ease of reference, we define an *outer iteration* of Algorithm 2 as one iteration that updates x^k to x^{k+1}, and an *inner iteration* as one call to either Algorithm 3 or Algorithm 4. The following theorem establishes that the number of calls to Algorithm 3 at each outer iteration of Algorithm 2 is finite, ensuring that Algorithm 2 is well-defined. The proof of this result is provided in Section 6.2.

Theorem 3 (well-definedness of Algorithm 2). *Suppose that Assumption 1 holds. Let $\{\gamma_k\}$ be generated by Algorithm 2, $\gamma_\nu(\epsilon_g)$ be defined in (9), and*

$$\sigma(\epsilon_g) := \max\{\gamma_{-1}, r\gamma_\nu(\epsilon_g)\}, \qquad T := \big\lceil \log(\sigma(\epsilon_g)/\gamma_{-1})/\log r \big\rceil_+ + 2, \tag{25}$$

where γ_{-1} and r are the inputs of Algorithm 2. Then, the number of calls of Algorithm 3 at the kth iteration of Algorithm 2 is at most T, and $\gamma_k \le \sigma(\epsilon_g)$ holds for all $k \ge 0$. Moreover, the total number of calls of Algorithms 3 and 4 during the first s outer iterations of Algorithm 2 is at most $T + 2s$.

Algorithm 2 A parameter-free Newton-CG method for problem (1)

input: tolerance $\epsilon_g \in (0,1)$, starting point x^0, CG-accuracy parameter $\zeta \in (0,1)$, trial regularization parameter $\gamma_{-1} > 0$, backtracking ratios $r > 1, \theta \in (0,1)$, line-search parameter $\eta \in (0,1)$, probability parameter $\delta \in (0,1)$; **optional input:** tolerance $\epsilon_H \in (0,1)$.

for $k = 0,1,2,\ldots$ **do**

▷ *This part aims to improve first-order stationarity by calling Algorithm 3.*

if $\|\nabla f(x^k)\| > \epsilon_g$ **then**
  Set $H_k = \nabla^2 f(x^k)$, $g^k = \nabla f(x^k)$, and $\sigma_0 = \max\{\gamma_{-1}, \gamma_{k-1}/r\}$;
  for $t = 0,1,2,\ldots$ **do**
    Set $\sigma_t = r^t \sigma_0$;
    Call Algorithm 3 (Appendix A) with $H = H_k$, $\varepsilon = (\sigma_t \epsilon_g)^{1/2}$, $g = g^k$, accuracy parameter ζ, and $U = 0$ to obtain outputs d, d_type;
    if d_type=NC **then**
      Set

$$d_k^t = -\operatorname{sgn}(d^T g^k) \max\{1, 1/\sigma_t\} \frac{|d^T H_k d|}{\|d\|^3} d;$$

    else {d_type=SOL}
      Set

$$d_k^t = d;$$

    end if
    if d_type=SOL **then**
      if $f(x^k + d_k^t) \le f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \le \epsilon_g$ **then** set $j_t = 0$ and **break** the inner loop;
      else if $6\|d_k^t\| \ge (\epsilon_g/\sigma_t)^{1/2}$ **then**
        Check whether there exists any nonnegative integer j satisfying

$$\theta^j \ge \min\{1, 2(1-\eta)\theta(\epsilon_g/\sigma_t)^{1/4}/(3\|d_k^t\|^{1/2})\}, \tag{21}$$

$$f(x^k + \theta^j d_k^t) \le f(x^k) - \eta(\sigma_t \epsilon_g)^{1/2} \theta^{2j} \|d_k^t\|^2; \tag{22}$$

        If such j exists, set j_t as the smallest nonnegative integer such that (22) holds and **break** the inner loop;
      end if
    else if d_type=NC **then**
      Check whether there exists any nonnegative integer j satisfying

$$\theta^{j-1} \ge \min\{1, 1/\sigma_t\}, \tag{23}$$

$$f(x^k + \theta^j d_k^t) \le f(x^k) - \eta \min\{1, \sigma_t\} \theta^{2j} \|d_k^t\|^3/4; \tag{24}$$

      If such j exists, set j_t as the smallest nonnegative integer such that (24) holds and **break** the inner loop;
    end if
  end for
  Set $(\alpha_k, \gamma_k, d^k) = (\theta^{j_t}, \sigma_t, d_k^t)$;
else if $\|\nabla f(x^k)\| \le \epsilon_g$ and ϵ_H is not provided **then**
  Output x^k and terminate;

▷ *This part aims to improve second-order stationarity by calling Algorithm 4.*

else if $\|\nabla f(x^k)\| \le \epsilon_g$ and ϵ_H is provided **then**
  Call Algorithm 4 (Appendix B) with $H = \nabla^2 f(x^k)$, $\varepsilon = \epsilon_H$, and probability parameter δ;
  if Algorithm 4 certifies that $\lambda_{\min}(\nabla^2 f(x^k)) \ge -\epsilon_H$ **then**
    Output x^k and terminate;
  else {Sufficiently negative curvature direction v returned by Algorithm 4}
    Set $\gamma_k = \gamma_{k-1}$ and

$$d^k \leftarrow -\operatorname{sgn}(v^T \nabla f(x^k)) |v^T \nabla^2 f(x^k) v| v;$$

  end if
  Find $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer j such that

$$f(x^k + \theta^j d^k) \le f(x^k) - \eta \theta^{2j} \|d^k\|^3/2;$$

end if

▷ *This part updates the next iterate.*

Set $x^{k+1} = x^k + \alpha_k d^k$;

end for

The next theorem states the iteration and operation complexity of Algorithm 2 for finding an ϵ_g-FOSP. Its proof is deferred to Section 6.2.

Theorem 4. *Suppose that Assumption 1 holds with some $H_\nu > 0$ and $\nu \in [0,1]$, and ϵ_H is not provided for Algorithm 2. Let $\epsilon_g \in (0,1)$ be given, $f_{\rm low}$, U_H, $c_{\rm nc}$, $\sigma(\epsilon_g)$, and T be respectively given in* (4), (14), *and* (25), *η and θ be given in Algorithm 1, and*

$$\hat{c}_{\rm sol} := \frac{\eta}{6}\min\left\{\frac{1}{6}, \left(\frac{2(1-\eta)\theta}{3}\right)^2\right\}, \tag{26}$$

$$\overline{K}_1 = \left\lceil \frac{f(x^0) - f_{\rm low}}{\min\{\hat{c}_{\rm sol}, c_{\rm nc}\}}\sigma(\epsilon_g)^{1/2}\epsilon_g^{-3/2}\right\rceil + 1. \tag{27}$$

Then the following statements hold.

(i) **(iteration complexity)** *Algorithm 2 requires at most $\overline{K}_1$ outer iterations and $T + 2\overline{K}_1$ inner iterations, where*

$$\overline{K}_1 = \mathcal{O}\left(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)}\right), \tag{28}$$

$$T + 2\overline{K}_1 = \mathcal{O}\big(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)}\big). \tag{29}$$

Moreover, its output x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ for some $0 \le k \le \overline{K}_1$.

(ii) **(operation complexity)** *Algorithm 2 requires at most*

$$\widetilde{\mathcal{O}}\Big(\min\big\{n, U_H^{1/2}/(H_\nu\epsilon_g^\nu)^{1/(2+2\nu)}\big\}H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)}\Big)$$

gradient evaluations and Hessian-vector products of f.

Remark 4. *From Theorems 1 and 4, we observe that Algorithm 2 achieves the same order of iteration and operation complexity as Algorithm 1 for finding an ϵ_g-FOSP of problem* (1)*. Moreover, the iteration complexity matches the lower complexity bound stated in* (2) *(see also [12, 13]).*

The following theorem presents iteration and operation complexity of Algorithm 2 for finding a stochastic (ϵ_g, ϵ_H)-SOSP. Its proof is deferred to Section 6.2.

Theorem 5. *Suppose that Assumption 1 holds with some $H_\nu > 0$ and $\nu \in (0,1]$, and $\epsilon_H \in (0,1)$ is provided for Algorithm 2. Let U_H, K_2, T and $\overline{K}_1$ be defined in* (4), (18), (25) *and* (27)*, respectively. Then the following statements hold.*

(i) **(iteration complexity)** *Algorithm 2 requires at most $\overline{K}_1 + 2K_2 - 1$ outer iterations and $T + 2\overline{K}_1 + 4K_2 - 2$ inner iterations, where*

$$\overline{K}_1 + 2K_2 - 1 = \mathcal{O}\left(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\right), \tag{30}$$

$$T + 2\overline{K}_1 + 4K_2 - 2 = \mathcal{O}\big(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\big). \tag{31}$$

Also, its output x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x^k)) \ge -\epsilon_H$ with probability at least $1-\delta$ for some $0 \le k \le \overline{K}_1 + 2K_2 - 1$.

(ii) **(operation complexity)** *Algorithm 2 requires at most*

$$\widetilde{\mathcal{O}}\Big(\min\big\{n, U_H^{1/2}/\epsilon_g^{1/4}\big\}\big(H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)} + H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\big) + \min\big\{n, (U_H/\epsilon_H)^{1/2}\big\}H_\nu^{2/\nu}\epsilon_H^{-(2+\nu)/\nu}\Big)$$

gradient evaluations and Hessian-vector products of f.

Remark 5. *From Theorems 2 and 5, we observe that Algorithm 2 achieves the same order of iteration complexity as Algorithm 1 for finding an (ϵ_g, ϵ_H)-SOSP of problem* (1) *with high probability.*

n	m	p	Objective value		CPU time (seconds)		Total subproblems	
			Algorithm 2	A-CRN	Algorithm 2	A-CRN	Algorithm 2	A-CRN
100	10	2.25	7.1×10^{-15}	1.7×10^{-14}	0.01	0.23	10.3	28.5
100	10	2.5	1.2×10^{-13}	1.8×10^{-13}	0.02	0.28	11.1	35.8
100	10	2.75	7.2×10^{-13}	3.0×10^{-12}	0.02	0.27	13.4	41.7
100	10	3	1.5×10^{-12}	4.5×10^{-12}	0.04	0.37	13.1	51.8
500	50	2.25	1.8×10^{-16}	2.6×10^{-15}	3.15	7.51	11.5	45.6
500	50	2.5	4.8×10^{-15}	5.9×10^{-15}	6.48	19.14	13.2	53.0
500	50	2.75	7.0×10^{-14}	3.7×10^{-14}	5.29	16.73	14.2	58.8
500	50	3.0	2.3×10^{-13}	3.3×10^{-13}	3.61	8.92	15.3	63.8
1000	100	2.25	1.9×10^{-18}	3.3×10^{-17}	12.82	33.82	11.2	49.6
1000	100	2.5	3.1×10^{-15}	6.3×10^{-15}	16.23	37.34	14.4	58.9
1000	100	2.75	6.8×10^{-15}	3.0×10^{-15}	17.75	39.02	15.3	63.5
1000	100	3	2.8×10^{-14}	1.8×10^{-14}	18.67	43.51	16.5	67.2

Table 1: Numerical results for problem (32)

5 Numerical results

In this section, we conduct preliminary numerical experiments to test the performance of our parameter-free Newton-CG method (Algorithm 2), and compare it with the adaptive cubic regularized Newton method (Universal Method II) in [20]. All the algorithms are coded in Matlab, and all the computations are performed on a laptop with a 2.20 GHz Intel Core i9-14900HX processor and 32 GB of RAM.

5.1 Infeasibility detection problem

In this subsection, we consider the infeasibility detection problem (see [5]):

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m}\left(x^T A_i x + b_i^T x + c_i\right)_+^p, \tag{32}$$

where $p > 2$, $A_i \in \mathbb{R}^{n\times n}$, $b_i \in \mathbb{R}^n$, and $c_i \in \mathbb{R}$ for $1 \le i \le m$.

Our goal is to find a 10^{-4}-FOSP of problem (32) for the generated instances using Algorithm 2 and the adaptive cubic regularized Newton method [20, Universal Method II], and compare their performance. For the adaptive cubic regularized Newton method, we employ a gradient descent method to solve its cubic regularized subproblems, as suggested in [6]. The initial point for the gradient descent method is uniformly selected from the unit sphere, and the tolerances for the subproblems decrease over iterations. For both methods, we initialize with $x^0 = (0,\ldots,0)^T$, and choose the following parameter settings, which appear to suit each method well in terms of computational performance:

- $(\zeta, \gamma_{-1}, \theta, r, \eta) = (0.5, 10, 0.5, 2, 0.01)$ for Algorithm 2;
- $H_0 = 10$ for the adaptive cubic regularized Newton method.

The computational results for Algorithm 2 and the adaptive cubic regularized Newton method (abbreviated as A-CRN) applied to problem (32) are presented in Table 1. The first three columns of the table list the values of n, m, and p, respectively. The remaining columns present the average final objective value, the average CPU time, and the average total number of subproblems over 10 random instances for each triple (n, m, p). Here, a subproblem refers to either one cubic regularized subproblem solved by A-CRN or one damped Newton system solved by Algorithm 2. The results show that both methods produce solutions with comparable final objective values. However, Algorithm 2 significantly outperforms A-CRN [20] in terms of CPU time.

			Objective value		CPU time (seconds)		Total subproblems	
n	m	p	Algorithm 2	A-CRN	Algorithm 2	A-CRN	Algorithm 2	A-CRN
100	20	2.25	0.09	0.10	0.09	0.46	56.9	218.4
100	20	2.5	0.09	0.10	0.10	0.52	60.1	242.8
100	20	2.75	0.09	0.10	0.08	0.87	64.6	382.2
100	20	3	0.10	0.12	0.12	0.61	61.5	325.9
500	100	2.25	0.09	0.10	8.25	10.96	148.6	422.5
500	100	2.5	0.10	0.11	9.16	14.85	153.3	507.7
500	100	2.75	0.10	0.11	9.67	16.91	148.9	567.1
500	100	3	0.11	0.12	11.04	18.83	164.1	646.7
1000	200	2.25	0.10	0.11	29.53	34.92	228.7	631.5
1000	200	2.5	0.10	0.11	34.72	48.42	241.4	800.2
1000	200	2.75	0.11	0.12	39.81	70.43	249.1	856.8
1000	200	3.0	0.12	0.13	46.46	106.92	279.2	1208.9

Table 2: Numerical results for problem (33)

5.2 Single-layer neural networks problem

In this subsection, we consider the problem of training single-layer rectified power unit (RePU) neural networks (see [25]):

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m} \phi((a_i^T x)_+^p - b_i), \tag{33}$$

where $p > 2$, $\phi(t) = t^2/(1+t^2)$ is a nonconvex loss function (see [3, 7]), $a_i \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$ for $1 \le i \le m$.

For each triple (n, m, p), we randomly generate 10 instances of problem (33). In particular, we first randomly generate a_i, $1 \le i \le m$, with all its components sampled from the standard normal distribution. We then randomly generate $\bar{b}_i$, $1 \le i \le m$, according to the standard normal distribution, and set $b_i = |\bar{b}_i|$ for $1 \le i \le m$.

Our goal is to find a 10^{-4}-FOSP of problem (33) fof problem (33) for the generated instances using Algorithm 2 and the adaptive cubic regularized Newton method [20, Universal Method II], and compare their performance. For the adaptive cubic regularized Newton method, we employ a gradient descent approach to solve its cubic regularized subproblems, as suggested in [6]. The initial point for the gradient descent method is uniformly selected from the unit sphere, with tolerances for the subproblems decreasing over iterations. For both methods, we initialize $x_0 = (1/n, \ldots, 1/n)^T$, and adopt the same parameter settings for Algorithm 2 and the cubic regularized Newton method as those specified in Subsection 5.1.

The computational results for Algorithm 2 and the adaptive cubic regularized Newton method (abbreviated as A-CRN) for solving problem (33) are presented in Table 2. The first three columns of the table list the values of n, m, and p, respectively. The remaining columns present the average final objective, average CPU time, and average total number of subproblems over 10 random instances for each triple (n, m, p). Here, a subproblem refers to either a cubic regularized subproblem solved by A-CRN or a damped Newton system solved by Algorithm 2. We observe that both methods find a 10^{-4}-FOSP of (33) with comparable objective values. However, Algorithm 2 is substantially faster than A-CRN [20].

6 Proof of the main results

In this section we provide a proof of our main results presented in Sections 3 and 4, which are particularly Theorems 1-5.

To proceed, we first establish several technical lemmas. The following lemma demonstrates that ∇f admits a first-order approximation with a controllable error, which will play a crucial role in our subsequent analysis. This result is inspired by [29], where it was shown that a function with a Hölder continuous gradient admits a first-order approximation with a controllable error.

Lemma 1. *Under Assumption 1, the following inequality holds for any $\delta > 0$:*

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\| \leq \frac{1}{2}L(\delta)\|y-x\|^2 + \delta, \quad \forall x, y \in \Omega, \tag{34}$$

where

$$L(\delta) = \left(\frac{1-\nu}{2\delta(1+\nu)}\right)^{\frac{1-\nu}{1+\nu}} H_\nu^{\frac{2}{1+\nu}}, \quad \forall \delta > 0.^3 \tag{35}$$

Proof. When $\nu = 1$, it follows from (5) that (34) holds. For the rest of the proof, we suppose that $\nu \in (0,1)$. By Young's inequality, one has that $\tau s \leq \tau^p/p + s^q/q$ for all $\tau, s \geq 0$, where $p, q \geq 1$ satisfy $1/p + 1/q = 1$. Taking $\tau = t^{1+\nu}$, $p = 2/(1+\nu)$, and $q = 2/(1-\nu)$, we obtain that

$$t^{1+\nu} \leq \frac{1+\nu}{2s}t^2 + \frac{1-\nu}{2}s^{\frac{1+\nu}{1-\nu}}, \quad \forall t \geq 0, s > 0.$$

Further, letting $t = \|y-x\|$, $s = \left(\frac{2\delta(1+\nu)}{H_\nu(1-\nu)}\right)^{\frac{1-\nu}{1+\nu}}$, and multiplying both sides of the above inequality by $H_\nu/(1+\nu)$, we have

$$\frac{H_\nu}{1+\nu}\|y-x\|^{1+\nu} \leq \frac{1}{2}\left(\frac{1-\nu}{2\delta(1+\nu)}\right)^{\frac{1-\nu}{1+\nu}} H_\nu^{\frac{2}{1+\nu}}\|y-x\|^2 + \delta,$$

which along with (5) and (35) implies that (34) holds. □

The following lemma provides a useful property of $L(\cdot)$.

Lemma 2. *For any $a \geq 2$, we have*

$$L(\epsilon_g/a) \leq a\gamma_\nu(\epsilon_g)/8, \tag{36}$$

where $\gamma_\nu(\cdot)$ and $L(\cdot)$ are defined in (9) and (35), respectively.

Proof. By $a \geq 2$, $\nu \in [0,1]$, (9), and (35), one has

$$L\left(\frac{\epsilon_g}{a}\right) \overset{(35)}{=} \left(\frac{a(1-\nu)}{2\epsilon_g(1+\nu)}\right)^{\frac{1-\nu}{1+\nu}} H_\nu^{\frac{2}{1+\nu}}\epsilon_g^{-\frac{1-\nu}{1+\nu}} \leq \frac{a}{2}H_\nu^{\frac{2}{1+\nu}}\epsilon_g^{-\frac{1-\nu}{1+\nu}} \overset{(9)}{=} \frac{a}{8}\gamma_\nu(\epsilon_g),$$

where the first inequality is due to $\nu \in [0,1]$ and $\alpha^\alpha \leq 1$ for all $\alpha \in [0,1]$, and the second inequality is due to $\nu \in [0,1]$ and $a \geq 2$. Hence, the conclusion of this lemma holds. □

The next lemma provides two equivalent reformulations of the inequality $\gamma \geq \gamma_\nu(\epsilon_g)$. It will be repeatedly used in our subsequent analysis.

Lemma 3. *Let $\gamma_\nu(\epsilon_g)$ be defined in (9). Then, $\gamma \geq \gamma_\nu(\epsilon_g)$ is equivalent to each of the following two inequalities:*

$$(\gamma\epsilon_g)^{1/2}/H_\nu \geq 2^{1+\nu}(\epsilon_g/\gamma)^{\nu/2}, \tag{37}$$

$$(\gamma\epsilon_g)^{(1-\nu)/2}/H_\nu \geq 2^{1+\nu}/\gamma^\nu. \tag{38}$$

Proof. Dividing both sides of (37) by $\epsilon_g^{1/2}/(H_\nu\gamma^{\nu/2})$ and dividing both sides of (38) by $\epsilon_g^{(1-\nu)/2}/(H_\nu\gamma^\nu)$, we observe that (37) and (38) are both equivalent to $\gamma^{(1+\nu)/2} \geq 2^{1+\nu}H_\nu\epsilon_g^{(\nu-1)/2}$. Taking the $\left(\frac{1+\nu}{2}\right)$th root of this inequality, it becomes

$$\gamma \geq 4H_\nu^{2/(1+\nu)}\epsilon_g^{-(1-\nu)/(1+\nu)} \overset{(9)}{=} \gamma_\nu(\epsilon_g).$$

Hence, the conclusion of this lemma holds. □

[3] By convention, 0^0 is set to 1 throughout this paper.

6.1 Proof of the main results in Section 3

In this subsection, we first establish several technical lemmas and then use them to prove Theorems 1 and 2.

The following lemma presents some useful properties of the output of Algorithm 3 when applied to solving the damped Newton system (8). It is a direct consequence of Lemma 16 given in Appendix A.

Lemma 4. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with a type specified in d_type at some iteration k of Algorithm 1. Then the following statements hold.*

(i) *If d_type=SOL, then d^k satisfies*

$$(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \leq (d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}I)d^k, \tag{39}$$

$$\|d^k\| \leq 1.1(\gamma_\nu(\epsilon_g)\epsilon_g)^{-1/2}\|\nabla f(x^k)\|, \tag{40}$$

$$(d^k)^T\nabla f(x^k) = -(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}I)d^k, \tag{41}$$

$$\|(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}I)d^k + \nabla f(x^k)\| \leq \zeta(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|/2. \tag{42}$$

(ii) *If d_type=NC, then d^k satisfies $(d^k)^T\nabla f(x^k) \leq 0$ and*

$$(d^k)^T\nabla^2 f(x^k)d^k/\|d^k\|^2 = -\min\{1, \gamma_\nu(\epsilon_g)\}\|d^k\| \leq -(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}.$$

The next lemma shows that when the search direction d^k in Algorithm 1 is of type 'SOL', either both $f(x^k + d^k) \leq f(x^k)$ and $\|\nabla f(x^k + d^k)\| \leq \epsilon_g$ hold, or $\|d^k\|$ is uniformly bounded below.

Lemma 5. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with d_type=SOL at some iteration k of Algorithm 1. Then, either both $f(x^k + d^k) \leq f(x^k)$ and $\|\nabla f(x^k + d^k)\| \leq \epsilon_g$ hold, or $6\|d^k\| \geq (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$ holds, where $\gamma_\nu(\epsilon_g)$ is defined in* (9).

Proof. Since d^k results from Algorithm 3 with d_type=SOL, we see that $\|\nabla f(x^k)\| > \epsilon_g$ and (39)-(42) hold for d^k. Moreover, by $\|\nabla f(x^k)\| > \epsilon_g$ and (42), we conclude that $d^k \neq 0$. To prove this lemma, it suffices to show that both $f(x^k + d^k) \leq f(x^k)$ and $\|\nabla f(x^k + d^k)\| \leq \epsilon_g$ hold under the condition $6\|d^k\| < (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$. To this end, we assume that $6\|d^k\| < (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$ holds throughout the remainder of this proof.

We first prove $f(x^k+d^k) \leq f(x^k)$. Suppose for contradiction that $f(x^k+d^k) > f(x^k)$. Let $\varphi(\alpha) = f(x^k+\alpha d^k)$ for all α. Then $\varphi(1) > \varphi(0)$. Also, since $d^k \neq 0$, one has

$$\varphi'(0) = \nabla f(x^k)^T d^k \overset{(41)}{=} -(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}I)d^k \overset{(39)}{\leq} -(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 < 0.$$

In view of these and $\varphi(1) > \varphi(0)$, one can observe that there exists a local minimizer $\alpha_* \in (0,1)$ of φ such that $\varphi'(\alpha_*) = \nabla f(x^k + \alpha_* d^k)^T d^k = 0$ and $\varphi(\alpha_*) < \varphi(0)$. Notice that f is descent along the iterates of Algorithm 1 and hence $f(x^k) \leq f(x^0)$. This together with $\varphi(\alpha_*) < \varphi(0)$ implies that $f(x^k + \alpha_* d^k) < f(x^k) \leq f(x^0)$. Hence, (5) holds for $x = x^k$ and $y = x^k + \alpha_* d^k$. Using this, $\alpha_* \in (0,1)$, (39), (41), and $\nabla f(x^k + \alpha_* d^k)^T d^k = 0$, we deduce that

$$\begin{aligned}
&\tfrac{\alpha_*^{1+\nu} H_\nu}{1+\nu}\|d^k\|^{2+\nu} \overset{(5)}{\geq} \|d^k\|\|\nabla f(x^k + \alpha_* d^k) - \nabla f(x^k) - \alpha_*\nabla^2 f(x^k)d^k\| \\
&\geq (d^k)^T(\nabla f(x^k + \alpha_* d^k) - \nabla f(x^k) - \alpha_*\nabla^2 f(x^k)d^k) = -(d^k)^T\nabla f(x^k) - \alpha_*(d^k)^T\nabla^2 f(x^k)d^k \\
&\overset{(41)}{=} (1-\alpha_*)(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}I)d^k + 2\alpha_*(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \\
&\overset{(39)}{\geq} (1+\alpha_*)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \geq (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2.
\end{aligned}$$

Recall that $\|d^k\| \neq 0$. Dividing both sides of the last inequality by $H_\nu\|d^k\|^2/(1+\nu)$, we obtain that

$$\|d^k\|^\nu \geq \alpha_*^{1+\nu}\|d^k\|^\nu \geq (1+\nu)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/H_\nu, \tag{43}$$

where the first inequality is due to $\alpha_* \in (0,1)$. In addition, letting $\gamma = \gamma_\nu(\epsilon_g)$ in Lemma 3, we obtain from (37) that $(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/H_\nu \geq 2^{1+\nu}(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}$, which together with (43) implies that

$$\|d^k\|^\nu \geq (1+\nu)2^{1+\nu}(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}.$$

Using this and $6\|d^k\| < (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$, we obtain that $(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}/6^\nu > (1+\nu)2^{1+\nu}(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}$, which yields $(1+\nu)2^{1+\nu}6^\nu < 1$. This contradicts the fact $\nu \in [0,1]$. Hence, $f(x^k + d^k) \leq f(x^k)$ holds.

We next prove $\|\nabla f(x^k + d^k)\| \leq \epsilon_g$. As shown above, $f(x^k + d^k) \leq f(x^k)$ holds, which together with $f(x^k) \leq f(x^0)$ implies that $f(x^k + d^k) \leq f(x^k) \leq f(x^0)$. It then follows that (34) holds for $x = x^k$ and $y = x^k + d^k$. By this, (42), and $6\|d^k\| < (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$, one has

$$\begin{aligned}\|\nabla f(x^k + d^k)\| &\leq \|\nabla f(x^k + d^k) - \nabla f(x^k) - \nabla^2 f(x^k) d^k\| + \|(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I) d^k + \nabla f(x^k)\| \\ &\quad + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\| \\ &\leq \frac{L(\epsilon_g/2)}{2}\|d^k\|^2 + \frac{\epsilon_g}{2} + \frac{4+\zeta}{2}(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\| < \frac{L(\epsilon_g/2)}{72}\frac{\epsilon_g}{\gamma_\nu(\epsilon_g)} + \frac{\epsilon_g}{2} + \frac{4+\zeta}{12}\epsilon_g, \end{aligned} \tag{44}$$

where the second inequality follows from (34) and (42), and the third inequality is due to $6\|d^k\| < (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$. Also, notice from (36) with $a = 2$ that $L(\epsilon_g/2) \leq \gamma_\nu(\epsilon_g)/4$. Using this, (44), and $\zeta \in (0,1)$, we obtain that

$$\|\nabla f(x^k + d^k)\| \leq \frac{\epsilon_g}{288} + \frac{\epsilon_g}{2} + \frac{4+\zeta}{12}\epsilon_g < \epsilon_g.$$

Hence, $\|\nabla f(x^k + d^k)\| \leq \epsilon_g$ holds as desired. □

The next lemma shows that when the search direction d^k in Algorithm 1 is of type 'SOL', the line search step results in a sufficient reduction in f.

Lemma 6. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with d_type=SOL at some iteration k of Algorithm 1. Let U_g, $\gamma_\nu(\epsilon_g)$, and $c_{\rm sol}$ be defined in (4), (9), and (14), respectively. Then the following statements hold.*

(i) *The step length α_k is well-defined, and moreover, $\alpha_k \geq \min\{1, (1-\eta)\theta(\epsilon_g/U_g)^{1/2}/2\}$.*

(ii) *The next iterate $x^{k+1} = x^k + \alpha_k d^k$ satisfies either $\|\nabla f(x^{k+1})\| \leq \epsilon_g$ or*

$$f(x^k) - f(x^{k+1}) \geq c_{\rm sol}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2}. \tag{45}$$

Proof. Observe that f is descent along the iterates of Algorithm 1, which implies that $f(x^k) \leq f(x^0)$ and hence $\|\nabla f(x^k)\| \leq U_g$ due to (4). In addition, since d^k results from Algorithm 3 with d_type=SOL, one can see that $\|\nabla f(x^k)\| > \epsilon_g$ and (39)-(42) hold for d^k. Moreover, by $\|\nabla f(x^k)\| > \epsilon_g$ and (42), one can conclude that $d^k \neq 0$.

We first prove statement (i). If $\alpha_k = 1$, then $\alpha_k \geq \min\{1, (1-\eta)\theta(\epsilon_g/U_g)^{1/2}/2\}$ clearly holds. We now suppose that $\alpha_k < 1$. Claim that for all $j \geq 0$ that violate (11), it holds that

$$\theta^{(1+\nu)j} \geq (1-\eta)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/(H_\nu\|d^k\|^\nu). \tag{46}$$

Indeed, suppose that (11) is violated by some $j \geq 0$. We next prove that (46) holds for such j by considering two separate cases below.

Case 1) $f(x^k + \theta^j d^k) > f(x^k)$. Denote $\varphi(\alpha) = f(x^k + \alpha d^k)$. Then $\varphi(\theta^j) > \varphi(0)$. Also, since $d^k \neq 0$, one has

$$\varphi'(0) = \nabla f(x^k)^T d^k \overset{(41)}{=} -(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I)d^k \overset{(39)}{\leq} -(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 < 0.$$

Using these, we can observe that there exists a local minimizer $\alpha_* \in (0, \theta^j)$ of φ such that $\varphi'(\alpha_*) = \nabla f(x^k + \alpha_* d^k)^T d^k = 0$ and $\varphi(\alpha_*) < \varphi(0)$. The latter relation together with $f(x^k) \leq f(x^0)$ implies that $f(x^k + \alpha_* d^k) < f(x^k) \leq f(x^0)$. Hence, (5) holds for $x = x^k$ and $y = x^k + \alpha_* d^k$. Using this, (39), (41), $0 < \alpha_* < \theta^j \leq 1$, and $\nabla f(x^k + \alpha_* d^k)^T d^k = 0$, we deduce that

$$\begin{aligned} &\tfrac{\alpha_*^{1+\nu} H_\nu}{1+\nu}\|d^k\|^{2+\nu} \overset{(5)}{\geq} \|d^k\|\|\nabla f(x^k + \alpha_* d^k) - \nabla f(x^k) - \alpha_*\nabla^2 f(x^k) d^k\| \\ &\geq (d^k)^T(\nabla f(x^k + \alpha_* d^k) - \nabla f(x^k) - \alpha_*\nabla^2 f(x^k)d^k) = -(d^k)^T\nabla f(x^k) - \alpha_*(d^k)^T\nabla^2 f(x^k) d^k \\ &\overset{(41)}{=} (1-\alpha_*)(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I)d^k + 2\alpha_*(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \\ &\overset{(39)}{\geq} (1+\alpha_*)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \geq (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2. \end{aligned}$$

Recall that $\|d^k\| \neq 0$. Dividing both sides of the last inequality by $H_\nu \|d^k\|^{2+\nu}/(1+\nu)$, we obtain that

$$\alpha_*^{1+\nu} \geq (1+\nu)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/(H_\nu\|d^k\|^\nu) \geq (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/(H_\nu\|d^k\|^\nu),$$

which together with $\eta \in (0,1)$ and $\theta^j > \alpha_*$ implies that (46) holds in this case.

Case 2) $f(x^k + \theta^j d^k) \leq f(x^k)$. This together with $f(x^k) \leq f(x^0)$ implies that (6) holds for $x = x^k$ and $y = x^k + \theta^j d^k$. By this and the supposition that j violates (11), we obtain that

$$\begin{aligned}
&-\eta(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\theta^{2j}\|d^k\|^2 \leq f(x^k+\theta^j d^k) - f(x^k)\\
&\overset{(6)}{\leq} \theta^j \nabla f(x^k)^T d^k + \frac{\theta^{2j}}{2}(d^k)^T \nabla^2 f(x^k) d^k + \frac{H_\nu \theta^{(2+\nu)j}\|d^k\|^{2+\nu}}{(1+\nu)(2+\nu)}\\
&\overset{(41)}{=} -\theta^j (d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I)d^k + \frac{\theta^{2j}}{2}(d^k)^T \nabla^2 f(x^k) d^k + \frac{H_\nu \theta^{(2+\nu)j}\|d^k\|^{2+\nu}}{(1+\nu)(2+\nu)}\\
&= -\theta^j\left(1 - \frac{\theta^j}{2}\right)(d^k)^T(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I)d^k - \theta^{2j}(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 + \frac{H_\nu \theta^{(2+\nu)j}\|d^k\|^{2+\nu}}{(1+\nu)(2+\nu)}\\
&\overset{(39)}{\leq} -\theta^j(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 + \frac{H_\nu \theta^{(2+\nu)j}\|d^k\|^{2+\nu}}{(1+\nu)(2+\nu)}. \qquad (47)
\end{aligned}$$

Recall that $d^k \neq 0$. Dividing both sides of (47) by $H_\nu \theta^j \|d^k\|^{2+\nu}/[(1+\nu)(2+\nu)]$, and using $\theta, \eta \in (0,1)$ and $\nu \in [0,1]$, we obtain that

$$\theta^{(1+\nu)j} \geq (1+\nu)(2+\nu)(1-\theta^j\eta)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/(H_\nu\|d^k\|^\nu) \geq (1-\eta)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/(H_\nu\|d^k\|^\nu).$$

Hence, (46) also holds in this case.

Combining the above two cases, we conclude that (46) holds for all $j \geq 0$ violating (11). Letting $\gamma = \gamma_\nu(\epsilon_g)$ in Lemma 3, we know from (37) that $(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}/H_\nu \geq 2^{1+\nu}(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}$, which along with (46) implies that

$$\theta^{(1+\nu)j} \geq (1-\eta)2^{1+\nu}(\epsilon_g/\gamma_\nu(\epsilon_g))^{\nu/2}/\|d^k\|^\nu. \qquad (48)$$

In addition, notice from Algorithm 1 that when $\alpha_k < 1$, at least one of $f(x^k+d^k) \leq f(x^k)$ and $\|\nabla f(x^k+d^k)\| \leq \epsilon_g$ does not hold. It then follows from Lemma 5 that $6\|d^k\| \geq (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$. Using this and taking the $\left(\frac{1}{1+\nu}\right)$th root of both sides of (48), we deduce that

$$\begin{aligned}
\theta^j &\geq 2(1-\eta)^{1/(1+\nu)}\left(\frac{(\epsilon_g/\gamma_\nu(\epsilon_g))^{1/4}}{\|d^k\|^{1/2}}\right)^{2\nu/(1+\nu)}\\
&\geq 2(1-\eta)\left(\frac{(\epsilon_g/\gamma_\nu(\epsilon_g))^{1/4}}{\|d^k\|^{1/2}}\right)^{2\nu/(1+\nu)}\left(\frac{(\epsilon_g/\gamma_\nu(\epsilon_g))^{1/4}}{\sqrt{6}\|d^k\|^{1/2}}\right)^{(1-\nu)/(1+\nu)} \geq \frac{2(1-\eta)(\epsilon_g/\gamma_\nu(\epsilon_g))^{1/4}}{3\|d^k\|^{1/2}}. \qquad (49)
\end{aligned}$$

By this and $\theta \in (0,1)$, one can see that all $j \geq 0$ that violate (11) must be bounded above. It then follows that the step length α_k associated with (11) is well-defined. We next derive a lower bound for α_k. Notice from the definition of j_k in Algorithm 1 that $j = j_k - 1$ violates (11) and hence (49) holds for $j = j_k - 1$. Then, by (49) with $j = j_k - 1$ and $\alpha_k = \theta^{j_k}$, one has

$$\alpha_k = \theta^{j_k} \geq 2(1-\eta)\theta(\epsilon_g/\gamma_\nu(\epsilon_g))^{1/4}/(3\|d^k\|^{1/2}), \qquad (50)$$

which together with (40) and $\|\nabla f(x^k)\| \leq U_g$ implies

$$\alpha_k \geq 2(1-\eta)\theta\epsilon_g^{1/2}/(3\sqrt{1.1}\|\nabla f(x^k)\|^{1/2}) \geq (1-\eta)\theta(\epsilon_g/U_g)^{1/2}/2,$$

and hence statement (i) holds.

We next prove statement (ii). To prove this, it suffices to show that (45) holds under the condition $\|\nabla f(x^{k+1})\| > \epsilon_g$. To this end, we assume that $\|\nabla f(x^{k+1})\| > \epsilon_g$ holds, and prove (45) by considering two separate cases below.

Case 1) $\alpha_k = 1$. By this and the assumption $\|\nabla f(x^{k+1})\| > \epsilon_g$, one can observe from Algorithm 1 that (11) holds for $j = 0$. It then follows that $f(x^k + d^k) \le f(x^k) \le f(x^0)$, which implies that (34) holds for $x = x^k$ and $y = x^k + d^k$. By this, $\alpha_k = 1$, $\|\nabla f(x^{k+1})\| > \epsilon_g$, (34), and (42), one has

$$\begin{aligned}\epsilon_g < \|\nabla f(x^{k+1})\| = \|\nabla f(x^k + d^k)\| &\le \|\nabla f(x^k + d^k) - \nabla f(x^k) - \nabla^2 f(x^k) d^k\| \\ &\quad + \|(\nabla^2 f(x^k) + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} I) d^k + \nabla f(x^k)\| + 2(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\| \\ &\overset{(34)(42)}{\le} \frac{L(\epsilon_g/2)}{2}\|d^k\|^2 + \frac{\epsilon_g}{2} + \frac{4+\zeta}{2}(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|.\end{aligned}$$

Solving the above inequality for $\|d^k\|$ and using $\|d^k\| > 0$, we obtain that

$$\begin{aligned}\|d^k\| &\ge \frac{-(4+\zeta)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} + \sqrt{(4+\zeta)^2\gamma_\nu(\epsilon_g)\epsilon_g + 4L(\epsilon_g/2)\epsilon_g}}{2L(\epsilon_g/2)} \\ &= \frac{2\epsilon_g}{(4+\zeta)(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} + \sqrt{(4+\zeta)^2\gamma_\nu(\epsilon_g)\epsilon_g + 4L(\epsilon_g/2)\epsilon_g}} \ge \frac{2}{4+\zeta+\sqrt{(4+\zeta)^2+1}}\left(\frac{\epsilon_g}{\gamma_\nu(\epsilon_g)}\right)^{1/2},\end{aligned}$$

where the last inequality is due to $L(\epsilon_g/2) \le \gamma_\nu(\epsilon_g)/4$ (see (36) with $a = 2$). By the above inequality, $\alpha_k = 1$, and (11), one has

$$f(x^k) - f(x^{k+1}) \ge \eta(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\|d^k\|^2 \ge \eta\left(\frac{2}{4+\zeta+\sqrt{(4+\zeta)^2+1}}\right)^2 \frac{\epsilon_g^{3/2}}{\gamma_\nu(\epsilon_g)^{1/2}},$$

and hence statement (i) holds in this case.

Case 2) $\alpha_k < 1$. By this, one can observe from Algorithm 1 that at least one of $\|\nabla f(x^k + d^k)\| \le \epsilon_g$ and $f(x^k + d^k) \le f(x^k)$ does not hold. It then follows from Lemma 5 that $6\|d^k\| \ge (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$. Using this, (11), and (50), we can deduce that

$$f(x^k) - f(x^{k+1}) \overset{(11)}{>} \eta(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}\theta^{2j_k}\|d^k\|^2 \overset{(50)}{\ge} \eta\left(\frac{2(1-\eta)\theta}{3}\right)^2 \epsilon_g\|d^k\| \ge \frac{\eta}{6}\left(\frac{2(1-\eta)\theta}{3}\right)^2 \frac{\epsilon_g^{3/2}}{\gamma_\nu(\epsilon_g)^{1/2}},$$

where the last inequality is due to $6\|d^k\| \ge (\epsilon_g/\gamma_\nu(\epsilon_g))^{1/2}$. By the above inequality and the definition of $c_{\rm sol}$ in (14), one can see that (45) also holds in this case. □

The following lemma shows when the search direction d^k in Algorithm 1 is of type 'NC', the line search step results in a sufficient reduction on f as well.

Lemma 7. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with d_type=NC at some iteration k of Algorithm 1. Let $\gamma_\nu(\epsilon_g)$ and $c_{\rm nc}$ be defined in (9) and (14), respectively. Then the following statements hold.*

(i) *The step length α_k is well-defined, and $\alpha_k \ge \theta\min\{1, 1/\gamma_\nu(\epsilon_g)\}$.*

(ii) *The next iterate $x^{k+1} = x^k + \alpha_k d^k$ satisfies $f(x^k) - f(x^{k+1}) \ge c_{\rm nc}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2}$.*

Proof. Observe that f is descent along the iterates of Algorithm 1 and hence $f(x^k) \le f(x^0)$. Since d^k results from Algorithm 3 with d_type=NC, one can see from Lemma 4(ii) that

$$\nabla f(x^k)^T d^k \le 0, \quad (d^k)^T\nabla^2 f(x^k) d^k/\|d^k\|^2 = -\min\{1, \gamma_\nu(\epsilon_g)\}\|d^k\| \le -(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} < 0. \tag{51}$$

We first prove statement (i). If (12) holds for $j = 0$, then $\alpha_k = 1$, which together with $\theta \in (0,1)$ implies that $\alpha_k \ge \theta\min\{1, 1/\gamma_\nu(\epsilon_g)\}$ holds. We now suppose that (12) fails for $j = 0$. Claim that for all $j \ge 0$ that violate (12), it holds that

$$\theta^{\nu j} > (1-\eta/2)\min\{1, \gamma_\nu(\epsilon_g)\}^\nu(\gamma_\nu(\epsilon_g)\epsilon_g)^{(1-\nu)/2}/H_\nu. \tag{52}$$

Indeed, suppose that (12) is violated by some $j \geq 0$. We now show that (52) holds for such j by considering two separate cases below.

Case 1) $f(x^k + \theta^j d^k) > f(x^k)$. Let $\varphi(\alpha) = f(x^k + \alpha d^k)$. Then $\varphi(\theta^j) > \varphi(0)$. Also, by (51), one has

$$\varphi'(0) = \nabla f(x^k)^T d^k \leq 0, \quad \varphi''(0) = (d^k)^T \nabla^2 f(x^k) d^k < 0.$$

By these and $\varphi(\theta^j) > \varphi(0)$, it is not hard to observe that there exists a local minimizer $\alpha_* \in (0, \theta^j)$ of φ such that $\varphi(\alpha_*) < \varphi(0)$, namely, $f(x^k + \alpha_* d^k) < f(x^k)$. Further, by the second-order optimality condition of φ at α_*, one has $\varphi''(\alpha_*) = (d^k)^T f(x^k + \alpha_* d^k) d^k \geq 0$. Since $f(x^k + \alpha_* d^k) < f(x^k) \leq f(x^0)$, it follows that (3) holds for $x = x^k$ and $y = x^k + \alpha_* d^k$. Using this, the second relation in (51), and $(d^k)^T \nabla^2 f(x^k + \alpha_* d^k) d^k \geq 0$, we obtain that

$$\begin{aligned} H_\nu \alpha_*^\nu \|d^k\|^{2+\nu} &\geq \|d^k\|^2 \|\nabla^2 f(x^k + \alpha_* d^k) - \nabla^2 f(x^k)\| \geq (d^k)^T (\nabla^2 f(x^k + \alpha_* d^k) - \nabla^2 f(x^k)) d^k \\ &\geq -(d^k)^T \nabla^2 f(x^k) d^k = \min\{1, \gamma_\nu(\epsilon_g)\} \|d^k\|^3. \end{aligned} \tag{53}$$

Recall from (51) that $\|d^k\| \geq \max\{1, 1/\gamma_\nu(\epsilon_g)\} (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} > 0$. Using this, $\theta^j > \alpha_*$, and (53), we deduce that

$$\theta^{\nu j} \geq \alpha_*^\nu \overset{(53)}{\geq} \min\{1, \gamma_\nu(\epsilon_g)\} \|d^k\|^{1-\nu} / H_\nu \geq \min\{1, \gamma_\nu(\epsilon_g)\}^\nu (\gamma_\nu(\epsilon_g)\epsilon_g)^{(1-\nu)/2} / H_\nu,$$

which together with $\eta \in (0, 1)$ implies that (52) holds in this case.

Case 2) $f(x^k + \theta^j d^k) \leq f(x^k)$. This and $f(x^k) \leq f(x^0)$ imply that $f(x^k + \theta^j d^k) \leq f(x^k) \leq f(x^0)$. It then follows that (6) holds for $x = x^k$ and $y = x^k + \theta^j d^k$. By this, (51), and the supposition that j violates (12), one has

$$\begin{aligned} -\frac{\eta}{4} \min\{1, \gamma_\nu(\epsilon_g)\} \theta^{2j} \|d^k\|^3 &< f(x^k + \theta^j d^k) - f(x^k) \\ &\overset{(6)}{\leq} \theta^j \nabla f(x^k)^T d^k + \frac{\theta^{2j}}{2} (d^k)^T \nabla^2 f(x^k) d^k + \frac{H_\nu \theta^{(2+\nu)j}}{(1+\nu)(2+\nu)} \|d^k\|^{2+\nu} \\ &\overset{(51)}{\leq} -\frac{1}{2} \min\{1, \gamma_\nu(\epsilon_g)\} \theta^{2j} \|d^k\|^3 + \frac{H_\nu \theta^{(2+\nu)j}}{(1+\nu)(2+\nu)} \|d^k\|^{2+\nu}, \end{aligned}$$

which together with $\|d^k\| \geq \max\{1, 1/\gamma_\nu(\epsilon_g)\} (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2} > 0$, $\eta \in (0, 1)$, and $\nu \in [0, 1]$ implies that

$$\theta^{\nu j} > (1+\nu)(2+\nu)(1/2 - \eta/4) \min\{1, \gamma_\nu(\epsilon_g)\} \|d^k\|^{1-\nu} / H_\nu \geq (1 - \eta/2) \min\{1, \gamma_\nu(\epsilon_g)\}^\nu (\gamma_\nu(\epsilon_g)\epsilon_g)^{(1-\nu)/2} / H_\nu,$$

and hence (52) also holds in this case.

Combining the above two cases, we conclude that (52) holds for any $j \geq 0$ violating (12). Letting $\gamma = \gamma_\nu(\epsilon_g)$ in Lemma 3, we obtain from (38) that $(\gamma_\nu(\epsilon_g)\epsilon_g)^{(1-\nu)/2} / H_\nu \geq 2^{1+\nu} / \gamma_\nu(\epsilon_g)^\nu$, which together with (52) and $\eta \in (0, 1)$ implies that for any $j \geq 0$ violating (12),

$$\theta^{\nu j} > (1 - \eta/2) \min\{1, \gamma_\nu(\epsilon_g)\}^\nu 2^{1+\nu} / \gamma_\nu(\epsilon_g)^\nu > 2^\nu \min\{1, \gamma_\nu(\epsilon_g)\}^\nu / \gamma_\nu(\epsilon_g)^\nu. \tag{54}$$

When $\nu = 0$, one can see that (54) does not hold for $j = 0$, which implies that (12) holds for $j = 0$ and hence $\alpha_k = 1$. When $\nu \in (0, 1]$, one can see from (54) that all $j \geq 0$ violating (12) must be bounded above, and hence α_k is well-defined. We next derive a lower bound for α_k. Notice from the definition of j_k that $j = j_k - 1$ violates (12). Then, using (54) with $j = j_k - 1$ and $\alpha_k = \theta^{j_k}$, we see that $\alpha_k = \theta^{j_k} \geq \theta \min\{1, 1/\gamma_\nu(\epsilon_g)\}$. Hence, statement (i) holds.

We next prove statement (ii). Recall from (51) that $\|d^k\| \geq \max\{1, 1/\gamma_\nu(\epsilon_g)\} (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2}$. It then follows from this, $\alpha_k \geq \theta \min\{1, 1/\gamma_\nu(\epsilon_g)\}$, and (12) that

$$\begin{aligned} f(x^k) - f(x^{k+1}) &\geq \frac{\eta}{4} \min\{1, \gamma_\nu(\epsilon_g)\} \alpha_k^2 \|d^k\|^3 \\ &= \frac{\eta}{4} \theta^2 \min\{1, \gamma_\nu(\epsilon_g)\} \min\{1, 1/\gamma_\nu(\epsilon_g)\}^2 \max\{1, 1/\gamma_\nu(\epsilon_g)\}^3 (\gamma_\nu(\epsilon_g)\epsilon_g)^{3/2} \\ &= \frac{\eta}{4} \theta^2 \min\{1, \gamma_\nu(\epsilon_g)\}^3 \max\{1, 1/\gamma_\nu(\epsilon_g)\}^3 \epsilon_g^{3/2} / \gamma_\nu(\epsilon_g)^{1/2} = \frac{\eta}{4} \theta^2 \epsilon_g^{3/2} / \gamma_\nu(\epsilon_g)^{1/2}. \end{aligned}$$

This together with the definition of $c_{\rm nc}$ in (14) implies that statement (ii) holds. □

We are now ready to prove Theorem 1.

Proof of Theorem 1. (i) Recall from the assumption of this theorem that ϵ_H is not provided for Algorithm 1. It then follows that Algorithm 3 is called at each iteration of Algorithm 1, except the last iteration. Suppose for contradiction that the total number of iterations of Algorithm 1 is more than K_1. Observe from Algorithm 1 and Lemmas 6(ii) and 7(ii) that each call of Algorithm 3 results in a reduction on f at least by $\min\{c_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2}$. Using this and (4), we have

$$K_1 \min\{c_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2} \le \sum_{k=0}^{K_1-1}(f(x^k) - f(x^{k+1})) = f(x^0) - f(x^{K_1}) \le f(x^0) - f_{\rm low},$$

which contradicts the definition of K_1 given in (15). Hence, the total number of iterations of Algorithm 1 is no more than K_1. In addition, the relation (16) follows from (9), (15), and (14). Since ϵ_H is not provided, one can also observe from Algorithm 1 that its output x^k satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ for some $0 \le k \le K_1$. This completes the proof of statement (i) of Theorem 1.

(ii) Notice that f is descent along the iterates generated by Algorithm 1, which implies that $f(x^k) \le f(x^0)$ for all $0 \le k \le K_1$. Using this and (4), we have that $\|\nabla^2 f(x^k)\| \le U_H$ for all $0 \le k \le K_1$. By Theorem 6 with $(H, \varepsilon) = (\nabla^2 f(x^k), (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2})$ and the fact that $\|\nabla^2 f(x^k)\| \le U_H$ for all k, one can observe that the number of gradient evaluations and Hessian-vector products of f required by each call of Algorithm 3 with $U = 0$ in Algorithm 1 is at most $\widetilde{\mathcal{O}}(\min\{n, U_H^{1/2}/(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/4}\})$. In view of this and statement (i), we see that statement (ii) of Theorem 1 holds . $\square$

The following lemma shows that when the search direction d^k in Algorithm 1 is a negative curvature direction returned from Algorithm 4, the next iterate x^{k+1} produces a sufficient reduction on f.

Lemma 8. *Suppose that Assumption 1 holds with $\nu \in (0, 1]$, and the direction d^k results from Algorithm 4 at some iteration k of Algorithm 1. Let $c_{\rm meo}$ be defined in (17). Then the following statements hold.*

(i) *The step length α_k is well-defined, and $\alpha_k \ge \min\{1, \theta((1-\eta)/H_\nu)^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\}$.*

(ii) *The next iterate $x^{k+1} = x^k + \alpha_k d^k$ satisfies $f(x^k) - f(x^{k+1}) \ge c_{\rm meo}\epsilon_H^{(2+\nu)/\nu}$.*

Proof. Observe that f is descent along the iterates generated by Algorithm 1, which implies that $f(x^k) \le f(x^0)$. Since d^k results from Algorithm 4, it follows from Algorithm 1 that d^k is given in (10) with v being the vector returned from Algorithm 4 with $H = \nabla^2 f(x^k)$ and $\varepsilon = \epsilon_H$ that satisfies $\|v\| = 1$ and $v^T\nabla^2 f(x^k)v \le -\epsilon_H/2$. By these and (10), one can see that

$$\nabla f(x^k)^T d^k \le 0, \quad (d^k)^T\nabla^2 f(x^k)d^k/\|d^k\|^2 = -\|d^k\| = v^T\nabla^2 f(x^k)v \le -\epsilon_H/2 < 0. \tag{55}$$

We first prove statement (i). If (13) holds for $j = 0$, then we have $\alpha_k = 1$, which clearly implies that $\alpha_k \ge \min\{1, \theta[(1-\eta)/H_\nu]^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\}$. We now suppose that (13) fails for $j = 0$. Claim that for all $j \ge 0$ that violate (13), it holds that

$$\theta^j \ge ((1-\eta)/H_\nu)^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}. \tag{56}$$

Indeed, suppose that (13) is violated by some $j \ge 0$. We now show that (56) holds for such j by considering two separate cases below.

Case 1) $f(x^k + \theta^j d^k) > f(x^k)$. Let $\varphi(\alpha) = f(x^k + \alpha d^k)$. Then $\varphi(\theta^j) > \varphi(0)$. Also, by (55), one has

$$\varphi'(0) = \nabla f(x^k)^T d^k \le 0, \quad \varphi''(0) = (d^k)^T\nabla^2 f(x^k)d^k < 0.$$

By these and $\varphi(\theta^j) > \varphi(0)$, it is not hard to observe that there exists a local minimizer $\alpha_* \in (0, \theta^j)$ of φ such that $\varphi(\alpha_*) < \varphi(0)$, namely, $f(x^k + \theta^j d^k) < f(x^k)$. By the second-order optimality condition of φ, one has $\varphi''(\alpha_*) = (d^k)^T\nabla^2 f(x^k + \alpha_* d^k)d^k \ge 0$. Since $f(x^k + \alpha_* d^k) \le f(x^k) \le f(x^0)$, it follows that (3) holds for $x = x^k$ and $y = x^k + \alpha_* d^k$. Using this, the second relation in (55), and $(d^k)^T\nabla^2 f(x^k + \alpha_* d^k)d^k \ge 0$, we obtain that

$$H_\nu\alpha_*^\nu\|d^k\|^{2+\nu} \ge \|d^k\|^2\|\nabla^2 f(x^k + \alpha_* d^k) - \nabla^2 f(x^k)\| \ge (d^k)^T(\nabla^2 f(x^k + \alpha_* d^k) - \nabla^2 f(x^k))d^k$$

$$\geq -(d^k)^T\nabla^2 f(x^k)d^k = \|d^k\|^3.$$

Recall from (55) that $d^k \neq 0$. Dividing both sides of this inequality by $H_\nu\|d^k\|^{2+\nu}$ yields $\alpha_*^\nu \geq \|d^k\|^{1-\nu}/H_\nu$, which along with $\theta^j > \alpha_*$, $\nu \in (0,1]$, and $\|d^k\| \geq \epsilon_H/2$ implies that $\theta^j \geq (\epsilon_H/2)^{(1-\nu)/\nu}/H_\nu^{1/\nu}$ and hence (56) holds in this case.

Case 2) $f(x^k+\theta^j d^k) \leq f(x^k)$. This and $f(x^k) \leq f(x^0)$ imply that $f(x^k+\theta^j d^k) \leq f(x^k) \leq f(x^0)$. It then follows that (6) holds for $x = x^k$ and $y = x^k + \theta^j d^k$. By this and the supposition that j violates (13), one has

$$-\frac{\eta}{2}\theta^{2j}\|d^k\|^3 \leq f(x^k+\theta^j d^k) - f(x^k) \overset{(6)}{\leq} \theta^j \nabla f(x^k)^T d^k + \frac{\theta^{2j}}{2}(d^k)^T\nabla^2 f(x^k)d^k + \frac{H_\nu\theta^{(2+\nu)j}}{(1+\nu)(2+\nu)}\|d^k\|^{2+\nu}$$
$$\overset{(55)}{\leq} -\frac{\theta^{2j}}{2}\|d^k\|^3 + \frac{H_\nu\theta^{(2+\nu)j}}{2}\|d^k\|^{2+\nu},$$

where the last inequality is due to $\nu \in (0,1]$ and (55). By this and $d^k \neq 0$, one has $\theta^{\nu j} \geq (1-\eta)\|d^k\|^{1-\nu}/H_\nu$. This together with $\nu \neq 0$ implies that (56) holds in this case as well.

Combining the above two cases, we conclude that (56) holds for all $j \geq 0$ that violate (13). By this and $\theta \in (0,1)$, one can see that all $j \geq 0$ violating (13) must be bounded above. It then follows that the step length α_k associated with (13) is well-defined. We next derive a lower bound for α_k. Notice from the definition of j_k in Algorithm 1 that $j = j_k - 1$ violates (13) and hence (56) holds for $j = j_k - 1$. Then, by (56) with $j = j_k - 1$ and $\alpha_k = \theta^{j_k}$, one has $\alpha_k = \theta^{j_k} \geq \theta[(1-\eta)/H_\nu]^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}$. Hence, $\alpha_k \geq \min\{1, \theta[(1-\eta)/H_\nu]^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\}$ holds as desired.

We next prove statement (ii). Recall from (55) that $\|d^k\| \geq \epsilon_H/2$. In view of this, (13), and the fact that $\alpha_k \geq \min\{1, \theta[(1-\eta)/H_\nu]^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\}$, we obtain that

$$\begin{aligned} f(x^k) - f(x^{k+1}) &> \frac{\eta}{2}\alpha_k^2\|d^k\|^3 \geq \frac{\eta}{2}\min\left\{1, \theta((1-\eta)/H_\nu)^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\right\}^2(\epsilon_H/2)^3 \\ &= \frac{\eta}{2}\min\left\{(\epsilon_H/2)^{-(1-\nu)/\nu}, \theta((1-\eta)/H_\nu)^{1/\nu}\right\}^2(\epsilon_H/2)^{(2+\nu)/\nu} \\ &\geq \frac{\eta}{2}\min\left\{1, \theta((1-\eta)/H_\nu)^{1/\nu}\right\}^2(\epsilon_H/2)^{(2+\nu)/\nu}, \end{aligned}$$

where the last inequality is due to $\epsilon_H \in (0,1)$ and $\nu \in (0,1]$. By this inequality and the definition of $c_{\rm meo}$ in (17), one can see that statement (ii) holds. □

We are now ready to provide a proof of Theorem 2.

Proof of Theorem 2. (i) Let K_1 and K_2 be defined in (15) and (18), respectively. We first claim that the total number of calls of Algorithm 4 in Algorithm 1 is at most K_2. Indeed, suppose for contradiction that its total number of calls is more than K_2. Observe from Algorithm 1 and Lemma 8(ii) that each of these calls, except the last one, results in a reduction on f at least by $c_{\rm meo}\epsilon_H^{(2+\nu)/\nu}$. Since f is descent along the iterates of Algorithm 1 and $f(x^k) \geq f_{\rm low}$, the total amount of reduction on f resulting from the calls of Algorithm 4 in Algorithm 1 is at most $f(x^0) - f_{\rm low}$. Combining these observations, one has

$$K_2 c_{\rm meo}\epsilon_H^{(2+\nu)/\nu} \leq f(x^0) - f_{\rm low},$$

which contradicts the definition of K_2 in (18). Hence, the total number of calls of Algorithm 4 is at most K_2.

We next claim that the total number of calls of Algorithm 3 in Algorithm 1 is at most $K_1 + K_2 - 1$. Indeed, suppose for contradiction that its total number of calls is more than $K_1 + K_2 - 1$. Notice that if Algorithm 3 is called at some iteration k and generates x^{k+1} satisfying $\|\nabla f(x^{k+1})\| \leq \epsilon_g$, then Algorithm 4 must be called at the iteration $k+1$. In view of this and the fact that the total number of calls of Algorithm 4 is at most K_2, one can observe that the total number of such iterations is at most K_2. This along with the above supposition implies that the total number of iterations k of Algorithm 1 at which Algorithm 3 is called and generates the next iterate x^{k+1} satisfying $\|\nabla f(x^{k+1})\| > \epsilon_g$ is at least K_1. For each of these iterations k, we observe from Lemmas 6(ii) and 7(ii) that $f(x^k) - f(x^{k+1}) \geq \min\{c_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2}$. Since f is descent along the iterates of Algorithm 1 and $f(x^k) \geq f_{\rm low}$, the total amount of reduction on f resulting from these iterations k is at most $f(x^0) - f_{\rm low}$. It then follows that

$$K_1 \min\{c_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2} \leq f(x^0) - f_{\rm low},$$

which contradicts the definition of K_1 given in (15). Hence, the total number of calls of Algorithm 3 in Algorithm 1 is at most $K_1 + K_2 - 1$.

Based on the above claims and the fact that either Algorithm 3 or 4 is called at each iteration of Algorithm 1, we conclude that the total number of iterations of Algorithm 1 is at most $K_1 + 2K_2 - 1$. In addition, relation (19) follows from (9), (14), (15), (17), and (18). Moreover, one can observe that the output x^k of Algorithm 1 satisfies $\|\nabla f(x^k)\| \leq \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x^k)) \geq -\epsilon_H$ with probability at least $1-\delta$ for some $0 \leq k \leq K_1 + 2K_2 - 1$, where the latter part is due to Algorithm 4. This completes the proof of statement (i) of Theorem 2.

(ii) By Theorem 6 with $(H, \varepsilon) = (\nabla^2 f(x^k), (\gamma_\nu(\epsilon_g)\epsilon_g)^{1/2})$ and the fact that $\|\nabla^2 f(x^k)\| \leq U_H$, one can observe that the number of gradient evaluations and Hessian-vector products of f required by each call of Algorithm 3 with input $U = 0$ is at most $\widetilde{\mathcal{O}}(\min\{n, U_H^{1/2}/(\gamma_\nu(\epsilon_g)\epsilon_g)^{1/4}\})$. In addition, by Theorem 7 with $(H, \varepsilon) = (\nabla^2 f(x^k), \epsilon_H)$, $\|\nabla^2 f(x^k)\| \leq U_H$ and the fact that each iteration of Algorithm 4 requires only one Hessian-vector product of f, one can observe that the number of Hessian-vector products required by each call of Algorithm 4 is at most $\widetilde{\mathcal{O}}(\min\{n, (U_H/\epsilon_H)^{1/2}\})$. Using these and statement (i) of Theorem 2, we see that statement (ii) of Theorem 2 holds. □

6.2 Proof of the main results in Section 4

In this subsection, we establish several technical lemmas, and then provide a proof of Theorems 3, 4, and 5.

The following lemma presents some useful properties of the output of Algorithm 3 when applied to solving the damped Newton system (20). It is a direct consequence of Lemma 16 given in Appendix A.

Lemma 9. *Suppose that Assumption 1 holds and the direction d_k^t results from Algorithm 3 with a type specified in d_type at some inner iteration t of Algorithm 2. Then the following statements hold.*

(i) *If d_type=SOL, then d_k^t satisfies*

$$(\sigma_t\epsilon_g)^{1/2}\|d_k^t\|^2 \leq (d_k^t)^T(\nabla^2 f(x^k) + 2(\sigma_t\epsilon_g)^{1/2} I)d_k^t, \quad (57)$$

$$\|d_k^t\| \leq 1.1(\sigma_t\epsilon_g)^{-1/2}\|\nabla f(x^k)\|, \quad (58)$$

$$(d_k^t)^T\nabla f(x^k) = -(d_k^t)^T(\nabla^2 f(x^k) + 2(\sigma_t\epsilon_g)^{1/2} I)d_k^t, \quad (59)$$

$$\|(\nabla^2 f(x^k) + 2(\sigma_t\epsilon_g)^{1/2} I)d_k^t + \nabla f(x^k)\| \leq \zeta(\sigma_t\epsilon_g)^{1/2}\|d_k^t\|/2. \quad (60)$$

(ii) *If d_type=NC, then d_k^t satisfies $\nabla f(x^k)^T d_k^t \leq 0$ and*

$$(d_k^t)^T\nabla^2 f(x^k) d_k^t/\|d_k^t\|^2 = -\min\{1, \sigma_t\}\|d_k^t\| \leq -(\sigma_t\epsilon_g)^{1/2}.$$

The following lemma generalizes Lemma 5 for Algorithm 2 with any $\sigma_t \geq \gamma_\nu(\epsilon_g)$.

Lemma 10. *Suppose that Assumption 1 holds and the direction d_k^t results from Algorithm 3 with d_type=SOL at some inner iteration t of the kth outer iteration of Algorithm 2. Assume that $\sigma_t \geq \gamma_\nu(\epsilon_g)$ holds, where $\gamma_\nu(\epsilon_g)$ is given in (9). Then, either both $f(x^k + d_k^t) \leq f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ hold, or $6\|d_k^t\| \geq (\epsilon_g/\sigma_t)^{1/2}$ holds.*

Proof. Since d_k^t results from Algorithm 3 with d_type=SOL, we see that $\|\nabla f(u)\| > \epsilon_g$ and (57)-(60) hold for d_k^t. Moreover, by by $\|\nabla f(x^k)\| > \epsilon_g$ and (60), we conclude that $d_k^t \neq 0$. To prove this lemma, it suffices to show that both $f(x^k + d_k^t) \leq f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ hold under the condition $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$. To this end, we assume that $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$ holds throughout the remainder of this proof.

We first prove $f(x^k + d_k^t) \leq f(x^k)$. Suppose for contradiction that $f(x^k + d_k^t) > f(x^k)$. Using the same arguments as for (43) with $(d^k, \gamma_\nu(\epsilon_g))$ replaced by (d_k^t, σ_t), we can have $\|d_k^t\|^\nu \geq (1+\nu)(\sigma_t\epsilon_g)^{1/2}/H_\nu$. Letting $\gamma = \sigma_t$ in Lemma 3, we observe from (37) that $(\sigma_t\epsilon_g)^{1/2}/H_\nu \geq 2^{1+\nu}(\epsilon_g/\sigma_t)^{\nu/2}$. Using these, we obtain that

$\|d_k^t\|^\nu \geq (1+\nu)2^{1+\nu}(\epsilon_g/\sigma_t)^{\nu/2}$, which contradicts the assumption that $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$, given that $\nu \in [0,1]$. Hence, $f(x^k + d_k^t) \leq f(x^k)$ holds as desired.

We now prove $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$. Recall that $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$, $\sigma_t \geq \gamma_\nu(\epsilon_g)$, and $L(\epsilon_g/2) \leq \gamma_\nu(\epsilon_g)/4$ (see (36) with $a = 2$). By these and the same arguments as for (44) with $(d^k, \gamma_\nu(\epsilon_g))$ replaced by (d_k^t, σ_t), one can have

$$\begin{aligned}\|\nabla f(x^k + d_k^t)\| &\leq \frac{L(\epsilon_g/2)}{2}\|d_k^t\|^2 + \frac{\epsilon_g}{2} + \frac{4+\zeta}{2}(\sigma_t\epsilon_g)^{1/2}\|d_k^t\| \\ &< \frac{L(\epsilon_g/2)}{72}\frac{\epsilon_g}{\sigma_t} + \frac{\epsilon_g}{2} + \frac{4+\zeta}{12}\epsilon_g \leq \frac{\gamma_\nu(\epsilon_g)}{288}\frac{\epsilon_g}{\sigma_t} + \frac{\epsilon_g}{2} + \frac{4+\zeta}{12}\epsilon_g \leq \epsilon_g,\end{aligned}$$

where the second inequality is due to $6\|d_k^t\| < (\epsilon_g/\sigma_t)^{1/2}$, the third inequality follows from $L(\epsilon_g/2) \leq \gamma_\nu(\epsilon_g)/4$, and the last inequality is due to $\sigma_t \geq \gamma_\nu(\epsilon_g)$. Hence, $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ holds. □

The next lemma shows that when d_k^t generated in Algorithm 2 is associated with d_type=SOL and $\sigma_t \geq \gamma_\nu(\epsilon)$, Algorithm 2 breaks its inner loop at the inner iteration t.

Lemma 11. *Suppose that Assumption 1 holds and the direction d_k^t results from Algorithm 3 with d_type=SOL at some inner iteration t of the kth outer iteration of Algorithm 2. Assume that $\sigma_t \geq \gamma_\nu(\epsilon_g)$ holds, where $\gamma_\nu(\epsilon_g)$ is given in* (9). *Then, either both $f(x^k + d_k^t) \leq f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ hold, or there exists some nonnegative integer j satisfying* (21) *and* (22).

Proof. Since d_k^t results from Algorithm 3 with d_type=SOL, one can see that $\|\nabla f(x^k)\| > \epsilon_g$ and (57)-(60) hold for d_k^t. To prove this lemma, it suffices to show that if at least one of $f(x^k + d_k^t) \leq f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ does not hold, there exists some nonnegative integer j satisfying (21) and (22). To this end, we assume throughout the remainder of this proof that at least one of $f(x^k + d_k^t) \leq f(x^k)$ and $\|\nabla f(x^k + d_k^t)\| \leq \epsilon_g$ does not hold. It then follows from Lemma 10 that $6\|d_k^t\| \geq (\epsilon_g/\sigma_t)^{1/2}$.

If (22) holds with $j = 0$, then (21) and (22) hold for $j = 0$ and hence the conclusion of this lemma holds. Now, we suppose that (22) is violated by some $j \geq 0$. Using the same arguments as for (46) with $(d^k, \gamma_\nu(\epsilon_g))$ replaced by (d_k^t, σ_t), we can have that all $j \geq 0$ violating (22) satisfy

$$\theta^{(1+\nu)j} \geq (1-\eta)(\sigma_t\epsilon_g)^{1/2}/(H_\nu\|d_k^t\|^\nu). \tag{61}$$

Recall that $\sigma_t \geq \gamma_\nu(\epsilon_g)$. Then, letting $\gamma = \sigma_t$ in Lemma 3, we have from (37) that $(\sigma_t\epsilon_g)^{1/2}/H_\nu \geq 2^{1+\nu}(\epsilon_g/\sigma_t)^{\nu/2}$, which together with (61) implies that all $j \geq 0$ violating (22) satisfy

$$\theta^{(1+\nu)j} \geq (1-\eta)2^{1+\nu}(\epsilon_g/\sigma_t)^{\nu/2}/\|d_k^t\|^\nu.$$

Taking the $\left(\frac{1}{1+\nu}\right)$th root of both sides of the above inequality, and using $6\|d_k^t\| \geq (\epsilon_g/\sigma_t)^{1/2}$ and $\eta \in (0,1)$, we deduce that

$$\begin{aligned}\theta^j &\geq 2(1-\eta)^{1/(1+\nu)}\left(\frac{(\epsilon_g/\sigma_t)^{1/4}}{\|d_k^t\|^{1/2}}\right)^{2\nu/(1+\nu)} \\ &\geq 2(1-\eta)\left(\frac{(\epsilon_g/\sigma_t)^{1/4}}{\|d_k^t\|^{1/2}}\right)^{2\nu/(1+\nu)}\left(\frac{(\epsilon_g/\sigma_t)^{1/4}}{\sqrt{6}\|d_k^t\|^{1/2}}\right)^{(1-\nu)/(1+\nu)} \geq \frac{2(1-\eta)(\epsilon_g/\sigma_t)^{1/4}}{3\|d_k^t\|^{1/2}}.\end{aligned}$$

By this and $\theta \in (0,1)$, one can observe that all $j \geq 0$ that violate (22) must be bounded above. Let j_t be the smallest integer satisfying (22). Then, $j = j_t - 1$ satisfies the above inequality and hence

$$\theta^{j_t} \geq 2(1-\eta)\theta(\epsilon_g/\sigma_t)^{1/4}/(3\|d_k^t\|^{1/2}).$$

It follows that such j_t satisfies both (21) and (22). Hence, the conclusion of this lemma holds. □

The following lemma shows that when d_k^t generated in Algorithm 2 is associated with d_type=NC and $\sigma_t \geq \gamma_\nu(\epsilon)$, Algorithm 2 breaks its inner loop at the inner iteration t.

Lemma 12. *Suppose that Assumption 1 holds and the direction d_k^t results from Algorithm 3 with d_type=NC at some inner iteration t of the kth outer iteration of Algorithm 2. Assume that $\sigma_t \geq \gamma_\nu(\epsilon_g)$ holds, where $\gamma_\nu(\epsilon_g)$ is given in* (9)*. Then, there exists some nonnegative integer j satisfying* (23) *and* (24)*.*

Proof. Since d_k^t is associated with d_type of NC, we observe from Lemma 9(ii) that

$$\nabla f(x^k)^T d_k^t \leq 0, \quad (d_k^t)^T \nabla^2 f(x^k) d_k^t / \|d_k^t\|^2 = -\min\{1, \sigma_t\} \|d_k^t\| \leq -(\sigma_t \epsilon_g)^{1/2}.$$

If (24) holds with $j = 0$, then (23) and (24) hold for $j = 0$ and hence the conclusion of this lemma holds. Now, we suppose that (24) is violated by some $j \geq 0$. Using the same arguments as for (52) with $(d^k, \gamma_\nu(\epsilon_g))$ replaced by (d_k^t, σ_t), we can have that all $j \geq 0$ violating (24) satisfy

$$\theta^{\nu j} > (1 - \eta/2) \min\{1, \sigma_t\}^\nu (\sigma_t \epsilon_g)^{(1-\nu)/2} / H_\nu. \tag{62}$$

Recall that $\sigma_t \geq \gamma_\nu(\epsilon_g)$. Then, letting $\gamma = \sigma_t$ in Lemma 3, and using (38), one has $(\sigma_t \epsilon_g)^{(1-\nu)/2}/H_\nu \geq 2^{1+\nu}/\sigma_t^\nu$, which along with (62) and $\eta \in (0,1)$ implies that all $j \geq 0$ violating (24) satisfy

$$\theta^{\nu j} > (1 - \eta/2) \min\{1, \sigma_t\}^\nu 2^{1+\nu}/\sigma_t^\nu > 2^\nu \min\{1, 1/\sigma_t\}^\nu. \tag{63}$$

When $\nu = 0$, one can see that (63) does not hold for $j = 0$, which implies that (24) holds for $j = 0$. Also, (23) holds for $j = 0$ due to $\theta \in (0,1)$. When $\nu \in (0,1]$, one can see from (63) that all $j \geq 0$ violating (24) must be bounded above. Consequently, there exists some $j \geq 0$ such that (24) holds. Let j_t be the smallest nonnegative integer satisfying (24). Then $j_t = 0$ or (63) holds for $j = j_t - 1$. This together with $\theta \in (0,1)$ implies that $\theta^{j_t} \geq \theta \min\{1, 1/\sigma_t\}$. Hence, both (23) and (24) hold for $j = j_t$. This completes the proof of this lemma. □

We are now ready to prove Theorem 3.

Proof of Theorem 3. We first show that the number of calls of Algorithm 3 at each outer iteration k of Algorithm 2 is at most T, where T is given in (25). To this end, let us consider an arbitrary outer iteration k of Algorithm 2. Clearly, this statement holds if Algorithm 3 is not invoked at the iteration k. Now, suppose that Algorithm 3 is invoked at the iteration k. If Algorithm 2 breaks its inner loop at $t = 0$, then the number of calls of Algorithm 3 is 1, which is clearly bounded above by T. If Algorithm 2 does not break its inner loop at $t = 0$, one can see from Lemmas 10, 11 and 12 that Algorithm 2 must break its inner loop at $t = t_k$ for some $t_k \geq 1$ and $\sigma_{t_k - 1} < \gamma_\nu(\epsilon_g)$. Using, (25), and the fact that $\sigma_{t_k} = r\sigma_{t_k-1}$, we have

$$\sigma_{t_k} = r\sigma_{t_k - 1} < r\gamma_\nu(\epsilon_g) \leq \sigma(\epsilon_g). \tag{64}$$

In addition, notice from Algorithm 2 that $\sigma_{t_k} = r^{t_k}\sigma_0 \geq r^{t_k}\gamma_{-1}$. It then follows that $r^{t_k}\gamma_{-1} \leq \sigma(\epsilon_g)$, which implies that $t_k \leq T - 1$. Hence, the number of calls of Algorithm 3 at the outer iteration k of Algorithm 2 is at most T.

We next show that $\gamma_k \leq \sigma(\epsilon_g)$ for all $k \geq -1$ by induction. Indeed, $\gamma_{-1} \leq \sigma(\epsilon_g)$ holds due to the definition of $\sigma(\epsilon_g)$. Now, suppose that $\gamma_{k-1} \leq \sigma(\epsilon_g)$ holds for some $k \geq 0$. If $\|\nabla f(x^k)\| \leq \epsilon_g$, we see from Algorithm 2 that $\gamma_k = \gamma_{k-1}$ and hence $\gamma_k \leq \sigma(\epsilon_g)$ holds. If $\|\nabla f(x^k)\| > \epsilon_g$ and Algorithm 2 breaks its inner loop at $t = 0$, then $\gamma_k = \sigma_0 = \max\{\gamma_{-1}, \gamma_{k-1}/r\}$, which, together with $r > 1$ and the supposition $\gamma_{k-1} \leq \sigma(\epsilon_g)$, implies that $\gamma_k \leq \sigma(\epsilon_g)$. Otherwise, Algorithm 2 must break its inner loop at $t = t_k$ for some $t_k \geq 1$. By this and (64), one has $\gamma_k = \sigma_{t_k} \leq \sigma(\epsilon_g)$. This completes the induction. Hence, $\gamma_k \leq \sigma(\epsilon_g)$ holds as desired.

We finally show that the total number of calls of Algorithms 3 and 4 during the first s outer iterations of Algorithm 2 is at most $T + 2s$. For convenience, we let τ_k denote the number of calls of Algorithms 3 and 4 in the outer iteration k of Algorithm 2. If $\|\nabla f(x^k)\| \leq \epsilon_g$, then Algorithm 4 is invoked in the outer iteration k, $\gamma_k = \gamma_{k-1}$, and $\tau_k = 1$. Otherwise, Algorithm 3 is invoked in the outer iteration k, and we have either ($\gamma_k = \sigma_0$ and $\tau_k = 1$) or ($\gamma_k = r^{\tau_k - 1}\sigma_0$ and $\tau_k > 1$). By these, $r > 1$ and $\sigma_0 = \max\{\gamma_{-1}, \gamma_{k-1}/r\}$ (see Algorithm 2), one can obtain that $\gamma_k \geq r^{\tau_k - 2}\gamma_{k-1}$ for all $k \geq 0$. It then follows that

$$\sum_{k=0}^{s-1} \tau_k \leq \ln(\gamma_{s-1}/\gamma_{-1})/\ln r + 2s.$$

where together with $\gamma_{s-1} \leq \sigma(\epsilon_g)$ implies that the total number of calls of Algorithms 3 and 4 during the first s outer iterations of Algorithm 2 is at most $T + 2s$. □

The next lemma shows that when the search direction d^k in Algorithm 2 is of type 'SOL', the next iterate x^{k+1} either satisfies $\|\nabla f(x^{k+1})\| \le \epsilon_g$ or produces a sufficient decrease in f.

Lemma 13. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with d_type=SOL at some outer iteration k of Algorithm 2. Let $\hat{c}_{\rm sol}$ be defined in (26). Then, the next iterate $x^{k+1} = x^k + \alpha_k d^k$ satisfies either $\|\nabla f(x^{k+1})\| \le \epsilon_g$ or*

$$f(x^k) - f(x^{k+1}) \ge \hat{c}_{\rm sol}\epsilon_g^{3/2}/\gamma_k^{1/2}. \tag{65}$$

Proof. Since d^k results from Algorithm 3 with d_type=SOL, we observe from Algorithm 2 and Lemma 9(i) that $\|\nabla f(x^k)\| > \epsilon_g$ and (57)-(60) hold with (d_k^t, σ_t) replaced by (d^k, γ_k). In addition, since the next iterate x^{k+1} has already been generated, one can see from Algorithm 3 that at least one of $\|\nabla f(x^{k+1})\| \le \epsilon_g$ and $6\|d^k\| \ge [\epsilon_g/\gamma_k]^{1/2}$ holds. Therefore, to prove this lemma, it suffices to show that (65) holds if $\|\nabla f(x^{k+1})\| > \epsilon_g$. To this end, we suppose that $\|\nabla f(x^{k+1})\| > \epsilon_g$ holds through the remainder of the proof, which implies that $6\|d^k\| \ge [\epsilon_g/\gamma_k]^{1/2}$ holds. We next prove (65) by considering two separate cases below.

Case 1) $\alpha_k = 1$. One can see from (22) with $(d_k^t, \sigma_t, \theta^j) = (d^k, \gamma_k, 1)$ that

$$f(x^k + d^k) \le f(x^k) - \eta(\gamma_k\epsilon_g)^{1/2}\|d^k\|^2.$$

Using this and $6\|d^k\| \ge [\epsilon_g/\gamma_k]^{1/2}$, we obtain that $f(x^k) - f(x^{k+1}) \ge \eta\epsilon_g^{3/2}/(36\gamma_k^{1/2})$, which together with the definition of $\hat{c}_{\rm sol}$ in (26) implies that (65) holds.

Case 2) $\alpha_k < 1$. We can observe from Algorithm 2 that $\alpha_k = \theta^{j_t}$, where j_t satisfies (21) and (22) with $(d_k^t, \sigma_t) = (d^k, \gamma_k)$. It then follows that

$$\alpha_k \ge 2(1-\eta)\theta(\epsilon_g/\gamma_k)^{1/4}/(3\|d^k\|^{1/2}), \quad f(x^k + \alpha_k d^k) \le f(x^k) - \eta(\gamma_k\epsilon_g)^{1/2}\alpha_k^2\|d^k\|^2.$$

Using these inequalities, $x^{k+1} = x^k + \alpha_k d^k$, and $6\|d^k\| \ge [\epsilon_g/\gamma_k]^{1/2}$, we obtain that

$$f(x^k) - f(x^{k+1}) \ge \eta\left(\frac{2(1-\eta)\theta}{3}\right)^2 \epsilon_g\|d^k\| \ge \frac{\eta}{6}\left(\frac{2(1-\eta)\theta}{3}\right)^2 \epsilon_g^{3/2}/\gamma_k^{1/2},$$

which along with the definition of $\hat{c}_{\rm sol}$ in (26) implies that (65) holds. □

Our next lemma shows that when the search direction d^k in Algorithm 2 is of type 'NC', the next iterate x^{k+1} produces a sufficient decrease in f.

Lemma 14. *Suppose that Assumption 1 holds and the direction d^k results from Algorithm 3 with d_type=NC at some outer iteration k of Algorithm 2. Let $c_{\rm nc}$ be defined as in (14). Then, the next iterate $x^{k+1} = x^k + \alpha_k d^k$ satisfies*

$$f(x^k) - f(x^{k+1}) \ge c_{\rm nc}\epsilon_g^{3/2}/\gamma_k^{1/2}. \tag{66}$$

Proof. Since d^k is associated with d_type=NC, we observe from Lemma 9(ii) with $(d_k^t, \sigma_t) = (d^k, \gamma_k)$ that

$$\|d^k\| \ge \max\{1, 1/\gamma_k\}(\gamma_k\epsilon_g)^{1/2}. \tag{67}$$

In addition, we observe from Algorithm 2 that $\alpha_k = \theta^{j_t}$, where j_t satisfies (23) and (24) with $(d_k^t, \sigma_t) = (d^k, \gamma_k)$. By these and (67), one has

$$\begin{aligned} f(x^k) - f(x^{k+1}) &> \frac{\eta}{4}\min\{1,\gamma_k\}\alpha_k^2\|d^k\|^2 \\ &\ge \frac{\eta}{4}\min\{1,\gamma_k\}(\theta\min\{1,1/\gamma_k\})^2(\max\{1,1/\gamma_k\})^3(\gamma_k\epsilon_g)^{3/2} = \frac{\eta\theta^2}{4}\epsilon_g^{3/2}/\gamma_k^{1/2}, \end{aligned}$$

which along with the definition of $c_{\rm nc}$ in (14) implies that (66) holds. □

We are now ready to prove Theorem 4.

Proof of Theorem 4. (i) We first show that the number of outer iterations of Algorithm 2 is at most $\overline{K}_1$. Suppose for contradiction that the number of its outer iterations is more than $\overline{K}_1$. By Lemmas 13 and 14 and the assumption that $\epsilon_H \in (0,1)$ is not provided, we observe that in each outer iteration of Algorithm 2, except for the last one, the function f is reduced at least by $\min\{\hat{c}_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_k^{1/2}$. Using this, (4), and $\gamma_k \le \sigma(\epsilon_g)$ (see Theorem 3), we have

$$\overline{K}_1 \min\{\hat{c}_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\sigma(\epsilon_g)^{1/2} \le \overline{K}_1 \min\{\hat{c}_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_k^{1/2} \le \sum_{k=0}^{\overline{K}_1-1}[f(x^k)-f(x^{k+1})]$$
$$= f(x^0) - f(x^{\overline{K}_1}) \le f(x^0) - f_{\rm low},$$

which contradicts the definition of $\overline{K}_1$ in (27). Hence, the number of outer iterations of Algorithm 2 is at most $\overline{K}_1$.

Recall from above that the number of outer iterations of Algorithm 2 is at most $\overline{K}_1$. Using this and Theorem 3, we see that the total number of calls of Algorithm 3 in Algorithm 2 is at most $T + 2\overline{K}_1$. This along with the fact that Algorithm 3 is called once at each inner iteration of Algorithm 2 implies that the total number of inner iterations of Algorithm 2 is at most $T+2\overline{K}_1$. In addition, relations (28) and (29) follow from (9), (14), (25), (26), and (27). Since ϵ_H is not provided, one can observe that the output x^k of Algorithm 2 satisfies $\|\nabla f(x^k)\| \le \epsilon_g$. This completes the proof of statement (i) of Theorem 4.

(ii) Recall from above that the number of outer iterations of Algorithm 2 is at most $\overline{K}_1$. Suppose that Algorithm 2 terminates at some outer iteration K' with $K' < \overline{K}_1$. Notice from Lemmas 13 and 14 that each outer iteration of Algorithm 2, except for the last one, results in a reduction on f at least by $\min\{\hat{c}_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_k^{1/2}$. Hence,

$$\sum_{k=0}^{K'-1} \min\{\hat{c}_{\rm sol}, c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_k^{1/2} \le \sum_{k=0}^{K'-1}(f(x^k)-f(x^{k+1})) = f(x^0)-f(x^{K'}) \le f(x^0) - f_{\rm low},$$

where the last inequality folllows from (4). Rearranging the terms of this inequality, we obtain that

$$\sum_{k=0}^{K'-1} 1/\gamma_k^{1/2} \le (f(x^0)-f_{\rm low})\epsilon_g^{-3/2}/\min\{\hat{c}_{\rm sol}, c_{\rm nc}\}. \tag{68}$$

In addition, notice that f is descent along the iterates generated by Algorithm 2, which implies $f(x^k) \le f(x^0)$ for all $0 \le k < K'$. It then follows from (4) that $\|\nabla^2 f(x^k)\| \le U_H$ for all $0 \le k < K'$. By Theorem 6 with $(H,\varepsilon) = (\nabla^2 f(x^k), (\sigma_t\epsilon_g)^{1/2})$ and $\|\nabla^2 f(x^k)\| \le U_H$, we obtain that in the kth outer iteration of Algorithm 2, the number of gradient evaluations and Hessian-vector products required by the call of Algorithm 3 at its tth inner iteration is at most

$$\min\left\{n, \lceil (U_H^{1/2}/(\sigma_t\epsilon_g)^{1/4}+2)\psi(U_H/(\sigma_t\epsilon_g)^{1/2})\rceil\right\},$$

where ψ is given in Theorem 6. Further, by $\sigma_t \ge \sigma_0 = \max\{\gamma_{-1}, \gamma_{k-1}/r\}$ and the monotonicity of ψ, one can see that the above quantity is bounded above by

$$\min\left\{n, \lceil (U_H^{1/2}/(\gamma_{k-1}\epsilon_g/r)^{1/4}+2)\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})\rceil\right\}.$$

Let τ_k denote the number of calls of Algorithm 3 (i.e., the number of inner iterations) in the kth outer iteration of Algorithm 2. It follows from statement (i) of Theorem 4 that $\sum_{k=0}^{K'-1}\tau_k \le T + 2\overline{K}_1$, where $\overline{K}_1$ is given in (27). Also, recall from Theorem 3 that $\tau_k \le T$ for all $0 \le k \le K'-1$. Based on these observations, we obtain that the total number of gradient evaluations and Hessian-vector products of f required by the calls of Algorithm 3 in Algorithm 2 is bounded by

$$\sum_{k=0}^{K'-1}\tau_k \min\left\{n, \lceil (U_H^{1/2}/(\gamma_{k-1}\epsilon_g/r)^{1/4}+2)\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})\rceil\right\}$$
$$\le \min\left\{n\sum_{k=0}^{K'-1}\tau_k, \sum_{k=0}^{K'-1}\left([U_H^{1/2}/(\gamma_{k-1}\epsilon_g/r)^{1/4}+2]\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})+1\right)\tau_k\right\}$$

$$= \min\Big\{n\sum_{k=0}^{K'-1}\tau_k, U_H^{1/2}(r/\epsilon_g)^{1/4}\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})\sum_{k=0}^{K'-1}\gamma_{k-1}^{-1/4}\tau_k + 2\big(\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})+1\big)\sum_{k=0}^{K'-1}\tau_k\Big\}$$

$$\le \min\left\{n\sum_{k=0}^{K'-1}\tau_k, U_H^{1/2}(r/\epsilon_g)^{1/4}\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})T\sqrt{\overline{K}_1\sum_{k=0}^{K'-1}\gamma_{k-1}^{-1/2}} + 2\big(\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})+1\big)\sum_{k=0}^{K'-1}\tau_k\right\}$$

$$\le \min\Big\{n(T+2\overline{K}_1), U_H^{1/2}(r/\epsilon_g)^{1/4}\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})T\sqrt{\overline{K}_1\big((f(x^0)-f_{\rm low})\epsilon_g^{-3/2}/\min\{\hat{c}_{\rm sol},c_{\rm nc}\}+\gamma_{-1}^{-1/2}\big)}$$

$$+ 2\big(\psi(U_H/(\gamma_{-1}\epsilon_g)^{1/2})+1\big)(T+2\overline{K}_1)\Big\},$$

$$= \widetilde{\mathcal{O}}\Big(\min\big\{n, U_H^{1/2}/(H_\nu^{1/(2+2\nu)}\epsilon_g^{-\nu/(2+2\nu)})\big\}H_\nu^{1/(1+\nu)}\epsilon_g^{-(2+\nu)/(1+\nu)}\Big),$$

where the first inequality is due to $\min\{a_1,b_1\}+\min\{a_2,b_2\}\le\min\{a_1+a_2,b_1+b_2\}$ for all $a_1,a_2,b_1,b_2\in\mathbb{R}$, the second inequality follows from Cauchy-Schwarz inequality and $(\sum_{k=0}^{K'-1}\tau_k^2)^{1/2}\le T(K')^{1/2}<T\overline{K}_1^{1/2}$ because $\tau_k\le T$ for all $0\le k\le K'-1$ and $K'<\overline{K}_1$, the last inequality is due to $\sum_{k=0}^{K'-1}\tau_k\le T+2\overline{K}_1$ and (68), and the last equality follows from (28) and (29) and the definition of ψ in Theorem 6. Hence, statement (ii) of Theorem 4 holds. □

The following lemma shows that when the search direction d^k in Algorithm 2 is a negative curvature direction returned from Algorithm 4, the next iterate x^{k+1} produces a sufficient reduction on f. Its proof is identical to that of Lemma 8, and thus is omitted here.

Lemma 15. *Suppose that Assumption 1 holds with $H_\nu>0$ and $\nu\in(0,1]$, and d^k results from Algorithm 4 at some outer iteration k of Algorithm 2. Let $c_{\rm meo}$ be defined in (17). Then the following statements hold.*

(i) *The step length α_k is well defined, and $\alpha_k\ge\min\big\{1,\theta((1-\eta)/H_\nu)^{1/\nu}(\epsilon_H/2)^{(1-\nu)/\nu}\big\}$.*

(ii) *The next iterate $x^{k+1}=x^k+\alpha_kd^k$ satisfies $f(x^k)-f(x^{k+1})\ge c_{\rm meo}\epsilon_H^{(2+\nu)/\nu}$.*

We are now ready to prove Theorem 5.

Proof of Theorem 5. (i) Let K_2 and $\overline{K}_1$ be defined in (18) and (27), respectively. Observe from Algorithm 2 and Lemma 15(ii) that each call of Algorithm 4, except the last one, results in a reduction on f at least by $c_{\rm meo}\epsilon_H^{(2+\nu)/\nu}$. By this and similar arguments as used in the proof of Theorem 2(i), one can claim that the total number of calls of Algorithm 4 in Algorithm 2 is at most K_2.

In addition, we claim that the total number of calls of Algorithm 3 in Algorithm 2 is at most $\overline{K}_1+K_2-1$. Indeed, suppose for contradiction that its total number of calls is more than $\overline{K}_1+K_2-1$. Notice that if Algorithm 3 is called at some iteration k and generates x^{k+1} satisfying $\|\nabla f(x^{k+1})\|\le\epsilon_g$, then Algorithm 4 must be called at the iteration $k+1$. In view of this and the fact that the total number of calls of Algorithm 4 is at most K_2, one can observe that the total number of such iterations is at most K_2. This along with the above supposition implies that the total number of iterations k of Algorithm 2 at which Algorithm 3 is called and generates the next iterate x^{k+1} satisfying $\|\nabla f(x^{k+1})\|>\epsilon_g$ is at least $\overline{K}_1$. For each of these iterations k, we observe from Lemmas 6(ii) and 7(ii) and Theorem 3 that

$$f(x^k)-f(x^{k+1})\ge\min\{\hat{c}_{\rm sol},c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_k^{1/2}\ge\min\{\hat{c}_{\rm sol},c_{\rm nc}\}\epsilon_g^{3/2}/\sigma(\epsilon_g)^{1/2}.$$

Since f is descent along the iterates of Algorithm 2 and $f(x^k)\ge f_{\rm low}$, the total amount of reduction on f resulting from these iterations k is at most $f(x^0)-f_{\rm low}$. It then follows that

$$\overline{K}_1\min\{c_{\rm sol},c_{\rm nc}\}\epsilon_g^{3/2}/\gamma_\nu(\epsilon_g)^{1/2}\le f(x^0)-f_{\rm low},$$

which contradicts the definition of $\overline{K}_1$ given in (27). Hence, the total number of calls of Algorithm 3 in Algorithm 2 is at most $\overline{K}_1+K_2-1$.

Based on the above claims and the fact that either Algorithm 3 or 4 is called at each iteration of Algorithm 2, we conclude that the total number of iterations of Algorithm 2 is at most $\overline{K}_1+2K_2-1$. Using this and

Theorem 3, we see that the total number of calls of Algorithm 3 in Algorithm 2 is at most $T + 2\overline{K}_1 + 4K_2 - 2$. This along with the fact that Algorithm 3 is called once at each inner iteration of Algorithm 2 implies that the total number of inner iterations of Algorithm 2 is at most $T + 2\overline{K}_1 + 4K_2 - 2$. In addition, relations (30) and (31) follow from (9), (14), (17), (18), (25), (26), and (27). Moreover, one can observe that the output x^k of Algorithm 2 satisfies $\|\nabla f(x^k)\| \le \epsilon_g$ deterministically and $\lambda_{\min}(\nabla^2 f(x^k)) \ge -\epsilon_H$ with probability at least $1-\delta$ for some $0 \le k \le \overline{K}_1 + 2K_2 - 1$, where the latter part is due to Algorithm 4. This completes the proof of statement (i) of Theorem 5.

(ii) Notice that f is descent along the iterates generated by Algorithm 2, which implies $f(x^k) \le f(x^0)$ for each iteration k. It then follows from (4) that $\|\nabla^2 f(x^k)\| \le U_H$ for each iteration k. Recall from the above proof that the total number of iterations of Algorithm 2 is at most $\overline{K}_1 + 2K_2 - 1$ and the total number of calls of Algorithm 3 in Algorithm 2 is at most $T + 2\overline{K}_1 + 4K_2 - 2$. In view of Theorem 6 with $(H, \varepsilon) = (\nabla^2 f(x^k), (\sigma_t \epsilon_g)^{1/2})$ and the fact that $\|\nabla^2 f(x^k)\| \le U_H$ and $\sigma_t \ge \gamma_{-1}$, one can observe that the number of gradient evaluations and Hessian-vector products of f required by each call of Algorithm 3 with input $U = 0$ is at most $\widetilde{\mathcal{O}}(\min\{n, U_H^{1/2}/\epsilon_g^{1/4}\})$. Using these, we obtain that the total number of gradient evaluations and Hessian-vector products of f required by the calls of Algorithm 3 in Algorithm 2 is bounded by

$$\widetilde{\mathcal{O}}((T + \overline{K}_1 + K_2) \min\{n, U_H^{1/2}/\epsilon_g^{1/4}\}). \tag{69}$$

In addition, by Theorem 7 with $(H, \varepsilon) = (\nabla^2 f(x^k), \epsilon_H)$, $\|\nabla^2 f(x^k)\| \le U_H$ and the fact that each iteration of the Lanczos method requires only one Hessian-vector product of f, one can observe that the number of Hessian-vector products required by each call of Algorithm 4 in Algorithm 2 is at most $\widetilde{\mathcal{O}}(\min\{n, (U_H/\epsilon_H)^{1/2}\})$. Recall from the above proof that the total number of calls of Algorithm 4 in Algorithm 2 is at most K_2. Hence, the total number of Hessian-vector products required by all calls of Algorithm 4 in Algorithm 2 is at most $\widetilde{\mathcal{O}}(K_2 \min\{n, (U_H/\epsilon_H)^{1/2}\})$. Using this, (31), and (69), we see that statement (ii) of Theorem 5 holds. □

Acknowledgments

The first author was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The second author was partially supported by National Science Foundation Award IIS-2347592. The third author was partially supported by the National Science Foundation Award IIS-2211491, the Office of Naval Research Award N00014-24-1-2702, and the Air Force Office of Scientific Research Award FA9550-24-1-0343.

References

[1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199, 2017.

[2] Z. Allen-Zhu and Y. Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31, 2018.

[3] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[4] E. G. Birgin and J. M. Martínez. The use of quadratic regularization with a cubic descent condition for unconstrained optimization. *SIAM J. Optim.*, 27(2):1049–1074, 2017.

[5] R. H. Byrd, F. E. Curtis, and J. Nocedal. Infeasibility detection and SQP methods for nonlinear optimization. *SIAM J. Optim.*, 20(5):2281–2299, 2010.

[6] Y. Carmon and J. Duchi. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM J. Optim.*, 29(3):2146–2178, 2019.

[7] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *International conference on machine learning*, pages 654–663, 2017.

[8] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM J. Optim.*, 28(2):1751–1772, 2018.

[9] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points I. *Math. Program.*, 184(1-2):71–120, 2020.

[10] 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(2):245–295, 2011.

[11] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130(2):295–319, 2011.

[12] C. Cartis, N. I. Gould, and P. L. Toint. Optimal Newton-type methods for nonconvex smooth optimization problems. *Technical Report Optimi. Online*, 2011.

[13] C. Cartis, N. I. Gould, and P. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3711–3750. World Scientific, 2018.

[14] C. Cartis, N. I. Gould, and P. L. Toint. Universal regularization methods: varying the power, the smoothness and the accuracy. *SIAM J. Optimi.*, 29(1):595–615, 2019.

[15] C. Cartis, N. I. Gould, and P. L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM J. Optimi.*, 30(1):513–541, 2020.

[16] F. E. Curtis, D. P. Robinson, C. W. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM J. Optim.*, 31(1):518–544, 2021.

[17] F. E. Curtis, D. P. Robinson, and M. Samadi. A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *Math. Program.*, 162:1–32, 2017.

[18] F. E. Curtis, D. P. Robinson, and M. Samadi. An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *IMA J. Numer. Anal.*, 39(3):1296–1327, 2019.

[19] P. Dvurechensky. Gradient method with inexact oracle for composite non-convex optimization. *arXiv preprint arXiv:1703.09180*, 2017.

[20] G. N. Grapiglia and Y. Nesterov. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM J. Optim.*, 27(1):478–506, 2017.

[21] C. He, Z. Lu, and T. K. Pong. A Newton-CG based augmented Lagrangian method for finding a second-order stationary point of nonconvex equality constrained optimization with complexity guarantees. *SIAM J. Optim.*, 33(3):1734–1766, 2023.

[22] M. Ito, Z. Lu, and C. He. A parameter-free conditional gradient method for composite minimization under Hölder condition. *J. Mach. Learn. Res.*, 24:1–34, 2023.

[23] C. Jin, P. Netrapalli, and M. I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pages 1042–1085, 2018.

[24] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13(4):1094–1122, 1992.

[25] B. Li, S. Tang, and H. Yu. Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. *arXiv preprint arXiv:1903.05858*, 2019.

[26] H. Li and Z. Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $\mathcal{O}(\epsilon^{-7/4})$ complexity. In *International Conference on Machine Learning*, pages 12901–12916, 2022.

[27] J. M. Martínez and M. Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *J. Glob. Optim.*, 68:367–385, 2017.

[28] N. Marumo and A. Takeda. Parameter-free accelerated gradient descent for nonconvex minimization. *SIAM J. Optimi.*, 34(2):2093–2120, 2024.

[29] Y. Nesterov. Universal gradient methods for convex optimization problems. *Math. Program.*, 152(1-2):381–404, 2015.

[30] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1):177–205, 2006.

[31] C. W. Royer, M. O'Neill, and S. J. Wright. A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Math. Program.*, 180(1-2):451–488, 2020.

[32] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM J. Optim.*, 28(2):1448–1477, 2018.

[33] Y. Xu, R. Jin, and T. Yang. Neon+: Accelerated gradient methods for extracting negative curvature for non-convex optimization. *arXiv preprint arXiv:1712.01033*, 2017.

[34] C. Zhang and R. Jiang. Riemannian adaptive regularized Newton methods with Hölder continuous Hessians. *arXiv preprint arXiv:2309.04052*, 2023.

Appendix

A A capped conjugate gradient method

We present a capped CG method in Algorithm 3, which was proposed in [31, Algorithm 1] for finding either an approximate solution to the linear system (7) or a sufficiently negative curvature direction of the associated coefficient matrix. Its details can be found in [31, Section 3.1].

The following lemma present some useful properties of Algorithm 3 below, which are adopted from [31, Lemma 3].

Lemma 16. *Consider applying Algorithm 3 with input $U = 0$ to the linear system (7) with $g \neq 0$, $\varepsilon > 0$, and H being an $n \times n$ symmetric matrix. Let d be the output of Algorithm 3 with a type specified in d_type. Then the following statements hold.*

(i) *If d_type=SOL, then d satisfies*

$$\varepsilon\|d\|^2 \leq d^T(H + 2\varepsilon I)d, \qquad \|d\| \leq 1.1\varepsilon^{-1}\|g\|,$$
$$d^T g = -d^T(H + 2\varepsilon I)d, \qquad \|(H + 2\varepsilon I)d + g\| \leq \zeta\varepsilon\|d\|/2.$$

(ii) *If d_type=NC, then d satisfies $d^T g \leq 0$ and $d^T H d/\|d\|^2 \leq -\varepsilon$.*

The following theorem presents the iteration complexity of Algorithm 3.

Theorem 6 (iteration complexity of Algorithm 3). *Consider applying Algorithm 3 with input $U = 0$ to the linear system (7) with $g \neq 0$, $\varepsilon > 0$, and H being an $n \times n$ symmetric matrix. Then the number of iterations of Algorithm 3 is at most*

$$\min\left\{n, \left\lceil \left((\|H\|/\varepsilon)^{1/2} + 2\right)\psi(\|H\|/\varepsilon) \right\rceil \right\} = \widetilde{\mathcal{O}}\left(\min\left\{n, (\|H\|/\varepsilon)^{1/2}\right\}\right),$$

where $\psi(t) = \ln(144((t + 2)^{1/2} + 1)^2(t + 2)^6/\zeta^2)$.

Algorithm 3 A capped conjugate gradient method

Inputs: symmetric matrix $H \in \mathbb{R}^{n\times n}$, vector $g \neq 0$, damping parameter $\varepsilon > 0$, desired relative accuracy $\zeta \in (0,1)$.
Optional input: scalar $U \geq 0$ (set to 0 if not provided).
Outputs: d_type, d.
Secondary outputs: final values of $U, \kappa, \hat{\zeta}, \tau$, and T.
Set

$$\bar{H} := H + 2\varepsilon I, \quad \kappa := \frac{U+2\varepsilon}{\varepsilon}, \quad \hat{\zeta} := \frac{\zeta}{3\kappa}, \quad \tau := \frac{\sqrt{\kappa}}{\sqrt{\kappa}+1}, \quad T := \frac{4\kappa^4}{(1-\sqrt{\tau})^2},$$

$y^0 \leftarrow 0, r^0 \leftarrow g, p^0 \leftarrow -g, j \leftarrow 0.$
if $(p^0)^T \bar{H} p^0 < \varepsilon \|p^0\|^2$ **then**
  Set $d \leftarrow p^0$ and terminate with d_type = NC;
else if $\|Hp^0\| > U\|p^0\|$ **then**
  Set $U \leftarrow \|Hp^0\|/\|p^0\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
end if
while TRUE **do**
  $\alpha_j \leftarrow (r^j)^T r^j / (p^j)^T \bar{H} p^j$; {Begin Standard CG Operations}
  $y^{j+1} \leftarrow y^j + \alpha_j p^j$;
  $r^{j+1} \leftarrow r^j + \alpha_j \bar{H} p^j$;
  $\beta_{j+1} \leftarrow \|r^{j+1}\|^2 / \|r^j\|^2$;
  $p^{j+1} \leftarrow -r^{j+1} + \beta_{j+1} p^j$; {End Standard CG Operations}
  $j \leftarrow j+1$;
  if $\|Hp^j\| > U\|p^j\|$ **then**
    Set $U \leftarrow \|Hp^j\|/\|p^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $\|Hy^j\| > U\|y^j\|$ **then**
    Set $U \leftarrow \|Hy^j\|/\|y^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $\|Hr^j\| > U\|r^j\|$ **then**
    Set $U \leftarrow \|Hr^j\|/\|r^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $(y^j)^T \bar{H} y^j < \varepsilon \|y^j\|^2$ **then**
    Set $d \leftarrow y^j$ and terminate with d_type = NC;
  else if $\|r^j\| \leq \hat{\zeta}\|r^0\|$ **then**
    Set $d \leftarrow y^j$ and terminate with d_type = SOL;
  else if $(p^j)^T \bar{H} p^j < \varepsilon \|p^j\|^2$ **then**
    Set $d \leftarrow p^j$ and terminate with d_type = NC;
  else if $\|r^j\| > \sqrt{T}\tau^{j/2}\|r^0\|$ **then**
    Compute α_j, y^{j+1} as in the main loop above;
    Find $i \in \{0, \ldots, j-1\}$ such that

$$(y^{j+1} - y^i)^T \bar{H} (y^{j+1} - y^i) < \varepsilon \|y^{j+1} - y^i\|^2;$$

    Set $d \leftarrow y^{j+1} - y^i$ and terminate with d_type = NC;
  end if
end while

Proof. From [31, Lemma 1], we know that the number of iterations of Algorithm 3 is bounded by $\min\{n, J(U,\varepsilon,\zeta)\}$, where $J(U,\varepsilon,\zeta)$ is the smallest integer J such that $\sqrt{T}\tau^{J/2} \leq \hat{\zeta}$, with $U, \hat{\zeta}, T$ and τ being the values returned by Algorithm 3. In addition, it was shown in [31, Section 3.1] that $J(U,\varepsilon,\zeta) \leq \lceil(\sqrt{\kappa}+1/2)\ln(144(\sqrt{\kappa}+1)^2\kappa^6/\zeta^2)\rceil$, where $\kappa = U/\varepsilon + 2$ is an output by Algorithm 3. Also, we can observe that $\sqrt{\kappa} \leq (U/\varepsilon)^{1/2} + \sqrt{2} \leq (U/\varepsilon)^{1/2} + 3/2$. Combining these, we obtain that $J(U,\varepsilon,\zeta) \leq \lceil[(U/\varepsilon)^{1/2}+2]\ln(144((U/\varepsilon+2)^{1/2}+1)^2(U/\varepsilon+2)^6/\zeta^2)\rceil$. Notice from Algorithm 3 that the output $U \leq \|H\|$. Using these, we obtain the conclusion as desired. □

B A randomized Lanczos based minimum eigenvalue oracle

In this part we present the randomized Lanczos method proposed in [31, Section 3.2], which can be used as a minimum eigenvalue oracle for Algorithms 1 and 2. As briefly discussed in Section 3, this oracle outputs either a sufficiently negative curvature direction of H or a certificate that H is nearly positive semidefinite with high probability. More detailed motivation and explanation of it can be found in [31, Section 3.2].

The following theorem justifies that Algorithm 4 is a suitable minimum eigenvalue oracle for Algorithms 1

Algorithm 4 A randomized Lanczos based minimum eigenvalue oracle

Input: symmetric matrix $H \in \mathbb{R}^{n\times n}$, tolerance $\varepsilon > 0$, and probability parameter $\delta \in (0,1)$.
Output: a sufficiently negative curvature direction v satisfying $v^T Hv \le -\varepsilon/2$ and $\|v\| = 1$; or a certificate that $\lambda_{\min}(H) \ge -\varepsilon$ with probability at least $1-\delta$.
Apply the Lanczos method [24] to estimate $\lambda_{\min}(H)$ starting with a random vector uniformly generated on the unit sphere, and run it for at most

$$N(\varepsilon,\delta) := \min\left\{n, 1 + \left\lceil \frac{\ln(2.75n/\delta^2)}{2}\sqrt{\frac{\|H\|}{\varepsilon}}\right\rceil\right\} \tag{70}$$

iterations. If a unit vector v with $v^T Hv \le -\varepsilon/2$ is found at some iteration, terminate immediately and return v.

and 2. Its proof is identical to that of [31, Lemma 2] and thus omitted.

Theorem 7 (iteration complexity of Algorithm 4). *Consider Algorithm 4 with tolerance $\varepsilon > 0$, probability parameter $\delta \in (0,1)$, and symmetric matrix $H \in \mathbb{R}^{n\times n}$ as its input. Then it either finds a sufficiently negative curvature direction v satisfying $v^T Hv \le -\varepsilon/2$ and $\|v\| = 1$ or certifies that $\lambda_{\min}(H) \ge -\varepsilon$ holds with probability at least $1-\delta$ in at most $N(\varepsilon,\delta)$ iterations, where $N(\varepsilon,\delta)$ is defined in* (70).

Notice that $\|H\|$ is required in Algorithm 4. In general, computing $\|H\|$ may not be cheap when n is large. Nevertheless, $\|H\|$ can be efficiently estimated via a randomization scheme with high confidence (e.g., see the discussion in [31, Appendix B3]).