Mathematical Programming
https://doi.org/10.1007/s10107-019-01362-7

FULL LENGTH PAPER

Series A

A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization

Clément W. Royer[1] · Michael O'Neill[2] · Stephen J. Wright[2]

Received: 9 March 2018 / Accepted: 5 January 2019

Abstract

We consider minimization of a smooth nonconvex objective function using an iterative algorithm based on Newton's method and the linear conjugate gradient algorithm, with explicit detection and use of negative curvature directions for the Hessian of the objective function. The algorithm tracks Newton-conjugate gradient procedures developed in the 1980s closely, but includes enhancements that allow worst-case complexity results to be proved for convergence to points that satisfy approximate first-order and second-order optimality conditions. The complexity results match the best known results in the literature for second-order methods.

Keywords Smooth nonconvex optimization · Newton's method · Conjugate gradient method · Optimality conditions · Worst-case complexity

Mathematics Subject Classification 49M05 · 49M15 · 65F10 · 65F15 · 90C06 · 90C60

Research supported by NSF Awards IIS-1447449, 1628384, 1634597, and 1740707; AFOSR Award FA9550-13-1-0138; Subcontracts 3F-30222 and 8F-30039 from Argonne National Laboratory; and Award N660011824020 from the DARPA Lagrange Program.

✉ Stephen J. Wright
swright@cs.wisc.edu

Clément W. Royer
croyer2@wisc.edu

Michael O'Neill
moneill@cs.wisc.edu

[1] Wisconsin Institute of Discovery, University of Wisconsin, 330 N. Orchard St., Madison, WI 53715, USA

[2] Computer Sciences Department, University of Wisconsin, 1210 W. Dayton St., Madison, WI 53706, USA

Published online: 19 January 2019

1 Introduction

We consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice Lipschitz continuously differentiable function that is nonconvex in general. We further assume that f is bounded below for all x, by some constant f_{low}. Although the Hessian $\nabla^2 f(x)$ is well defined for such functions, we assume that full evaluation of this matrix is undesirable from a computational viewpoint, though we assume that Hessian-vector products of the form $\nabla^2 f(x)v$ can be computed with reasonable efficiency, for arbitrary vectors v, as is often the case when n is large.

Unconstrained minimization of nonconvex smooth functions of many variables is a much-studied paradigm in optimization. Approaches such as limited-memory BFGS and nonlinear conjugate gradient are widely used to tackle (1), particularly in the case of large dimension n. Another popular approach, known as "Newton-CG," applies the (linear) conjugate gradient (CG) method to the second-order Taylor-series approximation of f around the current iterate x_k. Each iteration of CG requires computation of one Hessian-vector product of the form $\nabla^2 f(x_k)v$. A trust-region variant of Newton-CG, due to Steihaug [27], terminates the CG iterations when sufficient accuracy is achieved in minimizing the quadratic approximation, when a CG step leads outside the trust region, or when negative curvature is encountered in $\nabla^2 f(x_k)$. A line-search variant presented in [25] applies CG until some convergence criterion is satisfied, or until negative curvature is encountered, in which case the search direction reverts to the negative gradient.

Theoretical guarantees for Newton-CG algorithms have been provided, e.g. in [12,15–17,25]. Convergence analysis for such methods typically shows that accumulation points are stationary, that is, they satisfy the first-order optimality condition $\nabla f(x) = 0$. Local linear or superlinear convergence to a point satisfying second-order sufficient conditions is sometimes also proved for Newton-CG methods. Although several Newton-type methods have been analyzed from a global complexity perspective [8], particularly in terms of outer iterations and derivative evaluations, bounds that explicitly account for the use of inexact Newton-CG techniques have received less attention in the optimization literature. Meanwhile, with the recent upsurge of interest in complexity, several new algorithms have been proposed that have good global complexity guarantees. We review some such contributions in Sect. 2. In most cases, these new methods depart significantly from those seen in the traditional optimization literature, and there are questions surrounding their practical appeal.

Our aim in this paper is to develop a method that hews closely to the Newton-CG approach, but which comes equipped with certain safeguards and enhancements that allow worst-case complexity results to be proved. At each iteration, we use CG to solve a slightly damped version of the Newton equations, monitoring the CG iterations for evidence of indefiniteness in the Hessian. If the CG process terminates with an approximation to the Newton step, we perform a backtracking line search along this

direction. Otherwise, we step along a negative curvature direction for the Hessian, obtained either from the CG procedure on the Newton equations, or via some auxiliary computation (possibly another CG process). In either case, we can show that significant decrease can be attained in f at each iteration, at reasonable computational cost (in terms of the number of gradient evaluations or Hessian-vector products), allowing worst-case complexity results to be proved.

The remainder of the paper is organized as follows. We position our work within the existing literature in Sect. 2. Our algorithm is described in Sect. 3. Section 4 contains the complexity analysis, showing both a deterministic upper bound on the computation required to attain approximate first-order conditions (Sect. 4.2) and a high-probability upper bound on the computation required to satisfy approximate second-order necessary conditions (Sect. 4.3). Section 5 contains some conclusions and discussion. Several technical results and proofs related to CG are gathered in the Appendix.

Assumptions, Background, Notation Our algorithm seeks a point that approximately satisfies second-order necessary conditions for optimality, that is,

$$\|\nabla f(x)\| \leq \epsilon_g, \quad \lambda_{\min}(\nabla^2 f(x)) \geq -\epsilon_H, \tag{2}$$

for specified small positive tolerances ϵ_g and ϵ_H. (Here and subsequently, $\|\cdot\|$ denotes the Euclidean norm, or its induced norms on matrices.) We make the following standard assumptions throughout.

Assumption 1 The level set $\mathcal{L}_f(x_0) = \{x | f(x) \leq f(x_0)\}$ is compact.

Assumption 2 The function f is twice uniformly Lipschitz continuously differentiable on an open neighborhood of $\mathcal{L}_f(x_0)$ that includes the trial points generated by the algorithm. We denote by L_H the Lipschitz constant for $\nabla^2 f$ on this neighborhood.

Note that Assumption 2 is made for simplicity of exposition; slightly weaker variants could be used at the expense of some complication in the analysis.

Under these two assumptions, there exist scalars f_{low}, $U_g > 0$, and $U_H > 0$ such that the following are satisfied for $x \in \mathcal{L}_f(x_0)$:

$$f(x) \geq f_{\text{low}}, \quad \|\nabla f(x)\| \leq U_g, \quad \|\nabla^2 f(x)\| \leq U_H. \tag{3}$$

We observe that U_H is a Lipschitz constant for the gradient.

For any x and d such that Assumption 2 is satisfied at x and $x + d$, we have

$$f(x+d) \leq f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x) d + \frac{L_H}{6}\|d\|^3. \tag{4}$$

Notationally, we use order notation $\mathcal{O}$ in the usual sense, whereas $\tilde{\mathcal{O}}(\cdot)$ represents $\mathcal{O}$ with logarithmic terms omitted. We use such notation in bounding iteration count and computational effort, and focus on the dependencies of such complexities on ϵ_g and ϵ_H. (In one of our final results—Corollary 2—we also show explicitly the dependence on n and U_H.)

2 Complexity in nonconvex optimization

In recent years, many algorithms have been proposed for finding a point that satisfies conditions (2), with iteration complexity and computational complexity bounded in terms of ϵ_g and ϵ_H. We review several works most relevant to this paper here, and relate their results to our contributions. For purposes of computational complexity, we define the unit of computation to be one Hessian-vector product *or* one gradient evaluation, implicitly using the observation from computational/algorithmic differentiation [18] that these two operations differ in cost only by a modest factor, independent of the dimension n.

Classical second-order convergent trust-region schemes [12] can be shown to satisfy (2) after at most $\mathcal{O}(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\})$ iterations [10]. For the class of second-order algorithms (that is, algorithms which rely on second-order derivatives and Newton-type steps) the best known iteration bound is $\mathcal{O}(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\})$. This bound was first established for a form of cubic regularization of Newton's method [24]. Following this paper, numerous other algorithms have also been proposed which match this bound, see for example [3,8,13,14,23].

A recent trend in complexity analysis of these methods also accounts for the computational cost of each iteration, thus yielding a bound on the computational complexity. Two independent proposals, respectively based on adapting accelerated gradient to the nonconvex setting [6] and approximately solving the cubic subproblem [1], require $\tilde{\mathcal{O}}(\epsilon^{-7/4})$ operations (with high probability, showing dependency only on ϵ) to find a point x that satisfies

$$\|\nabla f(x)\| \leq \epsilon \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x)) \geq -\sqrt{U_H \epsilon}. \tag{5}$$

The difference of a factor of $\epsilon^{-1/4}$ with the results presented above arises from the cost of computing a negative curvature direction of $\nabla^2 f(x_k)$ and/or the cost of solving a linear system. The probabilistic nature of the bound is generally due to the introduction of randomness in the curvature estimation process; see [2,28] for two recent examples. A complexity bound of the same type was also established for a variant of accelerated gradient free of negative curvature computation, that regularly adds a random perturbation to the iterate when the gradient norm is small [19].

In an interesting followup to [6], an algorithm based on accelerated gradient with a nonconvexity monitor was proposed [5]. It requires at most $\tilde{\mathcal{O}}(\epsilon^{-7/4})$ iterations to satisfy (5) with high probability. However, if one is concerned only with satisfying the gradient condition $\|\nabla f(x)\| \leq \epsilon$, the $\tilde{\mathcal{O}}(\epsilon^{-7/4})$ bound holds deterministically. Note that this bound represents an improvement over the $\mathcal{O}(\epsilon^{-2})$ of steepest descent and classical Newton's method [7]. The improvement is due to a modification of the accelerated gradient paradigm that allows for deterministic detection and exploitation of negative curvature directions in regions of sufficient nonconvexity.

In a previous work [26], two authors of the current paper proposed a Newton-based algorithm in a line-search framework which has an iteration complexity of $\mathcal{O}(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\})$ when the subproblems are solved exactly, and a computational complexity of $\tilde{\mathcal{O}}\left(\epsilon^{-7/4}\right)$ Hessian-vector products and/or gradient evaluations, when the subproblems are solved inexactly using CG and the randomized Lanczos algorithm.

Compared to the accelerated gradient methods, this approach aligns more closely with traditional optimization practice, as described in Sect. 1.

Building on [26], the current paper describes a similar line-search framework with inexact Newton steps, but uses a modified version of CG to solve the system of Newton equations, without first checking for positive definiteness of the coefficient matrix. The modification is based in part on a convexity monitoring device introduced in the accelerated gradient algorithms mentioned above. An implicit cap is imposed on the number of CG iterations that are used to solve the damped Newton system. We show that once this cap is reached, either the damped Newton system has been solved to sufficient accuracy or else a direction of "sufficiently negative curvature" has been identified for the Hessian. (A single extra CG iteration may be needed to identify the negative curvature direction, in much the same manner as in [5] for accelerated gradient.) In contrast to the previous work [26], no estimate of the smallest eigenvalue of the Hessian is required prior to computing a Newton step. In addition to removing potentially unnecessary computation, this approach allows a deterministic result for first-order optimality to be proved, as in [5].

We are deliberate in our use of CG rather than accelerated gradient as the method of choice for minimizing the quadratic objective that arises in the damped Newton step. When applied to strongly convex quadratics, both approaches have similar asymptotic linear convergence rates that depend only on the extreme eigenvalues of the Hessian, and both can be analyzed using the same potential function [20] and viewed as two instances of an underlying "idealized algorithm" [21]. However, CG has several advantages: It has a rich convergence theory that depends on the full spectrum of eigenvalues; it is fully adaptive, requiring no prior estimates of the extreme eigenvalues; and its practical performance on convex quadratics is superior. (See, for example, [25, Chapter 5] for a description of these properties.) Further, as we prove in this paper (Sect. 3.1), CG can be adapted to detect nonconvexity efficiently in a quadratic function, without altering its core properties. We show in addition (see Sect. 3.2 and Appendix B) that by applying CG to a linear system with a random right-hand side, we can find a direction of negative curvature in an indefinite matrix efficiently, with the same iteration complexity as the randomized Lanczos process of [22] used elsewhere.

The practical benefits of CG in large-scale optimization have long been appreciated. We establish here that with suitable enhancements, methods based on CG can also be equipped with good complexity properties as well.

3 Damped-Newton/Capped-CG method with negative curvature steps

We describe our algorithm in this section, starting with its two major components. The first component, described in Sect. 3.1, is a linear conjugate gradient procedure that is used to solve a slightly damped Newton system. This procedure includes enhancements to detect indefiniteness in the Hessian and to restrict the number of iterations. Because of this implicit bound on the number of iterations, we refer to it as "Capped CG." The second component (see Sect. 3.2) is a "minimum eigenvalue oracle" that seeks a

direction of negative curvature for a symmetric matrix, along with its corresponding vector. The main algorithm is described in Sect. 3.3.

Algorithm 1 Capped Conjugate Gradient

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

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

$y_0 \leftarrow 0, r_0 \leftarrow g, p_0 \leftarrow -g, j \leftarrow 0;$
if $p_0^\top \bar{H} p_0 < \epsilon \|p_0\|^2$ **then**
  Set $d = p_0$ and terminate with d_type=NC;
else if $\|Hp_0\| > M\|p_0\|$ **then**
  Set $M \leftarrow \|Hp_0\|/\|p_0\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
end if
while TRUE **do**
  $\alpha_j \leftarrow r_j^\top r_j / p_j^\top \bar{H} p_j$; {Begin Standard CG Operations}
  $y_{j+1} \leftarrow y_j + \alpha_j p_j$;
  $r_{j+1} \leftarrow r_j + \alpha_j \bar{H} p_j$;
  $\beta_{j+1} \leftarrow (r_{j+1}^\top r_{j+1})/(r_j^\top r_j)$;
  $p_{j+1} \leftarrow -r_{j+1} + \beta_{j+1} p_j$; {End Standard CG Operations}
  $j \leftarrow j+1$;
  if $\|Hp_j\| > M\|p_j\|$ **then**
    Set $M \leftarrow \|Hp_j\|/\|p_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $\|Hy_j\| > M\|y_j\|$ **then**
    Set $M \leftarrow \|Hy_j\|/\|y_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $\|Hr_j\| > M\|r_j\|$ **then**
    Set $M \leftarrow \|Hr_j\|/\|r_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;
  end if
  if $y_j^\top \bar{H} y_j < \epsilon \|y_j\|^2$ **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 $p_j^\top \bar{H} p_j < \epsilon \|p_j\|^2$ **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

$$\frac{(y_{j+1}-y_i)^\top \bar{H}(y_{j+1}-y_i)}{\|y_{j+1}-y_i\|^2} < \epsilon; \tag{6}$$

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

3.1 Capped conjugate gradient

Conjugate Gradient (CG) is a widely used technique for solving linear equations with symmetric positive definite coefficient matrices or, equivalently, minimizing strongly convex quadratic functions. We devise a modified CG algorithm and apply it to a system of the form $\bar{H}y = -g$, where $\bar{H} = H + 2\epsilon I$ is a damped version of the symmetric matrix H, which is our notational proxy for the Hessian $\nabla^2 f(x_k)$.

Algorithm 1 presents our Capped CG procedure. The main loop consists of classical CG iterations. When $\bar{H} \succeq \epsilon I$, Algorithm 1 will generate the same iterates as a classical conjugate gradient method applied to $\bar{H}y = -g$, and terminate at an inexact solution of this linear system. When $\bar{H} \not\succeq \epsilon I$, the features added to Algorithm 1 cause a direction d to be identified along which $d^\top \bar{H} d < \epsilon \|d\|^2$ or, equivalently, $d^\top H d < -\epsilon \|d\|^2$—a direction of "sufficiently negative curvature" for H. Directions d of this type are encountered most obviously when they arise as iterates y_j or search directions p_j in the CG iterations. But evidence of the situation $\bar{H} \not\succeq \epsilon I$ can arise more subtly, when the residual norms $\|r_j\|$ decrease more slowly than we would expect if the eigenvalues of $\bar{H}$ were bounded below by ϵ. Accordingly, Algorithm 1 checks residual norms for slow decrease, and if such behavior is detected, it uses a technique based on one used for accelerated gradient methods in [5] to recover a direction d such that $d^\top \bar{H} d < \epsilon \|d\|^2$.

Algorithm 1 may be called with an optional input M that is meant to be an upper bound on $\|H\|$. Whether or not this parameter is supplied, it is updated so that at any point in the execution of the algorithm, M is an upper bound on the maximum curvature of H revealed to that point. Other parameters that depend on M (namely, κ, $\hat{\zeta}$, τ, and T) are updated whenever M is updated.

The following lemma justifies our use of the term "capped" in connection with Algorithm 1. Regardless of whether the condition $\bar{H} \succeq \epsilon I$ is satisfied, the number of iterations will not exceed a certain number $J(M, \epsilon, \zeta)$ that we subsequently show to be $\tilde{\mathcal{O}}(\epsilon^{-1/2})$. (We write J for $J(M, \epsilon, \zeta)$ in some of the subsequent discussion, to avoid clutter.)

Lemma 1 *The number of iterations of Algorithm 1 is bounded by*

$$\min\{n, J(M, \epsilon, \zeta)\},$$

where $J = J(M, \epsilon, \zeta)$ is the smallest integer such that $\sqrt{T}\tau^{J/2} \leq \hat{\zeta}$ where M, $\hat{\zeta}$, T, and τ are the values returned by the algorithm. If all iterates y_i generated by the algorithm are stored, the number of matrix-vector multiplications required is bounded by

$$\min\{n, J(M, \epsilon, \zeta)\} + 1. \tag{7}$$

If the iterates y_i must be regenerated in order to define the direction d returned after (6), this bound becomes $2\min\{n, J(M, \epsilon, \zeta)\} + 1$.

Proof If the full n iterations are performed, without any of the termination conditions being tripped, the standard properties of CG (see Appendix A) ensure that the final

residual r_n is zero, so that the condition $\|r_n\| \leq \hat{\zeta}\|r_0\|$ is satisfied, and termination occurs.

Since no more than n iterations are performed, the upper bound M is updated at most a finite number of times, so the quantity J is well defined.

Supposing that $J < n$, we note from the definition of J that $\sqrt{T}\tau^{J/2}\|r_0\| \leq \hat{\zeta}\|r_0\|$. Thus at least one of the following two conditions must be satisfied at iteration J: $\|r_J\| \leq \hat{\zeta}\|r_0\|$ or $\|r_J\| > \sqrt{T}\tau^{J/2}\|r_0\|$. In either case, termination will occur at iteration J, unless it has occurred already at a previous iteration.

To derive (7), note that the main workload at each iteration j is computation of a single matrix-vector product Hp_j after the increment of j (since matrix-vector products involving the matrices H and $\bar{H}$ and the vectors y_j and r_j can be computed in terms of this vector, in an additional $O(n)$ operations). (The "+1" in (7) accounts for the initial matrix-vector multiplication Hp_0 performed prior to entering the loop.)

If we do not store additional information, we need to regenerate the information needed to compute the direction d satisfying (6) by re-running the iterations of CG, possibly up to the second-to-last iteration. This fact accounts for the additional cost of $\min\{n, J(M, \epsilon, \zeta)\}$ in the no-storage case.[1] □

Note that $J(M, \epsilon, \zeta)$ is an increasing function of M, since $\hat{\zeta}$ is a decreasing function of M, while T and τ (and thus $\sqrt{T}\tau^{j/2}$) are increasing in M. If U_H is known in advance, we can call Algorithm 1 with $M = U_H$ and use $J(U_H, \epsilon, \zeta)$ as the bound. Alternately, we can call Algorithm 1 with $M = 0$ and let it adjust M as needed during the computation. Since the final value of M will be at most U_H, and since $J(M, \epsilon, \zeta)$ is an increasing function of M, the quantity $J(U_H, \epsilon, \zeta)$ provides the upper bound on the number of iterations in this case too.

We can estimate J by taking logs in its definition, as follows:

$$J \leq \frac{2\ln(\hat{\zeta}/\sqrt{T})}{\ln(\tau)} = \frac{\ln(\hat{\zeta}^2/T)}{\ln\left(\frac{\sqrt{\kappa}}{\sqrt{\kappa}+1}\right)} = \frac{\ln(T/\hat{\zeta}^2)}{\ln(1+1/\sqrt{\kappa})} \leq \left(\sqrt{\kappa}+\frac{1}{2}\right)\ln\left(\frac{T}{\hat{\zeta}^2}\right),$$

where we used $\ln(1+\frac{1}{t}) \geq \frac{1}{t+1/2}$ to obtain the latest inequality. By replacing T, τ, $\hat{\zeta}$, κ by their definitions in Algorithm 1, and using $\frac{1}{1-\sqrt{\tau}} = \frac{1+\sqrt{\tau}}{1-\tau} \leq \frac{2}{1-\tau}$, we obtain

$$\begin{aligned} J(M, \epsilon, \zeta) &\leq \min\left\{n, \left\lceil\left(\sqrt{\kappa}+\frac{1}{2}\right)\ln\left(\frac{144\left(\sqrt{\kappa}+1\right)^2\kappa^6}{\zeta^2}\right)\right\rceil\right\} \\ &= \min\left\{n, \tilde{\mathcal{O}}(\epsilon^{-1/2})\right\}. \end{aligned} \tag{8}$$

[1] Interestingly, as we show in Appendix A, the ratios on the left-hand side of (6) can be calculated without knowledge of y_i for $i = 0, 1, \ldots, j-1$, provided that we store the scalar quantities α_i and $\|r_i\|^2$ for $i = 0, 1, \ldots, j-1$.

3.2 Minimum eigenvalue oracle

A minimum eigenvalue oracle is needed in the main algorithm to either return a direction of "sufficient negative curvature" in a given symmetric matrix, or else return a certificate that the matrix is almost positive definite. This oracle is stated as Procedure 2.

Procedure 2 Minimum Eigenvalue Oracle

Inputs: Symmetric matrix $H \in \mathbb{R}^{n\times n}$, tolerance $\epsilon > 0$;
Optional input: Upper bound on Hessian norm M;
Outputs: An estimate λ of $\lambda_{\min}(H)$ such that $\lambda \leq -\epsilon/2$, and vector v with $\|v\| = 1$ such that $v^\top H v = \lambda$ OR a certificate that $\lambda_{\min}(H) \geq -\epsilon$. In the latter case, when the certificate is output, it is false with probability δ for some $\delta \in [0, 1)$.

To implement this oracle, we can use any procedure that finds the smallest eigenvalue of H to an absolute precision of $\epsilon/2$ with probability at least $1 - \delta$. This probabilistic property encompasses both deterministic and randomized instances of Procedure 2. In Sect. 4.3, we will establish complexity results under this general setting, and analyze the impact of the threshold δ. Several possibilities for implementing Procedure 2 have been proposed in the literature, with various guarantees. An exact, deterministic computation of the minimum eigenvalue and eigenvector (through a full Hessian evaluation and factorization) would be a valid choice for Procedure 2 (with $\delta = 0$ in that case), but is unsuited to our setting in which Hessian-vector products and vector operations are the fundamental operations. Strategies that require only gradient evaluations [2,28] may offer similar guarantees to those discussed below.

We focus on two inexact, randomized approaches for implementing Procedure 2. The first is the Lanczos method, which finds the smallest eigenvalue of the restriction of a given symmetric matrix to a Krylov subspace based on some initial vector. When the starting vector is chosen randomly, the dimension of the Krylov subspace increases by one at each Lanczos iteration, with high probability (see Appendix B and [22]). To the best of our knowledge, [6] was the first paper to propose a complexity analysis based on the use of randomized Lanczos for detecting negative curvature. The key result is the following.

Lemma 2 *Suppose that the Lanczos method is used to estimate the smallest eigenvalue of H starting with a random vector uniformly generated on the unit sphere, where $\|H\| \leq M$. For any $\delta \in [0, 1)$, this approach finds the smallest eigenvalue of H to an absolute precision of $\epsilon/2$, together with a corresponding direction v, in at most*

$$\min\left\{n, 1 + \left\lceil \frac{1}{2}\ln(2.75n/\delta^2)\sqrt{\frac{M}{\epsilon}}\right\rceil\right\} \quad \textit{iterations}, \tag{9}$$

with probability at least $1 - \delta$.

Proof If $\frac{\epsilon}{4M} \geq 1$, we have $-\frac{\epsilon}{4}I \prec -MI \preceq H \preceq MI \prec \frac{\epsilon}{4}I$. Therefore, letting b be the (unit norm) random start of the Lanczos method, we obtain

$$b^\top Hb \leq M < \frac{\epsilon}{4} = -\frac{\epsilon}{4} + \frac{\epsilon}{2} < -M + \frac{\epsilon}{2} \leq \lambda_{\min}(H) + \frac{\epsilon}{2},$$

thus the desired conclusion holds at the initial point.

We now suppose that $\frac{\epsilon}{4M} \in (0, 1)$. By setting $\bar{\epsilon} = \frac{\epsilon}{4M}$ in Lemma 9, we have that when k is at least the quantity in (9), the estimate $\xi_{\min}(H, b, k)$ of the smallest eigenvalue after k iterations of Lanczos applied to H starting from vector b satisfies the following bound, with probability at least $1 - \delta$:

$$\xi_{\min}(H, b, k) - \lambda_{\min}(H) \leq \bar{\epsilon}(\lambda_{\max}(H) - \lambda_{\min}(H)) \leq \frac{\epsilon}{2} \frac{\lambda_{\max}(H) - \lambda_{\min}(H)}{2M} \leq \frac{\epsilon}{2},$$

as required. □

Procedure 2 can be implemented by outputting the approximate eigenvalue λ for H, determined by the randomized Lanczos process, along with the corresponding direction v, provided that $\lambda \leq -\epsilon/2$. When $\lambda > -\epsilon/2$, Procedure 2 returns the certificate that $\lambda_{\min}(H) \geq -\epsilon$, which is correct with probability at least $1 - \delta$.

The second approach to implementing Procedure 2 is to apply the classical CG algorithm to solve a linear system in which the coefficient matrix is a shifted version of the matrix H and the right-hand side is random. This procedure has essentially identical performance to Lanczos in terms of the number of iterations required to detect the required direction of sufficiently negative curvature, as the following theorem shows.

Theorem 1 *Suppose that Procedure 2 consists in applying the standard CG algorithm (see Appendix A) to the linear system*

$$\left(H + \tfrac{1}{2}\epsilon I\right) d = b,$$

where b is chosen randomly from a uniform distribution over the unit sphere. Let M satisfying $\|H\| \leq M$ and $\delta \in (0, 1)$ be given. If $\lambda_{\min}(H) < -\epsilon$, then with probability at least $1 - \delta$, CG will yield a direction v satisfying the conditions of Procedure 2 in a number of iterations bounded above by (9). Conversely, if CG runs for this number of iterations without encountering a direction of negative curvature for $H + \frac{1}{2}\epsilon I$, then $\lambda_{\min}(H) \geq -\epsilon$ with probability at least $1 - \delta$.

We prove this result, and give some additional details of the CG implementation, in Appendices A and B. We also present in Appendix B.3 a variant of the randomized-Lanczos implementation of Procedure 2 that does not require prior knowledge of the bound M such that $\|H\| \leq M$. In this variant, M itself is also estimated via randomized Lanczos, and the number of iterations required does not different significantly from (9). It follows from this result, together with our observation above that M can also be obtained adaptively inside Algorithm 1, that knowledge of the bound on $\|\nabla^2 f(x)\|$ is not needed at all in implementing our method.

3.3 Damped Newton-CG

Algorithm 3 presents our method for finding a point that satisfies (2). It uses two kinds of search directions. Negative curvature directions (that are also first-order descent

steps) are used when they are either encountered in the course of applying the Capped CG method (Algorithm 1) to the damped Newton equations, or found explicitly by application of Procedure 2. The second type of step is an inexact damped Newton step, which is the other possible outcome of Algorithm 1. For both types of steps, a backtracking line search is used to identify a new iterate that satisfies a sufficient decrease condition, that depends on the cubic norm of the step. Such a criterion is instrumental in establishing optimal complexity guarantees in second-order methods [3,13,14,26].

Algorithm 3 Damped Newton-CG

Inputs: Tolerances $\epsilon_g > 0$, $\epsilon_H > 0$; backtracking parameter $\theta \in (0, 1)$; starting point x_0; accuracy parameter $\zeta \in (0, 1)$; sufficient decrease parameter $\eta > 0$;
Optional input: Upper bound $M > 0$ on Hessian norm;
for $k = 0, 1, 2, \ldots$ **do**
  if $\|\nabla f(x_k)\| > \epsilon_g$ **then**
    Call Algorithm 1 with $H = \nabla^2 f(x_k)$, $\epsilon = \epsilon_H$, $g = \nabla f(x_k)$, accuracy parameter ζ and M if provided, to obtain outputs d, d_type;
    if d_type=NC **then**
      $d_k \leftarrow -\text{sgn}(d^\top g) \frac{|d^\top \nabla^2 f(x_k) d|}{\|d\|^2} \frac{d}{\|d\|}$;
    else {d_type=SOL}
      $d_k \leftarrow d$;
    end if
    Go to **Line Search**;
  else
    Call Procedure 2 with $H = \nabla^2 f(x_k)$, $\epsilon = \epsilon_H$ and M if provided;
    if Procedure 2 certifies that $\lambda_{\min}(\nabla^2 f(x_k)) \geq -\epsilon_H$ **then**
      Terminate;
    else {direction of sufficient negative curvature found}
      $d_k \leftarrow -\text{sgn}(v^\top g) \frac{|v^\top \nabla^2 f(x_k) v|}{\|v\|^2} v$ (where v is the output from Procedure 2) and go to **Line Search**;
    end if
  end if
  Line Search: Compute a step length $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer such that

$$f(x_k + \alpha_k d_k) < f(x_k) - \frac{\eta}{6} \alpha_k^3 \|d_k\|^3; \tag{10}$$

  $x_{k+1} \leftarrow x_k + \alpha_k d_k$;
end for

In its deployment of two types of search directions, our method is similar to Steihaug's trust-region Newton-CG method [27], which applies CG (starting from a zero initial guess) to solve the Newton equations but, if it encounters a negative curvature direction during CG, steps along that direction to the trust-region boundary. It differs from the line-search Newton-CG method described in [25, Section 7.1] in that it makes use of negative curvature directions when they are encountered, rather than discarding them in favor of a steepest-descent direction. Algorithm 3 improves over both approaches in having a global complexity theory for convergence to both approximate first-order points, and points satisfying the approximate second-order conditions (2).

In Sect. 4, we will analyze the global complexity properties of our algorithm. Local convergence could also be of interest, in particular, it is probably possible to prove rapid convergence of Algorithm 3 once it reaches the neighborhood of a strict local minimum. We believe that such results would be complicated and less enlightening than the complexity guarantees, so we restrict our study to the latter.

4 Complexity analysis

In this section, we present a global worst-case complexity analysis of Algorithm 3. Elements of the analysis follow those in the earlier paper [26]. The most technical part appears in Sect. 4.1 below, where we show that the Capped CG procedure returns (deterministically) either an inexact Newton step or a negative curvature direction, both of which can be used as the basis of a successful backtracking line search. These properties are used in Sect. 4.2 to prove complexity results for convergence to a point satisfying the approximate first-order condition $\|\nabla f(x)\| \leq \epsilon_g$. Section 4.3 proves complexity results for finding approximate second-order points (2), leveraging properties of the minimum eigenvalue oracle, Procedure 2.

4.1 Properties of Capped CG

We now explore the properties of the directions d that are output by our Capped CG procedure, Algorithm 1. The main result deals with the case in which Algorithm 1 terminates due to insufficiently rapid decrease in $\|r_j\|$, showing that the strategy for identifying a direction of sufficient negative curvature for H is effective.

Theorem 2 *Suppose that the main loop of Algorithm 1 terminates with $j = \hat{J}$, where*

$$\hat{J} \in \{1, \ldots, \min\{n, J(M, \epsilon, \zeta)\}\},$$

(where $J(M, \epsilon, \zeta)$ is defined in Lemma 1 and (8)) because the fourth termination test is satisfied and the three earlier conditions do not hold, that is, $y_{\hat{J}}^\top \bar{H} y_{\hat{J}} \geq \epsilon \|y_{\hat{J}}\|^2$, $p_{\hat{J}}^\top \bar{H} p_{\hat{J}} \geq \epsilon \|p_{\hat{J}}\|^2$, and

$$\|r_{\hat{J}}\| > \max\left\{\hat{\zeta}, \sqrt{T}\tau^{\hat{J}/2}\right\} \|r_0\|. \tag{11}$$

where M, T, $\hat{\zeta}$, and τ are the values returned by Algorithm 1. Then $y_{\hat{J}+1}$ is computed by Algorithm 1, and we have

$$\frac{(y_{\hat{J}+1} - y_i)^\top \bar{H}(y_{\hat{J}+1} - y_i)}{\|y_{\hat{J}+1} - y_i\|^2} < \epsilon, \quad \text{for some } i \in \{0, \ldots, \hat{J} - 1\}. \tag{12}$$

The proof of Theorem 2 is quite technical, and can be found in Appendix C. It relies on an argument previously used to analyze a strategy based on accelerated gradient [5, Appendix A.1], itself inspired by a result of Bubeck [4], but it needs some additional

steps that relate specifically to CG. The part of our proof that corresponds to [5, Appendix A.1] is simplified in some respects, thanks to the use of CG and the fact that a quadratic (rather than a nonlinear) function is being minimized in the subproblem.

Having shown that Algorithm 1 is well-defined, we summarize the properties of its outputs.

Lemma 3 *Let Assumptions 1 and 2 hold, and suppose that Algorithm 1 is invoked at an iterate x_k of Algorithm 3 (so that $\|\nabla f(x_k)\| > \epsilon_g > 0$). Let d_k be the vector obtained in Algorithm 3 from the output d of Algorithm 1. Then, one of the two following statements holds:*

1. d_type = SOL, *and the direction d_k satisfies*

$$d_k^\top (\nabla^2 f(x_k) + 2\epsilon_H I) d_k \geq \epsilon_H \|d_k\|^2, \tag{13a}$$

$$\|d_k\| \leq 1.1 \epsilon_H^{-1} \|\nabla f(x_k)\|, \tag{13b}$$

$$\|\hat{r}_k\| \leq \frac{1}{2} \epsilon_H \zeta \|d_k\|, \tag{13c}$$

where

$$\hat{r}_k := (\nabla^2 f(x_k) + 2\epsilon_H I) d_k + \nabla f(x_k); \tag{14}$$

2. d_type = NC, *and the direction d_k satisfies $d_k^\top \nabla f(x_k) \leq 0$ as well as*

$$\frac{d_k^\top \nabla^2 f(x_k) d_k}{\|d_k\|^2} = -\|d_k\| \leq -\epsilon_H. \tag{15}$$

Proof For simplicity of notation, we use $H = \nabla^2 f(x_k)$ and $g = \nabla f(x_k)$ in the proof. Suppose first that d_type = SOL. In that case, we have from the termination conditions in Algorithm 1 and (14) that

$$d_k^\top (H + 2\epsilon_H I) d_k \geq \epsilon_H \|d_k\|^2, \tag{16a}$$

$$\|\hat{r}_k\| \leq \hat{\zeta} \|g\|, \tag{16b}$$

where $\hat{\zeta}$ was returned by the algorithm. We immediately recognize (13a) in (16a). We now prove (13b). Observe first that (16a) yields

$$\epsilon_H \|d_k\|^2 \leq d_k^\top (H + 2\epsilon_H I) d_k \leq \|d_k\| \|(H + 2\epsilon_H I) d_k\|,$$

so from (14) we have

$$\begin{aligned} \|d_k\| &\leq \epsilon_H^{-1} \|(H + 2\epsilon_H I) d_k\| = \epsilon_H^{-1} \| - g + \hat{r}_k\| \\ &= \epsilon_H^{-1} \sqrt{\|g\|^2 + \|\hat{r}_k\|^2} \leq \epsilon_H^{-1} \sqrt{1 + \hat{\zeta}^2} \|g\|, \end{aligned}$$

where we used (16b) to obtain the final bound, together with the equality $\|-g+\hat{r}_k\|^2 = \|g\|^2+\|\hat{r}_k\|^2$, which follows from $g^\top \hat{r}_k = r_0^\top \hat{r}_k = 0$, by orthogonality of the residuals in CG (see Lemma 7, Property 2). Since $\hat{\zeta} \leq \zeta/(3\kappa) \leq 1/6$ by construction, we have $\|d_k\| \leq \sqrt{37/36}\epsilon_H^{-1}\|g\| \leq 1.1\epsilon_H^{-1}\|g\|$, proving (13b).

The bound (13c) follows from (16b) and the logic below:

$$\|\hat{r}_k\| \leq \hat{\zeta}\|g\| \leq \hat{\zeta}\left(\|(H+2\epsilon_H I)d_k\| + \|\hat{r}_k\|\right) \leq \hat{\zeta}\left((M+2\epsilon_H)\|d_k\| + \|\hat{r}_k\|\right)$$
$$\Rightarrow \quad \|\hat{r}_k\| \leq \frac{\hat{\zeta}}{1-\hat{\zeta}}(M+2\epsilon_H)\|d_k\|,$$

where M is the value returned by the algorithm. We finally use $\hat{\zeta} < 1/6$ to arrive at

$$\frac{\hat{\zeta}}{1-\hat{\zeta}}(M+2\epsilon_H) \leq \frac{6}{5}\hat{\zeta}(M+2\epsilon_H) = \frac{6}{5}\frac{\zeta\epsilon_H}{3} < \frac{1}{2}\zeta\epsilon_H,$$

yielding (13c).

In the case of d_type=NC, we recall that Algorithm 3 defines

$$d_k = -\text{sgn}(d^\top g)\frac{|d^\top H d|}{\|d\|^2}\frac{d}{\|d\|} \tag{17}$$

where d denotes the direction obtained by Algorithm 1. It follows immediately that $d_k^\top g \leq 0$. Since d_k and d are collinear, we also have that

$$\frac{d_k^\top(H+2\epsilon_H I)d_k}{\|d_k\|^2} = \frac{d^\top(H+2\epsilon_H I)d}{\|d\|^2} \leq \epsilon_H \;\Rightarrow\; \frac{d_k^\top H d_k}{\|d_k\|^2} \leq -\epsilon_H.$$

By using this bound together with (17), we obtain

$$\|d_k\| = \frac{|d^\top H d|}{\|d\|^2} = \frac{|d_k^\top H d_k|}{\|d_k\|^2} = -\frac{d_k^\top H d_k}{\|d_k\|^2} \geq \epsilon_H,$$

proving (15). □

4.2 First-order complexity analysis

We now find a bound on the number of iterations and the amount of computation required to identify an iterate x_k for which $\|\nabla f(x_k)\| \leq \epsilon_g$. We consider in turn the two types of steps (approximate damped Newton and negative curvature), finding a lower bound on the descent in f achieved on the current iteration in each case. We then prove an upper bound on the number of iterations required to satisfy these approximate first-order conditions (Theorem 3) and an upper bound on the number of gradient evaluations and Hessian-vector multiplications required (Theorem 4).

We start with a lemma concerning the approximate damped Newton steps.

Lemma 4 *Suppose that Assumptions 1 and 2 hold. Suppose that at iteration k of Algorithm 3, we have $\|\nabla f(x_k)\| > \epsilon_g$, so that Algorithm 1 is called. When Algorithm 1 outputs a direction d_k with d_type = SOL, then the backtracking line search requires at most $j_k \leq j_{sol} + 1$ iterations, where*

$$j_{sol} \;=\; \left\lceil \frac{1}{2} \log_\theta \left(\frac{3(1-\zeta)}{L_H + \eta} \frac{\epsilon_H^2}{1.1 U_g} \right) \right\rceil_+ , \tag{18}$$

and the resulting step $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$f(x_k) - f(x_{k+1}) \;\geq\; c_{sol} \min \left(\|\nabla f(x_{k+1})\|^3 \epsilon_H^{-3}, \epsilon_H^3 \right), \tag{19}$$

where

$$c_{sol} = \frac{\eta}{6} \min \left\{ \left[\frac{4}{\sqrt{(4+\zeta)^2 + 8 L_H} + 4 + \zeta} \right]^3 , \left[\frac{3\theta^2(1-\zeta)}{L_H + \eta} \right]^3 \right\}.$$

Proof The proof tracks closely that of [26, Lemma 13]. The only significant difference is that equation (65) of [26], which is instrumental to the proof and requires a probabilistic assumption on $\lambda_{\min}(\nabla^2 f(x_k))$, is now ensured *deterministically* by (13a) from Lemma 3. As a result, both the proof and the result are deterministic. □

When $\|\nabla f(x_{k+1})\| \leq \epsilon_g$, the estimate (19) may not guarantee a "significant" decrease in f at this iteration. However, in this case, the approximate first-order condition $\|\nabla f(x)\| \leq \epsilon_g$ holds at the *next* iteration, so that Algorithm 3 will invoke Procedure 2 at iteration $k+1$, leading either to termination with satisfaction of the conditions (2) or to a step that reduces f by a multiple of ϵ_H^3, as we show in Theorem 4 below.

We now address the case in which Algorithm 1 returns a negative curvature direction to Algorithm 3 at iteration k. The backtracking line search guarantees that a sufficient decrease will be achieved at such an iteration. Although the Lipschitz constant L_H appears in our result, our algorithm (in contrast to [5]) does not require this constant to be known or estimated.

Lemma 5 *Suppose that Assumptions 1 and 2 hold. Suppose that at iteration k of Algorithm 3, we have $\|\nabla f(x_k)\| > \epsilon_g$, so that Algorithm 1 is called. When Algorithm 1 outputs d_type = NC, the direction d_k (computed from d in Algorithm 3) has the following properties: The backtracking line search terminates with step length $\alpha_k = \theta^{j_k}$ with $j_k \leq j_{nc} + 1$, where*

$$j_{nc} := \left\lceil \log_\theta \left(\frac{3}{L_H + \eta} \right) \right\rceil_+ , \tag{20}$$

and the resulting step $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$f(x_k) - f(x_k + \alpha_k d_k) \;\geq\; c_{nc} \epsilon_H^3, \tag{21}$$

with

$$c_{nc} := \frac{\eta}{6} \min\left\{1, \frac{27\theta^3}{(L_H + \eta)^3}\right\}.$$

Proof By Lemma 3, we have from (15) that

$$d_k^\top \nabla^2 f(x_k) d_k = -\|d_k\|^3 \leq -\epsilon_H \|d_k\|^2 \tag{22}$$

The result can thus be obtained exactly as in [26, Lemma 1]. □

We are ready to state our main result for first-order complexity.

Theorem 3 *Let Assumptions 1 and 2 hold. Then, defining*

$$\bar{K}_1 := \left\lceil \frac{f(x_0) - f_{\text{low}}}{\min\{c_{sol}, c_{nc}\}} \max\left\{\epsilon_g^{-3}\epsilon_H^3, \epsilon_H^{-3}\right\} \right\rceil,$$

some iterate x_k, $k = 0, 1, \ldots, \bar{K}_1 + 1$ generated by Algorithm 3 will satisfy

$$\|\nabla f(x_k)\| \leq \epsilon_g. \tag{23}$$

Proof Suppose for contradiction that $\|\nabla f(x_k)\| > \epsilon_g$ for all $k = 0, 1, \ldots, \bar{K}_1 + 1$, so that

$$\|\nabla f(x_{l+1})\| > \epsilon_g, \quad l = 0, 1, \ldots, \bar{K}_1. \tag{24}$$

Algorithm 1 will be invoked at each of the first $\bar{K}_1 + 1$ iterates of Algorithm 3. For each iteration $l = 0, 1, \ldots, \bar{K}_1$ for which Algorithm 1 returns d_type = SOL, we have from Lemma 4 and (24) that

$$f(x_l) - f(x_{l+1}) \geq c_{sol} \min\left\{\|\nabla f(x_{l+1})\|^3 \epsilon_H^{-3}, \epsilon_H^3\right\} \geq c_{sol} \min\left\{\epsilon_g^3 \epsilon_H^{-3}, \epsilon_H^3\right\}. \tag{25}$$

For each iteration $l = 0, 1, \ldots, \bar{K}_1$ for which Algorithm 1 returns d_type = NC, we have by Lemma 5 that

$$f(x_l) - f(x_{l+1}) \geq c_{nc} \epsilon_H^3. \tag{26}$$

By combining these results, we obtain

$$\begin{aligned} f(x_0) - f(x_{\bar{K}_1+1}) &\geq \sum_{l=0}^{\bar{K}_1} (f(x_l) - f(x_{l+1})) \\ &\geq \sum_{l=0}^{\bar{K}_1} \min\{c_{sol}, c_{nc}\} \min\left\{\epsilon_g^3 \epsilon_H^{-3}, \epsilon_H^3\right\} \end{aligned}$$

$$= (\bar{K}_1 + 1) \min\{c_{sol}, c_{nc}\} \min\left\{\epsilon_g^3 \epsilon_H^{-3}, \epsilon_H^3\right\}$$
$$> f(x_0) - f_{\text{low}}.$$

where we used the definition of $\bar{K}_1$ for the final inequality. This inequality contradicts the definition of f_{low} in (3), so our claim is proved. □

If we choose ϵ_H in the range $[\epsilon_g^{1/3}, \epsilon_g^{2/3}]$, this bound improves over the classical $\mathcal{O}(\epsilon_g^{-2})$ rate of gradient-based methods. The choice $\epsilon_H = \epsilon_g^{1/2}$ yields the rate $\mathcal{O}(\epsilon_g^{-3/2})$, which is known to be optimal among second-order methods [9].

Recalling that the workload of Algorithm 1 in terms of Hessian-vector products depends on the index J defined by (8), we obtain the following corollary. (Note the mild assumption on the quantities of M used at each instance of Algorithm 1, which is satisfied provided that this algorithm is always invoked with an initial estimate of M in the range $[0, U_H]$.)

Corollary 1 *Suppose that the assumptions of Theorem 3 are satisfied, and let $\bar{K}_1$ be as defined in that theorem and $J(M, \epsilon_H, \zeta)$ be as defined in (8). Suppose that the values of M used or calculated at each instance of Algorithm 1 satisfy $M \leq U_H$. Then the number of Hessian-vector products and/or gradient evaluations required by Algorithm 3 to output an iterate satisfying (23) is at most*

$$(2 \min\{n, J(U_H, \epsilon_H, \zeta)\} + 2)(\bar{K}_1 + 1).$$

For n sufficiently large, this bound is $\tilde{\mathcal{O}}\left(\max\left\{\epsilon_g^{-3}\epsilon_H^{5/2}, \epsilon_H^{-7/2}\right\}\right)$, while if $J(U_H, \epsilon_H, \zeta) \geq n$, the bound is $\tilde{\mathcal{O}}\left(n \max\left\{\epsilon_g^{-3}\epsilon_H^3, \epsilon_H^{-3}\right\}\right)$.

Proof From Lemma 1, the number of Hessian-vector multiplications in the main loop of Algorithm 1 is bounded by $\min\{n, J(U_H, \epsilon_H, \zeta) + 1\}$. An additional $\min\{n, J(U_H, \epsilon_H, \zeta)\}$ Hessian-vector products may be needed to return a direction satisfying (6), if Algorithm 1 does not store its iterates y_j. Each iteration also requires a single evaluation of the gradient ∇f, giving a bound of $(2 \min\{n, J(U_H, \epsilon_H, \zeta)\} + 2)$ on the workload per iteration of Algorithm 3. We multiply this quantity by the iteration bound from Theorem 3 to obtain the result. □

By setting $\epsilon_H = \epsilon_g^{1/2}$, we obtain from this corollary a computational bound of $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ (for n sufficiently large), which matches the deterministic first-order guarantee obtained in [5], and also improves over the $\mathcal{O}(\epsilon_g^{-2})$ computational complexity of gradient-based methods.

4.3 Second-order complexity results

We now find bounds on iteration and computational complexity of finding a point that satisfies (2). In this section, as well as using results from Sects. 4.1 and 4.2, we also need to use the properties of the minimum eigenvalue oracle, Procedure 2. To this end, we make the following generic assumption.

Assumption 3 For every iteration k at which Algorithm 3 calls Procedure 2, and for a specified failure probability δ with $0 \leq \delta \ll 1$, Procedure 2 either certifies that $\nabla^2 f(x_k) \succeq -\epsilon_H I$ or finds a vector of curvature smaller than $-\epsilon_H/2$ in at most

$$N_{\text{meo}} := \min\left\{n, 1 + \left\lceil \mathcal{C}_{\text{meo}} \epsilon_H^{-1/2} \right\rceil\right\} \tag{27}$$

Hessian-vector products, with probability $1-\delta$, where $\mathcal{C}_{\text{meo}}$ depends at most logarithmically on δ and ϵ_H.

Assumption 3 encompasses the strategies we mentioned in Sect. 3.2. Assuming the bound U_H on $\|H\|$ is available, for both the Lanczos method with a random starting vector and the conjugate gradient algorithm with a random right-hand side, (27) holds with $\mathcal{C}_{\text{meo}} = \ln(2.75n/\delta^2)\sqrt{U_H}/2$. When a bound on $\|H\|$ is not available in advance, it can be estimated efficiently with minimal effect on the overall complexity of the method, as shown in Appendix B.3.

The next lemma guarantees termination of the backtracking line search for a negative curvature direction, regardless of whether it is produced by Algorithm 1 or Procedure 2. As in Lemma 4, the result is deterministic.

Lemma 6 *Suppose that Assumptions 1 and 2 hold. Suppose that at iteration k of Algorithm 3, the search direction d_k is of negative curvature type, obtained either directly from Procedure 2 or as the output of Algorithm 1 and d_type = NC. Then the backtracking line search terminates with step length $\alpha_k = \theta^{j_k}$ with $j_k \leq j_{nc} + 1$, where j_{nc} is defined as in Lemma 5, and the decrease in the function value resulting from the chosen step length satisfies*

$$f(x_k) - f(x_k + \alpha_k d_k) \;\geq\; \frac{c_{nc}}{8}\epsilon_H^3, \tag{28}$$

with c_{nc} is defined in Lemma 5.

Proof Lemma 5 shows that the claim holds (with a factor of 8 to spare) when the direction of negative curvature is obtained from Algorithm 1. When the direction is obtained from Procedure 2, we have by the scaling of d_k applied in Algorithm 3 that

$$d_k^\top \nabla^2 f(x_k) d_k = -\|d_k\|^3 \leq -\frac{1}{2}\epsilon_H \|d_k\|^2 < 0, \tag{29}$$

from which it follows that $\|d_k\| \geq \frac{1}{2}\epsilon_H$. The result can now be obtained by following the proof of Lemma 5, with $\frac{1}{2}\epsilon_H$ replacing ϵ_H. □

We are now ready to state our iteration complexity result for Algorithm 3.

Theorem 4 *Suppose that Assumptions 1, 2, and 3 hold, and define*

$$\bar{K}_2 := \left\lceil \frac{3(f(x_0) - f_{\text{low}})}{\min\{c_{sol}, c_{nc}/8\}} \max\{\epsilon_g^{-3}\epsilon_H^3, \epsilon_H^{-3}\} \right\rceil + 2, \tag{30}$$

where constants c_{sol} and c_{nc} are defined in Lemmas 4 and 5, respectively. Then with probability at least $(1-\delta)^{\bar{K}_2}$, Algorithm 3 terminates at a point satisfying (2) in at most $\bar{K}_2$ iterations. (With probability at most $1-(1-\delta)^{\bar{K}_2}$, it terminates incorrectly within $\bar{K}_2$ iterations at a point for which $\|\nabla f(x_k)\| \leq \epsilon_g$ but $\lambda_{\min}(\nabla^2 f(x)) < -\epsilon_H$.)

Proof Algorithm 3 terminates incorrectly with probability δ at any iteration at which Procedure 2 is called, when Procedure 2 certifies erroneously that $\lambda_{\min}(\nabla^2 f(x)) \geq -\epsilon_H$. Since an erroneous certificate can only lead to termination, an erroneous certificate at iteration k means that Procedure 2 did not produce an erroneous certificate at iterations 0 to $k-1$. By a disjunction argument, we have that the overall probability of terminating with an erroneous certificate during the first $\bar{K}_2$ iterations is bounded by $1-(1-\delta)^{\bar{K}_2}$. Therefore, with probability at least $(1-\delta)^{\bar{K}_2}$, no incorrect termination occurs in the first $\bar{K}_2$ iterations.

Suppose now for contradiction that Algorithm 3 runs for $\bar{K}_2$ iterations without terminating. That is, for all $l = 0, 1, \ldots, \bar{K}_2$, we have either $\|\nabla f(x_l)\| > \epsilon_g$ or $\lambda_{\min}(\nabla^2 f(x_l)) < -\epsilon_H$. We perform the following partition of the set of iteration indices:

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3 = \{0, 1, \ldots, \bar{K}_2 - 1\}, \tag{31}$$

where $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$ are defined as follows.

Case 1 $\mathcal{K}_1 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 \ : \ \|\nabla f(x_l)\| \leq \epsilon_g\}$. At each iteration $l \in \mathcal{K}_1$, Algorithm 3 calls Procedure 2, which does not certify that $\lambda_{\min}(\nabla^2 f(x_l)) \geq -\epsilon_H$ (since the algorithm continues to iterate) but rather returns a direction of sufficient negative curvature. By Lemma 6, the step along this direction leads to an improvement in f that is bounded as follows:

$$f(x_l) - f(x_{l+1}) \geq \frac{c_{nc}}{8}\epsilon_H^3. \tag{32}$$

Case 2 $\mathcal{K}_2 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 \ : \ \|\nabla f(x_l)\| > \epsilon_g \text{ and } \|\nabla f(x_{l+1})\| > \epsilon_g\}$. Algorithm 3 calls Algorithm 1 at each iteration $l \in \mathcal{K}_2$, returning either an approximate damped Newton or a negative curvature direction. By combining Lemmas 4 and 5, we obtain a decrease in f satisfying

$$\begin{aligned} f(x_l) - f(x_{l+1}) &\geq \min\{c_{sol}, c_{nc}\} \min\left\{\|\nabla f(x_{l+1})\|^3 \epsilon_H^{-3}, \epsilon_H^3\right\} \\ &\geq \min\{c_{sol}, c_{nc}/8\} \min\left\{\epsilon_g^3 \epsilon_H^{-3}, \epsilon_H^3\right\}. \end{aligned} \tag{33}$$

Case 3 $\mathcal{K}_3 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 \ : \ \|\nabla f(x_l)\| > \epsilon_g \geq \|\nabla f(x_{l+1})\|\}$. Because $\|\nabla f(x_{l+1})\|$ may be small in this case, we can no longer bound the decrease in f by an expression such as (33). We can however guarantee at least that $f(x_l) - f(x_{l+1}) \geq 0$. Moreover, provided that $l < \bar{K}_2 - 1$, we have from $\|\nabla f(x_{l+1})\| \leq \epsilon_g$ that the next iterate $l+1$ is in $\mathcal{K}_1$. Thus, a significant decrease in f will be attained at the *next* iteration, and we have

$$|\mathcal{K}_3| \leq |\mathcal{K}_1| + 1. \tag{34}$$

We now consider the total decrease in f over the span of $\bar{K}_2$ iterations, which is bounded by $f(x_0) - f_{\text{low}}$ as follows:

$$\begin{aligned} f(x_0) - f_{\text{low}} &\geq \sum_{l=0}^{\bar{K}_2 - 1} (f(x_l) - f(x_{l+1})) \\ &\geq \sum_{l \in \mathcal{K}_1} (f(x_l) - f(x_{l+1})) + \sum_{l \in \mathcal{K}_2} (f(x_l) - f(x_{l+1})) \end{aligned} \tag{35}$$

where both sums in the final expression are nonnegative. Using first the bound (32) for the sum over $\mathcal{K}_1$, we obtain

$$f(x_0) - f_{\text{low}} \;\geq\; |\mathcal{K}_1| \frac{c_{nc}}{8} \epsilon_H^3 \;\Leftrightarrow\; |\mathcal{K}_1| \leq \frac{f(x_0) - f_{\text{low}}}{c_{nc}/8} \epsilon_H^{-3}. \tag{36}$$

Applying (33) to the sum over $\mathcal{K}_2$ leads to

$$|\mathcal{K}_2| \leq \frac{f(x_0) - f_{\text{low}}}{\min\{c_{sol}, c_{nc}/8\}} \max\{\epsilon_g^{-3} \epsilon_H^3, \epsilon_H^{-3}\}. \tag{37}$$

Using these bounds together with (34), we have

$$\begin{aligned} \bar{K}_2 &= |\mathcal{K}_1| + |\mathcal{K}_2| + |\mathcal{K}_3| \\ &\leq 2|\mathcal{K}_1| + |\mathcal{K}_2| + 1 \\ &\leq 3 \max\{|\mathcal{K}_1|, |\mathcal{K}_2|\} + 1 \\ &\leq \frac{3(f(x_0) - f_{\text{low}})}{\min\{c_{sol}, c_{nc}/8\}} \max\{\epsilon_g^{-3} \epsilon_H^3, \epsilon_H^{-3}\} + 1 \\ &\leq \bar{K}_2 - 1, \end{aligned}$$

giving the required contradiction. □

We note that when $\delta < 1/\bar{K}_2$ in Theorem 4, a technical result shows that $(1-\delta)^{\bar{K}_2} \geq 1 - \delta \bar{K}_2$. In this case, the qualifier "with probability at least $(1-\delta)^{\bar{K}_2}$" in the theorem can be replaced by "with probability at least $1 - \delta \bar{K}_2$" while remaining informative.

Finally, we provide an operation complexity result: a bound on the number of Hessian-vector products and gradient evaluations necessary for Algorithm 3 to find a point that satisfies (2).

Corollary 2 *Suppose that assumptions of Theorem 4 hold, and let $\bar{K}_2$ be defined as in (30). Suppose that the values of M used or calculated at each instance of Algorithm 1 satisfy $M \leq U_H$. Then with probability at least $(1-\delta)^{\bar{K}_2}$, Algorithm 3 terminates at a point satisfying (2) after at most*

$$(\max\{2 \min\{n, J(U_H, \epsilon_H, \zeta)\} + 2, N_{\text{meo}}\})\, \bar{K}_2$$

Hessian-vector products and/or gradient evaluations. (With probability at most $1-(1-\delta)^{\bar{K}_2}$*, it terminates incorrectly with this complexity at a point for which* $\|\nabla f(x_k)\| \leq \epsilon_g$ *but* $\lambda_{\min}(\nabla^2 f(x)) < -\epsilon_H$*.)*

For n *sufficiently large, and assuming that* $\delta < 1/\bar{K}_2$*, the bound is* $\tilde{\mathcal{O}}\left(\max\left\{\epsilon_g^{-3}\epsilon_H^{5/2}, \epsilon_H^{-7/2}\right\}\right)$*, with probability at most* $1-\bar{K}_2\delta$*.*

Proof The proof follows by combining Theorem 4 (which bounds the number of iterations) with Lemma 1 and Assumption 3 (which bound the workload per iteration). □

By setting $\epsilon_H = \epsilon_g^{1/2}$ and assuming that n is sufficiently large, we recover (with high probability) the familiar complexity bound of order $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$, matching the bound of accelerated gradient-type methods such as [1,6,19].

5 Discussion

We have presented a Newton-CG approach for smooth nonconvex unconstrained minimization that is close to traditional variants of this method, but incorporates additional checks and safeguards that enable convergence to a point satisfying approximate second-order conditions (2) with guaranteed complexity. This was achieved by exploiting the properties of Lanczos-based methods in two ways. First, we used CG to compute Newton-type steps when possible, while monitoring convexity during the CG iterations to detect negative curvature directions when those exist. Second, by exploiting the close relationship between the Lanczos and CG algorithms, we show that both methods can be used to detect negative curvature of a given symmetric matrix with high probability. Both techniques are endowed with complexity guarantees, and can be combined within a Newton-CG framework to match the best known bounds for second-order algorithms on nonconvex optimization [11].

Nonconvexity detection can be introduced into CG in ways other than those used in Algorithm 1. For instance, we can drop the *implicit* cap on the number of CG iterations that is due to monitoring of the condition $\|r_j\| > \sqrt{T}\tau^{j/2}\|r_0\|$ and use of the negative curvature direction generation procedure (6) from Algorithm 1, and instead impose an *explicit* cap (smaller by a factor of approximately 4 than $J(M, \epsilon, \zeta)$) on the number of CG iterations. In this version, if the explicit cap is reached without detection of a direction of sufficient negative curvature for $\bar{H}$, then Procedure 2 is invoked to find one. This strategy comes equipped with essentially the same high-probability complexity results as Theorem 4 and Corollary 2, but it lacks the deterministic approximate-first-order complexity guarantee of Theorem 3. On the other hand, it is more elementary, both in the specification of the Capped CG procedure and the analysis.

A common feature to the Capped CG procedures described in Algorithm 1 and in the above paragraph, which also emerges in most Newton-type methods with good complexity guarantees [11], is the need for high accuracy in the step computation. That is, only a small residual is allowed in the damped Newton system at the approximate solution. Looser restrictions are typically used in practical algorithms, but our tighter bounds appear to be necessary for the complexity analysis. Further investigation of the

differences between our procedure in this paper and practical Newton-CG procedures is a subject of ongoing research.

Acknowledgements We thank sincerely the associate editor and two referees, whose comments led us to improve the presentation and to derive stronger results.

Funding Funding was provided by National Science Foundation (Grant Nos. 1447449, 1628384, 1634597, 1740707), Air Force Office of Scientific Research (Grant No. FA9550-13-1-0138) and Argonne National Laboratory (Grant Nos. 3F-30222, 8F-30039) and DARPA (Grant No. N660011824020).

A Linear conjugate gradient: relevant properties

In this appendix, we provide useful results for the classical CG algorithm, that also apply to the "standard CG" operations within Algorithm 1. To this end, and for the sake of discussion in Appendix B, we sketch the standard CG method in Algorithm 4, reusing the notation of Algorithm 1.

Algorithm 4 Conjugate Gradient

Inputs: Symmetric matrix $\bar{H}$, vector g;
$r_0 \leftarrow g$, $p_0 \leftarrow -r_0$, $y_0 \leftarrow 0$, $j \leftarrow 0$;
while $p_j^\top \bar{H} p_j > 0$ and $r_j \neq 0$ **do**
  $\alpha_j \leftarrow r_j^\top r_j / p_j^\top \bar{H} p_j$;
  $y_{j+1} \leftarrow y_j + \alpha_j p_j$;
  $r_{j+1} \leftarrow r_j + \alpha_j \bar{H} p_j$;
  $\beta_{j+1} \leftarrow (r_{j+1}^\top r_{j+1})/(r_j^\top r_j)$;
  $p_{j+1} \leftarrow -r_{j+1} + \beta_{j+1} p_j$;
  $j \leftarrow j + 1$;
end while

Here and below, we refer often to the following quadratic function:

$$q(y) := \frac{1}{2} y^\top \bar{H} y + g^\top y, \tag{38}$$

where $\bar{H}$ and g are the matrix and vector parameters of Algorithms 1 or 4. When $\bar{H}$ is positive definite, the minimizer of q is identical to the unique solution of $\bar{H} y = -g$. CG can be viewed either as an algorithm to solve $\bar{H} y = -g$ or as an algorithm to minimize q.

The next lemma details several properties of the conjugate gradient method to be used in the upcoming proofs.

Lemma 7 *Suppose that j iterations of the CG loop are performed in Algorithm 1 or 4. Then, we have*

$$\frac{p_i^\top \bar{H} p_i}{\|p_i\|^2} > 0 \quad \text{for all } i = 0, 1, \ldots, j-1. \tag{39}$$

Moreover, the following properties hold.

1. $y_i \in \text{span}\,\{p_0, \ldots, p_{i-1}\}$, $i = 1, 2, \ldots, j$.
2. $r_i \in \text{span}\,\{p_0, \ldots, p_i\}$ *for all* $i = 1, 2, \ldots, j$, *and*

$$r_i^\top v = 0, \quad \textit{for all } v \in \text{span}\,\{p_0, \ldots, p_{i-1}\} \textit{ and all } i = 1, 2, \ldots, j.$$

   (In particular, $r_i^\top r_l = 0$ *if* $0 \le l < i \le j$. *If* $j = n$, *then* $r_n = 0$.*)*
3. $\|r_i\| \le \|p_i\|$, $i = 0, 1, \ldots, j$.
4. $r_i^\top p_i = -\|r_i\|^2$, $i = 0, 1, \ldots, j$.
5. $p_i^\top \bar{H} p_k = 0$ *for all* $i, k = 0, 1, \ldots, j$ *with* $k \neq i$.
6. $p_i = -\sum_{k=0}^{i} (\|r_i\|^2 / \|r_k\|^2) r_k$, $i = 0, 1, \ldots, j$.
7. $$q(y_{i+1}) = q(y_i) - \frac{\|r_i\|^4}{2 p_i^\top \bar{H} p_i}, \quad i = 0, 1, \ldots, j-1.$$
8. $r_i^\top \bar{H} r_i \ge p_i^\top \bar{H} p_i$, $i = 0, 1, \ldots, j$.

Proof Since CG has not terminated prior to iteration j, (39) clearly holds. All properties then follow from the definition of the CG process, and most are proved in standard texts (see, for example, [25, Chapter 5]). Property 8 is less commonly used, so we provide a proof here.

The case $i = 0$ is immediate since $r_0 = -p_0$ and there is equality. When $i \ge 1$, we have:

$$p_i = -r_i + \frac{\|r_i\|^2}{\|r_{i-1}\|^2} p_{i-1} \;\Leftrightarrow\; r_i = -p_i + \frac{\|r_i\|^2}{\|r_{i-1}\|^2} p_{i-1}.$$

(Note that if iteration i is reached, we cannot have $\|r_{i-1}\| = 0$.) It follows that

$$\begin{aligned} r_i^\top \bar{H} r_i &= p_i^\top \bar{H} p_i - 2 \frac{\|r_i\|^2}{\|r_{i-1}\|^2} p_i^\top \bar{H} p_{i-1} + \frac{\|r_i\|^4}{\|r_{i-1}\|^4} p_{i-1}^\top \bar{H} p_{i-1} \\ &= p_i^\top \bar{H} p_i + \frac{\|r_i\|^4}{\|r_{i-1}\|^4} p_{i-1}^\top \bar{H} p_{i-1}, \end{aligned}$$

as $p_i^\top \bar{H} p_{i-1} = 0$ by Property 5 above. Since iteration i has been reached, p_{i-1} is a direction of positive curvature, and we obtain $r_i^\top \bar{H} r_i \ge p_i^\top \bar{H} p_i$, as required. □

We next address an important technical point about Algorithm 1: the test (6) to identify a direction of negative curvature for H after an insufficiently rapid rate of reduction in the residual norm $\|r_j\|$ has been observed. As written, the formula (6) suggests both that previous iterations y_i, $i = 1, 2, \ldots, j-1$ must be stored (or regenerated) and that additional matrix-vector multiplications (specifically, $\bar{H}(y_{j+1} - y_i)$, $i = 0, 1, \ldots$) must be performed. We show here that in fact (6) can be evaluated at essentially no cost, provided we store two extra scalars at each iteration of CG: the quantities α_k and $\|r_k\|^2$, for $k = 0, 1, \ldots, j$.

Lemma 8 *Suppose that Algorithm 1 computes iterates up to iteration $j+1$. Then, for any $i \in \{0, \ldots, j\}$, we can compute (6) as*

$$\frac{(y_{j+1} - y_i)^\top \bar{H}(y_{j+1} - y_i)}{\|y_{j+1} - y_i\|^2} = \frac{\sum_{k=i}^{j} \alpha_k \|r_k\|^2}{\sum_{\ell=0}^{j} \left[\sum_{k=\max\{\ell,i\}}^{j} \alpha_k \|r_k\|^2\right]^2 / \|r_\ell\|^2}.$$

Proof By definition, $y_{j+1} - y_i = \sum_{k=i}^{j} \alpha_k p_k$. By conjugacy of the p_k vectors, we have

$$(y_{j+1} - y_i)^\top \bar{H}(y_{j+1} - y_i) = \sum_{k=i}^{j} \alpha_k^2 p_k^\top \bar{H} p_k = \sum_{k=i}^{j} \alpha_k \|r_k\|^2, \tag{40}$$

where we used the definition of α_k to obtain the last equality. Now we turn our attention to the denominator. Using Property 6 of Lemma 7, we have that

$$y_{j+1} - y_i = \sum_{k=i}^{j} \alpha_k p_k = \sum_{k=i}^{j} \alpha_k \left(- \sum_{\ell=0}^{k} \frac{\|r_k\|^2}{\|r_\ell\|^2} r_\ell \right),$$

By rearranging the terms in the sum, we obtain

$$y_{j+1} - y_i = - \sum_{k=i}^{j} \sum_{\ell=0}^{k} \alpha_k \|r_k\|^2 \frac{r_\ell}{\|r_\ell\|^2} = - \sum_{\ell=0}^{j} \left[\sum_{k=\max\{\ell,i\}}^{j} \alpha_k \|r_k\|^2 \right] \frac{r_\ell}{\|r_\ell\|^2}.$$

Using the fact that the residuals $\{r_\ell\}_{\ell=0,1,\ldots,j}$ form an orthogonal set (by Property 2 of Lemma 7), we have that

$$\|y_{j+1} - y_i\|^2 = \sum_{\ell=0}^{j} \frac{1}{\|r_\ell\|^2} \left[\sum_{k=\max\{\ell,i\}}^{j} \alpha_k \|r_k\|^2 \right]^2.$$

Combining this with (40) gives the desired result. □

B Implementing Procedure 2 via Lanczos and conjugate gradient

In the first part of this appendix (Appendix B.1) we outline the randomized Lanczos approach and describe some salient convergence properties. The second part (Appendix B.2) analyzes the CG method (Algorithm 4) applied to a (possibly nonconvex) quadratic function with a random linear term. We show that the number of iterations required by CG to detect nonpositive curvature in an indefinite matrix is the same as the number required by Lanczos, when the two approaches are initialized in a consistent way, thereby proving Theorem 1. As a result, both techniques are implementations of Procedure 2 that satisfy Assumption 3, provided than an upper bound

M on $\|H\|$ is known. In the third part (Appendix B.3), we deal with the case in which a bound on $\|H\|$ is not known a priori, and describe a version of the randomized Lanczos scheme which obtains an overestimate of this quantity (to high probability) during its first phase of execution. The complexity of this version differs by only a modest multiple from the complexity of the original method, and still satisfies Assumption 3.

B.1 Randomized Lanczos

Consider first the Lanczos algorithm applied to a symmetric, n-by-n matrix $\bar{H}$ and a starting vector $b \in \mathbb{R}^n$ with $\|b\| = 1$. After $t+1$ iterations, Lanczos constructs a basis of the t-th Krylov subspace defined by

$$\mathcal{K}_t(b, \bar{H}) = \text{span}\{b, \bar{H}b, \ldots, \bar{H}^t b\}. \tag{41}$$

The Lanczos method can compute estimates of the minimum and maximum eigenvalues of $\bar{H}$. For $t = 0, 1, \ldots$, those values are given by

$$\xi_{\min}(\bar{H}, b, t) = \min_z \ z^\top \bar{H} z \quad \text{subject to } \|z\|_2 = 1, z \in \mathcal{K}_t(b, \bar{H}), \tag{42a}$$

$$\xi_{\max}(\bar{H}, b, t) = \max_z \ z^\top \bar{H} z \quad \text{subject to } \|z\|_2 = 1, z \in \mathcal{K}_t(b, \bar{H}). \tag{42b}$$

The Krylov subspaces satisfy a shift invariance property, that is, for any $\hat{H} = a_1 I + a_2 H$ with $(a_1, a_2) \in \mathbb{R}^2$, we have that

$$\mathcal{K}_t(b, \hat{H}) = \mathcal{K}_t(b, H) \quad \text{for } t = 0, 1, \ldots \tag{43}$$

Properties of the randomized Lanczos procedure are explored in [22]. The following key result is a direct consequence of Theorem 4.2(a) from the cited paper, along with the shift invariance property mentioned above.

Lemma 9 *Let $\bar{H}$ be an $n \times n$ symmetric matrix, let b be chosen from a uniform distribution over the sphere $\|b\| = 1$, and suppose that $\bar{\epsilon} \in [0, 1)$ and $\delta \in (0, 1)$ are given. Suppose that $\xi_{\min}(\bar{H}, b, k)$ and $\xi_{\max}(\bar{H}, b, k)$ are defined as in (42). Then after k iterations of randomized Lanczos, the following convergence condition holds:*

$$\lambda_{\max}(\bar{H}) - \xi_{\max}(\bar{H}, b, k) \le \bar{\epsilon}(\lambda_{\max}(\bar{H}) - \lambda_{\min}(\bar{H})) \quad \text{with probability at least } 1 - \delta, \tag{44}$$

provided k satisfies

$$k = n \quad \text{or} \quad 1.648\sqrt{n}\exp\left(-\sqrt{\bar{\epsilon}}(2k-1)\right) \le \delta. \tag{45}$$

A sufficient condition for (44) *thus is*

$$k \geq \min\left\{n, 1 + \left\lceil \frac{1}{4\sqrt{\bar{\epsilon}}} \ln(2.75n/\delta^2) \right\rceil\right\}. \tag{46}$$

Similarly, we have that

$$\xi_{\min}(\bar{H}, b, k) - \lambda_{\min}(\bar{H}) \leq \bar{\epsilon}(\lambda_{\max}(\bar{H}) - \lambda_{\min}(\bar{H})) \quad \text{with probability at least } 1 - \delta \tag{47}$$

for k satisfying the same conditions (45) *or* (46).

B.2 Lanczos and conjugate gradient as minimum eigenvalue oracles

Lemma 2 implies that using the Lanczos algorithm to generate the minimum eigenvalue of $\bar{H}$ from (42a) represents an instance of Procedure 2 satisfying Assumption 3. The sequence of iterates $\{z_t\}$ given by $z_0 = b$ and

$$z_{t+1} \in \arg\min_z \frac{1}{2} z^\top \bar{H} z \quad \text{subject to } \|z\|_2 = 1, z \in \mathcal{K}_t(b, \bar{H}), \text{ for } t = 0, 1, \ldots \tag{48}$$

eventually yields a direction of sufficient negative curvature, when such a direction exists.

Consider now Algorithm 4 applied to $\bar{H}$, and $g = -b$. By Property 2 of Lemma 7, we can see that if Algorithm 4 does not terminate with $j \leq t$, for some given index t, then y_{t+1}, r_{t+1}, and p_{t+1} are computed, and we have

$$\text{span}\{p_0, \ldots, p_i\} = \text{span}\{r_0, \ldots, r_i\} = \mathcal{K}_i(b, \bar{H}), \quad \text{for } i = 0, 1, \ldots, t, \tag{49}$$

because $\{r_\ell\}_{\ell=0}^i$ is a set of $i+1$ orthogonal vectors in $\mathcal{K}_i(b, \bar{H})$. Thus $\{p_0, \ldots, p_i\}$, $\{r_0, \ldots, r_i\}$, and $\{b, \bar{H}b, \ldots, \bar{H}^i b\}$ are all bases for $\mathcal{K}_i(b, \bar{H})$, $i = 0, 1, \ldots, t$. As long as they are computed, the iterates of Algorithm 4 satisfy

$$y_{t+1} := \arg\min_y \frac{1}{2} y^\top \bar{H} y - b^\top y \quad \text{subject to } y \in \mathcal{K}_t(b, \bar{H}), \text{ for } t = 0, 1, \ldots. \tag{50}$$

The sequences defined by (48) (for Lanczos) and (50) (for CG) are related via the Krylov subspaces. We have the following result about the number of iterations required by CG to detect non-positive-definiteness.

Theorem 5 *Consider applying Algorithm 4 to the quadratic function* (38)*, with* $g = -b$ *for some* b *with* $\|b\| = 1$*. Let* J *be the smallest value of* $t \geq 0$ *such that* $\mathcal{K}_t(b, \bar{H})$ *contains a direction of nonpositive curvature, so that* J *is also the smallest index* $t \geq 0$ *such that* $z_{t+1}^\top \bar{H} z_{t+1} \leq 0$*, where* $\{z_j\}$ *are the Lanczos iterates from* (48)*. Then Algorithm 4 terminates with* $j = J$*, with* $p_j^\top \bar{H} p_J \leq 0$.

Proof We consider first the case of $J = 0$. Then $z_1 = b/\|b\|$ and $b^\top \bar{H} b \leq 0$, so since $p_0 = -r_0 = b$, we have $p_0^\top \bar{H} p_0 \leq 0$, so the result holds in this case. We assume that $J \geq 1$ for the remainder of the proof.

Suppose first that Algorithm 4 terminates with $j = t$, for some t satisfying $1 \leq t \leq J$, because of a zero residual — $r_t = 0$—without having encountered nonpositive curvature. In that case, we can show that $\bar{H}^t b \in \text{span}\{b, \ldots, \bar{H}^{t-1} b\}$.

We can invoke (49) with t replaced by $t - 1$ since Algorithm 4 has not terminated at iteration $t - 1$. By the recursive definition of r_{t-1} within Algorithm 4, there exist coefficients τ_i and σ_i such that

$$r_{t-1} = -b + \sum_{i=1}^{t-1} \tau_i \bar{H}^i b, \quad p_{t-1} = \sum_{i=0}^{t-1} \sigma_i \bar{H}^i b.$$

Since $r_t = 0$, we have again from Algorithm 4 that

$$0 = r_{t-1} + \alpha_{t-1} \bar{H} p_{t-1} = -b + \sum_{i=1}^{t-1} (\tau_i + \alpha_{t-1}\sigma_{i-1}) \bar{H}^i b + \alpha_{t-1}\sigma_{t-1} \bar{H}^t b. \tag{51}$$

The coefficient $\alpha_{t-1}\sigma_{t-1}$ must be nonzero, because otherwise this expression would represent a nontrivial linear combination of the basis elements $\{b, \bar{H}b, \ldots, \bar{H}^{t-1}b\}$ of $\mathcal{K}_{t-1}(b, \bar{H})$ that is equal to zero. It follows from this observation and (51) that $\bar{H}^t b \in \text{span}\{b, \bar{H}b, \ldots, \bar{H}^{t-1}b\} = \mathcal{K}_{t-1}(b, \bar{H})$, as required.

Consequently,

$$\mathcal{K}_t(b, \bar{H}) = \text{span}\{b, \bar{H}b, \ldots, \bar{H}^t b\} = \text{span}\{b, \ldots, \bar{H}^{t-1} b\} = \mathcal{K}_{t-1}(b, \bar{H}).$$

By using a recursive argument on the definition of $\mathcal{K}_i(b, \bar{H})$ for $i = t, \ldots, J$, we arrive at $\mathcal{K}_{t-1}(b, \bar{H}) = \mathcal{K}_J(b, \bar{H})$. Thus there is a value of t smaller than J such that $\mathcal{K}_t(b, \bar{H})$ contains a direction of nonpositive curvature, contradicting our definition of J. Thus we cannot have termination of Algorithm 4 with $j \leq J$ unless $p_j^\top \bar{H} p_j \leq 0$.

Suppose next that CG terminates with $j = t$ for some $t > J$. It follows that $p_j^\top \bar{H} p_j > 0$ for all $j = 0, 1, \ldots, J$. By definition of J, there is a nonzero vector $z \in \mathcal{K}_J(b, \bar{H})$ such that $z^\top \bar{H} z \leq 0$. On the other hand, we have $\mathcal{K}_J(b, \bar{H}) = \text{span}\{p_0, p_1, \ldots, p_J\}$ by (49), thus we can write $z = \sum_{j=0}^{J} \gamma_j p_j$, for some scalars γ_j, $j = 0, 1, \ldots, J$. By Property 5 of Lemma 7, we then have

$$0 \geq z^\top \bar{H} z = \sum_{j=0}^{J} \gamma_j^2 p_j^\top \bar{H} p_j.$$

Since $p_j^\top \bar{H} p_j > 0$ for every $j = 0, 1, \ldots, J$, and not all γ_j can be zero (because $z \neq 0$), the final summation is strictly positive, a contradiction.

Suppose now that CG terminates at some $j < J$. Then $p_j^\top \bar{H} p_j \leq 0$, and since $p_j \in \mathcal{K}_j(b, \bar{H})$, it follows that $\mathcal{K}_j(b, \bar{H})$ contains a direction of nonpositive curvature, contradicting the definition of J.

We conclude that Algorithm 4 must terminate with $j = J$ and $p_J^\top \bar{H} p_J \leq 0$, as claimed. □

Theorem 5 is a generic result that does not require b to be chosen randomly. It does not guarantee that Lanczos will detect nonpositive curvature in $\bar{H}$ whenever present, because b could be orthogonal to the subspace corresponding to the nonpositive curvature, so the Lanczos subspace never intersects with the subspace of negative curvature. When b is chosen randomly from a uniform distribution over the unit ball, however, we can certify the performance of Lanczos, as we have shown in Lemma 2 based on Lemma 9 above. We can exploit Theorem 5 to obtain the same performance for CG, as stated in Theorem 1. We restate this result as a corollary, and prove it now.

Corollary 3 *Let b be distributed uniformly on the unit ball and H be a symmetric n-by-n matrix, with $\|H\| \leq M$. Given $\delta \in [0, 1)$, define*

$$\bar{J} := \min \left\{ n, 1 + \left\lceil \frac{\ln(2.75n/\delta^2)}{2} \sqrt{\frac{M}{\epsilon}} \right\rceil \right\}. \tag{52}$$

Consider applying Algorithm 4 with $\bar{H} := H + \frac{1}{2}\epsilon I$ and $g = -b$. Then, the following properties hold:

(a) If $\lambda_{\min}(H) < -\epsilon$, then with probability at least $1 - \delta$, there is some index $j \leq \bar{J}$ such that Algorithm 4 terminates with a direction p_j such that $p_j^\top H p_j \leq -(\epsilon/2)\|p_j\|^2$.

(b) if Algorithm 4 runs for $\bar{J}$ iterations without terminating, then with probability at least $1 - \delta$, we have that $\lambda_{\min}(H) \geq -\epsilon$.

Proof We will again exploit the invariance of the Krylov subspaces to linear shifts given by (43). This allows us to make inferences about the behavior of Algorithm 4 applied to $\bar{H}$ from the behavior of the Lanczos method applied to H, which has been described in Lemma 2.

Suppose first that $\lambda_{\min}(H) < -\epsilon$. By Lemma 2, we know that with probability at least $1 - \delta$, the Lanczos procedure returns a vector v such that $\|v\| = 1$ and $v^\top H v \leq -(\epsilon/2)$ after at most $\bar{J}$ iterations. Thus, for some $j \leq \bar{J}$, we have $v \in \mathcal{K}_j(b, H) = \mathcal{K}_j(b, \bar{H})$, and moreover $v^\top \bar{H} v \leq 0$ by definition of $\bar{H}$, so the Krylov subspace $\mathcal{K}_j(b, \bar{H})$ contains directions of nonpositive curvature, for some $j \leq \bar{J}$. It then follows from Theorem 5 that $p_j^\top \bar{H} p_j \leq 0$ for some $j \leq \bar{J}$. To summarize: If $\lambda_{\min}(H) < -\epsilon$, then with probability $1 - \delta$, Algorithm 4 applied to $\bar{H}$ and $g = -b$ will terminate with some p_j such that $p_j^\top \bar{H} p_j \leq 0$ for some j with $j \leq \bar{J}$. The proof of (a) is complete.

Suppose now that Algorithm 4 applied to $\bar{H}$ and $g = -b$ runs for $\bar{J}$ iterations without terminating, that is $p_j^\top \bar{H} p_j > 0$ for $j = 0, 1, \ldots, \bar{J}$. It follows from the logic of Theorem 5 that $\mathcal{K}_{\bar{J}}(b, \bar{H})$ contains no directions of nonpositive curvature for $\bar{H}$. Equivalently, there is no direction of curvature less than $-\epsilon/2$ for H in $\mathcal{K}_{\bar{J}}(b, H)$. By Lemma 2, this certifies with probability at least $1 - \delta$ that $\lambda_{\min}(H) \geq -\epsilon$, establishing (b). □

B.3 Randomized Lanczos with internal estimation of a bound on $\|H\|$

The methods discussed in Sect. B.2 assume knowledge of an upper bound on the considered matrix, denoted by M. When no such bound is known, we show here that it is possible to estimate it within the Lanczos procedure. Algorithm 5 details the method; we show that it can be used as an instance of Procedure 2 satisfying Assumption 3 when the optional parameter M is not provided.

Algorithm 5 consists in applying the Lanczos method on H starting with a random vector b. We first run Lanczos for j_M iterations, where j_M does not depend on any estimate on the minimum or maximum eigenvalue and instead targets a fixed accuracy. After this initial phase of j_M iterations, we have approximations of the extreme eigenvalues $\xi_{\max}(H, b, j_M)$ and $\xi_{\min}(H, b, j_M)$ from (42). An estimate M of $\|H\|$ is then given by:

$$M = 2 \max \{|\xi_{\max}(H, b, j_M)|, |\xi_{\min}(H, b, j_M)|\}. \tag{53}$$

We show below that $\|H\| \leq M \leq 2\|H\|$, with high probability. This value can then be used together with a tolerance ϵ to define a new iteration limit for the Lanczos method. After this new iteration limit is reached, we can either produce a direction of curvature at most $-\epsilon/2$, or certify with high probability that $\lambda_{\min}(H) \succeq -\epsilon I$—the desired outcomes for Procedure 2.

Algorithm 5 Lanczos Method with Upper Bound Estimation

Inputs: Symmetric matrix $H \in \mathbb{R}^{n \times n}$, tolerance $\epsilon > 0$.
Internal parameters: probability $\delta \in [0, 1)$, vector b uniformly distributed on the unit sphere.
Outputs: Estimate λ of $\lambda_{\min}(H)$ such that $\lambda \leq -\epsilon/2$, and vector v with $\|v\| = 1$ such that $v^\top H v = \lambda$ OR certificate that $\lambda_{\min}(H) \geq -\epsilon$. If the certificate is output, it is false with probability δ.
Set $j_M = \min \left\{ n, 1 + \left\lceil \frac{1}{2} \ln(25n/\delta^2) \right\rceil \right\}$.
Perform j_M iterations of Lanczos starting from b.
Compute $\xi_{\min}(H, b, j_M)$ and $\xi_{\max}(H, b, j_M)$, and set M according to (53).
Set $j_{\text{total}} = \min \left\{ j_M, 1 + \left\lceil \frac{1}{2} \ln(25n/\delta^2) \sqrt{\frac{M}{\epsilon}} \right\rceil \right\}$.
Perform $\max\{0, j_{\text{total}} - j_M\}$ additional iterations of Lanczos.
Compute $\xi_{\min}(H, b, j_{\text{total}})$.
if $\xi_{\min}(H, b, j_{\text{total}}) \leq -\epsilon/2$ **then**
  Output $\lambda = \xi_{\min}(H, b, j_{\text{total}})$ and a unit vector v such that $v^\top H v = \lambda$.
else
  Output $\lambda = \xi_{\min}(H, b, j_{\text{total}})$ as a certificate that $\lambda_{\min}(H) \geq -\epsilon$. This certificate is false with probability δ.
end if

Algorithm 5 could be terminated earlier, in fewer than j_{total} iterations, when a direction of sufficient negative is encountered. For simplicity, we do not consider this feature, but observe that it would not affect the guarantees of the method, described in Lemma 10 below.

Lemma 10 *Consider Algorithm 5 with input parameters H and ϵ, and internal parameters δ and b. The method outputs a value λ such that*

$$\lambda \leq \lambda_{\min}(H) + \frac{\epsilon}{2} \tag{54}$$

in at most

$$\min\left\{n, 1 + \max\left\{\left\lceil \frac{1}{2}\ln(25n/\delta^2)\right\rceil, \left\lceil \frac{1}{2}\ln(25n/\delta^2)\sqrt{\frac{2\|H\|}{\epsilon}}\right\rceil\right\}\right\} \tag{55}$$

matrix-vector multiplications by H, with probability at least $1-\delta$.

Proof We begin by showing that the first phase of Algorithm 5 yields an accurate estimate of $\|H\|$ with high probability. We assume that $\|H\| > 0$ as the result is trivially true otherwise. By setting $\delta \leftarrow \delta/3$ and $\bar{\epsilon} = \frac{1}{4}$ in Lemma 9, we obtain that the following inequalities hold, each with probability at least $1-\delta/3$:

$$\xi_{\max}(H, b, j_M) \geq \lambda_{\max}(H) - \tfrac{1}{4}(\lambda_{\max}(H) - \lambda_{\min}(H)), \tag{56a}$$

$$\xi_{\min}(H, b, j_M) \leq \lambda_{\min}(H) + \tfrac{1}{4}(\lambda_{\max}(H) - \lambda_{\min}(H)). \tag{56b}$$

We consider the various possibilities for $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ separately, showing in each case that M defined by (53) has $\|H\| \leq M \leq 2\|H\|$.

- When $\lambda_{\max}(H) \geq \lambda_{\min}(H) \geq 0$, we have $\xi_{\max}(H, b, j_M) \geq \frac{3}{4}\lambda_{\min}(H)$ and $0 \leq \xi_{\min}(H, b, j_M) \leq \xi_{\max}(H, b, j_M)$, so that

$$M = 2\xi_{\max}(H, b, j_M) \geq \frac{3}{2}\lambda_{\max}(H) = \frac{3}{2}\|H\|,$$
$$M = 2\xi_{\max}(H, b, j_M) \leq 2\lambda_{\max}(H) = 2\|H\|,$$

  as required.
- When $\lambda_{\min}(H) \leq \lambda_{\max}(H) \leq 0$, we have $\xi_{\min}(H, b, j_M) \leq \frac{3}{4}\lambda_{\min}(H) \leq 0$ and $\xi_{\min}(H, b, j_M) \leq \xi_{\max}(H, b, j_M) \leq 0$, so that

$$M = 2|\xi_{\min}(H, b, j_M)| \geq \frac{3}{2}|\lambda_{\min}(H)| = \frac{3}{2}\|H\|,$$
$$M = 2|\xi_{\min}(H, b, j_M)| \leq 2|\lambda_{\min}(H)| = 2\|H\|,$$

  as required.
- When $\lambda_{\min}(H) \leq 0 \leq \lambda_{\max}(H)$ and $-\lambda_{\min}(H) \leq \lambda_{\max}(H)$, we have $\lambda_{\max}(H) - \lambda_{\min}(H) \leq 2\lambda_{\max}(H)$, so from (56a), it follows that $\xi_{\max}(H, b, j_M) \geq \frac{1}{2}\lambda_{\max}(H) = \frac{1}{2}\|H\|$, and so

$$M \geq 2\xi_{\max}(H, b, j_M) \geq \|H\|,$$
$$M = 2\max\{|\xi_{\max}(H, b, j_M)|, |\xi_{\min}(H, b, j_M)|\}$$
$$\leq 2\max\{|\lambda_{\max}(H)|, |\lambda_{\min}(H)|\} = 2\|H\|,$$

  as required.

– When $\lambda_{\min}(H) \le 0 \le \lambda_{\max}(H)$ and $-\lambda_{\min}(H) \ge \lambda_{\max}(H)$, we have $\lambda_{\max}(H) - \lambda_{\min}(H) \le -2\lambda_{\min}(H)$, so from (56b), it follows that $\xi_{\min}(H, b, j_M) \le \frac{1}{2}\lambda_{\min}(H) \le 0$, and so

$$\begin{aligned} M &\ge 2|\xi_{\min}(H, b, j_M)| \ge |\lambda_{\min}(H)| = \|H\|, \\ M &= 2\max\{|\xi_{\max}(H, b, j_M)|, |\xi_{\min}(H, b, j_M)|\} \\ &\le 2\max\{|\lambda_{\max}(H)|, |\lambda_{\min}(H)|\} = 2\|H\|, \end{aligned}$$

as required.

Since each of the bounds in (56) holds with probability at least $1 - \delta/3$, both hold with probability at least $1 - 2\delta/3$, by a union bound argument.

We finally consider the complete run of Algorithm 5, which requires j_{total} iterations of Lanczos. If our estimate M is accurate, we have by setting $\delta \leftarrow \delta/3$ and $M \leftarrow 2\|H\|$ in Lemma 2 that $\lambda = \xi_{\min}(H, b, j_{\text{total}})$ satisfies (54) with probability $1 - \delta/3$. By using a union bound to combine this probability with the probabilities of estimating M appropriately, we obtain the probability of at least $1 - \delta$.

In conclusion, Algorithm 5 runs j_{total} iterations of Lanczos (each requiring one matrix-vector multiplication by H) and terminates correctly with probability at least $1 - \delta$. □

The lemma above shows that Algorithm 5 is an instance of Procedure 2 that does not require an a priori bound on $\|H\|$. Assuming $\|H\| \le U_H$, we further observe that Algorithm 5 satisfies the conditions of Assumption 3 with $\mathcal{C}_{\text{meo}} = \frac{\ln(25n/\delta^2)}{\sqrt{2}}\sqrt{U_H}$, which is within a modest constant multiple of the one obtained for the Lanczos method with knowledge of $\|H\|$ or U_H.

C Proof of Theorem 2

Proof The proof proceeds by contradiction: If we assume that all conditions specified in the statement of the theorem hold, and in addition that

$$\frac{(y_{\hat{\jmath}+1} - y_i)^\top \bar{H}(y_{\hat{\jmath}+1} - y_i)}{\|y_{\hat{\jmath}+1} - y_i\|^2} \ge \epsilon, \quad \text{for all } i = 0, 1, \ldots, \hat{J} - 1, \tag{57}$$

then we must have

$$\|r_{\hat{\jmath}}\| \le \sqrt{T}\tau^{\hat{\jmath}/2}\|r_0\|, \tag{58}$$

contradicting (11). Note that

$$\frac{(y_{\hat{\jmath}+1} - y_{\hat{\jmath}})^\top \bar{H}(y_{\hat{\jmath}+1} - y_{\hat{\jmath}})}{\|y_{\hat{\jmath}+1} - y_{\hat{\jmath}}\|^2} = \frac{\alpha_{\hat{\jmath}} p_{\hat{\jmath}}^\top \bar{H}(\alpha_{\hat{\jmath}} p_{\hat{\jmath}})}{\|\alpha_{\hat{\jmath}} p_{\hat{\jmath}}\|^2} = \frac{p_{\hat{\jmath}}^\top \bar{H} p_{\hat{\jmath}}}{\|p_{\hat{\jmath}}\|^2} \ge \epsilon \tag{59}$$

by assumption, therefore we can consider (57) to hold for $i = 0, 1, \ldots, \hat{J}$. Moreover, recalling the definition (38) of the quadratic function q, we have for any $i = 0, 1, \ldots, \hat{J}$ that

$$q(y_{\hat{J}+1}) = q(y_i) + \nabla q(y_i)^\top (y_{\hat{J}+1} - y_i) + \frac{1}{2}(y_{\hat{J}+1} - y_i)^\top \bar{H}(y_{\hat{J}+1} - y_i).$$

Thus, (57) can be reformulated as

$$q(y_{\hat{J}+1}) \geq q(y_i) + r_i^\top (y_{\hat{J}+1} - y_i) + \frac{\epsilon}{2}\|y_{\hat{J}+1} - y_i\|^2, \quad \text{for all } i = 0, 1, \ldots, \hat{J}, \tag{60}$$

where we used $\nabla q(y_i) = r_i$ and the definitions (57), (59) of strong convexity along the directions $y_{\hat{J}+1} - y_i$. In the remainder of the proof, and similarly to [5, Proof of Proposition 1], we will show that (60) leads to the contradiction (58), thus proving that (12) holds.

We define the sequence of functions Φ_j, $j = 0, 1, \ldots \hat{J}$ as follows:

$$\Phi_0(z) := q(y_0) + \frac{\epsilon}{2}\|z - y_0\|^2,$$

and for $j = 0, \ldots, \hat{J} - 1$:

$$\Phi_{j+1}(z) := \tau \Phi_j(z) + (1 - \tau)\left(q(y_j) + r_j^\top (z - y_j) + \frac{\epsilon}{2}\|z - y_j\|^2\right). \tag{61}$$

Since each Φ_j is a quadratic function with Hessian ϵI, it can be written as follows:

$$\Phi_j(z) \;=\; \Phi_j^* + \frac{\epsilon}{2}\|z - v_j\|^2, \tag{62}$$

where v_j is the unique minimizer of Φ_j, and $\Phi_j^* = \Phi_j(v_j)$ is the minimum value of Φ_j. (Note that $v_0 = y_0 = 0$ and $\Phi_0^* = q(y_0) = 0$.)

Defining

$$\psi(y) := q(y_0) - q(y) + \frac{\epsilon}{2}\|y - y_0\|^2 = \Phi_0(y) - q(y), \tag{63}$$

we give a short inductive argument to establish that

$$\Phi_j(y_{\hat{J}+1}) \leq q(y_{\hat{J}+1}) + \tau^j \psi(y_{\hat{J}+1}), \quad j = 0, 1, \ldots, \hat{J}. \tag{64}$$

For $j = 0$, (64) holds because $\Phi_0(y) = q(y) + \psi(y)$ by definition. Assuming that (64) holds for some index $j \geq 0$, we find by first applying (61) (with $z = y_{\hat{J}+1}$) and then (60) (with $i = j$) that

$$\begin{aligned}\Phi_{j+1}(y_{\hat{J}+1}) &= \tau \Phi_j(y_{\hat{J}+1}) + (1 - \tau)\left(q(y_j) + r_j^\top (y_{\hat{J}+1} - y_j) + \frac{\epsilon}{2}\|y_{\hat{J}+1} - y_j\|^2\right) \\ &\leq \tau \Phi_j(y_{\hat{J}+1}) + (1 - \tau) q(y_{\hat{J}+1}).\end{aligned}$$

Thus, we have

$$\begin{aligned}\Phi_{j+1}(y_{\hat{J}+1}) &\leq \tau\Phi_j(y_{\hat{J}+1}) + (1-\tau)q(y_{\hat{J}+1}) \\ &\leq \tau q(y_{\hat{J}+1}) + \tau^{j+1}\psi(y_{\hat{J}+1}) + (1-\tau)q(y_{\hat{J}+1}) \quad \text{from (64)} \\ &= q(y_{\hat{J}+1}) + \tau^{j+1}\psi(y_{\hat{J}+1}),\end{aligned}$$

which proves (64) for $j+1$, and thus completes the inductive argument.

We next prove another technical fact about the relationship between $q(y_j)$ and Φ_j^*, namely,

$$q(y_j) \leq \Phi_j^*, \quad j = 0, 1, \ldots, \hat{J}. \tag{65}$$

We establish this result by an inductive argument that is quite lengthy and technical; we note the termination of this phase of the proof clearly below.

This result trivially holds (with equality) at $j = 0$. Supposing that it holds for some $j = 0, 1, \ldots, \hat{J} - 1$, we will prove that it also holds for $j+1$.

By using Properties 7 and 8 of Lemma 7, and also $\|\bar{H}r_j\| \leq (M + 2\epsilon)\|r_j\|$, we have

$$q(y_{j+1}) = q(y_j) - \frac{\|r_j\|^4}{2\, p_j^\top \bar{H} p_j} \leq q(y_j) - \frac{\|r_j\|^4}{2\, r_j^\top \bar{H} r_j} \leq q(y_j) - \frac{\|r_j\|^2}{2(M+2\epsilon)}.$$

It follows from induction hypothesis $q(y_j) \leq \Phi_j^*$ that

$$\begin{aligned}q(y_{j+1}) &\leq q(y_j) - \frac{\|r_j\|^2}{2(M+2\epsilon)} = \tau q(y_j) + (1-\tau)q(y_j) - \frac{\|r_j\|^2}{2(M+2\epsilon)} \\ &\leq \tau\Phi_j^* + (1-\tau)q(y_j) - \frac{\|r_j\|^2}{2(M+2\epsilon)}.\end{aligned} \tag{66}$$

By taking the derivative in (61), and using (62), we obtain

$$\begin{aligned}&\nabla\Phi_{j+1}(z) = \tau\nabla\Phi_j(z) + (1-\tau)\left[r_j + \epsilon(z - y_j)\right] \\ &\quad\Rightarrow\quad \epsilon(z - v_{j+1}) = \epsilon\tau(z - v_j) + (1-\tau)\left(r_j + \epsilon(z - y_j)\right).\end{aligned}$$

By rearranging the above relation (and noting that the z terms cancel out), we obtain:

$$v_{j+1} \;=\; \tau v_j - \frac{1-\tau}{\epsilon} r_j + (1-\tau) y_j. \tag{67}$$

It follows from this expression together with Properties 1 and 2 of Lemma 7 that

$$v_j \in \text{span}\left\{p_0, p_1, \ldots, p_{j-1}\right\}, \quad j = 1, 2, \ldots, \hat{J}. \tag{68}$$

(The result holds for $j = 1$, from (67) we have $v_1 \in \text{span}\{v_0, r_0, y_0\} = \text{span}\{r_0\} = \text{span}\{p_0\}$, and an induction based on (67) can be used to establish (68) for the other

values of j.) By combining the expressions (62) for Φ_j, Φ_{j+1} with the recurrence formula (61) for Φ_{j+1}, we obtain

$$\begin{aligned}\Phi_{j+1}^* + \frac{\epsilon}{2}\|y_j - v_{j+1}\|^2 &= \Phi_{j+1}(y_j) \\ &= \tau\Phi_j(y_j) + (1-\tau)q(y_j) \\ &= \tau\left(\Phi_j^* + \frac{\epsilon}{2}\|y_j - v_j\|^2\right) + (1-\tau)q(y_j)\end{aligned}$$

and therefore

$$\Phi_{j+1}^* = \tau\left(\Phi_j^* + \frac{\epsilon}{2}\|y_j - v_j\|^2\right) + (1-\tau)q(y_j) - \frac{\epsilon}{2}\|y_j - v_{j+1}\|^2. \tag{69}$$

On the other hand, we have by (67) that

$$\begin{aligned}\|y_j - v_{j+1}\|^2 &= \left\|\tau(y_j - v_j) + \frac{1-\tau}{\epsilon}r_j\right\|^2 \\ &= (\tau^2\|y_j - v_j\|^2 + \frac{(1-\tau)^2}{\epsilon^2}\|r_j\|^2 + \frac{2}{\epsilon}(1-\tau)\tau r_j^\top(y_j - v_j) \\ &= \tau^2\|y_j - v_j\|^2 + \frac{(1-\tau)^2}{\epsilon^2}\|r_j\|^2,\end{aligned} \tag{70}$$

where the last relation comes from $r_j \perp \text{span}\{p_0, \ldots, p_{j-1}\}$ (Property 2 of Lemma 7) and (68) in the case $j \geq 1$, and immediately in the case $j = 0$, since $y_0 = v_0 = 0$. By combining (69) and (70), we arrive at:

$$\begin{aligned}\Phi_{j+1}^* &= \tau\left(\Phi_j^* + \frac{\epsilon}{2}\|y_j - v_j\|^2\right) + (1-\tau)q(y_j) - \frac{\epsilon}{2}\|y_j - v_{j+1}\|^2 \\ &= \tau\left(\Phi_j^* + \frac{\epsilon}{2}\|y_j - v_j\|^2\right) + (1-\tau)q(y_j) \\ &\quad - \frac{\epsilon}{2}\tau^2\|y_j - v_j\|^2 - \frac{(1-\tau)^2}{2\epsilon}\|r_j\|^2 \\ &= \tau\Phi_j^* + \frac{\epsilon}{2}\left[\tau - \tau^2\right]\|y_j - v_j\|^2 + (1-\tau)q(y_j) - \frac{(1-\tau)^2}{2\epsilon}\|r_j\|^2 \\ &= \tau\Phi_j^* + \frac{\epsilon}{2}(1-\tau)\tau\|y_j - v_j\|^2 + (1-\tau)q(y_j) - \frac{(1-\tau)^2}{2\epsilon}\|r_j\|^2 \\ &\geq \tau\Phi_j^* + (1-\tau)q(y_j) - \frac{(1-\tau)^2}{2\epsilon}\|r_j\|^2 \\ &\geq q(y_{j+1}) + \frac{1}{2(M+2\epsilon)}\|r_j\|^2 - \frac{(1-\tau)^2}{2\epsilon}\|r_j\|^2.\end{aligned} \tag{71}$$

where the last inequality comes from (66). By using the definitions of τ and κ in Algorithm 1, we have

$$\frac{(1-\tau)^2}{2\epsilon} = \frac{1}{2\epsilon(\sqrt{\kappa}+1)^2} \leq \frac{1}{2\epsilon\kappa} = \frac{1}{2(M+2\epsilon)}.$$

It therefore follows from (71) that $q(y_{j+1}) \leq \Phi^*_{j+1}$. At this point, we have shown that when $q(y_j) \leq \Phi^*_j$ for $j = 0, 1, \ldots, \hat{J}-1$, it follows that $q(y_{j+1}) \leq \Phi^*_{j+1}$, establishing the inductive step. As a result, our proof of (65) is complete.

By substituting $j = \hat{J}$ into (65), we obtain $q(y_{\hat{J}}) \leq \Phi^*_{\hat{J}}$, which in combination with (64) with $j = \hat{J}$, and the definition (62), yields

$$q(y_{\hat{J}}) - q(y_{\hat{J}+1}) \leq \Phi^*_{\hat{J}} - q(y_{\hat{J}+1}) \leq \Phi_{\hat{J}}(y_{\hat{J}+1}) - q(y_{\hat{J}+1}) \leq \tau^{\hat{J}}\psi(y_{\hat{J}+1}). \quad (72)$$

By substitution from (63), we obtain

$$q(y_{\hat{J}}) - q(y_{\hat{J}+1}) \;\leq\; \tau^{\hat{J}}\left(q(y_0) - q(y_{\hat{J}+1}) + \frac{\epsilon}{2}\|y_0 - y_{\hat{J}+1}\|^2\right). \quad (73)$$

We now depart from the analysis of [5], and complete the proof of this result by expressing (73) in terms of residual norms. On the left-hand side, we have

$$\begin{aligned} q(y_{\hat{J}}) - q(y_{\hat{J}+1}) &= r_{\hat{J}+1}^\top (y_{\hat{J}} - y_{\hat{J}+1}) + \frac{1}{2}(y_{\hat{J}} - y_{\hat{J}+1})^\top \bar{H}(y_{\hat{J}} - y_{\hat{J}+1}) \\ &= \frac{1}{2}(y_{\hat{J}} - y_{\hat{J}+1})^\top \bar{H}(y_{\hat{J}} - y_{\hat{J}+1}) \end{aligned}$$

because $r_{\hat{J}+1}^\top (y_{\hat{J}} - y_{\hat{J}+1}) = r_{\hat{J}+1}^\top (\alpha_{\hat{J}} p_{\hat{J}}) = 0$ by Lemma 7, Property 2. We thus have from (59) that

$$\begin{aligned} q(y_{\hat{J}}) - q(y_{\hat{J}+1}) &\geq \frac{\epsilon}{2}\|y_{\hat{J}} - y_{\hat{J}+1}\|^2 \\ &= \frac{\epsilon}{2}\|\alpha_{\hat{J}} p_{\hat{J}}\|^2 \\ &\geq \frac{\epsilon}{2(M+2\epsilon)^2}\|\bar{H}(\alpha_{\hat{J}} p_{\hat{J}})\|^2 \qquad (\text{since } \|\bar{H} p_{\hat{J}}\| \leq (M+2\epsilon)\|p_{\hat{J}}\|) \\ &= \frac{\epsilon}{2(M+2\epsilon)^2}\|\bar{H}(y_{\hat{J}} - y_{\hat{J}+1})\|^2 \\ &= \frac{\epsilon}{2(M+2\epsilon)^2}\|r_{\hat{J}} - r_{\hat{J}+1}\|^2 \qquad (\text{since } r_j = g + \bar{H} y_j) \\ &= \frac{\epsilon}{2(M+2\epsilon)^2}(\|r_{\hat{J}}\|^2 + \|r_{\hat{J}+1}\|^2) \qquad (\text{by Lemma 7, Property 2}) \\ &\geq \frac{\epsilon}{2(M+2\epsilon)^2}\|r_{\hat{J}}\|^2, \end{aligned} \quad (74)$$

On the right-hand side of (73), because of the strong convexity condition (60) at $i = 0$, we have

$$\begin{aligned} q(y_0) - q(y_{\hat{J}+1}) + \frac{\epsilon}{2}\|y_0 - y_{\hat{J}+1}\|^2 &\leq -\nabla q(y_0)^\top (y_{\hat{J}+1} - y_0) \\ &= -r_0^\top (y_{\hat{J}+1} - y_0) \leq \|r_0\|\|y_{\hat{J}+1} - y_0\|. \end{aligned}$$

Moreover, we have

$$\|y_{\hat{J}+1} - y_0\| \;=\; \left\| \sum_{i=0}^{\hat{J}} \alpha_i p_i \right\| \;\leq\; \sum_{i=0}^{\hat{J}} \alpha_i \|p_i\| \;=\; \sum_{i=0}^{\hat{J}} \frac{\|r_i\|^2}{p_i^\top \bar{H} p_i} \|p_i\|,$$

where the last relation follows from the definition of α_i. By combining these last two bounds, and using Property 3 of Lemma 7, we obtain

$$q(y_0) - q(y_{\hat{J}+1}) + \frac{\epsilon}{2}\|y_0 - y_{\hat{J}+1}\|^2 \;\leq\; \|r_0\| \sum_{i=0}^{\hat{J}} \|r_i\| \frac{\|p_i\|^2}{p_i^\top \bar{H} p_i} \;\leq\; \|r_0\| \frac{1}{\epsilon} \sum_{i=0}^{\hat{J}} \|r_i\|, \tag{75}$$

because $p_j^\top \bar{H} p_j \geq \epsilon \|p_j\|^2$ for $j = 0, 1, \ldots, \hat{J}$ by assumption.

To bound the sum in (75), we recall that since Algorithm 1 did not terminate until iteration $\hat{J}$, the residual norms $\|r_i\|$ at all iterations $i = 0, 1, \ldots, \hat{J} - 1$ must have decreased at the expected convergence rate. In particularly, we have $\|r_i\| \leq \sqrt{T}\tau^{i/2}\|r_0\|$ for the *possibly smaller* versions of $\sqrt{T}$ and τ that prevailed at iteration i, so certainly $\|r_i\| \leq \sqrt{T}\tau^{i/2}\|r_0\|$ for the final values of these parameters. Thus for $i = 0, 1, \ldots, \hat{J} - 1$, we have

$$\|r_i\| \leq \sqrt{T}\tau^{i/2}\|r_0\| \leq \tau^{(i-\hat{J})/2}\|r_{\hat{J}}\|,$$

where we used $\|r_{\hat{J}}\| \geq \sqrt{T}\tau^{\hat{J}/2}\|r_0\|$ (from (11)) for the last inequality. Observing that this bound also holds (trivially) for $i = \hat{J}$, we obtain by substituting in (75) that

$$\begin{aligned} q(y_0) - q(y_{\hat{J}+1}) + \frac{\epsilon}{2}\|y_0 - y_{\hat{J}+1}\|^2 &\leq \|r_0\| \frac{1}{\epsilon} \sum_{i=0}^{\hat{J}} \tau^{(i-\hat{J})/2}\|r_{\hat{J}}\| \\ &\leq \|r_0\| \frac{\tau^{-\hat{J}/2}}{\epsilon}\|r_{\hat{J}}\| \sum_{i=0}^{\hat{J}} (\sqrt{\tau})^i \\ &\leq \|r_0\| \frac{\tau^{-\hat{J}/2}}{\epsilon}\|r_{\hat{J}}\| \frac{1}{1-\sqrt{\tau}}. \end{aligned} \tag{76}$$

Applying successively (74), (73) and (76) finally yields:

$$\begin{aligned} \frac{\epsilon}{2(M+2\epsilon)^2}\|r_{\hat{J}}\|^2 &\leq q(y_{\hat{J}}) - q(y_{\hat{J}+1}) \\ &\leq \tau^{\hat{J}} \left(q(y_0) - q(y_{\hat{J}+1}) + \frac{\epsilon}{2}\|y_0 - y_{\hat{J}+1}\|^2 \right) \\ &\leq \tau^{\hat{J}} \|r_0\| \|r_{\hat{J}}\| \frac{\tau^{-\hat{J}/2}}{\epsilon} \frac{1}{1-\sqrt{\tau}}. \end{aligned}$$

After rearranging this inequality and dividing by $\|r_{\hat{j}}\| > 0$, we obtain

$$\|r_{\hat{j}}\| \leq \frac{2(M+2\epsilon)^2}{\epsilon^2} \frac{\tau^{\hat{j}/2}}{1-\sqrt{\tau}} \|r_0\| = \sqrt{T}\tau^{\hat{j}/2}\|r_0\|. \tag{77}$$

We have thus established (58) which, as we noted earlier, contradicts (11). Thus (57) cannot be true, so we have established (12), as required. □

References

1. Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., Ma, T.: Finding approximate local minima faster than gradient descent. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017), PMLR (2017)
2. Allen-Zhu, Z., Li, Y.: NEON2: finding local minima via first-order oracles. In: Proceedings of the 32nd Conference on Neural Information Processing Systems (2018)
3. Birgin, E.G., Martínez, J.M.: The use of quadratic regularization with a cubic descent condition for unconstrained optimization. SIAM J. Optim. **27**, 1049–1074 (2017)
4. Bubeck, S.: Convex optimization: algorithms and complexity. Found. Trends© Mach. Learn. **8**, 231–357 (2015)
5. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: "Convex until proven guilty": dimension-free acceleration of gradient descent on non-convex functions. In: International Conference on Machine Learning, vol. 70, 6–11 August 2017, International Convention Centre, Sydney, Australia, PMLR, pp. 654–663 (2017)
6. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Accelerated methods for non-convex optimization. SIAM J. Optim. **28**, 1751–1772 (2018)
7. Cartis, C., Gould, N.I.M., Toint, P.L.: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. SIAM J. Optim. **20**, 2833–2852 (2010)
8. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. Math. Program. **127**, 245–295 (2011)
9. Cartis, C., Gould, N.I.M., Toint, P.L.: Optimal Newton-type methods for nonconvex optimization. Technical Report naXys-17-2011, Department of Mathematics, FUNDP, Namur (B) (2011)
10. Cartis, C., Gould, N.I.M., Toint, P.L.: Complexity bounds for second-order optimality in unconstrained optimization. J. Complex. **28**, 93–108 (2012)
11. Cartis, C., Gould, N.I.M., Toint, P.L.: Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. arXiv:1709.07180 (2017)
12. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia (2000)
13. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}\left(\epsilon^{-3/2}\right)$ for nonconvex optimization. Math. Program. **162**, 1–32 (2017)
14. Curtis, F.E., Robinson, D.P., Samadi, M.: An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. IMA J. Numer. Anal. (2018). https://doi.org/10.1093/imanum/dry022
15. Dembo, R.S., Steihaug, T.: Truncated-Newton algorithms for large-scale unconstrained optimization. Math. Program. **26**, 190–212 (1983)
16. Fasano, G., Lucidi, S.: A nonmonotone truncated Newton–Krylov method exploiting negative curvature directions, for large-scale unconstrained optimization. Optim. Lett. **3**, 521–535 (2009)
17. Gould, N.I.M., Lucidi, S., Roma, M., Toint, P.L.: Exploiting negative curvature directions in linesearch methods for unconstrained optimization. Optim. Methods Softw. **14**, 75–98 (2000)
18. Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Frontiers in Applied Mathematics, 2nd edn. SIAM, Philadelphia (2008)
19. Jin, C., Netrapalli, P., Jordan, M.I.: Accelerated gradient descent escapes saddle points faster than gradient descent. In: Proceedings of the 31st Conference on Learning Theory, PMLR, pp. 1042–1085 (2018)

20. Karimi, S., Vavasis, S.A.: A unified convergence bound for conjugate gradient and accelerated gradient. arXiv:1605.00320 (2016)
21. Karimi, S., Vavasis, S.A.: A single potential governing convergence of conjugate gradient, accelerated gradient and geometric descent. arXiv:1712.09498 (2017)
22. Kuczyński, J., Woźniakowski, H.: Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. SIAM J. Matrix Anal. Appl. **13**, 1094–1122 (1992)
23. Martínez, J.M., Raydan, M.: Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. J. Glob. Optim. **68**, 367–385 (2017)
24. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. **108**, 177–205 (2006)
25. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)
26. Royer, C.W., Wright, S.J.: Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. SIAM J. Optim. **28**, 1448–1477 (2018)
27. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. **20**, 626–637 (1983)
28. Xu, Y., Jin, R., Yang, T.: First-order stochastic algorithms for escaping from saddle points in almost linear time. In: Proceedings of the 32nd Conference on Neural Information Processing Systems (2018)