Globally linearly convergent nonlinear conjugate gradients without Wolfe line search

Arnold Neumaier
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: Arnold.Neumaier@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~neum`

Morteza Kimiaei
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: kimiaeim83@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~kimiaei`

Behzad Azmi
Department of Mathematics and Statistics
University of Konstanz
D-78457 Konstanz, Germany
email: behzad.azmi@uni-konstanz.de

Abstract. This paper introduces a new nonlinear conjugate gradient (CG) method using an efficient gradient-free line search method. Unless function values diverge to $-\infty$, global convergence to a stationary point is proved for continuously differentiable objective functions with Lipschitz continuous gradient, and global linear convergence if this stationary point is a strong local minimizer. The n-iterations termination of the method also is addressed for strictly convex quadratic functions posed in $\mathbb{R}^n$. Moreover, $\mathcal{O}(\varepsilon^{-2})$ complexity bounds for the number of functions and gradient evaluations are derived for approximating a stationary point. These will be improved to $\mathcal{O}(\log \varepsilon^{-1})$ for objective functions having only a strong minimizer and no other stationary points. A measure for zigzagging strength is also introduced. Relying on this, a minimal zigzagging strength of the method is shown. Numerical results on the unconstrained `CUTEst` test problems illustrate that the new method is competitive with the best nonlinear state-of-the-art CG methods proposed in the literature.

Keywords. Unconstrained optimization; nonlinear conjugate gradients; complexity; zigzagging

2000 AMS Subject Classification: primary 90C56.

Acknowledgment The second author acknowledges the financial support of the Austrian Science Foundation under Project No. P 34317. Earlier versions of this paper benefitted from discussions with Waltraud Huyer and Hermann Schichl.

May 19, 2023

1 Introduction

This paper discusses a new nonlinear conjugate gradient (CG) method for the unconstrained optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array} \tag{1}$$

satisfying the following assumptions (A1) and (A2):
(A1) The function f is bounded below, i.e.,

$$\underline{f} := \inf_{\alpha \geq 0} f(x(\alpha)) > -\infty. \tag{2}$$

(A2) The function f is continuously differentiable on $\mathbb{R}^n$, and its gradient $g(x) = f'(x)^T$ is Lipschitz continuous with Lipschitz constant $\overline{\gamma}$, i.e.,

$$\|g(x) - g(x')\|_* \leq \overline{\gamma}\|x - x'\| \quad \text{with } \overline{\gamma} > 0. \tag{3}$$

Here $\|\cdot\|$ is an arbitrary norm and $\|\cdot\|_*$ is its dual norm, defined by

$$\|y\|_* := \sup\{y^T x \mid \|x\| \leq 1\}.$$

They satisfy the **generalized Cauchy–Schwarz inequality**

$$|y^T s| \leq \|y\|_* \|s\| \quad \text{for } s, y \in \mathbb{R}^n.$$

We call a point $\widehat{x} \in \mathbb{R}^n$ a **strong local minimizer** of f if f is twice continuously differentiable in a neighborhood of $\widehat{x}$, the gradient $g(\widehat{x})$ of f at $\widehat{x}$ vanishes, and the Hessian $G(\widehat{x})$ of f at $\widehat{x}$ is positive definite.

For any sequence $x^0, x^1, x^2, \ldots$ of feasible points and $\ell = 0, 1, 2, \ldots$, we write

$$\begin{aligned} f_\ell &:= f(x^\ell), \quad g^\ell := g(x^\ell), \\ s^\ell &:= x^{\ell+1} - x^\ell, \quad y^\ell := g^{\ell+1} - g^\ell. \end{aligned} \tag{4}$$

1.1 Related work

The unconstrained optimization problem has a very long history which we do not trace here; see, instead, the books by Fletcher [13] or Nocedal & Wright [31]. Here we only discuss the state-of-the-art concerning nonlinear conjugate gradient methods used in this context.

1.1.1 Standard convergence theory and complexity

We say that the search directions p^ℓ ($\ell \geq 1$) satisfy the **bounded angle condition** if

$$\frac{(g^\ell)^T p^\ell}{\|g^\ell\|_* \|p^\ell\|} \leq -\delta < 0 \quad \text{for } \ell = 1, 2, \ldots. \tag{5}$$

The reason for this terminology is that if the norms are Euclidean, the ratio is the cosine of the angle between g^ℓ and p^ℓ.

Warth & Werner [37] call a line search **efficient** if it always returns step sizes satisfying

$$(f(x) - f(x + \alpha p)) \frac{\|p\|^2}{(g(x)^T p)^2} \geq \zeta, \tag{6}$$

where ζ is a fixed positive real number. The following basic convergence result is a simple consequence of their result. Note that linear convergence by itself does not imply a complexity statement since it is unclear how many function values are used in each line search. Bounding this number is an essential part of the analysis of our new method (see Theorem 3.3)

1.1 Theorem. *Given an optimization method that uses search directions satisfying the angle condition* (5) *and computes its points by*

$$x^{\ell+1} = x^\ell + \alpha_\ell p^\ell, \quad \alpha_\ell > 0, \tag{7}$$

where p^ℓ is a descent direction.

(i) If the line searches are efficient then

$$\inf_\ell \|g^\ell\|_* = 0. \tag{8}$$

(ii) If the x^ℓ converge to a strong local minimizer $\widehat{x}$ then there are constants $q \in]0, 1[$ and $c > 0$ such that

$$\|x^\ell - \widehat{x}\| \leq cq^\ell, \quad \|g^\ell\|_* \leq \overline{\gamma} cq^\ell \ \text{ for all } \ell \geq 0. \tag{9}$$

Proof. (i) (8) follows directly from the angle condition and Lemma 2.1 and Satz 2.2 of [37].

(ii) The assumption implies that f is uniformly convex in some neighborhood M of $\widehat{x}$. By assumption, there is an index k such that $x^\ell \in M$ for all $\ell \geq k$. Then, by the proof of Lemma 2.1 in [37],

$$f_{\ell+1} - f(\widehat{x}) \leq q^2 (f_\ell - f(\widehat{x}))$$

holds for some $0 < q < 1$ and all $\ell \geq k$, so that

$$f_\ell - f(\widehat{x}) \leq q^{2(\ell-k)} (f_k - f(\widehat{x})) \ \text{ for } \ell \geq k.$$

Now (3) and the arguments in the proof of Lemma 2.1 of [37] imply that (9) holds for some $c > 0$ and all $\ell \geq k$ (a self-contained proof is in the preprint [29, Theorem 4.2]). By increasing c, (9) for all ℓ. □

Recently, Neumaier & Kimiaei [30] proposed a new line search method `CLS`, which guarantees that the sufficient descent condition

$$\mu(\alpha)|\mu(\alpha) - 1| \geq \beta, \ \text{ with fixed } \beta > 0 \tag{10}$$

holds for the Goldstein quotient (Goldstein [15])

$$\mu(\alpha) := \frac{f(x+\alpha p) - f(x)}{\alpha g(x)^T p} \quad \text{for} \quad \alpha > 0. \tag{11}$$

In the first iteration, if $\mu(\alpha) < 1$ holds, `CLS` computes the step size

$$\widehat{\alpha} = \frac{\alpha}{2(1-\mu(\alpha))} \tag{12}$$

to ensure termination with at most two function evaluations when f is almost quadratic. Until (10) does not hold, `CLS` performs a simple bisection scheme since f is far from quadratic and bounded. `CLS` performs either interpolation with (12), or extrapolation with a constant factor $q > 1$, to obtain a bracket $[\underline{\alpha}, \overline{\alpha}]$ with $\underline{\alpha} > 0$ and $\overline{\alpha} < \infty$. Once `CLS` finds such a bracket, the geometric mean of $\underline{\alpha}$ and $\overline{\alpha}$ is chosen as a new step size in the next iteration. After `CLS` terminates with $\alpha > 0$ satisfying (10), then the condition

$$f(x) - f(x+\alpha' p) \geq \frac{2\beta}{\overline{\gamma}} \frac{(g(x)^T p)^2}{\|p\|^2} \tag{13}$$

is satisfied for any step size α' with $f(x+\alpha' p) \leq f(x+\alpha p)$ ([30, Theorem 3.1]). Here $\overline{\gamma}$ is from (3).

For any descent optimization algorithm using `CLS`, Neumaier & Kimiaei [30, Theorem 4.2] proved a complexity of $\mathcal{O}(\varepsilon^{-2})$ for the number of iterations and function evaluations under the condition that for fixed tuning parameters $0 < \kappa < \lambda < \infty$ the initial step sizes α_{init} and the maximum step size $\alpha_{\max}$ satisfy the condition

$$\frac{\kappa |g(x)^T p|}{\|p\|^2} \leq \alpha_{\text{init}} \leq \alpha_{\max} \leq \frac{\lambda |g(x)^T p|}{\|p\|^2}. \tag{14}$$

1.1.2 Nonlinear CG methods

In the literature, nonlinear CG methods (without preconditioner) are generally described in terms of search directions of the form

$$d^\ell = -g^\ell + \beta_{\ell-1} d^{\ell-1} \tag{15}$$

and corresponding updates

$$x^{\ell+1} = x^\ell + \gamma_\ell d^\ell. \tag{16}$$

The first nonlinear CG method, introduced by Fletcher & Reeves [14], uses (15)

$$\beta_{\ell-1} := \frac{(g^\ell)^T g^\ell}{(g^{\ell-1})^T g^{\ell-1}}. \tag{17}$$

For a quadratic function $f(x) = \frac{1}{2} x^T A x - b^T x$ with symmetric positive definite Hessian A, they showed the equivalence with the CG method of Hestenes & Stiefel [22] for solving the linear system of equations $Ax = b$, and hence finitely terminates with $g^\ell = 0$ for $\ell \leq n$.

Many other formulas for the βs, sharing this property, are in use. The most prominent examples are, with $y^{\ell-1} := g^\ell - g^{\ell-1}$,

$$\beta_\ell^{FR} = \frac{(g^\ell)^T g^\ell}{(g^{\ell-1})^T g^{\ell-1}} \quad \text{(Fletcher \& Reeves [14])},$$

$$\beta_\ell^{PR} = \frac{(y^{\ell-1})^T g^\ell}{(g^{\ell-1})^T g^{\ell-1}} \quad \text{(Polak \& Ribiere [34])},$$

$$\beta_\ell^{P} = \max(0, \beta_\ell^{PR}) \quad \text{(Powell [35])},$$

$$\beta_\ell^{CD} = \frac{(g^\ell)^T g^\ell}{(y^{\ell-1})^T d^{\ell-1}} \quad \text{(Fletcher [13])},$$

$$\beta_\ell^{HS} = \frac{(y^{\ell-1})^T g^\ell}{(y^{\ell-1})^T d^{\ell-1}} \quad \text{(Hestenes \& Stiefel [22])}.$$

Further variants can be found in the thorough survey of nonlinear CG methods by Hager & Zhang [21]. In Section 4, we compare two versions of our CG method with 21 CG methods from the literature [1, 2, 4, 8, 9, 10, 12, 14, 18, 22, 23, 26, 27, 28, 32, 34, 39], listed in Table 1 in Section 4.2.

For the optimization of quadratic functions, all these formulas are equivalent in exact arithmetic. But they have different properties in finite precision arithmetic and in generalizations to the optimization of nonquadratic functions. In particular, on nonquadratic problems, some of these formulas may exhibit convergence difficulties such as jamming, where many consecutive excessively small steps are taken. In addition, in order that the search directions generated are descent directions, restrictions on the previous step sizes are needed.The survey by Hager & Zhang discusses these problems in more detail.

The survey [21] also gives derivations of the global convergence results known at the time. Almost all (cf. Section 1.1.5) previously known globally convergent nonlinear CG methods determine their step sizes γ_ℓ by enforcing some version of the Wolfe conditions [38], thus involving gradients at each trial point.

1.1.3 Restart techniques

Practical CG methods use a restart strategy in which the CG direction is replaced by the steepest descent direction, to achieve a minimizer of a quadratic function after at most n iterations. Powell [36] gave three reasons why the traditional restart procedure is not good, and used a two-terms CG direction instead of the steepest descent direction for restart iterations. Then, Beale [5] retained the restart procedure of Powell for restart iterations and used the three-terms CG directions for non-restart iterations. Dai & Yuan [10] used these two restart procedures as an algorithm called the Beale & Powell restart algorithm, and showed by example that this algorithm may not converge. To obtain convergence results for the Beale & Powell restart algorithm, they restricted the CG parameters (generate only positive values) and constructed a modified Beale & Powell restarted algorithm. The improved Beale & Powell restart algorithm initializes the restart counter $t = 1$ and then updates this counter by $t = \ell - 1$ if $\ell - t \geq n$ or the condition

$$\left|(g^{\ell-1})^T g^\ell\right| > c_1 \|g^\ell\|_*^2 \quad \text{for } \ell \geq 2,$$

holds, where $0 < c_1 < 1$. Then the CG direction d^ℓ and corresponding updates $x^{\ell+1}$ is computed according to whether or not restart is required:
• If $\ell > t+1$, the three-terms CG direction for restart iterations

$$d^\ell = -g^\ell + \beta_\ell^{HS} d^{\ell-1} + \gamma_\ell d^t \tag{18}$$

is computed, where γ_ℓ is β_ℓ^{HS} with the difference that $y^\ell = g^{\ell+1} - g^\ell$ is replaced by $y^t = g^{t+1} - g^t$. As long as the condition

$$-c_3\|g^\ell\|_*^2 \le (g^\ell)^T d^\ell \le -c_2\|g^\ell\|_*^2 \text{ with } 0 < c_2 < 1 < c_3 < \infty \tag{19}$$

does not hold, $t = \ell - 1$ is chosen and d^ℓ is recomputed by (18). Once the condition (19) holds, d^ℓ computed by (18) is sufficiently downhill; hence the new point

$$x^{\ell+1} = x^\ell + \alpha_\ell d^\ell \tag{20}$$

is computed, where α_ℓ satisfies Wolfe conditions.
• If $\ell = t+1$, $\gamma_\ell = 0$ is chosen and the two-terms CG direction d^ℓ is computed by (18), regardless of whether or not the condition (19) holds, and the new point is computed by (20).

Very recently, Chan-Renous-Legoubin & Royer [7] employed a backtracking line search along several known CG directions with a nonstandard restart condition. They restarted their algorithm if at least one of the two conditions

$$(g^\ell)^T d^\ell < -c_1\|g^\ell\|_*^{1+p}, \quad 0 < c_1 \le 1,\ p \ge 0 \tag{21}$$

$$\|d^\ell\| < c_2\|g^\ell\|_*^q, \quad c_2 \ge 1,\ q \ge 0 \tag{22}$$

is violated.

1.1.4 Hybrid conjugate gradient methods

Section 4 compares only algorithms that use exclusively CG directions. There are, however, some software packages that use CG as a basic technique, but have been enhanced with a considerable number of modifications to improve practical performance, see e.g., `CG_DESCENT` of Hager & Zhang [17, 18, 19], `LMBOPT` [24] of Kimiaei et al. [24], `ASACG` of Hager & Zhang [17, 18, 19, 20], and `CGOPT` of Liu et al. [25]. Most of these solvers were comprehensively compared in [24] on the unconstrained and bound-constrained `CUTEst` test problems of Gould et al. [16]. Thus, for numerical comparison, we restrict ourselves here only to pure CG algorithms.

1.1.5 Complexity for conjugate gradient methods

Cartis et al. [6] proved a complexity of $\mathcal{O}(\varepsilon^{-2})$ for a general class of gradient-free line search algorithms along *gradient-related search directions*, defined as those satisfying the conditions

$$(g^\ell)^T d^\ell \le -\kappa_1\|g^\ell\|_*^2, \quad \|d^\ell\| \le \kappa_2\|g^\ell\|_* \quad \text{for } \kappa_1, \kappa_2 > 0. \tag{23}$$

In fact, concerning the search directions, the conditions (23) imply the bounded angle condition (5) with $\delta = \kappa_1/\kappa_2$, and thus, the same complexity result as their follows analogously by our Theorem 3.3(i) below. But, to the best of our knowledge, there is no result in the literature that any of the previously known CG formulas leads to search directions satisfying (23). Hence their complexity analysis does not apply to CG methods.

Very recently, a complexity result for a nonlinear CG method was given by Chan-Renous-Legoubin & Royer [7]. They proved that

$$\mathcal{O}(\varepsilon^{-2}) + \mathcal{O}\Big(\varepsilon^{-\max\{1+p,2(1+p-q)\}}\Big)$$

function evaluations are sufficient for reaching a point x^ℓ with $\|g^\ell\|_* \leq \varepsilon$. The best complexity $O(\varepsilon^{-2})$ is obtained when $p \leq \max(q, 1)$. For $p = q = 1$, their restart rule guarantees that the search directions satisfy (23), and as a consequence, the analysis of Cartis et al. [6] also applies. Their family of methods also does not use the Wolfe conditions. However, compared to our method, no requirement is imposed on the factors $\beta_{\ell-1}$ in (15), and convergence and complexity results are completely unaffected by their choice. Moreover, the restart conditions (21) and (22) may trigger a restart even when the objective function is quadratic, and the factors $\beta_{\ell-1}$ are chosen by one of the standard formulas, thus possibly impairing the finite termination property for quadratic functions.

1.2 An overview of our method

The present paper analyzes a new nonlinear CG method called `NCG`, formally specified in Algorithm 1. Compared to the traditional approaches, we can summarize here the following new features:

• Most of the known nonlinear CG methods need for their global convergence that the (strong) Wolfe conditions at each step hold. This involves the gradient evaluation at each trial point which might be expensive from the computational point of view. Our global convergence analysis of `NCG` does not rely on the Wolfe conditions and uses the so-called efficient line search methods which generate step sizes satisfying (6).

• A restart condition is built into the algorithm. This guarantees global linear convergence when it converges to a strong local minimizer without impairing the finite termination property of CG iterations for strictly convex quadratic functions.

• It is known that CG methods reduce the zigzagging effect observed in the steepest descent method. We quantify this effect by defining a measure of zigzagging strength (Section 2.1). Unless a restart is made, our CG direction is the search direction minimizing the zigzagging strength. This ensures that in our CG method, zigzagging is maximally reduced.

• `NCG` terminates after at most n iterations for strictly convex quadratic functions.

• `NCG` has the optimal complexity $\mathcal{O}(\varepsilon^{-2})$ for continuously differentiable objective function with Lipschitz continuous gradients. Moreover, it preserves the standard optimal complexity $\mathcal{O}(\log \varepsilon^{-1})$ for strongly convex functions.

1.2.1 Search directions with minimal zigzagging

In each iteration, `NCG` uses a new nonlinear CG direction along which `CLS` is tried. Unlike other CG directions (cf. Hager & Zhang [21]), our CG direction minimizes a measure of zigzagging strength, the squared preconditioned distance from the previous search direction. As a consequence, the amount of zigzagging in consecutive search directions is minimized.

1.2.2 A restart condition guaranteeing global linear convergence

To enforce linear convergence when `NCG` converges to a strong local minimizer we guarantee in Theorem 2.3 below the angle condition (5) using a restart procedure that decides when to replace a poor search direction by a simplified Newton direction, using a symmetric and positive definite preconditioner B.

Unlike the restart procedure of Chan-Renous-Legoubin & Royer [7], which restarts the CG algorithm when the conditions (21) and (22) are violated, our restart is performed when at least one of the two conjugacy relations (32) and (33) is significantly violated or the number m of non-restart iterations reaches n. Like Dai & Yuan [10], this ensures that no restart is performed for quadratic functions.

1.2.3 A new convergence analysis

The global convergence of a new nonlinear CG method is proved in Theorem 2.3 that, unlike traditional nonlinear CG methods that require line search methods satisfying the Wolfe condition, uses the new gradient-free line search method. The new CG method is motivated by the desire to reduce the inefficiency of line search methods due to zigzagging of search directions discussed in Section 1.1.2. Our search direction is therefore chosen by minimizing (Theorem 2.1) a preconditioned distance from the previous search direction.

1.2.4 A complexity bound

We prove in Theorem 3.3 complexity bounds on the number of iterations of `NCG`. We find that `NCG` has the same order $\mathcal{O}(\varepsilon^{-2})$ of complexity as the CG method by Chan-Renous-Legoubin & Royer [7], and the complexity improves to $\mathcal{O}(\log \varepsilon^{-1})$ if the objective function has only a strong minimizer and no other stationary points.

2 A class of search directions

From now on we use the pair of **ellipsoidal norms**

$$\|p\| := \sqrt{p^T B p}, \quad \|g\|_* := \sqrt{g^T B^{-1} g} = \|B^{-1} g\| \tag{24}$$

defined in terms of a fixed symmetric positive definite matrix $B \in \mathbb{R}^{n\times n}$. Using a Cholesky factorization $B = R^T R$ and a linear transformation $p' = Rp$, $g' = R^{-T} g$, where R^{-T} denotes the transposed inverse of R, it is easy to check that these form a pair of dual norms, so that

$$|g^T p| \le \|g\|_* \|p\|.$$

The case without preconditioning is obtained for the identity matrix $B = I$, where both norms (24) become the Euclidean norm $\|s\|_2 := \sqrt{s^T s}$, which is its own dual.

If a symmetric and positive definite preconditioner B approximating the Hessian near the starting point is available, it is sensible to measure closeness in terms of distance in the ellipsoidal norms (24) associated with B. In the absence of such curvature information we may simply take $B = I$, giving the special case without preconditioning.

2.1 Search directions with minimal zigzagging

Starting with $x^0 = \binom{\xi}{\xi}$, the steepest descent method ($p^\ell = -g^\ell$) with exact line searches applied to the optimization problem

$$\begin{array}{ll} \min & f(x) = (x_1 - x_2)^2 + \varepsilon x_2^2 \\ \text{s.t.} & x \in \mathbb{R}^2 \end{array}$$

yields the sequence

$$x^{2\ell} = \xi(1+\varepsilon)^{-\ell}\begin{pmatrix}1\\1\end{pmatrix}, \quad x^{2\ell+1} = \xi(1+\varepsilon)^{-\ell-1}\begin{pmatrix}1+\varepsilon\\1\end{pmatrix},$$

with arbitrarily slow linear convergence as $\varepsilon \downarrow 0$. The reason is inefficient **zigzagging** of the search directions. Thus linear convergence by itself is no quality criterion, and the angle condition only serves as a guard against even slower sublinear convergence behavior (see Theorem 2.2).

In order to avoid zigzagging we propose to choose the search direction p as a vector satisfying $g^T p < 0$ that is closest to the previous search direction p_{old}. The preconditioned distance $(p - p_{\text{old}})^T B (p - p_{\text{old}})$ is called **zigzagging strength measure**. If this measure is small, almost no zigzagging occurs; otherwise, a large zigzagging occurs, as in the steepest descent direction.

In order that it is meaningful to compare two different search directions we note that for a sufficiently small step size α we obtain a gain in the function value of

$$f(x) - f(x + \alpha p) = -\alpha g^T p + o(\alpha).$$

Hence the infinitesimal quality of a direction is fully characterized by $\nu := g^T p$. We therefore compare only directions with the same value of ν; this is no restriction of generality since we may rescale an arbitrary direction to match any given value of ν.

2.1 Theorem. *Among all $p \in \mathbb{R}^n$ with $g^T p = -\nu < 0$, the squared preconditioned distance $(p - p_{\text{old}})^T B (p - p_{\text{old}})$ becomes minimal for*

$$p = p_{\text{old}} - \lambda B^{-1} g, \tag{25}$$

where

$$\lambda = \frac{\nu + g^T p_{\text{old}}}{g^T B^{-1} g}. \tag{26}$$

Proof. This optimization problem can be solved using Lagrange multipliers. We have to find a stationary point of the Lagrange function

$$L(p, \lambda) := \frac{1}{2}(p - p_{\text{old}})^T B (p - p_{\text{old}}) + \lambda g^T p,$$

giving the condition $B(p - p_{\text{old}}) + \lambda g = 0$, hence (25) holds. The Lagrange multiplier λ is determined from the constraint $g^T p = -\nu$, and yields (26). □

2.2 Global convergence

For a search direction of the form

$$p^\ell = \rho_\ell p^{\ell-1} - \lambda_\ell B^{-1} g^\ell, \tag{27}$$

we need

$$0 < \nu := -(g^\ell)^T p^\ell = -\rho_\ell (g^\ell)^T p^{\ell-1} + \lambda_\ell (g^\ell)^T B^{-1} g^\ell,$$

hence

$$\lambda_\ell = \frac{\nu + \rho_\ell (g^\ell)^T p^{\ell-1}}{(g^\ell)^T B^{-1} g^\ell}. \tag{28}$$

For $\rho_\ell = 1$, this agrees with the direction derived from the zigzagging avoiding argument in Theorem 2.1; for $\rho_\ell = 0$, we get the simplified Newton direction $-B^{-1} g^\ell$, up to a constant factor. Thus search directions of the form (27) look like a flexible choice.

2.2 Theorem. *Let $\nu > 0$ and suppose that (27) and (28) hold for all ℓ with $|\rho_\ell| \le 1$. Then, for these ℓ,*

$$(p^\ell)^T B p^\ell - (p^{\ell-1})^T B p^{\ell-1} \le \frac{\nu^2}{(g^\ell)^T B^{-1} g^\ell}. \tag{29}$$

Moreover, if an efficient line search is used, the number L of iterations to reach

$$\|g^L\|_* < \varepsilon \tag{30}$$

is bounded by

$$L \le 1 + \xi \varepsilon^2 (e^{C/\varepsilon^2} - 1), \tag{31}$$

where $\xi := \|s^1\| / |(g^1)^T s^1|$ and $C := \dfrac{\overline{\gamma}}{2\beta}(f_0 - \underline{f})$.

Proof. By $\sigma_\ell := |(g^\ell)^T s^\ell| > 0$, we have

$$\begin{aligned}(p^\ell)^T B p^\ell &= \rho_\ell^2 (p^{\ell-1})^T B p^{\ell-1} - 2\rho_\ell \lambda_\ell (g^\ell)^T p^{\ell-1} + \lambda_\ell^2 (g^\ell)^T B^{-1} g^\ell \\ &= \rho_\ell^2 (p^{\ell-1})^T B p^{\ell-1} + \frac{\nu^2 - \left(\rho_\ell (g^\ell)^T p^{\ell-1}\right)^2}{(g^\ell)^T B^{-1} g^\ell}.\end{aligned}$$

By (28), (29) follows since $\rho_\ell^2 \le 1$. In terms of the ellipsoidal norms (24), (29) reads

$$\frac{1}{\nu^2}\left(\|p^\ell\|^2 - \|p^{\ell-1}\|^2\right) \le \frac{1}{\|g^\ell\|_*^2}.$$

Since $s^\ell = \alpha_\ell p^\ell$ and $\sigma_\ell = \alpha_\ell \nu$, we find

$$\frac{\|s^\ell\|^2}{\sigma_\ell^2} - \frac{\|s^{\ell-1}\|^2}{\sigma_{\ell-1}^2} \le \|g^\ell\|_*^2.$$

If L is the first index with (30), we conclude that

$$\frac{\|s^\ell\|^2}{\sigma_\ell^2} - \frac{\|s^{\ell-1}\|^2}{\sigma_{\ell-1}^2} \le \frac{1}{\|g^\ell\|_*^2} \le \frac{1}{\varepsilon^2} \quad \text{for } \ell < L.$$

Summation over all steps gives

$$\frac{\|s^\ell\|^2}{\sigma_\ell^2} \le \xi + \frac{\ell - 1}{\varepsilon^2} \quad \text{for } \ell < L.$$

Since the line search is efficient, by (13), we have

$$\frac{\overline{\gamma}}{2\beta}(f_\ell - f_{\ell+1}) \ge \frac{\sigma_\ell^2}{\|s^\ell\|^2} \ge \left(\xi + \frac{\ell - 1}{\varepsilon^2}\right)^{-1} = \frac{\varepsilon^2}{\ell - 1 + \xi\varepsilon^2} \quad \text{for } \ell < L.$$

For any $z > 0$, we have

$$\frac{1}{\ell - 1 + z} \ge \log\left(1 + \frac{1}{\ell - 1 + z}\right) = \log(\ell + z) - \log(\ell - 1 + z),$$

$$\sum_{\ell=1}^{L-1} \frac{1}{\ell - 1 + z} \ge \log(L - 1 + z) - \log z = \log\left(1 + \frac{L-1}{z}\right).$$

Summation over all steps gives

$$\varepsilon^2 \log\left(1 + \frac{L-1}{\xi\varepsilon^2}\right) \le \sum_{\ell=1}^{L-1} \frac{\varepsilon^2}{\ell - 1 + \xi\varepsilon^2} \le \frac{\overline{\gamma}}{2\beta} \sum_{\ell=1}^{L-1} (f_\ell - f_{\ell+1}) \le \frac{\overline{\gamma}}{2\beta}(f_0 - \underline{f}) := C,$$

resulting in

$$L \le 1 + \xi\varepsilon^2 (e^{C/\varepsilon^2} - 1).$$

□

The bound (31) is extremely poor, and will be improved under additional conditions.

2.3 A sufficient condition for the angle condition

2.3 Theorem. *Under the conditions of Theorem 2.2, suppose that an efficient line search is used and there are positive constants κ_1 and κ_2 such that, for all sufficiently large ℓ, either p^ℓ is parallel to the simplified Newton direction $-B^{-1}g^\ell$ or the conditions*

$$(g^\ell)^T B^{-1} g^\ell \le \kappa_1 (y^{\ell-1})^T B^{-1} y^{\ell-1}, \tag{32}$$

$$(y^{\ell-1})^T p^{\ell-1} \le \kappa_2 \nu \tag{33}$$

hold (where $y^{\ell-1} := g^\ell - g^{\ell-1}$).

(i) If $\widehat{x}$ is a strong local minimizer then the angle condition (5) holds for some $\delta > 0$.

(ii) If f has a strong local minimizer $\widehat{x}$ and no other stationary point then convergence is globally linear and the number of iterations to reach a point x with

$$\|g(x)\|_* \le \varepsilon \tag{34}$$

is $\mathcal{O}(\log \varepsilon^{-1})$. In particular, this is the case when f is strongly convex.

(iii) If the initial step sizes are chosen such that (14) holds and the line search `CLS` *is used, then the number of function evaluations needed to reach (34) is $\mathcal{O}(\log \varepsilon^{-1})$.*

Proof. (i) Since the eigenvalues of a positive definite matrix are positive, the requirements on $\widehat{x}$ imply that there are positive constants $\underline{\gamma}, \overline{\gamma}$ and a ball C around $\widehat{x}$ such that for all $x \in C$, the eigenvalues of the Hessian $G(x)$ are in $[\underline{\gamma}, \overline{\gamma}]$. The remainder form of Taylor's theorem now implies that for $x, x' \in C$, the condition (3) and

$$\frac{1}{2}\underline{\gamma}\|x' - x\|^2 \le f(x') - f(x) - g(x)^T(x' - x) \le \frac{1}{2}\overline{\gamma}\|x' - x\|^2. \tag{35}$$

Interchanging x and x' in the first inequality of (35), adding the two formulas, and applying the generalized Cauchy–Schwarz inequality gives

$$\underline{\gamma}\|x' - x\|^2 \le (g(x') - g(x))^T(x' - x) \le \|g(x') - g(x)\|_* \|x' - x\|. \tag{36}$$

Since $\widehat{x}$ is assumed to be a strong local minimizer, relations (3) and (36) apply for x, x' sufficiently close to $\widehat{x}$, and give

$$\underline{\gamma}\|g(x') - g(x)\|_* \, \|x' - x\| \le \overline{\gamma}(g(x') - g(x))^T(x' - x).$$

Substituting $x' = x^\ell$ and $x = x^{\ell-1}$ and using (32), we find after division by $\alpha_{\ell-1}$ that

$$\underline{\gamma}\|y^{\ell-1}\|_* \, \|p^{\ell-1}\| \le \overline{\gamma}(y^{\ell-1})^T p^{\ell-1} \le \overline{\gamma}\kappa_2\nu$$

for all sufficiently large ℓ for which (32) and (33) hold. For these ℓ,

$$\begin{aligned} (g^\ell)^T B^{-1} g^\ell \cdot (p^{\ell-1})^T B p^{\ell-1} &\le \kappa_1 (y^{\ell-1})^T B^{-1} y^{\ell-1} \cdot (p^{\ell-1})^T B p^{\ell-1} \\ &\le \kappa_1 \|y^{\ell-1}\|_*^2 \|p^{\ell-1}\|^2 \\ &\le \kappa_1 \Big(\overline{\gamma}\kappa_2\nu/\underline{\gamma}\Big)^2 = c\nu^2 \end{aligned}$$

for some constant $c > 0$. Now (29) implies

$$\frac{(p^\ell)^T B p^\ell}{\nu^2} \leq \frac{(p^{\ell-1})^T B p^{\ell-1}}{\nu^2} + \frac{1}{(g^\ell)^T B^{-1} g^\ell} \leq \frac{c+1}{(g^\ell)^T B^{-1} g^\ell}.$$

Thus

$$\frac{\nu^2}{(p^\ell)^T B p^\ell \cdot (g^\ell)^T B^{-1} g^\ell} \geq \frac{1}{c+1} \tag{37}$$

for sufficiently large ℓ satisfying (32) and (33). But if (32) or (33) are violated, p^ℓ is the simplified Newton direction, for which (37) holds trivially. Since $0 < \nu = -(g^\ell)^T p^\ell$, this shows that the left hand side of (5) is bounded away from zero. Hence the angle condition (5) holds.

(ii) By (i) and Theorem 1.1(ii), (9) is satisfied with $0 < q < 1$. Thus, convergence is globally linear and at most

$$\ell = \left\lceil \frac{\log \overline{\gamma} c \varepsilon^{-1}}{\log(1/q)} \right\rceil = \mathcal{O}(\log \varepsilon^{-1})$$

iterations are required to satisfy (34).

(iii) The number $\mathcal{O}(\log \varepsilon^{-1})$ of function evaluations is the product of the number $\mathcal{O}(\log \varepsilon^{-1})$ of iterations and the number $\mathcal{O}(1)$ of function evaluations of `CLS` (see [30, Theorem 4.1]). □

3 A new nonlinear CG method

In this section, we describe NCG, our new nonlinear CG method.

Algorithm 1 NCG, **nonlinear CG method**

1: **Purpose**: NCG finds local minimizer of $f(x)$ (or a stationary point only)

2: **Input**: x^0 (starting point) and B (preconditioner)

3: **Tuning parameters**: $\kappa_1, \kappa_2 > 0$ and $m \geq n$ (parameters for CG restart), ε (minimum threshold for the gradient norm), $0 < \alpha_{\max} < \infty$ (maximum step size), and $0 < \kappa < \lambda < \infty$ (parameters for the initial step size).

4: **Requirements**: B symmetric and positive definite

5: **for** $\ell = 0, 1, \ldots,$ **do**
6: compute $g^\ell = g(x^\ell)$, $h^\ell = B^{-1}g^\ell$, and $\omega_\ell = (g^\ell)^T h^\ell$;
7: **if** $\omega_\ell \leq \varepsilon^2$, **break**; **end**; ▷ x^ℓ stationary
8: **if** $\ell = 0$, restart $= 1$;
9: **else** ▷ check whether restart is required
10: $\omega' = (h^\ell)^T g^{\ell-1}$;
11: restart1 $= (\omega_\ell > \kappa_1(\omega_\ell - 2\omega' + \omega_{\ell-1}))$;
12: restart2 $= (|(g^\ell)^T p^{\ell-1} + \nu| > \kappa_2 \nu)$;
13: restart $=$ (restart1 or restart2 or $n_{ng} \geq m$);
14: **end**;
15: **if** restart, ▷ restart was done
16: compute $\nu = \omega_\ell$ and $p^\ell = -h^\ell$; initialize $n_{cg} = 0$;
17: **else** ▷ no zigzagging CG direction
18: compute $\lambda_\ell = \dfrac{\nu + (g^\ell)^T p^{\ell-1}}{\omega_\ell}$ and $p^\ell = p^{\ell-1} - \lambda_\ell h^\ell$; update $n_{cg} = n_{cg} + 1$;
19: **end**;
20: choose the initial step size α_{init} of CLS2 such that

$$\frac{\kappa |g(x^\ell)^T p^\ell|}{\|p^\ell\|^2} \leq \alpha_{\text{init}} \leq \alpha_{\max} \leq \frac{\lambda |g(x^\ell)^T p^\ell|}{\|p^\ell\|^2}; \tag{38}$$

21: determine α_ℓ by CLS2 with $x^{\ell+1} = x(\alpha_\ell) = x^\ell + \alpha_\ell p^\ell$ and $f_{\ell+1} = f(x^{\ell+1})$;
22: **end for**
23: **return** $x^{\ell+1}$ and $f_{\ell+1}$;

NCG uses a CG restart process (lines 11-13) that guarantees good complexity bounds, without impairing finite termination for quadratic functions. NCG uses the line search CLS2, a modification of CLS from [30]. The restart conditions and CLS2 are discussed in Subsection 3.2. The complexity of NCG is obtained in Subsection 3.3.

By Theorem 2.1, our new search direction is not too different from the old one. Thus, f is expected to behave along the new search path like along the old one. The initial step size

α_{init} should satisfy the condition (14).

To start the iteration we take $p_{\text{old}} = 0$. In order to guarantee linear convergence, we may need to reset p_{old} to zero also at suitable later stages. We call this a **restart**; the precise restart conditions used come from Theorem 2.3. For $B \neq I$, i.e., if preconditioning is used, one should store $h^\ell := B^{-1}g^\ell$ in the computation of $\omega_\ell := (g^\ell)^T h^\ell$, for later use in the computation of p^ℓ. Finally, note that, by Theorem 2.1, $\nu = -(g^\ell)^T p^\ell$ remains constant as long as no restart is made. The result is Algorithm 1. It is a **nonlinear CG method** since by Theorem 3.2 below, it is for a quadratic function f with positive definite Hessian equivalent to the preconditioned CG method for solving positive definite linear systems of equations.

Since in exact precision arithmetic $\omega_\ell = 0$ is guaranteed, line 7 of Algorithm 1 uses instead the condition $\omega_\ell = \|g^\ell\|_*^2 \leq \varepsilon^2$ for a finite termination of `NCG`.

`NCG` uses (27)–(28) with $\rho_k = 0$ or $\rho_k = 1$, hence our convergence results apply. Theorem 2.2 and Theorem 2.3 imply the following global convergence result.

3.1 Theorem. *The points x^ℓ produced by the nonlinear CG method of Algorithm 1 satisfy* (8) *and in case of convergence to a strong local minimizer, the convergence is globally linear.*

Like all nonlinear CG algorithms, `NCG` can be implemented using very little storage only: Apart from what is needed for a Cholesky factor of the preconditioner, we need 4 vectors of storage (for x, g, p, and $h = B^{-1}g$). Without preconditioning ($B = I$), even 3 vectors suffice.

If no preconditioning is used ($B = I$) and all λ_ℓ are positive, our formulas can be cast into the traditional CG form (15)–(16) if we use the scaled vectors

$$d^\ell := \lambda_\ell^{-1} p^\ell = \lambda_\ell^{-1} p^{\ell-1} - g^\ell = -g^\ell + \frac{\lambda_{\ell-1}}{\lambda_\ell} d^{\ell-1} \tag{39}$$

and the correspondence

$$\beta_{\ell-1} := \frac{\lambda_{\ell-1}}{\lambda_\ell}, \quad \gamma_\ell := \frac{\alpha_\ell}{\lambda_\ell}.$$

Thus the two choices of search directions appear to be equivalent. However, directions of the form (27) are more flexible than directions with the the traditional formula (15) since no sign restriction applies to the λ_ℓ.

3.1 A modification of `CLS`

To ensure that the line search is exact and takes two function evaluations only when applied to strictly convex quadratic functions we modify the line search `CLS` from [30]. We call the modified version `CLS2`.

Unlike `CLS`, `CLS2` does not stop if the first step is efficient, which means that the sufficient decrease condition (10) holds in the first iteration. Instead, it stores the function value

Algorithm 2 `CLS2`, **curved line search**

Purpose: `CLS2` finds a step size α with $\mu(\alpha)|\mu(\alpha) - 1| \geq \beta$, and guarantees that for strictly convex quadratic functions, an exact line search is done.

Input: $x(\alpha)$ (search path), $f_0 = f(x(0))$ (initial function value), $\nu = -g(x(0))^T x'(0)$ (directional derivative)

Tuning parameters: α_{init} (initial step size), $\alpha_{\max}$ (maximal step size), $\beta \in]0, \frac{1}{4}[$ (parameter for efficiency), $Q > 1$ (factor for extrapolation and interpolation)

Requirements: $\nu > 0$, $0 < \alpha_{\text{init}} \leq \alpha_{\max} \leq \infty$

Initialization: `first`=1; `firstok` = 0; $\underline{\alpha} = 0$; $\overline{\alpha} = \infty$; $\alpha = \alpha_{\text{init}}$;

while 1 **do**
  compute $f_a = f(x(\alpha))$ and the Goldstein quotient $\mu(\alpha) = (f_0 - f_a)/(\alpha\nu)$;
  if $\mu(\alpha)|\mu(\alpha) - 1| \geq \beta$, ▷ sufficient descent condition was satisfied
    if `first`, $\alpha_1 = \alpha$; $f_1 = f_a$; **else**, **break**; **end**
    `firstok` = `first`;
  end
  if `firstok` and not `first`, $\alpha = \alpha_1$; $f_a = f_1$; **break**; **end**
  if $\mu(\alpha) > \frac{1}{2}$, $\underline{\alpha} = \alpha$;
  elseif $\alpha = \alpha_{\max}$, **break**;
  else, set $\overline{\alpha} = \alpha$; ▷ linear decrease or more
  end
  if `first`, ▷ initially check whether function is almost quadratic or not
    `first` = 0;
    if $\mu(\alpha) < 1$, $\alpha = \frac{1}{2}\alpha/(1 - \mu(\alpha))$; **else** $\alpha = \alpha Q$; **end**
  else
    if $\overline{\alpha} = \infty$, expand to $\alpha = \alpha Q$; ▷ extrapolation was done
    elseif $\underline{\alpha} = 0$, compute $\alpha = \frac{1}{2}\alpha/(1 - \mu(\alpha))$; ▷ interpolation was done
    else, calculate $\alpha = \sqrt{\underline{\alpha}\,\overline{\alpha}}$; ▷ interval was found; geometric mean was computed
    end
  end
  restrict $\alpha = \min(\alpha, \alpha_{\max})$;
  end
end while
return α and f_a;

at the first trial point and the corresponding step size. In general, `CLS2` agrees with `CLS`, except that it may accept the second step size rather than the first one since it performs at least two function evaluations. In particular, the properties proved in [30, Theorem 4.1] remain valid.

If $f(x + \alpha p)$ is a strictly convex quadratic, then $\mu(\alpha) < 1$ for all $\alpha \in \mathbb{R}$, and the second iteration performs a quadratic interpolation step. Thus the line search is exact and the second iteration is efficient. Therefore `CLS2` stops with exactly two function evaluations.

If $f(x + \alpha p)$ is linear or a strictly concave quadratic, the descent direction is a direction of infinite descent, and the line search ends after extrapolations reaching $\alpha = \alpha_{\max}$. Thus by general properties of CG methods, quadratic functions are minimized after at most n iterations using at most n gradient evaluations and $2n$ function evaluations.

In detail, the difference between `CLS` and `CLS2` consists of the following three changes:
- The initialization `firstOk` $= 0$ is added to line 5 of `CLS` in line 5 of `CLS2`.
- Lines 7–8 of `CLS` are replaced by lines 7–12 of `CLS2`.
- Line 25 of `CLS` is replaced by line 29 of `CLS2`.

The Boolean variables `first` and `firstOk` serve to ensure that behavior described. `CLS2` initializes `first` $= 1$ and `firstOk` $= 0$. If the first iteration is efficient, `firstok` is changed to true. On the next iterations, `first` is then changed to false. The Boolean variable `first` ensures that the quadratic case is handled optimally, and the Boolean variable `firstOk` ensures that the line search is terminated in the second iteration if the the first iteration is efficient, but the second iteration is inefficient. In this case, the values of the first iteration are restored.

3.2 Finite termination on quadratics

Our motivation for the CG restart is to guarantee a finite termination on quadratic functions. Theorem 3.2 below shows that in exact precision arithmetic, Algorithm 1 for quadratic functions terminates after at most n gradient evaluations with a minimizer or with a direction of infinite descent, though in finite precision arithmetic it may take more than n gradients evaluations and $2n$ function evaluations to find such a minimizer.

3.2 Theorem. *Applied to quadratic functions*

$$f(x) = \gamma + c^T x + \frac{1}{2} x^T G x, \tag{40}$$

Algorithm 1 for $m \geq n$ performs no restarts and produces the same sequence of x^ℓ as the nonlinear CG method by Fletcher and Reeves. In particular, Algorithm 1 stops for quadratic functions after at most n iterations with a minimizer or with a direction of infinite descent.

Proof. We have

$$p^\ell = p^{\ell-1} - \lambda_\ell B^{-1} g^\ell, \quad x^{\ell+1} = x^\ell + \alpha_\ell p^\ell.$$

For a quadratic function (40) we have $g^\ell = c + G x^\ell$, hence

$$g^\ell - g^{\ell-1} = G(x^\ell - x^{\ell-1}) = \alpha_{\ell-1} G p^{\ell-1}.$$

For quadratic functions, CLS2 becomes exact, hence

$$\alpha_\ell = \frac{-(g^\ell)^T p^\ell}{(p^\ell)^T G p^\ell} = \frac{\nu}{(p^\ell)^T G p^\ell}$$

as long as no restarts are made. Now $\nu = -(g^{\ell-1})^T p^{\ell-1}$, hence

$$\lambda_\ell = \frac{(g^\ell - g^{\ell-1})^T p^{\ell-1}}{\omega_\ell} = \frac{\nu}{\omega_\ell} > 0, \quad \beta_{\ell-1} = \frac{\lambda_{\ell-1}}{\lambda_\ell} = \frac{\omega_\ell}{\omega_{\ell-1}}.$$

Since an exact line search is used, the result of the algorithm is the same for an arbitrary rescaling of the search direction. Thus we may rewrite the iteration in terms of the d^ℓ computed by (39) and get for $B = I$ equivalence with the Fletcher-Reeves CG method. Fletcher & Reeves [14] showed the equivalence with the CG method of Hestenes & Stiefel [22] for solving the linear system of equations $g(x) = c + Gx = 0$. They proved the well-known conjugacy properties

$$(g^\ell)^T p^k = (g^\ell)^T B^{-1} g^k = 0 \quad \text{for } k \leq \ell - 1,$$

which imply that given the restrictions $\kappa_1, \kappa_2 > 0$, no restarts will be made for $\ell \leq m$.

Hestenes & Stiefel showed that for positive definite G, their algorithm stops after at most n iterations with a solution of the linear system, hence with the minimizer of $f(x)$. If G is not positive definite, the algebra remains the same, except that it is now possible that a line search ends with a direction of infinite descent. Thus, if $m \geq n$, our CG method for quadratic functions after at most n iterations with a minimizer or with a direction of infinite descent.

The case with a preconditioner is easily reduced to the case $B = I$ by means of a linear transformation of the vector x of variables; hence the same properties hold for any symmetric and positive definite B. □

Since locally all twice continuously differentiable functions are well approximated by a quadratic, the final remark in the proof also holds locally for general C^2-functions with CLS. Thus, close to a strong local minimizer, Algorithm 1 shares the excellent local convergence behavior of the quadratic case. The latter is surveyed in Axelsson & Lindskog [3] of the preconditioned linear CG method.

In particular, when a good starting point is available, no restarts are made. Far away from a minimizer, however, a strong deviation from quadratic behavior may cause a restart. In particular, whenever very little progress is made while the gradient is still large, $g^\ell \approx g^{\ell-1}$, hence $y^{\ell-1} \approx 0$, and a restart is made. Thus jamming, a problem for the standard implementation of the nonlinear CG method by Fletcher & Reeves [14] is not possible.

3.3 Complexity of NCG

The following complexity result for NCG yields a bound on the number of iterations (= number of gradient evaluations) and function evaluations of NCG. Note that our result also

holds for $m = \infty$, where no restart is imposed after a fixed number of iterations. But the numerical results in Section 4 show that setting $m = \infty$ reduces the efficiency and robustness of NCG.

3.3 Theorem. *Given constants $0 < \kappa < \lambda < \infty$ and $0 < m \leq \infty$, suppose that the initial step sizes are chosen such that (14) holds. Then:*

(i) The number of function values needed by NCG to reach a point x with (34) is $\mathcal{O}(\varepsilon^{-2})$.

(ii) If the sublevel set $\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\}$ is bounded then, starting with x^0, at least one subsequence of the points generated by NCG converges to a stationary point.

(iii) If f has a strong local minimizer $\widehat{x}$ and no other stationary point then convergence is globally linear and the number of function values needed by NCG with $m \geq n$ to reach a point x with (34) is $\mathcal{O}(\log \varepsilon^{-1})$. In particular, this is the case when f is strongly convex.

Proof. (i) Denote by R the index set of restart iterations and by R^c the index set of non-restart iterations. For $\ell \in R$, the search direction $p^\ell = -B^{-1}g^\ell$ satisfies the bounded angle condition (5) since

$$(g^\ell)^T p^\ell = -(g^\ell)^T B^{-1} g^\ell = -\|g^\ell\|_*^2, \quad \|p^\ell\| = \sqrt{(p^\ell)^T B p^\ell} = \sqrt{(g^\ell)^T B^{-1} g^\ell} = \|g^\ell\|_*;$$

hence

$$\frac{((g^\ell)^T p^\ell)^2}{\|p^\ell\|^2} \geq \delta^2 \|g^\ell\|_*^2 \quad \text{for } \ell \in R. \tag{41}$$

Denote $f_\ell := f(x^\ell)$ and $f_{\ell+1} := f(x^\ell + \alpha_\ell p^\ell)$, and suppose that the algorithm ends at x^L with

$$\|g(x^L)\|_* < \varepsilon \leq \|g(x^\ell)\|_* \quad \text{for } \ell < L. \tag{42}$$

We now find an upper bound on the number L of iterations of NCG. Substituting (41) into (13), we obtain

$$f_\ell - f_{\ell+1} \geq \frac{2\beta}{\overline{\gamma}} \delta^2 \|g(x^\ell)\|_*^2 \geq \frac{2\beta}{\overline{\gamma}} \delta^2 \varepsilon^2 \quad \text{for } L > \ell \in R. \tag{43}$$

For $\ell \in R^c$, we have $f_\ell - f_{\ell+1} > 0$ and so $\sum_{\ell \in R^c} (f_\ell - f_{\ell+1}) > 0$. Hence (2) and (43) imply

$$f_0 - \underline{f} \geq f_0 - f_L = \sum_{\ell=0}^{L-1} (f_\ell - f_{\ell+1}) = \sum_{\ell \in R} (f_\ell - f_{\ell+1}) + \sum_{\ell \in R^c} (f_\ell - f_{\ell+1}) \geq \frac{2\beta}{\gamma} \delta^2 \varepsilon^2 |R|,$$

leading to

$$|R| \leq C\varepsilon^{-2} \quad \text{with } C := \frac{\overline{\gamma}(f_0 - \underline{f})}{2\beta\delta^2}.$$

Since $|R^c| \leq m|R|$, this proves

$$L = |R| + |R^c| \leq (m+1)|R| \leq (m+1)C\varepsilon^{-2}.$$

By [30, Theorem 4.1], the number of function evaluations of CLS2 in each iteration is bounded by a constant. Therefore, this also holds for CLS2. Hence the number of function evaluations of NCG is $\mathcal{O}(L) = \mathcal{O}(\varepsilon^{-2})$.

(ii) By (i), $\inf_{\ell \geq 0} \|g(x^\ell)\|_* = 0$, which together with a standard compactness argument gives the result.

(iii) follows directly from Theorem 2.3(iii). □

4 Numerical results

In this section, we compare our algorithm NCG with the CG methods listed in Table 1 on all 507 unconstrained test problems with dimensions 2 to 9000 from the CUTEst collection by Gould et al. [16].

NCG is Algorithm 1 with the new CG direction and the tuning parameters

$$\kappa_1 = 1,\ \kappa_2 = 10,\ \kappa = 10^{-10},\ \lambda = 10^{-2},\ m = 2n + 10,\ l_{\max} = 20,\ \beta = 0.02.$$

To have a fair comparison, the preconditioner B was chosen an identity matrix in the NCG algorithm. The initial step size for CLS in the ℓth iteration of NCG was computed by

$$\alpha_{\text{init}} = \max(\kappa\alpha_0^\ell, \min(\alpha_h^\ell, \lambda\alpha_0^\ell)) \ \text{ with } \alpha_0^\ell = |(g(x^\ell))^T p^\ell| / \|p^\ell\|^2,$$

satisfying the condition (38). Here the ℓth heuristic step size α_h was computed heuristically as in goodStep of the LMBOPT solver by Kimiaei et al. [24]. We also compare this default NCG with NCG–, which stands for NCG with $m = \infty$.

To compute the CG parameter β_k, most CG methods need the curvature condition to guarantee that the condition $d_k^T y_k > 0$ holds for the nonconvex functions since this condition appears in β_k. Here d_k is computed by (15) and $y_k = g_{k+1} - g_k$. Except for NCG and NCG–, all CG methods tested use along the CG directions the strong Wolfe line search cvsrch by Moré & Thuente [33], with the default values for its tuning parameters.

4.1 Efficiency and robustness

We denote by $\mathcal{S}$ the list of compared solvers, by $\mathcal{P}$ the list of problems, and by $c_{p,s}$ the cost measure of the solver $s \in \mathcal{S}$ to solve the problem $p \in \mathcal{P}$. Our cost measures are the number nf of function evaluations, the number ng of gradient evaluations, nf2g = nf + 2ng, and times sec in second.

To find approximate local minimum of the unconstrained problems, we say that a CG method is most *efficient* if it has a lowest cost measure and is most *robust* if it has a highest number of solved problems compared to the other compared CG methods on unconstrained test problem from the CUTEst collection.

Using these cost measures, the efficiency and robustness of CG methods can be identified by performance profile of Dolan and Moré [11]. The performance profile of the solver s

$$\rho_s(\tau) := \frac{1}{|\mathcal{P}|}\Big|\Big\{p \in \mathcal{P} \;\Big|\; pr_{p,s} := \frac{c_{p,s}}{\min(c_{p,\overline{s}} \mid \overline{s} \in S)} \leq \tau\Big\}\Big|. \tag{44}$$

is the fraction of problems that the performance ratio $pr_{p,s}$ is at most τ. In particular, the fraction of problems that the solver s wins compared to the other solvers is $\rho_s(1)$ and for sufficiently large τ the fraction of problems that the solver s can solve is $\rho_s(\tau)$.

Each algorithm was terminated once one of the termination criteria given in the first row of Table 1 below was satisfied. These impose upper bounds on $\|g\|_\infty$, `sec`, and `nf2g`. For a given list S of solvers and each given cost measure c_s, the **partial efficiency**

$$e_{s,p} := \begin{cases} 1/pr_{p,s} & \text{if the solver } s \text{ solves the problem } p, \\ 0 & \text{otherwise} \end{cases}$$

of the solver s measures the strength of the solver s relative to an ideal solver corresponding to the best solver for the problem p in percent, rounded to integers. The **efficiency** e_s of the solver s to solve all $p \in P$ is the sum of $e_{s,p}$ over $p \in P$. The efficiency with respect to the cost measures `nf`, `ng`, `nf2g`, and `sec` are called `nf` *efficiency*, `ng` *efficiency*, `nf2g` *efficiency*, and `sec` *efficiency*, respectively. The other columns of the table contain the number of solved problems by the solvers, the `nf2g` efficiency, the `ng` efficiency, the `nf` efficiency, and the `sec` efficiency.

4.2 A comparison of CG methods

Table 1 and the performance profiles of Figure 1 suummarize the results of our numerical experiments. We see that due to restarts after a suitably fixed finite number of iterations, `NCG` is slightly more robust and efficient than `NCG`–. In the following analysis, we therefore ignore `NCG`–, and compare `NCG` with the other 21 CG methods from Table 1.

By inspecting the results we may conclude that:
• `NCG`, `DL+`, `FA`, `NYF`, and `MBA` are most robust among all compared CG methods. Hence in terms of robustness `NCG` is competitive with the best state-of-the-art CG methods.
• In terms of the `ng` efficiency, `NCG` is most efficient since it is 23% more efficient than the second best method `DL+`.
• In terms of the `nf` efficiency, `DL+` is most efficient since it is 5% more efficient than the second best method `NCG`.
• In terms of the `nf2g` efficiency, `NCG` is 11% more efficient than the second best method `DL+`.
• In terms of the `sec` efficiency, `DL+` and `DK+` are 12% more efficient than `NCG`.

Table 1: The summary results for all problems

stopping test: $\|g\|_\infty \leq 10^{-6}$, `sec` ≤ 300, `nf2g` $\leq 20n + 10^4$						
425 of 507 problems solved			mean efficiency e_s in %			
dim∈[2,9000]			for cost measure			
solver	reference	solved	`nf2g`	`ng`	`nf`	`sec`
NCG	Algorithm 1 with default m	393	59	64	51	39
DL+	Dai & Liao [9, with CG parameter (2.26)]	392	48	41	56	51
FA	Faramarzi & Amini [12]	388	44	37	51	47
NYF	Narushima et al. [28]	384	33	28	41	40
MBA	Mirhoseini et al. [32]	383	37	31	45	41
NCG–	Algorithm 1 with $m = \infty$	381	55	59	48	35
LH	Lotfi & Hosseini [27]	381	34	29	42	41
HS	Hestenes & Stiefel [22]	379	34	29	41	38
DK+	Dai & Kou [8, with CG parameter (2.32)]	377	46	39	53	51
HZ	Hager & Zhang [18, with CG parameter (1.3)]	375	37	31	44	41
HZ+	Hager & Zhang [18, with CG parameter (1.6)]	374	37	31	44	40
DL	Dai & Liao [9, with CG parameter (2.6)]	372	38	33	45	42
DK	Dai & Kou [8, with CG parameter (2.31)]	371	44	38	51	48
PR	Polak & Ribière [34]	367	31	26	38	36
BG	Babaie-Kafaki & Ghanbari [4, the `ZZ` method]	362	30	25	37	35
BG+	Babaie-Kafaki & Ghanbari [4, the `MZZ` method]	361	29	24	35	34
BA	Aminifard & Babaie-Kafaki [2]	357	20	16	24	25
LS	Liu & Storey [26]	349	29	24	36	33
AFP	Amini et al. [1]	343	33	27	41	37
DY	Dai & Yuan [10]	299	31	27	36	32
FR	Fletcher & Reeves [14]	267	23	19	28	24
YYZ	Yuan et al. [39]	250	15	12	17	16
IKKA	Ibrahim et al. [23]	197	7	6	9	10

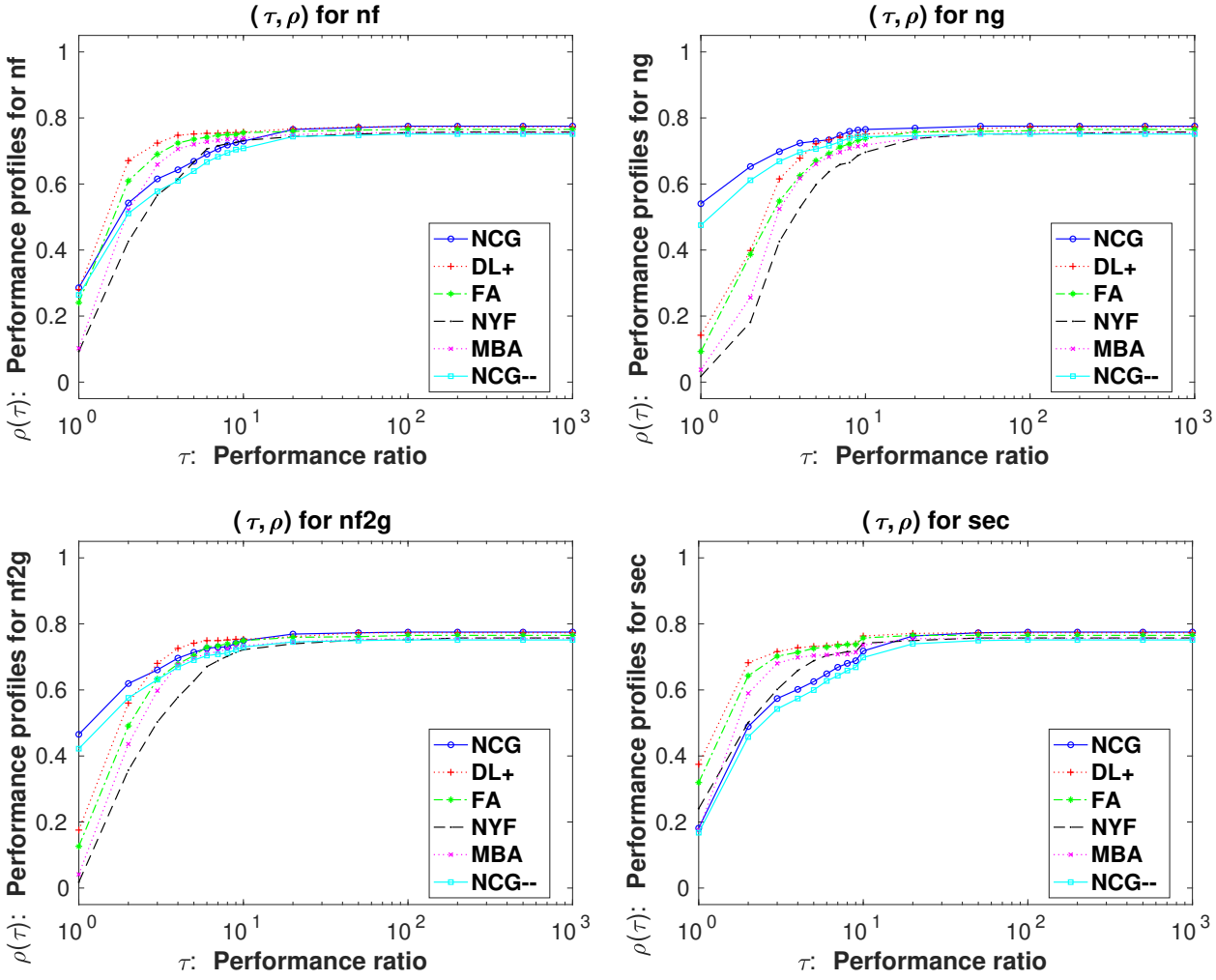

Figure 1: The performance profiles in terms of `nf`, `ng`, `nf2g`, and `sec` for 6 more robust CG methods. Problems solved by no solver are ignored. All compared solvers used the budgets `secmax` $= 300$ and `nfmax` $= 20n + 10000$.

5 Conclusion

This paper proposes a new nonlinear CG algorithm (`NCG`) for unconstrained optimization problems. The global convergence of `NCG` has been obtained without enforcing the strong Wolfe conditions but rather using an arbitrary efficient line search method that does not use any gradient evaluations. A restart condition ensures global linear convergence if it converges to a strong local minimizer without affecting the finite termination for strictly convex quadratic functions. Unless a restart is made within `NCG`, zigzagging is maximally reduced based on an introduced zigzagging strength.

It has been shown that `NCG` needs at most $\mathcal{O}(\varepsilon^{-2})$ function evaluations to find a ε-stationary point. This complexity result reduces to $\mathcal{O}(\log \varepsilon^{-1})$ provided that $f(x)$ has a strong local minimizer and no other stationary point.

Our numerical results on the 507 unconstrained `CUTEst` test problems illustrate that `NCG` is competitive with the best state-of-the-art CG methods in terms of the robustness and efficiency.

References

[1] K. Amini, P. Faramarzi, and N. Pirfalah. A modified Hestenes–Stiefel conjugate gradient method with an optimal property. *Optim. Methods Softw.* **34** (2018), 770–782.

[2] Z. Aminifard and S. Babaie-Kafaki. Dai–Liao extensions of a descent hybrid nonlinear conjugate gradient method with application in signal processing. *Numer. algorithms.* **89** (2022), 1369–1387.

[3] O. Axelsson and G. Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numer. Math.* **48** (1986), 499–523.

[4] S. Babaie-Kafaki and R. Ghanbari. Two modified three-term conjugate gradient methods with sufficient descent property. *Optim. Lett.* **8** (2014), 2285–2297.

[5] E. M. Beale. A deviation of conjugate gradients. *Numerical methods for nonlinear optimization.* (1972), 39–43.

[6] C. Cartis, Ph. R. Sampaio, and Ph. L. Toint. Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization. *Optimization* **64**(2015), 1349–1361.

[7] R. Chan-Renous-Legoubin and C. W. Royer. A nonlinear conjugate gradient method with complexity guarantees and its application to nonconvex regression. *EURO J. Comput. Optim.* **10** (2022), 100044.

[8] Y. H. Dai and C. X. Kou. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **23** (2013), 296–320.

[9] Y. H. Dai and L. Z. Liao. New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **43** (2001), 87–101.

[10] Y. Dai and Y. Yuan. Convergence properties of Beale-Powell restart algorithm. *Sci. China Ser. A-Math.* **41** (1998), 1142–1150.

[11] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.* **91** (January 2002), 201–213.

[12] P. Faramarzi and K. Amini. A modified spectral conjugate gradient method with global convergence. *J. Optim. Theory Appl.* **182** (2019), 667–690.

[13] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, Ltd (2000).

[14] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer J.* **7** (1964), 149–154.

[15] A. A. Goldstein. On steepest descent. *J. SIAM, Ser. A: Control* **3** (1965), 147–151.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Comput. Optim. Appl.* **60** (2015), 545–557.

[17] W. W. Hager and H. Zhang. CG_DESCENT user's guide. Technical report, Department of Mathematics, University of Florida, Gainesville, FL (2004).

[18] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **16** (2005), 170–192.

[19] W. W. Hager and H. Zhang. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Softw.* **32** (2006), 113–137.

[20] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.* **17** (2006), 526–557.

[21] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2** (2006), 35–58.

[22] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **49** (1952), 409–436.

[23] A. H. Ibrahim, P. Kumam, A. Kamandi, and A. B. Abubakar. An efficient hybrid conjugate gradient method for unconstrained optimization. *Optim. Methods Softw.* **37** (2022), 1370–1383.

[24] M. Kimiaei, A. Neumaier, and B. Azmi. LMBOPT: a limited memory method for bound-constrained optimization. *Math. Program. Comput.* **14** (2022) 271–318.

[25] Z. Liu, H. Liu, Y. H. Dai. An improved Dai–Kou conjugate gradient algorithm for unconstrained optimization. *Comput. Optim. Appl.* **75** (2020), 145–167.

[26] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1: theory *J. Optim. Theory Appl.* **69** (1991), 129—137.

[27] M. Lotfi and S. M. Hosseini. An efficient hybrid conjugate gradient method with sufficient descent property for unconstrained optimization. *Optim. Methods Softw.* **37** (2022), 1725–1739.

[28] Y. Narushima, H. Yabe and J. A. Ford. A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. *SIAM J. Optim.* **21** (2011), 212–230.

[29] A. Neumaier and B. Azmi. Line search and convergence in bound-constrained optimization. Unpublished manuscript, University of Vienna (2019). `http://www.optimization-online.org/DB_HTML/2019/03/7138.html`.

[30] A. Neumaier and M. Kimiaei. An efficient gradient-free line search. Preprint, University of Vienna (2022). `https://optimization-online.org/?p=21115`

[31] J. Nocedal and S. Wright. *Numerical optimization.* Springer Science & Business Media (2006).

[32] N. Mirhoseini, S. Babaie-Kafaki and Z. Aminifard. A Nonmonotone Scaled Fletcher–Reeves Conjugate Gradient Method with Application in Image Reconstruction. *Bull. Malays. Math. Sci. Soc.*. **45** (2022), 2885–2904.

[33] J. J. Moré, D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *(ACM) Trans. Math. Softw.* **20** (1994), 286–307.

[34] E. Polak and G. Ribière. Note sur la convergence de directions conjugées. *Rev. Francaise Informat Recherche Opertionelle* 3e Année. **16** (1969), 35–43.

[35] M. J. D. Powell. Convergence properties of algorithms for nonlinear optimization. *SIAM Rev.* **28** (1986), 487–500.

[36] M. J. D. Powell. Restart procedures for the conjugate gradient method. *Math. Program.* **12** (1977), 241–254.

[37] W. Warth and J. Werner. Effiziente Schrittweitenfunktionen bei unrestringierten Optimierungsaufgaben. *Computing* **19** (1977), 59–72.

[38] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.* **11** (1969), 226–235.

[39] G. Yuan, H. Yang, M. Zhang. Adaptive three-term PRP algorithms without gradient Lipschitz continuity condition for nonconvex functions *Numer. algorithms.* **91** (2022), 145–160.