SIAM J. OPTIM.
Vol. 29, No. 3, pp. 1988–2025

THE CONJUGATE RESIDUAL METHOD IN LINESEARCH AND TRUST-REGION METHODS*

MARIE-ANGE DAHITO† AND DOMINIQUE ORBAN†

Abstract. The minimum residual method (MINRES) of Paige and Saunders [*SIAM J. Numer. Anal.*, 12 (1975), pp. 617–629], which is often the method of choice for symmetric linear systems, is a generalization of the conjugate residual method (CR), proposed by Hestenes and Stiefel [*J. Res. Natl. Bur. Stand. (U.S.)*, 49 (1952), pp. 409–436]. Like the conjugate gradient method (CG), CR possesses properties that are desirable for unconstrained optimization, but is only defined for symmetric positive-definite operators. CR's main property, that it minimizes the residual, is particularly appealing in inexact Newton methods for optimization, typically used in a linesearch context. CR is also relevant in a trust-region context as it causes a monotonic decrease of convex quadratic models [D. C.-L. Fong and M. A. Saunders, *SQU J. Sci.*, 17 (2012), pp. 44–62]. We investigate modifications that make CR suitable, even in the presence of negative curvature, and perform comparisons on convex and nonconvex problems with CG. We complete our investigation with an extension suitable for nonlinear least-squares problems. Our experiments reveal that CR performs as well as or better than CG, and mainly yields savings in operator-vector products.

Key words. unconstrained optimization, conjugate residual method, inexact Newton method, trust-region method, conjugate gradient method

AMS subject classifications. 49M15, 49M37, 65F10, 65F20, 65K05, 90C30

DOI. 10.1137/18M1204255

1. Introduction.

We consider the large-scale unconstrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\; f(x), \tag{1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is twice-continuously differentiable and possibly nonconvex. We wish to identify a first-order critical point of (1), i.e., $x_*\in\mathbb{R}^n$ such that $\nabla f(x_*)=0$, using either a linesearch or a trust-region method. In a linesearch method, we compute a step as an approximate solution of

$$Hs=-g, \tag{2}$$

with $H=H^T\approx\nabla^2 f(x)$ and $g=\nabla f(x)$, while in a trust-region method, the step is an approximate solution of

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\; m(s) \quad \text{subject to } \|s\|\le\Delta, \qquad m(s):=g^Ts+\tfrac{1}{2}s^THs, \tag{3}$$

where $\Delta>0$ is the trust-region radius. We consider the inexact solution of (2) and (3) via an iterative method. Throughout the paper, we assume that H is symmetric.

The conjugate gradient method (CG) (Hestenes and Stiefel (1952)) has long been a workhorse in optimization because of its desirable properties. In particular, it is well suited to (2) because H is often forced to be positive definite so as to obtain a descent

*Received by the editors July 31, 2018; accepted for publication (in revised form) April 4, 2019; published electronically July 25, 2019.
https://doi.org/10.1137/18M1204255
Funding: The second author's research was partially supported by an NSERC Discovery Grant.
†GERAD and Department of Mathematics and Industrial Engineering, École Polytechnique, Montréal, QC, Canada (marie-ange.dahito@polymtl.ca, dominique.orban@gerad.ca).

direction. Dembo and Steihaug (1983) developed modifications of CG to cope with directions of negative curvature in the context of inexact Newton methods that ensure global and fast local convergence. CG is also well suited to (3) because the value of m is monotonically decreasing along the CG iterations and the norm of the iterates is monotonically increasing. Thus if the iterates were to leave the trust region, they would never return. The kth CG iterate solves

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ m(s) \quad \text{subject to } s \in K_k,$$

where $K_k = \text{span}\{g, Hg, H^2g, \ldots, H^{k-1}g\}$ is the kth Krylov subspace. Steihaug (1983) allowed indefinite H by describing truncated CG, in which directions of negative curvature are followed to the boundary of the trust region. Yuan (2000) showed that the approximate solution s^C identified by the truncated CG is such that $m(s^C) \leq \frac{1}{2}m(s^*) \leq 0$, where s^* is a global solution of (3).

The conjugate residual method (CR) was introduced by Hestenes and Stiefel (1952) and Stiefel (1955), although the name is not mentioned in the first paper. Like CG, CR is a Krylov subspace method for solving (2) with positive definite H. At iteration $k = 1, 2, \ldots$, CR minimizes the residual norm $\|Hs + g\|_2$ in K_k. The search occurs along directions p_k that are conjugate with respect to H^2, while the residuals $r_k := -g - Hs_k$ are conjugate with respect to H, which explains the name of the method.

In inexact Newton methods, we seek a step s such that $\|Hs + g\| \leq \tau \|g\|$ for a certain tolerance $0 < \tau < 1$. Monotonicity of the residual norm seems like an appealing property to attain this stopping condition. Fong and Saunders (2012) established that when H is positive definite, the values of m decrease monotonically along the CR iterations and the norm of the iterates is also monotonically increasing. Those observations lead us to believe that CR could also be well suited to computing steps in both linesearch and trust-region methods.

We describe modifications of CR analogous to those of CG to cope with directions of negative curvature in both linesearch and trust-region methods, and establish that global and local convergence properties are preserved.

Our numerical experiments on convex problems indicate that CR performs comparably to CG. CR exhibits a slight advantage over CG in terms of function, gradient, and Hessian-vector evaluations on nonconvex problems, which makes it a viable general-purpose subproblem solver.

Paige and Saunders (1975) described the minimum residual method (MINRES), which generates the same iterates as CR in the definite case but is more general in that it also solves indefinite systems. Because the implementation of MINRES is substantially more complicated than that of CR, we do not consider it in this paper. We note however that it should be possible to implement our modified versions of CR as part of MINRES.

The rest of this paper is organized as follows. In section 2, we introduce the basic CR and its main properties. Section 3 gives a variant of CR in a linesearch inexact-Newton context, corresponding global and local convergence results, and numerical comparisons with CG. In section 4, we study CR in a trust-region context, provide theoretical results ensuring global convergence, and report on numerical experience. In section 5, we provide a variant of CR suitable for the solution of nonlinear least-squares problems in a trust-region context, and present numerical comparisons with the corresponding variant of CG. Concluding remarks appear in section 6. Complete details of our numerical experiments are provided in the appendix.

Notation. We use the Euclidean norm throughout. Uppercase Latin letters denote matrices, lowercase Latin letters denote vectors, Greek letters denote scalars.

2. Derivation of CR. Luenberger (1970) and Fong (2011) established properties of CR when H is positive definite. We base our description of CR on the pseudocode of Fong and Saunders (2012) with some modifications explained below.

The main idea behind CR is to solve (2) with H positive definite by solving the equivalent problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \ \bar{m}(s), \qquad \bar{m}(s) := \tfrac{1}{2}\|\nabla m(s)\|^2 = \tfrac{1}{2}\|g + Hs\|^2. \tag{4}$$

Note that $\nabla \bar{m}(s) = H(g + Hs)$ and $\nabla^2 \bar{m}(s) = H^2$. For any s, let $r := -g - Hs$ denote the residual of (2) so that $\nabla \bar{m}(s) = -Hr$. The algorithm starts from any s_0 and initializes $r_0 = -g - Hs_0$. It is customary to set $s_0 := 0$ and $r_0 := -g$. Given an iterate s_k and a descent direction p_k, CR computes a step length $\alpha_{k+1} > 0$ such that $s_{k+1} := s_k + \alpha_{k+1} p_k$ minimizes $\bar{m}$ in the direction p_k, i.e., such that $\nabla \bar{m}(s_{k+1})^T p_k = 0$. Thus, by construction, CR produces a monotonically decreasing sequence $\{\|r_k\|\}$.

Because $\bar{m}$ is quadratic,

$$\nabla \bar{m}(s_{k+1}) = \nabla \bar{m}(s_k) + \alpha_{k+1} \nabla^2 \bar{m}(s_k) p_k \tag{5}$$

and

$$\nabla \bar{m}(s_{k+1})^T p_k = -r_k^T H p_k + \alpha_{k+1} p_k^T H^2 p_k, \tag{6}$$

so that

$$\alpha_{k+1} = r_k^T H p_k / p_k^T H^2 p_k.$$

As we show in (16), $r_k^T H p_k = r_k^T H r_k$ when H is positive definite. Thanks to (5), the residual at s_{k+1} can be updated as $r_{k+1} = r_k - \alpha_{k+1} H p_k$. The next search direction is defined as $p_{k+1} := r_{k+1} + \beta_{k+1} p_k$, where β_{k+1} is chosen so that $p_{k+1}^T H^2 p_k = 0$, i.e., $\beta_{k+1} = -r_{k+1}^T H^2 p_k / p_k^T H^2 p_k$. Luenberger (1970, Theorem 1) shows that a consequence of the choice of p_{k+1} is that the search directions are H^2-conjugate and that $\beta_{k+1} = r_{k+1}^T H r_{k+1} / r_k^T H r_k$.

Algorithm 1 describes the classical CR. It differs from the one presented by Fong and Saunders (2012) by the addition of $\rho_k = \|r_k\|^2$, $\nu_k = \|r_k\|$, and a stopping criterion with absolute and relative tolerances.

Motivated by saddle-point systems and constrained problems, Luenberger (1970) modified CR to solve indefinite systems. His algorithm deviates from the standard CR when the step length computed is zero. Indeed, for singular H, $\alpha_{k+1} = 0$ may occur because $r_k^T H r_k$ may be equal to zero. In that case, the search direction $p_k = r_k$ and (6) implies that $\nabla \bar{m}(s_{k+1})^T p_k = 0$, so that the iterations are effectively stuck. An inconvenience of his approach is that one must decide numerically when the step length should be treated as zero.

For indefinite H, $\alpha_{k+1} < 0$ occurs when r_k is a direction of negative curvature: we follow a descent direction p_k for $\|Hs + g\|$, which may be an ascent direction for the quadratic model m. Such a situation is illustrated in the following simple example.

Example 2.1. For $n = 2$, let $m(s_1, s_2) = \frac{1}{2}(s_1^2 - s_2^2)$ and assume the first iterate is $(0, 1)$. The first CR search direction is the residual $p_0 = r_0 = (0, 1)$, which is a direction of negative curvature. The formula for α_1 above indicates that $\alpha_1 < 0$ and the next iterate is $(0, 0)$, a move that reduces $\|g + Hs\|$ to zero, but that causes an increase in $m(s)$.

Thus, our strategy differs in that we check the curvature and not the step length, in the spirit of the modified CG developed by Dembo and Steihaug (1983). The properties of standard CR continue to hold because our changes only apply when negative curvature is encountered, at which point the iterations stop.

Algorithm 1 CR for (2).

Require: H, g, $\tau_a > 0$, $\tau_r > 0$

Initialize: $k = 0$, $s_0 = 0$, $r_0 = -g$, $u_0 = Hr_0$, $\zeta_0 = r_0^T u_0$, $p_0 = r_0$, $q_0 = u_0$, $\rho_0 = r_0^T r_0$, $\nu_0 = \sqrt{\rho_0}$

while $\nu_k > \tau_a + \tau_r \|g\|$ **do**

- $k \leftarrow k + 1$
- $\alpha_k = \zeta_{k-1} / \|q_{k-1}\|^2$
- $s_k = s_{k-1} + \alpha_k p_{k-1}$
- $r_k = r_{k-1} - \alpha_k q_{k-1}$
- $\rho_k = \rho_{k-1} - \alpha_k \zeta_{k-1}$ $\qquad \rho_k = \|r_k\|^2$
- $\nu_k = \sqrt{\rho_k}$
- $u_k = Hr_k$
- $\zeta_k = r_k^T u_k$ $\qquad \zeta_k = r_k^T H r_k$
- $\beta_k = \zeta_k / \zeta_{k-1}$
- $p_k = r_k + \beta_k p_{k-1}$
- $q_k = u_k + \beta_k q_{k-1}$ $\qquad q_k = Hp_k$

return s_k

Theorem 1 summarizes properties of CR that we use frequently.

THEOREM 1. *In Algorithm* 1 *assume H is positive definite and $r_i \neq 0$ for $i = 0, \ldots, k$. The following properties hold:*

$$p_i^T H^2 p_j = 0, \quad i \neq j, \quad i, j = 0, \ldots, k, \tag{7}$$

$$r_i^T H r_j = 0, \quad i \neq j, \quad i, j = 0, \ldots, k, \tag{8}$$

$$p_i^T H r_j = 0, \quad 0 \leq i < j \leq k, \tag{9}$$

$$r_i^T H r_i > 0, \quad i = 0, \ldots, k, \tag{10}$$

$$\operatorname{span}\{r_0, r_1, \ldots, r_k\} = \operatorname{span}\{g, Hg, \ldots, H^k g\}, \tag{11}$$

$$\|s_i\| < \|s_j\|, \quad i < j, \quad i, j = 0, \ldots, k, \tag{12}$$

$$\alpha_i > 0, \quad i = 1, \ldots, k, \tag{13}$$

$$p_i^T r_j > 0, \quad 0 \leq i, j \leq k. \tag{14}$$

Proof. Properties (7), (8), and (9) respectively come from Luenberger (1970, Theorem 1(a), (d), and (b)), (10) follows from the positive definiteness of H, (12) was proven by Fong (2011, Theorem 2.1.6), and (13)–(14) come from Fong (2011, Theorem 2.1.5).

By construction, s_i minimizes the norm of the residual in K_i. We show by induction that $r_i \in K_{i+1}$ and $p_i \in K_{i+1}$. Initially, $r_0 = p_0 = g \in K_1$. Let the assumption be satisfied for index i. We have $r_{i+1} = r_i - \alpha_{i+1} H p_i$. Since r_i and p_i are in K_{i+1}, we know $Hp_i \in K_{i+2}$ and so $r_{i+1} \in K_{i+2}$. Also, $p_{i+1} = r_{i+1} + \beta_{i+1} p_i$ so $p_{i+1} \in K_{i+2}$, which establishes (11). ∎

Our next result highlights quantities computed recursively in Algorithm 1.

THEOREM 2. *Assume H is positive definite in Algorithm 1. For all $k \geq 0$,*

$$q_k = Hp_k, \tag{15}$$

$$\zeta_k = r_k^T q_k = r_k^T H r_k = r_k^T H p_k, \tag{16}$$

$$\rho_k = \|r_k\|^2, \tag{17}$$

$$\nu_k = \|r_k\|. \tag{18}$$

Proof. Equality (15) can be established by induction. Initially, $q_0 = u_0 = Hr_0 = Hp_0$. Now assume that the property holds for index k. Then, by recurrence,

$$q_{k+1} = u_{k+1} + \beta_{k+1} q_k = Hr_{k+1} + \beta_{k+1} Hp_k = H(r_{k+1} + \beta_{k+1} p_k) = Hp_{k+1}.$$

We have $r_k^T q_k = r_k^T(u_k + \beta_k q_{k-1})$. But (9) and (15) yield $r_k^T q_{k-1} = 0$, so $r_k^T q_k = r_k^T H r_k = \zeta_k$, which establishes (16).

Because $\rho_0 = r_0^T r_0 = \|r_0\|^2$, (17) holds for $k = 0$. Then, (9), (15), (16), and a recursion assumption yield

$$\|r_{k+1}\|^2 = r_{k+1}^T (r_k - \alpha_{k+1} q_k) = (r_k - \alpha_{k+1} q_k)^T r_k = \rho_k - \alpha_{k+1} \zeta_k = \rho_{k+1}.$$

Property (18) follows because $\nu_k = \sqrt{\rho_k}$. ∎

The following result guides the detection of nonpositive curvature in the next sections, and parallels a similar result for CG. Its proof is as in Dembo and Steihaug (1983) with directions d_k replaced by residuals r_k.

THEOREM 3 (Dembo and Steihaug (1983, Theorem A.5)). *Consider Algorithm 1. Let l be the number of distinct eigenvalues of H. If g has a nonzero projection on each eigenspace and if*

$$r_i^T H r_j = 0, \quad i, j = 0, \ldots, l-1 \ (i \neq j), \tag{19}$$

$$r_i^T H r_i > 0, \quad i = 0, \ldots, l-1, \tag{20}$$

$$\operatorname{span}\{r_0, \ldots, r_{l-1}\} = \operatorname{span}\{g, Hg, \ldots, H^{l-1} g\}, \tag{21}$$

then H is positive definite.

3. CR in a linesearch context.

In this section, we are interested in solving (1) by way of a linesearch method. The main motivation is that because CR produces monotonic $\|r_k\|$, it appears suitable for computing steps in an inexact Newton scheme. The linesearch subproblem is described by (2). As f may be nonconvex, H may not be positive definite.

3.1. Linesearch CR algorithm.

Let $\pi_i = p_i^T p_i$, $\rho_i = r_i^T r_i$ for $i = 0, 1, \ldots, k$ and $\epsilon > 0$ be a constant value. In Algorithm 2 we modify CR in the context of a linesearch method to solve (2) when H is not necessarily positive definite.

The main difference from Algorithm 1 resides in the condition on line 4 and recursion for π_k. We first note that Theorem 1 continues to hold for Algorithm 2 for as long as the search directions and residuals are directions of positive curvature.

COROLLARY 1. *Let $\epsilon > 0$. If $p_i^T H p_i > \epsilon \pi_i$ and $r_i^T H r_i > \epsilon \rho_i$ for $i = 0, \ldots, k$, then the properties of Theorem 1 hold for Algorithm 2 at iterations $i = 0, \ldots, k$.*

Because of Theorem 3, there may be an iteration k such that $r_k^T H r_k \leq 0$ if H is not positive definite and Algorithm 2 does not terminate earlier. Thus, the residual

Algorithm 2 Modified CR (linesearch version).

Require: H, g, $\tau_a > 0$, $\tau_r > 0$, $\epsilon > 0$

Initialize: $k = 0$, $s_0 = 0$, $r_0 = -g$, $u_0 = Hr_0$, $\zeta_0 = r_0^T u_0$, $p_0 = r_0$, $q_0 = u_0$, $\delta_0 = \zeta_0$, $\rho_0 = r_0^T r_0$, $\mu_0 = \rho_0$, $\pi_0 = \rho_0$, $\nu_0 = \sqrt{\rho_0}$

while $\nu_k > \tau_a + \tau_r \|g\|$ **do**

  $k \leftarrow k + 1$

  if $\delta_{k-1} \le \epsilon \pi_{k-1}$ or $\zeta_{k-1} \le \epsilon \rho_{k-1}$ **then** *(near) negative curvature detected*

    if $k = 1$ **then**

      return $-g$

    else

      return s_{k-1}

  $\alpha_k = \zeta_{k-1}/\|q_{k-1}\|^2$

  $s_k = s_{k-1} + \alpha_k p_{k-1}$

  $r_k = r_{k-1} - \alpha_k q_{k-1}$

  $\rho_k = \rho_{k-1} - \alpha_k \zeta_{k-1}$ $\qquad \rho_k = \|r_k\|^2$

  $\nu_k = \sqrt{\rho_k}$ $\qquad \nu_k = \|r_k\|$

  $u_k = Hr_k$

  $\zeta_k = r_k^T u_k$ $\qquad \zeta_k = r_k^T H r_k$

  $\beta_k = \zeta_k/\zeta_{k-1}$

  $p_k = r_k + \beta_k p_{k-1}$

  $\pi_k = \rho_k + 2\beta_k(\mu_{k-1} - \alpha_k \delta_{k-1}) + \beta_k^2 \pi_{k-1}$ $\qquad \pi_k = \|p_k\|^2$

  $\mu_k = \rho_k + \beta_k(\mu_{k-1} - \alpha_k \delta_{k-1})$ $\qquad \mu_k = p_k^T r_k$

  $q_k = u_k + \beta_k q_{k-1}$ $\qquad q_k = Hp_k$

  $\delta_k = \zeta_k + \beta_k^2 \delta_{k-1}$ $\qquad \delta_k = p_k^T H p_k$

return s_k

will eventually signal the presence of nonpositive curvature. However, because of (7), we know that the search directions are conjugate with respect to H^2 and not H. We must check the sign of $p_k^T H p_k$ explicitly at each iteration to discover whether p_k is a direction of positive curvature or not. Example 2.1 illustrates that p_k can be an ascent direction if H is not positive definite. If nearly negative curvature is detected, either the previous iterate s_{k-1} is returned if $k \ge 2$ or $-g$ is returned if $k = 1$, as it is known to be a descent direction.

THEOREM 4. *Whether H is positive definite or not, identities* (15)–(18) *continue to hold for Algorithm* 2. *The following equalities also hold:*

$$\delta_k = p_k^T H p_k, \tag{22}$$

$$\mu_k = p_k^T r_k, \tag{23}$$

$$\pi_k = \|p_k\|^2. \tag{24}$$

Proof. The proof of (15)–(18) is the same as in that of Theorem 2. We have $\delta_0 = \zeta_0$ and $\forall k \ge 1$, $\delta_k = \zeta_k + \beta_k^2 \delta_{k-1}$. By induction we can show that $\delta_k = p_k^T H p_k$. Indeed, since $p_0 = r_0$ we have $\delta_0 = p_0^T H p_0$. Now suppose the property is satisfied for $k \ge 0$. Then,

$$p_{k+1}^T H p_{k+1} = p_{k+1}^T q_{k+1} = (r_{k+1} + \beta_{k+1} p_k)^T (u_{k+1} + \beta_{k+1} q_k).$$

Since H is symmetric, $p_k^T u_{k+1} = r_{k+1}^T q_k = 0$ according to (9). This leads to

$$p_{k+1}^T H p_{k+1} = r_{k+1}^T u_{k+1} + \beta_{k+1}^2 p_k^T q_k = \zeta_{k+1} + \beta_{k+1}^2 \delta_k = \delta_{k+1}.$$

Thus, (22) is established.

We have $\mu_0 = \rho_0 = \|r_0\|^2$. But $r_0 = p_0$, so (23) is satisfied at iteration 0. Suppose by induction that it is verified at iteration $k \geq 0$. Then

$$\begin{aligned} p_{k+1}^T r_{k+1} &= (r_{k+1} + \beta_{k+1} p_k)^T r_{k+1} \\ &= \rho_{k+1} + \beta_{k+1} p_k^T r_{k+1} \\ &= \rho_{k+1} + \beta_{k+1} p_k^T (r_k - \alpha_{k+1} q_k) \\ &= \rho_{k+1} + \beta_{k+1} \mu_k - \alpha_{k+1} \beta_{k+1} \delta_k \\ &= \mu_{k+1}, \end{aligned}$$

which establishes (23).

Similarly, (24) can be shown by induction. First, $\pi_0 = \rho_0 = \|p_0\|^2$. If $k \geq 0$ satisfies the property, then

$$p_{k+1}^T p_{k+1} = (r_{k+1} + \beta_{k+1} p_k)^T (r_{k+1} + \beta_{k+1} p_k) = \rho_{k+1} + 2\beta_{k+1} r_{k+1}^T p_k + \beta_{k+1}^2 \pi_k.$$

The update for r_{k+1} yields

$$r_{k+1}^T p_k = (r_k - \alpha_{k+1} q_{k-1})^T p_k = \mu_k - \alpha_{k+1} \delta_k.$$

Finally,

$$\pi_{k+1} = \rho_{k+1} + 2\beta_{k+1}(\mu_k - \alpha_{k+1}\delta_k) + \beta_{k+1}^2 \pi_k. \qquad \square$$

The last result of this section follows from Theorem 3 and justifies that if g is not orthogonal to the eigenspaces of H associated with nonpositive eigenvalues, Algorithm 2 eventually detects nonpositive curvature.

THEOREM 5 (Dembo and Steihaug (1983, Theorem 2.4)). *In Algorithm 2, let $\tau_a = \tau_r = \epsilon = 0$. If H is not positive definite, either*
1. *$r_k^T H r_k \leq 0$ for a certain iteration k, or*
2. *g is orthogonal to the eigenspaces associated with the nonpositive eigenvalues of H.*

3.2. Global convergence. In this subsection we show that the truncated-Newton method paired with Algorithm 2 is globally convergent. Our analysis follows that of Dembo and Steihaug (1983), and so does Algorithm 3.

LEMMA 1 (Dembo and Steihaug (1983, Lemma A.2)). *Let $\epsilon > 0$ be as in Algorithm 2. Assume that there exists $M > 0$ such that $\|H_j\| \leq M$ for all j in Algorithm 3. If $g_j \neq 0$, there exist constants $\gamma_0 > 0$ and $\gamma_1 > 0$ that only depend on ϵ and M such that*

$$g_j^T s_j \leq -\gamma_0 \|g_j\|^2 \quad \text{and} \tag{27}$$

$$\|s_j\| \leq \gamma_1 \|g_j\|. \tag{28}$$

Proof. In step 5 of Algorithm 3, Algorithm 2 is initialized with $r_0 = -g_j$. If $r_0^T H_j r_0 \leq \epsilon \rho_0$, then $s_j = -g_j$ and we may choose $\gamma_0 = 1$ in (27). Otherwise,

Algorithm 3 Truncated Newton method for (1).

Require: x_0, $\epsilon_a > 0$, $\epsilon_r > 0$

Compute: $f_0 = f(x_0)$, $g_0 = g(x_0)$, set $j = 0$
while $\|g_j\| > \epsilon_a + \epsilon_r\|g_0\|$ **do**
  $j \leftarrow j + 1$
  Choose $H_{j-1} = H_{j-1}^T \approx \nabla^2 f(x_{j-1})$, $\tau_a > 0$, $\tau_r > 0$
  Compute s_{j-1} such that $\|H_{j-1}s_{j-1} + g_{j-1}\| \le \tau_a + \tau_r\|g_{j-1}\|$ *use Algorithm 2*
  Compute $t_{j-1} > 0$ that satisfies the Wolfe conditions

$$f(x_{j-1} + t_{j-1}s_{j-1}) \le f_{j-1} + \alpha t_{j-1} g_{j-1}^T s_{j-1}, \qquad \alpha \in (0, \tfrac{1}{2}) \tag{25}$$

$$\nabla f(x_{j-1} + t_{j-1}s_{j-1})^T s_{j-1} \ge \beta g_{j-1}^T s_{j-1}, \qquad \beta \in (\alpha, 1) \tag{26}$$

  $x_j = x_{j-1} + t_{j-1}s_{j-1}$
  $g_j = \nabla f(x_j)$

let Algorithm 2 terminate after k iterations with s_{j_k}, which corresponds to s_j in Algorithm 3. The kth iteration of Algorithm 2 sets $s_{j_k} = \sum_{i=1}^k \alpha_i p_{i-1}$, where $\alpha_i = r_{i-1}^T H_j r_{i-1}/\|H_j p_{i-1}\|^2$. Thus, $g_j^T s_{j_k} = -\sum_{i=1}^k \alpha_i r_0^T p_{i-1}$. For $i = 1, \dots, k$, we have from Corollary 1 and (10) that $r_{i-1}^T H_j r_{i-1} > 0$, while (14) yields $r_0^T p_{i-1} > 0$. Hence, for $i = 1, \dots, k$,

$$g_j^T s_{j_k} \le -r_0^T p_0 \frac{r_0^T H_j r_0}{\|H_j p_0\|^2} \le -\|g_j\|^2 \frac{g_j^T H_j g_j}{\|H_j\|^2 \|g_j\|^2} \le -\frac{g_j^T H_j g_j}{\|H_j\|^2}.$$

But $g_j^T H_j g_j > \epsilon g_j^T g_j$, so $g_j^T s_{j_k} \le -\epsilon\|g_j\|^2/\|H_j\|^2 \le -\epsilon\|g_j\|^2/M^2$. Thus, (27) holds with $\gamma_0 = \min(1, \epsilon/M^2)$.

Similarly, if $s_j = s_{j_k} = -g_j$, we may choose $\gamma_1 = 1$ in (28). Otherwise, we have from (16) and Theorem 4 that

$$\|s_{j_k}\| = \left\| \sum_{i=1}^k \frac{r_{i-1}^T H_j r_{i-1}}{\|H_j p_{i-1}\|^2} p_{i-1} \right\| \le \sum_{i=1}^k \frac{p_{i-1}^T H_j r_{i-1}}{\|H_j p_{i-1}\|^2} \|p_{i-1}\| \le \sum_{i=1}^k \frac{\|p_{i-1}\|^2 \|H_j\| \|r_{i-1}\|}{\|H_j p_{i-1}\|^2}.$$

Because Algorithm 2 did not stop before iteration k, we have for $i = 1, \dots, k$ that $\epsilon\|p_{i-1}\|^2 < p_{i-1}^T H_j p_{i-1} \le \|p_{i-1}\|\|H_j p_{i-1}\|$, so that $\|H_j p_{i-1}\| > \epsilon\|p_{i-1}\|$. Moreover, $\|H_j\| \le M$ and, by design, Algorithm 2 ensures that $\|r_{i-1}\| \le \|r_0\|$ for all $i = 1, \dots, k$. Combining the above yields

$$\|s_{j_k}\| \le k\frac{M}{\epsilon^2}\|r_0\| \le n\frac{M}{\epsilon^2}\|g_j\|,$$

which concludes the proof with $\gamma_1 = \max(1, n\frac{M}{\epsilon^2})$. □

Global convergence of Algorithm 3 follows from the next two results, whose proofs are identical to those of Dembo and Steihaug (1983, Theorems 2.1 and A.3).

THEOREM 6 (Dembo and Steihaug (1983, Theorem A.3)). *Consider Algorithm* 3 *in which steps are computed using Algorithm* 2. *The sequence of iterates* $\{x_j\}$ *is well defined and* $\lim_{j\to\infty} \|g_j\| = 0$.

THEOREM 7 (Dembo and Steihaug (1983, Theorem 2.1)). *If the sequence $\{x_j\}$ generated by Algorithm 3 has a limit point x^* where $H(x^*)$ is positive definite, then the whole sequence $\{x_j\}$ converges to x^*.*

3.3. Local convergence. We now show that Algorithm 3 has fast local convergence properties provided the tolerances τ_a and τ_r are chosen appropriately.

LEMMA 2. *Consider Algorithm 3 in which steps are computed using Algorithm 2, and assume that $g_j \neq 0$ and that there exists $M > 0$ such that $\|H_j\| \leq M$ and H_j is positive definite for all j. If $\tau_a = 0$ and $\tau_r = o(1)$, then $\|g_j + H_j s_j\| = o(\|s_j\|)$.*

Proof. The termination condition of Algorithm 2 is $\|g_j + H_j s_j\| \leq \tau_j$ with $\tau_j = \tau_a + \tau_r \|g_j\| = o(\|g_j\|)$ by assumption. Moreover, Algorithm 2 performs at least one iteration. Indeed, if this were not the case and $s_j = 0$, that would mean that $\|g_j\| \leq \tau_r \|g_j\|$, and therefore that $g_j = 0$, so that Algorithm 3 would have stopped earlier. In the call to Algorithm 2 at the jth iteration of Algorithm 3, we denote by $p_{j_0} = -g_j$ the initial search direction, by α_{j_1} the first step length, and by $s_{j_1} = \alpha_{j_1} p_{j_0}$ the updated iterate. Because H_j is positive definite,

$$g_j^T H_j g_j = p_{j_0}^T H_j p_{j_0} > \epsilon \|p_{j_0}\|^2 = \epsilon \|g_j\|^2.$$

By (12), $\|s_j\| \geq \|s_{j_1}\| = \alpha_{j_1} g_j$. Thus,

$$\frac{\|g_j + H_j s_j\|}{\|s_j\|} \leq \frac{\tau_j}{\alpha_{j_1}\|g_j\|} = \frac{\|H_j g_j\|^2}{g_j^T H_j g_j} \frac{\tau_j}{\|g_j\|} \leq \frac{\|H_j\|^2 \|g_j\|^2}{g_j^T H_j g_j} \frac{\tau_j}{\|g_j\|} \leq \frac{M^2}{\epsilon} \frac{\tau_j}{\|g_j\|}.$$

Because $\tau_j = o(\|g_j\|)$, we have $\|g_j + H_j s_j\| = o(\|s_j\|)$. ☐

Lemma 2 yields the following result, which parallels Theorem 2.2 of Dembo and Steihaug (1983) and is an application of Theorem 6.4 of Dennis and Moré (1977).

THEOREM 8. *Let the sequence of iterates $\{x_j\}$ generated by Algorithm 3 converge to x^* such that $H(x^*)$ is positive definite. If the Wolfe conditions (25) and (26) are satisfied and $\|g_j + H_j s_j\| = o(\|s_j\|)$, then for ϵ sufficiently small in Algorithm 2, there exists an iteration $\bar{j}$ such that the stopping criterion on the residual norm in Algorithm 2 is satisfied and such that for all $j \geq \bar{j}$, $t_j = 1$ is an acceptable step length for the linesearch.*

The following result states the local convergence rate. The proof is identical to that of Theorem 2.3 of Dembo and Steihaug (1983).

THEOREM 9 (Dembo and Steihaug (1983, Theorem 2.3)). *Let the sequence $\{x_j\}$ generated by Algorithm 3 converge to x^* where $H(x^*)$ is positive definite. Suppose that H is Lipschitz continuous at x^*. If the jth iteration of Algorithm 3 sets $\tau_a = 0$ and $\tau_r = \min(1/j, \|g(x_j)\|^t)$ for some $0 < t \leq 1$, then $\{x_j\}$ converges at a rate $1 + t$.*

3.4. Numerical results. In this section, we compare the performance of Algorithm 3 using CG or CR to compute steps. We use the Julia[1] programming language (version 0.7) to implement the truncated CG algorithm of Dembo and Steihaug (1983), Algorithm 2, and Algorithm 3.

We use convex and nonconvex unconstrained problems from the CUTEst collection of Gould, Orban, and Toint (2015), accessed via the CUTEst.jl[2] Julia interface, and

[1] See julialang.org.
[2] See github.com/JuliaSmoothOptimizers/CUTEst.jl.

the modified `CUTE` problems of Lukšan, Matonoha, and Vlček (2010) as implemented in the Julia library OptimizationProblems.jl.[3] We exclude the CUTEst nonlinear least-squares problems for which there exists a variant formulated as a feasibility problem in which the equality constraints play the role of the residual, as those are evaluated in section 5. We keep the least-squares problems that are only available in the form of unconstrained problems with a sum-of-squares objective. We further eliminate problems with fewer than 10 variables. There remain 91 problems from CUTEst and 7 problems from OptimizationProblems that are not available in CUTEst. We use all problems in their default dimension with the exception of *scosine*, for which we use `N=100` instead of the default `N=5000` as the latter took over 1.5 hours to solve.

Among our problems, 16 are known to be convex (see Fourer et al. (2010)): *arglina*, *arglinb*, *arglinc*, *bdqrtic*, *clplatea*, *clplateb*, *clplatec*, *dixon3dq*, *dqdrtic*, *dqrtic*, *engval1*, *nondquar*, *power*, *quartc*, *tridia*, and *vardim*. Another 58 are known to be nonconvex. The remaining 24 have unknown convexity.

In Algorithm 2 and truncated CG, we set the maximum number of iterations to the number of variables. We impose a maximum of 10,000 iterations in Algorithm 3 and set $\epsilon_a = \epsilon_r = 10^{-6}$, and $\tau_a = 0$, $\tau_r = \min(0.1, \sqrt{\|g_j\|})$ in the hope of encouraging local superlinear convergence—see Theorem 9. Finally, we use a simple Armijo backtracking linesearch with $\alpha = 10^{-4}$, which only ensures (25). This is the way Algorithm 3 is often implemented in practice.

Our results are presented in the form of $\log_2$-scaled performance profiles of the number of objective evaluations, gradient evaluations, Hessian-vector products, and the sum of the three measures. We report results on convex and nonconvex problems separately, as convex problems do not trigger the modifications to CR pertaining to negative curvature. We also report results on the entire problem set.

Figure 1 contains the profiles for convex problems and shows that CG and CR perform equivalently, though CR fails on one problem.

Figure 2 illustrates that both methods are essentially equivalent on nonconvex problems, except that CG shows savings in terms of gradient evaluations, and therefore in terms of iterations of Algorithm 3. CR shows slight savings in terms of Hessian-vector products, indicating that fewer iterations of Algorithm 2 than of CG are necessary to attain the stopping criterion. CR and CG fail on five and four problems, respectively.

Profiles for all 98 problems together appear in Figure 3, where we observe the same trends as in Figure 2. CR fails on 9 problems overall while CG fails on 5 problems. On those, the maximum number of iterations of Algorithm 3 is reached, except for problem *parkch*, where the magnitude of the negative curvature directions identified by Algorithm 2 becomes so large that we eventually evaluate the log likelihood objective with arguments that result in NaNs.

4. CR in a trust-region context.

Most large-scale implementations of trust-region methods compute an approximate solution s of (3) using the truncated CG method of Steihaug (1983). The basic mechanism is to temporarily ignore the trust-region constraint on the step and apply CG as if m were convex. The fundamental properties of CG, reviewed in subsection 4.1, make it particularly well suited to this task. Steihaug (1983) described simple modifications to CG to account for situations where the next CG iterate lies outside the trust region, or where the current CG search direction p is a direction of nonpositive curvature for m, i.e., $p^T Hp \le 0$. Yuan (2000) established that truncated CG yields an approximate minimizer s^C such that

[3] See github.com/JuliaSmoothOptimizers/OptimizationProblems.jl.

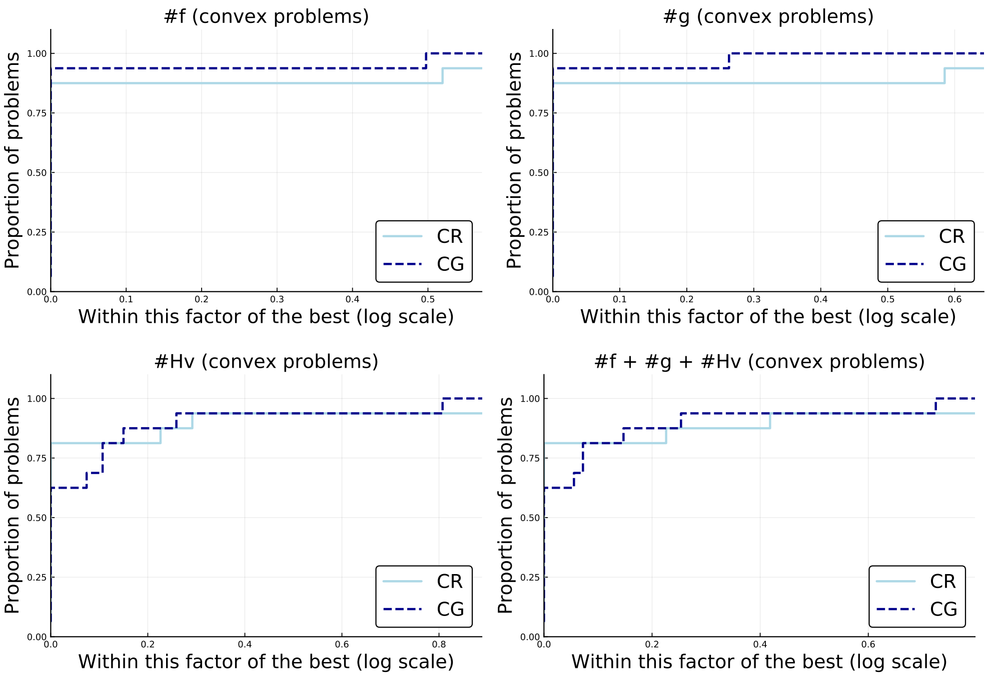

FIG. 1. *Performance of linesearch CR and CG to solve* 16 *convex problems in terms of evaluations of f, g, and products with H.*

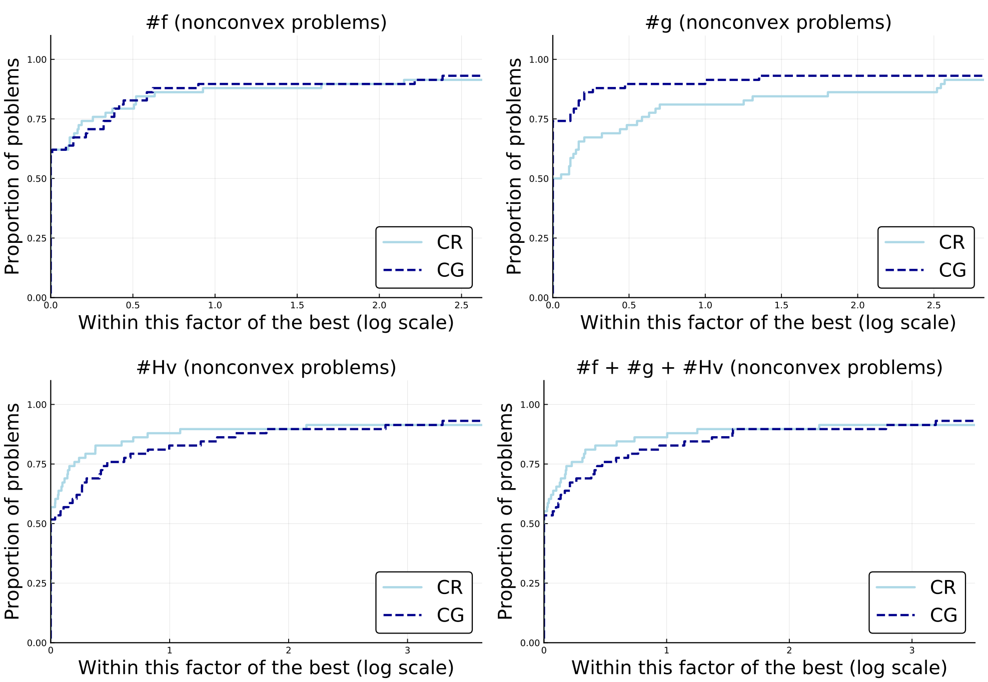

FIG. 2. *Performance of linesearch CR and CG on* 58 *nonconvex problems in terms of evaluations of f, g, and products with H.*

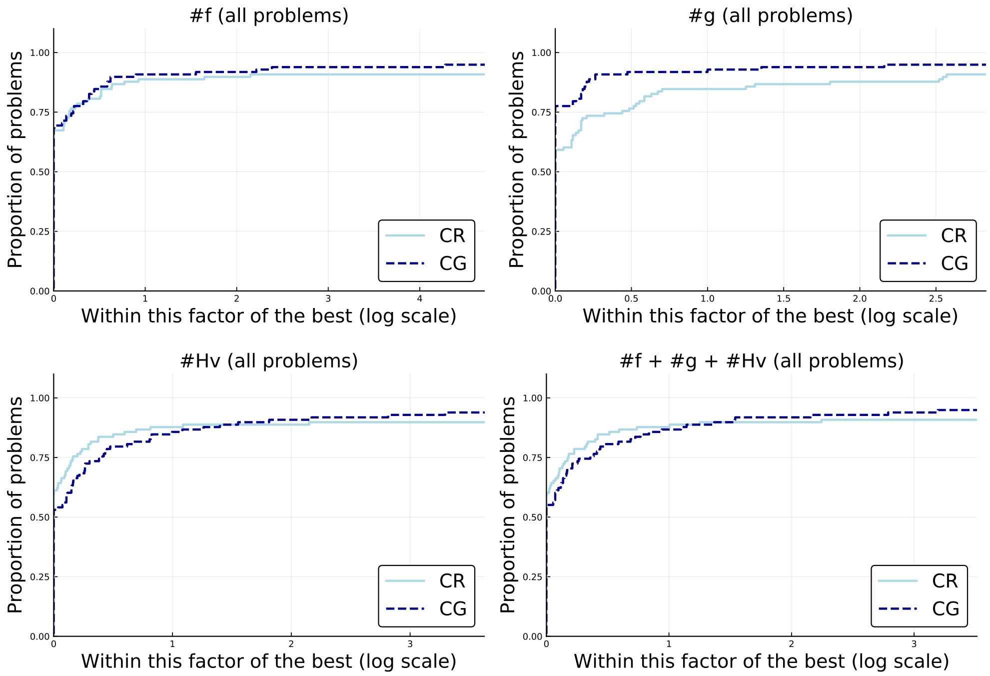

FIG. 3. *Performance of linesearch CR and CG on* 98 *problems in terms of evaluations of* f, g, *and products with* H.

$m(s^C) \le \frac{1}{2}m(s^*) \le 0$, where s^* is a global minimizer of (3). Algorithm 4 gives the trust-region process. In the remainder of this section, we introduce a version of CR to be used at line 7. See Conn, Gould, and Toint (2000) for a comprehensive account of trust-region methods.

4.1. Background. Convergence of trust-region methods to a stationary point hinges around the concept of decrease obtained at the Cauchy point, which is determined by the solution of

$$\underset{\alpha}{\text{minimize}} \;\; m_j(-\alpha g_j) \quad \text{subject to} \;\; 0 \le \alpha \le \Delta_j/\|g_j\|,$$

where $m_j(s) := g_j^T s + \frac{1}{2}s^T H_j s$ is defined as in (3), and Δ_j is the current trust-region radius. The Cauchy point is the minimizer of the model along the steepest-descent direction and inside the trust region. It may or may not lie on the boundary. If m_j is convex along the steepest-descent direction but the unconstrained minimizer lies outside the trust region, or if m_j is concave along the steepest-descent direction, the Cauchy point lies on the boundary. However we compute an approximate solution s_j of (3), and convergence will be guaranteed provided we satisfy the sufficient decrease condition, which requires that our step achieve at least a fixed fraction of Cauchy decrease at each trust-region subproblem. The sufficient-decrease condition demands that there exist a constant $\kappa \in (0, 1)$, independent of j, such that

$$m_j(s_j) \le -\kappa \|g_j\| \min\left(\frac{\|g_j\|}{1 + \|H_j\|}, \Delta_j\right). \tag{29}$$

We refer the reader to subsection 6.3.4 of Conn, Gould, and Toint (2000) for additional information.

Algorithm 4 Trust-region method for (1).

Require: x_0, $\Delta_0 > 0$, $\epsilon_a > 0$, $\epsilon_r > 0$, $0 < \eta_1 \le \eta_2 < 1$, $0 < \gamma_1 \le \gamma_2 < 1$
evaluate $f_0 = f(x_0)$, $g_0 = \nabla f(x_0)$
initialize $j = 0$
while $\|g_j\| > \epsilon_a + \epsilon_r \|g_0\|$ **do**
  $j \leftarrow j + 1$
  choose $H_{j-1} = H_{j-1}^T \approx \nabla^2 f(x_{j-1})$
  choose absolute and relative tolerances $0 < \tau_a, \tau_r < 1$
  compute s_{j-1} by approximately solving (3); *use Algorithm 5*
    stop if $\|H_{j-1}s_{j-1} + g_{j-1}\| \le \tau_a + \tau_r \|g_{j-1}\|$ or $\|s_{j-1}\| = \Delta_{j-1}$
  compute $\sigma_{j-1} = (f_{j-1} - f(x_{j-1} + s_{j-1}))/(m_{j-1}(0) - m_{j-1}(s_{j-1}))$
  if $\sigma_{j-1} < \eta_1$ **then** *unsuccessful iteration*
    $x_j = x_{j-1}$, $f_j = f_{j-1}$, $g_j = g_{j-1}$
    set $\Delta_j \in [\gamma_1 \Delta_{j-1}, \gamma_2 \Delta_{j-1}]$
  else *successful iteration*
    $x_j = x_{j-1} + s_{j-1}$
    evaluate $f_j = f(x_j)$, $g_j = \nabla f(x_j)$
    if $\sigma_{j-1} \ge \eta_2$ **then** *very successful iteration*
      set $\Delta_j \in [\Delta_{j-1}, \infty)$
    else
      set $\Delta_j \in [\gamma_2 \Delta_{j-1}, \Delta_{j-1}]$
return x_j

If we neglect the trust-region constraint temporarily, the first-order optimality condition of (3) may be stated as the symmetric linear system $H_j s = -g_j$. For simplicity, we drop the subscript j in the rest of this section as we will only be interested in a single trust-region subproblem, and rewrite the linear system as

$$Hs = -g. \tag{30}$$

Accordingly, our quadratic model becomes $m(s) = g^T s + \frac{1}{2} s^T H s$.

Both CG and CR can be constructed from the Lanczos (1950) process, which theoretically generates an orthonormal basis $\{v_1, v_2, v_3, \dots\}$ of the increasing Krylov subspaces $\text{span}\{-g, -Hg, -H^2 g, \dots\}$.

CG has several desirable properties that make it a natural candidate for (3), even when H is indefinite. Those properties follow from the definition of the method and Theorem 2.1 of Steihaug (1983) and are summarized in the following result.

THEOREM 10. *Assume H is positive definite and s_k are the iterates generated by CG on $Hs = -g$. Then,*

1. $\|s_{k+1}\| > \|s_k\|$ *for* $k = 0, 1, \dots,$
2. $s_k \in \text{argmin}\,\{m(s) \mid s \in \text{span}\{v_1, \dots, v_k\}\}$ *for* $k = 1, 2, \dots,$
3. *the function* $\alpha \mapsto m(s_k + \alpha(s_{k+1} - s_k))$ *is monotonically decreasing over* $[0,\, 1]$ *for* $k = 0, 1, \dots.$

The properties of Theorem 10 are important in the solution of (3) for the following reasons. First, property 2 is relevant because minimizing $m(s)$ is the end goal. For the case in which the solution of (3) lies in the trust region, standard CG will identify it in at most n iterations in exact arithmetic. If the problem is convex and the solution lies outside the trust region, property 1 of the iterates implies that there will be an iteration k such that $\|s_k\| \le \Delta$ but $\|s_{k+1}\| > \Delta$. By property 3, we may compute $\alpha \in [0, 1]$ such that $\|s_k + \alpha(s_{k+1} - s_k)\| = \Delta$ and possibly further improve m. If H is indefinite, CG is guaranteed to observe negative curvature along one of its search directions provided Δ is sufficiently large, because those search directions, denoted p_k, are *conjugate* with respect to H, i.e, they satisfy $p_i^T H p_k = 0$ if $i \neq k$. Note that this property is not guaranteed for the Lanczos vectors v_i. Thus if CG encounters a situation where $\|s_k\| < \Delta$ and p_k is a direction of negative curvature ($p_k^T H p_k < 0$), then p_k may simply be followed to the boundary of the trust region because it is guaranteed to be a descent direction. Further information on the procedure just outlined, referred to as the *truncated conjugated gradient* method, or the *Steihaug–Toint* strategy, may be found in Steihaug (1983) and subsection 7.5.1 of Conn, Gould, and Toint (2000).

In contrast, CR aims to reduce the residual norm $\|g + Hs\|$ at each iteration. The following result summarizes properties of CR on a positive definite system that are relevant in a trust-region context.

THEOREM 11 (Fong and Saunders (2012, Theorems 2.3 and 2.5)). *Assume H is positive definite and s_k are the iterates generated by CR on $Hs = -g$. Then,*

1. $\|s_{k+1}\| \ge \|s_k\|$ *for* $k = 0, 1, \ldots,$
2. $s_k \in \operatorname{argmin}\{\|g + Hs\| \mid s \in \operatorname{span}\{v_1, \ldots, v_k\}\}$ *for* $k = 1, 2, \ldots,$
3. $m(s_k + t(s_{k+1} - s_k)) \le m(s_k)$ *for* $t \in [0,\ 1]$ *and* $k = 0, 1, \ldots,$
4. $m(s_{k+1}) < m(s_k)$ *for* $k = 0, 1, \ldots.$

Theorem 11 shows that CR possesses properties closely related to those of CG and suggests that CR may be viable as a trust-region subproblem solver. In the next section, we examine how standard CR may be modified to solve (3).

4.2. Truncated CR. In this subsection, we describe the theory behind our truncated CR for trust regions. To explain the strategy, we consider curvatures and projections in exact arithmetic. These quantities will be relaxed numerically.

If H is not positive definite, m is nonconvex and Algorithm 1 no longer ensures that the search directions p_k are descent directions, as shown by Example 2.1. Fong (2011, Theorem 2.1.5) shows that $-p_k^T \nabla m(s_k) = p_k^T r_k > 0$ as long as $p_k^T H p_k > 0$, and $\alpha_{k+1} > 0$ as long as $r_k^T H r_k > 0$. Theorem 1 shows that in CR, the search directions p_k are conjugate with respect to H^2, while the residuals r_k are conjugate with respect to H for as long as positive curvature holds. Thus, $r_k^T H r_k$ is a reliable indicator of convexity, but it is possible to observe $p_k^T H p_k \le 0$ before $r_k^T H r_k \le 0$, and so we must monitor the sign of both quantities at each iteration.

In Algorithm 1, we see that if $r_k^T H r_k = 0$, then $\beta_k = \zeta_k = 0$, so $p_k = r_k$. Therefore, $r_k^T H r_k = 0$ is a special case of $p_k^T H p_k = 0$. But $\zeta_k = 0$ implies $\alpha_{k+1} = 0$, so if nothing is changed in Algorithm 1, an error occurs because $\zeta_{k+1} = r_{k+1}^T H r_{k+1} = 0$ and β_{k+1} is undefined.

If either p_k or r_k is a negative curvature direction, we follow the one that produces the best decrease as described in Case III below. Our overall strategy is such that as soon as nonpositive curvature is detected, s_k is updated one last time and the algorithm stops.

Algorithm 5 describes truncated CR. A separate procedure, described later in Algorithm 6, returns a boolean `terminate`, a search direction p_{k-1}, and a step length α_k. As long as `terminate` is `false`, Algorithm 5 is equivalent to Algorithm 1 used in a trust-region context; `terminate` is `true` when either nonpositive curvature is discovered or s_k is on the boundary of the trust region. If m is convex, or if a solution

Algorithm 5 CR for (3) (trust-region version).

Require: H, g, $\Delta > 0$, $\tau_a > 0$, $\tau_r > 0$

Initialize: $k = 0$, $s_0 = 0$, $r_0 = -g$, $u_0 = Hr_0$, $\zeta_0 = r_0^T u_0$, $p_0 = r_0$, $q_0 = u_0$, $\delta_0 = \zeta_0$, $\rho_0 = r_0^T r_0$, $\nu_0 = \sqrt{\rho_0}$, $\mu_0 = \rho_0$

while $\nu_k > \tau_a + \tau_r \|g\|$ **do**

  $k \leftarrow k + 1$

  compute `terminate`, α_k, and p_{k-1} using Algorithm 6

  $s_k = s_{k-1} + \alpha_k p_{k-1}$

  if `terminate` **then return** s_k *m is nonconvex or $\|s_k\| = \Delta$*

  $r_k = r_{k-1} - \alpha_k q_{k-1}$

  $\rho_k = \rho_{k-1} - \alpha_k \zeta_{k-1}$ $\rho_k = \|r_k\|^2$

  $\nu_k = \sqrt{\rho_k}$ $\nu_k = \|r_k\|$

  $u_k = Hr_k$

  $\zeta_k = r_k^T u_k$ $\zeta_k = r_k^T H r_k$

  $\beta_k = \zeta_k / \zeta_{k-1}$

  $p_k = r_k + \beta_k p_{k-1}$

  $q_k = u_k + \beta_k q_{k-1}$ $q_k = Hp_k$

  $\mu_k = \rho_k + \beta_k(\mu_{k-1} - \alpha_k \delta_{k-1})$ $\mu_k = p_k^T r_k$

  $\delta_k = \zeta_k + \beta_k^2 \delta_{k-1}$ $\delta_k = p_k^T H p_k$

return s_k

of (2) lies inside the trust region and is found before any nonpositive curvature is discovered, Algorithm 5 is equivalent to Algorithm 1.

Whether or not we discover nonpositive curvature along p_{k-1} or r_{k-1}, we must monitor the sign of α_k and select a final iterate inside the trust region. Because r_{k-1} is always a descent direction, the direction of best decrease among p_{k-1} and r_{k-1} is followed to either the minimum of the quadratic along that direction, or the boundary of the trust region. Example 2.1 shows that for indefinite H, p_{k-1} may not be a descent direction. Thus, we define $\alpha_{p+} > 0$ and $\alpha_{p-} < 0$ as the step lengths to the boundary in the direction p_{k-1}, and $\alpha_r > 0$ as the step length in the direction r_{k-1}. Note that α_{p+}, α_{p-}, and α_r can be computed at a moderate cost because ρ_{k-1} contains the value of $\|r_{k-1}\|^2$ and $\|p_{k-1}\|^2$ can be recurred as in Algorithm 2.

In Algorithm 6, we use the notation "$\alpha \leftarrow$ *value*1 **if** *condition* **else** *value*2" to mean that if *condition* evaluates to true, α receives *value*1, and receives *value*2 otherwise. Let α_{p*} and α_{r*} be the optimal step lengths along p_{k-1} and r_{k-1}, respectively. At iteration k of Algorithm 5, we define

$$\begin{aligned} \xi_k &:= m(s_{k-1} + \alpha_{p*} p_{k-1}) - m(s_{k-1} + \alpha_{r*} r_{k-1}) \\ &= -\alpha_{p*} p_{k-1}^T r_{k-1} + \alpha_{r*} \|r_{k-1}\|^2 + \tfrac{1}{2}(\alpha_{p*}^2 p_{k-1}^T H p_{k-1} - \alpha_{r*}^2 r_{k-1}^T H r_{k-1}) \\ &= -\alpha_{p*} \mu_{k-1} + \alpha_{r*} \rho_{k-1} + \tfrac{1}{2}(\alpha_{p*}^2 \delta_{k-1} - \alpha_{r*}^2 \zeta_{k-1}), \end{aligned} \tag{31}$$

which evaluates the decrease along p_{k-1} compared to that along r_{k-1}.

Below, line numbers refer to Algorithm 6. We distinguish several cases.

Case I. $p_{k-1}^T H p_{k-1} = 0$ (line 3), which covers the $r_{k-1}^T H r_{k-1} = 0$ case by (33): $m(s_{k-1} + \alpha_k p_{k-1}) = m(s_{k-1}) - \alpha_k r_{k-1}^T p_{k-1}$ is linear along p_{k-1}. If H is not positive definite, (14) may not hold. We consider two subcases.

I.1. $p_{k-1}^T r_{k-1} = 0$ (line 5): m is constant along p_{k-1}. We reset $p_{k-1} \leftarrow r_{k-1}$, which is a descent direction. For any step length α, $m(s_{k-1} + \alpha r_{k-1}) = m(s_{k-1}) - \alpha\|r_{k-1}\|^2 + \frac{1}{2}\alpha^2 r_{k-1}^T H r_{k-1} = m(s_{k-1}) - \alpha\rho_{k-1} + \frac{1}{2}\alpha^2\zeta_{k-1}$ is stationary when $\alpha\zeta_{k-1} = \rho_{k-1}$. Thus, we set α_k on line 8 as follows.
  (a) $\zeta_{k-1} > 0$: m decreases from $\alpha = 0$ to $\alpha_b = \rho_{k-1}/\zeta_{k-1}$. To remain inside the trust region, we choose $\alpha_k = \min(\alpha_r, \alpha_b)$.
  (b) $\zeta_{k-1} \le 0$: m is unbounded below along r_{k-1}. We set $\alpha_k = \alpha_r$.

I.2. $p_{k-1}^T r_{k-1} \neq 0$ (line 10): m is linear but not constant along p_{k-1} and α_k must be such that $\alpha_k p_{k-1}$ is a descent direction. As r_{k-1} is a descent direction, we consider the decrease in m along both directions and choose the best. Call α_{p*} the optimal step length along p_{k-1}, which is α_{p+} if $\mu_{k-1} = p_{k-1}^T r_{k-1} > 0$, and α_{p-} otherwise, because m is linear along p_{k-1}. Call α_{r*} the optimal step length along r_{k-1}, i.e., $\min(\alpha_b, \alpha_r)$ if $\zeta_{k-1} > 0$ and α_r otherwise. According to (31),

$$\xi_k = -\alpha_{p*}\mu_{k-1} + \alpha_{r*}\rho_{k-1} - \tfrac{1}{2}\alpha_{r*}^2\zeta_{k-1},$$

which is computed at line 14. If $\xi_k > 0$, the best decrease occurs along r_{k-1}, which is then chosen as the search direction with $\alpha_k = \alpha_{r*}$. Otherwise, we select p_{k-1} with $\alpha_k = \alpha_{p*}$.

Numerically, we relax the condition $\delta_{i-1} = 0$ to $|\delta_{i-1}| \le \epsilon\|p_{i-1}\|\|q_{i-1}\|$ and $\mu_{i-1} = 0$ to $|\mu_{i-1}| \le \epsilon\|p_{i-1}\|\nu_{i-1}$ for a user-defined tolerance $\epsilon > 0$.

Case II. $p_k^T H p_k > 0$ and $r_k^T H r_k > 0$ (line 17): m is convex along p_k and r_k, so Corollary 2 applies and standard CR is applied within the trust region.

Case III. In all other situations, $p_{k-1}^T H p_{k-1} > 0$ and $r_{k-1}^T H r_{k-1} < 0$ (line 22), $p_{k-1}^T H p_{k-1} < 0$ and $r_{k-1}^T H r_{k-1} > 0$ (line 28), or $p_{k-1}^T H p_{k-1} < 0$ and $r_{k-1}^T H r_{k-1} < 0$ (line 35): negative curvature is detected. We compute ξ_k to select the direction of best decrease between r_{k-1} and p_{k-1}, and follow it to the minimum of m or the trust-region boundary.

4.3. Main properties.

THEOREM 12. *In Algorithm* 5, *the properties of Theorem* 1 *continue to hold for positive definite* H.

Proof. As long as $\delta_{i-1} > 0$ and $\mu_{i-1} > 0$, $i = 0, 1, \ldots, k$, Algorithm 5 coincides with Algorithm 1. For H positive definite, this is ensured at all iterations. ∎

COROLLARY 2. *If* $\delta_{i-1} > 0$ *and* $\mu_{i-1} > 0$ *for* $i = 0, 1, \ldots, k$, *then the properties of Theorem* 1 *continue to hold at those iterations in Algorithm* 5.

THEOREM 13. *Whether* H *is positive definite or not, Algorithm* 5 *continues to satisfy* (15)–(18), (22), *and* (23) *for as long as* `terminate` *is* `false`. *In addition, for* $k \ge 0$,

$$\xi_k = m(s_{k-1} + \alpha_k p_{k-1}) - m(s_{k-1} + \alpha_r r_{k-1}), \tag{32}$$

$$\zeta_k = 0 \implies \delta_k = 0. \tag{33}$$

Proof. The proof of (15)–(18) is as in Theorem 2. That of (22) and (23) is as in Theorem 4.

The identity (32) follows from (31) and our choice for α_k and α_r in Algorithm 5. If $\zeta_k = 0$, lines 12 and 13 of Algorithm 5 yield $\beta_k = 0$ and $p_k = r_k$, and $\delta_k = 0$. ∎

Note that (33) also holds for all previous variants of CR.

4.4. Convergence analysis. Here, we establish that a step computed by Algorithm 5 satisfies the sufficient-decrease condition (29), and therefore convergence of Algorithm 4 to a stationary point is guaranteed under standard assumptions on (1).

Algorithm 6 Step computation for Algorithm 5.

Require: $\Delta > 0$, ζ_{k-1}, δ_{k-1}, ρ_{k-1}, μ_{k-1}, ν_{k-1}, s_{k-1}, r_{k-1}, p_{k-1}, q_{k-1}, $\epsilon > 0$

`terminate = false`
compute $\alpha_{p+} > 0$, $\alpha_{p-} < 0$ such that $\|s_{k-1} + \alpha p_{k-1}\| = \Delta$, $\alpha \in \{\alpha_{p+}, \alpha_{p-}\}$
if $|\delta_{k-1}| \le \epsilon \|p_{k-1}\| \|q_{k-1}\|$ **then** *$p_{k-1}^T H p_{k-1} \approx 0$*
    `terminate = true` *m is nonconvex*
    if $|\mu_{k-1}| \le \epsilon \|p_{k-1}\| \nu_{k-1}$ **then** *$p_{k-1}^T r_{k-1} \approx 0$*
        $p_{k-1} \leftarrow r_{k-1}$
        compute $\alpha_r > 0$ such that $\|s_{k-1} + \alpha_r r_{k-1}\| = \Delta$
        $\alpha_k \leftarrow \min(\alpha_r, \rho_{k-1}/\zeta_{k-1})$ **if** $\zeta_{k-1} > 0$ **else** α_r
    else
        compute $\alpha_r > 0$ such that $\|s_{k-1} + \alpha_r r_{k-1}\| = \Delta$
        if $\zeta_{k-1} > 0$ **then**
            $\alpha_r \leftarrow \min(\alpha_r, \rho_{k-1}/\zeta_{k-1})$
        $\alpha_k \leftarrow \alpha_{p+}$ **if** $\mu_{k-1} > 0$ **else** α_{p-}
        $\xi_k = -\alpha_k \mu_{k-1} + \alpha_r \rho_{k-1} - \frac{1}{2}\alpha_r^2 \zeta_{k-1}$
        if $\xi_k > 0$ **then** *$m(s_{k-1} + \alpha_r r_{k-1}) < m(s_{k-1} + \alpha_k p_{k-1})$*
            $p_{k-1} \leftarrow r_{k-1}$, $\alpha_k \leftarrow \alpha_r$
else if $\delta_{k-1} > 0$ and $\zeta_{k-1} > 0$ **then**
    $\alpha_k = \zeta_{k-1}/\|q_{k-1}\|^2$
    if $\alpha_k \ge \alpha_{p+}$ **then**
        `terminate = true` *s_k is on the boundary of the trust region*
        $\alpha_k \leftarrow \alpha_{p+}$
else if $\delta_{k-1} > 0$ and $\zeta_{k-1} < 0$ **then**
    `terminate = true`
    $\alpha_k \leftarrow \min(\alpha_{p+}, \mu_{k-1}/\delta_{k-1})$ **if** $\mu_{k-1} > 0$ **else** $\max(\alpha_{p-}, \mu_{k-1}/\delta_{k-1})$
    compute $\alpha_r > 0$ such that $\|s_{k-1} + \alpha_r r_{k-1}\| = \Delta$
    $\xi_k = -\alpha_k \mu_{k-1} + \alpha_r \rho_{k-1} + \frac{1}{2}(\alpha_k^2 \delta_{k-1} - \alpha_r^2 \zeta_{k-1})$
    if $\xi_k > 0$ **then** $p_{k-1} \leftarrow r_{k-1}$, $\alpha_k \leftarrow \alpha_r$
else if $\delta_{k-1} < 0$ and $\zeta_{k-1} > 0$ **then**
    `terminate = true`
    $\alpha_k \leftarrow \alpha_{p+}$ **if** $\mu_{k-1} > 0$ **else** α_{p-}
    compute $\alpha_r > 0$ such that $\|s_{k-1} + \alpha_r r_{k-1}\| = \Delta$
    $\alpha_r \leftarrow \min(\alpha_r, \rho_{k-1}/\zeta_{k-1})$
    $\xi_k = -\alpha_k \mu_{k-1} + \alpha_r \rho_{k-1} + \frac{1}{2}(\alpha_k^2 \delta_{k-1} - \alpha_r^2 \zeta_{k-1})$
    if $\xi_k > 0$ **then** $p_{k-1} \leftarrow r_{k-1}$, $\alpha_k \leftarrow \alpha_r$
else if $\delta_{k-1} < 0$ and $\zeta_{k-1} < 0$ **then**
    `terminate = true`
    compute $\alpha_r > 0$ such that $\|s_{k-1} + \alpha_r r_{k-1}\| = \Delta$
    $\alpha_k \leftarrow \alpha_{p+}$ **if** $\mu_{k-1} > 0$ **else** α_{p-}
    $\xi_k = -\alpha_k \mu_{k-1} + \alpha_r \rho_{k-1} + \frac{1}{2}(\alpha_k^2 \delta_{k-1} - \alpha_r^2 \zeta_{k-1})$
    if $\xi_k > 0$ **then** $p_{k-1} \leftarrow r_{k-1}$, $\alpha_k \leftarrow \alpha_r$
return `terminate`, α_k, p_{k-1}

Both CG and CR begin by performing a search along the steepest-descent direction $-g$. The CG step length $\alpha_C > 0$ is determined by minimizing

$$m(-\alpha g) = -\alpha\|g\|^2 + \tfrac{1}{2}\alpha^2 g^T H g.$$

The CR step length $\alpha_M > 0$ is determined by minimizing

$$R(-\alpha g) = \tfrac{1}{2}\|g - \alpha H g\|^2.$$

Both minimizations must take the trust region into account. By definition, the first CG iterate is precisely the Cauchy point. If $g^T H g \le 0$, both methods step to the boundary and thus achieve the same decrease. If $g^T H g > 0$, the unconstrained minimizers are

$$\alpha_C = \frac{\|g\|^2}{g^T H g} \qquad \text{and} \qquad \alpha_M = \frac{g^T H g}{g^T H^2 g}.$$

In addition,

$$m_C := m(-\alpha_C g) = -\tfrac{1}{2}\frac{\|g\|^4}{g^T H g}, \tag{34}$$

$$m_M := m(-\alpha_M g) = \frac{g^T H g}{g^T H^2 g}\left(\tfrac{1}{2}\frac{(g^T H g)^2}{g^T H^2 g} - \|g\|^2\right). \tag{35}$$

Theorem 11 implies $m_M \le 0$, which yields $\tfrac{1}{2}\alpha_M \le \alpha_C$. Lemma 3 is more precise.

LEMMA 3. *Let g be such that $g^T H g > 0$. Then $\alpha_M \le \alpha_C$.*

Proof. Let $y = Hg$. The Cauchy–Schwartz inequality states that $(g^T y)^2 \le \|g\|^2\,\|y\|^2$, i.e., $\|Hg\|^2 \ge (g^T H g)^2/\|g\|^2$. Thus

$$\alpha_M = \frac{g^T H g}{\|Hg\|^2} \le \frac{(g^T H g)\,\|g\|^2}{(g^T H g)^2} = \alpha_C. \qquad \square$$

Assume first that the CR minimizer along $-g$ lies inside the trust region. Lemma 3 implies

$$\frac{(g^T H g)^2}{g^T H^2 g} \le \|g\|^2,$$

so that (35) yields $m_M \le -\tfrac{1}{2}\alpha_M\|g\|^2$. But

$$\alpha_M \ge \frac{g^T H g}{(1 + \|H\|)\, g^T H g} = \frac{1}{1 + \|H\|},$$

and therefore

$$m_M \le -\tfrac{1}{2}\frac{1}{1 + \|H\|}\,\|g\|^2.$$

Suppose now that the CG minimizer lies on the boundary or outside the trust region. In this case, we reset $\alpha_C = \Delta/\|g\|$ and obtain

$$m_C := m(-\alpha_C g) = \Delta\left(\tfrac{1}{2}\Delta\frac{g^T H g}{\|g\|^2} - \|g\|\right).$$

If the CR minimizer also lies on the boundary or outside the trust region, $m_M = m_C$. If on the other hand $\alpha_M < \Delta/\|g\|$, (35) yields

$$m_M < \frac{\Delta}{\|g\|}\left(\frac{1}{2}\frac{\Delta}{\|g\|}\, g^T H g - \|g\|^2\right) = m_C.$$

By Theorem 11, in the event that $g^T H g > 0$ and $\alpha_M < \Delta/\|g\|$, subsequent CR iterations further reduce the value of $m(s)$. Should CR encounter a direction of negative curvature, Algorithm 5 guarantees that no increase in the value of $m(s)$ can result, but that further decrease may occur. Thus in all cases, CR improves upon its first iterate and therefore yields an approximate solution of (3) that satisfies the sufficient-decrease condition (29).

4.5. Numerical results. We use the same benchmark problems as in subsection 3.4, with the addition of *indefm*, which is nonconvex, and *sscosine*, whose convexity is unknown.

Algorithms 4 to 6 are implemented in Julia v0.7. Truncated CG is implemented as described by Steihaug (1983).

Algorithm 4 uses $\epsilon_a = \epsilon_r = 10^{-6}$, $\Delta_0 = 10$, $\eta_1 = 10^{-4}$, $\eta_2 = 0.99$, and $\gamma_1 = \gamma_2 = \frac{1}{3}$. Line 17 is changed to $\Delta_{j+1} = 3\Delta_j$, and line 19 becomes $\Delta_{j+1} = \Delta_j$. We impose a maximum of 10,000 iterations.

Truncated CG and CR have a maximum number of iterations equal to the number of variables in the problem. Algorithms 5 and 6 set ϵ to machine precision for the detection of negative curvature. Finally, at iteration j of the trust-region algorithm, the tolerance for the inner iterations is $\tau_a + \tau_r\|g_j\|$ with $\tau_a = 0$ and $\tau_r = \min(0.1, \sqrt{\|g_j\|})$. This choice of τ_r is again inspired by Theorem 9, because when x_k approaches an isolated minimizer, we expect the trust-region constraint to be inactive.

Both variants perform equivalently on convex problems, as illustrated by the profiles of Figure 4. In particular, both variants solve all problems.

On nonconvex problems, CR and CG fail to solve two and three problems, respectively. Figure 5 shows that CR performs better for all measures, which is a surprise given the CG optimality property of minimizing the quadratic objective as long as it is convex. The advantage is especially apparent in terms of Hessian-vector products.

Figure 6 shows that the same trend persists on the entire set of 100 problems. CR and CG fail on three and five problems, respectively.

5. Extension to nonlinear least squares. Suppose the objective of (1) is $f(x) = \frac{1}{2}\|F(x)\|^2$, where $F(x) = (f_1(x), \dots, f_l(x))$. The classical approach of Levenberg (1944) and Marquardt (1963) may be implemented by replacing $m(s)$ in (3) with the Gauss–Newton model and solving (1) using Algorithm 4. Thus, at each iteration the subproblem to solve is

$$\text{(36)} \quad \underset{s\in\mathbb{R}^n}{\text{minimize}}\ m^{\text{GN}}(s) \quad \text{subject to } \|s\| \le \Delta, \qquad m^{\text{GN}}(s) := \tfrac{1}{2}\|J(x)s + F(x)\|^2,$$

where $J(x)$ is the Jacobian of F at x. Both CG and CR can be used to solve (36). Products with $J(x)^T J(x)$ may be decoupled, and this gives rise to variants named CGLS[4] (section 10 of Hestenes and Stiefel (1952)) and CRLS (Fong (2011)) specific to linear least squares.[5]

[4]The name CGLS appears to have been coined by Paige and Saunders (1982).

[5]The earliest reference mentioning CRLS, though not under that name, that we are aware of is Björck (1979).

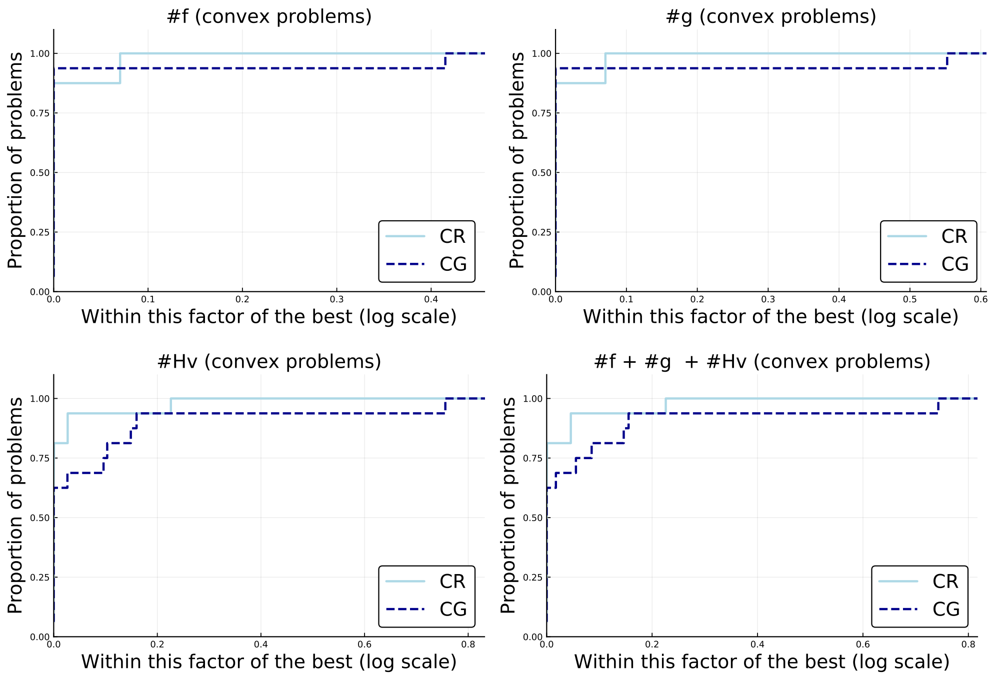

Fig. 4. *Performance of trust-region CR and CG on* 16 *convex problems in terms of evaluations of f, g, and products with H.*

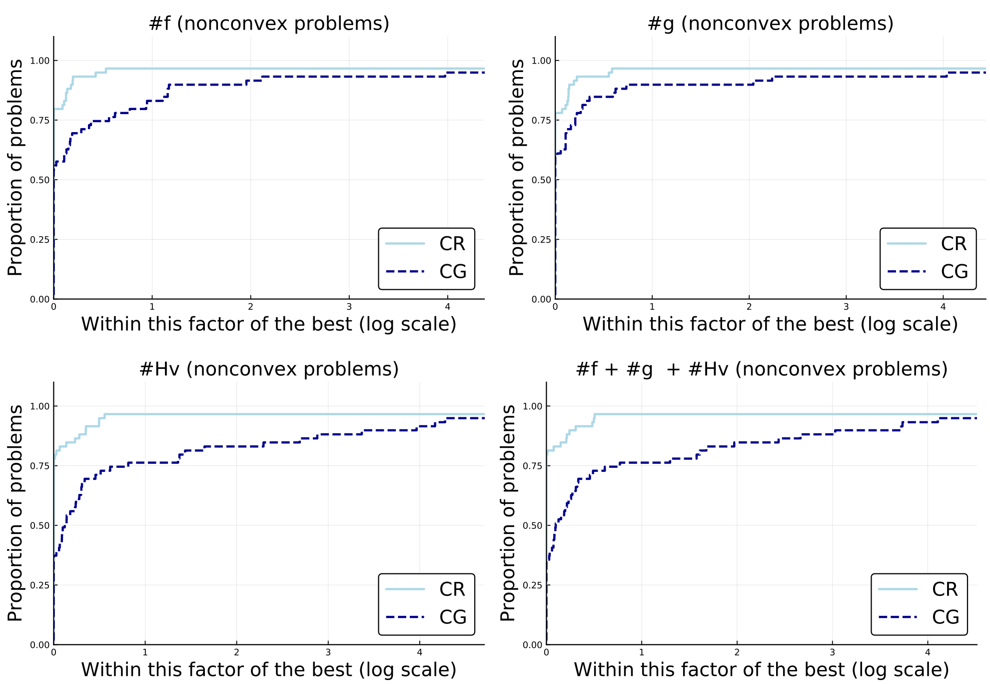

Fig. 5. *Performance of trust-region CR and CG on* 59 *nonconvex problems in terms of evaluations of f, g, and products with H.*

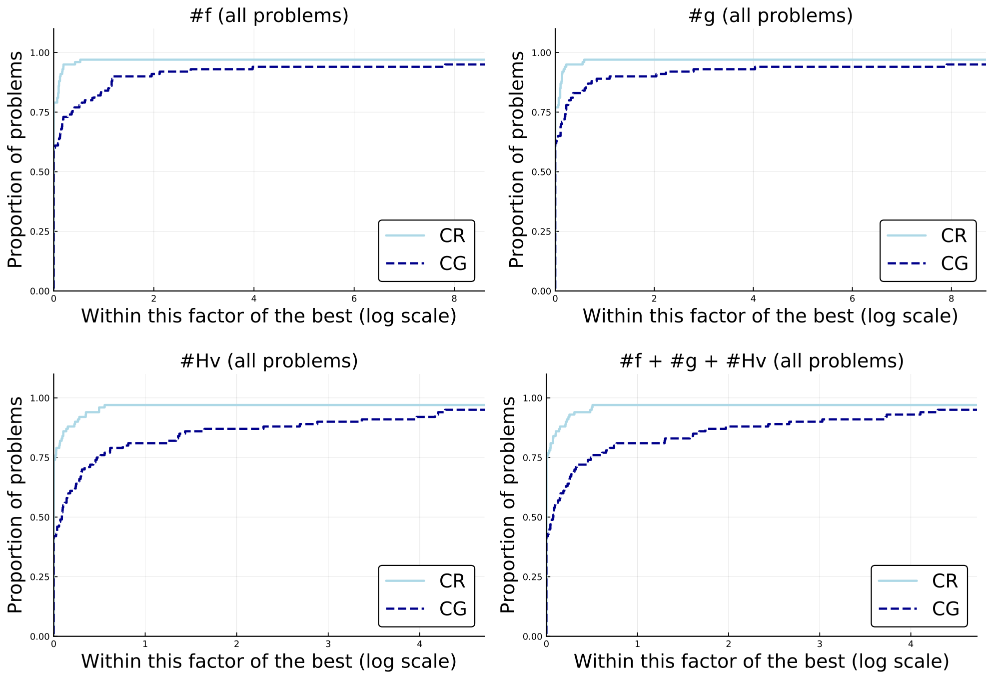

Fig. 6. *Performance of trust-region CR and CG on* 100 *problems in terms of evaluations of* f*,* g*, and products with* H*.*

5.1. CRLS for trust region. For simplicity we use J and F to refer respectively to $J(x)$ and $F(x)$ in the description of the algorithm.

Algorithm 7 is a modification of CRLS adapted to a trust-region context, with radius Δ as input parameter. Note that $\nabla f(x) = J^T F = -\tilde{r}_0$.

At any iteration k, let $\alpha_p > 0$ be the step length to the trust-region boundary in the direction p_{k-1}. Because $J^T J$ is positive definite or semidefinite, the curvature of m^{GN} can only be zero or positive. If $p_{k-1}^T J^T J p_{k-1} = 0$, then $J p_{k-1} = q_{k-1} = 0$, and m^{GN} is constant in the direction p_{k-1}. In that case, we select $-\nabla m^{\text{GN}}(s_{k-1}) = \tilde{r}_{k-1}$ as the new search direction. We compute the step length to the minimizer of m^{GN} in the direction $\tilde{r}_{k-1}$, and either step to the minimizer or stop at the trust-region boundary if the minimizer lies outside.

5.2. Numerical results. In exact arithmetic, CGLS and CRLS are equivalent to LSQR (Paige and Saunders (1982)) and LSMR (Fong and Saunders (2011)), respectively, which are based on the Golub and Kahan (1965) process and are numerically preferable. LSQR and LSMR require the same number of operator-vector products per iteration as CGLS and CRLS. Björck, Elfving, and Strakoš (1998) analyzed several versions of CGLS, notably using recurred residuals, and compared their numerical stability. They concluded that for CGLS to be as stable as LSQR and achieve similar accuracy, the product $J^T r$ must be computed explicitly rather than being recurred, and that results in an extra operator-vector product per iteration. Björck and Saunders (2017) performed similar comparisons between CRLS and LSMR and concluded that in its default version, CRLS is inferior to LSMR and is unable to achieve comparable accuracy. They devised a version of CRLS that requires an extra operator-vector product per iteration that is competitive with LSMR. Kloek (2012) performed similar

Algorithm 7 CRLS for (36).

Require: J, F, $\Delta > 0$, $\tau_a > 0$, $\tau_r > 0$, $\epsilon > 0$

Initialize: $k = 0$, $s_0 = 0$, $r_0 = -F$, $\tilde{r}_0 = J^T r_0$, $w_0 = J\tilde{r}_0$, $\zeta_0 = w^T w$, $p_0 = \tilde{r}_0$, $q_0 = w_0$

while $\|\tilde{r}_{k-1}\| > \tau_a + \tau_r \|\tilde{r}_0\|$ **do**

    $k \leftarrow k + 1$

    $v_k = J^T q_{k-1}$

    $\alpha_k = \zeta_{k-1} / \|v\|^2$

    compute $\alpha_p > 0$ such that $\|s_{k-1} + \alpha_p p_{k-1}\| = \Delta$

    if $\|q_{k-1}\|^2 \le \epsilon \|p_{k-1}\| \|v_k\|$ **then** *(near) zero curvature detected*

        $p_{k-1} \leftarrow \tilde{r}_{k-1}$

        $\alpha_k \leftarrow \min(\alpha_p, \|\tilde{r}_{k-1}\|^2 / \zeta_{k-1})$

        $s_k = s_{k-1} + \alpha_k p_{k-1}$

        return s_k

    else

        if $\alpha_k \ge \alpha_p$ **then**

            $\alpha_k \leftarrow \alpha_p$

            $s_k = s_{k-1} + \alpha_k p_{k-1}$

            return s_k

    $s_k = s_{k-1} + \alpha_k p_{k-1}$

    $r_k = r_{k-1} - \alpha_k q_{k-1}$

    $\tilde{r}_k = J^T r_k$

    $w_k = J\tilde{r}_k$

    $\zeta_k = w_k^T w_k$

    $\beta_k = \zeta_k / \zeta_{k-1}$

    $p_k = \tilde{r}_k + \beta_k p_{k-1}$

    $q_k = w_k + \beta_k q_{k-1}$

return s_k

experiments and made similar observations. For the above reasons, in our experiments, we use LSQR and LSMR as implemented in the package Krylov.jl,[6] with appropriate changes to accommodate a trust-region constraint.

We use nonlinear least-squares problems implemented in Julia from the NLSProblems.jl[7] collection, together with those from CUTEst.jl that are available in the form of feasibility problems where the equality constraints play the role of the residual. We eliminate problems with fewer than 10 variables. We exclude *ba-l21*, *ba-l49*, and *ba-l52* as they require more than 1.5 hours to be solved. We further eliminate *mgh11* because its objective is not $\mathcal{C}^2$. In total, we run 97 problems.

The trust-region parameters are as in subsection 4.5. The maximum number of LSQR and LSMR iterations is $n + l$, $\tau_a = 0$, and $\tau_r = \min(0.1, \sqrt{\|g_j\|})$.

Figure 7 gives the profiles using $\#F$, $\#Ju + \#J^T v$, and $\#F + \#Ju + \#J^T v$. The profiles show that LSQR and LSMR perform equivalently, with a slight advantage for LSMR in terms of residual evaluations. Both methods fail on *ba-l73* and *oscipane*.

[6] See github.com/JuliaSmoothOptimizers/Krylov.jl.

[7] See github.com/JuliaSmoothOptimizers/NLSProblems.jl.

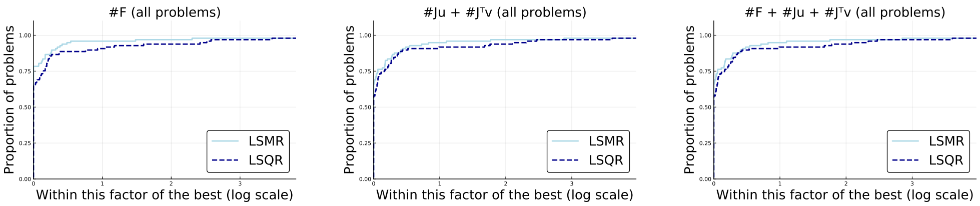

FIG. 7. *Performance of trust-region LSQR and LSMR on* 97 *nonlinear least-squares problems in terms of* $\#F$*,* $\#Ju + \#J^Tv$*, and the sum of both.*

6. Discussion. Most implementations of linesearch inexact Newton and trust-region methods with iterative step computation employ the conjugate gradient method. While CG is conceptually the correct method in a trust-region context because of its minimization property of the quadratic model, the CG residual is typically erratic and it is difficult to justify why it should be the method of choice in a linesearch inexact Newton context. When applied to a convex quadratic model, CR shares similarities with CG. Although CR does not minimize the quadratic model at each iteration, its value decreases monotonically. By construction, CR also produces a monotonic residual, which *is* minimized at each iteration. Our adaptations of CR to handle nonpositive curvature in both linesearch and trust-region contexts show that CR performs as well as or slightly better than CG, particularly in terms of Hessian-vector products. Our extension to nonlinear least-squares problems via the LSMR implementation of CRLS shows that it behaves comparably to the LSQR implementation of CGLS in a trust-region context.

It is counter-intuitive that CR performs comparably to CG in a linesearch context, but outperforms it in a trust-region context. One possible explanation is that Algorithm 5 differs more from Algorithm 1 than the truncated CG of Steihaug (1983) differs from plain CG. Indeed, when negative curvature is detected, Algorithm 6 sometimes explores a two-dimensional subspace to improve the step, whereas truncated CG simply follows the negative curvature direction. In truncated CG, it is also possible to compute the model decrease along the residual and select the step that yields the best decrease. However, doing so costs an extra operator-vector product and an extra dot product to compute $r_k^THr_k$. Future research may reveal that CR outperforms CG in linesearch contexts on specific applications, or on different test problems. In sum, we believe that CR is a viable alternative to CG in optimization.

Our implementations are designed to save computations and recur a number of quantities such as $p_k^THp_k$ and $r_k^THr_k$. It is conceivable that such recurrence formulae are subject to accumulation of rounding errors, especially on ill-conditioned problems, though we have not observed damaging results in our experiments. A finite-precision arithmetic analysis such as that of Björck, Elfving, and Strakoš (1998) would shed light on the matter.

MINRES (Paige and Saunders (1975)) should be the preferred implementation of CR, and it generalizes CR to indefinite systems. We have not used MINRES in the present research because its implementation is substantially more involved and certain quantities of interest, such as $p_k^THp_k$, are not readily available. However, variants of MINRES adapted to linesearch and trust-region contexts would be highly relevant.

We have deliberately left questions of preconditioning aside as they are typically application dependent. A study of the behavior of CR with generic—e.g., diagonal or incomplete Cholesky—preconditioners is left for future work.

The satisfactory performance of CR illustrated in this research raises the question of whether it would also be a worthwhile subproblem solver in constrained optimization, e.g., in projected direction methods for bound-constrained problems such as that of Lin and Moré (1998).

Finally, in trust-region methods, Yuan (2000, Theorem 2) establishes that the decrease in the quadratic model achieved by truncated CG is at least half of that obtained at a global solution of the trust-region subproblem. This is an important result and it is relevant to determine whether a similar result holds for CR.

7. Appendix. In this section, we provide detailed results for each variant of CR and CG. Problems marked with "$\star$" reached the maximum number of iterations. Problem *parkch* is marked with "+" in Table 1 to indicate that NaNs were generated during the iterations, as described in subsection 3.4.

7.1. Detailed results for the linesearch method. Detailed results for each problem are given in Table 1 and Table 2.

Table 1: Solution of 98 nonlinear problems with linesearch CR.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
arglina	200	2.000e+02	1.000e+03	5.8e−13	5.7e+01	2	2	2	1
arglinb	200	9.963e+01	8.651e+15	1.2e−01	1.4e+15	2	2	2	1
arglinc	200	1.011e+02	8.353e+15	7.8e−02	1.4e+15	2	2	2	1
bdqrtic	5000	2.001e+04	1.129e+06	9.2e−01	1.5e+06	10	10	40	9
box	10000	−1.865e+03	0.000e+00	2.7e−06	5.0e+01	12	5	14	4
boxpower	20000	4.716e−02	1.764e+05	2.1e−02	1.7e+05	8	7	18	6
broydn7d	5000	1.854e+03	1.760e+04	7.6e−04	1.1e+03	1977	1782	7610	1781
brybnd	5000	2.050e−06	1.249e+05	6.5e−03	7.8e+03	9	9	92	8
chainwoo	4000	2.388e+03	1.445e+07	1.7e−01	4.2e+05	5291	3454	85956	3453
chnrosnb_mod	100	1.115e−05	1.764e+04	6.1e−03	6.4e+03	200	112	1195	111
chnrsnbm	50	2.275e−09	8.633e+03	2.7e−04	4.4e+03	126	70	1311	69
clplatea	5041	−1.259e−02	0.000e+00	1.1e−06	1.0e−01	8	7	646	6
clplateb	5041	−5.095e−03	0.000e+00	3.6e−07	1.2e−02	4	4	411	3
clplatec	5041	−5.021e−03	0.000e+00	7.3e−07	9.9e−02	5	5	10961	4
cosine	10000	−9.999e+03	8.775e+03	8.7e−07	7.2e+01	8	7	19	6
cragglvy	5000	1.688e+03	2.749e+06	1.1e−01	2.8e+05	13	13	92	12
curly10	10000	−1.003e+06	−6.306e−01	1.3e−04	1.3e+02	21	19	86127	18
curly20	10000	−1.003e+06	−1.344e+00	2.4e−05	3.0e+02	25	21	99683	20
curly30	10000	−1.003e+06	−2.190e+00	7.6e−05	5.1e+02	28	22	106049	21
dixmaana	3000	1.000e+00	2.850e+04	2.6e−04	1.2e+03	10	8	20	7
dixmaanb	3000	1.000e+00	4.724e+04	2.1e−04	2.0e+03	8	8	14	7
dixmaanc	3000	1.000e+00	8.248e+04	2.8e−04	3.7e+03	9	9	16	8
dixmaand	3000	1.000e+00	1.586e+05	2.2e−03	7.6e+03	10	10	18	9
dixmaane	3000	1.000e+00	2.850e+04	2.6e−04	1.2e+03	10	8	20	7
dixmaanf	3000	1.000e+00	4.104e+04	2.7e−04	1.9e+03	12	12	252	11
dixmaang	3000	1.000e+00	7.607e+04	1.3e−03	3.6e+03	12	12	152	11
dixmaanh	3000	1.001e+00	1.517e+05	3.6e−03	7.4e+03	12	12	64	11
dixmaani	3000	1.000e+00	2.002e+04	3.3e−05	1.0e+03	11	11	1260	10
dixmaanj	3000	1.000e+00	3.900e+04	4.7e−04	1.8e+03	13	13	166	12
dixmaank	3000	1.000e+00	7.400e+04	9.7e−04	3.6e+03	13	13	122	12
dixmaanl	3000	1.000e+00	1.496e+05	2.4e−03	7.4e+03	13	13	86	12
dixmaanm	3000	1.001e+00	9.358e+03	3.1e−04	4.4e+02	9	9	524	8

Continued on next page

Table 1 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
dixmaann	3000	1.000e+00	2.018e+04	4.9e−04	1.0e+03	13	13	290	12
dixmaano	3000	1.001e+00	3.635e+04	1.1e−03	2.0e+03	13	13	198	12
dixmaanp	3000	1.002e+00	7.128e+04	2.5e−03	4.0e+03	13	13	136	12
dixon3dq*	10000	9.661e−04	8.000e+00	1.8e−05	5.7e+00	10001	10001	42496	10000
dqdrtic	5000	3.143e−05	9.041e+06	1.1e−02	8.5e+04	5	5	16	4
dqrtic	5000	4.137e+09	6.241e+17	8.1e+06	1.3e+13	13	13	52	12
edensch	2000	1.200e+04	7.358e+06	2.8e−02	1.0e+05	14	14	48	13
eg2	5000	5.549e+03	2.949e+05	1.7e−03	8.8e+03	9	9	34	8
engval1	5000	5.549e+03	2.949e+05	1.7e−03	8.8e+03	9	9	34	8
errinros	50	4.022e+01	1.102e+05	4.1e−02	1.2e+05	32	27	341	26
errinros_mod	100	7.862e+01	3.140e+05	1.9e−01	1.9e+05	22	18	93	17
errinrsm	50	3.873e+01	1.529e+05	1.1e−01	1.3e+05	25	20	209	19
extrosnb	1000	8.403e−03	3.996e+05	3.8e−02	3.8e+04	48	30	216	29
fletbv3m	5000	−2.326e+05	1.982e+02	3.2e−05	4.4e+01	293	292	528	291
fletcbv2*	5000	−5.003e−01	−5.003e−01	3.7e−06	4.4e−06	10001	10001	32266	10000
fletcbv3*	5000	−2.286e+08	1.982e+02	5.0e+01	4.4e+01	10001	10001	13158	10000
fletcbv3_mod*	100	−8.684e−02	−1.879e−02	2.8e−03	1.6e−03	10001	10001	10001	10000
fletchbv*	5000	−1.916e+19	−2.302e+11	4.5e+09	2.8e+09	15817	10001	29997	10000
fletchcr	1000	7.199e−11	9.990e+02	2.2e−05	6.3e+01	2726	1634	37090	1633
fminsrf2*	5625	1.000e+00	2.846e+01	2.5e−06	3.3e−01	10133	10001	27849	10000
fminsurf	5625	1.000e+00	2.859e+01	2.8e−07	3.3e−01	227	38	16029	37
freuroth	5000	6.082e+05	5.049e+06	4.4e−03	5.5e+04	28	10	43	9
genhumps	5000	1.735e−07	1.281e+08	3.0e−04	6.0e+03	7947	7530	28065	7529
genrose	500	1.000e+00	1.870e+03	2.2e−06	3.0e+02	915	457	7380	456
genrose_nash	100	1.000e+00	4.041e+02	8.7e−05	1.3e+02	196	110	790	109
hilbertb	10	2.216e−14	5.102e+02	6.8e−07	1.1e+02	6	6	12	5
indef*	5000	−2.499e+07	4.603e+03	7.1e+01	8.0e+01	10001	10001	20062	10000
liarwhd	5000	1.671e−04	2.925e+06	2.6e−02	4.8e+05	13	13	36	12
mancino	100	1.059e−01	1.103e+12	9.1e+02	2.9e+09	6	6	12	5
modbeale	20000	1.695e−02	1.264e+07	2.4e−01	3.1e+05	14	13	150	12
ncb20	5010	−1.463e+03	1.000e+04	2.6e−04	2.8e+02	238	218	1561	217
ncb20b	5000	7.351e+03	1.000e+04	7.2e−05	2.8e+02	14	10	203	9
noncvxu2	5000	1.182e+04	3.235e+11	2.6e+00	3.3e+06	3291	3287	13292	3286
noncvxun	5000	1.217e+04	3.335e+11	2.7e+00	3.6e+06	2343	2337	9987	2336
nondia	5000	2.570e−04	2.000e+06	1.3e−02	2.0e+06	3	3	4	2
nondquar	5000	4.830e−03	5.006e+03	1.9e−02	2.0e+04	17	15	112	14
parkch$^+$	15	*nan*	2.150e+03	*nan*	2.5e+04	24	14	97	13
penalty2	200	4.712e+13	4.712e+13	3.6e+00	1.6e+07	12	12	232	11
penalty3	200	1.016e−03	1.584e+09	7.9e−01	2.2e+06	49	20	115	19
powellsg	5000	5.539e−04	2.688e+05	4.8e−03	1.6e+04	14	14	90	13
power	10000	9.819e+07	2.501e+15	8.7e+07	1.2e+14	13	13	76	12
quartc	5000	4.137e+09	6.241e+17	8.1e+06	1.3e+13	13	13	52	12
schmvett	5000	−1.499e+04	−1.429e+04	3.2e−05	7.5e+01	7	7	80	6
scosine*	100	−4.591e+01	8.688e+01	9.1e+04	8.1e+02	10225	10001	25764	10000
scurly10	10000	3.636e+24	7.006e+31	8.3e+23	1.3e+30	13	13	116	12
scurly20	10000	4.752e+25	9.031e+32	1.1e+25	1.6e+31	13	13	108	12
scurly30	10000	1.914e+26	4.163e+33	4.7e+25	7.5e+31	13	13	104	12
sensors	100	−2.109e+03	−5.648e+01	6.2e−09	7.1e+01	33	18	49	17
sinquad	5000	−6.749e+06	6.561e−01	4.6e−03	5.1e+03	53	18	62	17

Continued on next page

Table 1 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
sparsine	5000	1.022e−01	5.173e+07	2.3e+00	3.0e+06	9	9	429	8
sparsqur	10000	9.566e−02	1.406e+07	7.3e−01	1.2e+06	13	13	70	12
spmsrtls	4999	1.341e−11	4.141e+03	1.0e−06	7.7e+01	16	15	554	14
srosenbr	5000	1.831e−08	4.850e+04	1.2e−04	1.1e+04	8	8	18	7
ssbrybnd	5000	6.737e−04	1.249e+05	3.0e−01	9.0e+05	18	17	16568	16
stratec	50	0.000e+00	0.000e+00	0.0e+00	0.0e+00	1	1	0	0
testquad	5000	4.741e+01	1.250e+09	1.5e+01	4.4e+07	7	7	748	6
tointgor	50	1.374e+03	5.074e+03	1.9e−05	6.0e+02	9	9	234	8
tointgss	5000	1.000e+01	4.499e+04	2.1e−04	4.2e+02	4	4	6	3
tointpsp	50	2.256e+02	1.828e+03	5.6e−05	1.1e+02	53	20	226	19
tointqor	50	1.175e+03	2.335e+03	1.9e−04	2.1e+02	6	6	66	5
tquartic	5000	2.749e−15	8.100e−01	1.5e−09	1.8e+00	43	33	71	32
tridia	5000	2.749e−15	8.100e−01	1.5e−09	1.8e+00	43	33	71	32
vardim	200	1.149e+08	3.257e+16	7.3e+09	1.6e+16	13	13	24	12
vareigvl	50	1.052e−13	1.126e+02	1.1e−06	4.2e+01	7	7	59	6
watson	12	1.602e−07	3.000e+01	4.8e−06	2.1e+02	14	14	112	13
woods	4000	7.877e+03	1.919e+07	1.1e−02	5.2e+05	10	10	38	9

Table 2: Solution of 98 nonlinear problems with linesearch CG.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
arglina	200	2.000e+02	1.000e+03	5.8e−13	5.7e+01	2	2	2	1
arglinb	200	9.963e+01	8.651e+15	1.2e−01	1.4e+15	2	2	2	1
arglinc	200	1.011e+02	8.353e+15	4.1e−01	1.4e+15	2	2	2	1
bdqrtic	5000	2.001e+04	1.129e+06	8.4e−01	1.5e+06	10	10	40	9
box	10000	−1.865e+03	0.000e+00	1.5e−06	5.0e+01	12	5	14	4
boxpower	20000	4.716e−02	1.764e+05	2.0e−02	1.7e+05	8	7	20	6
broydn7d	5000	1.875e+03	1.760e+04	4.2e−05	1.1e+03	1040	300	4326	299
brybnd	5000	1.510e−06	1.249e+05	7.2e−03	7.8e+03	8	8	78	7
chainwoo	4000	1.535e+02	1.445e+07	1.0e−01	4.2e+05	5324	1450	79310	1449
chnrosnb_mod	100	6.582e−08	1.764e+04	2.3e−03	6.4e+03	213	99	1192	98
chnrsnbm	50	4.301e−09	8.633e+03	2.1e−03	4.4e+03	100	56	1176	55
clplatea	5041	−1.259e−02	0.000e+00	3.0e−07	1.0e−01	8	7	773	6
clplateb	5041	−5.095e−03	0.000e+00	2.2e−07	1.2e−02	4	4	456	3
clplatec	5041	−5.021e−03	0.000e+00	2.7e−08	9.9e−02	5	5	9370	4
cosine	10000	−9.999e+03	8.775e+03	9.1e−07	7.2e+01	8	7	19	6
cragglvy	5000	1.688e+03	2.749e+06	1.0e−01	2.8e+05	13	13	102	12
curly10	10000	−1.003e+06	−6.306e−01	6.9e−05	1.3e+02	27	14	116337	13
curly20	10000	−1.003e+06	−1.344e+00	1.2e−04	3.0e+02	29	15	138593	14
curly30	10000	−1.003e+06	−2.190e+00	3.2e−04	5.1e+02	35	15	127354	14
dixmaana	3000	1.000e+00	2.850e+04	2.6e−04	1.2e+03	11	9	24	8
dixmaanb	3000	1.000e+00	4.724e+04	1.9e−04	2.0e+03	8	8	14	7
dixmaanc	3000	1.000e+00	8.248e+04	2.8e−04	3.7e+03	9	9	16	8
dixmaand	3000	1.000e+00	1.586e+05	2.1e−03	7.6e+03	10	10	18	9
dixmaane	3000	1.000e+00	2.850e+04	2.6e−04	1.2e+03	11	9	24	8
dixmaanf	3000	1.000e+00	4.104e+04	1.7e−03	1.9e+03	14	12	444	11
dixmaang	3000	1.000e+00	7.607e+04	1.8e−03	3.6e+03	15	13	202	12
dixmaanh	3000	1.000e+00	1.517e+05	4.1e−03	7.4e+03	18	13	225	12
dixmaani	3000	1.000e+00	2.002e+04	4.5e−05	1.0e+03	11	11	3696	10

Continued on next page

Table 2 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
dixmaanj	3000	1.000e+00	3.900e+04	1.4e−03	1.8e+03	12	12	128	11
dixmaank	3000	1.001e+00	7.400e+04	3.3e−03	3.6e+03	12	12	94	11
dixmaanl	3000	1.000e+00	1.496e+05	1.9e−03	7.4e+03	13	13	100	12
dixmaanm	3000	1.000e+00	9.358e+03	3.0e−04	4.4e+02	9	9	1382	8
dixmaann	3000	1.001e+00	2.018e+04	9.8e−04	1.0e+03	17	13	445	12
dixmaano	3000	1.000e+00	3.635e+04	1.9e−03	2.0e+03	20	15	395	14
dixmaanp	3000	1.000e+00	7.128e+04	1.9e−03	4.0e+03	17	15	326	14
dixon3dq	10000	5.207e−10	8.000e+00	6.4e−06	5.7e+00	6	6	21578	5
dqdrtic	5000	4.340e−09	9.041e+06	6.6e−04	8.5e+04	5	5	16	4
dqrtic	5000	2.888e+09	6.241e+17	6.8e+06	1.3e+13	13	13	56	12
edensch	2000	1.200e+04	7.358e+06	8.0e−03	1.0e+05	14	13	46	12
eg2	5000	5.549e+03	2.949e+05	1.5e−03	8.8e+03	9	9	34	8
engval1	5000	5.549e+03	2.949e+05	1.5e−03	8.8e+03	9	9	34	8
errinros	50	4.033e+01	1.102e+05	5.5e−02	1.2e+05	48	26	361	25
errinros_mod	100	7.838e+01	3.140e+05	5.8e−02	1.9e+05	41	25	148	24
errinrsm	50	3.855e+01	1.529e+05	5.6e−02	1.3e+05	34	24	280	23
extrosnb	1000	8.625e−03	3.996e+05	3.3e−02	3.8e+04	37	26	196	25
fletbv3m	5000	−2.492e+05	1.982e+02	2.7e−05	4.4e+01	66	50	119	49
fletcbv2	5000	−5.003e−01	−5.003e−01	1.7e−08	4.4e−06	2	2	9884	1
fletcbv3*	5000	−1.302e+09	1.982e+02	3.3e+01	4.4e+01	10001	10001	15043	10000
fletcbv3_mod*	100	−8.701e−02	−1.879e−02	2.8e−03	1.6e−03	10001	10001	10001	10000
fletchbv*	5000	−3.044e+22	−2.302e+11	1.3e+10	2.8e+09	14970	10001	20057	10000
fletchcr	1000	4.824e−13	9.990e+02	2.3e−05	6.3e+01	1758	1488	33547	1487
fminsrf2	5625	1.000e+00	2.846e+01	9.7e−09	3.3e−01	465	47	101670	46
fminsurf	5625	1.000e+00	2.859e+01	5.3e−08	3.3e−01	349	43	72166	42
freuroth	5000	6.082e+05	5.049e+06	4.2e−03	5.5e+04	25	11	53	10
genhumps	5000	2.405e−06	1.281e+08	1.2e−03	6.0e+03	5593	4720	18566	4719
genrose	500	1.000e+00	1.870e+03	2.3e−04	3.0e+02	829	281	6893	280
genrose_nash	100	1.000e+00	4.041e+02	4.2e−06	1.3e+02	209	71	742	70
hilbertb	10	3.437e−14	5.102e+02	8.4e−07	1.1e+02	6	6	12	5
indef*	5000	−2.322e+09	4.603e+03	3.2e+02	8.0e+01	10348	10001	29124	10000
liarwhd	5000	1.626e−04	2.925e+06	2.6e−02	4.8e+05	13	13	36	12
mancino	100	1.101e−01	1.103e+12	9.2e+02	2.9e+09	6	6	13	5
modbeale	20000	1.299e−03	1.264e+07	7.3e−02	3.1e+05	9	9	106	8
ncb20	5010	−1.458e+03	1.000e+04	9.4e−06	2.8e+02	76	38	734	37
ncb20b	5000	7.351e+03	1.000e+04	2.6e−04	2.8e+02	65	20	1992	19
noncvxu2	5000	1.161e+04	3.235e+11	1.9e+00	3.3e+06	2755	941	8218	940
noncvxun	5000	1.163e+04	3.335e+11	7.1e−01	3.6e+06	3127	942	11348	941
nondia	5000	2.570e−04	2.000e+06	1.3e−02	2.0e+06	3	3	4	2
nondquar	5000	1.301e−03	5.006e+03	1.3e−02	2.0e+04	24	18	196	17
parkch	15	1.624e+03	2.150e+03	3.6e−03	2.5e+04	27	22	280	21
penalty2	200	4.712e+13	4.712e+13	1.3e+00	1.6e+07	12	12	250	11
penalty3	200	1.011e−03	1.584e+09	7.3e−01	2.2e+06	43	18	118	17
powellsg	5000	2.065e−03	2.688e+05	1.3e−02	1.6e+04	13	13	86	12
power	10000	4.738e+07	2.501e+15	8.0e+07	1.2e+14	13	13	80	12
quartc	5000	2.888e+09	6.241e+17	6.8e+06	1.3e+13	13	13	56	12
schmvett	5000	−1.499e+04	−1.429e+04	2.8e−05	7.5e+01	7	7	78	6
scosine*	100	−8.359e+01	8.688e+01	2.7e+03	8.1e+02	10186	10001	1232544	10000
scurly10	10000	1.409e+24	7.006e+31	7.2e+23	1.3e+30	13	13	130	12

Continued on next page

Table 2 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
scurly20	10000	1.847e+25	9.031e+32	9.1e+24	1.6e+31	13	13	124	12
scurly30	10000	1.006e+26	4.163e+33	4.3e+25	7.5e+31	13	13	102	12
sensors	100	−2.109e+03	−5.648e+01	1.8e−11	7.1e+01	28	16	49	15
sinquad	5000	−6.749e+06	6.561e−01	4.3e−03	5.1e+03	37	16	54	15
sparsine	5000	5.750e−03	5.173e+07	1.1e+00	3.0e+06	47	23	3018	22
sparsqur	10000	6.964e−02	1.406e+07	6.5e−01	1.2e+06	13	13	64	12
spmsrtls	4999	2.706e−10	4.141e+03	1.8e−05	7.7e+01	47	17	981	16
srosenbr	5000	1.916e−08	4.850e+04	1.2e−04	1.1e+04	8	8	18	7
ssbrybnd	5000	3.273e−06	1.249e+05	1.7e−01	9.0e+05	349	76	35100	75
stratec	50	0.000e+00	0.000e+00	0.0e+00	0.0e+00	1	1	0	0
testquad	5000	1.768e+00	1.250e+09	1.5e+01	4.4e+07	7	7	1020	6
tointgor	50	1.374e+03	5.074e+03	7.5e−05	6.0e+02	8	8	218	7
tointgss	5000	1.000e+01	4.499e+04	3.8e−05	4.2e+02	4	4	6	3
tointpsp	50	2.256e+02	1.828e+03	1.3e−05	1.1e+02	31	14	182	13
tointqor	50	1.175e+03	2.335e+03	1.2e−05	2.1e+02	7	7	86	6
tquartic	5000	1.474e−15	8.100e−01	1.1e−09	1.8e+00	30	22	58	21
tridia	5000	1.474e−15	8.100e−01	1.1e−09	1.8e+00	30	22	58	21
vardim	200	1.149e+08	3.257e+16	7.3e+09	1.6e+16	13	13	24	12
vareigvl	50	9.344e−13	1.126e+02	4.7e−06	4.2e+01	8	8	66	7
watson	12	1.597e−07	3.000e+01	1.2e−05	2.1e+02	13	13	100	12
woods	4000	7.877e+03	1.919e+07	1.8e−01	5.2e+05	10	10	37	9

7.2. Detailed results for the trust-region method. Detailed results on individual problems are given in Table 3 and Table 4.

Table 3: Solution of 100 nonlinear problems with trust-region CR.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
arglina	200	2.000e+02	1.000e+03	1.8e−13	5.7e+01	3	3	4	2
arglinb	200	1.011e+02	8.353e+15	3.1e−03	1.4e+15	3	3	4	2
arglinc	200	1.011e+02	8.353e+15	3.1e−03	1.4e+15	3	3	4	2
bdqrtic	5000	2.001e+04	1.129e+06	9.2e−01	1.5e+06	10	10	29	9
box	10000	−1.865e+03	0.000e+00	3.5e−07	5.0e+01	7	7	17	6
boxpower	20000	4.736e−02	1.764e+05	6.9e−02	1.7e+05	11	6	22	10
broydn7d	5000	1.846e+03	1.760e+04	9.8e−04	1.1e+03	428	422	2239	427
brybnd	5000	6.074e−08	1.249e+05	1.1e−03	7.8e+03	11	11	67	10
chainwoo	4000	2.799e+03	1.445e+07	3.7e−01	4.2e+05	5656	3545	68977	5655
chnrosnb_mod	100	2.487e−08	1.764e+04	6.7e−04	6.4e+03	184	122	1901	183
chnrsnbm	50	4.924e−08	8.633e+03	9.7e−04	4.4e+03	101	68	965	100
clplatea	5041	−1.259e−02	0.000e+00	5.2e−07	1.0e−01	21	15	681	20
clplateb	5041	−5.095e−03	0.000e+00	3.6e−07	1.2e−02	4	4	414	3
clplatec	5041	−5.021e−03	0.000e+00	7.3e−07	9.9e−02	5	5	10965	4
cosine	10000	−9.999e+03	8.775e+03	3.0e−05	7.2e+01	12	12	26	11
cragglvy	5000	1.688e+03	2.749e+06	1.1e−01	2.8e+05	13	13	57	12
curly10	10000	−1.003e+06	−6.306e−01	3.9e−05	1.3e+02	24	21	48883	23
curly20	10000	−1.003e+06	−1.344e+00	2.6e−04	3.0e+02	24	21	47039	23
curly30	10000	−1.003e+06	−2.190e+00	7.5e−05	5.1e+02	26	22	48385	25
dixmaana	3000	1.000e+00	2.850e+04	5.5e−04	1.2e+03	9	9	19	8
dixmaanb	3000	1.000e+00	4.724e+04	1.8e−03	2.0e+03	8	8	14	7

Continued on next page

Table 3 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
dixmaanc	3000	1.000e+00	8.248e+04	1.6e−04	3.7e+03	10	10	18	9
dixmaand	3000	1.000e+00	1.586e+05	5.0e−03	7.6e+03	10	10	18	9
dixmaane	3000	1.000e+00	2.209e+04	3.8e−05	1.1e+03	11	11	204	10
dixmaanf	3000	1.000e+00	4.104e+04	1.1e−03	1.9e+03	12	12	91	11
dixmaang	3000	1.001e+00	7.607e+04	3.2e−03	3.6e+03	12	12	46	11
dixmaanh	3000	1.001e+00	1.517e+05	2.0e−03	7.4e+03	13	13	60	12
dixmaani	3000	1.000e+00	2.002e+04	2.4e−04	1.0e+03	11	11	297	10
dixmaanj	3000	1.000e+00	3.900e+04	9.4e−04	1.8e+03	13	13	79	12
dixmaank	3000	1.001e+00	7.400e+04	2.5e−03	3.6e+03	13	13	56	12
dixmaanl	3000	1.001e+00	1.496e+05	5.7e−03	7.4e+03	13	13	46	12
dixmaanm	3000	1.000e+00	9.358e+03	6.1e−05	4.4e+02	10	10	507	9
dixmaann	3000	1.001e+00	2.018e+04	9.3e−04	1.0e+03	13	13	123	12
dixmaano	3000	1.000e+00	3.635e+04	6.3e−04	2.0e+03	14	14	134	13
dixmaanp	3000	1.001e+00	7.128e+04	1.7e−03	4.0e+03	14	14	89	13
dixon3dq	10000	1.195e−04	8.000e+00	4.9e−06	5.7e+00	6	6	10799	5
dqdrtic	5000	6.563e−06	9.041e+06	5.1e−03	8.5e+04	8	8	18	7
dqrtic	5000	1.741e+09	6.241e+17	4.2e+06	1.3e+13	21	21	54	20
edensch	2000	1.200e+04	7.358e+06	2.4e−02	1.0e+05	13	13	30	12
eg2	1000	−9.989e+02	−8.406e+02	6.0e−09	5.4e+02	4	4	6	3
engval1	5000	5.549e+03	2.949e+05	7.9e−04	8.8e+03	11	11	29	10
errinros	50	4.034e+01	1.102e+05	1.1e−01	1.2e+05	42	26	382	41
errinros_mod	100	7.858e+01	3.140e+05	1.4e−01	1.9e+05	30	18	220	29
errinrsm	50	3.866e+01	1.529e+05	8.4e−02	1.3e+05	36	20	300	35
extrosnb	1000	4.800e−03	3.996e+05	1.5e−02	3.8e+04	61	34	343	60
fletbv3m	5000	−2.394e+05	1.982e+02	5.0e−06	4.4e+01	16	12	30	15
fletcbv2	5000	−5.003e−01	−5.003e−01	1.7e−08	4.4e−06	2	2	4843	1
fletcbv3*	5000	−1.941e+09	1.982e+02	3.6e+01	4.4e+01	10001	9728	25199	10000
fletcbv3_mod	100	−2.047e+00	−1.879e−02	1.6e−08	1.6e−03	43	39	103	42
fletchbv*	5000	−2.372e+17	−2.302e+11	3.7e+09	2.8e+09	10001	10000	25863	10000
fletchcr	1000	8.100e−11	9.990e+02	5.0e−05	6.3e+01	2347	1564	28041	2346
fminsrf2	5625	1.000e+00	2.846e+01	2.2e−09	3.3e−01	221	212	1522	220
fminsurf	5625	1.000e+00	2.859e+01	5.0e−07	3.3e−01	710	703	2689	709
freuroth	5000	6.082e+05	5.049e+06	4.2e−03	5.5e+04	12	12	34	11
genhumps	5000	5.468e−06	1.281e+08	1.4e−03	6.0e+03	6453	6222	25419	6452
genrose	500	1.000e+00	1.870e+03	4.1e−05	3.0e+02	378	308	4362	377
genrose_nash	100	1.000e+00	4.041e+02	4.1e−05	1.3e+02	96	77	931	95
hilbertb	10	2.216e−14	5.102e+02	6.8e−07	1.1e+02	6	6	11	5
indef*	5000	−2.024e+13	4.603e+03	7.1e+01	8.0e+01	10001	7246	30578	10000
indefm	100000	−1.005e+07	9.207e+04	1.3e−06	3.6e+02	24	20	69	23
liarwhd	5000	1.825e−02	2.925e+06	2.7e−01	4.8e+05	13	13	31	12
mancino	100	5.784e−02	1.103e+12	6.6e+02	2.9e+09	12	11	24	11
modbeale	20000	1.041e−02	1.264e+07	2.1e−01	3.1e+05	10	10	53	9
ncb20	5010	−1.459e+03	1.000e+04	2.6e−05	2.8e+02	64	52	526	63
ncb20b	5000	7.351e+03	1.000e+04	8.7e−05	2.8e+02	33	21	1214	32
noncvxu2	5000	1.326e+04	3.235e+11	3.2e+00	3.3e+06	935	864	4710	934
noncvxun	5000	1.335e+04	3.335e+11	3.5e+00	3.6e+06	906	824	4605	905
nondia	5000	2.570e−04	2.000e+06	1.3e−02	2.0e+06	3	3	4	2
nondquar	5000	1.519e−03	5.006e+03	5.7e−03	2.0e+04	31	19	229	30
parkch	15	1.624e+03	2.150e+03	1.8e−02	2.5e+04	28	20	234	27

Continued on next page

Table 3 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
penalty2	200	4.712e+13	4.712e+13	3.6e+00	1.6e+07	12	12	127	11
penalty3	200	1.002e−03	1.584e+09	1.9e−01	2.2e+06	32	25	117	31
powellsg	5000	2.085e−03	2.688e+05	1.3e−02	1.6e+04	14	14	57	13
power	10000	6.432e+07	2.501e+15	6.0e+07	1.2e+14	14	14	54	13
quartc	5000	1.741e+09	6.241e+17	4.2e+06	1.3e+13	21	21	54	20
schmvett	5000	−1.499e+04	−1.429e+04	2.5e−06	7.5e+01	8	8	55	7
scosine	100	−9.900e+01	8.688e+01	7.7e−04	8.1e+02	262	242	38423	261
scurly10	10000	1.602e+24	7.006e+31	4.1e+23	1.3e+30	15	15	74	14
scurly20	10000	2.013e+25	9.031e+32	5.3e+24	1.6e+31	15	15	72	14
scurly30	10000	8.866e+25	4.163e+33	2.4e+25	7.5e+31	15	15	71	14
sensors	100	−2.055e+03	−5.648e+01	1.9e−10	7.1e+01	17	15	51	16
sinquad	5000	−6.757e+06	6.561e−01	6.8e−05	5.1e+03	14	14	38	13
sparsine	5000	3.184e−02	5.173e+07	8.0e−01	3.0e+06	10	10	262	9
sparsqur	10000	1.693e−01	1.406e+07	1.1e+00	1.2e+06	13	13	39	12
spmsrtls	4999	1.647e−08	4.141e+03	5.0e−05	7.7e+01	14	14	207	13
srosenbr	5000	1.831e−08	4.850e+04	1.2e−04	1.1e+04	8	8	16	7
ssbrybnd	5000	2.227e−03	1.249e+05	4.9e−01	9.0e+05	27	19	7783	26
sscosine	5000	−4.997e+03	4.387e+03	5.4e−03	5.9e+03	441	316	1022191	440
stratec	10	2.212e+03	2.818e+03	1.4e−02	4.7e+04	49	39	302	48
testquad	5000	7.180e+00	1.250e+09	5.6e+00	4.4e+07	9	9	456	8
tointgor	50	1.374e+03	5.074e+03	2.3e−05	6.0e+02	9	9	125	8
tointgss	5000	1.000e+01	4.499e+04	1.0e−05	4.2e+02	9	9	24	8
tointpsp	50	2.256e+02	1.828e+03	6.7e−05	1.1e+02	29	24	126	28
tointqor	50	1.175e+03	2.335e+03	1.8e−05	2.1e+02	7	7	47	6
tquartic	5000	2.424e−21	8.100e−01	1.4e−08	1.8e+00	11	11	24	10
tridia	5000	4.377e−04	1.250e+07	1.0e−01	4.1e+05	9	9	580	8
vardim	200	1.149e+08	3.257e+16	7.3e+09	1.6e+16	13	13	24	12
vareigvl	50	6.242e−11	1.126e+02	2.7e−05	4.2e+01	8	7	33	7
watson	12	1.602e−07	3.000e+01	4.8e−06	2.1e+02	14	14	69	13
woods	4000	7.877e+03	1.919e+07	4.4e−02	5.2e+05	9	9	25	8

Table 4: Solution of 100 nonlinear problems with trust-region CG.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
arglina	200	2.000e+02	1.000e+03	1.8e−13	5.7e+01	3	3	4	2
arglinb	200	1.011e+02	8.353e+15	4.6e−02	1.4e+15	3	3	4	2
arglinc	200	1.011e+02	8.353e+15	4.6e−02	1.4e+15	3	3	4	2
bdqrtic	5000	2.001e+04	1.129e+06	8.4e−01	1.5e+06	10	10	29	9
box	10000	−1.865e+03	0.000e+00	1.9e−07	5.0e+01	10	9	25	9
boxpower	20000	4.736e−02	1.764e+05	6.9e−02	1.7e+05	12	7	29	11
broydn7d	5000	1.825e+03	1.760e+04	4.2e−05	1.1e+03	480	472	2345	479
brybnd	5000	6.154e−07	1.249e+05	4.3e−03	7.8e+03	10	10	57	9
chainwoo	4000	2.856e+03	1.445e+07	9.9e−02	4.2e+05	4939	3223	46889	4938
chnrosnb_mod	100	4.134e−07	1.764e+04	3.3e−03	6.4e+03	173	117	1768	172
chnrsnbm	50	4.507e−10	8.633e+03	4.7e−04	4.4e+03	102	69	955	101
clplatea	5041	−1.259e−02	0.000e+00	3.2e−07	1.0e−01	28	22	1150	27
clplateb	5041	−5.095e−03	0.000e+00	2.2e−07	1.2e−02	4	4	459	3
clplatec	5041	−5.021e−03	0.000e+00	2.7e−08	9.9e−02	5	5	9374	4
cosine	10000	−9.999e+03	8.775e+03	2.5e−05	7.2e+01	12	12	26	11

Continued on next page

Table 4 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	#f	#g	#Hv	#it
cragglvy	5000	1.688e+03	2.749e+06	1.1e−01	2.8e+05	13	13	62	12
curly10	10000	−1.003e+06	−6.306e−01	5.1e−05	1.3e+02	24	20	57435	23
curly20	10000	−1.003e+06	−1.344e+00	1.7e−04	3.0e+02	22	18	64559	21
curly30	10000	−1.003e+06	−2.190e+00	4.2e−04	5.1e+02	18	15	59948	17
dixmaana	3000	1.000e+00	2.850e+04	4.7e−04	1.2e+03	9	9	19	8
dixmaanb	3000	1.000e+00	4.724e+04	1.9e−03	2.0e+03	8	8	14	7
dixmaanc	3000	1.000e+00	8.248e+04	1.6e−04	3.7e+03	10	10	18	9
dixmaand	3000	1.000e+00	1.586e+05	4.7e−03	7.6e+03	10	10	18	9
dixmaane	3000	1.000e+00	2.209e+04	5.7e−05	1.1e+03	11	11	251	10
dixmaanf	3000	1.000e+00	4.104e+04	9.8e−04	1.9e+03	23	12	587	22
dixmaang	3000	1.000e+00	7.607e+04	2.3e−03	3.6e+03	27	13	892	26
dixmaanh	3000	1.000e+00	1.517e+05	4.8e−03	7.4e+03	29	14	294	28
dixmaani	3000	1.000e+00	2.002e+04	2.7e−04	1.0e+03	11	11	762	10
dixmaanj	3000	1.000e+00	3.900e+04	4.5e−04	1.8e+03	28	14	815	27
dixmaank	3000	1.000e+00	7.400e+04	1.8e−03	3.6e+03	13	13	68	12
dixmaanl	3000	1.001e+00	1.496e+05	5.0e−03	7.4e+03	13	13	49	12
dixmaanm	3000	1.000e+00	9.358e+03	5.7e−05	4.4e+02	10	10	1589	9
dixmaann	3000	1.000e+00	2.018e+04	5.9e−05	1.0e+03	29	15	1922	28
dixmaano	3000	1.000e+00	3.635e+04	1.4e−03	2.0e+03	27	14	987	26
dixmaanp	3000	1.002e+00	7.128e+04	3.2e−03	4.0e+03	13	13	85	12
dixon3dq	10000	5.207e−10	8.000e+00	6.4e−06	5.7e+00	6	6	10794	5
dqdrtic	5000	1.111e−09	9.041e+06	4.3e−04	8.5e+04	8	8	18	7
dqrtic	5000	6.837e+09	6.241e+17	1.2e+07	1.3e+13	20	20	53	19
edensch	2000	1.200e+04	7.358e+06	7.5e−02	1.0e+05	14	14	37	13
eg2	1000	−9.989e+02	−8.406e+02	6.0e−09	5.4e+02	4	4	6	3
engval1	5000	5.549e+03	2.949e+05	5.8e−04	8.8e+03	11	11	31	10
errinros	50	4.009e+01	1.102e+05	1.0e−01	1.2e+05	62	40	586	61
errinros_mod	100	7.846e+01	3.140e+05	1.2e−01	1.9e+05	39	23	314	38
errinrsm	50	3.856e+01	1.529e+05	1.1e−01	1.3e+05	41	24	331	40
extrosnb	1000	5.872e−03	3.996e+05	3.0e−02	3.8e+04	54	31	269	53
fletbv3m	5000	−2.394e+05	1.982e+02	1.3e−06	4.4e+01	18	14	34	17
fletcbv2	5000	−5.003e−01	−5.003e−01	1.7e−08	4.4e−06	2	2	4943	1
fletcbv3*	5000	−7.478e+12	1.982e+02	3.7e+01	4.4e+01	10001	9993	24984	10000
fletcbv3_mod	100	−2.034e+00	−1.879e−02	5.0e−07	1.6e−03	32	26	73	31
fletchbv*	5000	−1.074e+21	−2.302e+11	3.8e+09	2.8e+09	10001	9999	24970	10000
fletchcr	1000	1.308e−12	9.990e+02	4.4e−05	6.3e+01	2390	1415	29831	2389
fminsrf2	5625	1.000e+00	2.846e+01	3.1e−07	3.3e−01	1473	1469	3612	1472
fminsurf	5625	1.000e+00	2.859e+01	4.0e−09	3.3e−01	1510	1506	3631	1509
freuroth	5000	6.082e+05	5.049e+06	5.4e−02	5.5e+04	11	11	28	10
genhumps*	5000	2.154e+01	1.281e+08	3.1e+00	6.0e+03	10001	9726	32001	10000
genrose	500	1.000e+00	1.870e+03	7.0e−05	3.0e+02	582	465	4275	581
genrose_nash	100	1.000e+00	4.041e+02	6.4e−05	1.3e+02	164	128	1100	163
hilbertb	10	3.437e−14	5.102e+02	8.4e−07	1.1e+02	6	6	11	5
indef*	5000	−4.959e+07	4.603e+03	9.8e+02	8.0e+01	10001	10001	29258	10000
indefm	100000	−9.865e+06	9.207e+04	1.3e−05	3.6e+02	377	328	1240	376
liarwhd	5000	1.535e−02	2.925e+06	2.5e−01	4.8e+05	14	14	33	13
mancino	100	2.340e−02	1.103e+12	4.2e+02	2.9e+09	12	11	23	11
modbeale	20000	3.158e−04	1.264e+07	2.8e−02	3.1e+05	11	11	65	10
ncb20	5010	−1.460e+03	1.000e+04	6.0e−05	2.8e+02	71	60	412	70

Continued on next page

Table 4 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#f$	$\#g$	$\#Hv$	$\#it$
ncb20b	5000	7.351e+03	1.000e+04	7.9e−05	2.8e+02	31	21	1535	30
noncvxu2	5000	1.159e+04	3.235e+11	9.5e−01	3.3e+06	3628	3558	12209	3627
noncvxun	5000	1.227e+04	3.335e+11	2.8e+00	3.6e+06	3930	3882	12459	3929
nondia	5000	2.570e−04	2.000e+06	1.3e−02	2.0e+06	3	3	4	2
nondquar	5000	1.581e−03	5.006e+03	1.0e−02	2.0e+04	31	19	246	30
parkch	15	1.624e+03	2.150e+03	4.6e−03	2.5e+04	32	23	223	31
penalty2	200	4.712e+13	4.712e+13	1.3e+00	1.6e+07	12	12	136	11
penalty3	200	1.002e−03	1.584e+09	3.5e−01	2.2e+06	35	26	129	34
powellsg	5000	2.277e−03	2.688e+05	1.4e−02	1.6e+04	14	14	57	13
power	10000	3.252e+07	2.501e+15	5.6e+07	1.2e+14	14	14	55	13
quartc	5000	6.837e+09	6.241e+17	1.2e+07	1.3e+13	20	20	53	19
schmvett	5000	−1.499e+04	−1.429e+04	7.3e−06	7.5e+01	7	7	50	6
scosine	100	−9.900e+01	8.688e+01	5.8e−04	8.1e+02	336	301	27244	335
scurly10	10000	2.554e+24	7.006e+31	1.2e+24	1.3e+30	14	14	76	13
scurly20	10000	3.493e+25	9.031e+32	1.5e+25	1.6e+31	14	14	71	13
scurly30	10000	1.703e+26	4.163e+33	7.1e+25	7.5e+31	14	14	66	13
sensors	100	−2.041e+03	−5.648e+01	2.1e−05	7.1e+01	17	14	48	16
sinquad	5000	−6.757e+06	6.561e−01	6.5e−05	5.1e+03	17	17	45	16
sparsine	5000	1.499e−03	5.173e+07	7.4e−01	3.0e+06	10	10	461	9
sparsqur	10000	1.322e−01	1.406e+07	9.9e−01	1.2e+06	13	13	41	12
spmsrtls	4999	2.434e−10	4.141e+03	4.2e−05	7.7e+01	32	25	523	31
srosenbr	5000	1.916e−08	4.850e+04	1.2e−04	1.1e+04	8	8	16	7
ssbrybnd	5000	2.774e−04	1.249e+05	5.6e−01	9.0e+05	6086	4554	143194	6085
sscosine*	5000	−3.992e+03	4.387e+03	6.0e+03	5.9e+03	10001	9847	1955112	10000
stratec	10	2.212e+03	2.818e+03	2.7e−02	4.7e+04	52	40	310	51
testquad	5000	4.323e−01	1.250e+09	7.4e+00	4.4e+07	9	9	632	8
tointgor	50	1.374e+03	5.074e+03	3.8e−04	6.0e+02	8	8	104	7
tointgss	5000	1.000e+01	4.499e+04	4.5e−07	4.2e+02	9	9	25	8
tointpsp	50	2.256e+02	1.828e+03	1.0e−06	1.1e+02	51	38	193	50
tointqor	50	1.175e+03	2.335e+03	4.9e−05	2.1e+02	7	7	44	6
tquartic	5000	2.439e−21	8.100e−01	1.4e−08	1.8e+00	11	11	24	10
tridia	5000	1.432e−05	1.250e+07	9.7e−02	4.1e+05	9	9	648	8
vardim	200	1.149e+08	3.257e+16	7.3e+09	1.6e+16	13	13	24	12
vareigvl	50	6.049e−14	1.126e+02	9.6e−07	4.2e+01	9	8	40	8
watson	12	1.597e−07	3.000e+01	1.2e−05	2.1e+02	13	13	62	12
woods	4000	7.877e+03	1.919e+07	5.9e−02	5.2e+05	9	9	25	8

7.3. Detailed results for nonlinear least squares trust-region method. Detailed results for individual problems are given in Table 5 and Table 6.

Table 5: Solution of 97 nonlinear least-squares problems with LSMR.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
10foldtr	1000	8.1e+32	5.0e+39	1.2e+32	1.6e+38	26	76	76	25
argtrig	200	8.9e−09	3.3e+01	1.6e−04	1.3e+03	9	631	631	8
artif	5002	2.6e−08	9.1e+02	7.4e−04	9.5e+02	16	873	873	15
arwhdne	500	7.0e+01	1.2e+03	1.6e−03	2.0e+03	144	419	419	143
ba-l1	57	2.3e−10	6.4e+04	2.2e−02	3.1e+05	7	34	34	6

Continued on next page

Table 5 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
ba-l16	200	8.9e−09	3.3e+01	1.6e−04	1.3e+03	9	631	631	8
ba-l1sp	57	5.5e−09	6.4e+04	1.2e−01	3.1e+05	8	40	40	7
ba-l73*	33753	2.2e+07	1.2e+08	7.7e+06	3.1e+08	10001	32592	32592	10000
bdvalue	5002	5.2e−12	5.2e−12	2.1e−07	2.1e−07	1	1	1	0
bratu2d	5184	4.1e−16	1.5e−03	6.0e−09	1.9e−02	5	2066	2066	4
bratu2dt	5184	8.8e−13	4.5e−03	3.5e−07	3.2e−02	5	2045	2045	4
bratu3d	4913	2.8e−14	1.2e+00	3.4e−07	1.5e+00	6	296	296	5
brownale	200	7.8e−09	1.0e+06	1.5e−03	2.8e+05	2	4	4	1
broydn3d	5000	1.5e−11	2.5e+03	2.0e−05	2.8e+02	7	36	36	6
broydnbd	5000	1.5e−08	6.2e+04	3.7e−04	3.9e+03	12	77	77	11
cbratu2d	3200	3.7e−15	7.8e−03	7.9e−08	5.8e−02	5	692	692	4
cbratu3d	3456	3.4e−16	1.6e+00	8.1e−08	2.1e+00	6	173	173	5
chandheu	500	4.5e−08	1.7e+01	1.9e−06	2.9e+00	13	127	127	12
channel	9600	4.1e−01	2.9e+07	7.1e−01	9.4e+05	8	40	40	7
chnrsbne	50	1.5e−09	3.8e+03	1.0e−04	1.8e+03	95	998	998	94
cyclic3	100002	3.6e+14	5.0e+22	1.6e+11	9.5e+17	48	142	142	47
deconvne	63	6.8e−10	5.5e+01	1.2e−06	5.3e+01	24	433	433	23
dmn15102	66	4.4e+03	1.8e+06	4.9e+01	5.1e+07	47	459	459	46
dmn15103	99	2.4e+02	2.0e+06	3.5e+01	5.3e+07	77	1725	1725	76
dmn15332	66	1.1e+02	2.5e+05	4.6e+00	5.8e+06	179	3298	3298	178
dmn15333	99	7.5e+01	2.4e+05	3.8e+00	5.5e+06	171	6922	6922	170
dmn37142	66	1.1e+02	1.3e+05	3.1e+00	3.8e+06	145	2190	2190	144
dmn37143	99	1.7e+02	7.5e+04	3.3e+00	3.8e+06	32	137	137	31
eigena	2550	1.8e−07	2.0e+04	2.9e−04	4.5e+02	100	4095	4095	99
eigenau	2550	1.8e−07	2.0e+04	2.9e−05	4.5e+02	103	4224	4224	102
eigenb	2550	6.3e−06	5.0e+01	3.7e−06	1.9e+01	1192	74133	74133	1191
eigenc	2652	2.4e−06	5.6e+03	8.2e−05	2.4e+02	430	53610	53610	429
hatfldg	25	1.6e−10	1.4e+01	2.0e−05	2.5e+01	10	69	69	9
hydcar20	99	6.5e−02	6.7e+02	3.2e−03	3.6e+03	39	2823	2823	38
hydcar6	29	9.8e−03	3.5e+02	1.4e−03	2.2e+03	43	1188	1188	42
integreq	502	9.3e−17	1.4e+00	1.4e−08	2.1e+00	5	17	17	4
inteqne	12	2.4e−17	3.2e−02	7.1e−09	3.1e−01	4	14	14	3
kss	1000	1.8e+03	2.0e+15	6.1e+04	1.9e+11	20	58	58	19
luksan11	100	3.2e−14	3.1e+02	2.5e−06	1.1e+02	298	3434	3434	297
luksan12	98	2.1e+03	1.6e+04	2.5e−03	3.0e+03	38	314	314	37
luksan13	98	1.3e+04	3.2e+04	4.0e−03	4.2e+03	45	312	312	44
luksan14	98	6.2e+01	1.3e+04	2.7e−03	5.1e+03	19	191	191	18
luksan15	100	1.8e+00	1.4e+04	2.7e−03	7.1e+03	9	34	34	8
luksan16	100	1.8e+00	6.5e+03	6.6e−03	1.5e+04	7	31	31	6
luksan17	100	2.5e−01	8.4e+05	1.1e−01	2.0e+05	27	269	269	26
luksan21	100	9.3e−14	5.0e+01	6.8e−08	1.4e+00	17	1365	1365	16
luksan22	100	4.3e+02	1.2e+04	3.4e−03	3.6e+03	1639	15884	15884	1638
lukšan-vlček5.1	20	8.1e−08	2.3e+03	2.4e−04	1.5e+03	61	529	529	60

Continued on next page

Table 5 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
lukšan-vlček5.11	20	2.7e−08	4.5e+00	3.3e−06	6.2e+00	8	43	43	7
lukšan-vlček5.12	21	1.9e−08	4.2e+01	4.3e−06	2.7e+01	9	89	89	8
lukšan-vlček5.13	20	1.8e−07	2.5e+02	2.8e−05	6.9e+01	8	37	37	7
lukšan-vlček5.14	20	1.1e−03	1.6e+05	1.0e−02	5.7e+04	9	37	37	8
lukšan-vlček5.15	21	4.2e−01	6.4e+06	3.3e−01	4.1e+05	17	68	68	16
lukšan-vlček5.16	21	2.0e−07	5.6e+01	3.1e−05	5.2e+01	8	43	43	7
lukšan-vlček5.17	21	2.0e−07	1.4e+02	3.0e−05	7.0e+01	8	44	44	7
lukšan-vlček5.18	21	1.9e−09	1.5e+01	1.3e−06	7.1e+00	9	33	33	8
lukšan-vlček5.2	20	3.5e+01	7.8e+03	6.2e−03	6.9e+03	12	92	92	11
lukšan-vlček5.3	20	1.5e−05	2.2e+03	1.5e−03	1.5e+03	8	48	48	7
lukšan-vlček5.4	20	1.7e+00	2.4e+03	5.5e−03	5.7e+03	25	214	214	24
methanb8	31	9.3e−06	5.2e−01	2.4e−04	2.9e+02	16	309	309	15
methanl8	31	3.1e−02	2.2e+03	1.4e−02	2.8e+04	51	1322	1322	50
mgh19	11	4.4e−02	1.6e+01	7.1e−06	2.2e+01	57	507	507	56
mgh21	20	2.2e−11	1.2e+02	3.0e−06	3.7e+02	31	110	110	30
mgh22	20	5.5e−06	5.4e+02	4.0e−04	5.1e+02	9	44	44	8
mgh25	10	3.7e−06	1.1e+06	5.3e−02	2.2e+06	9	25	25	8
mgh26	10	1.4e−05	3.5e−03	6.4e−07	5.0e−02	29	245	245	28
mgh27	10	3.1e−07	1.4e+02	7.8e−05	1.7e+02	5	14	14	4
mgh28	10	2.0e−16	3.9e−04	4.0e−09	2.0e−02	4	35	35	3
mgh29	10	2.4e−17	3.2e−02	7.1e−09	3.1e−01	4	14	14	3
mgh30	10	1.2e−11	1.1e+01	2.1e−05	2.5e+01	6	33	33	5
mgh31	10	1.4e−12	1.8e+02	9.9e−06	4.1e+02	7	27	27	6
mgh32	10	5.0e+00	2.5e+01	7.0e−16	6.3e+00	2	4	4	1
mgh33	10	2.3e+00	4.3e+06	1.4e−09	3.1e+06	2	4	4	1
mgh34	10	3.1e+00	2.0e+06	4.3e−10	1.6e+06	2	4	4	1
mgh35	10	3.3e−03	1.7e−02	1.1e−06	6.7e−01	30	228	228	29
morebvne	10	1.9e−15	8.0e−03	1.7e−08	1.2e−01	4	35	35	3
msqrta	1024	5.5e−07	4.0e+03	7.4e−05	1.7e+02	47	16071	16071	46
msqrtb	1024	2.2e−08	4.0e+03	1.6e−05	1.7e+02	34	6992	6992	33
nystrom5	18	6.5e−09	9.1e−01	2.9e−07	3.1e+00	34	564	564	33
nzf1	13	3.9e−15	5.0e+03	3.4e−07	9.3e+02	7	36	36	6
osborne2	11	1.6e−01	1.6e+00	3.0e−06	3.2e+00	81	839	839	80
oscigrne	100000	3.4e−04	3.1e+08	1.0e+02	1.1e+09	8	45	45	7
oscipane*	10	5.0e−01	5.0e−01	2.7e−06	5.0e−01	10001	29658	29658	10000
penlt1ne	10	1.1e−06	7.4e+04	1.5e−03	1.5e+04	8	22	22	7
porous1	5184	9.8e−04	5.0e+06	9.8e+00	1.3e+07	12	3560	3560	11
porous2	5184	3.0e−05	5.4e+06	1.4e+00	1.4e+07	14	4411	4411	13
semicn2u	5002	2.5e−03	9.8e+03	2.6e−04	3.4e+02	179	3604	3604	178
spin	1327	9.8e−04	6.7e+04	4.4e−03	2.2e+04	12	46	46	11
spin2	102	5.9e−09	1.6e+04	3.5e−03	4.4e+03	7	25	25	6
spmsqrt	4999	1.7e−10	2.1e+03	2.6e−05	3.9e+01	13	242	242	12
vandaniums	22	4.8e+00	7.5e+00	1.6e−06	1.2e+00	7	23	23	6

Continued on next page

Table 5 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
vardimne	10	3.7e−06	1.1e+06	5.3e−02	2.2e+06	9	25	25	8
watsonne	12	2.0e−06	1.5e+01	3.2e−05	4.3e+01	7	32	32	6
woodsne	4000	4.5e+00	1.5e+08	1.5e+00	6.4e+06	31	143	143	30
yatp1ne	2600	2.7e−05	2.6e+07	1.0e−02	4.4e+05	31	99	99	30
yatp1sq	123200	1.6e−01	1.3e+09	2.3e+00	8.1e+06	20	60	60	19
yatp2sq	123200	3.0e−08	3.8e+09	1.3e−02	3.8e+05	35	102	102	34

Table 6: Solution of 97 nonlinear least-squares problems with LSQR.

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
10foldtr	1000	8.1e+32	5.0e+39	1.2e+32	1.6e+38	26	76	76	25
argtrig	200	1.1e−11	3.3e+01	7.6e−05	1.3e+03	8	654	654	7
artif	5002	1.4e−09	9.1e+02	3.7e−05	9.5e+02	18	1022	1022	17
arwhdne	500	7.0e+01	1.2e+03	1.6e−03	2.0e+03	147	430	430	146
ba-l1	57	1.7e−09	6.4e+04	7.1e−02	3.1e+05	7	35	35	6
ba-l16	200	1.1e−11	3.3e+01	7.6e−05	1.3e+03	8	654	654	7
ba-l1sp	57	2.1e−09	6.4e+04	9.8e−02	3.1e+05	8	42	42	7
ba-l73*	33753	7.4e+07	1.2e+08	5.3e+07	3.1e+08	10001	37754	37754	10000
bdvalue	5002	5.2e−12	5.2e−12	2.1e−07	2.1e−07	1	1	1	0
bratu2d	5184	5.7e−14	1.5e−03	3.1e−07	1.9e−02	4	1417	1417	3
bratu2dt	5184	1.1e−13	4.5e−03	9.2e−07	3.2e−02	5	2022	2022	4
bratu3d	4913	1.5e−15	1.2e+00	1.4e−07	1.5e+00	6	246	246	5
brownale	200	7.8e−09	1.0e+06	1.5e−03	2.8e+05	2	4	4	1
broydn3d	5000	1.1e−10	2.5e+03	6.6e−05	2.8e+02	7	35	35	6
broydnbd	5000	2.4e−07	6.2e+04	1.8e−03	3.9e+03	12	62	62	11
cbratu2d	3200	1.2e−19	7.8e−03	1.9e−09	5.8e−02	5	690	690	4
cbratu3d	3456	3.3e−17	1.6e+00	5.4e−08	2.1e+00	6	143	143	5
chandheu	500	9.8e−09	1.7e+01	5.7e−07	2.9e+00	14	143	143	13
channel	9600	3.2e−01	2.9e+07	1.9e−01	9.4e+05	9	46	46	8
chnrsbne	50	2.1e−08	3.8e+03	5.8e−04	1.8e+03	101	955	955	100
cyclic3	100002	3.6e+14	5.0e+22	1.6e+11	9.5e+17	48	142	142	47
deconvne	63	1.3e−09	5.5e+01	3.0e−06	5.3e+01	24	457	457	23
dmn15102	66	4.3e+03	1.8e+06	4.5e+01	5.1e+07	269	2542	2542	268
dmn15103	99	1.7e+02	2.0e+06	3.8e+01	5.3e+07	174	2075	2075	173
dmn15332	66	5.5e+02	2.5e+05	5.5e+00	5.8e+06	957	11189	11189	956
dmn15333	99	4.7e+01	2.4e+05	5.4e+00	5.5e+06	1888	34421	34421	1887
dmn37142	66	1.1e+02	1.3e+05	3.7e+00	3.8e+06	114	1324	1324	113
dmn37143	99	1.4e+02	7.5e+04	5.1e−01	3.8e+06	97	439	439	96
eigena	2550	4.7e−08	2.0e+04	2.3e−05	4.5e+02	95	3943	3943	94
eigenau	2550	8.6e−06	2.0e+04	2.0e−04	4.5e+02	105	4139	4139	104
eigenb	2550	9.6e−07	5.0e+01	4.3e−06	1.9e+01	995	75269	75269	994
eigenc	2652	1.5e−08	5.6e+03	1.5e−04	2.4e+02	408	39554	39554	407
hatfldg	25	4.9e−15	1.4e+01	9.9e−08	2.5e+01	10	66	66	9

Continued on next page

Table 6 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
hydcar20	99	7.5e−03	6.7e+02	2.1e−03	3.6e+03	233	34516	34516	232
hydcar6	29	4.7e−03	3.5e+02	2.0e−03	2.2e+03	82	2357	2357	81
integreq	502	8.9e−17	1.4e+00	1.4e−08	2.1e+00	5	17	17	4
inteqne	12	2.3e−17	3.2e−02	7.0e−09	3.1e−01	4	14	14	3
kss	1000	1.8e+03	2.0e+15	6.1e+04	1.9e+11	20	58	58	19
luksan11	100	1.9e−15	3.1e+02	6.1e−07	1.1e+02	336	3041	3041	335
luksan12	98	2.1e+03	1.6e+04	1.3e−03	3.0e+03	42	318	318	41
luksan13	98	1.3e+04	3.2e+04	3.9e−03	4.2e+03	76	447	447	75
luksan14	98	6.2e+01	1.3e+04	4.6e−03	5.1e+03	16	157	157	15
luksan15	100	1.8e+00	1.4e+04	2.6e−03	7.1e+03	9	34	34	8
luksan16	100	1.8e+00	6.5e+03	1.3e−03	1.5e+04	8	36	36	7
luksan17	100	2.5e−01	8.4e+05	8.4e−02	2.0e+05	28	332	332	27
luksan21	100	2.5e−14	5.0e+01	8.3e−08	1.4e+00	17	1210	1210	16
luksan22	100	4.3e+02	1.2e+04	2.4e−03	3.6e+03	332	2154	2154	331
lukšan-vlček5.1	20	8.2e−11	2.3e+03	1.7e−04	1.5e+03	61	468	468	60
lukšan-vlček5.11	20	2.1e−08	4.5e+00	2.7e−06	6.2e+00	8	45	45	7
lukšan-vlček5.12	21	2.2e−07	4.2e+01	2.5e−05	2.7e+01	8	67	67	7
lukšan-vlček5.13	20	1.3e−07	2.5e+02	2.1e−05	6.9e+01	8	37	37	7
lukšan-vlček5.14	20	8.0e−04	1.6e+05	7.5e−03	5.7e+04	9	36	36	8
lukšan-vlček5.15	21	2.1e−02	6.4e+06	7.8e−02	4.1e+05	20	88	88	19
lukšan-vlček5.16	21	1.4e−07	5.6e+01	2.3e−05	5.2e+01	8	45	45	7
lukšan-vlček5.17	21	2.0e−07	1.4e+02	3.0e−05	7.0e+01	8	44	44	7
lukšan-vlček5.18	21	1.9e−09	1.5e+01	1.3e−06	7.1e+00	9	33	33	8
lukšan-vlček5.2	20	3.5e+01	7.8e+03	6.1e−03	6.9e+03	11	82	82	10
lukšan-vlček5.3	20	5.6e−06	2.2e+03	6.4e−04	1.5e+03	8	49	49	7
lukšan-vlček5.4	20	1.7e+00	2.4e+03	4.4e−03	5.7e+03	29	223	223	28
methanb8	31	1.4e−05	5.2e−01	1.7e−04	2.9e+02	11	174	174	10
methanl8	31	1.7e−02	2.2e+03	2.2e−02	2.8e+04	66	1714	1714	65
mgh19	11	4.4e−02	1.6e+01	1.0e−05	2.2e+01	66	618	618	65
mgh21	20	1.8e−11	1.2e+02	2.7e−06	3.7e+02	33	112	112	32
mgh22	20	4.0e−06	5.4e+02	3.5e−04	5.1e+02	9	44	44	8
mgh25	10	3.7e−06	1.1e+06	5.3e−02	2.2e+06	9	25	25	8
mgh26	10	1.4e−05	3.5e−03	3.4e−07	5.0e−02	34	212	212	33
mgh27	10	3.1e−07	1.4e+02	7.8e−05	1.7e+02	5	14	14	4
mgh28	10	4.8e−16	3.9e−04	6.2e−09	2.0e−02	3	25	25	2
mgh29	10	2.3e−17	3.2e−02	7.0e−09	3.1e−01	4	14	14	3
mgh30	10	7.1e−12	1.1e+01	1.8e−05	2.5e+01	6	33	33	5
mgh31	10	1.3e−12	1.8e+02	1.0e−05	4.1e+02	7	27	27	6
mgh32	10	5.0e+00	2.5e+01	2.5e−16	6.3e+00	2	4	4	1
mgh33	10	2.3e+00	4.3e+06	1.1e−09	3.1e+06	2	4	4	1
mgh34	10	3.1e+00	2.0e+06	6.0e−10	1.6e+06	2	4	4	1
mgh35	10	3.3e−03	1.7e−02	1.1e−06	6.7e−01	36	267	267	35
morebvne	10	1.9e−15	8.0e−03	1.6e−08	1.2e−01	4	35	35	3

Continued on next page

Table 6 — continued from previous page

Model	nvar	$f(x)$	$f(x_0)$	$\|g(x)\|$	$\|g(x_0)\|$	$\#F$	$\#Ju$	$\#J^Tv$	$\#it$
msqrta	1024	2.3e−09	4.0e+03	1.7e−05	1.7e+02	36	7492	7492	35
msqrtb	1024	8.0e−11	4.0e+03	4.3e−06	1.7e+02	31	5290	5290	30
nystrom5	18	5.0e−09	9.1e−01	7.2e−07	3.1e+00	39	630	630	38
nzf1	13	1.8e−09	5.0e+03	2.3e−04	9.3e+02	7	32	32	6
osborne2	11	2.0e−02	1.6e+00	2.8e−06	3.2e+00	29	246	246	28
oscigrne	100000	4.1e−04	3.1e+08	1.7e+02	1.1e+09	8	44	44	7
oscipane*	10	5.0e−01	5.0e−01	1.0e−05	5.0e−01	10001	29658	29658	10000
penlt1ne	10	1.1e−06	7.4e+04	1.5e−03	1.5e+04	8	22	22	7
porous1	5184	9.4e−08	5.0e+06	1.4e+00	1.3e+07	10	4469	4469	9
porous2	5184	3.8e−07	5.4e+06	2.8e+00	1.4e+07	10	4378	4378	9
semicn2u	5002	1.8e−04	9.8e+03	3.0e−04	3.4e+02	368	15488	15488	367
spin	1327	9.2e−04	6.7e+04	1.0e−02	2.2e+04	13	49	49	12
spin2	102	1.2e−08	1.6e+04	4.3e−03	4.4e+03	7	25	25	6
spmsqrt	4999	1.6e−13	2.1e+03	3.3e−07	3.9e+01	12	291	291	11
vandaniums	22	4.8e+00	7.5e+00	1.6e−06	1.2e+00	7	21	21	6
vardimne	10	3.7e−06	1.1e+06	5.3e−02	2.2e+06	9	25	25	8
watsonne	12	4.4e−08	1.5e+01	3.5e−05	4.3e+01	8	41	41	7
woodsne	4000	1.9e+01	1.5e+08	5.0e+00	6.4e+06	32	149	149	31
yatp1ne	2600	1.2e−06	2.6e+07	2.2e−03	4.4e+05	34	107	107	33
yatp1sq	123200	1.6e−01	1.3e+09	2.3e+00	8.1e+06	26	81	81	25
yatp2sq	123200	2.3e−08	3.8e+09	1.1e−02	3.8e+05	35	102	102	34

Acknowledgments. The authors express their gratitude to Michael Saunders for comments on an earlier draft of the present research, to Åke Björck for providing a copy of Björck (1979), and to two anonymous referees for comments that greatly improved this paper.

REFERENCES

Å. Björck (1979), *Use of conjugate gradients for solving linear least squares problems*, in Conjugate Gradient Methods and Similar Techniques, I. S. Duff, ed., AERE, Harwell, UK, pp. 49–71.

Å. Björck and M. A. Saunders (2017), *Krylov Subspace Algorithms for Overdetermined and Underdetermined Linear Systems*, Technical report, Stanford University, Stanford, CA.

Å. Björck, T. Elfving, and Z. Strakoš (1998), *Stability of conjugate gradient and Lanczos methods for linear least squares problems*, SIAM J. Matrix Anal. Appl., 19, pp. 720–736, DOI: 10.1137/S089547989631202X.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint (2000), *Trust-Region Methods*, MOS-SIAM Ser. Optim. 1, SIAM, Philadelphia, PA, DOI: 10.1137/1.9780898719857.

R. S. Dembo and T. Steihaug (1983), *Truncated-Newton algorithms for large-scale unconstrained optimization*, Math. Program., 26, pp. 190–212.

J. E. Dennis, Jr., and J. J. Moré (1977), *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19, pp. 46–89.

D. C.-L. Fong (2011), *Minimum-Residual Methods for Sparse Least-Squares Using Golub–Kahan Bidiagonalization*, Ph.D. thesis, Stanford University, Stanford, CA.

D. C.-L. Fong and M. A. Saunders (2011), *LSMR: An iterative algorithm for sparse least-squares problems*, SIAM J. Sci. Comput., 33, pp. 2950–2971, DOI: 10.1137/10079687X.

D. C.-L. Fong and M. A. Saunders (2012), *CG versus MINRES: An empirical comparison*, SQU J. Sci., 17, pp. 44–62.

R. Fourer, C. Maheshwari, A. Neumaier, D. Orban, and H. Schichl (2010), *Convexity and concavity detection in computational graphs*, INFORMS J. Comput., 22, pp. 26–43, DOI: 10.1287/ijoc.1090.0321.

G. H. Golub and W. Kahan (1965), *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2, pp. 205–224, DOI: 10.1137/0702016.

N. I. M. Gould, D. Orban, and Ph. L. Toint (2015), *CUTEst: A Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization*, Comput. Optim. Appl., 60, pp. 545–557, DOI: 10.1007/s10589-014-9687-3.

M. R. Hestenes and E. Stiefel (1952), *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand. (U.S.), 49, pp. 409–436.

T. Kloek (2012), *Conjugate Gradients and Conjugate Residuals Type Methods for Solving Least Squares Problems from Tomography*, Bachelor's thesis, TU Delft, Delft, Netherlands.

C. Lanczos (1950), *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand. (U.S.), 45, pp. 225–280.

K. Levenberg (1944), *A method for the solution of certain problems in least squares*, Q. J. Appl. Math., 2, pp. 164–168.

C.-J. Lin and J. J. Moré (1998), *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9, pp. 1100–1127, DOI: 10.1137/S1052623498345075.

D. G. Luenberger (1970), *The conjugate residual method for constrained minimization problems*, SIAM J. Numer. Anal., 7, pp. 390–398.

L. Lukšan, C. Matonoha, and J. Vlček (2010), *Modified CUTE Problems for Sparse Unconstrained Optimization*, Technical report 1081, Institute of Computer Science, Academy of Science of the Czech Republic, Prague.

D. Marquardt (1963), *An algorithm for least-squares estimation of nonlinear parameters*, J. Soc. Ind. Appl. Math., 11, pp. 431–441, DOI: 10.1137/0111030.

C. C. Paige and M. A. Saunders (1975), *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12, pp. 617–629, DOI: 10.1137/0712047.

C. C. Paige and M. A. Saunders (1982), *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8, pp. 43–71.

T. Steihaug (1983), *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20, pp. 626–637, DOI: 10.1137/0720042.

E. Stiefel (1955), *Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme*, Comment. Math. Helv., 29, pp. 157–179.

Y. Yuan (2000), *On the truncated conjugate gradient method*, Math. Program., 87, pp. 561–573, DOI: 10.1007/s101070050012.