Extended-Krylov-subspace methods for trust-region and norm-regularization subproblems

Hussam Al Daas and Nicholas I. M. Gould*

November 13, 2025

Abstract

We consider an effective new method for solving trust-region and norm-regularization problems that arise as subproblems in many optimization applications. We show that the solutions to such subproblems lie on a manifold of approximately very low rank as a function of their controlling parameters (trust-region radius or regularization weight). Based on this, we build a basis for this manifold using an efficient extended-Krylov-subspace iteration that involves a single matrix factorization. The problems within the subspace using such a basis may be solved at very low cost using effective high-order root-finding methods. This then provides an alternative to common methods using multiple factorizations or standard Krylov subspaces. We provide numerical results to illustrate the effectiveness of our `TREK`/`NREK` approach.

1 Introduction

Given an n by n symmetric (Hessian) matrix A, an n-vector b, a scalar "radius" $\Delta > 0$, and the Euclidean norm $\|\cdot\|$, the trust-region subproblem is to find

$$x_* = \arg\min_{x \in \mathbb{R}^n} \ \tfrac{1}{2} x^T A x - b^T x \ \text{ subject to } \ \|x\| \leq \Delta. \tag{1.1}$$

Such problems arise in trust-region methods for (non-convex) nonlinear optimization [5, 22]. A variety of methods have been proposed to solve them, some based on matrix factorization [15, 20], others on approximations from Krylov subspaces [4, 12] and some on reformulation as eigenproblems [1, 23, 24, 26]. It is well known [10, 20] that the solution $x(\sigma)$ satisfies the first-order optimality condition

$$(A + \sigma I)x(\sigma) = b, \tag{1.2}$$

*Computational Mathematics Theme, Scientific Computing Department, Rutherford Appleton Laboratory, Chilton, OX11 0QX, England (UK). Email: hussam.al-daas@stfc.ac.uk, nick.gould@stfc.ac.uk

where $\|x(\sigma)\| \leq \Delta$ and the scalar shift

$$\sigma \geq \sigma_{\min} \stackrel{\text{def}}{=} \max(0, -\lambda_1(A)) \tag{1.3}$$

satisfies the complementarity condition $\sigma(\|x(\sigma)\| - \Delta) = 0$; here $\{\lambda_1(A), \ldots, \lambda_n(A)\}$ are the eigenvalues of A, in increasing order, i.e., $\lambda_i(A) \leq \lambda_{i+1}(A)$, for $i = 1, \ldots, n-1$. That is, either, for positive semi-definite A,

$$Ax(0) = b \;\text{ with }\; \|x(0)\| \leq \Delta \tag{1.4}$$

or, with unrestricted A,

$$(A + \sigma I)x(\sigma) = b \;\text{ with }\; \|x(\sigma)\| = \Delta \;\text{ for some }\; \sigma \geq \sigma_{\min}. \tag{1.5}$$

We refer to the first possibility as the trivial—or interior—case, while the second is the general one; there is an additional probability-zero possibility, commonly called the hard case—this occurs if b is orthogonal to all the eigenvectors for the leftmost eigenvalue $\lambda_1(A)$ and the radius is too large—that we shall ignore for the time being. If we can rule out the trivial and hard cases, (1.5) then suggests an iteration in which, for a sequence of positive σ, $x(\sigma)$ is found by solving the linear system (1.2), and its suitability assessed by examining the size of the residual $\|x(\sigma)\| - \Delta$.

An interesting question is, therefore, what we can say about the manifold $\mathcal{S}(A, b) = \text{span}\{x(\sigma)\}$ as σ varies between $\sigma_{\min}$ and $+\infty$. There is no loss of generality, from a theoretical perspective, in considering a problem with a diagonal Hessian, since if A has the spectral factorization

$$A = U\Lambda U^T, \tag{1.6}$$

involving an orthogonal matrix U whose columns are its eigenvectors, and a diagonal matrix Λ whose entries are its eigenvalues, (1.1) is equivalent to

$$\arg\min_{y \in \mathbb{R}^k n} \; \tfrac{1}{2} y^T \Lambda y - d^T y \;\text{ subject to }\; \|y\| \leq \Delta \tag{1.7}$$

under the transformation

$$y = U^T x, \;\text{ where }\; d = U^T b.$$

Consider, then, problem (1.7) with $n = 10^4$, d a vector of ones, eigenvalues λ_i, $i = 1, \ldots, n$, in $(0, 1]$ that are either (i) equi-spaced at i/n, (ii) clustered at the lower end at i^2/n^2, (iii) clustered at the higher end at $1 - (i-1)^2/n^2$, or (iv) logarithmically-distributed in $(10^{-15}, 1]$ as $\exp(\log(10^{-15}) + (i-1)\beta)$, where $\beta = (\log(1) - \log(10^{-15})/(n-1)$, and $p = 10n$ evaluation points $\sigma = \sigma_j = j/n$ for $j = 1, \ldots, p$ in $(0, 10]$. In this case, the components of $y(\sigma)$ from the first-order optimality system

$$(\Lambda + \sigma I)y(\sigma) = d$$

for (1.7) are $y_i(\sigma) = 1/(\lambda_i + \sigma)$. If we approximate the solution manifold $\mathcal{S}(\Lambda, d)$ by the discrete set $\{y(\sigma_j), j = 1, \ldots, p\}$ at the evaluation points, and compute the singular value decomposition of the resulting n by p matrix S with entries

$$s_{i,j} = d_i/(\lambda_i + \sigma_j) \;\text{ for }\; i = 1, \ldots, n \;\text{ and }\; j = 1, \ldots, p, \tag{1.8}$$

we find 37 significant singular values (those that are larger than 10^{-15} of the largest) in case (i), and for cases (ii)–(iv) there are 38, 36 and 35 respectively. The precise singular values are given in Table 1.1.

(i): equi-spaced		(ii): clustered lower		(iii): clustered higher		(iv): logarithmically	
2.53e+04	1.32e-03	1.34e+05	6.33e-03	1.86e+04	4.84e-04	1.09e+06	4.64e-03
1.44e+04	4.99e-04	4.38e+04	2.48e-03	1.09e+04	1.74e-04	7.32e+04	1.81e-03
6.89e+03	1.87e-04	1.92e+04	9.70e-04	5.12e+03	6.23e-05	2.09e+04	7.04e-04
3.07e+03	7.00e-05	8.81e+03	3.78e-04	2.21e+03	2.22e-05	7.75e+03	2.72e-04
1.33e+03	2.60e-05	3.95e+03	1.46e-04	9.19e+02	7.85e-06	3.23e+03	1.05e-04
5.64e+02	9.64e-06	1.73e+03	5.65e-05	3.76e+02	2.77e-06	1.39e+03	4.04e-05
2.36e+02	3.55e-06	7.42e+02	2.17e-05	1.51e+02	9.70e-07	5.96e+02	1.55e-05
9.78e+01	1.30e-06	3.15e+02	8.33e-06	6.03e+01	3.39e-07	2.52e+02	5.90e-06
4.01e+01	4.76e-07	1.33e+02	3.18e-06	2.38e+01	1.18e-07	1.05e+02	2.24e-06
1.63e+01	1.73e-07	5.54e+01	1.21e-06	9.27e+00	4.07e-08	4.35e+01	8.51e-07
6.55e+00	6.29e-08	2.29e+01	4.59e-07	3.58e+00	1.40e-08	1.79e+01	3.21e-07
2.61e+00	2.27e-08	9.44e+00	1.74e-07	1.37e+00	4.80e-09	7.30e+00	1.21e-07
1.04e+00	8.18e-09	3.86e+00	6.54e-08	5.20e-01	1.64e-09	2.96e+00	4.54e-08
4.07e-01	2.93e-09	1.57e+00	2.46e-08	1.96e-01	5.57e-10	1.19e+00	1.70e-08
1.59e-01	1.05e-09	6.34e-01	9.19e-09	7.33e-02	1.89e-10	4.79e-01	6.35e-09
6.18e-02	3.73e-10	2.55e-01	3.43e-09	2.72e-02	6.37e-11	1.91e-01	2.36e-09
2.38e-02	1.32e-10	1.02e-01	1.28e-09	1.00e-02	2.14e-11	7.60e-02	
9.14e-03	4.69e-11	4.05e-02	4.74e-10	3.67e-03		3.01e-02	
3.48e-03		1.61e-02	1.76e-10	1.34e-03		1.18e-02	

Table 1.1: Singular values of dominant manifold subspace for four eigenvalue distributions

It is then remarkable that the manifold appears to be spanned by so few significant vectors, that is, it is approximately of very low rank. This turns out not to be a coincidence, and we shall exploit this throughout.

This paper is organised as follows. In Section 2, we explain why the solution manifold of interest is (approximately) low rank. We follow this in Section 3 by considering methods that build approximations to this manifold. A particularly appealing approach may be constructed from a so-called extended Krylov subspace, and we show how in Section 4. This then provides the basis for methods aimed at the trust-region subproblem (1.1), and we provide details for positive-definite A in Section 5. We follow this with a description of how we may extend this for general A in Section 6. Numerical experiments follow in Section 7. We show that the same ideas trivially extend to the regularized-norm problem in Section 8, and finally we conclude in Section 9.

2 Cauchy matrices and accurate low-rank approximation

The matrix (1.8) is an example of what is commonly known as a generalised Cauchy—or Cauchy-like—matrix [27]. To be more specific, suppose that we know the ordered, increasing eigenvalues $\lambda_i \in [\lambda_1, \lambda_n]$ of A (and thus Λ), and have picked a set of distinct shifts $\sigma_i \in [\sigma_{\min}^+, \sigma_{\max}]$, $i = 1, \ldots, p$, where

$$\sigma_{\min}^+ \stackrel{\text{def}}{=} \max(0, -\lambda_1 + \delta) \text{ and } \sigma_{\max} > \sigma_{\min}^+$$

for some $\delta > 0$. Then $\sigma_{\min} = \max(0, -\lambda_1) \leq \sigma_{\min}^+$, and moreover $\sigma_{\min}^+ \geq 0 > -\delta \geq -\lambda_1$ whenever $\lambda_1 \geq \delta$ and $\sigma_{\min}^+ \geq -\lambda_1 + \delta > -\lambda_1$ otherwise. Thus $-\sigma_{\min}^+ < \lambda_1$, indeed $\sigma_{\min}^+ + \lambda_1 \geq \delta$, and the intervals $[-\sigma_{\max}, -\sigma_{\min}^+]$ and $[\lambda_1, \lambda_n]$ are non-overlapping.

To investigate the rank of S, clearly rows i and j of S are linearly dependent if $\lambda_i = \lambda_j$, and thus $\text{rank}(S) = \text{rank}(\bar{S})$, where $\bar{S}$ comprises the rows of S in which each set of such dependencies is replaced by a single representative.

For any m by n matrix X with decreasing singular values $\sigma_1(x) \geq \sigma_2(X) \geq \ldots \geq \sigma_k(X)$ for increasing k, the ϵ-rank, $\text{rank}_\epsilon(X)$, of X is the smallest integer k for which $\sigma_{k+1}(X) \leq \epsilon\sigma_1(X)$. Then Beckermann and Townsend [2, §4.2] show that[1]

$$\text{rank}_\epsilon(\bar{S}) \leq \left\lceil \log(16\gamma) \log(4/\epsilon)/\pi^2 \right\rceil,$$

involving the cross-ratio

$$\gamma = \left| \frac{(\sigma_{\max} + \lambda_1)(\sigma_{\min}^+ + \lambda_n)}{(\sigma_{\max} + \lambda_n)(\sigma_{\min}^+ + \lambda_1)} \right|,$$

because we have removed repeated eigenvalues, the shifts are distinct and the two sets are non-overlapping. Notice that the cross ratio is bounded from above as $\sigma_{\min}^+ + \lambda_1 \geq \delta > 0$, and approaches $(\sigma_{\min}^+ + \lambda_n)/(\sigma_{\min}^+ + \lambda_1)$ as $\sigma_{\max} \to \infty$; for positive definite A, the latter is simply the spectral condition number. In particular, the log dependence on both the cross ratio and the precision ϵ are significant in keeping the bound small, and its dimension independence is also noteworthy. For the examples illustrated in Table 1.1, this bound is 44, a very reasonable (upper-bound) approximation of the computed values.

3 Manifold approximation from Krylov subspaces

Now that we know that the solution we seek to (1.1) *for any* Δ lies in an approximately low-dimensional subspace, our task is to identify a suitable (for convenience, orthogonal) n by ℓ basis matrix V_ℓ for this subspace. Once we do so, we may recover an approximate

[1] Stronger bounds are possible [2] in terms of the so-called Grötzsch ring function, and other bounds are also known, e.g. [16].

solution $x_\ell = V_\ell y_\ell$, where

$$y_\ell = \underset{y\in\mathbb{R}^k}{\arg\min}\ \ \tfrac{1}{2} y^T P_\ell y - b_\ell^T y \ \text{ subject to } \ \|y\| \le \Delta. \tag{3.1}$$

with $P_\ell = V_\ell^T A V_\ell$ and $b_\ell = V_\ell^T b$. The fact that we expect ℓ to be small provides an opportunity to use methods that would be inappropriate has it been larger.

An obvious approach might be to simply select a set of $\sigma_i \ge \sigma_{\min}^+$ for $i = 1, \ldots, \ell$, compute the resulting solutions $x(\sigma_i)$ and build an orthogonal basis of the set $\{x(\sigma_i)\}$ by, for example, the modified Gram-Schmidt process. This has two fundamental issues. Firstly, it requires that we solve ℓ shifted linear systems, and this is not without cost. Secondly, it is not known *a priori* how big ℓ should be. The latter may be addressed as follows. Suppose that y_ℓ is the solution to (3.1) with $\|y_\ell\| = \Delta$. If so, there is a shift $\sigma_\ell \ge 0$, just as in (1.2), for which

$$0 = (P_\ell + \sigma_\ell I) y_\ell - b_\ell = V_\ell^T \left[(A + \sigma_\ell I) x_\ell - b \right] \tag{3.2}$$

where $x_\ell = V_\ell y_\ell$ and $\|x_\ell\| = \Delta$, since V_ℓ has orthogonal columns (i.e., $V_\ell^T V_\ell = I$). As the solution we seek satisfies $(A + \sigma I)x = b$, and as V_ℓ will be an increasingly better approximation to the low-dimensional subspace that contains the required solution, it suffices to monitor the residual $r_\ell = (A + \sigma_\ell I) x_\ell - b$ and to stop when $\|r_\ell\|$ is sufficiently small. This, however, does not avoid the cost of multiple factorizations; rival methods, e.g., [15, 20], also use factorizations, but experience suggests that relatively few often suffice. Another approach for constructing a good approximation of the manifold is to use a method based on a rational Krylov subspace [25, 6]. However, this will still require solving ℓ shifted linear systems.

The alternative is to try to build a good approximation to the manifold by other means. We shall resort to methods based on the extended Krylov subspace (see Section 4).

To justify this, first note that given a shift σ, (1.2) and (1.6) give that

$$x(\sigma) = (A + \sigma I)^{-1} b = U y(\sigma),$$

where

$$y_i(\sigma) = \frac{d_i}{\lambda_i + \sigma}, \quad i = 1, \ldots, n, \quad \text{and} \quad d = U^T b.$$

Consider two cases and first suppose that $|\sigma| < |\lambda_i|$, for $i = 1, \ldots, n$. Then using a Laurent series [28], we have

$$y_i(\sigma) = \frac{d_i}{\lambda_i} \sum_{j=0}^{\infty} \left(\frac{-\sigma}{\lambda_i} \right)^j.$$

The smaller the ratio $|\sigma/\lambda_i|$ the faster the sequence $(-\sigma/\lambda_i)^j$ reaches machine precision and few terms would be sufficient to approximate $y_i(\sigma)$. In other words, $y(\sigma)$ can be very well approximated using a polynomial with a moderate degree in Λ^{-1}, and hence $x(\sigma)$ can be well approximated using a polynomial with a moderate degree in A^{-1} applied to b. In the other case, when $|\sigma| > |\lambda_i|$, for $i = 1, \ldots, n$, a similar argument using Taylor series shows that $x(\sigma)$ can be well approximated using a polynomial with moderate degree in A applied to b.

An extended Krylov subspace is by definition the space that constructs the sum space of both Krylov subspaces $\mathcal{K}(A, b) = \text{span}\{A^i b\}_{i\geq 0}$ and $\mathcal{K}(A^{-1}, b) = \text{span}\{A^{-i} b\}_{i\geq 0}$, and constructs the aforementioned polynomials required to handle the two ends of the behaviour spectrum. Hence, these observations provide some intuition as to why we choose an extended Krylov method (see Section 4) to solve the trust-region subproblem. Similar arguments have been made by Druskin and Knizhnerman [8] for different applications.

For a better understanding of the convergence of the extended-Krylov-subspace method when solving shifted linear systems with a symmetric positive definite matrix whose spectrum is contained in $[\lambda_1, \lambda_n]$ and shifts lying outside $(-\lambda_n - \delta_n, -\lambda_1 + \delta_1)$ for $\delta_1, \delta_n > 0$, we provide a convergence analysis in Appendix A that is heavily based on that by Knizhnerman and Simoncini [18]. This shows that there will be an element $x_i(\sigma)$ of the extended Krylov subspace $\mathcal{K}_i(A, b)$, defined in (4.1), for which

$$\|x_i(\sigma) - x(\sigma)\| \leq c\mu^{2i}$$

for some constants c and $\mu < 1$ and all $i \geq 0$.

4 The EKS algorithm

Our aim is thus to construct efficiently an orthogonal basis V_{2m+1} of the evolving extended Krylov subspaces

$$\mathcal{K}_{2m}(A, b) = \text{span}\{b, A^{-1}b, Ab, A^{-2}b, A^2 b, \ldots, A^{-m}b\} \tag{4.1}$$

and

$$\mathcal{K}_{2m+1}(A, b) = \mathcal{K}_{2m}(A, b) + \text{span}\{A^m b\}.$$

Fortunately, much of the groundwork is already in place. We suppose, for the time being, that A is positive definite. Then Jagels and Reichel[2] [17] have shown how to build such a basis if the order[3] in which the components $A^{-m}b$ and $A^m b$ arise is interchanged, that is if instead

$$\mathcal{K}'_{2m}(A, b) = \text{span}\{b, Ab, A^{-1}b, A^2 b, A^{-2}b, \ldots, A^m b\}$$

and

$$\mathcal{K}_{2m+1}(A, b) = \mathcal{K}'_{2m}(A, b) + \text{span}\{A^{-m}b\}.$$

[2]They also propose a second, more expensive algorithm that copes with indefinite A.

[3]We shall say why our order is more convenient later.

Using the same logic as in [17, Alg.2.1], that applies to $\mathcal{K}'_{2m}(A,b)$, it is straightforward to derive the following algorithm, Algorithm 4.1 on this page, that instead uses the order $\mathcal{K}_{2m}(A,b)$.

Algorithm 4.1: The EKS algorithm to find an orthogonal basis of $\mathcal{K}_{2m+1}(A,b)$

Input: symmetric $A \in \mathrm{IR}^{n\times n}$, $b \in \mathrm{IR}^n$ and subspace dimension bound $m \geq 1$.
Output: orthonormal $V_{2m+1} =$
$(v_0 : v_{-1} : v_1 : \cdots : v_{k-1} : v_{-k} : v_k : v_{-k-1} : v_{k+1} : \cdots : v_{-m} : v_m) \in \mathrm{IR}^{n\times(2m+1)}$.
Algorithm:

$\delta_0 := \|b\|$;
if $\delta_0 = 0$ **stop**, subspace completed;
$v_0 := b/\delta_0$, $u := A^{-1}v_0$;
$\beta_0 := u^T v_0$, $u \leftarrow u - \beta_0 v_0$;
$\delta_{-1} := \|u\|$;
for $k = 1, 2, \ldots, m$ **do**
  if $\delta_{-k} = 0$ **stop**, subspace completed;
  $v_{-k} := u/\delta_{-k}$, $u := Av_{-k}$;
  $\alpha_{k-1} := u^T v_{k-1}$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$;
  $\alpha_{-k} := u^T v_{-k}$, $u \leftarrow u - \alpha_{-k}v_{-k}$;
  $\delta_k := \|u\|$;
  if $\delta_k = 0$ **stop**, subspace completed;
  $v_k := u/\delta_k$, $u := A^{-1}v_k$;
  $\beta_{-k} := u^T v_{-k}$, $u \leftarrow u - \beta_{-k}v_{-k}$;
  $\beta_k := u^T v_k$, $u \leftarrow u - \beta_k v_k$;
  if $k < m$ **then**
    $\delta_{-k-1} := \|u\|$;

Thus

$$\delta_0 v_0 = b$$
$$\delta_{-1}v_{-1} = A^{-1}v_0 - \beta_0 v_0 \tag{4.2}$$
$$\delta_k v_k = Av_{-k} - \alpha_{-k}v_{-k} - \alpha_{k-1}v_{k-1} \quad \text{for } k = 1, \ldots, m \tag{4.3}$$
$$\delta_{-k-1}v_{-k-1} = A^{-1}v_k - \beta_{-k}v_{-k} - \beta_k v_k \quad \text{for } k = 1, \ldots, m-1 \tag{4.4}$$

Each iteration of the main loop requires a solve and a product involving A, the initialization requires a single solve. The remaining operations simply involve scalar products and vector sums. The decision whether and when to stop in the main loop is the focus of Sections 5 and 6, but observe that the extended Krylov subspace has definitely been completed if ever $\delta_i = 0$.

We consider the symmetric $\mathrm{IR}^{(2m+1)\times(2m+1)}$ "projected" matrix

$$P_{2m+1} = V_{2m+1}^T A V_{2m+1},$$

whose i, j-entry we denote as $p_{i,j}$, as well as its leading $2k$ and $2k+1$ sub-blocks

$$P_{2k} = V_{2k}^T A V_{2k} \text{ and } P_{2k+1} = V_{2k+1}^T A V_{2k+1},$$

where

$$V_{2k} = (v_0 : v_{-1} : v_1 : \cdots : v_{k-1} : v_{-k}) \text{ and } V_{2k+1} = (V_{2k} : v_k),$$

that play the roles of A_k in (3.1). It follows from (4.3) that

$$Av_{-k} = \alpha_{k-1} v_{k-1} + \alpha_{-k} v_{-k} + \delta_k v_k \text{ for } k = 1, \ldots, m \tag{4.5}$$

and the orthogonality of the v_i then gives

$$p_{2k-1,2k} = v_{k-1}^T A v_{-k} = \alpha_{k-1}, \;\; p_{2k,2k} = v_{-k}^T A v_{-k} = \alpha_{-k}, \;\; p_{2k+1,2k} = v_k^T A v_{-k} = \delta_k \tag{4.6}$$

and

$$p_{i,2k} = v_i^T A v_{-k} = 0 \text{ for } i \neq 2k-1, -2k \text{ and } 2k+1.$$

By symmetry of P_{2m+1}, we also have

$$p_{2k,2k-1} = \alpha_{k-1}, \;\; p_{2k,2k} = \alpha_{-k}, \;\; p_{2k,2k+1} = \delta_k \text{ and } p_{2k,i} = 0 \text{ for the remaining } i.$$

Thus, from (4.5) and (4.6), we have

$$Av_{-k} = p_{2k,2k-1} v_{k-1} + p_{2k,2k} v_{-k} + p_{2k,2k+1} v_k \text{ for } k = 1, \ldots, m, \tag{4.7}$$

where formally any $p_{2k,i} = 0$ if $i \leq 0$. Similarly, (4.4) gives

$$A^{-1} v_k = \beta_{-k} v_{-k} + \beta_k v_k + \delta_{-k-1} v_{-k-1} \text{ for } k = 1, \ldots, m-1$$

and thus that

$$\begin{aligned} \beta_k A v_k &= v_k - \beta_{-k} A v_{-k} - \delta_{-k-1} A v_{-k-1} \\ &= -\beta_{-k} \alpha_{k-1} v_{k-1} - \beta_{-k} \alpha_{-k} v_{-k} + (1 - \beta_{-k} \delta_k - \delta_{-k-1} \alpha_k) v_k \\ &\quad - \delta_{-k-1} \alpha_{-k-1} v_{-k-1} - \delta_{-k-1} \delta_{k+1} v_{k+1} \end{aligned} \tag{4.8}$$

using (4.5). Again, the orthogonality of the v_i and symmetry implies that[4]

$$\begin{aligned} p_{2k-1,2k+1} &= p_{2k+1,2k-1} = v_{k-1}^T A v_k = -\frac{\beta_{-k}\alpha_{k-1}}{\beta_k}, \\ p_{2k,2k+1} &= p_{2k+1,2k} = v_{-k}^T A v_k = -\frac{\beta_{-k}\alpha_{-k}}{\beta_k}, \\ p_{2k+1,2k+1} &= v_k^T A v_k = \frac{1-\beta_{-k}\delta_k - \delta_{-k-1}\alpha_k}{\beta_k}, \\ p_{2k+2,2k+1} &= p_{2k+1,2k+2} = v_{-k-1}^T A v_k = -\frac{\delta_{-k-1}\alpha_{-k-1}}{\beta_k}, \\ p_{2k+3,2k+1} &= p_{2k+1,2k+3} = v_{k+1}^T A v_k = -\frac{\delta_{-k-1}\delta_{k+1}}{\beta_k} \end{aligned} \tag{4.9}$$

and

$$p_{i,2k+1} = p_{2k+1,i} = v_i^T A v_k = 0 \text{ for } i \neq -k-1, k-1, k, -k \text{ and } k+1.$$

Thus, it follows from from (4.8) and (4.9) that

$$\begin{aligned} Av_k &= p_{2k-1,2k+1} v_{k-1} + p_{2k,2k+1} v_{-k} + p_{2k+1,2k+1} v_k + \\ & \quad p_{2k+2,2k+1} v_{-k-1} + p_{2k+3,2k+1} v_{k+1} \text{ for } k = 1, \ldots, m-1, \end{aligned} \tag{4.10}$$

where now formally any $p_{2k+1,i} = 0$ if $i \leq 0$. Moreover, note that

$$p_{2k,2k+1} = v_{-k}^T A v_k = \delta_k \text{ and } p_{2k+2,2k+1} = v_{-k-1}^T A v_k = \alpha_k$$

because of (4.6), and

$$-\frac{\beta_{-k}\alpha_{k-1}}{\beta_k} = -\frac{\delta_{-k}\delta_k}{\beta_{k-1}}$$

by comparing $p_{2k-1,2k+1}$ with $p_{2k-3,2k-1}$ for subsequent k. Finally, if follows from (4.2), (4.5) with $k = 1$ and the orthogonality of the v_i, that[5]

$$p_{1,1} = v_0^T A v_0 = \frac{1-\delta_{-1}\alpha_0}{\beta_0}, \quad p_{2,1} = v_{-1}^T A v_0 = -\frac{\delta_{-1}\alpha_{-1}}{\beta_0} \equiv \alpha_0, \quad p_{3,1} = v_1^T A v_0 = -\frac{\delta_{-1}\delta_1}{\beta_0}, \tag{4.11}$$

where the alternative definition of $p_{1,2}$ follows by symmetry and from (4.6) with $k = 0$.

[4]When $\delta_{-k-1} = 0$, the last three terms in (4.9) are $p_{2k+1,2k+1} = (1 - \beta_{-k}\delta_k)/\beta_k$ and $p_{2k+2,2k+1} = p_{2k+3,2k+1} = 0$, and thus do not require that we calculate α_k, α_{-k-1} and δ_{k+1}.

[5]When $\delta_{-1} = 0$, $p_{1,1} = 1/\beta_0$ and $p_{1,2} = p_{1,3} = 0$, and thus α_0, α_{-1} and δ_1 need not be calculated.

Thus P_{2k+3} is a symmetric, pentadiagonal matrix of the form

$$
P_{2k+3} = \begin{bmatrix}
p_{1,1} & \alpha_0 & p_{3,1} & 0 & 0 & 0 & 0 & \cdots \\
\alpha_0 & \alpha_{-1} & \delta_1 & 0 & 0 & 0 & 0 & \cdots \\
p_{3,1} & \delta_1 & p_{3,3} & \alpha_1 & p_{5,3} & 0 & 0 & \cdots \\
0 & 0 & \alpha_1 & \alpha_{-2} & \delta_2 & 0 & 0 & \cdots \\
0 & 0 & p_{5,3} & \delta_2 & p_{5,5} & \alpha_2 & p_{7,5} & \cdots \\
0 & 0 & 0 & 0 & \alpha_2 & \alpha_{-3} & \delta_3 & \cdots \\
0 & 0 & 0 & 0 & p_{7,5} & \delta_3 & p_{7,7} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
$$

$$
\equiv \begin{bmatrix}
p_{1,1} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \vdots & \ddots & \alpha_{k-2} & p_{2k-1,2k-3} & 0 & 0 & 0 & 0 \\
0 & \vdots & \alpha_{k-2} & \alpha_{-k+1} & \delta_{k-1} & 0 & 0 & 0 & 0 \\
0 & \vdots & p_{2k-1,2k-3} & \delta_{k-1} & p_{2k-1,2k-1} & \alpha_{k-1} & p_{2k-1,2k+1} & 0 & 0 \\
0 & \vdots & 0 & 0 & \alpha_{k-1} & \alpha_{-k} & \delta_k & 0 & 0 \\
0 & \vdots & 0 & 0 & p_{2k+1,2k-1} & \delta_k & p_{2k+1,2k+1} & \alpha_k & p_{2k+1,2k+3} \\
0 & \vdots & 0 & 0 & 0 & 0 & \alpha_k & \alpha_{-k-1} & \delta_{k+1} \\
0 & \vdots & 0 & 0 & 0 & 0 & p_{2k+3,2k+1} & \delta_{k+1} & p_{2k+3,2k+3}
\end{bmatrix}, \quad (4.12)
$$

where the entries $p_{2k+1,2k+1}$ and $p_{2k+1,2k-1}$ (etc) are as given above; the leading sub-matrices P_{2k} and P_{2k+1} are the entries in the two "hashed" boxes. In other words, we may grow the (lower-triangular parts of the) submatrices P_i of P_{2m+1}, $i = 1, \ldots, 2m+1$, simply by introducing, for each successive column, the entries

$$
\begin{aligned}
p_{2k+1,2k+1} &= \frac{1 - \beta_{-k}\delta_k - \delta_{-k-1}\alpha_k}{\beta_k}, \quad p_{2k+2,2k+1} = \alpha_k \text{ and} \\
p_{2k+3,2k+1} &= -\frac{\delta_{-k-1}\delta_{k+1}}{\beta_k} \quad \text{for odd } i = 2k+1
\end{aligned} \quad (4.13)
$$

and

$$
p_{2k,2k} = \alpha_{-k}, \quad p_{2k+1,2k} = \delta_k \text{ and } p_{2k+2,2k} = 0 \text{ for even } i = 2k, \quad (4.14)
$$

starting with the first column (4.11) when $i = 1$.

Trivially, it also follows from the orthogonality of V_k that

$$
V_{2m+1}^T b = \|b\| e_1, \quad (4.15)
$$

where e_1 is the first unit vector, and thus $b_k = \|b\| e_1$ in (3.1) for this choice of V_k.

We may also combine (4.7) and (4.10) to deduce that

$$
AV_{2k} = V_{2k}P_{2k} + p_{2k+1,2k-1}v_k e_{2k-1}^T + p_{2k+1,2k}v_k e_{2k}^T \quad (4.16)
$$

and

$$AV_{2k+1} = V_{2k+1}P_{2k+1} + (p_{2k+2,2k+1}v_{-k-1} + p_{2k+3,2k+1}v_{k+1})e_{2k+1}^T \tag{4.17}$$

These identities will shortly be used to develop suitable stopping rules for our algorithm.

It is perhaps worth mentioning that lines 4–5 and the first part of 8, 9–11 and the first part of 13 and 14–15, 17 and the first part of 8 in Algorithm 4.1 are mathematically equivalent to

$$\begin{aligned} \beta_0 &:= u^T v_0, \\ \delta_{-1} &:= \sqrt{\|u\|^2 - \beta_0^2}, \\ v_{-1} &:= [u - \beta_0 v_0]/\delta_{-1} \end{aligned}$$

$$\begin{aligned} \alpha_{k-1} &:= u^T v_{k-1}, \alpha_{-k} := u^T v_{-k}, \\ \delta_k &:= \sqrt{\|u\|^2 - \alpha_{k-1}^2 - \alpha_{-k}^2}, \\ v_k &:= [u - \alpha_{k-1}v_{k-1} - \alpha_{-k}v_{-k}]/\delta_k \end{aligned}$$

and

$$\begin{aligned} \beta_{-k} &:= u^T v_{-k}, \beta_k := u^T v_k^T, \\ \delta_{-k-1} &:= \sqrt{\|u\|^2 - \beta_{-k}^2 - \beta_k^2}, \\ v_{-k-1} &:= [u - \beta_{-k}v_{-k} - \beta_k v_k]/\delta_{-k-1}. \end{aligned}$$

However, numerical experience suggests that this form is unstable, in a manner akin to that of Gram-Schmidt orthogonalisation of a set of vectors and how such instability may be cured by its modified form [11, §5.2.7–5.2.8].

5 Solving the convex trust-region subproblem

Having determined an orthogonal basis V_ℓ of $\mathcal{K}_\ell(A, b)$ using the EKS algorithm, we now turn to the trust-region subproblem (3.1) within this subspace. As we have just seen, the Hessian $P_\ell = V_\ell^T A V_\ell$ is symmetric and sparse (pentadiagonal), while the first-order term b_ℓ is a scalar multiple $\|b\|$ of the ℓ-dimensional first unit vector. Significantly, we are expecting ℓ to be small, and thus the subproblem is low dimensional.

Thus we might apply off-the-shelf trust-region software such as GQT [20] or TRS [15] from GALAHAD [13]. However, the alternative we prefer is to proceed as in (1.6), that is to compute the spectral factorization

$$P_\ell = U_\ell \Lambda_\ell U_\ell^T \tag{5.1}$$

involving matrices of eigenvectors U_ℓ and (diagonal) eigenvalues Λ_ℓ, and to find $y_\ell = U_\ell z_\ell$, where

$$z_\ell = \arg\min_{z \in \mathbb{R}^\ell} \ \tfrac{1}{2} z^T \Lambda_\ell z - d_\ell^T z \ \text{ subject to } \ \|z\| \leq \Delta \tag{5.2}$$

and $d_\ell = U_\ell^T b_\ell$. Significantly (5.1) is inexpensive since ℓ is small, and the Hessian of (5.2) is diagonal. This then enables very fast and accurate solution using high-order, factorization-free iterative schemes, such as `TRS_solve_diagonal` [15] from GALAHAD.

Having computed the solution y_ℓ to (3.1) as above (or by any other means), together with the corresponding shift σ_ℓ for which both

$$(P_\ell + \sigma_\ell I)y_\ell = \|b\|e_1 \tag{5.3}$$

and $\|x_\ell\| = \Delta$ hold, according to (3.2), it remains to see how big is the residual

$$r_\ell \stackrel{\text{def}}{=} (A + \sigma_\ell I)x_\ell - b,$$

where $x_\ell = V_\ell y_\ell$. When $\ell = 2k+1$, we find that

$$\begin{aligned} Ax_{2k+1} &= AV_{2k+1}y_{2k+1} \\ &= V_{2k+1}P_{2k+1}y_{2k+1} + (p_{2k+2,2k+1}v_{-k-1} + p_{2k+3,2k+1}v_{k+1})e_{2k+1}^T y_{2k+1} \\ &= V_{2k+1}(\|b\|e_1 - \sigma_{2k+1}y_{2k+1}) + (p_{2k+2,2k+1}v_{-k-1} + p_{2k+3,2k+1}v_{k+1})e_{2k+1}^T y_{2k+1} \\ &= b - \sigma_{2k+1}x_{2k+1} + (p_{2k+2,2k+1}v_{-k-1} + p_{2k+3,2k+1}v_{k+1})e_{2k+1}^T y_{2k+1} \end{aligned}$$

using (4.17), (5.3), the orthogonality of the columns of V_{2k+1}, and the relationship $V_{2k+1}\|b\|e_1 = b$. Thus

$$r_{2k+1} = (p_{2k+2,2k+1}v_{-k-1} + p_{2k+3,2k+1}v_{k+1})e_{2k+1}^T y_{2k+1}$$

and hence

$$\|r_{2k+1}\| = |y_{2k+1,2k+1}|\sqrt{p_{2k+2,2k+1}^2 + p_{2k+3,2k+1}^2}, \tag{5.4}$$

where $y_{2k+1,2k}$ and $y_{2k+1,2k+1}$ are the $2k$-th and $2k+1$-st components of y_{2k+1}, using the orthogonality of v_{-k-1} and v_{k+1}. Thus the norm of the residual may be computed at almost no extra cost using data that is already available. Identical reasoning, when $\ell = 2k$ but now using (4.16), shows that

$$r_{2k} = (p_{2k+1,2k-1}e_{2k-1}^T y_{2k} + p_{2k+1,2k}e_{2k}^T y_{2k})v_k$$

and hence in this case

$$\|r_{2k}\| = |p_{2k+1,2k-1}y_{2k,2k-1} + p_{2k+1,2k}y_{2k,2k}|. \tag{5.5}$$

Note that in the exceptional case where $\delta_i = 0$, the corresponding $\|r_i\| = 0$, and thus termination will always occur at this step.

The `TREK` algorithm embeds solutions of (3.1) after every computation of v_{-k} and v_k in Algorithm 1; at these stages the matrices V_{2k} and V_{2k+1}, respectively, are complete, and we have the data required to solve the appropriate subproblems (3.1). Armed with such solutions, we terminate as soon as $\|r_{2k}\|$ from (5.5) or $\|r_{2k+1}\|$ from (5.4) is deemed small enough. One further important point is that after computing $u := A^{-1}v_0 = A^{-1}b/\|b\|$ before we start the main loop in the `EKS` algorithm, it is trivial to check for the interior case (1.4). Thus if we check and find $\|u\|\|b\| \leq \Delta$ in `TREK`, the solution is interior, and we exit[6] with $x_* = \|b\|u$. For completeness we state the full TREK algorithm (Algorithm 5.1) on the current page.

Algorithm 5.1: The TREK algorithm to solve the trust-region subproblem (1.1)

Input: symmetric $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $\Delta > 0$, a stopping tolerance $\epsilon > 0$ and an iteration bound $m \geq 1$.

Output: $x_* = \text{(approx)} \arg\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^TAx - b^Tx$ subject to $\|x\| \leq \Delta$.

Algorithm:

$x := A^{-1}b$;
if $\|x\| \leq \Delta$ **then**
  exit with the interior solution $x_* = x$, shift $\sigma_* = 0$ and $k_{\text{stop}} = 0$;
$\delta_0 := \|b\|$, $v_0 := b/\delta_0$, $u := x/\delta_0$;
$\beta_0 := u^Tv_0$, $u \leftarrow u - \beta_0 v_0$;
$\delta_{-1} := \|u\|$;
for $k = 1, 2, \dots, m$ **do**
  if $\delta_{-k} > 0$ **then**
    $v_{-k} := u/\delta_{-k}$, $u := Av_{-k}$;
    $\alpha_{k-1} := u^Tv_{k-1}$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$;
    $\alpha_{-k} := u^Tv_{-k}$, $u \leftarrow u - \alpha_{-k}v_{-k}$;
    $\delta_k := \|u\|$;
  if $k > 1$ **then**
    update P_{2k-1} from P_{2k-2} using (4.13) and form $b_{2k-1} = (b_{2k-2}^T, 0)^T$;
  else
    initialize P_1 from (4.11) and set $b_1 = \delta_0$;
  $y_{2k-1} := \arg\min \frac{1}{2}y^TP_{2k-1}y - b_{2k-1}^Ty$ s.t. $\|y\| \leq \Delta$ and its optimal shift σ_{2k-1};
  compute $\|r_{2k-1}\|$ from (5.4);
  if $\|r_{2k-1}\| \leq \epsilon$ **then**
    exit with the interior solution $x_* = V_{2k-1}y_{2k-1}$, shift $\sigma_* = \sigma_{2k-1}$ and $k_{\text{stop}} = k$;
  if $\delta_k > 0$ **then**
    $v_k := u/\delta_k$, $u := A^{-1}v_k$;
    $\beta_{-k} := u^Tv_{-k}$, $u \leftarrow u - \beta_{-k}v_{-k}$;
    $\beta_k := u^Tv_k$, $u \leftarrow u - \beta_k v_k$;
  if $k < m$ **then**
    if $\delta_k > 0$ **then**
      $\delta_{-k-1} := \|u\|$;
    update P_{2k} from P_{2k-1} using (4.14) and form $b_{2k} = (b_{2k-1}^T, 0)^T$;
    $y_{2k} := \arg\min \frac{1}{2}y^TP_{2k}y - b_{2k}^Ty$ s.t. $\|y\| \leq \Delta$ together with its optimal shift σ_{2k};
    compute $\|r_{2k}\|$ from (5.5);
    if $\|r_{2k}\| \leq \epsilon$ **then**
      exit with the interior solution $x_* = V_{2k}y_{2k}$, shift $\sigma_* = \sigma_{2k}$ and $k_{\text{stop}} = k$;

---

[6] Actually, we compute $x = A^{-1}b$, exit with $x_* = x$ if $\|x\| \leq \Delta$, and otherwise continue with $u = x/\|b\|$.

5.1 Resolves

The need to resolve a problem with the same A and b, but a smaller Δ is a common occurrence in trust-region methods, since such a mechanism is used to guarantee convergence when the current solution has proved unsatisfactory [5]. This is easy to accommodate within Algorithm 5.1; the data generated by the previous call to the algorithm is retained, and the '"for loop" re-entered with k taking the value k_{stop} that it last had. The motivation is simply that although the segment of the manifold $\mathcal{S}(A,b)$ we have observed may now have to expand slightly, it may well be that there is little actual difference; the data provided in Appendix C indicates that in only two of ninety six cases examined were any further extended Krylov iterations necessary, and thus that resolves are generally very inexpensive. Examining the bound (A.6) given in Proposition A.2 in Appendix A, and looking at Figure A.1, we see that the convergence rate of the extended Krylov subspace improves for σ beyond a certain point, which is consistent with this observation. Although other methods can also take advantage of resolves, all that we are aware of still require significant extra computation.

5.2 Other elliptical norms

Although the trust-region is often defined in terms of the Euclidean norm, it can be that this promotes some variables at the expense of others. To cope with this, it is sometimes more effective to pose the problem as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2} x^T A x - b^T x \;\; \text{subject to} \;\; \|Px\| \leq \Delta, \tag{5.6}$$

where P is an easily-invertible matrix—a diagonal P is often a simple but effective choice. On writing

$$x_{\text{P}} = Px, \tag{5.7}$$

the minimizer of (5.6) is formally $x_* = P^{-1} x_{\text{P}*}$, where

$$x_{\text{P}*} = \underset{x \in \mathbb{R}^n}{\arg\min} \;\; \tfrac{1}{2} {x_{\text{P}}}^T A_{\text{P}} x_{\text{P}} - {b_{\text{P}}}^T x_{\text{P}} \;\; \text{subject to} \;\; \|x_{\text{P}}\| \leq \Delta \tag{5.8}$$

and $A_{\text{P}} = P^{-T} A P^{-1}$ and $b_{\text{P}} = P^{-T} b$. Since the `TREK` algorithm described above merely relies on products with A_{P} and its inverse ${A_{\text{P}}}^{-1} = P A^{-1} P^T$, once again we merely need a factorization of A (to find $A^{-1}v$ for given v) so long as means of inverting P and its transpose are available. Commonly a positive-definite scaling matrix S that is (in some sense) close to A is known, and in this case a decomposition, for example a sparse Cholesky factorization, $S = P^T P$ gives the appropriate P. We provide details of the necessary changes to devise suitable algorithms to solve the resulting problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2} x^T A x - b^T x \;\; \text{subject to} \;\; \|x\|_S \leq \Delta, \tag{5.9}$$

where $\|x\|_S \stackrel{\text{def}}{=} \sqrt{x^T S x}$, in Appendix B. We illustrate one such algorithm here as Algorithm 5.2 on the following page. Note that here the generated vectors v_i now form an S-orthonormal basis for the extended Krylov subspace.

It is also worth noting that the extended Krylov subspace (4.1) generated by A_{P} and b_{P},

$$\mathcal{K}_{2m+1}(A_{\mathrm{P}}, b_{\mathrm{P}}) = \mathrm{span}\{b_{\mathrm{P}}, A_{\mathrm{P}}^{-1} b_{\mathrm{P}}, A_{\mathrm{P}} b_{\mathrm{P}}, \ldots, A_{\mathrm{P}}^{-m} b_{\mathrm{P}}, A_{\mathrm{P}}^{m} b_{\mathrm{P}}\}, \tag{5.10}$$

is

$$P^{-T} \mathrm{span}\{b, (SA^{-1})b, (AS^{-1})b, \ldots, (SA^{-m})b, (AS^{-1})^m b\},$$

and it is then possible to derive a suitable variant of Algorithm 4.1 using S and its inverse rather than P and P^{-1}.

Algorithm 5.2: The `TREK` algorithm to solve the trust-region subproblem (5.9)

Input: symmetric $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, symmetric, positive-definite $S \in \mathbb{R}^{n\times n}$, $\Delta > 0$, a stopping tolerance $\epsilon > 0$ and an iteration bound $m \geq 1$.

Output: $x_* = \text{(approx)} \arg\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^TAx - b^Tx$ subject to $\|x\|_S \leq \Delta$.

Algorithm:

$x := A^{-1}b$, $u := Sx$;
if $\sqrt{u^Tx} \leq \Delta$ **then**
  exit with the interior solution $x_* = x$, shift $\sigma_* = 0$ and $k_{\text{stop}} = 0$;
$w := S^{-1}b$, $\delta_0 := \sqrt{b^Tw}$;
$v_0 := w/\delta_0$, $q_0 := b/\delta_0$, $w := u/\delta_0$, $u := x/\delta_0$;
$\beta_0 := u^Tq_0$, $u \leftarrow u - \beta_0 v_0$;
$w = Su$, $\delta_{-1} := \sqrt{u^Tw}$;
for $k = 1, 2, \ldots, m$ **do**
  if $\delta_{-k} > 0$ **then**
    $v_{-k} := u/\delta_{-k}$, $q_{-k} := w/\delta_{-k}$;
    $w := Av_{-k}$, $u := S^{-1}w$;
    $\alpha_{-k} := w^Tv_{-k}$, $u \leftarrow u - \alpha_{-k}v_{-k}$, $w \leftarrow w - \alpha_{-k}q_{-k}$;
    $\alpha_{k-1} := w^Tv_{k-1}$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$, $w \leftarrow w - \alpha_{k-1}q_{k-1}$;
    $\delta_k := \sqrt{u^Tw}$;
  if $k > 1$ **then**
    update P_{2k-1} from P_{2k-2} using (4.13) and form $b_{2k-1} = (b_{2k-2}^T, 0)^T$;
  else
    initialize P_1 from (4.11) and set $b_1 = \delta_0$;
  $y_{2k-1} := \arg\min \frac{1}{2}y^TP_{2k-1}y - b_{2k-1}^Ty$ s.t. $\|y\| \leq \Delta$ and its optimal shift σ_{2k-1};
  compute $\|r_{2k-1}\|$ from (5.4);
  if $\|r_{2k-1}\| \leq \epsilon$ **then**
    exit with the interior solution $x_* = V_{2k-1}y_{2k-1}$, shift $\sigma_* = \sigma_{2k-1}$ and $k_{\text{stop}} = k$;
  if $\delta_k > 0$ **then**
    $v_k := u/\delta_k$, $q_k := w/\delta_k$;
    $u := A^{-1}q_k$, $w := Su$;
    $\beta_{-k} := w^Tv_{-k}$, $u \leftarrow u - \beta_{-k}v_{-k}$, $w \leftarrow w - \beta_{-k}q_{-k}$;
    $\beta_k := w^Tv_k$, $u \leftarrow u - \beta_k v_k$, $w \leftarrow w - \beta_k q_k$;
  if $k < m$ **then**
    if $\delta_k > 0$ **then**
      $\delta_{-k-1} := \sqrt{u^Tw}$;
    update P_{2k} from P_{2k-1} using (4.14) and form $b_{2k} = (b_{2k-1}^T, 0)^T$;
    $y_{2k} := \arg\min \frac{1}{2}y^TP_{2k}y - b_{2k}^Ty$ s.t. $\|y\| \leq \Delta$ together with its optimal shift σ_{2k};
    compute $\|r_{2k}\|$ from (5.5);
    if $\|r_{2k}\| \leq \epsilon$ **then**
      exit with the interior solution $x_* = V_{2k}y_{2k}$, shift $\sigma_* = \sigma_{2k}$ and $k_{\text{stop}} = k$;

6 Solving the general trust-region subproblem

For non-convex problems—indeed if A is not positive definite—it may be awkward (or even impossible if A is singular) to use an extended Krylov subspace involving products with A^{-1}. Fortunately, it is straightforward to deal with this possibility using the subspace generated by a diagonally-shifted matrix $A_{\sigma_{\rm S}} = A + \sigma_{\rm S} I$, for some suitable $\sigma_{\rm S} > 0$, and its inverse.

To see why, first note that the interior-solution case cannot occur if A is indefinite, and thus the solution we seek must lie on the trust-region boundary. That is, we must satisfy (1.5). We may rewrite this as

$$(A_{\sigma_{\rm S}} + \theta I) x_{\sigma_{\rm S}}(\theta) = b \;\text{ with }\; \|x_{\sigma_{\rm S}}(\theta)\| = \Delta, \tag{6.1}$$

where $x(\sigma) \equiv x_{\sigma_{\rm S}}(\theta)$ and

$$\theta = \sigma - \sigma_{\rm S} \geq \sigma_{\min} - \sigma_{\rm S} = \max(-\sigma_{\rm S}, -\lambda_1(A) - \sigma_{\rm S}).$$

Thus we seek the unique root of (6.1) but now over the interval $\theta \in [\sigma_{\min} - \sigma_{\rm S}, \infty)$. Given $\sigma_{\rm S}$, it is trivial to adapt any bracketing root-finding procedure to account for this extended interval.

We have in mind that the shift should chosen be so that $A_{\sigma_{\rm S}}$ is positive definite. The smallest shift for which $A_{\sigma_{\rm S}}$ is guaranteed to be positive semi-definite is $\sigma_{\min}$ from (1.3) so any value slightly larger than this is possible. But a far cheaper option that does not rely on knowing the spectrum of A, once A has been found not to be positive definite, is to use the value

$$\sigma_{\rm G} = -\min_{1\leq i\leq n} \left[a_{ii} - \sum_{1\leq j\neq i\leq n} |a_{i,j}| \right] + \epsilon_{\rm G}$$

from the Gershgorin theorem, requiring a single pass through the values of A; we have found that this with the heuristic value $\epsilon_{\rm G} = \sqrt{\epsilon_{\rm M}} \max_{1\leq i,j\leq n} |a_{i,j}|$ where $\epsilon_{\rm M}$ is the machine precision is effective.

As before, we generate the extended Krylov subspace $\mathcal{K}_\ell(A_{\sigma_{\rm S}}, b)$ using Algorithm 4.1. If the σ we seek lies to the right of $\sigma_{\rm S}$ (i.e., $\sigma \geq \sigma_{\rm S}$ and hence $\theta \geq 0$), the Beckermann-Townsend [2, §4.2] analysis outlined in Section 2 continues to hold, and we can expect that $x_{\sigma_{\rm S}}(\theta)$ will lie in an approximately low-dimensional subspace that will be uncovered by Algorithm 4.1.

It is also worth noting that although we have now generated $P_\ell = V_\ell^T A_{\sigma_{\rm S}} V_\ell$, we equally have

$$V_\ell^T A V_\ell = V_\ell^T A_{\sigma_{\rm S}} V_\ell - \sigma_{\rm S} V_\ell^T V_\ell = P_\ell - \sigma_{\rm S} I$$

since the columns of v_ℓ are orthonormal. Thus subspace trust-region subproblems of the form (3.1) can be formulated as

$$y_\ell = \underset{y \in \mathbb{R}^k}{\arg\min} \ \tfrac{1}{2} y^T (P_\ell - \sigma_S I) y - b_\ell^T y \ \text{ subject to } \ \|y\| \leq \Delta. \tag{6.2}$$

More particularly, if we rely on the spectral factorization (1.6), the diagonal subproblem (5.2) becomes

$$z_\ell = \underset{z \in \mathbb{R}^\ell}{\arg\min} \ \tfrac{1}{2} z^T (\Lambda_\ell - \sigma_S I) z - d_\ell^T z \ \text{ subject to } \ \|z\| \leq \Delta.$$

Finally, we return briefly to the thorny issue of the so-called hard case, that is when b is orthogonal to the subspace $\mathcal{E}_1$ of all the eigenvectors for the leftmost eigenvalue $\lambda_1(A) < 0$, and the radius is too large. In this case, the solution we seek is of the form[7]

$$x_+ + \theta y, \ \text{ where } \ x_+ = \lim_{\sigma \xrightarrow{+} -\lambda_1(A)} x(\sigma), \ \ y \in \mathcal{E}_1, \ \text{ and } \ \|x_+ + \theta y\| = \Delta,$$

when $\|x_+\| < \Delta$. No Krylov method based on A and b (extended or otherwise) will see the eigenspace $\mathcal{E}_1$ in the hard case in exact arithmetic, although rounding errors can gradually introduce it. Since we have never observed the hard case in practice for anything other than contrived examples, our only precaution is to add a tiny (pseudo-random) perturbation to b if requested. We certainly do this when $b = 0$.

6.1 Other elliptical norms

For the general problem defined in terms of an elliptical norm $\|x\|_S = \sqrt{x^T S x}$ for a given symmetric positive definite $S = P^T P$, a similar approach is possible. Transforming via (5.7) as in Subsection 5.2, the (non-interior) optimality condition (1.2) becomes

$$(A_P + \sigma I) x_P(\sigma) = b_P \ \text{ with } \ \|x_P(\sigma)\| = \Delta \tag{6.3}$$

for some

$$\sigma \geq \sigma_{P\min} \stackrel{\text{def}}{=} \max(0, -\lambda_1(A_P)),$$

or equivalently

$$(A + \sigma S) x(\sigma) = b \ \text{ with } \ \|x(\sigma)\|_S = \Delta;$$

note that $\lambda_1(A_P)$ is equivalently the leftmost eigenvalue $\lambda_1(A, S)$ of the symmetric matrix pencil $A + \lambda S$. The equivalent version of (6.1) is then

$$(A_{\sigma_S} + \theta S) x_{\sigma_S}(\theta) = b \ \text{ with } \ \|x_{\sigma_S}(\theta)\|_S = \Delta, \tag{6.4}$$

where

$$\theta = \sigma - \sigma_S \geq \sigma_{P\min} - \sigma_S = \max(-\sigma_S, -\lambda_1(A, S) - \sigma_S).$$

[7] Note that x_+ exists since b lies in the range of $A - \lambda_1(A) I$ as b is orthogonal to $\mathcal{E}_1$.

Although it is less obvious in general how to find a lower bound on $\lambda_1(A, S)$ in general, it is possible to use a variant of the inexpensive Gershgorin bound in the commonly-occurring case where S is strictly diagonally dominant (see [15, §3.3.2] for details). Our software packages make the strict diagonal dominance of S a prerequisite.

7 Numerical experiments

We now provide some evidence that our new approach is a useful alternative to common existing methods. We consider the set of 96 unconstrained test problems in the current release (2025-09-09) of the CUTEst optimization test examples [14] that have 1000 or more variables. For these, we generate the gradient and Hessian at the provided initial point, and use these for $-b$ and A respectively.

We compare our new method (Algorithm 5.1 implemented as TREK), using the initial (Gershgorin-based) shifting strategy outlined in Section 6, with the multi-factorization method [15] available as TRS within GALAHAD [13] and the iterative method GLTR [12] from the same library; GLTR only requires Hessian-vector products which can be an asset in some cases. Since all three methods can take advantage of information gathered when solving a problem with one radius and are now faced with another with identical data but a smaller radius, we consider each problem with radii, $\Delta = 1$ and then 0.5. TREK uses an iteration bound $m = 100$, but almost always[8] terminates with far-fewer k. Default, and roughly equivalent stopping rules were applied; TREK terminates as soon as (5.4) or (5.5) is smaller than 10^{-10}.

We perform our experiments on four cores of a PC with sixteen Intel Core i9-9900 CPU 3.10GHz processors, and thirty two Gbytes of memory. The codes are all from the GALAHAD library compiled with -Ofast optimization by gfortran, and use is made of OpenMP-tuned BLAS and LAPACK. Factorization is performed, when needed, by the HSL_MA57 sparse symmetric linear solver [9], enabled via GALAHAD's SLS package, and relies on the tuned BLAS for good performance. On some examples, particularly those whose Hessians are banded, further experiments not listed here revealed that LAPACK's PBTR band solver is a good alternative.

We summarize our findings in Figure 7.1 using the performance profile [7] in which relative times are ranked. The profile is built using the data given in Appendix C.

We did not expect any of the methods tested to be an overall winner, and indeed this is the case. Clearly, the factorization-free method GLTR is frequently very effective, particularly when factorization is a significant overhead. But we were encouraged to see that TREK performs well in other cases, and the requirement of a single factorization often proves decisive when compared to the multi-factorization approach used by TRS.[9]

[8] In four instances, m was too small, but increasing the value to 400 cured the problem.

[9] TRS also failed in three cases because the shifted matrix $A + \sigma I$ became too close to singularity to progress.

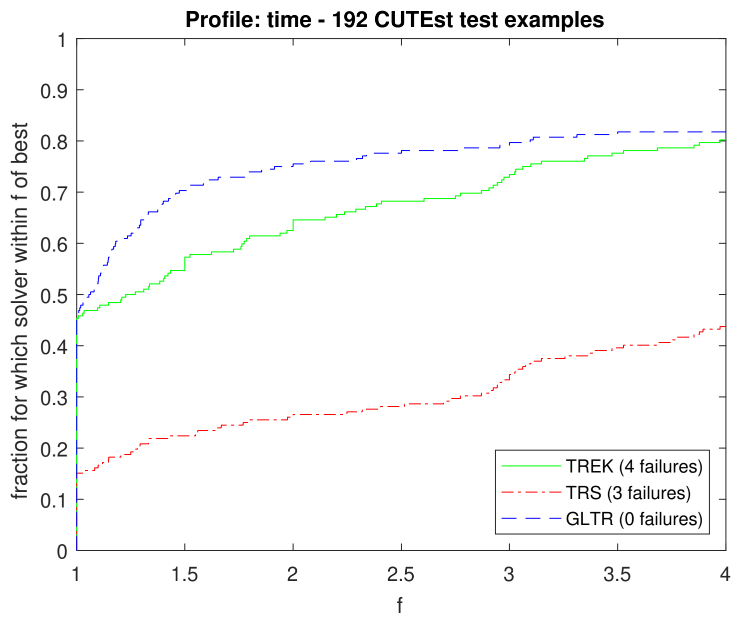

Figure 7.1: Performance profile comparing three trust-region subproblem solvers on larger CUTEst examples.

8 Regularization subproblems

Another important subproblem employed by methods for nonlinear optimization uses norm-regularization [3, 4, 21]. Again, given an n by n symmetric (Hessian) matrix A, an n-vector b and the Euclidean norm $\|\cdot\|$, but now a regularization weight $\rho > 0$ and a power $r \geq 2$, the norm-regularization subproblem is to find

$$x_* = \arg\min_{x \in \mathbb{R}^n} \; \tfrac{1}{2} x^T A x - b^T x + \frac{\rho}{r}\|x\|^r. \tag{8.1}$$

Once again, the solution $x(\sigma)$ satisfies the first-order optimality condition (1.2), but now

$$\sigma = \rho \|x(\sigma)\|^{r-2} \tag{8.2}$$

for some $\sigma \geq \sigma_{\min}$ [3, 21]. Thus the manifold of solutions is identical to that for the trust-region subproblem, but the one of interest satisfies (8.2) instead of (1.5); there is no interior-solution possibility in this case.

Since this is the case, it is trivial to recast the TREK algorithm for (8.1), the only differences are that the steps that calculate y_{2k} and y_{2k+1} and their shifts find instead

$$y_{2k} := \arg\min \tfrac{1}{2} y^T P_{2k} y - b_{2k}^T y + \frac{\rho}{r}\|y\|^r \;\text{ and its optimal shift } \sigma_{2k} \tag{8.3}$$

and

$$y_{2k+1} := \arg\min \tfrac{1}{2} y^T P_{2k+1} y - b_{2k+1}^T y + \frac{\rho}{r}\|y\|^r \text{ and its optimal shift } \sigma_{2k+1}, \tag{8.4}$$

and that the check for an interior solution before the main iteration loop is omitted. We refer to the resulting algorithm (and the GALAHAD package that implements it) as `NREK`. The subproblems (8.3) and (8.4) are solved just as those for the trust-region case by finding the spectral decompositions of P_{2k} and P_{2k+1}, and transforming so that the resulting (small) subproblem is diagonal. Thereafter the high-order, factorization-free diagonal regularization subproblem subroutine `RQS_solve_diagonal` [15] from GALAHAD is employed. Other elliptical norms may be handled just as described in Subsection 5.2.

9 Conclusions

We have provided an alternative to currently popular methods for solving quadratic trust-region and regularization problems. This lies somewhere between multiple factorization methods that were first popularised by Moré and Sorensen [20] and CG/Lanczos-based factorization-free methods such as `GLTR` [12]. The novel features are firstly that it is shown that the manifold of all possible solutions lies on an (approximately) very low-rank manifold, and secondly that by extending the Krylov subspace to include products with A^{-1} as well as A, this manifold may be computed very efficiently. This then adds a further tool to the arsenal of solvers that trust-region and cubic-regularization methods for general optimization methods rely on. Unsurprisingly, the new method is not always the best, but in some cases it is. It is also remarkable that empirically that the extended Krylov-subspace approach requires so few iterations to build the a good approximation of the solution manifold; examination of the results in Appendix C illustrates only a few instances where more than 20 iterations are required. We provide analysis to explain why this should be so, although this may well be tightened in future.

Software that implements the algorithms will be available very shortly[10] as part of the GALAHAD [13] fortran library, and there are interfaces to other popular languages such as C, Python and Julia, and tools such as Matlab. The codes are designed to solve not only a single instance, but a sequence in which the regularization is tightened, without recourse to expensive re-evaluations; the current subspace V_k is extended if it is insufficient as the regularization changes. The new packages will be rolled out as optional subproblem solvers in other GALAHAD packages in due course.

[10]The `TREK` algorithm with its documentation is now available as a beta release, and once C, Python, Julia and Matlab interfaces have been finished, it will be simple to extend these to the norm regularisation case discussed in Section 8 as the `NREK` package.

References

[1] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda. Solving the trust-region subproblem by a generalized eigenvalue problem. *SIAM Journal on Optimization*, 27(1):269–291, 2017.

[2] B. Beckermann and A. Townsend. On the singular values of matrices with displacement structure. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1227–1248, 2017.

[3] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming, Series A*, 127(2):245–295, 2011.

[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation, and Perspectives.* SIAM, Philadelphia, 2022.

[5] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods.* SIAM, Philadelphia, 2000.

[6] H. Al Daas and D. Palitta. Minimal residual rational Krylov subspace method for sequences of shifted linear systems, 2025.

[7] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[8] V. Druskin and L. Knizhnerman. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM Journal on Matrix Analysis and Applications*, 19(3):755–771, 1998.

[9] I. S. Duff and J. K. Reid. Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, 22(2):227–257, 1996.

[10] D. M. Gay. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.

[11] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[12] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[13] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, 29(4):353–372, 2003.

[14] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst : a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015.

[15] N. I. M. Gould, D. P. Robinson, and H. S. Thorne. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computation*, 2(1):21–57, 2010.

[16] L. Grasedyck. Existence of a low rank or H matrix approximant to the solution of a Sylvester equation. *Numerical Linear Algebra with Applications*, 11(4):371–389, 2004.

[17] C. Jagels and L. Reichel. The extended Krylov subspace method and orthogonal Laurent polynomials. *Linear Algebra and its Applications*, 431(3-4):441–458, 2009.

[18] L. Knizhnerman and V. Simoncini. Convergence analysis of the extended Krylov subspace method for the Lyapunov equation. *Numerische Mathematik*, 118(3):567–586, 2011.

[19] G. S. Kocharyan. On approximation by rational functions in the complex domain (in Russian). *Izvestiya Akademii Nauk Armyanskoi SSR, Seriya Fiziko-Matematicheskikh Nauk*, 11(4):53–77, 1958.

[20] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[21] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108(1):177–205, 2006.

[22] J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

[23] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming, Series B*, 77(2):273–299, 1997.

[24] M. Rojas, S. A. Santos, and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM Journal on Optimization*, 11(3):611–646, 2001.

[25] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 1984.

[26] S. A. Santos and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. Technical Report TR95-20, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 1995.

[27] Cauchy matrix—Wikipedia, the free encyclopedia, 2024. [Online; accessed 19-November 2024].

[28] Laurent series—Wikipedia, the free encyclopedia, 2024. [Online; accessed 29-December 2024].

A Convergence analysis

The first j-th matrix products of the EKS algorithm (Algorithm 4.1) generate an orthogonal basis for the extended Krylov subspace $\mathcal{K}_j(A, b)$. A natural question is then, how closely do the elements of this subspace lie to components of the manifold $x(\sigma)$. That is to say, for a given σ, is there an entry in $\mathcal{K}_j(A, b)$ close to $x(\sigma)$, and if so, how close?

Consider the symmetric positive-definite matrix A with smallest and largest eigenvalues $\lambda_1 = \lambda_1(A)$ and $\lambda_n = \lambda_n(A)$, and consider the interval $\mathcal{W} = [\lambda_1, \lambda_n] \subset \mathbb{R}_+^*$. In this section, we repeat for completeness, with minor variations, the theory developed by Knizhnerman and Simoncini [18] to prove that the convergence of the extended-Krylov-subspace method to build an approximate basis for the solution manifold $\{(A + \sigma I)^{-1} b\}$, for all $\sigma \notin -\mathcal{W} - (-\delta_1, \delta_n) = -(\lambda_1 - \delta_1, \lambda_n + \delta_n)$, and $\delta_1, \delta_n > 0$, is linear at worst. The proof relies heavily on 19th to mid 20th century complex analysis, and we refer the reader to [18] for the details; the technical buildup may appear somewhat fearsome to the uninitiated.

Let D denote the closed unit disk, let $\Psi : \bar{\mathbb{C}} \setminus D \to \bar{\mathbb{C}} \setminus W$ be the Riemann mapping for $\mathcal{W}$ that preserves infinity and has positive derivatives there and let $\Phi = \Psi^{-1}$ be its inverse. That is, $\Psi = \Phi^{-1}$ and both Φ and Ψ are holomorphic on their domains. For the specific set $\mathcal{W}$, we have that $\Phi(t)$ and $\Psi(t)$ are real $t \notin \mathcal{W}$, and explicitly

$$\Psi(t) = c + \frac{r}{2}\left(t + \frac{1}{t}\right) \text{ and}$$

$$\Phi(t) = \begin{cases} \dfrac{t-c}{r} + \sqrt{\left(\dfrac{t-c}{r}\right)^2 - 1} & \text{for } t \leq -\lambda_n \\ \left(\dfrac{t-c}{r} + \sqrt{\left(\dfrac{t-c}{r}\right)^2 - 1}\right)^{-1} & \text{for } t \geq -\lambda_1 \end{cases}$$

where $c = (\lambda_n + \lambda_1)/2$ and $r = (\lambda_n - \lambda_1)/2$. We then define the function (a finite Blaschke product)

$$B(w) = w\frac{1 - \overline{\Phi(0)}w}{\Phi(0) - w},$$

and note that this function satisfies $|B(w)| = 1$ for $|w| = 1$ and

$$B(w) < 1 \tag{A.1}$$

for $|w| < 1$ (see, [18]).

Armed with these definitions, let $\phi_{2\ell}$ and $\phi_{2\ell+1}$ be the (Takenaka–Malmquist) set of rational functions

$$\phi_{2\ell}(w) = B(w)^{\ell} \text{ and } \phi_{2\ell+1}(w) = -\frac{\sqrt{1 - |\Phi(0)|^{-2}}}{1 - \Phi(0)^{-1}w} wB(w)^{\ell}. \tag{A.2}$$

for $\ell \in \mathbb{N}$. For these, we have

$$\frac{1}{2\pi} \int_{|w|=1} \phi_p(w)\phi_m(w)|dw| = \delta_{m,p}$$

and

$$\max_{m\in\mathbb{N}} \max_{|w|=1} \phi_m(w) \leq c_{\phi}$$

for all $m, p \in \mathbb{N}$ and some finite constant c_{ϕ}.

Now consider the Faber-Dzhrbashyan rational functions

$$M_m(z) = \frac{1}{2\pi i} \int_{\Gamma_R} \frac{\phi_m(\Phi(\xi))}{\xi - z} d\xi \text{ for } z \notin \overline{G_R} \text{ and } m \in \mathbb{N},$$

where

$$\Gamma_R = \{z \in \mathbb{C} \setminus W : |\Phi(z)| = R\}, \quad G_R = W \cup \{z \in \mathbb{C} \setminus W : |\Phi(z)| < R\}$$

and $R > 1$. In particular, M_m is a Faber transformation of ϕ_m, and it then follows that

$$M_{2k}(z) = \frac{p_{2k}(z)}{z^k} \text{ and } M_{2k+1}(z) = \frac{p_{2k+1}(z)}{z^{k+1}} \tag{A.3}$$

where p_m is a polynomial of degree m. Furthermore,

$$|M_m(z)| \leq c_M \tag{A.4}$$

for some finite constant c_M for all m and all $z \in \mathcal{W}$.

Proposition A.1. *Given A, Φ, Ψ, ϕ_k, M_k and $\mathcal{W}$ as above, we have that*

$$(zI - A)^{-1} = \frac{1}{\Phi(z)\Psi'(\Phi(z))} \sum_{k=0}^{\infty} \phi_k(\Phi(z)^{-1}) M_k(A) \text{ for every } z \notin \mathcal{W} \tag{A.5}$$

Proof. In general, we have the identity [19]

$$\frac{\Psi'(w)}{\Psi(w)-u} = \frac{1}{w}\sum_{k=0}^{\infty}\overline{\phi_k\left(\frac{1}{\bar{w}}\right)}M_k(u).$$

The special (matrix-valued) case for which $u = A$ and $\Psi(w) = zI$ leads directly to (A.5). □

Proposition A.2. *Let $\sigma \in \mathbb{R}\setminus(-\lambda_n - \delta_n, -\lambda_1 + \delta_1)$. The EKS algorithm (Algorithm 4.1) used to approximate $x(\sigma) = (A + \sigma I)^{-1}b$ converges at worst linearly with a rate*

$$\mu = \sqrt{|E(\sigma)|} < 1, \tag{A.6}$$

where $E : \mathbb{R}\setminus[-\lambda_n, -\lambda_1] \to \mathbb{R}$ is

$$E(\sigma) \stackrel{\text{def}}{=} B\left(\Phi(-\sigma)^{-1}\right).$$

Furthermore, if $\sigma \geq 0$,

$$\mu \leq \frac{\kappa^{1/4}-1}{\kappa^{1/4}+1}, \tag{A.7}$$

where $\kappa = \lambda_n/\lambda_1$.

Proof. Employing (A.5) with $z = -\sigma$ immediately gives

$$\begin{aligned} x(\sigma) &= -((-\sigma)I - A)^{-1}b = \frac{1}{\Phi(-\sigma)\Psi'(\Phi(-\sigma))}\sum_{i=0}^{\infty}\phi_i(\Phi(-\sigma)^{-1})M_i(A)b \\ &= \sum_{i=0}^{\infty} a_i M_i(A)b, \end{aligned} \tag{A.8}$$

where

$$a_i = \frac{\phi_i(\Phi(-\sigma)^{-1})}{\Phi(-\sigma)\Psi'(\Phi(-\sigma))}.$$

Using (A.2) with $w = -\sigma$, we may bound

$$|a_i| \leq \frac{|\phi_i(\Phi(-\sigma)^{-1})|}{|\Phi(-\sigma)||\Psi'(\Phi(-\sigma))|} \leq \frac{|B\left(\Phi(-\sigma)^{-1}\right)^{i/2}|}{|\Phi(-\sigma)||\Psi'(\Phi(-\sigma))|} \leq c_1\mu^i, \tag{A.9}$$

where

$$c_1 = \frac{1}{\Psi'\left(\Phi(-\lambda_1 - \delta_1)\right)}$$

is a constant, and

$$\mu = \sqrt{|B\left(\Phi(-\sigma)^{-1}\right)|} \equiv \sqrt{|E(\sigma)|} < 1 \tag{A.10}$$

from (A.1) as $|\Phi(-\sigma)^{-1}| < 1$.

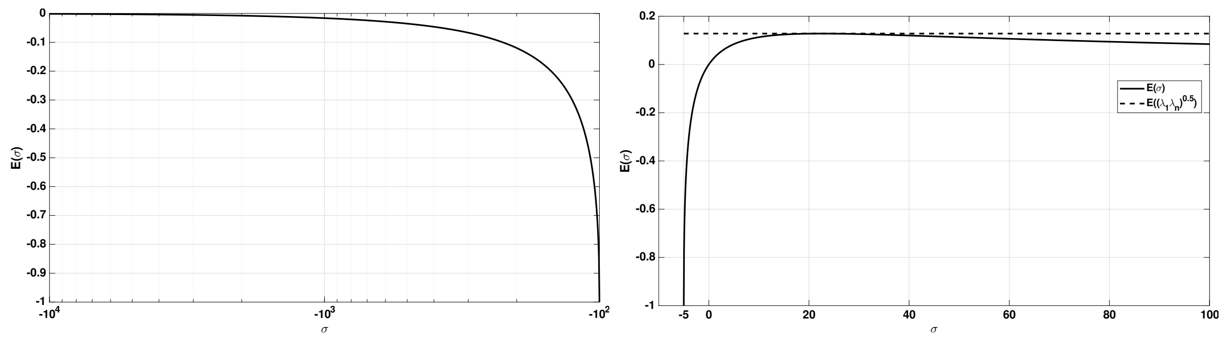

Figure A.1: Upper bound on the convergence rate of extended Krylov method to solve the shifted linear system with a shift σ. The spectrum of the matrix is in $[5, 100]$.

Since $M_i(A)b \in \mathcal{K}_i(A, b)$ for $i = 1, \ldots, 2k$ by definition (4.1) of $\mathcal{K}_i(A, b)$ and from (A.3), the estimate

$$x_j \stackrel{\text{def}}{=} \sum_{i=0}^{j} a_i M_i(A) b$$

is in $\mathcal{K}_j(A, b)$ for $j = 1, \ldots 2k$. But then we have

$$\|x(\sigma) - x_j\| \leq \sum_{i=j+1}^{\infty} |a_i| \|M_i(A)\| \|b\| \leq c\mu^j.$$

from (A.8), (A.4) and and (A.9) respectively, where $c = c_M c_1 \|b\| \mu/(1 - \mu)$ as $\mu < 1$.

A simple analysis, e.g., [18, Prop.4.1] shows that the function E evolves as follows:

σ	$-\infty$		$-\lambda_n$	$-\lambda_1$		0		$\sqrt{\lambda_1 \lambda_n}$		∞
$E(\sigma)$	0	$\searrow$	-1	-1	$\nearrow$	0	$\nearrow$	$\left(\frac{\kappa^{1/4}-1}{\kappa^{1/4}+1}\right)^2$	$\searrow$	0

where $\kappa = \frac{\lambda_n}{\lambda_1}$; as an illustration, Figure A.1 depicts the function E on $\mathbb{R} \setminus (-\lambda_n, -\lambda_1)$ with $\lambda_1 = 5$ and $\lambda_n = 100$. Therefore, for symmetric positive-definite A and a shift $\sigma \notin [-\lambda_n, -\lambda_1]$, the convergence rate of EKS in approximating the solution of the shifted linear system $(A + \sigma I)x = b$ is $\sqrt{|E(\sigma)|}$, and if $\sigma \geq 0$, we have $E(\sigma) \leq \left(\frac{\kappa^{1/4}-1}{\kappa^{1/4}+1}\right)^2$, and thus (A.7) follows from (A.10). $\square$

Theorem A.3. *The extended Krylov method to approximate a solution manifold* $\{x(\sigma)\}_{\sigma \notin (-\lambda_n - \delta_n, -\lambda_1 + \delta_1)}$ *converges at worst linearly with a convergence rate*

$$\mu = \max\left(\sqrt{|E(-\lambda_n - \delta_n)|}, \sqrt{|E(-\lambda_1 + \delta_1)|}, \left(\frac{\kappa^{1/4} - 1}{\kappa^{1/4} + 1}\right)\right),$$

where E is defined in Proposition A.2 and $\kappa = \lambda_n/\lambda_1$.

Proof. The proof follows straightforwardly from Proposition A.2, and the discussion of the evolution of $E(\sigma)$ given there. $\square$

B Modified norm trust region subproblem solver

Here we provide a detailed description of the modifications to the TREK algorithm (Algorithm 5.1) that are necessary when using the trust region norm for which $\|x\|_S^2 = x^T S x$, with $S = P^T P$, as discussed in Section 5.2. As we stated there, it suffices to apply Algorithm 5.1 with the matrix $A_{\rm P} = P^{-T} A P^{-1}$ and vector $b_{\rm P} = P^{-T} b$, and to recover the solution via $x_* = P^{-1} x_{{\rm P}*}$. Since $S^{-1} = P^{-1} P^{-T}$, this is formally as Algorithm B.1 on the next page.

Under the transformation $w_k = P^{-1} v_k$ i.e., $v_k = P w_k$, and $u = Pv$, i.e., $v = P^{-1} u$ (and then replacing $v_k \rightarrow w_k$ and $u \rightarrow v$ so that the resulting changes may be easily compared with Algorithm 5.1), and introducing temporary workspace vectors z and s, we may rewrite Algorithm B.1 as Algorithm B.2 on page 30.

While it might appear that Algorithm B.2 requires multiple products of S with selected vectors, as needed to calculate the terms $v_i^T S u$ that define α_i and β_i, fortunately this is not the case. There are two ways to avoid this. In the first, we maintain appropriate vectors $q_i \stackrel{\rm def}{=} S v_i$. We record this as Algorithm B.3 on page 31.

In practice, use two arrays q_+ and q_-, and store and successively overwrite $q_+ = q_k$ for $k \geq 0$ and $q_- = q_k$ for $k < 0$ with $k = 1, 2, \ldots, m$.

The second way to avoid multiple products with S is to use q_i as before, but now to update $w \stackrel{\rm def}{=} S u$ as it appears. We record this variant as Algorithm 5.2 on page 16. Once again, in practice, use two arrays q_+ and q_-, and store and successively overwrite $q_+ = q_k$ for $k \geq 0$ and $q_- = q_k$ for $k < 0$.

Computationally, the main difference is that Algorithm B.3 requires an additional S product per cycle, while Algorithm 5.2 has an additional four vector subtractions.

Algorithm B.1: The `TREK` algorithm to solve the trust-region subproblem (5.6)

Input: symmetric $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, symmetric, positive-definite $S \in \mathbb{R}^{n\times n}$, $\Delta > 0$, a stopping tolerance $\epsilon > 0$ and an iteration bound $m \geq 1$.

Output: $x_* = \text{(approx)} \arg\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^T Ax - b^T x$ subject to $\|Px\| \leq \Delta$.

Algorithm:

$x := A^{-1}b$, $w := Px$;
if $\|w\| \leq \Delta$ **then**
  exit with the interior solution $x_* = x$ and shift $\sigma_* = 0$;
$\delta_0 := \|P^{-T}b\|$, $v_0 := P^{-T}b/\delta_0$, $u := w/\delta_0$;
$\beta_0 := u^T v_0$;
$u \leftarrow u - \beta_0 v_0$;
$\delta_{-1} := \|u\|$;
for $k = 1, 2, \ldots, m$ **do**
  if $\delta_{-k} > 0$ **then**
    $v_{-k} := u/\delta_{-k}$;
    $u := P^{-T}AP^{-1}v_{-k}$;
    $\alpha_{k-1} := u^T v_{k-1}$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$;
    $\alpha_{-k} := u^T v_{-k}$, $u \leftarrow u - \alpha_{-k}v_{-k}$;
    $\delta_k := \|u\|$;
  if $k > 1$ **then**
    update P_{2k-1} from P_{2k-2} using (4.13) and form $b_{2k-1} = (b_{2k-2}^T, 0)^T$;
  else
    initialize P_1 from (4.11) and set $b_1 = \delta_0$;
  $y_{2k-1} := \arg\min \frac{1}{2}y^T P_{2k-1}y - b_{2k-1}^T y$ s.t. $\|y\| \leq \Delta$ and its optimal shift σ_{2k-1};
  compute $\|r_{2k-1}\|$ from (5.4);
  if $\|r_{2k-1}\| \leq \epsilon$ **then**
    exit with the interior solution $x_* = V_{2k-1}y_{2k-1}$, shift $\sigma_* = \sigma_{2k-1}$ and $k_{\text{stop}} = k$;
  if $\delta_k > 0$ **then**
    $v_k := u/\delta_k$, $u := PA^{-1}P^T v_k$;
    $\beta_{-k} := u^T v_{-k}$, $u \leftarrow u - \beta_{-k}v_{-k}$;
    $\beta_k := u^T v_k$, $u \leftarrow u - \beta_k v_k$;
  if $k < m$ **then**
    if $\delta_k > 0$ **then**
      $\delta_{-k-1} := \|u\|$;
    update P_{2k} from P_{2k-1} using (4.14) and form $b_{2k} = (b_{2k-1}^T, 0)^T$;
    $y_{2k} := \arg\min \frac{1}{2}y^T P_{2k}y - b_{2k}^T y$ s.t. $\|y\| \leq \Delta$ together with its optimal shift σ_{2k};
    compute $\|r_{2k}\|$ from (5.5);
    if $\|r_{2k}\| \leq \epsilon$ **then**
      exit with the interior solution $x_* = V_{2k}y_{2k}$, shift $\sigma_* = \sigma_{2k}$ and $k_{\text{stop}} = k$;

Algorithm B.2: The `TREK` algorithm to solve the trust-region subproblem (5.9)

Input: symmetric $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, symmetric, positive-definite $S \in \mathbb{R}^{n\times n}$, $\Delta > 0$, a stopping tolerance $\epsilon > 0$ and an iteration bound $m \geq 1$.

Output: $x_* = \text{(approx)} \arg\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^TAx - b^Tx$ subject to $\|x\|_S \leq \Delta$.

Algorithm:

$x := A^{-1}b$, $u := Sx$;

if $\sqrt{u^Tx} \leq \Delta$ **then**

- **exit** with the interior solution $x_* = x$ and shift $\sigma_* = 0$;

$w := S^{-1}b$, $\delta_0 := \sqrt{b^Tw}$;

$v_0 := w/\delta_0$, $w := u/\delta_0$, $u := x/\delta_0$;

$\beta_0 := v_0^TSu$, $u \leftarrow u - \beta_0 v_0$;

$w = Su$, $\delta_{-1} := \sqrt{u^Tw}$;

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

- **if** $\delta_{-k} > 0$ **then**
  - $v_{-k} := u/\delta_{-k}$, $w := Av_{-k}$, $u := S^{-1}w$;
  - $\alpha_{-k} := v_{-k}^TSu$, $u \leftarrow u - \alpha_{-k}v_{-k}$;
  - $\alpha_{k-1} := v_{k-1}^TSu$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$;
  - $w = Su$, $\delta_k := \sqrt{u^Tw}$;
- **if** $k > 1$ **then**
  - update P_{2k-1} from P_{2k-2} using (4.13) and form $b_{2k-1} = (b_{2k-2}^T, 0)^T$;
- **else**
  - initialize P_1 from (4.11) and set $b_1 = \delta_0$;
- $y_{2k-1} := \arg\min \frac{1}{2}y^TP_{2k-1}y - b_{2k-1}^Ty$ s.t. $\|y\| \leq \Delta$ and its optimal shift σ_{2k-1};
- compute $\|r_{2k-1}\|$ from (5.4);
- **if** $\|r_{2k-1}\| \leq \epsilon$ **then**
  - **exit** with the interior solution $x_* = V_{2k-1}y_{2k-1}$, shift $\sigma_* = \sigma_{2k-1}$ and $k_{\text{stop}} = k$;
- **if** $\delta_k > 0$ **then**
  - $v_k := u/\delta_k$, $w := Sv_k$, $u := A^{-1}w$;
  - $\beta_{-k} := v_{-k}^TSu$, $u \leftarrow u - \beta_{-k}v_{-k}$;
  - $\beta_k := v_k^TSu$, $u \leftarrow u - \beta_kv_k$;
- **if** $k < m$ **then**
  - **if** $\delta_k > 0$ **then**
    - $w = Su$, $\delta_{-k-1} := \sqrt{u^Tw}$;
  - update P_{2k} from P_{2k-1} from (4.12) and form $b_{2k} = (b_{2k-1}^T, 0)^T$;
  - $y_{2k} := \arg\min \frac{1}{2}y^TP_{2k}y - b_{2k}^Ty$ s.t. $\|y\| \leq \Delta$ together with its optimal shift σ_{2k};
  - compute $\|r_{2k}\|$ from (5.5);
  - **if** $\|r_{2k}\| \leq \epsilon$ **then**
    - **exit** with the interior solution $x_* = V_{2k}y_{2k}$, shift $\sigma_* = \sigma_{2k}$ and $k_{\text{stop}} = k$;

Algorithm B.3: The TREK algorithm to solve the trust-region subproblem (5.9)

Input: symmetric $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, symmetric, positive-definite $S \in \mathbb{R}^{n\times n}$, $\Delta > 0$, a stopping tolerance $\epsilon > 0$ and an iteration bound $m \geq 1$.

Output: $x_* = \text{(approx)} \arg\min_{x\in\mathbb{R}^n} \ \frac{1}{2}x^TAx - b^Tx$ subject to $\|x\|_S \leq \Delta$.

Algorithm:

$x := A^{-1}b$, $u := Sx$;
if $\sqrt{u^Tx} \leq \Delta$ **then**
  exit with the interior solution $x_* = x$ and shift $\sigma_* = 0$;
$w := S^{-1}b$, $\delta_0 := \sqrt{b^Tw}$;
$v_0 := w/\delta_0$, $q_0 := b/\delta_0$, $w := u/\delta_0$, $u := x/\delta_0$;
$\beta_0 := u^Tq_0$, $u \leftarrow u - \beta_0 v_0$;
$w = Su$, $\delta_{-1} := \sqrt{u^Tw}$;
for $k = 1, 2, \ldots, m$ **do**
  if $\delta_{-k} > 0$ **then**
    $v_{-k} := u/\delta_{-k}$, $q_{-k} := w/\delta_{-k}$;
    $w := Av_{-k}$, $u := S^{-1}w$;
    $\alpha_{-k} := u^Tq_{-k}$, $u \leftarrow u - \alpha_{-k}v_{-k}$;
    $\alpha_{k-1} := u^Tq_{k-1}$, $u \leftarrow u - \alpha_{k-1}v_{k-1}$;
    $w = Su$, $\delta_k := \sqrt{u^Tw}$;
  if $k > 1$ **then**
    update P_{2k-1} from P_{2k-2} using (4.13) and form $b_{2k-1} = (b_{2k-2}^T, 0)^T$;
  else
    initialize P_1 from (4.11) and set $b_1 = \delta_0$;
  $y_{2k-1} := \arg\min \frac{1}{2}y^TP_{2k-1}y - b_{2k-1}^Ty$ s.t. $\|y\| \leq \Delta$ and its optimal shift σ_{2k-1};
  compute $\|r_{2k-1}\|$ from (5.4);
  if $\|r_{2k-1}\| \leq \epsilon$ **then**
    exit with the interior solution $x_* = V_{2k-1}y_{2k-1}$, shift $\sigma_* = \sigma_{2k-1}$ and $k_{\text{stop}} = k$;
  if $\delta_k > 0$ **then**
    $v_k := u/\delta_k$, $q_k := w/\delta_k$;
    $u := A^{-1}q_k$;
    $\beta_{-k} := u^Tq_{-k}$, $u \leftarrow u - \beta_{-k}v_{-k}$;
    $\beta_k := u^Tq_k$, $u \leftarrow u - \beta_k v_k$;
  if $k < m$ **then**
    if $\delta_k > 0$ **then**
      $w = Su$, $\delta_{-k-1} := \sqrt{u^Tw}$;
    update P_{2k} from P_{2k-1} using (4.14) and form $b_{2k} = (b_{2k-1}^T, 0)^T$;
    $y_{2k} := \arg\min \frac{1}{2}y^TP_{2k}y - b_{2k}^Ty$ s.t. $\|y\| \leq \Delta$ together with its optimal shift σ_{2k};
    compute $\|r_{2k}\|$ from (5.5);
    if $\|r_{2k}\| \leq \epsilon$ **then**
      exit with the interior solution $x_* = V_{2k}y_{2k}$, shift $\sigma_* = \sigma_{2k}$ and $k_{\text{stop}} = k$;

C Detailed numerical experiments

Here we summarize the results from our numerical experiments that lead to Figure 7.1. As we said, we consider the set of 96 unconstrained test problems in the current release (2025-09-09) of the CUTEst optimization test examples [14] that have 1000 or more variables, evaluate the gradient and Hessian to provide $-b$ and A, and for each a pair of radii $\Delta = 1$ and then 0.5. We apply the GALAHAD solvers TREK (Algorithm 5.1), TRS [15] and GLTR [12] in each case; when $\Delta = 0.5$, we use data accumulated from the $\Delta = 1$ case to "warm start" the second solve as each solver provides for this.

In Table C.1 we report the name of each test problem, the size of the instance chosen, n (many CUTEst examples have multiple dimensions), the radius value, Δ and the optimal objective function found. For TREK, we give the terminal iteration count, k_{stop}, which is the number of solves with A required, and the clock time required. For TRS, we report the number of factorizations and clock time required, while for GLTR, we state the number of vector products involving A and again the time. The symbol † indicates an instance that the iteration bound $m = 100$ was reached, while and ‡ indicates that a factorization failed through singularity.

Table C.1: Subproblem solves for CUTEst examples: Complete results

				TREK		TRS		GLTR	
name	n	Δ	$f(x_*)$	k_{stop}	time	facts	time	prods	time
ARWHEAD	5000	1.0	-9.99800000E+03	0	0.00	2	0.00	2	0.00
ARWHEAD	5000	0.5	-9.99800000E+03	1	0.00	3	0.00	1	0.00
BA-L16LS	66462	1.0	-4.00647969E+09	10	1.18	6	1.94	69	2.76
BA-L16LS	66462	0.5	-1.27074021E+09	10	0.00	10	1.43	34	1.42
BA-L16LS	66462	1.0	-4.00647969E+09	10	0.82	6	0.94	69	2.55
BA-L16LS	66462	0.5	-1.27074021E+09	10	0.00	10	0.68	34	1.34
ARWHEAD	5000	1.0	-9.99800000E+03	-1	0.00	2	0.00	2	0.00
ARWHEAD	5000	0.5	-9.99800000E+03	1	0.00	3	0.00	1	0.00
BA-L16LS	66462	1.0	-4.00647969E+09	10	0.84	6	0.94	69	2.60
BA-L16LS	66462	0.5	-1.27074021E+09	10	0.00	10	0.73	34	1.32
BA-L21LS	34134	1.0	-1.43350953E+08	100†	0.88	8	0.39	1129	18.09
BA-L21LS	34134	0.5	-1.27987567E+08	100†	0.00	12	0.27	564	8.95
BA-L49LS	23769	1.0	-2.32581840E+10	6	0.25	8	0.45	27	0.40
BA-L49LS	23769	0.5	-5.82477997E+09	6	0.00	11	0.20	13	0.21
BA-L52LS	192627	1.0	-1.78720629E+10	31	4.65	11	9.18	173	27.82
BA-L52LS	192627	0.5	-1.26606809E+10	31	0.01	16	4.79	86	13.82
BA-L73LS	33753	1.0	-9.88938047E+10	4	0.37	10	0.93	25	0.56
BA-L73LS	33753	0.5	-2.47259422E+10	4	0.00	13	0.35	12	0.29
BDQRTIC	5000	1.0	-4.70328224E+05	4	0.01	4	0.01	13	0.00
BDQRTIC	5000	0.5	-4.37969301E+05	4	0.00	8	0.01	6	0.00
BOX	10000	1.0	-4.99665149E+01	1	0.01	3	0.01	3	0.00
BOX	10000	0.5	-2.49841935E+01	1	0.00	6	0.01	1	0.00
BOXPOWER	20000	1.0	-1.27930301E+05	3	0.01	3	0.02	5	0.00
BOXPOWER	20000	0.5	-7.39842920E+04	3	0.00	6	0.02	2	0.00
BROYDN3DLS	5000	1.0	-5.47141790E+02	6	0.01	3	0.02	19	0.00
BROYDN3DLS	5000	0.5	-2.78620552E+02	6	0.00	6	0.01	9	0.00
BROYDN7D	5000	1.0	-1.04198440E+03	4	0.01	3	0.01	11	0.00
BROYDN7D	5000	0.5	-5.29359803E+02	4	0.00	6	0.01	5	0.00
BROYDNBDLS	5000	1.0	-7.54822794E+03	5	0.02	3	0.01	15	0.01
BROYDNBDLS	5000	0.5	-3.83192245E+03	5	0.00	6	0.01	7	0.00
BRYBND	5000	1.0	-7.54822794E+03	5	0.02	3	0.01	15	0.01
BRYBND	5000	0.5	-3.83192245E+03	5	0.00	6	0.01	7	0.00
CHAINWOO	4000	1.0	-4.18808861E+05	4	0.01	2	0.01	9	0.00
CHAINWOO	4000	0.5	-2.10745351E+05	4	0.00	4	0.01	4	0.00
COSINE	10000	1.0	-7.33802606E+01	5	0.03	2	0.02	11	0.00
COSINE	10000	0.5	-3.63234227E+01	5	0.00	4	0.01	5	0.00
CRAGGLVY	5000	1.0	-2.70357001E+05	2	0.02	3	0.01	5	0.00
CRAGGLVY	5000	0.5	-1.38612789E+05	2	0.00	6	0.01	2	0.00
CURLY10	10000	1.0	-2.55466912E+03	52	0.11	3	0.05	157	0.04
CURLY10	10000	0.5	-6.72371854E+02	52	0.00	6	0.04	78	0.02
CURLY20	10000	1.0	-9.12107858E+03	49	0.11	3	0.05	205	0.07
CURLY20	10000	0.5	-2.35575388E+03	49	0.00	6	0.04	102	0.03
CURLY30	10000	1.0	-1.97300910E+04	41	0.15	3	0.10	235	0.11
CURLY30	10000	0.5	-5.06068395E+03	41	0.00	6	0.06	117	0.04
DIXMAANB	3000	1.0	-1.94571746E+03	4	0.01	3	0.01	11	0.00
DIXMAANB	3000	0.5	-9.82394549E+02	4	0.00	6	0.01	5	0.00
DIXMAANC	3000	1.0	-3.67420974E+03	4	0.02	3	0.01	11	0.00

Table C.1: Subproblem solves for CUTEst examples: Complete results (continued)

				TREK		TRS		GLTR	
name	n	Δ	$f(x_*)$	k_{stop}	time	facts	time	prods	time
DIXMAANC	3000	0.5	-1.85594245E+03	4	0.00	6	0.01	5	0.00
DIXMAAND	3000	1.0	-7.40783941E+03	4	0.01	3	0.01	11	0.00
DIXMAAND	3000	0.5	-3.74285037E+03	4	0.00	6	0.01	5	0.00
DIXMAANE1	3000	1.0	-1.03848735E+03	4	0.01	3	0.00	11	0.00
DIXMAANE1	3000	0.5	-5.25114487E+02	4	0.00	6	0.00	5	0.00
DIXMAANF	3000	1.0	-1.83737329E+03	4	0.01	3	0.01	11	0.00
DIXMAANF	3000	0.5	-9.28137690E+02	4	0.00	6	0.01	5	0.00
DIXMAANG	3000	1.0	-3.56193345E+03	4	0.01	3	0.01	11	0.00
DIXMAANG	3000	0.5	-1.79971801E+03	4	0.00	6	0.01	5	0.00
DIXMAANH	3000	1.0	-7.28769893E+03	4	0.01	3	0.01	11	0.00
DIXMAANH	3000	0.5	-3.68269064E+03	4	0.00	6	0.01	5	0.00
DIXMAANI1	3000	1.0	-1.00064454E+03	4	0.01	3	0.00	11	0.00
DIXMAANI1	3000	0.5	-5.06141189E+02	4	0.00	6	0.00	5	0.00
DIXMAANJ	3000	1.0	-1.79984433E+03	4	0.01	3	0.01	11	0.00
DIXMAANJ	3000	0.5	-9.09324798E+02	4	0.00	6	0.01	5	0.00
DIXMAANK	3000	1.0	-3.52376478E+03	4	0.01	3	0.01	11	0.00
DIXMAANK	3000	0.5	-1.78058448E+03	4	0.00	6	0.01	5	0.00
DIXMAANL	3000	1.0	-7.24829529E+03	4	0.01	3	0.01	11	0.00
DIXMAANL	3000	0.5	-3.66293885E+03	4	0.00	6	0.01	5	0.00
DIXMAANM1	3000	1.0	-4.27493497E+02	5	0.01	3	0.01	13	0.00
DIXMAANM1	3000	0.5	-2.16348948E+02	5	0.00	6	0.01	6	0.00
DIXMAANN	3000	1.0	-9.97842749E+02	5	0.01	3	0.01	13	0.00
DIXMAANN	3000	0.5	-5.05235340E+02	5	0.00	6	0.01	6	0.00
DIXMAANO	3000	1.0	-1.90145642E+03	5	0.01	3	0.01	13	0.00
DIXMAANO	3000	0.5	-9.63139983E+02	5	0.00	6	0.01	6	0.00
DIXMAANP	3000	1.0	-3.85352076E+03	5	0.01	3	0.01	13	0.00
DIXMAANP	3000	0.5	-1.95233964E+03	5	0.00	6	0.01	6	0.00
DIXON3DQ	10000	1.0	-4.35180402E+00	19	0.02	5	0.03	45	0.01
DIXON3DQ	10000	0.5	-2.48148299E+00	19	0.00	8	0.02	22	0.00
DQDRTIC	5000	1.0	-8.50546818E+04	3	0.00	2	0.00	7	0.00
DQDRTIC	5000	0.5	-4.25775883E+04	3	0.00	4	0.00	3	0.00
DQRTIC	5000	1.0	-1.33489191E+13	2	0.00	2	0.00	5	0.00
DQRTIC	5000	0.5	-6.67448869E+12	2	0.00	4	0.00	2	0.00
EDENSCH	2000	1.0	-9.90061935E+04	3	0.00	2	0.00	7	0.00
EDENSCH	2000	0.5	-4.96303271E+04	3	0.00	4	0.00	3	0.00
EG2	1000	1.0	-1.73066127E+02	0	0.00	2	0.00	1	0.00
EG2	1000	0.5	-1.64667237E+02	1	0.00	4	0.00	0	0.00
EIGENALS	2550	1.0	-1.01221801E+03	7	0.75	3	0.85	25	0.04
EIGENALS	2550	0.5	-4.77552997E+02	7	0.00	6	0.79	12	0.02
EIGENBLS	2550	1.0	-3.52937908E+01	8	0.70	3	0.89	25	0.04
EIGENBLS	2550	0.5	-1.80972156E+01	8	0.00	5	0.62	12	0.02
EIGENCLS	2652	1.0	-4.70591220E+02	9	0.78	3	0.91	31	0.04
EIGENCLS	2652	0.5	-2.33514688E+02	9	0.00	6	0.93	15	0.02
ENGVAL1	5000	1.0	-8.67081566E+03	3	0.01	2	0.01	9	0.00
ENGVAL1	5000	0.5	-4.35940622E+03	3	0.00	4	0.01	4	0.00
EXTROSNB	1000	1.0	-3.66203611E+04	4	0.00	3	0.00	11	0.00
EXTROSNB	1000	0.5	-1.86350898E+04	4	0.00	6	0.00	5	0.00
FLETBV3M	5000	1.0	-4.36166522E+01	3	0.01	2	0.01	7	0.00
FLETBV3M	5000	0.5	-2.18337960E+01	3	0.00	4	0.01	3	0.00
FLETCBV2	5000	1.0	-3.88209445E-06	13	0.01	4	0.01	9737	412.07
FLETCBV2	5000	0.5	-2.02683952E-06	16	0.00	8	0.01	4868	0.53
FLETCBV3	5000	1.0	-4.36172752E+01	3	0.01	2	0.01	7	0.04
FLETCBV3	5000	0.5	-2.18341109E+01	3	0.00	4	0.01	3	0.00
FLETCHBV	5000	1.0	-2.74868337E+09	3	0.01	2	0.01	9	0.00
FLETCHBV	5000	0.5	-1.37709990E+09	3	0.00	4	0.01	4	0.00
FLETCHCR	1000	1.0	-1.08786732E+01	1	0.00	3	0.00	3	0.00
FLETCHCR	1000	0.5	-1.05114880E+01	1	0.00	7	0.00	1	0.00
FMINSRF2	5625	1.0	-2.74210232E-01	44	0.04	4	0.03	107	0.03
FMINSRF2	5625	0.5	-1.42608286E-01	44	0.00	8	0.03	53	0.01
FMINSURF	5625	1.0	-2.74396123E-01	44	5.28	4	8.28	107	0.03
FMINSURF	5625	0.5	-1.42697922E-01	44	0.00	8	8.05	53	0.01
FREUROTH	5000	1.0	-5.51793805E+04	3	0.01	3	0.01	7	0.00
FREUROTH	5000	0.5	-2.75854293E+04	3	0.00	6	0.01	3	0.00
GENHUMPS	5000	1.0	-6.64118303E+03	6	0.03	3	0.02	13	0.00
GENHUMPS	5000	0.5	-3.16552923E+03	6	0.00	6	0.01	6	0.00
INDEF	5000	1.0	-2.10493952E+03	6	0.01	103‡	0.12	13	0.00
INDEF	5000	0.5	-5.26825583E+02	6	0.00	207‡	0.25	6	0.00
INDEFM	100000	1.0	-4.20748078E+04	3	0.09	11	0.38	11	0.05
INDEFM	100000	0.5	-1.05192638E+04	3	0.00	16	0.23	5	0.03
JIMACK	3549	1.0	-4.37368349E-01	100†	0.45	26	0.57	3201	17.07
JIMACK	3549	0.5	-2.41925217E-01	100†	0.00	31	0.14	1600	0.62
LIARWHD	5000	1.0	-4.61798034E+05	1	0.00	3	0.00	3	0.01
LIARWHD	5000	0.5	-2.36034628E+05	1	0.00	6	0.00	1	0.00
MODBEALE	20000	1.0	-3.04981491E+05	2	0.05	2	0.03	7	0.01
MODBEALE	20000	0.5	-1.52960464E+05	2	0.00	4	0.03	3	0.00
MOREBV	5000	1.0	-7.72418940E-14	9	0.01	107‡	0.13	295	0.17
MOREBV	5000	0.5	-7.72418940E-14	54	0.03	116	0.02	294	0.02
MSQRTALS	1024	1.0	-3.70040042E+02	4	0.16	4	0.14	27	0.01
MSQRTALS	1024	0.5	-1.75608597E+02	5	0.00	7	0.11	13	0.01
MSQRTBLS	1024	1.0	-3.69312117E+02	4	0.16	4	0.14	27	0.01
MSQRTBLS	1024	0.5	-1.75279355E+02	5	0.00	7	0.12	13	0.01
NCB20	5010	1.0	-2.77922014E+02	6	0.03	4	0.02	27	0.07

Table C.1: Subproblem solves for CUTEst examples: Complete results (continued)

				TREK		TRS		GLTR	
name	n	Δ	$f(x_*)$	k_{stop}	time	facts	time	prods	time
NCB20	5010	0.5	-1.39893162E+02	6	0.00	7	0.02	13	0.03
NCB20B	5000	1.0	-2.77950996E+02	6	0.03	4	0.02	27	0.05
NCB20B	5000	0.5	-1.39907519E+02	6	0.00	7	0.02	13	0.02
NONCVXU2	5000	1.0	-3.33554888E+06	2	0.21	2	0.34	5	0.00
NONCVXU2	5000	0.5	-1.66777663E+06	2	0.00	4	0.34	2	0.00
NONCVXUN	5000	1.0	-3.56003262E+06	2	0.02	2	0.02	5	0.00
NONCVXUN	5000	0.5	-1.78001885E+06	2	0.00	4	0.02	2	0.00
NONDIA	5000	1.0	-1.49970308E+06	1	0.01	3	0.01	3	0.00
NONDIA	5000	0.5	-8.75226609E+05	1	0.00	6	0.01	1	0.00
NONDQUAR	5000	1.0	-3.33683482E+03	43	0.03	10	0.02	117	0.01
NONDQUAR	5000	0.5	-3.33513581E+03	43	0.00	15	0.01	58	0.00
NONMSQRT	4900	1.0	-1.15116171E+03	8	0.07	5	0.03	41	0.17
NONMSQRT	4900	0.5	-5.50172607E+02	8	0.00	9	0.03	20	0.08
OSCIGRAD	100000	1.0	-5.34690566E+08	8	0.31	5	0.23	43	0.13
OSCIGRAD	100000	0.5	-4.31017580E+08	8	0.00	9	0.20	21	0.09
PENALTY1	1000	1.0	-2.43960328E+13	1	0.07	2	0.07	1	0.00
PENALTY1	1000	0.5	-1.21985172E+13	1	0.00	4	0.05	0	0.00
POWELLSG	5000	1.0	-1.57803913E+04	2	0.01	3	0.01	7	0.00
POWELLSG	5000	0.5	-7.99992041E+03	2	0.00	6	0.01	3	0.00
POWER	10000	1.0	-1.13407283E+14	5	13.59	3	29.47	11	0.00
POWER	10000	0.5	-5.72243018E+13	5	0.00	6	29.24	5	0.00
QUARTC	5000	1.0	-1.33489191E+13	2	0.00	2	0.00	5	0.00
QUARTC	5000	0.5	-6.67448869E+12	2	0.00	4	0.00	2	0.00
SBRYBND	5000	1.0	-9.63032705E+11	20	0.03	16	0.03	179	0.06
SBRYBND	5000	0.5	-2.40760941E+11	20	0.00	21	0.02	89	0.03
SCHMVETT	5000	1.0	-7.28426352E+01	10	0.01	3	0.01	35	0.01
SCHMVETT	5000	0.5	-3.68824395E+01	10	0.00	6	0.01	17	0.01
SCOSINE	5000	1.0	-8.25379341E+10	13	0.03	11	0.02	187	0.02
SCOSINE	5000	0.5	-2.06344856E+10	13	0.00	13	0.01	93	0.01
SCURLY10	10000	1.0	-1.28077199E+30	4	0.04	3	0.03	11	0.00
SCURLY10	10000	0.5	-6.43068284E+29	4	0.00	6	0.03	5	0.00
SCURLY20	10000	1.0	-1.63614356E+31	4	0.07	3	0.04	11	0.00
SCURLY20	10000	0.5	-8.21427988E+30	4	0.00	6	0.04	5	0.00
SCURLY30	10000	1.0	-7.47542209E+31	4	0.09	3	0.06	9	0.01
SCURLY30	10000	0.5	-3.75274481E+31	4	0.00	6	0.06	4	0.00
SINQUAD	5000	1.0	-7.12672063E+03	2	0.00	4	0.00	5	0.00
SINQUAD	5000	0.5	-2.75259696E+03	2	0.00	8	0.00	2	0.00
SPARSINE	5000	1.0	-2.92330968E+06	4	0.48	3	1.16	11	0.00
SPARSINE	5000	0.5	-1.46942254E+06	4	0.00	6	1.18	5	0.00
SPARSQUR	10000	1.0	-1.19555581E+06	7	3.09	2	5.80	15	0.01
SPARSQUR	10000	0.5	-6.09140674E+05	7	0.00	4	5.86	7	0.00
SPMSRTLS	4999	1.0	-8.16740777E+01	7	0.02	3	0.01	21	0.01
SPMSRTLS	4999	0.5	-3.97487352E+01	7	0.00	6	0.01	10	0.00
SROSENBR	5000	1.0	-1.15690489E+02	9	0.01	4	0.01	5	0.00
SROSENBR	5000	0.5	-6.52884475E+01	9	0.00	8	0.01	2	0.00
SSBRYBND	5000	1.0	-5.98063290E+06	17	0.03	12	0.04	145	0.09
SSBRYBND	5000	0.5	-1.51010118E+06	17	0.00	15	0.02	72	0.04
SSCOSINE	5000	1.0	-5.13429696E+05	12	0.02	16	0.02	155	0.02
SSCOSINE	5000	0.5	-1.28378413E+05	12	0.00	19	0.01	77	0.01
TOINTGSS	5000	1.0	-4.23179011E+02	4	0.02	2	0.02	9	0.00
TOINTGSS	5000	0.5	-2.11839554E+02	4	0.00	4	0.01	4	0.00
TQUARTIC	5000	1.0	-2.91590249E-02	1	0.00	3	0.00	3	0.00
TQUARTIC	5000	0.5	-1.66434134E-02	1	0.00	6	0.00	1	0.00
TRIDIA	10000	1.0	-1.14762126E+06	7	0.02	3	0.02	15	0.00
TRIDIA	10000	0.5	-5.75687744E+05	7	0.00	6	0.02	7	0.00
VAREIGVL	5000	1.0	-1.03187418E+04	3	2.68	3	4.55	9	0.00
VAREIGVL	5000	0.5	-5.19645395E+03	3	0.00	6	4.58	4	0.00
WOODS	4000	1.0	-5.13132992E+05	2	0.01	3	0.01	7	0.00
WOODS	4000	0.5	-2.57913812E+05	2	0.00	6	0.01	3	0.00
YATP1LS	10200	1.0	-2.47600401E+06	1	4.29	3	11.48	3	0.01
YATP1LS	10200	0.5	-1.24008348E+06	1	0.00	5	7.65	1	0.00
YATP2LS	10200	1.0	-6.17000891E+04	1	8.21	3	11.89	3	0.00
YATP2LS	10200	0.5	-3.11097632E+04	1	0.00	5	7.77	1	0.00