

DELFT UNIVERSITY OF TECHNOLOGY

REPORT 11-09

COMPUTATIONAL AND SENSITIVITY ASPECTS OF EIGENVALUE-BASED METHODS FOR THE
LARGE-SCALE TRUST-REGION SUBPROBLEM

MARIELBA ROJAS, BJØRN H. FOTLAND, AND TROND STEIHAUG

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2011

Copyright © 2011 by Department of Applied Mathematical Analysis, Delft, The Netherlands.

No part of the Journal may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission from Department of Applied Mathematical Analysis, Delft University of Technology, The Netherlands.

Computational and Sensitivity Aspects of Eigenvalue-Based Methods for the Large-Scale Trust-Region Subproblem

Marielba Rojas*

Bjørn H. Fotland†

Trond Steihaug‡

August 29, 2011

Abstract

The trust-region subproblem of minimizing a quadratic function subject to a norm constraint arises in the context of trust-region methods in optimization and in the regularization of discrete forms of ill-posed problems, including non-negative regularization by means of interior-point methods. A class of efficient methods and software for solving large-scale trust-region subproblems is based on a parametric-eigenvalue formulation of the subproblem. The solution of a sequence of large symmetric eigenvalue problems is the main computation in these methods. In this work, we study the robustness and performance of eigenvalue-based methods for the large-scale trust-region subproblem. We describe the eigenvalue problems and their features, and discuss the computational challenges they pose as well as some approaches to handle them. We present results from a numerical study of the sensitivity of solutions to the trust-region subproblem to eigenproblem solutions.

1 Introduction

Consider the problem of minimizing a quadratic function subject to a norm constraint:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Hx + g^T x, \\ \text{s.t.} \quad & \|x\| \leq \Delta \end{aligned} \tag{1}$$

where H is an $n \times n$ real, symmetric matrix, g is an n -dimensional vector, Δ is a positive scalar, and $\|\cdot\|$ is the Euclidean norm. We assume that n is large and that matrix-vector products with H can be efficiently computed. Optimality conditions for problem (1) are presented in Lemma 1.1 from [48].

Lemma 1.1 (see [48]). *A feasible vector $x_* \in \mathbb{R}^n$ is a solution to (1) with corresponding Lagrange multiplier λ_* if and only if x_*, λ_* satisfy $(H - \lambda_* I)x_* = -g$, $H - \lambda_* I$ positive semidefinite, $\lambda_* \leq 0$, and $\lambda_*(\Delta - \|x_*\|) = 0$.*

Proof. See [48]. □

Problem (1) is known in optimization as the trust-region subproblem (TRS) arising in the widely-used, state-of-the-art trust-region methods [7, 35]. The main computation in trust-region iterations is

*Delft Institute of Applied Mathematics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (marielba.rojas@tudelft.nl).

†Department of Informatics, University of Bergen, Bergen, Norway. The author is currently with WesternGeco, Stavanger, Norway.

‡Department of Informatics, University of Bergen, Postboks 7803, N-5020 Bergen, Norway (Trond.Steihaug@ii.uib.no).

the solution of a TRS at each step. The following special case of the TRS arises in the regularization of discrete forms of linear ill-posed problems:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T A^T A x - (A^T b)^T x , \\ \text{s.t.} \quad & \|x\| \leq \Delta \end{aligned} \tag{2}$$

with A a discretized operator and b a data vector perturbed by noise such that b is not in the range of A . It is well known (cf. [9, 42, 45]) that (2) is equivalent to Tikhonov regularization [53, 54], with the Lagrange multiplier associated with the norm constraint corresponding to the Tikhonov regularization parameter. The regularization of linear problems by means of (2) requires the solution of one TRS only. Nonlinear ill-posed problems can be solved by means of trust-region methods which require the solution of a sequence of problems of type (2). The TRS in regularization is usually a very challenging problem owing to the presence of high-degree singularities (cf. [42, 45]). Moreover, constraints are often needed in order to model physical properties. This is the case in image restoration, where solutions are arrays of pixel values of color or light intensity which are non-negative properties. Note that the image restoration problem can be formulated as a TRS with additional non-negativity constraints:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T A^T A x - (A^T b)^T x . \\ \text{s.t.} \quad & \|x\| \leq \Delta \\ & x \geq 0 \end{aligned} \tag{3}$$

The interior-point method TRUST_μ for solving (3) was proposed in [46] (see also [4]). The method is based on a logarithmic barrier approach to handle the non-negativity constraints and requires the solution of a sequence of TRS that may be ill-conditioned. The TRS solutions are used to compute dual variables and the duality gap used in the convergence criterion, and the corresponding Lagrange multipliers are used to update a scalar barrier parameter. Therefore, the TRS solution and associated multiplier must be computed very accurately.

The TRS can be efficiently solved by means of the Newton iteration proposed in [31] when the Cholesky factorization of matrices of the form $H - \lambda I$ can be efficiently computed. If H is not explicitly available or is too large, different strategies are needed. Until the mid 1990's, the only available method for solving large-scale trust-region subproblems was the truncated Conjugate Gradients method proposed in [51], and this is still one of the best choices in the context of trust-region methods. However, regularization problems can prove challenging for the technique. Note also that the method computes an approximate TRS solution but not the Lagrange multiplier.

The method in [51] is a so-called *approximate* technique since it does not aim to satisfy the optimality conditions in Lemma 1.1. Note that solving the TRS to optimality will guarantee quadratic convergence of Newton's method combined with the trust-region globalization strategy. Moreover, optimality is required in certain situations such as in the non-negative regularization approach in [46].

In recent years, new *nearly-exact* methods have appeared [11, 41, 43, 44, 50] that aim to compute solutions to large-scale trust-region subproblems that satisfy the optimality conditions. New approximate methods have also been proposed in [15, 16]. Most of the new techniques are also suitable for regularization.

In this work, we focus on the eigenvalue-based techniques which include [11, 41, 43, 44, 50]. In particular, we study computational and sensitivity issues for the LSTRS method [43, 44]. A MATLAB software package implementing LSTRS has been in the public domain for a few years. The LSTRS software has been successfully used or recommended in the literature in the context of optimization and also in large-scale engineering applications (cf. [1, 2, 3, 5, 10, 20, 21, 22, 23, 24, 25, 26, 29, 30, 34, 36, 37, 38, 40, 58]). TRUST_μ , which is based on LSTRS, has also been used in applications and as guideline

for developing new methods (cf. [20, 33, 56, 57]). Many of the applications rely on the efficiency and robustness of the LSTRS method, and this fact was the main motivation for this work.

As mentioned before, the main computation at every iteration of eigenvalue-based methods for the TRS is the solution of a parametric eigenvalue problem that may be computationally challenging, in particular in regularization problems such as (2) and (3). Therefore, in this work we focus on computational and sensitivity aspects associated with these eigenvalue problems. The presentation is organized as follows. In Section 2, we briefly describe eigenvalue-based TRS methods. In Section 3, we discuss the features of the parametric eigenvalue problems arising in TRS methods, the computational challenges they pose, and the strategies used in LSTRS to handle those challenges. In Section 4, we present a numerical sensitivity study of LSTRS solutions with respect to the eigenproblem solutions. Section 5 contains concluding remarks.

2 Eigenvalue-Based TRS Methods

One approach for developing large-scale methods for solving (1) is based on the following fact. It can be shown (see [42]) that there always exists an optimal value of a scalar parameter α such that a solution x to (1) can be computed from a solution $y = (1, x^T)^T$ to

$$\begin{aligned} \min \quad & \frac{1}{2}y^T B_\alpha y \\ \text{s.t.} \quad & y^T y \leq 1 + \Delta^2 \\ & e_1^T y = 1 \end{aligned} \tag{4}$$

where $B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix}$.

Notice that a solution to (4) is an eigenvector with non-zero first component corresponding to the smallest eigenvalue of B_α . Notice also that the eigenvalues of H and B_α are related. The Cauchy Interlace Theorem (cf. [39]) establishes that the eigenvalues of B_α interlace the eigenvalues of H . In particular, the smallest eigenvalue of B_α is a lower bound for the smallest eigenvalue of H . This implies that solving an eigenvalue problem for the smallest eigenvalue of B_α and a corresponding eigenvector with non-zero first component yields x and λ that automatically satisfy the first two optimality conditions in Lemma 1.1 for any value of α .

These facts suggest designing an iteration for computing α based on the solution of eigenvalue problems for B_α . The methods in [11, 41, 43, 44, 50] propose such iterations. The semidefinite programming approach used in [11, 41] works mainly on a primal problem switching to the dual when a difficult situation (the so-called *hard case*) is detected. The method in [50] also switches iterations in the presence of the hard case. LSTRS is a unified iteration that incorporates all cases. In all of the methods above, the main computation per iteration is the solution of an eigenvalue problem for an eigenpair associated with the smallest eigenvalue of B_α . In LSTRS, an additional eigenpair (corresponding to another eigenvalue) is needed. In both families of methods, the eigenpairs are used to update α by means of rational interpolation and similar safeguarding strategies are used to ensure global convergence of the iteration. We refer the reader to [11, 41, 43, 44] for more details about the theory and computational aspects of this kind of TRS methods.

Figure 1 shows the LSTRS algorithm, where δ_1 denotes the smallest eigenvalue of H , and $\lambda_1(\alpha)$ and $\lambda_i(\alpha)$ denote the smallest eigenvalue and the i th eigenvalue of B_α , respectively. Note that indeed the eigenvalue computation is the main cost of the iteration. We discuss the features and challenges associated with this eigenvalue computation in Section 3.

Input: $H \in \mathbb{R}^{n \times n}$, symmetric; $g \in \mathbb{R}^n$; $\Delta > 0$; tolerances $(\epsilon_\Delta, \epsilon_{HC}, \epsilon_{Int}, \epsilon_\nu, \epsilon_\alpha)$.
Output: x^* , solution to TRS and Lagrange multiplier λ^* .

```

1: Initialization
2:   Compute  $\delta_U \geq \delta_1$ , initialize  $\alpha_U$  and  $\alpha_0$ , set  $k = 0$     %  $\alpha_U \geq \alpha_k$ 
3:   Compute eigenpairs  $\{\lambda_1(\alpha_0), (\nu_1, u_1^T)^T\}$ , and  $\{\lambda_i(\alpha_0), (\nu_2, u_2^T)^T\}$  of  $B_{\alpha_0}$ 
4:   Initialize  $\alpha_L$     %  $\alpha_L \leq \alpha_k$ 
5: repeat
6:   Adjust  $\alpha_k$  (might need to compute eigenpairs)
7:   Update  $\delta_U = \min \left\{ \delta_U, \frac{u_1^T A u_1}{u_1^T u_1} \right\}$ 
8:   if  $\|g\| |\nu_1| > \epsilon_\nu \sqrt{1 - \nu_1^2}$  then
9:     Set  $\lambda_k = \lambda_1(\alpha_k)$ ,  $x_k = \frac{u_1}{\nu_1}$ , and update  $\alpha_L$  or  $\alpha_U$ 
10:  else
11:    Set  $\lambda_k = \lambda_i(\alpha_k)$ ,  $x_k = \frac{u_2}{\nu_2}$ , and  $\alpha_U = \alpha_k$ 
12:  end if
13:  Compute  $\alpha_{k+1}$  by 1-point ( $k = 0$ ) or 2-point interpolation scheme
14:  Safeguard  $\alpha_{k+1}$  and set  $k = k + 1$ 
15:  Compute eigenpairs  $\{\lambda_1(\alpha_k), (\nu_1, u_1^T)^T\}$ , and  $\{\lambda_i(\alpha_k), (\nu_2, u_2^T)^T\}$  of  $B_{\alpha_k}$ 
16: until convergence

```

Figure 1: The LSTRS Method.

3 Eigenvalue Problems in TRS Methods

We consider the parametric eigenvalue problem:

$$\begin{pmatrix} \alpha_k & g^T \\ g & H \end{pmatrix} \begin{pmatrix} \nu \\ u \end{pmatrix} = \lambda \begin{pmatrix} \nu \\ u \end{pmatrix} \quad (5)$$

with H, g as above, and α_k a real parameter that is iteratively updated such that $\{\alpha_k\}$ is a convergent sequence. As before, we assume that H is large, that it might not be explicitly available, and that matrix-vector products with H can be efficiently computed. We are interested in solving (5) for the algebraically smallest eigenvalue and a corresponding eigenvector with non-zero first component.

Several methods exist for the efficient solution of large-scale symmetric eigenvalue problems such as (5). We mention three approaches: the Implicitly Restarted Lanczos Method (IRLM) [28, 49], the Nonlinear Lanczos (Arnoldi) Method (NLLM) [55], and the Jacobi-Davidson Method [47]. All methods are matrix-free in the sense that they rely on matrix-vector multiplications only.

The main features of the IRLM include: limited-memory, the ability to compute several eigenpairs at a time, and the possibility of choosing *one* initial vector. Features of the NLLM include: limited-memory through the use of restarts, the ability to compute only one eigenpair at a time, the possibility of choosing *several* initial vectors, and the possibility of incorporating preconditioning. The Jacobi-Davidson method is similar to the NLLM. Both the IRLM and the NLLM have been successfully used in the context of LSTRS and of [11, 41]. The performance of the Jacobi-Davidson method in the context of trust-region methods is yet to be studied.

To efficiently solve problems of type (5) arising in TRS methods, an eigensolver must be able to handle the special features of these problems. Some of the computational issues associated with the solution of the eigenproblems, along with the strategies used in LSTRS, are discussed in Sections 3.1 through 3.3.

3.1 Eigenvalues close to zero

The solution of (5) is particularly challenging for methods based on matrix-vector multiplications when the eigenvalues of interest are close to zero. In this case, every matrix-vector product will annihilate components of the resulting vector precisely in the desired directions. In regularization, it is often the case that the eigenvalues of interest are close to zero.

In [44], this situation is handled by means of a Tchebyshev Spectral Transformation. Namely, we construct a Tchebyshev polynomial T that is as large as possible on λ_1 and as small as possible on an interval containing the remaining eigenvalues of B_α . We then compute the eigenvalues of $T(B_\alpha)$ instead of the eigenvalues of B_α . In LSTRS, a polynomial of degree ten is used. Hence, the number of matrix-vector products increases accordingly. However, the convergence of the IRLM is usually enhanced in this way and in the context of LSTRS for regularization this is often the only way to handle certain challenging ill-posed problems (cf. [45]). After convergence, the eigenvalues of B_α are recovered via Rayleigh quotients with the converged eigenvectors. No special strategy is used to handle this case when the NLLM is used as eigensolver.

3.2 Clustered eigenvalues

In regularization problems, the singular values of A are usually clustered and very close to zero with no apparent gap (cf. [17]). The eigenvalues of $H = A^T A$ inherit this feature. The interlacing properties discussed before imply that if the smallest k eigenvalues of H are small and clustered then, eigenvalues 2 through k of B_α will also be small and clustered.

The situation for λ_1 , the smallest eigenvalue of B_α , is as follows. Recall that λ_1 is a lower bound for δ_1 , the smallest eigenvalue of H . The distance between λ_1 and δ_1 depends on the value of α , which in turn

depends on the value of Δ . For values of Δ smaller than a certain critical value, the smallest eigenvalue of B_α is well-separated from the rest and Lanczos-type methods can compute it very efficiently. For larger values of Δ , λ_1 is well-separated from δ_1 only at early iterations. As the process converges (and α_k approaches the optimal value), λ_1 gets closer to δ_1 and, in regularization, to the cluster. Figure 2 illustrates this situation for a test problem from [18]. The figure shows the eigenvalues of H and B_α for the optimal value of α , for three trust-region subproblems differing only in the value of Δ . We can observe that for Δ small (top plot), λ_1 is well separated from δ_1 . Increasing Δ makes the gap between λ_1 and δ_1 decrease (middle plot). For large Δ , λ_1 and δ_1 are indistinguishable (bottom plot).

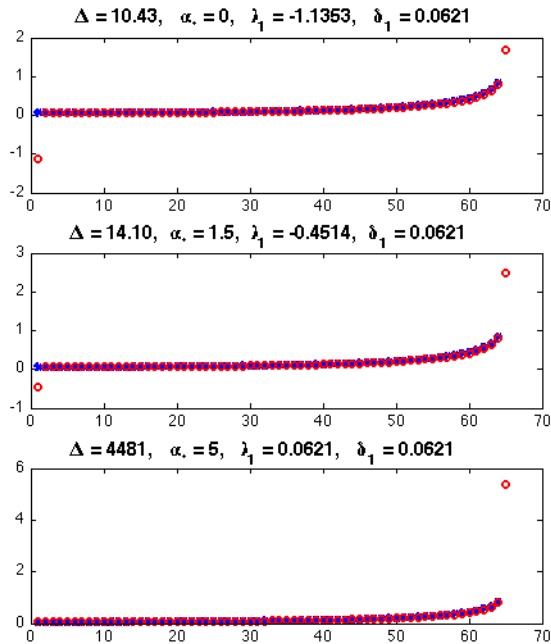


Figure 2: Eigenvalues of H (dot) and B_α (circle) for different values of Δ (and α_*). Problem **heat** from [18].

It is often the case in regularization that Δ exceeds the critical value that leads to the cluster situation. In practice, this often means that the number of vectors required by the IRLM or by the NLLM must be increased. This is the only strategy followed at this moment in LSTRS.

3.3 Efficiency

We now discuss the performance of LSTRS in terms of number of matrix-vector products (MVP). Comparisons of LSTRS with other state-of-the-art methods for large-scale trust-region subproblems seem to indicate an advantage for LSTRS [44], especially for regularization problems. This was to be expected since LSTRS was designed with focus on this kind of problems. Recently [27], significant reductions in the number of MVP have been obtained at a moderate cost in storage by means of the NLLM. Preliminary results in [27] indicate that the performance of [11, 41] can also improve significantly by using the NLLM. We expect that further improvements are possible, for example, by incorporating preconditioning. This is the subject of current research.

4 Sensitivity of TRS solutions

In TRS methods such as LSTRS, which are based on the solution of parametric eigenvalue problems of type (5), the eigenvalue problems are embedded in an outer iteration (see Figure 1). Hence, a relevant question is how accurate the eigenvalue problems must be solved in order for the outer iteration to converge. Note that theoretical convergence is not an issue in LSTRS, since both global and local superlinear convergence are proven features (cf. [43]). However, the practical convergence speed is more interesting for practitioners. In general, we would like to solve the eigenvalue problems as fast and accurately as possible while maintaining fast (practical) convergence of the outer iteration. This issue is particularly relevant in the large-scale case in which iterative (inexact) methods must be used to solve the eigenvalue problems. In this section, we present a numerical study designed to investigate how sensitive the trust-region solution is to random perturbations of exact eigenpairs. The sensitivity properties of the solution would indicate if the eigenproblems must be solved very accurately or not. We also investigate the effect of eigenpair perturbations on the performance of LSTRS. Our study is an extension of the one presented in [12]. Related numerical investigations involving random perturbations can be found in [8, 14, 13, 32, 52].

The sensitivity study was carried out in MATLAB R2009b on a MacBookPro with a 2.66 GHz processor and 4 GB of RAM, running Mac OS X version 10.6.8 (Snow Leopard). The floating-point arithmetic was IEEE standard double precision with machine precision $2^{-52} \approx 2.2204 \times 10^{-16}$.

Our study consisted of a set of experiments on regularization problems of type (2) where A and b were taken from the test set [18]. Regularization problems were chosen since as mentioned before, they usually yield very difficult TRS. The experiments consisted of adding random perturbations to the exact eigenvalue or eigenvector computed at each iteration of LSTRS. An *exact* eigenvalue or eigenvector is one computed to working precision. In MATLAB, these were computed with the routine `eig` (QR method). Both absolute and relative perturbations were used as well as two distributions (uniform and Gaussian). No further assumptions were made on the stochastic properties of the perturbations, as they might not be valid in general (cf. [6, 19, 32]). Note that the goal of the experiments was not to simulate roundoff error, but rather the *approximation* error incurred when an *exact* computation (within working precision) is replaced by an *inexact* one.

Instances of problem (2) were solved for A , an $m \times n$ matrix and b , an m -dimensional vector such that $b = b_o + s$, with $b_o = Ax_o$, x_o the desired solution available from the test set, and s a vector of Gaussian noise. The values of Δ as well as some of the LSTRS settings were chosen to favor boundary solutions for the TRS. The following three eigensolvers, which are available in LSTRS, were tested: `eig` (QR method), `eigs` (MATLAB's interface to ARPACK), and `tcheigs` (`eigs` combined with a Tchebyshev spectral transformation). In Sections 4.1 and 4.2, we report results for problem `shaw` from [18] with $m = n = 100$ and a fixed Gaussian noise vector with noise level $\|s\|/\|b\|$ equal to 10^{-2} . Experiments with other problems from the test set and other noise vectors yielded similar results. The results are discussed in Section 4.3.

In the remainder of this section, x_p and x_u denote the solutions computed by LSTRS using perturbed and unperturbed eigenvalues (or eigenvectors), respectively. E_p denotes the relative error $\frac{\|x_p - x_u\|}{\|x_u\|}$.

4.1 Eigenvalue Perturbations

In the first set of experiments, we solved the eigenvalue problems with MATLAB's `eig` routine and then perturbed the eigenvalue at each LSTRS iteration in the following way. Given an (unperturbed) eigenvalue λ_u and a perturbation level $\varepsilon \in (0, 1)$, a perturbed eigenvalue λ_p was constructed as follows:

- $\lambda_p = \lambda_u + \varepsilon\rho$, for *absolute* perturbations

- $\lambda_p = \lambda_u(1 + \varepsilon\rho)$, for *relative* perturbations

where ρ was a random number in $(-1/2, 1/2)$ for uniform distribution and in $(-1, 1)$ for Gaussian. We ran 100 examples corresponding to different perturbations ρ_i , $i = 1, \dots, 100$ sorted in increasing order. We used the following values for ε : 10^{-5} , 10^{-4} , 10^{-3} , 10^{-2} , and 10^{-1} .

Figure 3 shows results for $\varepsilon = 10^{-2}$, and absolute perturbations with uniform distribution. Note that this is equivalent to relative perturbations with a perturbation level of order one. To see this, put $\lambda_u + \varepsilon_A\rho = \lambda_u(1 + \varepsilon_R\rho)$, fix either ε_A (perturbation level for absolute approach) or ε_R (perturbation level for relative approach), and solve for the other. In plot (a), we can observe an apparent linear behavior of $\|x_p\|$ with respect to the perturbations. Plot (b) shows the difference between $\|x_p\|$ and $\|x_u\|$. In plot (c), we show the relative error E_p together with the relative error of x_{inex} (a solution computed by LSTRS using `eigs` as eigensolver) and x_u . This plot shows that the relative error in x_p is of the same order of the relative error in x_{inex} . Thus indicating that, for this perturbation level ε , the random perturbations seem to accurately simulate the approximation error introduced by an inexact eigensolver. Results for relative perturbations were similar as were results for Gaussian distribution.

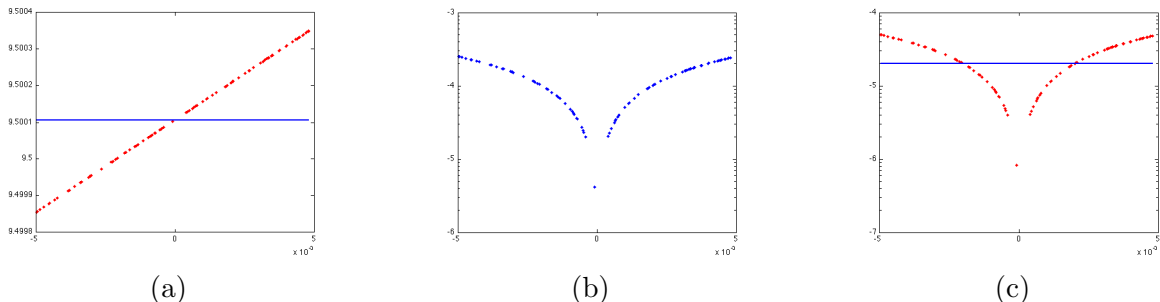


Figure 3: Eigenvalue perturbations. (a) $\|x_p\|$ (dotted) and $\|x_u\|$ (solid); (b) $|\|x_u\| - \|x_p\||$ in logarithmic scale; (c) relative errors $\|x_p - x_u\|/\|x_u\|$ (dotted) and $\|x_{inex} - x_u\|/\|x_u\|$ (solid) in logarithmic scale. On the x -axis, perturbations.

We also conducted a second set of experiments seeking to simulate the solution of the eigenproblems to different accuracy by means of random perturbations. For this purpose, the inexact eigensolvers `eigs` and `tcheigs` were used as eigensolvers in LSTRS. As before, perturbations were introduced in the eigenvalue. Both absolute and relative perturbations were used as well as two distributions (uniform and Gaussian).

Figure 4 presents LSTRS performance results for all eigensolvers. We report: the perturbation level (ε); the average number of (outer) LSTRS iterations (IT); the average number of eigenvalue problems solved (EV); and the maximum value of the relative error in x_p with respect to x_u ($\max E_p$). For `eigs` and `tcheigs`, the average number of matrix-vector products (MVP) is also reported.

The maximum values of E_p (last column in all tables) show that the solution of the TRS based on perturbed eigenvalues remains close to the solution based on unperturbed ones for all perturbation levels and that E_p is of lower order than the perturbation level ε . Regarding performance, measured in MVP, we observe that this was essentially not affected by the perturbations, except for the largest perturbation level $\varepsilon = 10^{-1}$. In this case, we observe an increase in MVP of approximately 2% for `eigs` and 20% for `tcheigs`, with respect to the unperturbed case. Also for $\varepsilon = 10^{-1}$, we observe an increase of approximately 25% in the number of iterations for all eigensolvers. Note that similar results were obtained for relative perturbations and for Gaussian distribution. Figure 4 shows results for absolute perturbations with uniform distribution, which were the worst results in terms of $\max E_p$.

ε	IT	EV	max E_p
0	7.0	8.0	0
10^{-5}	7.0	8.0	4.98e-08
10^{-4}	7.0	8.0	4.98e-07
10^{-3}	7.0	8.0	4.98e-06
10^{-2}	7.0	8.0	4.98e-05
10^{-1}	8.7	9.7	2.07e-04

eig

ε	IT	EV	MVP	max E_p
0	7.0	8.0	542.0	0
10^{-5}	7.0	8.0	542.0	4.98e-08
10^{-4}	7.0	8.0	542.0	4.98e-07
10^{-3}	7.0	8.0	542.0	4.98e-06
10^{-2}	7.0	8.0	543.0	4.98e-05
10^{-1}	8.8	9.8	556.5	2.11e-04

eigs

ε	IT	EV	MVP	max E_p
0	7.0	8.0	564.0	0
10^{-5}	7.0	8.0	564.0	4.98e-08
10^{-4}	7.0	8.0	564.0	4.98e-07
10^{-3}	7.0	8.0	564.0	4.98e-06
10^{-2}	7.0	8.0	564.6	4.98e-05
10^{-1}	8.8	9.8	688.2	2.11e-04

tcheigs

Figure 4: Eigenvalue perturbations. Performance results and maximum relative error in x_p with respect to x_u .

4.2 Eigenvector Perturbations

As in Section 4.1, we performed a first set of experiments in which we solved the eigenvalue problems with MATLAB’s `eig` routine and then perturbed the eigenvector computed at each LSTRS iteration. Given an (unperturbed) eigenvector y_u and a perturbation level $\varepsilon \in (0, 1)$, a perturbed eigenvector y_p was constructed as follows:

- $y_p = y_u + \varepsilon \rho r$, for *absolute* perturbations
- $y_p = y_u \times (e + \varepsilon \rho r)$, for *relative* perturbations

with ρ a random number in $(-1/2, 1/2)$ for uniform distribution and in $(-1, 1)$ for Gaussian; e the vector of all ones; and \times denoting entry-wise multiplication. The vector r was a fixed random vector normalized such that $\|r\| = 1$ and whose entries had uniform distribution in $(0, 1)$ if ρ was uniformly distributed, or Gaussian distribution if ρ was Gaussian.

We ran 100 examples corresponding to different perturbations ρ_i , $i = 1, \dots, 100$ sorted in increasing order. We used the following values for ε : 10^{-5} , 10^{-4} , 10^{-3} , 10^{-2} , and 10^{-1} .

Figure 5 shows results for $\varepsilon = 10^{-5}$, and absolute perturbations with uniform distribution. In plot (a), we can again observe the linear behavior of $\|x_p\|$ with respect to the perturbations. Plot (b) shows the difference between $\|x_p\|$ and $\|x_u\|$. In plot (c), we show the relative error E_p together with the relative error of x_{inex} (a solution computed by LSTRS using `eigs` as eigensolver) and x_u . This plot shows that, for this perturbation level ε , the relative error in x_p is of the same order as the relative error in x_{inex} . As before, results for relative perturbations were similar as were results for Gaussian distribution.

Note that the same experiment was performed using different normalized random vectors r_i . Similar results were obtained in this case, although a larger number of examples (500 instead of 100) was needed to expose the trends.

The results of the second set of experiments, where we also introduced random perturbations in the eigenvectors computed with iterative (inexact) eigensolvers, are shown in Figure 6 where we report: the perturbation level (ε); the average number of (outer) LSTRS iterations (IT); the average number of eigenvalue problems solved (EV); and the maximum value of the relative error in x_p with respect to x_u (max E_p). For `eigs` and `tcheigs`, the average number of matrix-vector products (MVP) is also reported.

Figure 6 shows that the relative error E_p is of the same order as or of lower order than the perturbation level ε . Regarding performance, measured in MVP, we observe the following. For the eigensolvers `eig` and `tcheigs` and as in Section 4.1, performance was essentially not affected by perturbations, except for $\varepsilon = 10^{-1}$. In this case, we observe an increase of approximately 25% in the number of iterations (`eig`), and between 7% and 80% increase in MVP (`tcheigs`), with respect to the unperturbed case. For `eigs`, we observe MVP of between 2 and 5 times those of the unperturbed case. Note also that the deterioration

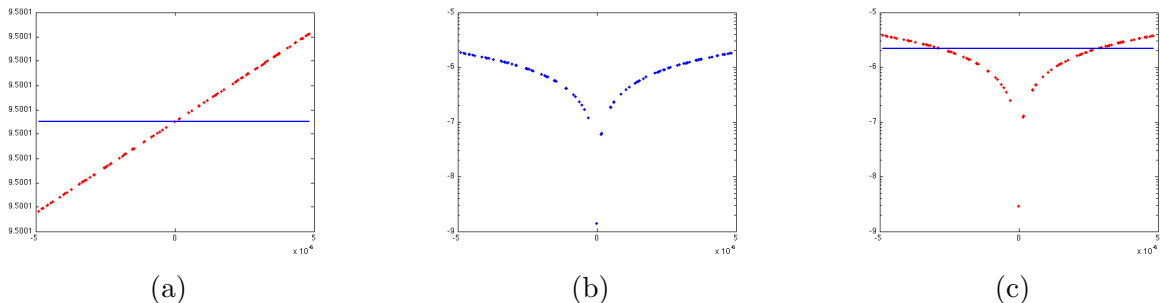


Figure 5: Eigenvector perturbations. (a) $\|x_p\|$ (dotted) and $\|x_u\|$ (solid); (b) $|\|x_u\| - \|x_p\||$ in logarithmic scale; (c) relative errors $\|x_p - x_u\|/\|x_u\|$ (dotted) and $\|x_{inex} - x_u\|/\|x_u\|$ (solid) in logarithmic scale. On the x -axis, perturbations.

in performance begins at a smaller perturbation level ($\varepsilon = 10^{-4}$) than in Section 4.1. The mild effect of perturbations when `eig` and `tcheigs` are used as eigensolvers can be accounted for by the fact that these eigensolvers perform very accurate calculations, whereas the eigenpairs computed by `eigs` are *more inexact*. Note that similar results were obtained for relative perturbations and for Gaussian distribution. Figure 4 shows the results of the experiment for absolute perturbations with uniform distribution, which were the worst results in terms of $\max E_p$.

ε	IT	EV	$\max E_p$
0	7.0	8.0	0
10^{-5}	7.0	8.0	3.79e-06
10^{-4}	7.0	8.0	3.79e-05
10^{-3}	7.0	8.0	1.12e-03
10^{-2}	7.0	8.0	1.03e-02
10^{-1}	8.7	9.7	8.40e-02

`eig`

ε	IT	EV	MVP	$\max E_p$
0	7.0	8.0	542.0	0
10^{-5}	7.0	8.0	542.0	6.58e-06
10^{-4}	7.4	9.5	1270.1	2.19e-04
10^{-3}	7.8	10.0	1436.8	1.06e-03
10^{-2}	9.8	12.2	1962.8	5.36e-03
10^{-1}	13.3	16.3	2844.2	8.06e-02

`eigs`

ε	IT	EV	MVP	$\max E_p$
0	7.0	8.0	564.0	0
10^{-5}	7.0	8.0	564.0	6.58e-06
10^{-4}	7.0	8.0	564.9	6.55e-05
10^{-3}	6.9	7.9	556.4	8.45e-04
10^{-2}	8.3	8.6	605.4	9.84e-03
10^{-1}	14.5	14.6	1030.5	6.56e-02

`tcheigs`

Figure 6: Eigenvector perturbations. Performance results and maximum relative error in x_p with respect to x_u .

4.3 Discussion

The results in Section 4.1 seem to indicate that LSTRS solutions are not significantly affected by perturbations in the eigenvalues computed with any of the available eigensolvers. In particular, Figure 4 shows that the relative error E_p remains small and is of lower order than the perturbation level. The results of these experiments seem to indicate that the effect of eigenvalue perturbations on both accuracy and performance of LSTRS is limited and therefore, that the method is stable with respect to perturbations in the eigenvalues.

In Section 4.2, similar results in terms of E_p were obtained for eigenvector perturbations. However, in that case the maximum E_p was larger than for eigenvalue perturbations. We also observed a dramatic deterioration in performance when eigenvector perturbations were introduced in combination with the inexact eigensolver `eigs`. This was to be expected since the eigenvector y influences the TRS solution in a more direct way than the eigenvalue. Namely, the last n (normalized) components of y form the approximate solution to problem (1) at each iteration of LSTRS. In particular, in the last LSTRS iteration the last n (normalized) components of y form the solution. By introducing perturbations in the

eigenvector, we are effectively changing the execution “path” of the algorithm. It is thus most remarkable and a strong indication of robustness, that even with this expected deterioration in performance and the fact that the safeguarding mechanisms were rarely triggered in these experiments, the final solution does not deviate excessively from the unperturbed solution.

5 Concluding Remarks

We considered computational and sensitivity issues arising in eigenvalue-based methods for the large-scale trust-region subproblem using the LSTRS method as a case study. We described the eigenvalue problems and their relevant features for computational purposes, emphasizing challenges and strategies for dealing with them. A main aspect of this work is a numerical sensitivity study of the TRS solution with respect to the eigenvalue solutions. The results of this study seem to indicate that LSTRS is stable with respect to eigenpair perturbations and that the eigenvalue problems can be solved to low accuracy. Current and future work in this area includes the use of preconditioning, further experiments concerning sensitivity, and the design of strategies to guarantee the computation of all the desired eigenpairs and of at least one eigenvector with the desired structure. A similar study for other eigenvalue-based methods for the large-scale TRS such as [11, 41] is yet to be done.

Acknowledgments. M. Rojas would like to thank Professor Kees Roos and his group in the Software Technology Department at the Delft University of Technology in The Netherlands for his hospitality during several visits from 2006 through 2009.

References

- [1] R. Andreani, E.G. Birgin, J.M. Martínez, and M.L. Schuverdt. On augmented Lagrangian methods with general lower-level constraints. *SIAM J. Optimiz.*, 18(4):1286–1309, 2007.
- [2] M. Andretta, E.G. Birgin, and J.M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization. *Optimization*, 54:305–325, 2005.
- [3] A. Barbero and S. Sra. Fast algorithms for total-variation based optimization. Technical Report 194, Max Planck Institute for Biological Cybernetics, Tübingen, Germany August 2010.
- [4] Ø. Bergmann and T. Steihaug. Solving a TRS which has linear inequality constraints. *Optim. Method. Softw.* DOI: 10.1080/10556788.2011.582501, 2011.
- [5] E.G. Birgin, J.M. Martínez, and M. Raydan. Algorithm 813: SPG – Software for Convex-Constrained Optimization. *ACM T. Math. Software*, 27(3):340–349, 2001.
- [6] F. Chaitin-Chatelin and V. Frayssé. *Lectures on Finite Precision Computations*. SIAM, Philadelphia, 1996.
- [7] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *Trust Region Methods*. SIAM, Philadelphia, 2000.
- [8] J.J. Dongarra, C.B. Moler, and J.H. Wilkinson. Improving the accuracy of computed eigenvalues and eigenvectors. *SIAM J. Numer. Anal.*, 20(1):23–45, 1983.
- [9] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17:134–145, 1977.

- [10] A.P. Eriksson, C. Olsson, and F. Kahl. Image segmentation with context. *Lect. Notes Comp. Sci.*, 4522:283–292, 2007.
- [11] C. Fortin and H. Wolkowicz. The trust-region subproblem and semidefinite programming. *Optim. Method. Softw.*, 19(1):41–67, 2004.
- [12] B.H. Fotland. A matrix-free method for the regularisation with unrestricted variables. Master’s thesis, Department of Informatics, University of Bergen, Bergen, Norway, June 2008.
- [13] G.H. Golub and H. Melbø. Stochastic error estimates for iterative linear solvers: part 2. Unpublished report, 2000.
- [14] G.H. Golub and H. Melbø. A stochastic approach to error estimates for iterative linear solvers: part 1. *BIT*, 41(5):977–985, 2001.
- [15] N.I.M. Gould, S. Lucidi, M. Roma, and Ph.L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM J. Optimiz.*, 9(2):504–525, 1999.
- [16] W.W. Hager. Minimizing a quadratic over a sphere. *SIAM J. Optimiz.*, 12:188–208, 2001.
- [17] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.
- [18] P.C. Hansen. Regularization Tools version 4.0 for Matlab 7.3. *Numer. Algo.*, 46:189–194, 2007.
- [19] N.J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 2nd edition, 2002.
- [20] M.E. Hochstenbach, N. McNinch, and L. Reichel. Discrete ill-posed least-squares problems with a solution norm constraint. CASA-Report 11-37, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, June 2011.
- [21] E.S. Hons, A.K. Khandani, and W. Tong. An optimized transmitter precoding scheme for synchronous DS-CDMA. *IEEE T. Commun.*, 54(1):32–36, 2006.
- [22] N. Hurley and M. Zhang. Analysis of methods for novel case selection. In *20th IEEE International Conference on Tools with Artificial Intelligence, ICTAI ’08*, pages 217–224, 2008.
- [23] S. Kosinov and S. Marchand-Maillet. Evaluation of distance-based discriminant analysis and its kernelized extension in visual object recognition. In *Proceedings of the 7th International on signal/image processing and pattern recognition*, 2004.
- [24] S. Kosinov, S. Marchand-Maillet, and T. Pun. Iterative majorization approach to the distance-based discriminant analysis. In C. Weihs and W. Gaul, editors, *Studies in Classification, Data Analysis, and Knowledge Organization. Classification - the Ubiquitous Challenge, Proceedings of the 28th Annual Conference of the Gesellschaft für Klassifikation e.V.*, pages 168–175, 2004.
- [25] S. Kosinov and T. Pun. Distance-based discriminant analysis method and its applications. *Pattern Anal. Applic.*, 11:227–246, 2008.
- [26] J. Lampe. Solving regularized total least squares problems based on eigenproblems. Ph.D. thesis, Department of Applied Mathematics, Technical University of Hamburg, Hamburg-Harburg, 2010.
- [27] J. Lampe, M. Rojas, D.C. Sorensen, and H. Voss. Accelerating the LSTRS algorithm. *SIAM J. Sci. Comput.*, 33(1):175–194, 2011.

- [28] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998.
- [29] L. Luksan, M. Tuma, J. Hartman, J. Vlcek, N. Ramesova, M. Siska, and C. Matonoha. Ufo 2006 interactive system for universal functional optimization. Technical Report 977, Institute of Computer Science Academy of Sciences of the Czech Republic, Prague, December 2006.
- [30] J.L. Mead and R.A. Renaut. Least squares problems with inequality constraints as quadratic constraints. *Linear Algebra Appl.*, 432(8):1936–1949, 2008.
- [31] J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comp.*, 4(3):553–572, 1983.
- [32] J.J. Moré and Wild. Estimating computational noise. *SIAM J. Sci. Comput.*, 33(3):1292–1314, 2011.
- [33] S. Morigi, R. Plemmons, L. Reichel, and F. Sgallari. A hybrid multilevel-active set method for large box-constrained linear discrete ill-posed problems. *Calcolo*, 48(1):89–105, 2009.
- [34] G.P. Nava. Inverse sound rendering : In-situ estimation of surface acoustic impedance for acoustic simulation and design of real indoor environments. Ph.D. thesis, Graduate School of Information Science and Technology, University of Tokyo, Japan, 2006.
- [35] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
- [36] C. Olsson, A.P. Eriksson, and F. Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. IEEE Conference on Computer Vision and Pattern Recognition, 2007.
- [37] C. Olsson, A.P. Eriksson, and F. Kahl. Improved spectral relaxation methods for binary quadratic optimization problems. *Comput. Vis. Image. Und.*, 112(1):3–13, 2008.
- [38] Y.C. Park, M.H. Chang, and T.-Y. Lee. A new deterministic global optimization method for general twice-differentiable constrained nonlinear programming problems. *Eng. Optimiz.*, 39(4):397–411, 2007.
- [39] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs NY, 1980.
- [40] D.T. Phan. Lagrangian duality-based branch and bound algorithms for optimal power flow. IBM Research Report RC25142 (W1104-042), Mathematics, April 2011.
- [41] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Program.*, 77(2):273–299, 1997.
- [42] M. Rojas. A large-scale trust-region approach to the regularization of discrete ill-posed problems. Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, Technical Report TR98-19, May 1998.
- [43] M. Rojas, S.A. Santos, and D.C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM J. Optimiz.*, 11(3):611–646, 2000.
- [44] M. Rojas, S.A. Santos, and D.C. Sorensen. Algorithm 873: LSTRS: MATLAB software for large-scale trust-region subproblems and regularization. *ACM T. Math. Software*, 34(2):11, 2008.
- [45] M. Rojas and D.C. Sorensen. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. *SIAM J. Sci. Comput.*, 23(3):1843–1861, 2002.

- [46] M. Rojas and T. Steihaug. An interior-point trust-region-based method for large-scale non-negative regularization. *Inverse Probl.*, 18:1291–1307, 2002.
- [47] G.L.G. Sleijpen and H.A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. A.*, 17:401–425, 1996.
- [48] D.C. Sorensen. Newton’s method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2):409–426, 1982.
- [49] D.C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. A.*, 13(1):357–385, 1992.
- [50] D.C. Sorensen. Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM J. Optimiz.*, 7(1):141–161, 1997.
- [51] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.
- [52] G.W. Stewart. Stochastic perturbation theory. *SIAM Rev.*, 32(4):579–610, 1990.
- [53] A.N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math.*, 4:1624–1627, 1963.
- [54] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems*. Halsted Press, New York, 1977.
- [55] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT*, 44:387–401, 2004.
- [56] Y. Wang and S. Ma. A fast subspace method for image deblurring. *Appl. Math. Comput.*, 215(6):2359–2377, 2009.
- [57] R. Zdunek and A. Cichocki. Nonnegative matrix factorization with quadratic programming. *Neurocomputing*, 71(10–12):2309–2320, 2008.
- [58] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender Systems*, RecSys ’08, pages 123–130, New York, NY, USA, 2008. ACM.