

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 [?].

Lemma 1.1 (see [?]). *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 [?]. □

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

*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).

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. [?, ?, ?]) that (??) is equivalent to Tikhonov regularization [?, ?], with the Lagrange multiplier associated with the norm constraint corresponding to the Tikhonov regularization parameter. The regularization of linear problems by means of (??) 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 (??). The TRS in regularization is usually a very challenging problem owing to the presence of high-degree singularities (cf. [?, ?]). 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 (??) was proposed in [?] (see also [?]). 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 [?] 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 [?], 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 [?] is a so-called *approximate* technique since it does not aim to satisfy the optimality conditions in Lemma ???. 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 [?].

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

In this work, we focus on the eigenvalue-based techniques which include [?, ?, ?, ?, ?]. In particular, we study computational and sensitivity issues for the LSTRS method [?, ?]. 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. [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]). TRUST_μ , which is based on LSTRS, has also been used in applications and as guideline for developing new methods (cf. [?, ?, ?, ?]). 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 (??) and (??). Therefore, in this work we focus on computational and sensitivity aspects associated with these eigenvalue problems. The presentation is organized as follows. In Section ??, we briefly describe eigenvalue-based TRS methods. In Section ??, 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 ??, we present a numerical sensitivity study of LSTRS solutions with respect to the eigenproblem solutions. Section ?? contains concluding remarks.

2 Eigenvalue-Based TRS Methods

One approach for developing large-scale methods for solving (??) is based on the following fact. It can be shown (see [?]) that there always exists an optimal value of a scalar parameter α such that a solution x to (??) 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 (??) 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. [?]) 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 ?? 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 [?, ?, ?, ?, ?] propose such iterations. The semidefinite programming approach used in [?, ?] works mainly on a primal problem switching to the dual when a difficult situation (the so-called *hard case*) is detected. The method in [?] 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 [?, ?, ?, ?] for more details about the theory and computational aspects of this kind of TRS methods.

Figure ?? 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 ??.

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) [?, ?], the Nonlinear Lanczos (Arnoldi) Method (NLLM) [?], and the Jacobi-Davidson Method [?]. 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 [?, ?]. 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 ?? through ??.

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 [?], 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. [?]). 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. [?]). 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 ?? illustrates this situation for a test problem from [?]. 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).

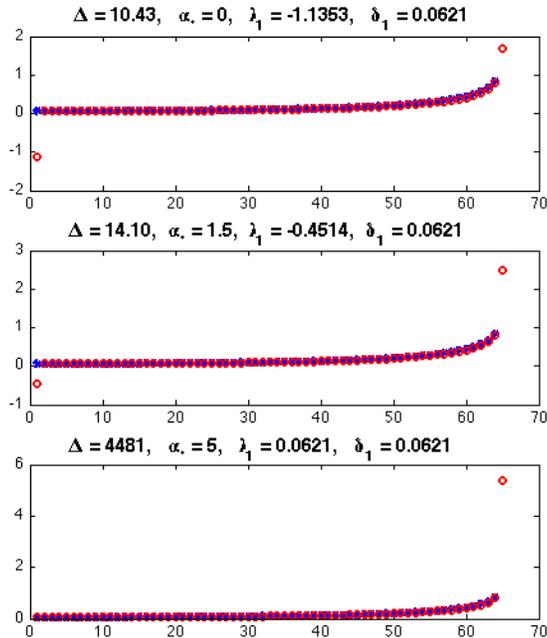


Figure 2: Eigenvalues of H (dot) and B_α (circle) for different values of Δ (and α_*). Problem **heat** from [?].

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 [?], especially for regularization problems. This was to be expected since LSTRS was designed with focus on this kind of problems. Recently [?], significant reductions in the number of MVP have been obtained at a moderate cost in storage by means of the NLLM. Preliminary results in [?] indicate that the performance of [?, ?] 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 (??), the eigenvalue problems are embedded in an outer iteration (see Figure ??). 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. [?]). 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 [?]. Related numerical investigations involving random perturbations can be found in [?, ?, ?, ?, ?].

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 (??) where A and b were taken from the test set [?]. 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. [?, ?, ?]). 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 (??) 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 ?? and ??, we report results for problem **shaw** from [?] 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 ??.

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