DELFT UNIVERSITY OF TECHNOLOGY

REPORT 13-01

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

Marielba Rojas, Bjørn H. Fotland, and Trond Steihaug

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2013

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

Marielba Rojas* Bjørn H. Fotland† Trond Steihaug‡

**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:

$$\begin{array}{ll} \min & \frac{1}{2}x^T H x + g^T x \ , \\ s.t. & \|x\| \leq \Delta \end{array} \tag{1}$$

where $H$ is an $n \times n$ real, symmetric matrix, $g$ is an $n$-dimensional vector, $\Delta$ 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 [41].

**Lemma 1.1** ([41]). *A feasible vector $x_* \in \mathbb{R}^n$ is a solution to (1) with corresponding Lagrange multiplier $\lambda_*$ if and only if $x_*, \lambda_*$ satisfy $(H - \lambda_* I)x_* = -g$, $H - \lambda_* I$ positive semidefinite, $\lambda_* \leq 0$, and $\lambda_*(\Delta - \|x_*\|) = 0$.*

**Proof.** See [41]. □

A special situation that is particularly challenging in practice is the so-called *hard case* defined as follows. If $\delta_1$ is the smallest eigenvalue of $H$ and $\mathcal{S}_1$ is the corresponding eigenspace, then the hard case is present when $H$ is indefinite, $g$ is orthogonal to $\mathcal{S}_1$, and $\Delta < \|(H - \delta_1 I)^\dagger g\|$. In the hard case, the solution to the TRS is not unique. When $g \perp \mathcal{S}_1$ this is called a *potential hard-case*.

Problem (1) is known in optimization as the trust-region subproblem (TRS) arising in the widely-used, state-of-the-art trust-region methods [6]. 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:

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

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. [8, 34, 37]) that (2) is equivalent to Tikhonov regularization [45, 46], 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 in the form of multiple instances of the hard case or potential hard case (cf. [34, 37]). 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:

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

The interior-point method TRUST$_\mu$ for solving (3) was proposed in [38]. 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.

Several methods have been proposed for solving the large-scale TRS (see [6, 36] and the references therein). In this work, we focus on eigenvalue-based techniques which include [10, 33, 35, 36, 43]. In particular, we study computational and sensitivity issues for the LSTRS method [35, 36]. 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, 9, 16, 17, 18, 19, 20, 21, 22, 25, 26, 29, 30, 31, 50]). TRUST$_\mu$, which is based on LSTRS, has also been used in applications and as guideline for developing new methods (cf. [16, 28, 48, 49]). 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 [34]) that there always exists an optimal value of a scalar parameter $\alpha$ such that a solution $x$ to (1) can be computed from a solution $y = (1, x^T)^T$ to

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

where $B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & H \end{pmatrix}$ and $e_1$ is the first canonical vector.

Notice that a solution to (4) is an eigenvector with non-zero first component corresponding to the smallest eigenvalue of $B_\alpha$. Notice also that the eigenvalues of $H$ and $B_\alpha$ are related. The Cauchy Interlace Theorem (cf. [32]) establishes that the eigenvalues of $B_\alpha$ interlace the eigenvalues of $H$. In particular, the smallest eigenvalue of $B_\alpha$ is a lower bound for the smallest eigenvalue of $H$. This implies that solving an eigenvalue problem for the smallest eigenvalue of $B_\alpha$ and a corresponding eigenvector with non-zero first component yields $x$ and $\lambda$ that automatically satisfy the first two optimality conditions in Lemma 1.1 *for any value of* $\alpha$.

These facts suggest designing an iteration for computing $\alpha$ based on the solution of eigenvalue problems for $B_\alpha$. The methods in [10, 33, 35, 36, 43] propose such iterations. The semidefinite programming approach used in [10, 33] works mainly on a primal problem switching to the dual when the hard case is detected. The method in [43] also switches iterations in the presence of the hard case. LSTRS is a unified iteration that incorporates all cases. The algorithm is described in Appendix A. 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 $B_\alpha$. In LSTRS, an additional eigenpair (corresponding to another eigenvalue) is needed. In both families of methods, the eigenpairs are used to update $\alpha$ by means of rational interpolation and similar safeguarding strategies are used to ensure global convergence of the iteration. We refer the reader to [10, 33, 35, 36] for more details about the theory and computational aspects of this kind of TRS methods. We discuss the features and challenges associated with the eigenvalue computation in Section 3.

## 3 Eigenvalue Problems in TRS Methods

We consider the parametric eigenvalue problem:

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

with $H, g$ as above, and $\alpha_k$ 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) [24, 42], the Nonlinear Lanczos (Arnoldi) Method (NLLM) [47], and the Jacobi-Davidson Method [40]. All methods are matrix-free in the sense that they rely on matrix-vector multiplications only. 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 [10, 33]. 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 [36], 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 $\lambda_1$ and as small as possible on an interval containing the remaining eigenvalues of $B_\alpha$. We then compute the eigenvalues of $T(B_\alpha)$ instead of the eigenvalues of $B_\alpha$. 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. [37]). After convergence, the eigenvalues of $B_\alpha$ 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. The eigenvalues of $H = A^T A$ inherit this feature. The interlacing properties discussed in Section 2 imply that if the smallest $q$ eigenvalues of $H$ are small and clustered then, eigenvalues 2 through $q$ of $B_\alpha$ will also be small and clustered.

The situation for $\lambda_1$, the smallest eigenvalue of $B_\alpha$, is as follows. Recall that $\lambda_1$ is a lower bound for $\delta_1$, the smallest eigenvalue of $H$. The distance between $\lambda_1$ and $\delta_1$ depends on the value of $\alpha$, which in turn depends on the value of $\Delta$. For values of $\Delta$ smaller than a certain critical value, the smallest eigenvalue of $B_\alpha$ is well-separated from the rest and Lanczos-type methods can compute it very efficiently. For larger values of $\Delta$, $\lambda_1$ is well-separated from $\delta_1$ only at early (outer) iterations of LSTRS. As the process converges (and $\alpha_k$ approaches the optimal value), $\lambda_1$ gets closer to $\delta_1$ and, in regularization, to the cluster. Figure 1 illustrates this situation for a test problem from [14]. The figure shows the eigenvalues of $H$ and $B_\alpha$ for the optimal value of $\alpha$, for three trust-region subproblems differing only in the value of $\Delta$. We can observe that for $\Delta$ small (top plot), $\lambda_1$ is well separated from $\delta_1$. Increasing $\Delta$ makes the gap between $\lambda_1$ and $\delta_1$ decrease (middle plot). For large $\Delta$, $\lambda_1$ and $\delta_1$ are indistinguishable (bottom plot).

It is often the case in regularization that $\Delta$ 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 the 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 [36], especially for regularization problems. This was to be expected since LSTRS was designed with focus on this kind of problems. Recently [23], significant reductions in the number of MVP have been obtained at a moderate cost in storage by means of the

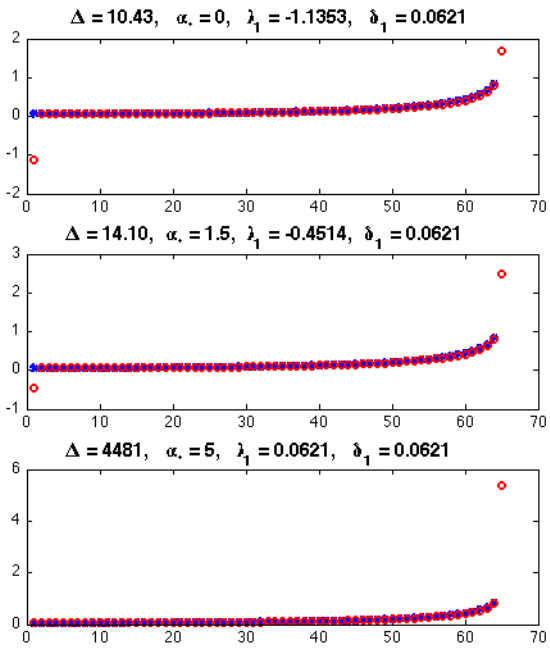



Figure 1: Eigenvalues of $H$ (dot) and $B_\alpha$ (circle) for different values of $\Delta$ (and $\alpha_*$). Problem **heat** from [14].

NLLM. Preliminary results in [23] indicate that the performance of [10, 33] 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: a numerical study

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 Appendix A, Figure 6). 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. [35]). However, the practical convergence speed is more interesting for practitioners. Ideally, we would like to solve the eigenvalue problems to the lowest accuracy (and hence at the lowest cost) required to maintain 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 [11]. Related numerical investigations involving random perturbations can be found in [7, 12, 13, 27, 44]. The remainder of this section is organized as follows. We describe the test problems in Section 4.1 and the experiments in Section 4.2. We present and discuss the results in sections 4.3 and 4.4, respectively.

### 4.1 Test problems

We used two kinds of test problems in our study: general TRS of type (1) and regularization problems of type (2). The two non-regularization problems had an indefinite Hessian. Additionally, the potential hard case was enforced on one of them. For the regularization problems, the matrix $A$ was an $m \times n$ matrix; $b$ was an $m$-dimensional vector such that $b = b_{true} + s$, with $b_{true} = Ax_{true}$; $x_{true}$ was the (true) solution to the inverse problem; and $s$ was a vector of Gaussian noise. The problems were generated with routines from the test set [14], which provided $A$, $b$, and $x_{true}$. The set used consisted of a small model regularization problem, two problems with different degrees of ill posedness and with a *rectangular* matrix $A$, and an image restoration problem of size 65536. For the image restoration problem, $b$ and $x_{true}$ were black and white photographs. Note that regularization problems were the main focus of the study since, as mentioned before, they usually yield very difficult TRS. In all problems, the trust-region radius $\Delta$ was chosen so that the (original, unperturbed) problem had a single boundary solution.

In sections 4.1.1 and 4.1.2, we describe the two non-regularization problems. Sections 4.1.3 through 4.1.5 contain a description of the regularization problems. A summary of the problems and their main features is presented in Section 4.1.6.

#### 4.1.1 Problem laplacian: a shifted Laplacian problem

In this problem, $H = L - 5I$, with $L$ the standard 2-D discrete Laplacian on the unit square based on a 5-point stencil with equally-spaced mesh points. The dimension of the problem was $n = 1024$. The trust-region radius was fixed at $\Delta = 100$. The vector $g$ was randomly generated with entries uniformly distributed on $(0, 1)$. A uniform noise vector of norm $10^{-8}$ was added to $g$.

#### 4.1.2 Problem udut: $UDU^T$ problem

In this problem, the Hessian matrix was defined as $H = UDU^T$ with $D$ a diagonal matrix with elements $d_1, \ldots, d_n$, and $U = I - 2uu^T$ with $u^T u = 1$. The dimension of the problem was $n = 1000$. The elements of $D$ were randomly generated with a uniform distribution on $(-5, 5)$, then sorted in nondecreasing order and $d_1$ was set to $-5$. Both vectors $u$ and $g$ were randomly generated with entries selected from a uniform distribution on $(-0.5, 0.5)$. The vector $u$ was normalized to have unit length.

Note that the eigenvectors of $H$ are of the form $v_i = e_i - 2uu_i$, $i = 1, \ldots, n$, with $e_i$ the $i$th canonical vector and $u_i$ the $i$th component of the vector $u$. The vector $g$ was orthogonalized against $v_1 = e_1 - 2uu_1$, and a noise vector of norm $10^{-2}$ was added to $g$. Finally, $g$ was normalized to have unit norm. The trust-region radius $\Delta$ was chosen as follows. We first computed $x_{min} = -(H - d_1 I)^{\dagger} g$ and $\Delta_{min} = \|x_{min}\|$, and then set $\Delta = 0.1\Delta_{min}$.

#### 4.1.3 Problem shaw: instrument design

This test problem is a small model problem obtained by means of quadrature rules applied to a Fredholm integral equation of the first kind. The problem arises in instrument design and was first introduced in [39]. Problem **shaw** possesses features of both rank-deficient and ill-posed problems with 80% of the singular values of the order of machine precision but with a clear gap between the largest singular values and the rest. The dimension of the matrix $A$ was $100 \times 100$. The TRS was of dimension 100. The noise level in the vector $s$ mentioned above was $10^{-2}$. The trust-region radius was $\Delta = 9.5 < \|x_{true}\|$.

#### 4.1.4 Problems heat, mild and heat, severe: inverse heat equation

This problem is a discretized version of the inverse heat equation. The problem arises, for example, in the inverse heat conduction problem of determining the temperature on the surface of a body from transient

measurements of the temperature at a fixed location in the interior [4]. The equation is a Volterra integral equation

$$\gamma(y) = \int_0^y \mathcal{K}(y,t)\phi(t)dt, \quad 0 \le y \le 1,$$

where $\mathcal{K}(y,t) = k(y-t)$, with $k(t) = \frac{t^{-3/2}}{2\kappa\sqrt{\pi}} \exp\left(-\frac{1}{4\kappa^2 t}\right)$. The parameter $\kappa$ controls the degree of ill posedness.

We performed experiments with **heat, mild**, a mildly ill-posed problem ($\kappa = 5$) and **heat, severe**, a severely ill-posed problem ($\kappa = 1$). To generate the discrete problems, we used a modified version of routine **heat** from [14] that can generate rectangular problems. The dimension of the matrix $A$ was $m \times n$, with $m = 2000$ and $n = 1000$. The corresponding TRS was of dimension $n = 1000$. In order to generate more difficult TRS, no noise was added to the vector $b$ (cf. [34, 37]). For this problem, 20% of the singular values of the matrix $A$, and therefore of the eigenvalues of the Hessian $A^T A$, were zero to working precision. There was no gap in the singular spectrum.

#### 4.1.5 Problem paris: an image restoration problem

In image restoration, we want to recover an image from blurred and noisy data. This is a regularization problem in which the coefficient matrix $A$ represents the blurring operator and the data vector $b$ contains a vector version of a degraded image. In our study, the matrix $A$ was generated with the routine **blur** from [14]. The original image $x_{true}$ was a vector version of a black and white photograph of an art gallery in Paris. The noise level in the vector $s$ was $10^{-2}$ and $\Delta = 0.95\|x_{true}\|$. The dimension of the image was $256 \times 256$, the dimension of $A$ was $65536 \times 65536$, and the dimension of the resulting TRS was $n = 65536$.

#### 4.1.6 Summary of test problems

A summary of the test problems and their features is presented in Table 1.

| Problem | TRS dimension | Regularization | Features |
|---|---|---|---|
| **laplacian** | 1024 | no | indefinite Hessian |
| **udut** | 1000 | no | indefinite Hessian, potential hard case |
| **shaw** | 100 | yes | square matrix, ill-posed, rank-deficient |
| **heat, mild** | 1000 | yes | rectangular matrix, mildly ill-posed |
| **heat, severe** | 1000 | yes | rectangular matrix, severely ill-posed |
| **paris** | 65536 | yes | square matrix, severely ill-posed, large-scale |

Table 1: Test problems.

### 4.2 Experiments

In this section, we describe the experiments in our numerical study. The experiments were carried out in MATLAB R2011b on a MacBookPro with a 2.66 GHz processor and 4 GB of RAM, running Mac OS X version 10.7.4 (Lion). The floating-point arithmetic was IEEE standard double precision with machine precision $2^{-52} \approx 2.2204 \times 10^{-16}$. We performed two kinds of experiments to investigate the relationship between the inner iteration (eigensolver) and the outer iteration (main iteration) of LSTRS. For this purpose, we solved several instances of problems (1) and (2) using settings chosen to favor boundary solutions for the TRS. The specific settings can be found in Appendix B. The following three

eigensolvers, which are available in LSTRS, were tested: `eig` (QR method), `eigs` (MATLAB's interface to ARPACK [24]), and `tcheigs` (`eigs` combined with a Tchebyshev spectral transformation).

Note that not all eigensolvers were used for all problems. For instance, the non-regularization problems were not ill-posed and neither the Hessian nor the bordered matrix had eigenvalues close to zero. For this reason, it was not necessary to use the Tchebyshev transformation and therefore, only the two eigensolvers `eig` and `eigs` were used in this case. All eigensolvers could be used for the regularization problems of medium size. For the large-scale problem **paris**, only `tcheigs` could be used.

The first experiment aimed to study the effect on the accuracy of a TRS solution of random perturbations in either the eigenvalue or the eigenvector. The perturbations were used to simulate the errors that an iterative method would introduce. Note that the goal of the experiment 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. In our study, exact eigenvalues and eigenvectors were computed with MATLAB's routine `eig` (QR method). Note that owing to memory limitations, this experiment could not be performed on the large-scale problem **paris**.

The second experiment sought to simulate the solution of the eigenproblems to different accuracy by means of random perturbations. In this case, the perturbations were introduced in the eigenvalue or eigenvector computed with an inexact eigensolver. In our study, the inexact eigensolvers were `eigs` and `tcheigs`.

In both experiments, at each LSTRS iteration, the eigenvalue problem was solved and then either the eigenvalue or the eigenvector was perturbed. In both cases, 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. [5, 15, 27]). The perturbations were generated in the following way.

*Eigenvalue perturbations.* Given an (unperturbed) eigenvalue $\lambda_u$ and a perturbation level $\varepsilon \in (0, 1)$, a perturbed eigenvalue $\lambda_p$ was constructed as follows:

$$\lambda_p = \lambda_u + \varepsilon\rho, \text{ for } \textit{absolute} \text{ perturbations}$$

$$\lambda_p = \lambda_u(1 + \varepsilon\rho), \text{ for } \textit{relative} \text{ perturbations}$$

where $\rho$ was a random number in $(-1/2, 1/2)$ for uniform distribution and in $(-1, 1)$ for Gaussian.

*Eigenvector perturbations.* Given an (unperturbed) eigenvector $y_u$ and a perturbation level $\varepsilon \in (0, 1)$, a perturbed eigenvector $y_p$ was constructed as follows:

$$y_p = y_u + \varepsilon\rho r, \text{ for } \textit{absolute} \text{ perturbations}$$

$$y_p = y_u \times (e + \varepsilon\rho r), \text{ for } \textit{relative} \text{ perturbations}$$

with $\rho$ 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 $\rho$ was uniformly distributed, or Gaussian distribution if $\rho$ was Gaussian.

For each type of perturbation (eigenvalue and eigenvector, absolute and relative, uniform and Gaussian), we ran 100 examples corresponding to $\rho = \rho_i$, $i = 1, \cdots, 100$ with $\rho_i$'s sorted in increasing order. We used the following values for $\varepsilon$: $10^{-5}$, $10^{-4}$, $10^{-3}$, $10^{-2}$, and $10^{-1}$.

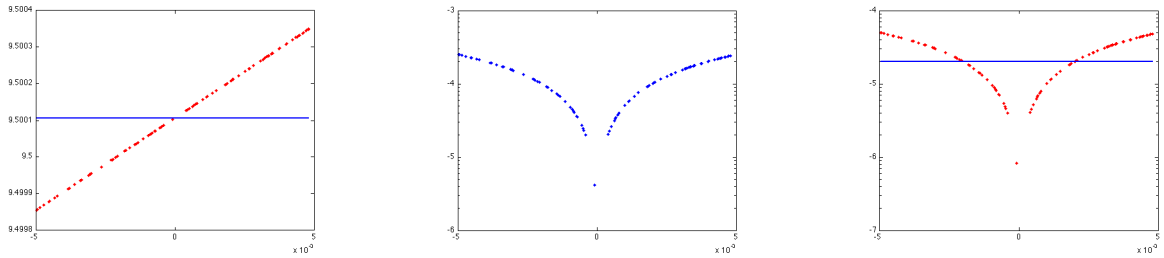
### 4.3 Results

In this section, we present the results from our numerical study. Results for the first experiment correspond to the perturbation level that best simulated the approximation error introduced by an inexact eigensolver. For this experiment, similar results were observed for all kinds of perturbations (uniform

and Gaussian, absolute and relative). Results for the second experiment correspond to the problem and perturbation level and kind (uniform or Gaussian, absolute or relative) for which the worst outcome was observed in terms of accuracy of the solutions. For this experiment, the performance results are averaged over the number of examples (100). Note that for those problems in which noise was present, the results correspond to a fixed noise vector. Experiments with other noise vectors yielded similar results. In the remainder of the paper, $x_p$ and $x_u$ denote the solutions computed by LSTRS using perturbed and unperturbed eigenvalues (or eigenvectors), respectively. $E_p$ denotes the relative error $\frac{\|x_p - x_u\|}{\|x_u\|}$. We present results of the first and second experiments in sections 4.3.1 and 4.3.2, respectively. Results for the more realistic large-scale image restoration problem **paris** are presented separately in Section 4.3.3.

### 4.3.1 First Experiment

*Eigenvalue perturbations.* Figure 2 shows results for problem **shaw**, $\varepsilon = 10^{-2}$, 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 $\varepsilon_A$ (perturbation level for absolute approach) or $\varepsilon_R$ (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 in $x_{inex}$ (a solution computed by LSTRS using `eigs` as eigensolver) with respect to $x_u$. This plot shows that the relative error in $x_p$ is of the same order as the relative error in $x_{inex}$. Thus indicating that, for this perturbation level $\varepsilon$, the random perturbations seem to accurately simulate the approximation error introduced by an inexact eigensolver. Similar results were obtained for all the test problems for which this experiment was performed, although the perturbation level that simulated the relative error in $x_{inex}$ was problem-dependent: $10^{-3}$ for **laplacian**, $10^{-5}$ for **udut**, $10^{-4}$ for **heat, mild**, and $10^{-5}$ for **heat, severe**.



(a) (b) (c)

Figure 2: 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.

*Eigenvector perturbations.* Similar results to the ones obtained for eigenvalue perturbations were obtained for all the test problems when eigenvector perturbations were used. Only the perturbation level that simulated the relative error in $x_{inex}$ was different (and again problem-dependent): $10^{-5}$ for **shaw**, $10^{-4}$ for **laplacian** and **udut**, and $10^{-5}$ for **heat, mild** and **heat, severe**. Note that the same experiment was performed using different normalized random vectors to generate the perturbations and a similar behavior was observed in this case, although a larger number of examples (500 instead of 100) was needed to expose the trends.

### 4.3.2 Second Experiment

We now present an overview of the results of the second experiment. Figure 3 shows the behavior in accuracy (max $E_p$) and performance in terms of matrix-vector products (MVP) for problem **heat, severe**, when eigenvalue or eigenvector perturbations were introduced. The plots were generated from the data in Table 7, Appendix C. Detailed results for all the test problems can be found in Appendix C.

*Eigenvalue perturbations.* For most problems, we observed that the accuracy was overall equal to or higher than the perturbation level. The only exceptions were **heat, severe** (the severely ill-posed problem) and **udut**. For **heat, severe**, lower accuracy was obtained for the three smaller perturbations, while accuracy of the same order was obtained for the two largest perturbations. As it will be discussed later in this section, an improvement in accuracy for the two largest perturbation levels was also observed when eigenvector perturbations were used. The MVP remained the same for the three smallest perturbations but increased to 15% and 20% (with respect to the results for unperturbed eigenvalues), respectively, for the two largest. A similar behavior was observed for problem **udut**.

*Eigenvector perturbations.* For most problems, we observed that the accuracy was overall of the same order as or higher than the perturbation level (although in general lower than for eigenvalue perturbations), and that eigenvector perturbations had a bigger impact on both accuracy and performance than eigenvalue perturbations. The only exceptions were again **heat, severe** and **udut**. For **heat, severe**, we obtained better accuracy than for eigenvalue perturbations for the two smallest perturbations. Performance remained equal or similar to the unperturbed case for the two smallest perturbation levels, improved by 2% and 16% for the third and fourth largest perturbation levels, and increased for the largest level. This behavior was more marked for problem **udut**, where an improvement in performance of approximately 19% was observed for all perturbation levels except the largest for which an improvement of 13% was obtained. Note that in both cases the unperturbed problems were very close to the hard case. In problem **udut**, the potential hard case was artificially enforced and it would seem that our choice of $\Delta = 0.1\Delta_{min}$ was not large enough to avoid the hard-case. Problem **heat, severe** is severely ill-posed and in this case, several of the eigenvectors corresponding to the smallest eigenvalue of the Hessian are orthogonal to $g$. In this situation, the first component of the relevant eigenvector is very small and the adjustment procedure for $\alpha$ (see Appendix A, Figure 7) is triggered. As a consequence, the LSTRS iterations may be expensive. In both cases, the perturbations that we introduced in the eigenvectors seem to have had the effect of moving the LSTRS iterates away from the hard case rendering the problems easier to solve. This is similar to the positive effect that noise in the data has on the properties of the TRS (cf. [37]).

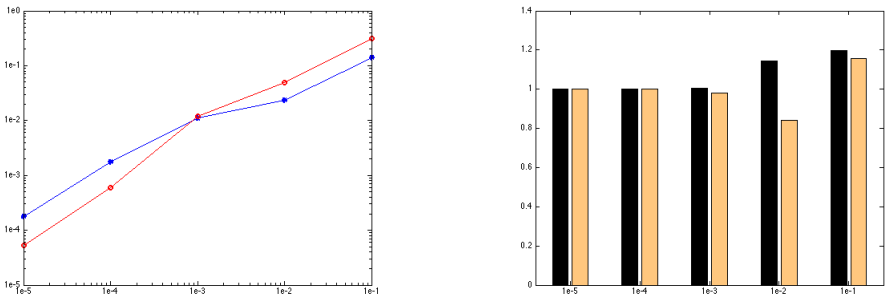



Figure 3: Left, y-axis: max $E_p$ in logarithmic scale for eigenvalue perturbations (dot) and eigenvector perturbations (circle). Right, y-axis: ratio (MVP $x_p$)/(MVP $x_u$) for eigenvalue perturbations (dark) and eigenvector perturbations (clear). On the $x$-axis, perturbation level. Problem **heat, severe**, eigensolver `tcheigs`.

### 4.3.3 An image restoration problem

In this section, we present results of the sensitivity study for problem **paris**, a large-scale image restoration problem. Since the problem was large and since the matrix $A$ was very ill-conditioned with a large cluster of singular values that were numerically zero, only the eigensolver `tcheigs` could be used in this case. Therefore, only the second experiment could be performed on this problem. The results of the experiment are shown in Figure 4. The plots are based on the data in Table 8, Appendix C.

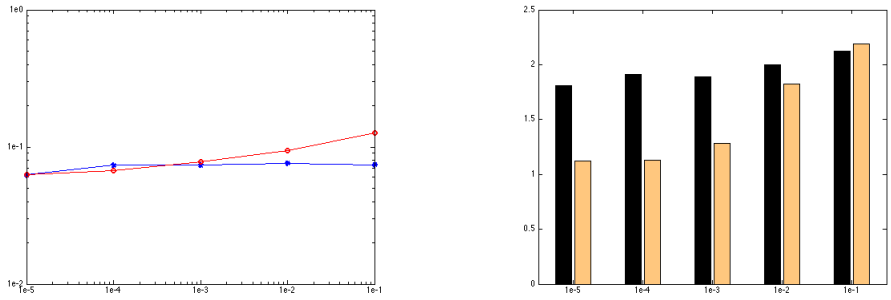



Figure 4: Left, y-axis: max $E_p$ in logarithmic scale for eigenvalue perturbations (dot) and eigenvector perturbations (circle). Right, y-axis: ratio (MVP $x_p$)/(MVP $x_u$) for eigenvalue perturbations (dark) and eigenvector perturbations (clear). On the $x$-axis, perturbation level. Problem **paris**, eigensolver `tcheigs`.

We observe that for both eigenvalue and eigenvector perturbations, the accuracy with respect to the solution of the unperturbed problem is not high. The solution seems to be very sensitive to small perturbations in the eigenpairs. However, the accuracy is of the same order for all perturbations, including large ones, except for the largest eigenvector perturbation. As mentioned above, this problem is very ill-conditioned and therefore, the same comments as for **heat, severe** apply here, in particular, regarding eigenvector perturbations (see Section 4.3.2). For both perturbations, the MVP increased by between 80% and two times with respect to the unperturbed case. However, for eigenvector perturbations, large MVP were only attained for the largest perturbations and only a minor increase was observed for the smallest ones, whereas for eigenvalue perturbations, a large number of MVP was observed for most perturbations.

Figure 5 shows the original and degraded images (top row) as well as three different restorations corresponding to unperturbed eigenpairs and to the two solutions that had maximum $E_p$ for eigenvalue and eigenvector perturbations (bottom row). For eigenvalues, the maximum $E_p$ was reached at perturbation level $10^{-2}$ and for eigenvectors, at $10^{-1}$. Note that the solutions corresponding to perturbed eigenpairs have approximately the same relative error with respect to $x_{true}$, as the solution corresponding to unperturbed ones. The quality of the restored image is noticeably lower for the eigenvector perturbation that yielded the maximum $E_p$. Note that in all cases, the quality of the restoration could be improved by choosing a larger value for $\Delta$. Here, we used $\Delta < \|x_{true}\|$, a value that yields boundary solutions but that is smaller than the optimal one.

## 4.4 Discussion

The results of the first experiment show that the norm of the solution to perturbed problems increases linearly with the perturbation and remains close to the norm of the solution to the unperturbed problem. Moreover, we could always identify a perturbation level that simulated the error introduced by one of our

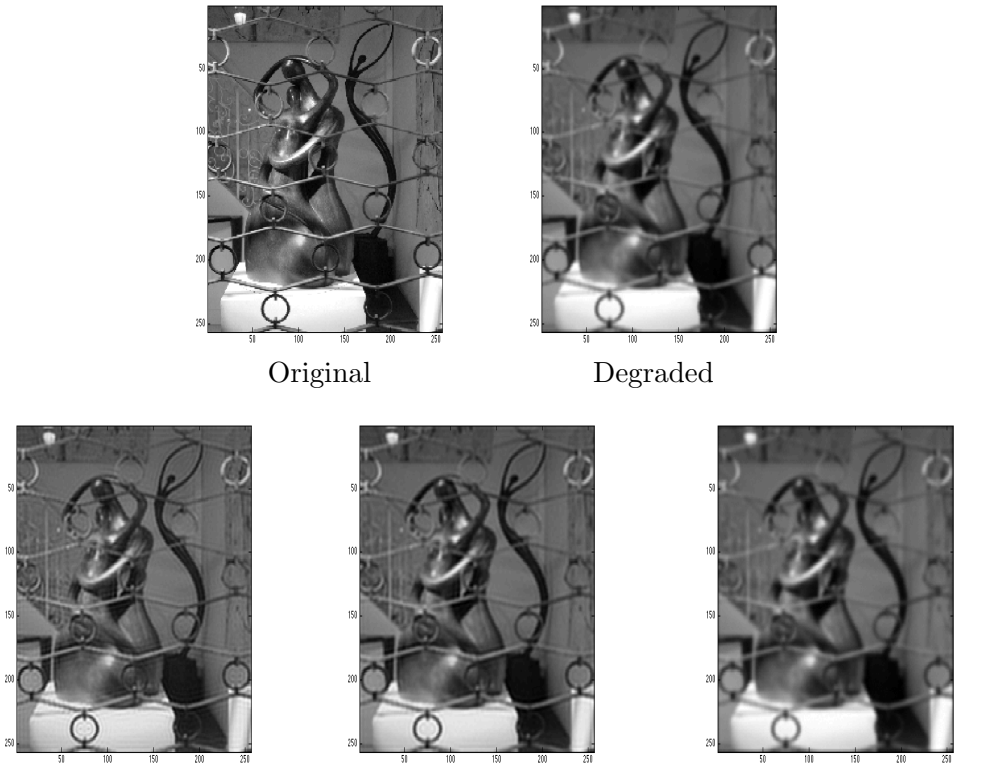



Figure 5: Top row: original and data images. Bottom row: solutions corresponding to unperturbed eigenpairs, to max $E_p$ for eigenvalue perturbations, and to max $E_p$ for eigenvector perturbations. The relative errors with respect to $x_{true}$ are 0.1057, 0.1325, and 0.1694, respectively.

inexact eigensolvers. Therefore, these results seem to indicate that the effect of inexact computations in the inner iteration of LSTRS is limited.

The results of the second experiment show several interesting aspects of the LSTRS method. We observed that for most problems the solution was more sensitive to perturbations in the eigenvectors than to perturbations in the eigenvalues. This was to be expected since the eigenvector influences the TRS solution in a more direct way than the eigenvalue. Namely, the last $n$ (normalized) components of the eigenvector 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 the eigenvector form the solution. However, note that the maximum relative error with respect to the solution of the unperturbed problem was of the same or lower order than the perturbation level for all kinds of perturbations. Only for the very ill-posed problems the behavior was different, with the relative error being one order larger than the perturbation level or, in the case of our large-scale test problem, remaining the same for all perturbation levels.

We also observed (a sometimes dramatic) deterioration in performance when perturbations were introduced, in particular when eigenvector perturbations were used in combination with the iterative eigensolvers on problems that were not very ill-posed or did not have singularities. Note that, since the iterates depend so closely on the eigenvector, by introducing perturbations in that 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 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.

An interesting behavior was observed for problems where the hard case or the potential hard case was present. In those cases, eigenvector perturbations seem to have had the effect of avoiding the singularities and, as a result, performance actually improved. Since the final solutions remained close to the solutions to the unperturbed problems, this would indicate that especially for these very difficult problems it may be advisable to solve the eigenproblems to low accuracy since the perturbation introduced by the inexact solver could be beneficial in those cases.

## 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. This is especially the case for difficult problems such as regularization problems, where the hard case is present. Current and future work in this area include 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 [10, 33] is yet to be done.

**Acknowledgments**. We would like to thank Dan Sorensen for providing an important reference. 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. Finally, we thank the three anonymous referees for their constructive comments.

## 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] A. Carasso. Determining surface temperatures from interior observations. *SIAM J. Appl. Math.*, 42:558–574, 1982.

[5] F. Chaitin-Chatelin and V. Frayssé. *Lectures on Finite Precision Computations.* SIAM, Philadelphia, 1996.

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

[7] 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.

[8] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17:134–145, 1977.

[9] A.P. Eriksson, C. Olsson, and F. Kahl. Image segmentation with context. *Lect. Notes Comp. Sci.*, 4522:283–292, 2007.

[10] C. Fortin and H. Wolkowicz. The trust-region subproblem and semidefinite programming. *Optim. Method. Softw.*, 19(1):41–67, 2004.

[11] 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. Available at `http://hdl.handle.net/1956/3207`.

[12] G.H. Golub and H. Melbø. Stochastic error estimates for iterative linear solvers: part 2. Unpublished report, 2000.

[13] G.H. Golub and H. Melbø. A stochastic approach to error estimates for iterative linear solvers: part 1. *BIT*, 41(5):977–985, 2001.

[14] P.C. Hansen. Regularization Tools version 4.0 for Matlab 7.3. *Numer. Algo.*, 46:189–194, 2007.

[15] N.J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 2nd edition, 2002.

[16] M.E. Hochstenbach, N. McNinch, and L. Reichel. Discrete ill-posed least-squares problems with a solution norm constraint. *Linear Algebra Appl.*, 436:3801–3818, 2012.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] S. Kosinov and T. Pun. Distance-based discriminant analysis method and its applications. *Pattern Anal. Applic.*, 11:227–246, 2008.

[22] 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.

[23] J. Lampe, M. Rojas, D.C. Sorensen, and H. Voss. Accelerating the LSTRS algorithm. *SIAM J. Sci. Comput.*, 33(1):175–194, 2011.

[24] 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.

[25] 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.

[26] J.L. Mead and R.A. Renaut. Least squares problems with inequality constraints as quadratic constraints. *Linear Algebra Appl.*, 432(8):1936–1949, 2008.

[27] J.J. Moré and S.F. Wild. Estimating computational noise. *SIAM J. Sci. Comput.*, 33(3):1292–1314, 2011.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, 1980.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] M. Rojas and T. Steihaug. An interior-point trust-region-based method for large-scale non-negative regularization. *Inverse Probl.*, 18:1291–1307, 2002.

[39] C.B. Shaw, Jr. Improvements of the resolution of an instrument by numerical solution of an integral equation. *J. Math. Anal. Appl.*, 37:83–112, 1972.

[40] 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.

[41] D.C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2):409–426, 1982.

[42] D.C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. A.*, 13(1):357–385, 1992.

[43] D.C. Sorensen. Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM J. Optimiz.*, 7(1):141–161, 1997.

[44] G.W. Stewart. Stochastic perturbation theory. *SIAM Rev.*, 32(4):579–610, 1990.

[45] A.N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math.*, 4:1624–1627, 1963.

[46] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems.* Halsted Press, New York, 1977.

[47] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT*, 44:387–401, 2004.

[48] Y. Wang and S. Ma. A fast subspace method for image deblurring. *Appl. Math. Comput.*, 215(6):2359–2377, 2009.

[49] R. Zdunek and A. Cichocki. Nonnegative matrix factorization with quadratic programming. *Neurocomputing*, 71(10–12):2309–2320, 2008.

[50] 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.

# A The LSTRS Method

In this section, we present the main components of the LSTRS algorithm from [35]: the main iteration (Figure 6), the procedure for adjusting the parameter $\alpha$ (Figure 7), and the procedure for safeguarding $\alpha$ (Figure 8). The 1- and 2-point rational (1D) interpolation schemes are described in detail in [35]. Note that they do not contribute significantly to the iteration cost. The solution of the parametric eigenvalue problem (Figure 6: lines 3 and 16, and (conditionally) line 6) is the most expensive computation at each iteration. In all algorithms: $H$, $g$, and $\Delta$ are as in (1), $B_\alpha$ is as in (4), $\delta_1$ is the smallest eigenvalue of $H$,

$\delta_U$ is an upper bound for $\delta_1$, and $\lambda_1(\alpha)$ and $\lambda_i(\alpha)$ denote the smallest eigenvalue and the $i$th eigenvalue of $B_\alpha$ for *any* $i > 1$, respectively; $\alpha_L$ and $\alpha_U$ are lower and upper bounds for $\alpha_k$, respectively. Input tolerances are described in Table 2. The stopping criterion consists of conditions for a boundary, quasi-optimal, and interior solution, as well as for the length of the safeguarding interval for $\alpha$ and a maximum number of iterations. For more details about the stopping criterion, see [35, 36] and also Table 2.

| | |
|---|---|
| $\varepsilon_\Delta$ | The desired relative accuracy in the norm of the trust-region solution. A boundary solution $x$ satisfies $\frac{\lvert\lVert x\rVert-\Delta\rvert}{\Delta} \leq \varepsilon_\Delta$ . |
| $\varepsilon_{HC}$ | The desired accuracy of a quasi-optimal solution. If $x_*$ is the true solution and $\widetilde{x}$ is the quasi-optimal solution, then $\psi(x_*) \leq \psi(\widetilde{x}) \leq (1-\varepsilon_{HC})\psi(x_*)$, where $\psi(x) = \frac{1}{2}x^T Hx + g^T x$. |
| $\varepsilon_{Int}$ | Used to declare the algebraically smallest eigenvalue of $B_\alpha$ positive in the test for an interior solution: $\lambda_1(\alpha)$ is considered positive if $\lambda_1(\alpha) > -\varepsilon_{Int}$ . |
| $\varepsilon_\alpha$ | The minimum relative length of the safeguarding interval for $\alpha$. The interval is too small when $\lvert\alpha_U - \alpha_L\rvert \leq \varepsilon_\alpha * \max\{\lvert\alpha_L\rvert, \lvert\alpha_U\rvert\}$ . |
| $\varepsilon_\nu$ | The minimum relative size of an eigenvector component. The component $\nu$ is small when $\lvert\nu\rvert \leq \varepsilon_\nu \lVert u\rVert / \lVert g\rVert$ . |

Table 2: Tolerances for LSTRS.

# B Settings for LSTRS

The settings used in the sensitivity study were the same as in [23, 36]. We include them here for completeness. In all cases, the initial vector for the first call to `eigs` was $v_0 = (1, \ldots, 1)^T/\sqrt{n+1}$. Default values were used for the parameters that are not mentioned.

## B.1 Problem laplacian

The number of vectors was 10 and 8 shifts were applied in each implicit restart. The remaining parameters were as follows: `epsilon.Delta = 1e-5`, `epsilon.HC = 1e-11`, `lopts.deltaU = 'mindiag'`, and `lopts.alpha0 = lopts.deltaU`.

## B.2 Problem udut

The number of vectors was 10 and 8 shifts were applied in each implicit restart. Other parameters were as follows: `epsilon.Delta = 1e-4`, `epsilon.HC = 1e-10`, `lopts.deltaU = 'mindiag'`, `lopts.alpha0 = lopts.deltaU`, `lopts.maxeigentol = 0.2`,

## B.3 Problem shaw

We set `epsilon.HC = 1e-16` and `epsilon.Int = 0` to favor boundary solutions; and `epsilon.Delta = 1e-2`; `lopts.max_eigentol = 0.4`.

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

Initialization
  Compute $\delta_U \geq \delta_1$, initialize $\alpha_U$ and $\alpha_0$, set $k = 0$
  Compute **eigenpairs** $\{\lambda_1(\alpha_0), (\nu_1, u_1^T)^T\}$, and $\{\lambda_i(\alpha_0), (\nu_i, u_i^T)^T\}$ of $B_{\alpha_0}$
  Initialize $\alpha_L$
**repeat**
  Adjust $\alpha_k$ using algorithm in Figure 7 (might need to compute **eigenpairs**)
  Update $\delta_U = \min\left\{\delta_U, \frac{u_1^T H u_1}{u_1^T u_1}\right\}$
  **if** $\|g\||\nu_1| > \varepsilon_\nu\sqrt{1-\nu_1^2}$ **then**
    Set $\lambda_k = \lambda_1(\alpha_k)$ and $x_k = \frac{u_1}{\nu_1}$
    **if** $\|x_k\| < \Delta$ **then** $\alpha_L = \alpha_k$ **end if**; **if** $\|x_k\| > \Delta$ **then** $\alpha_U = \alpha_k$ **end if**
  **else**
    Set $\lambda_k = \lambda_i(\alpha_k)$, $x_k = \frac{u_i}{\nu_i}$, and $\alpha_U = \alpha_k$
  **end if**
  Compute $\alpha_{k+1}$ by 1-point ($k = 0$) or 2-point interpolation scheme (see [35])
  Safeguard $\alpha_{k+1}$ using algorithm in Figure 8 and set $k = k + 1$
  Compute **eigenpairs** $\{\lambda_1(\alpha_k), (\nu_1, u_1^T)^T\}$, and $\{\lambda_i(\alpha_k), (\nu_i, u_i^T)^T\}$ of $B_{\alpha_k}$
**until** convergence

Figure 6: The LSTRS Method for solving (1).

**Input:** $\varepsilon_\nu, \varepsilon_\alpha \in (0, 1)$, $\alpha_L$, $\alpha_U$, $\alpha$ with $\alpha \in [\alpha_L, \alpha_U]$,
  eigenpairs $\{\lambda_1(\alpha), (\nu_1, u_1^T)^T\}$ and $\{\lambda_i(\alpha), (\nu_i, u_i^T)^T\}$ of $B_\alpha$
**Output:** $\alpha, \{\lambda_1(\alpha), (\nu_1, u_1^T)^T\}$ and $\{\lambda_i(\alpha), (\nu_i, u_i^T)^T\}$.
**while** $\|g\||\nu_1| \leq \varepsilon_\nu\sqrt{1-\nu_1^2}$ **and** $\|g\||\nu_i| \leq \varepsilon_\nu\sqrt{1-\nu_i^2}$
  **and** $|\alpha_U - \alpha_L| > \varepsilon_\alpha \max\{|\alpha_L|, |\alpha_U|\}$ **do**
  $\alpha_U = \alpha$; $\alpha = (\alpha_L + \alpha_U)/2$
  Compute eigenpairs $\{\lambda_1(\alpha), (\nu_1, u_1^T)^T\}$ and $\{\lambda_i(\alpha), (\nu_i, u_i^T)^T\}$ of $B_\alpha$
**end while**

Figure 7: Adjustment of $\alpha$.

**Input:** $\alpha$, $\delta_U \geq \delta_1$, $\alpha_L$, $\alpha_U$,
$x_j$, $\lambda_j$, $\phi_j = -g^T x_j$ and $\phi'_j = \|x_j\|^2$, for $j = k-1, k$.
**Output:** $\alpha \in [\alpha_L, \alpha_U]$.

1: **if** $\alpha \notin [\alpha_L, \alpha_U]$
2: **if** $k = 0$ **then** $\alpha = \delta_U + \phi_k + \phi'_k(\delta_U - \lambda_k)$
3: **else if** $\|x_k\| < \|x_{k-1}\|$ **then** $\alpha = \delta_U + \phi_k + \phi'_k(\delta_U - \lambda_k)$
4: **else** $\alpha = \delta_U + \phi_{k-1} + \phi'_{k-1}(\delta_U - \lambda_{k-1})$ **end if**
5: **if** $\alpha \notin [\alpha_L, \alpha_U]$ **then** set $\alpha = (\alpha_L + \alpha_U)/2$ **end if**
6: **end if**

Figure 8: Safeguarding of $\alpha$.

### B.4 Problems heat, mild and heat, severe

We set `lopts.heuristics = 1`; `epsilon.Delta = 1e-3` and `lopts.maxeigentol = 0.7` for **heat, mild** and `epsilon.Delta = 1e-2` and `lopts.maxeigentol = 0.4` for **heat, severe**.

### B.5 Problem paris

The settings were `epsilon_Delta = 1e-2` and `epsilon_HC = 1e-4`.

## C Numerical Results

This section presents the numerical results of our sensitivity study for each problem. As mentioned in Section 4.3, the results correspond to the kind of perturbation (uniform or Gaussian, relative or absolute) that yielded the worst results in terms of max $E_p$. This data was used to generate the plots in Section 4.3. In all tables, we report: the perturbation level ($\varepsilon$); 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.

### C.1 Problem laplacian

Table 3 shows results for eigenvalue and eigenvector perturbations. For eigenvalues, the results correspond to Gaussian distribution and relative perturbations. For eigenvectors, the results correspond to Gaussian distribution and absolute perturbations. For eigenvalue perturbations, we can observe that the solutions are close to the solution corresponding to unperturbed eigenvalues and that high accuracy was also obtained for the largest perturbation, $10^{-1}$. Performance deteriorated at perturbation level $10^{-2}$ for both `eig` and `eigs`. For `eigs`, the MVP increased by a factor of approximately two and three, respectively, for the two largest perturbations. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-3}$. For eigenvector perturbations, the error was of the same order as the perturbation. Performance increased for the three largest perturbations and decreased for the two smallest. For `eigs`, the MVP decreased by a factor of two and then increased to approximately the same level of the unperturbed problem. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-4}$.

Eigenvalue Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 5.0 | 5.0 | 0 |
| $10^{-5}$ | 5.0 | 5.0 | 1.09e-07 |
| $10^{-4}$ | 5.0 | 5.0 | 1.09e-06 |
| $10^{-3}$ | 5.1 | 5.1 | 8.75e-06 |
| $10^{-2}$ | 8.7 | 8.7 | 1.02e-05 |
| $10^{-1}$ | 15.5 | 15.5 | 1.03e-05 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 6.0 | 6.0 | 104.0 | 0 |
| $10^{-5}$ | 6.0 | 6.0 | 104.0 | 1.11e-07 |
| $10^{-4}$ | 6.0 | 6.0 | 104.0 | 1.10e-06 |
| $10^{-3}$ | 6.0 | 6.0 | 107.7 | 9.93e-06 |
| $10^{-2}$ | 10.4 | 10.4 | 210.8 | 9.92e-06 |
| $10^{-1}$ | 15.3 | 15.3 | 302.1 | 1.01e-05 |

`eigs`

Eigenvector Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 5.0 | 5.0 | 0 |
| $10^{-5}$ | 4.0 | 4.0 | 5.10e-05 |
| $10^{-4}$ | 4.0 | 4.0 | 5.38e-04 |
| $10^{-3}$ | 4.0 | 4.0 | 5.42e-03 |
| $10^{-2}$ | 5.3 | 5.3 | 2.72e-02 |
| $10^{-1}$ | 6.5 | 6.5 | 1.53e-01 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 6.0 | 6.0 | 104.0 | 0 |
| $10^{-5}$ | 4.0 | 4.0 | 55.0 | 5.36e-05 |
| $10^{-4}$ | 4.0 | 4.0 | 55.0 | 5.39e-04 |
| $10^{-3}$ | 4.0 | 4.0 | 55.1 | 5.45e-03 |
| $10^{-2}$ | 5.9 | 5.9 | 113.8 | 2.99e-02 |
| $10^{-1}$ | 5.9 | 5.9 | 111.5 | 1.02e-01 |

`eigs`

Table 3: Problem **laplacian**. Performance results and maximum relative error in $x_p$ with respect to $x_u$.

### C.1.1 Problem udut

Table 4 shows results for eigenvalue and eigenvector perturbations. For eigenvalues, the results correspond to Gaussian distribution and relative perturbations. For eigenvectors, the results correspond to Gaussian distribution and absolute perturbations. We observe that for eigenvalue perturbations, the solutions are close to the solution corresponding to unperturbed eigenvalues, although not as close as for problem **laplacian**. Observed also that relatively high accuracy was attained for most of the perturbations. Performance deteriorated at $10^{-4}$ for both `eig` and `eigs`. For `eigs`, the MVP increased by a factor of between two and three. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-5}$. For eigenvector perturbations, the error was of the same order as the perturbation. In this case, the number of MVP decreased for most perturbations, and was of the same order as for the unperturbed problem for the largest perturbation. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-4}$.

### C.1.2 Problem shaw

Table 5 shows results for eigenvalue and eigenvector perturbations. As before, we observe 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 the maximum relative error is of lower order than the perturbation level. Regarding performance, measured in MVP, we observe that this was essentially not affected by the perturbations, except for the largest perturbation level, $10^{-1}$. 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. For eigenvector perturbations, we observe a similar behavior. Regarding performance, measured in MVP, we observe the following. For the eigensolvers `eig` and `tcheigs`, performance was essentially not affected by perturbations, except for the largest perturbation. 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

Eigenvalue Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 5.0 | 5.0 | 0 |
| $10^{-5}$ | 5.0 | 5.0 | 9.79e-05 |
| $10^{-4}$ | 8.4 | 8.4 | 1.04e-04 |
| $10^{-3}$ | 11.3 | 11.3 | 1.13e-04 |
| $10^{-2}$ | 13.2 | 13.2 | 1.11e-04 |
| $10^{-1}$ | 19.6 | 19.6 | 1.11e-04 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 5.0 | 5.0 | 64.0 | 0 |
| $10^{-5}$ | 5.0 | 5.0 | 63.8 | 8.10e-05 |
| $10^{-4}$ | 10.5 | 10.5 | 130.1 | 1.17e-03 |
| $10^{-3}$ | 13.8 | 13.8 | 169.4 | 1.90e-04 |
| $10^{-2}$ | 14.7 | 14.7 | 179.8 | 1.17e-03 |
| $10^{-1}$ | 15.6 | 15.6 | 190.3 | 1.27e-04 |

`eigs`

Eigenvector Perturbation

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 5.0 | 5.0 | 0 |
| $10^{-5}$ | 4.0 | 4.0 | 9.62e-05 |
| $10^{-4}$ | 4.0 | 4.0 | 9.62e-04 |
| $10^{-3}$ | 4.0 | 4.0 | 9.62e-03 |
| $10^{-2}$ | 4.0 | 4.0 | 9.62e-02 |
| $10^{-1}$ | 4.7 | 4.7 | 7.69e-01 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 5.0 | 5.0 | 64.0 | 0 |
| $10^{-5}$ | 4.0 | 4.0 | 52.0 | 9.79e-05 |
| $10^{-4}$ | 4.0 | 4.0 | 52.0 | 9.80e-04 |
| $10^{-3}$ | 4.0 | 4.0 | 52.0 | 9.81e-03 |
| $10^{-2}$ | 4.3 | 4.3 | 56.1 | 9.59e-02 |
| $10^{-1}$ | 5.1 | 5.1 | 64.6 | 7.07e-01 |

`eigs`

Table 4: Problem **udut**. Performance results and maximum relative error in $x_p$ with respect to $x_u$.

respect to the unperturbed case. For `eigs`, we observe MVP of between two and five times those of the unperturbed case. Note also that the deterioration in performance begins at a smaller perturbation level ($10^{-4}$) than before. 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*.

Eigenvalue Perturbations

| $\varepsilon$ | 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`

| $\varepsilon$ | 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`

| $\varepsilon$ | 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`

Eigenvector Perturbations

| $\varepsilon$ | 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`

| $\varepsilon$ | 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`

| $\varepsilon$ | 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`

Table 5: Problem **shaw**. Performance results and maximum relative error in $x_p$ with respect to $x_u$.

### C.1.3 Problems heat, mild and heat, severe

Tables 6 shows results for eigenvalue and eigenvector perturbations on problem **heat, mild**. For eigenvalues, the results correspond to Gaussian distribution and relative perturbations. For eigenvectors, the results correspond to Gaussian distribution and absolute perturbations. For eigenvalue perturbations, the solutions are close to the solution corresponding to unperturbed eigenvalues, except for the two largest perturbations. Performance remained essentially the same for all but the largest perturbation. In the latter case, the MVP increased by approximately 22% for both `eigs` and `tcheigs`. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-4}$. For eigenvector perturbations, the error was of the same order as the perturbation for all eigensolvers. In this case, the increase in MVP for `eigs` started at perturbation level $10^{-4}$ and was between 7% and three times larger than for unperturbed problems. For `eig` and `tcheigs`, performance changed slightly for the largest perturbation only. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-5}$.

Eigenvalue Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 3.0 | 3.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 3.70e-06 |
| $10^{-4}$ | 3.0 | 3.0 | 3.71e-05 |
| $10^{-3}$ | 3.0 | 3.0 | 3.80e-04 |
| $10^{-2}$ | 2.9 | 2.9 | 1.21e-02 |
| $10^{-1}$ | 3.6 | 3.6 | 1.21e-02 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 3.0 | 3.0 | 94.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 94.0 | 3.70e-06 |
| $10^{-4}$ | 3.0 | 3.0 | 94.0 | 3.69e-05 |
| $10^{-3}$ | 3.0 | 3.0 | 94.0 | 3.60e-04 |
| $10^{-2}$ | 3.0 | 3.0 | 93.5 | 1.21e-02 |
| $10^{-1}$ | 3.8 | 3.8 | 115.1 | 1.19e-02 |

`eigs`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 3.0 | 3.0 | 419.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 419.0 | 3.70e-06 |
| $10^{-4}$ | 3.0 | 3.0 | 419.0 | 3.69e-05 |
| $10^{-3}$ | 3.0 | 3.0 | 419.0 | 3.60e-04 |
| $10^{-2}$ | 3.0 | 3.0 | 415.0 | 1.21e-02 |
| $10^{-1}$ | 3.8 | 3.8 | 511.9 | 1.19e-02 |

`tcheigs`

Eigenvector Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 3.0 | 3.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 2.67e-05 |
| $10^{-4}$ | 3.0 | 3.0 | 2.67e-04 |
| $10^{-3}$ | 3.0 | 3.0 | 2.67e-03 |
| $10^{-2}$ | 3.0 | 3.0 | 2.67e-02 |
| $10^{-1}$ | 3.5 | 3.5 | 2.02e-01 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 3.0 | 3.0 | 94.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 94.0 | 2.61e-05 |
| $10^{-4}$ | 3.0 | 3.0 | 100.5 | 2.67e-04 |
| $10^{-3}$ | 3.0 | 3.0 | 151.9 | 2.67e-03 |
| $10^{-2}$ | 3.0 | 3.0 | 211.6 | 2.67e-02 |
| $10^{-1}$ | 3.4 | 3.4 | 251.2 | 2.66e-01 |

`eigs`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 3.0 | 3.0 | 419.0 | 0 |
| $10^{-5}$ | 3.0 | 3.0 | 419.0 | 2.67e-05 |
| $10^{-4}$ | 3.0 | 3.0 | 419.0 | 2.67e-04 |
| $10^{-3}$ | 3.0 | 3.0 | 419.0 | 2.67e-03 |
| $10^{-2}$ | 3.0 | 3.0 | 419.0 | 2.67e-02 |
| $10^{-1}$ | 3.3 | 3.3 | 438.1 | 2.66e-01 |

`tcheigs`

Table 6: Problem **heat, mild**. Performance results and maximum relative error in $x_p$ with respect to $x_u$.

Table 7 show results for eigenvalue and eigenvector perturbations on problem **heat, severe**. The results correspond to Gaussian distribution and absolute perturbations. For eigenvalue perturbations, the solutions have lower accuracy than the perturbation level. For `eigs`, the number of MVP increased by between 30% and three times the cost for solving the unperturbed problem for the two largest perturbations, while for `tcheigs`, the increase was of between 14% and 20%. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-5}$. For eigenvector perturbations, we obtained higher accuracy than for eigenvalue perturbations for the two smallest perturbations. The MVP actually decreased for some of the perturbations by as much as 8% for `eigs` and 19% for `tcheigs`, while it increased by about two times for the largest perturbation. The perturbation that simulated the approximation error of the inexact solver `eigs` was $10^{-5}$.

Eigenvalue Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 6.0 | 7.0 | 0 |
| $10^{-5}$ | 6.0 | 7.0 | 1.90e-04 |
| $10^{-4}$ | 6.0 | 7.0 | 1.90e-03 |
| $10^{-3}$ | 6.0 | 7.0 | 1.64e-02 |
| $10^{-2}$ | 7.4 | 8.8 | 2.03e-02 |
| $10^{-1}$ | 15.1 | 19.2 | 1.88e-02 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 6.0 | 7.0 | 533.0 | 0 |
| $10^{-5}$ | 6.0 | 7.0 | 533.0 | 1.90e-04 |
| $10^{-4}$ | 6.0 | 7.0 | 533.0 | 1.90e-03 |
| $10^{-3}$ | 6.0 | 7.0 | 534.3 | 1.64e-02 |
| $10^{-2}$ | 7.7 | 9.1 | 693.0 | 2.03e-02 |
| $10^{-1}$ | 15.8 | 20.0 | 1546.9 | 1.90e-02 |

`eigs`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 9.0 | 10.0 | 797.0 | 0 |
| $10^{-5}$ | 9.0 | 10.0 | 797.0 | 1.77e-04 |
| $10^{-4}$ | 9.0 | 10.0 | 797.0 | 1.76e-03 |
| $10^{-3}$ | 9.0 | 10.0 | 799.7 | 1.10e-02 |
| $10^{-2}$ | 10.4 | 11.4 | 911.6 | 2.37e-02 |
| $10^{-1}$ | 10.8 | 12.0 | 953.9 | 1.41e-01 |

`tcheigs`

Eigenvector Perturbations

| $\varepsilon$ | IT | EV | max $E_p$ |
|---|---|---|---|
| 0 | 6.0 | 7.0 | 0 |
| $10^{-5}$ | 6.0 | 7.0 | 1.26e-04 |
| $10^{-4}$ | 6.0 | 7.0 | 9.19e-04 |
| $10^{-3}$ | 5.9 | 6.9 | 1.65e-02 |
| $10^{-2}$ | 5.9 | 6.9 | 2.87e-02 |
| $10^{-1}$ | 7.5 | 10.1 | 2.64e-01 |

`eig`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 6.0 | 7.0 | 533.0 | 0 |
| $10^{-5}$ | 6.0 | 7.0 | 533.0 | 4.67e-05 |
| $10^{-4}$ | 6.0 | 7.0 | 533.0 | 3.76e-04 |
| $10^{-3}$ | 5.9 | 6.9 | 527.2 | 1.65e-02 |
| $10^{-2}$ | 5.5 | 6.5 | 492.8 | 2.76e-02 |
| $10^{-1}$ | 9.1 | 11.8 | 923.4 | 2.67e-01 |

`eigs`

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 9.0 | 10.0 | 797.0 | 0 |
| $10^{-5}$ | 9.0 | 10.0 | 797.0 | 5.24e-05 |
| $10^{-4}$ | 9.0 | 10.0 | 797.0 | 6.04e-04 |
| $10^{-3}$ | 8.8 | 9.8 | 781.0 | 1.20e-02 |
| $10^{-2}$ | 7.4 | 8.4 | 671.5 | 4.88e-02 |
| $10^{-1}$ | 10.6 | 11.6 | 921.5 | 3.13e-01 |

`tcheigs`

Table 7: Problem **heat, severe**. Performance results and maximum relative error in $x_p$ with respect to $x_u$.

## C.2 Problem paris

Table 8 shows results for eigenvalue and eigenvector perturbations. For eigenvalues, the results correspond to normal distribution and relative perturbations. For eigenvectors, the results correspond to uniform distribution and absolute perturbations. The results are discussed in Section 4.3.3.

Eigenvalue Perturbations

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 2.0 | 2.0 | 318.0 | 0 |
| $10^{-5}$ | 5.6 | 5.6 | 576.4 | 6.24e-02 |
| $10^{-4}$ | 6.1 | 6.1 | 608.5 | 7.35e-02 |
| $10^{-3}$ | 6.0 | 6.0 | 600.3 | 7.35e-02 |
| $10^{-2}$ | 6.5 | 6.5 | 636.4 | 7.62e-02 |
| $10^{-1}$ | 7.0 | 7.0 | 676.3 | 7.40e-02 |

Eigenvector Perturbations

| $\varepsilon$ | IT | EV | MVP | max $E_p$ |
|---|---|---|---|---|
| 0 | 2.0 | 2.0 | 318.0 | 0 |
| $10^{-5}$ | 2.6 | 2.6 | 357.8 | 6.29e-02 |
| $10^{-4}$ | 2.6 | 2.6 | 358.4 | 6.76e-02 |
| $10^{-3}$ | 3.2 | 3.2 | 407.3 | 7.80e-02 |
| $10^{-2}$ | 5.7 | 5.7 | 579.0 | 9.36e-02 |
| $10^{-1}$ | 7.3 | 7.3 | 695.3 | 1.27e-01 |

Table 8: Problem **paris**. Eigenvalue and eigenvector perturbations, eigensolver `tcheigs`. Performance results and maximum relative error in $x_p$ with respect to $x_u$.