Math. Program., Ser. B (2009) 116:193–220
DOI 10.1007/s10107-007-0126-4

FULL LENGTH PAPER

Regularization using a parameterized trust region subproblem

Oleg Grodzevich · Henry Wolkowicz

Received: 9 May 2005 / Accepted: 30 March 2006 / Published online: 28 April 2007

Abstract We present a new method for regularization of ill-conditioned problems, such as those that arise in image restoration or mathematical processing of medical data. The method extends the traditional *trust-region subproblem*, TRS, approach that makes use of the *L-curve maximum curvature criterion*, a strategy recently proposed to find a good regularization parameter. We apply a parameterized trust region approach to estimate the region of maximum curvature of the L-curve and find the regularized solution. This exploits the close connections between various parameters used to solve TRS. A MATLAB code for the algorithm is tested and a comparison to the conjugate gradient least squares, CGLS, approach is given and analysed.

Keywords Regularization · Trust region subproblem · Ill-conditioned problems · L-curve · Image restoration

1 Introduction

Regularization centers on finding approximate solutions for least-squares problems such as

Research supported by The Natural Sciences and Engineering Research Council of Canada.

O. Grodzevich
Department of Management Sciences, University of Waterloo,
Waterloo, ON N2L 3G1, Canada
e-mail: ogrodzev@uwaterloo.ca

H. Wolkowicz (✉)
Department of Combinatorics and Optimization, University of Waterloo,
Waterloo, ON N2L 3G1, Canada
e-mail: hwolkowicz@uwaterloo.ca

$$\min_x \|Gx - d\|_2, \tag{1}$$

where G is an ill-conditioned $m \times n$ matrix and d is a vector of *observed data*. This problem arises from mathematical models $Gx = d$, where the data contains noise η,

$$Gx = Gx_{\text{true}} + \eta = d = d_{\text{true}} + \eta.$$

It is remarkable that, for many applications, a small amount of noise η can result in a solution x that has no relation to x_{true}, i.e. we can make the size of the error $\|\eta\|_2$ arbitrarily small, while the size of the error in the solution $\|x - x_{\text{true}}\|_2$ is arbitrarily large. (See e.g. the survey article [30] or the book [1].) The least-squares problem (1) typically arises from discretizations of linear equations in in nite dimensional spaces, e.g. $Tx = d$, where T is a compact operator and so has an unbounded inverse. This means that x is not a continuous function of the data d. Such problems are called *ill-posed* [15,16].

To obtain meaningful solutions to the mathematical model one often uses various methods of *regularization*; the classical reference is [40]. The aim is to find algorithms for constructing *generalized solutions* that are stable under small changes in the data d. One method uses the solution, x_ε, of the constrained least-squares problem:

$$\begin{aligned} r_\varepsilon := \quad & \min \quad \|Gx - d\|_2 \\ & \text{subject to } \|x\|_2 \le \varepsilon. \end{aligned} \tag{2}$$

The restriction on $\|x\|_2$ results in a larger residual error $\|Gx - d\|_2$ but reduces the propagated data error in $\|x\|_2$. As ε increases we reduce $\|Gx_\varepsilon - d\|_2$ and expect x_ε to approximate the best least-squares solution $x_{\text{true}} = G^\dagger d_{\text{true}}$, where $G^\dagger$ denotes the Moore-Penrose generalized inverse of G. However, in practice, the error propagation in x_ε stays small for small ε, but then diverges (from x_{true}) as $\varepsilon \downarrow 0$. (See *semiconvergence* in [28].) Regularization depends on choosing the *good (correct)* parameter ε. A Lagrange multiplier argument shows the equivalence between choosing the value for ε in (2) and choosing the corresponding correct value for the *Tikhonov regularization parameter* α in

$$(G^T G + \alpha^2 I)x_\alpha = G^T d \tag{3}$$

to obtain $x_\varepsilon = x_\alpha$.

By squaring the objective and the constraint, (2) can be reformulated as the so-called *trust region subproblem*, TRS , e.g. [8]:

$$(\text{TRS}) \qquad \begin{aligned} \mu_\varepsilon := \quad & \min \quad q(x) := x^T Ax - 2a^T x \\ & \text{subject to} \quad \|x\|_2^2 \le \varepsilon^2, \end{aligned} \tag{4}$$

where $A := G^T G$ is $n \times n$ symmetric (we assume $n \ge 2$ and G is nonsingular, though generally ill-conditioned), $a := G^T d$ is an n-vector, ε is a *positive* scalar, and x is the n-vector of unknowns. All matrix and vector entries are real. For TRS , we let x^* or x_ε

denote the optimal solution and $\lambda^* = \lambda_\varepsilon$ denote the corresponding optimal Lagrange multiplier. The relation between optimal values is $r_\varepsilon = \mu_\varepsilon + d^T d$. The TRS can be used to form the so-called *L-curve*,

$$\mathcal{L}(G, d) := \{(\log(\varepsilon), \log \|Gx_\varepsilon - d\|_2) : \varepsilon > 0,\ x_\varepsilon \text{ is optimal for TRS}\}. \quad (5)$$

A strategy introduced recently to find a good (correct) regularization parameter uses the point of maximum curvature, the *elbow*, on the *L-curve*, e.g. [19]. (See e.g. Fig. 1 for the L-curve of the deblurring problem which is described in Sect. 5.2.)

The classical algorithm for solving TRS, with a given trust region radius ε, is based on solving (3) for various choices of α, and using a Cholesky factorization of $G^T G + \alpha^2 I$, [27]. Each choice of α yields x_α that is optimal for TRS with trust region radius $\varepsilon = \|x_\alpha\|_2$. Extensions of this approach to the large sparse case that avoid the Cholesky factorization are given in [32,37]. These TRS algorithms are parametric methods that exploit the connection between TRS and α, x_α in (3). Therefore, they find points on the L-curve at each iteration.

Regularization that uses the equivalent problem to TRS

$$\min \|x\|_2 \text{ s.t. } \|Gx - d\|_2 \le \delta,$$

for appropriate δ, is introduced in the classical reference [40]. Other references for using TRS with a given parameter ε appears in e.g. [2]; and, more recently, a parameterized trust region approach to find the regularized solution is used in [35,34]. Another approach for regularization uses the conjugate gradient method, CGLS. This method is particularly efficient when an estimate for the norm of the error $\|e\|_2$, is known. Stopping rules for CGLS, for finding a regularized solution are given in [29,17].

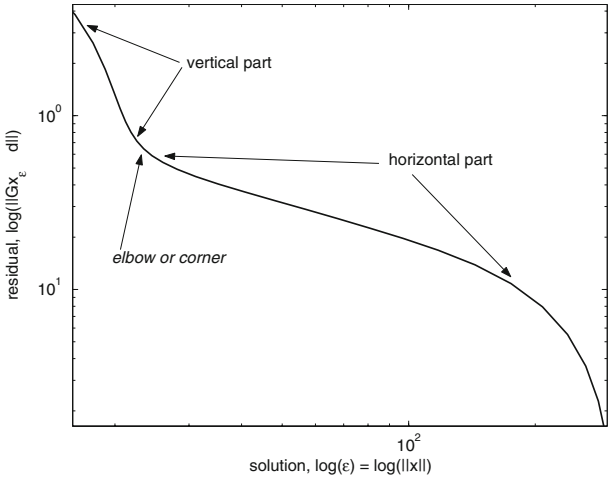

Fig. 1 The L-curve for the deblurring problem

1.1 Outline of main results

Each iteration of the parameterized trust region approach in [32] finds an optimal solution x_ε for different values of ε, i.e. it repeatedly computes points on the L-curve until it locates the correct point corresponding to a desired ε. In this paper we dynamically change ε and exploit the connections between ε, α and various other parameters used in [32], to efficiently move along the L-curve and estimate the region near the point of maximum curvature. This results in a regularized solution $x_\alpha = x_\varepsilon$. In addition, we compare our approach to the CGLS method, see e.g. [29].

In Sect. 2 we present the basic regularization theory that we need along with the analytic description of the L-curve. The L-curve curvature is discussed in Sect. 5. In Sect. 3 we apply the parametric TRS approach for regularization and show how it can be used to efficiently control the regularization parameter using the trust region radius ε. The details of our algorithm for finding the *good (correct)* regularization parameter are given in Sect. 4.

Numerical results are presented in Sect. 5. In Sect. 5.1 we consider an image restoration example. Concluding remarks are given in Sect. 6.

2 The L-curve

It is well known that the singular value decomposition (SVD) of the matrix G simplifies the L-curve analysis, see e.g. [19]. We write the SVD as $G = USV^T$, where matrix S is a diagonal $n \times n$ matrix consisting of singular values $\sigma_1 \le \cdots \le \sigma_n$, and U, V are orthogonal matrices. It follows that

$$\begin{aligned} \|x_\alpha\|_2^2 &= \textstyle\sum_{i=1}^n f_i^2 \left(\frac{U_{:i}^T d}{\sigma_i}\right)^2, \\ \|Gx_\alpha - d\|_2^2 &= \textstyle\sum_{i=1}^n (1-f_i)^2 \left(U_{:i}^T d\right)^2, \end{aligned} \tag{6}$$

where $f_i = \dfrac{\sigma_i^2}{\sigma_i^2 + \alpha^2}$ are the so-called *Tikhonov filter factors*. Note that $\|x_\alpha\|_2 < \|x_0\|_2$, $\forall \alpha > 0$, with $\|x_\alpha\|_2 \to 0$ as $\alpha \to \infty$. If G is invertible, then setting $\alpha = 0$ gives the unique solution x_0, i.e. $\|Gx_0 - d\|_2 = 0$ as all filter factors are equal to one. Moreover, adding uncorrelated noise η results in

$$\|x_0\|_2^2 = \sum_{i=1}^n \left(\frac{U_{:i}^T d_{\text{true}}}{\sigma_i} + \frac{U_{:i}^T \eta}{\sigma_i}\right)^2. \tag{7}$$

The error contributions from η in (7) can be large when the noise vector is orthogonal to the singular vectors $U_{:i}$'s corresponding to small singular values.

The situation continues to be problematic in the case of small singular values, even if the noise component is absent, i.e. the ratios $\frac{U_{:i}^T d_{\text{true}}}{\sigma_i}$ in (6) imply that we require:

$$\boxed{\text{the Fourier coefficients } |U_{:i}^T d_{\text{true}}| \text{ decay faster than the } \sigma_i} \tag{8}$$

This condition, known as *the Discrete Picard Condition*, e.g. [22], guarantees that the least-squares solution has a reasonable norm and thus is physically meaningful.

Example 1 (Failure of Picard condition; with no noise) We consider a Shaw problem from the Hansen MATLAB package (see [20]) with $n = 32$. This is a one-dimensional image restoration problem which is constructed via discretization of a Fredholm integral equation of the first kind (see [35]). The MATLAB **shaw** command produces the matrix G and the right-hand side vector d_{true}, as well as the true solution vector x_{true}.

We plot the Fourier coefficients $|U_{:i}^T d_{\text{true}}|$, the singular values σ_i and the ratio $\frac{|U_{:i}^T d_{\text{true}}|}{\sigma_i}$ in Fig. 2, marked with o, $\times$, and $-$, respectively. The Picard condition holds until the singular values (line marked with $\times$) reach the machine epsilon level (horizontal dashed line). The Picard condition fails for the larger indices due to round-off error. The norm of the least-squares solution computed using the SVD is $\sim 10^5$; while the true solution has norm $= 10$. A good approximation of the true solution is still recoverable via a truncated SVD, i.e. by setting all the singular values less than machine epsilon to 0.

We now study the L-curve described in (5), (see e.g. [22]. The curve usually features a strong L-shaped form with almost linear vertical and horizontal parts and a well distinguishable elbow or corner. (In this paper we use the nonstandard L-curve with the abscissa as $\log(\|x_\varepsilon\|_2)$, or equivalently $\log(\|x_\alpha\|_2)$.)

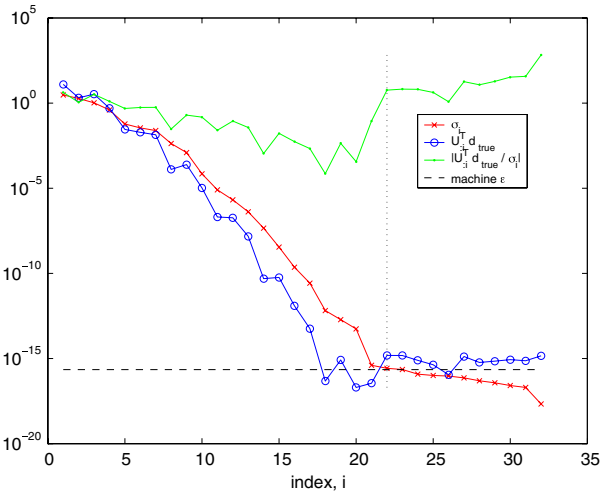

Fig. 2 Picard plot; Shaw problem; Example 1

In the presence of noise, as in (6),

$$\|x_\alpha\|_2^2 = \sum_{i=1}^{n} f_i^2 \left(\frac{U_{:i}^T d_{\text{true}}}{\sigma_i} + \frac{U_{:i}^T \eta}{\sigma_i} \right)^2 .$$

If we assume uncorrelated noise, then the expected value of the Fourier coefficients of η is independent of i, $\mathcal{E}\left(|U_{:i}^T \eta|\right) \approx \|\eta\|_2, \forall i$, i.e. the Picard condition can fail if there are small singular values.

Example 2 (Failure of Picard condition; with noise)
We consider a deblurring of a 20×20 image. (See Sect. 5.2 for problem details.)

1. Figure 3 shows the Picard plot for the unperturbed right-hand side. (We use the same markings as in Fig. 2, as explained in Example 1.) On average the Fourier coefficients corresponding to the unperturbed data vector (marked with the thick dark line) decay faster than the singular values. Hence, the Picard condition holds and the least-squares solution is meaningful in the absence of noise.
2. The Fourier coefficients for the noise vector η are plotted in Fig. 4. (We use the same markings as in Item 1 above.) Now on average the Fourier coefficients and the singular values stay on the same level and hence fail to satisfy the Picard condition.
3. As expected, the Picard plot for the Fourier coefficients for the perturbed (noisy) right-hand side $d_{\text{true}} + \eta$ levels off at approximately $\|\eta\|_2$; see Fig. 5, i.e. the Picard condition fails. (We use the same markings as in Item 1 above).

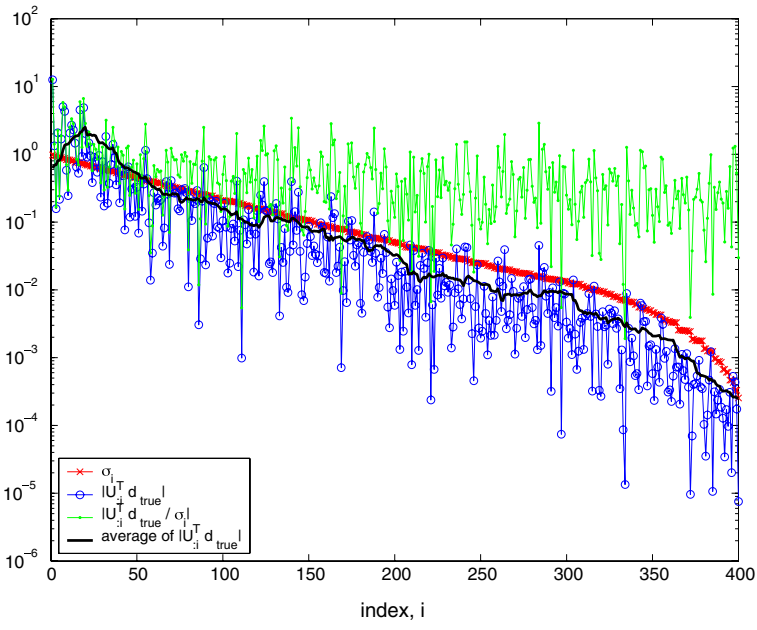

Fig. 3 Picard plot unperturbed right-hand side; Example 2, Item 1

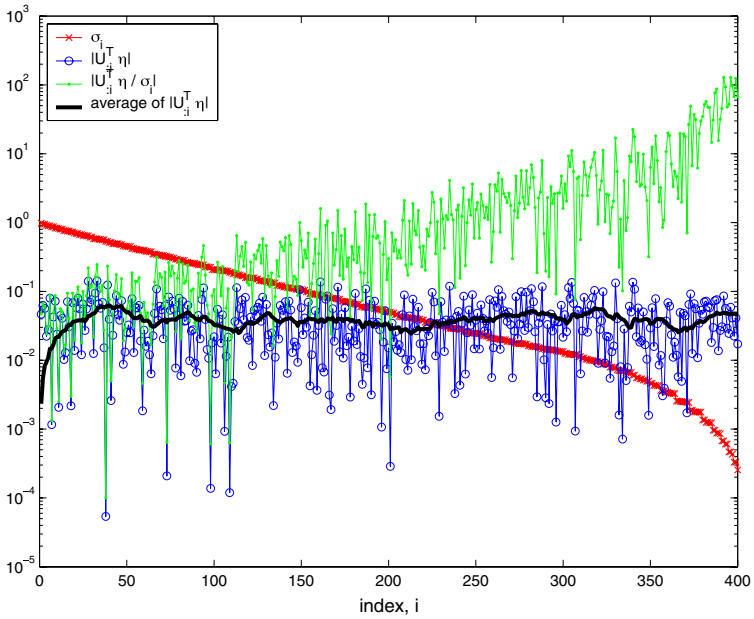

Fig. 4 Picard plot; noise vector; Example 2, Item 2

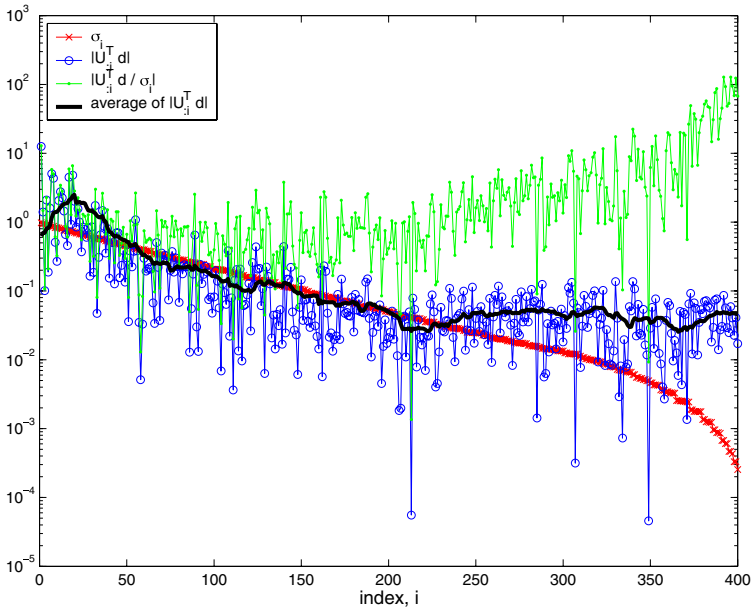

Fig. 5 Picard plot; perturbed right-hand side; Example 2, Item 3

For positive values of α, the filter factors f_i are approximately 1 if $\sigma_i \gg \alpha$ and approximately 0 if $\sigma_i \ll \alpha$. The filter factors control which terms in the summation contribute to the norm of the residual and the solution (6). The vertical part of the

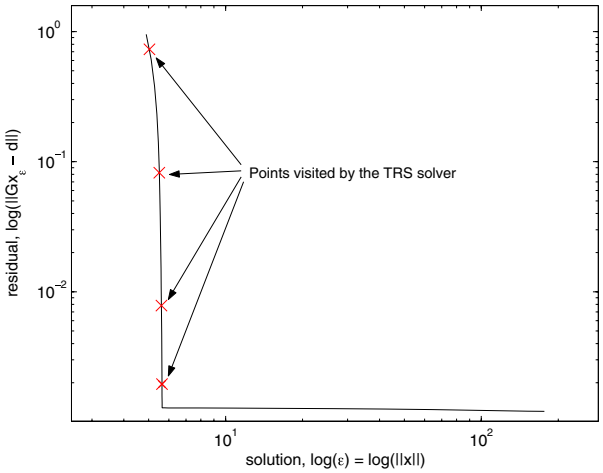

Fig. 6 Points on L-curve, while solving TRS

L-curve in Fig. 1 demonstrates that when the regularization parameter α is large (equivalently $\|x_\varepsilon\|_2 = \varepsilon$ is small), then the norm of the residual varies greatly with α, but, the norm of the solution $\|x_\alpha\|_2$ is almost unaffected, since all the terms corresponding to the small singular values are filtered out (this is also known as *oversmoothing* a solution). On the other hand, when α is small (equivalently the trust region radius ε is large), then small changes in α result in small changes in the norm of the residual; but, the small changes in α can cause large changes in the norm of the solution. This corresponds to the horizontal part of the L-curve.

Depending on the particular Picard plot, the smoothness of the transition between the vertical and horizontal parts can vary in a broad range. For example, the L-curve for the deblurring problem in Fig. 1 is not strongly L-shaped, though it is still possible to locate a distinguishable elbow. This is in sharp contrast to the L-curve in Fig. 6, where a distinct elbow is visible. This discussion is relevant only when the log–log scale is used. In a linear scale the plot is always convex, see e.g. [24].

This behaviour leads to the *L-curve criterion* for choosing the regularization parameter proposed in [19,23], i.e. one chooses the value of the parameter that corresponds to a point on the L-curve with maximum curvature (details on curvature calculation are given in Sect. 4.5). A point of maximum curvature coincides with an elbow that separates the regions where the solution is dominated by regularization errors (oversmoothing) and perturbation errors.

3 Trust region subproblem, TRS

In this section, we recall some of the details in the Rendl–Wolkowicz TRS algorithm, [32,8], denoted RW, and apply them to the regularization problem. We show that the RW algorithm visits a point on the L-curve at each iteration. Therefore, we can

dynamically change the trust region radius Δ to steer the algorithm to the elbow of the L-curve.

It is known ([10,37]) that $x^* = x_\varepsilon$ is a solution to TRS if and only if:

$$\left.\begin{array}{c} (A-\lambda^* I)x_\varepsilon = a, \\ A-\lambda^* I \succeq 0, \lambda^* \le 0 \end{array}\right\} \quad \text{dual feasibility}$$
$$\|x_\varepsilon\|_2^2 \le \varepsilon^2 \quad \text{primal feasibility} \tag{9}$$
$$\lambda^*(\|x_\varepsilon\|_2^2 - \varepsilon^2) = 0 \quad \text{complementary slackness}$$

for some (Lagrange multiplier) $\lambda^* = \lambda_\varepsilon$. The above conditions connect Tikhonov regularization (4) ([40]) with TRS , i.e. solving (3) with a particular value of the regularization parameter α, $\alpha^2 = -\lambda_\varepsilon$

$$(A - \lambda_\varepsilon I)x_{\lambda_\varepsilon} = a \tag{10}$$

is equivalent to solving (2) with a corresponding value of Δ. Also, for our applications $\lambda^* \le 0 < \lambda_1(A)$. Therefore, the optimal solution always lies on the boundary, $\|x^*\|_2 = \|x_\varepsilon\|_2 = \varepsilon$, and the so-called *easy case* holds for TRS . The objective value of (2) and ε correspond to a unique point on the L-curve, and vice-versa.

3.1 Building the L-curve using TRS

Example 3 (Moving along the L-curve) Figure 6 presents an L-curve for a sample Shaw problem created using the Hansen MATLAB package, regu([20]). The RW algorithm was then applied with a fixed trust region radius $\Delta = 6$. (The TRS optimum yields a point near the elbow.) It took 8 iterations to solve the TRS with a desired optimality tolerance $\delta = 10^{-8}$. We obtained four points on the L-curve. (The other four are located outside the interval of interest.) These four points give enough information to approximate the vertical part of the L-curve to the left of the elbow.

By exploiting the strong Lagrangian duality of TRS (see [36]), TRS can be reformulated as an unconstrained concave maximization problem, i.e.

$$\mu_\varepsilon = \min_x \max_{\lambda \le 0} L(x,\lambda) = \max_{\lambda \le 0} \min_x L(x,\lambda),$$

where $L(x,\lambda) = x^T Ax - 2a^T x + \lambda(\varepsilon^2 - \|x\|_2^2)$ denotes the *Lagrangian* of TRS . Define the symmetric $(n+1)\times(n+1)$ matrix

$$D(t) = \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix}, \tag{11}$$

and let $\lambda_1(D(t))$ denote its smallest eigenvalue. Further define the concave function

$$k(t) = (\varepsilon^2 + 1)\lambda_1(D(t)) - t, \quad t \in \mathbb{R}. \tag{12}$$

Then an unconstrained dual problem is given by

$$\mu_\varepsilon = \max_t k(t). \tag{13}$$

Furthermore, under assumptions of the *easy case*, $\lambda_1(D(t))$ is a singleton eigenvalue, and the derivative of $k(t)$ satisfies

$$k'(t) = (\varepsilon^2 + 1)y_0^2 - 1, \tag{14}$$

where $\begin{pmatrix} y_0 \\ x \end{pmatrix}$ is the normalized eigenvector for $\lambda_1(D(t))$, scaled so that $y_0 \geq 0$. (Note $y_0 > 0$ indicates the *easy case*.)

The L-curve is formed using ε in TRS as a parameter and finding the residual for the corresponding optimal x_ε. We now see that the L-curve can be formed using any of the following parameters:

t – control parameter in $K(t)$, $D(t)$,
ε – trust-region radius, norm of the solution $\|x_\varepsilon\|_2$,
α – Tikhonov regularization parameter,
λ_ε – optimal Lagrange multiplier for TRS .

From Sect. 2, we have $\lambda_\varepsilon = -\alpha^2$. However, changing between λ_ε, t and ε is computationally expensive. The following lemmas describe some of the relationships.

Lemma 1 *Given the parameter $\lambda_\varepsilon < 0$, the corresponding values of t and ε are related by*

$$\begin{aligned} t &= \lambda_\varepsilon + d^T G(G^T G - \lambda_\varepsilon I)^{-1} G^T d, \\ \lambda_\varepsilon &= \lambda_1(D(t)), \\ \varepsilon^2 &= d^T G(G^T G - \lambda_\varepsilon I)^{-2} G^T d. \end{aligned} \tag{15}$$

Proof The formula for t follows from Proposition 3.1 and Corollary 3.4 in [32]. The formula for ε follows from the optimality conditions (9), since the optimal solution x^* to TRS , that corresponds to the Lagrange multiplier $\lambda^* = \lambda_\varepsilon$, lies on the boundary. □

Lemma 2 *Given the parameter $t < d^T d$, the corresponding values of λ_ε and ε are given by*

$$\begin{aligned} \lambda_\varepsilon &= \lambda_1(D(t)), \\ \varepsilon^2 &= \frac{1 - y_0(t)^2}{y_0(t)^2}, \end{aligned} \tag{16}$$

where $y(t)$ is the normalized eigenvector corresponding to $\lambda_1(D(t))$ and $y_0(t)$ is its first component.

Proof The results follow from Theorem 3.7 in [32], e.g. we use the normalized eigenvector $\begin{pmatrix} y_0(t) \\ \bar{x}(t) \end{pmatrix}$ and find that

$$x_t = \frac{1}{y_0(t)} \bar{x}(t) \tag{17}$$

is a solution for TRS . This implies

$$y_0(t)^2 + \|\bar{x}(t)\|_2^2 = 1, \quad \|\bar{x}(t)\|_2 = y_0(t)\varepsilon.$$

□

Corollary 1 *The derivatives*

$$\lambda'_\varepsilon = \frac{\mathrm{d}\lambda_\varepsilon}{\mathrm{d}t} = (1+\varepsilon^2)^{-1} > 0, \tag{18}$$

$$\varepsilon' = \frac{\mathrm{d}\varepsilon}{\mathrm{d}t} = \frac{-(1+\varepsilon^2)}{2\varepsilon}\frac{\lambda''_\varepsilon}{\lambda'_\varepsilon} = \frac{1}{\varepsilon}\sum_{j\neq 1}\frac{y_j^2}{\lambda_j - \lambda_\varepsilon} > 0, \tag{19}$$

where $v_j = \binom{y_j}{x_j}$, *denotes the normalized eigenvectors of* $D(t)$ *for eigenvalues* λ_j, $j \neq 1$, *and*

$$\lambda''_\varepsilon = -2\sum_{j\neq 1}\frac{(y_j)^2}{(1+\varepsilon^2)(\lambda_j - \lambda_\varepsilon)}. \tag{20}$$

Proof From (14) and (16), the derivative of the smallest eigenvalue (see e.g. [31])

$$\begin{aligned}\frac{\mathrm{d}\lambda_\varepsilon}{\mathrm{d}t} &= \frac{\mathrm{d}\lambda_1(D(t)}{\mathrm{d}t} \\ &= \begin{pmatrix} y_0 \\ x \end{pmatrix}^T \frac{\mathrm{d}D(t)}{\mathrm{d}t}\begin{pmatrix} y_0 \\ x \end{pmatrix} \\ &= \begin{pmatrix} y_0 \\ x \end{pmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} y_0 \\ x \end{pmatrix} \\ &= y_0^2.\end{aligned}$$

This yields (18).

For a given ε, we have $k'(t) = (\varepsilon^2+1)\frac{\mathrm{d}\lambda_\varepsilon}{\mathrm{d}t} - 1 = 0$ at the optimum t. We can differentiate both sides with respect to t and solve for $\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}$ to get (19). The formula for λ''_ε can be found in [24,25,31], i.e. using $\frac{\mathrm{d}^2 D(t)}{\mathrm{d}t^2} = 0$, we get

$$\lambda''_\varepsilon = 2\sum_{j\neq 1}\frac{(y_j y_0)^2}{\lambda_\varepsilon - \lambda_j} = -2\sum_{j\neq 1}\frac{(y_j)^2}{(1+\varepsilon^2)(\lambda_j - \lambda_\varepsilon)}. \tag{21}$$

□

Lemma 3 *Given the parameter* $\varepsilon < \left\|G^{-1}d\right\|_2$, *the corresponding values of* t *and* λ_ε *can be obtained by solving* TRS *using the* RW *algorithm. The corresponding optimal solution satisfies* $\|x_\varepsilon\|_2 = \varepsilon$.

Proof The RW algorithm solves TRS with a fixed trust region radius t producing the optimal solution x_ε, the optimal Lagrange multiplier $\lambda^* = \lambda_\varepsilon$, and the corresponding parameter ε. And, $\|x_\varepsilon\|_2 = \varepsilon$, since the norm of the unconstrained minimum $\|x^*\|_2 = \left\|A^{-1}a\right\|_2 = \left\|G^{-1}d\right\|_2$. □

Combining the above lemmas we conclude that each of $t, \lambda_\varepsilon, \varepsilon, \alpha$ can be interchangeably used to parameterize the regularization problem and, hence, to describe points on the L-curve. In addition, we observe that the norm of the regularized solution is a monotonic function of these parameters.

Lemma 4 *Let* $\|x_t\|_2, \left\|x_{\lambda_\varepsilon}\right\|_2, \|x_\varepsilon\|_2, \|x_\alpha\|_2$ *denote the norms of the solutions of* (17),(10), (4), (3)*, respectively. Then*:

1. $\|x_t\|_2$ *is a monotonically increasing function of* t;
2. $\left\|x_{\lambda_\varepsilon}\right\|_2$ *is a monotonically increasing function of* λ_ε;
3. $\|x_\varepsilon\|_2$ *is a monotonically increasing function of* ε;
4. $\|x_\alpha\|_2$ *is a monotonically decreasing function of* α.

Therefore, the four parameters $t, \lambda_\varepsilon, \varepsilon, -\alpha^2$ *are pairwise isotonic.*

Proof The lemma follows from Theorem 3.7 in [32]. □

3.2 Parameter intervals of interest

The interval of uncertainty for the Tikhonov regularization parameter is $\alpha^2 < \infty$. Using the results in Sect. 3.1, this corresponds to the following.

Corollary 2 *The intervals of uncertainty for the parameters are*:

$$\begin{array}{rl} -\infty < \lambda_\varepsilon = \lambda_1(D(t)) = -\alpha^2 & \leq 0 \\ 0 < t = \lambda_\varepsilon + d^T G(G^T G - \lambda_\varepsilon I)^{-1} G^T d & \leq \|d\|_2^2 \\ 0 < \varepsilon = \left\|(G^T G - \lambda_\varepsilon I)^{-1} G^T d\right\|_2 & \leq \left\|G^{-1} d\right\|_2 \end{array}$$

The upper bounds correspond to the linear least squares solution.

Proof Follows directly from Lemmas 1 and 2. □

Note that if the largest singular value $\sigma_n = \sigma_n(G)$ is known, then the results of Sect. 2 imply that $-\sigma_n^2$ is a lower bound on λ_ε. The regularization algorithm keeps the values of the parameters within the described intervals.

4 Regularization algorithm

4.1 Flowchart

Algorithm 1: Trust-Region Based Regularization

```
♯ Initialization:
    λ = λ_low = −σ_n^2(G), λ_up = 0
    nd corresponding t(, x_λ, k(t)), for λ using (15), (10), (12)
    ε^2 = x_λ^T x_λ
    [κ_low^previous, κ_up^previous] = [κ_low, κ_up] = curvature(ε, k + d^T d, λ)
    iteration = 1
♯ Find three appropriate points on the L-curve:
while not (iteration ≥ 3 and κ_low > κ_up^previous) (max curv. region not yet defined)
do
    t = t − (ε^2 + 1)λ (triangle interpolation (22))
    ♯ Updates:
    nd corresponding λ = λ_1(D(t)), x_t, k(t)), for t, using (16),(17),(12)
    ε^2 = x_t^T x_t
    [κ_low, κ_up] = curvature(ε, k + d^T d, λ)
    if necessary, update bidiagonalization of G to improve precision
    κ_up^previous = κ_up
    iteration = iteration + 1
end
♯ Use simple interval bisection and the convex region defined by the last three
  generated points on the L-curve, to estimate the point of maximum curvature.
```

The interval used to estimate the curvature, $[\kappa_{low}, \kappa_{up}]$ = **curvature** $(\varepsilon, r^2, \lambda)$ is found by computing the lower and upper bounds (28), (29) using the current Lanczos bidiagonalized approximation of the matrix G, Sect. 4.5.

4.2 Initial L-curve point

Each iteration of our algorithm increases the value of ε, i.e. subsequent points are located to the right of previous ones. Hence, the initialization involves finding a value of λ (or t) located to the left of the elbow of the L-curve. One option is to start with the point corresponding to $\lambda = -\sigma_{\max}^2(G) = -\sigma_n^2(G)$, see Sect. 3.2.

If the largest singular value $\sigma_n(G)$ of the matrix G is not available, then the algorithm can be started with a point associated with a small enough value of λ, e.g. $\lambda = \frac{d^T d}{2}$. As discussed in Sect. 2, a *well-shaped* L-curve plot can be viewed as a linear horizontal plateau to the right of the elbow and a linear vertical part to the left of the elbow. For well shaped L-curve plots, small changes in ε would result in large changes in λ when we are on the horizontal part. Conversely, large changes in ε result in small

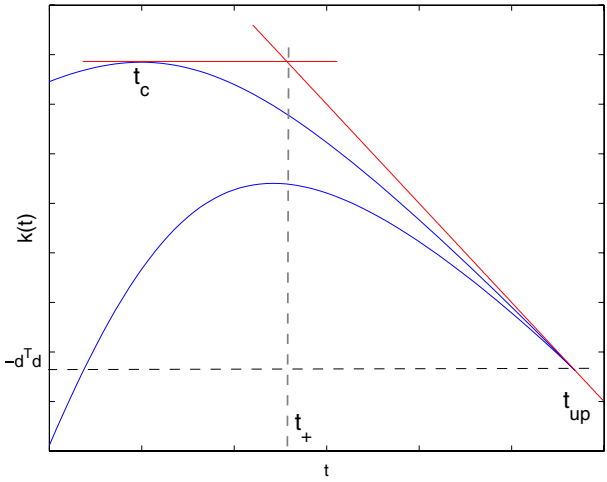

Fig. 7 k(t) and triangle interpolation

changes in ε when we are on the vertical part. This is explained by the structure of the singular value decomposition of the matrix G. The behaviour remains true for less well-behaved L-shaped plots. This tells us that points that lie on the plateau region correspond to the values of ε that are very close to $d^T d$. Thus, taking half of this value will put us onto the vertical part to the left of the elbow.

4.3 Computation of the step length

As Lemma 4 suggests, we can generate a point on the L-curve strictly to the right of the current point by increasing any of the parameters t, λ_ε, or ε. Our algorithm explicitly increases t by employing the relation between t, λ_ε and ε.

The key idea lies in the properties of the function $k(t) = k_\varepsilon(t)$. Recall (Sect. 3.1) that for each $\varepsilon > 0$, $k(t) = (\varepsilon^2+1)\lambda - t$ and $\mu_\varepsilon = \max_t k(t)$. Also, the function $k(t)$ is concave. Given the current iterate $t_c < t_{up}$ and the corresponding ε_c, we consider the current function $k(t)$ with $\varepsilon = \varepsilon_c$, which attains its maximum at the point $t = t_c$. Consider also the point $t_{up} = d^T d$. From Sect. 3.2 we know that $k(t_{up}) = -t_{up} = -d^T d$. Moreover, the derivative $k'(t_{up}) = -1$ does not depend on ε.

Figure 7 depicts what happens at the main step of the algorithm. The intersection of the tangent lines to the curve $k(t)$ at the current point t_c and at the upper bound point t_{up} yields a new point t_+, i.e. we exploit the slope $k'(t_{up}) = -1$ to get

$$\begin{aligned} t_{up} - t_+ &= k(t_c) - k(t_{up}) \\ t_+ &= t_{up} - k(t_c) + k(t_{up}) \\ t_+ &= -k(t_c). \end{aligned}$$

This gives the explicit expression for the new t,

$$t_+ = t_c - (\varepsilon^2 + 1)\lambda_1(D(t_c)). \tag{22}$$

Suppose that we have convergence in ε. If we take the limit on both sides of (22), then t cancels on both sides and we get $0 = (\varepsilon^2 + 1)\lambda_1(D(t_*))$, i.e. the only limit point t^* satisfies $\lambda_1(D(t_*)) = 0$ yielding $t^* = t_{up} = d^T d$. Therefore, the iteration cannot terminate prematurely and $t_{+1} > t_c$ at each iteration. So we only have to worry about stepping too far, in which case we backtrack appropriately. Moreover, as ε gets closer to $\lambda_1(A)$, the curve $\kappa(t)$ becomes linear to the right of t_c with slope -1. The geometry indicates that the step length gradually decreases as we go along the L-curve, allowing for exploring of the elbow.

With the new t we can compute the corresponding x_ε and ε as Lemma 2 suggests. Particular details on the eigenvalue computation of $\lambda_1(D(t_c))$ are given in Sect. 5.1.

4.4 Termination and elbow estimation

The termination condition is based on the L-curve maximum curvature criterion discussed in Sect. 3.1. The algorithm looks for a point on the L-curve that has the maximum (negative) curvature. Since the L-curve has a convex profile near the elbow, it is possible to isolate a region with the maximum curvature by keeping track of the curvature as the algorithm goes along the L-curve. The region of interest is defined by three points such that the middle one has a larger curvature value than the other two. After this profile is obtained, a simple bisection procedure is then performed to estimate the elbow location.

4.5 Curvature of the L-curve

Following [22], see also [18,19,23], let

$$\zeta := \|x_\varepsilon\|_2^2, \quad \hat{\zeta} := \log \zeta; \quad \rho := \|Gx_\varepsilon - d\|_2^2 = \mu_\varepsilon + d^T d, \quad \hat{\rho} := \log \rho.$$

And, recall that $\lambda^* = \lambda_\varepsilon$ is the optimal Lagrange multiplier for TRS . The L-curve is a plot of $\hat{\zeta}/2$ versus $\hat{\rho}/2$. Then the curvature κ of the L-curve, as a function of ε, is given by

$$\kappa_\varepsilon = 2\frac{\hat{\rho}'\hat{\zeta}'' - \hat{\rho}''\hat{\zeta}'}{\left((\hat{\rho}')^2 + (\hat{\zeta}')^2\right)^{3/2}}. \tag{23}$$

From Sect. 3, $\zeta = \varepsilon^2$, and therefore,

$$\hat{\zeta}' = \frac{\zeta'}{\zeta} = \frac{2}{\varepsilon} \quad \text{and} \quad \hat{\zeta}'' = -\frac{2}{\varepsilon^2}.$$

Furthermore,

$$\hat{\rho}' = \frac{\rho'}{\rho} = \frac{\mu_\varepsilon'}{\mu_\varepsilon} \quad \text{and} \quad \hat{\rho}'' = \frac{\mu_\varepsilon''\mu_\varepsilon - (\mu_\varepsilon')^2}{\mu_\varepsilon^2}.$$

Substituting these expressions into (23) we get

$$\begin{aligned}\kappa_\varepsilon &= 2\left(-\frac{\mu'_\varepsilon}{\mu_\varepsilon}\frac{2}{\varepsilon^2} - \frac{\mu''_\varepsilon\mu_\varepsilon-(\mu'_\varepsilon)^2}{\mu_\varepsilon^2}\frac{2}{\varepsilon}\right)\left(\left(\frac{\mu'_\varepsilon}{\mu_\varepsilon}\right)^2+\left(\frac{2}{\varepsilon}\right)^2\right)^{-3/2}\\ &= 4\varepsilon\mu_\varepsilon\left(\varepsilon(\mu'_\varepsilon)^2-\mu_\varepsilon\mu'_\varepsilon-\varepsilon\mu_\varepsilon\mu''_\varepsilon\right)\left(\varepsilon^2(\mu'_\varepsilon)^2+4\mu_\varepsilon^2\right)^{-3/2}\\ &= \varepsilon^2\mu_\varepsilon\left(2\varepsilon^2\lambda^{*2}-2\mu_\varepsilon\lambda^*-\varepsilon\mu_\varepsilon\left(\frac{\partial\lambda^*}{\partial\varepsilon}\right)\right)\left(\varepsilon^4\lambda^{*2}+\mu_\varepsilon^2\right)^{-3/2}.\end{aligned} \tag{24}$$

The last equality follows from (26) and (27) derived below.

The derivative $\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}$ can be found using implicit differentiation on the equation

$$\left\|(A-\lambda_\varepsilon I)^{-1}a\right\|_2^2-\varepsilon^2=0,$$

obtained after the substitution $x_\varepsilon=(A-\lambda_\varepsilon I)^{-1}a$. We get

$$\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}=\frac{\varepsilon}{a^T(A-\lambda_\varepsilon I)^{-3}a}. \tag{25}$$

Moreover, the optimal value

$$\begin{aligned}\mu_\varepsilon &= (x_\varepsilon)^TAx_\varepsilon-2a^Tx_\varepsilon\\ &= (x_\varepsilon)^TAx_\varepsilon-2a^Tx_\varepsilon-\lambda_\varepsilon(\|x_\varepsilon\|_2^2-\varepsilon^2)\\ &= -a^T(A-\lambda_\varepsilon I)^{-1}a+\lambda_\varepsilon\varepsilon^2.\end{aligned}$$

Then, using $a^T(A-\lambda_\varepsilon I)^{-2}a-\varepsilon^2=\|x_\varepsilon\|_2^2-\varepsilon^2=0$, we get

$$\begin{aligned}\frac{\partial\mu_\varepsilon}{\partial\varepsilon} &= a^T(A-\lambda_\varepsilon I)^{-2}a\left(-\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}\right)+\left(\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}\right)\varepsilon^2+2\lambda_\varepsilon\varepsilon\\ &= \left(-\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}\right)(a^T(A-\lambda_\varepsilon I)^{-2}a-\varepsilon^2)+2\lambda_\varepsilon\varepsilon\\ &= 2\lambda_\varepsilon\varepsilon\end{aligned} \tag{26}$$

and

$$\frac{\partial^2\mu_\varepsilon}{\partial\varepsilon^2}=2\left(\lambda_\varepsilon+\varepsilon\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}\right). \tag{27}$$

More details on these and other perturbation results can be found in e.g. [38].

4.5.1 Curvature estimation and Gauss quadrature

The numerical evaluation of the curvature (24) requires the (expensive) derivative $\frac{\partial\lambda_\varepsilon}{\partial\varepsilon}=\varepsilon/\left(a^T(A-\lambda_\varepsilon I)^{-3}a\right)$, see (25). This issue is addressed in [8,12–14]. One approach lies in obtaining both the upper and lower bounds

$$l_p(\alpha)\le v_p(\alpha)=d^TG(G^TG+\alpha I)^pG^Td\le u_p(\alpha),$$

where $\alpha = -\lambda_\varepsilon$ is a positive scalar, p is a negative integer ($p = -3$, $G^T G = A$ and $G^T d = a$ in (25)). These bounds are obtained using an iterative procedure and become tighter as the number of iterations, k, increases. We brie y outline the idea.

After k iterations applied to G, the Lanczos Bidiagonalization algorithm (e.g. [11]) produces a $(k+1)$-by-k lower bidiagonal matrix

$$B_k = \begin{bmatrix} \gamma_1 & & & \\ \delta_1 & \ddots & & \\ & \ddots & \ddots & \\ & & \ddots & \gamma_k \\ & & & \delta_k \end{bmatrix}, \text{ with } GV = VB_k, \quad V = \begin{bmatrix} v_1 \dots v_k \end{bmatrix},$$

such that the Gauss and Gauss–Radau quadrature rules for $l_p(\alpha)$ are de ned as

$$l_p(\alpha) = \left\| G^T d \right\|_2^2 e_1^T (B_k^T B_k + \alpha I)^p e_1 = \|d\|_2^2 e_1^T B_k (B_k^T B_k + \alpha I)^p B_k^T e_1, \quad (28)$$

$$u_p(\alpha) = \left\| G^T d \right\|_2^2 e_1^T (\tilde{U_k}^T \tilde{U_k} + \alpha I)^p e_1. \quad (29)$$

Here $\tilde{U_k}$ is $(k+1)$-by-k upper bidiagonal matrix obtained from B_k by a sequence of Givens rotations and by setting the main diagonal to zero.

Our implementation of the Lanczos Bidiagonalization algorithm allows restarting from the speci ed iteration (with v_k) if optional parameters are supplied. This enables one to increase the precision when necessary. This feature is exploited by the main algorithm that iterates by gradually decreasing α. Since for $p < 0$ and nonsingular B_k we have that:

$$\lim_{\alpha \searrow 0} l_p(\alpha) < \infty, \quad \lim_{\alpha \searrow 0} u_p(\alpha) = \infty,$$

it is natural that the bounds weaken as $\alpha \searrow 0$, which means that the precision has to be increased in order to compare two curvature intervals for the consecutive points on the L-curve.

Note that evaluating the expressions for $G_p(\alpha)$ and $R_p(\alpha)$ implies solving linear systems:

$$(\tilde{U_k}^T \tilde{U_k} + \alpha I)x = e_1,$$
$$(B_k^T B_k + \alpha I)x = B_k^T e_1.$$

The above equations are the normal equations for the linear least-squares problem, LLS ,

$$\min \left\| \begin{bmatrix} \tilde{U_k} \\ \sqrt{\alpha} I \end{bmatrix} x - \begin{bmatrix} 0 \\ e_1/\sqrt{\alpha} \end{bmatrix} \right\|_2$$
$$\min \left\| \begin{bmatrix} B_k \\ \sqrt{\alpha} I \end{bmatrix} x - \begin{bmatrix} e_1 \\ 0 \end{bmatrix} \right\|_2.$$

This means that the solution x for the linear least-squares problem satisfies the original linear system as well. We may, however, exploit the structure of LLS problems and solve them efficiently by a sequence of Givens rotations that produces the QR factorization. This approach is described in [7,9,26].

5 Numerics

5.1 Eigensolver issues

As shown in Sect. 3.1, obtaining a new L-curve point for a corresponding value of α requires solving for the smallest eigenpair of the matrix $D(x)$. In the case G is large and sparse, the same is true for $D(x)$, so one should use matrix-free iterative algorithms to compute the eigenpairs, e.g. Lanczos methods. As α increases, the gap between the first two smallest eigenvalues may become numerically zero. This can slow down the eigensolver substantially.

Under such numerical degeneracy a computation may converge to a wrong eigenpair, giving an incorrect eigenvector and an incorrect regularized solution. One way to control the eigensolution is to start with an initial eigenvalue smaller than the estimated one and, at the same time, relatively close to it. For iterative algorithms, it is possible to store a previous eigenpair and re-use it on the next step as an initial guess. This works only if the eigenvalue is about to increase at every subsequent iteration. We have employed this method in our Regularization Algorithm and it proved to be very efficient. For the eigenpair computation we used the MATLAB's `eigs` routine which implements a Lanczos-type matrix-free algorithm.

A different approach is to apply a spectral transformation to separate the first eigenvalue from the rest of the spectrum, i.e. preconditioning. In particular, a Tchebyshev polynomial transformation is discussed in [38] and [34].

5.2 Image deblurring example

We demonstrate how the algorithm works by considering a sample problem of deblurring an image. Problems of this nature occur often. For instance, one might need to deblur a photo taken by a space telescope or a satellite, see e.g. the forthcoming book [41].

For this particular example we take an image generated by the Hansen MATLAB package ([20]). Figure 8 shows the image generated by the `blur` command. This command also produces the blurring matrix G and the right-hand side d, i.e. observed data, computed as $d = Gx_{\text{true}} + \eta$. Where η represents the noise. Figure 9 shows the observed image.

The generated image is a 40-by-40 grayscale picture, which is stored as a vector x_{true} of size 1,600. The matrix G stands for the operator that represents degradation of the image caused by atmospheric turbulence blur, modelled by a Gaussian point-spread function,

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

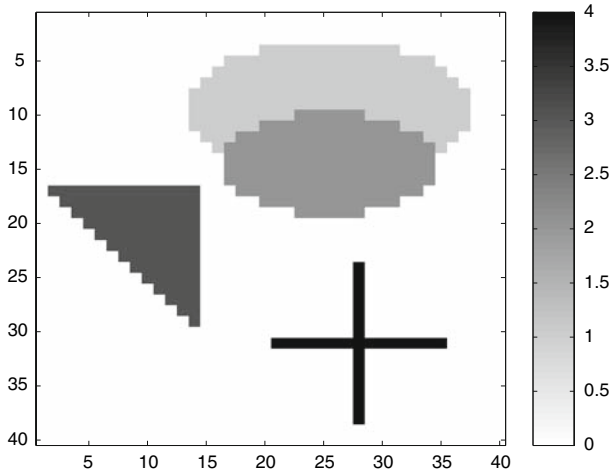

Fig. 8 Image deblurring example: original picture

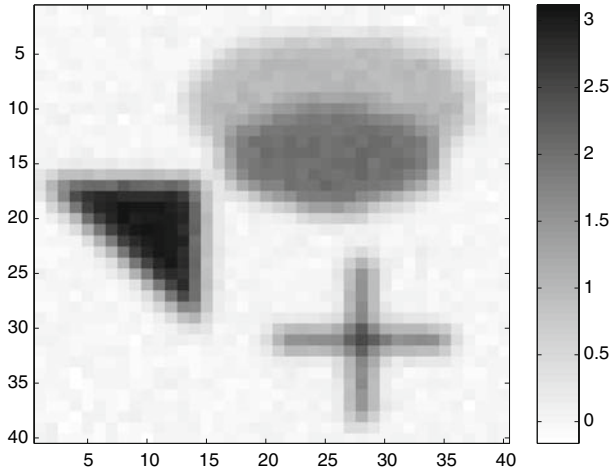

Fig. 9 Image deblurring example: observed data, blurred with added noise

Here the parameter a controls the smoothness (by defining the shape of the Gaussian point spread), and b stands for the bandwidth. It also follows that matrix G is sparse leading to a large sparse problem.

For our example we fix the parameters to be $a = 1$, $b = 5$. Noise η has a normal distribution with the mean of 0 and the standard deviation 0.05.

By computing the SVD we construct the true L-curve to get an idea where the solution is located. We can see that the curve is not strongly L-shaped, but both vertical and horizontal parts are still distinguishable (See Fig. 10). We also build a plot (dashed line) that shows how well the points on the L-curve approximates the true solution, i.e. for every point x we determine the quantity $\frac{\|x_{\text{true}} - x\|_2}{\|x_{\text{true}}\|_2}$ which we treat as the relative accuracy; the smaller the value, the better the approximation we obtain.

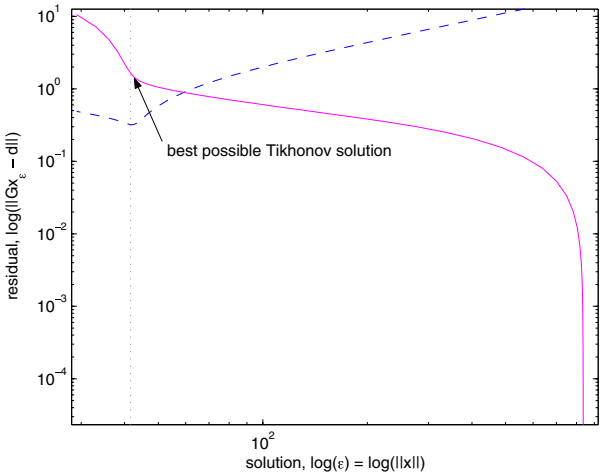

Fig. 10 Image deblurring example: corresponding L-curve

The minimum corresponds to the best possible solution that can be obtained using the Tikhonov regularization approach.

Figure11 shows points visited by the RPTRS Algorithm. For each point we present an associated solution image. See Figs. 12, 13, 14, 15, 16, 17. We can follow how the solution transforms as we go along the curve. For smaller values of the parameter the solution appears to be very smooth. The noise components are almost eliminated for these solutions. However, as we increase the regularization parameter, the noise starts to evolve. At the same time, pictures become sharper and represent a better approximation to the true solution. This behaviour continues until we hit the point #5 (see Fig16). Suddenly, the noise components overcome the real signal and the

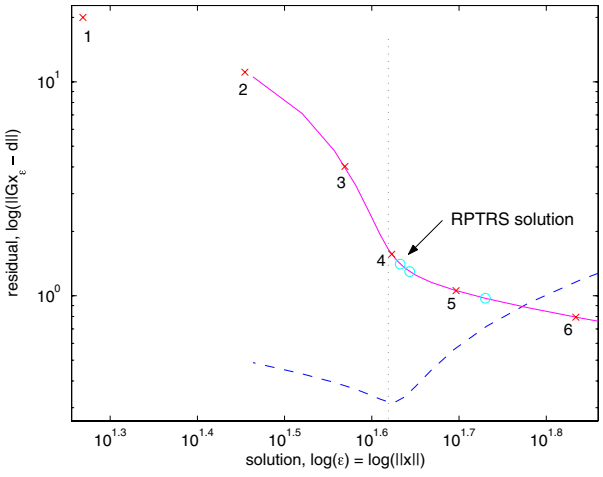

Fig. 11 Image deblurring example: corresponding L-curve with RPTRS points

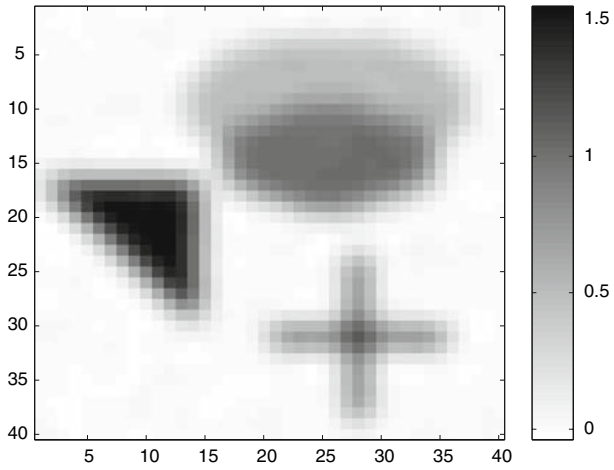

Fig. 12 Image deblurring example: point #1, $\tau = 652.166$, rel. acc. = 65.39%

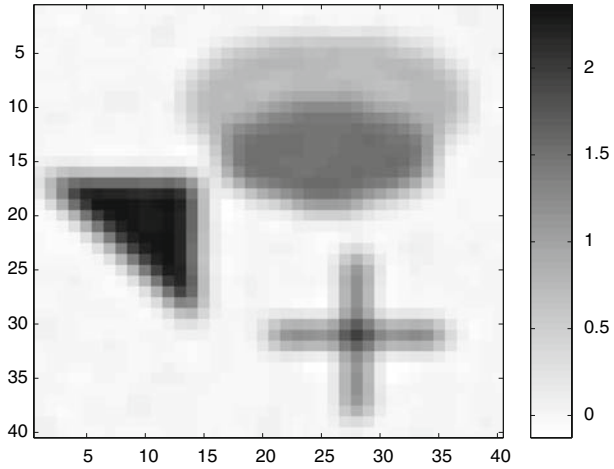

Fig. 13 Image deblurring example: point #2, $\tau = 994.155$, rel. acc. = 49.63%

solution becomes less distinguishable. The situation becomes even worse at the last point. The least-squares solution consists mostly of the noise components and contains practically no signal information.

The algorithm observes the changes in the curvature value and backtracks locating the elbow. Figure 11 demonstrates the undertaken steps. Points marked with a cross are visited during the main loop, and stars denote the final refinement steps. Note the proximity of the regularized solution to the best possible Tikhonov solution. The final RPTRS solution is shown on Fig. 18.

For more information and techniques on image de-blurring problems, we note the ongoing research based on wavelets (see e.g. [4–6]). We do not perform any comparison with these techniques in this paper.

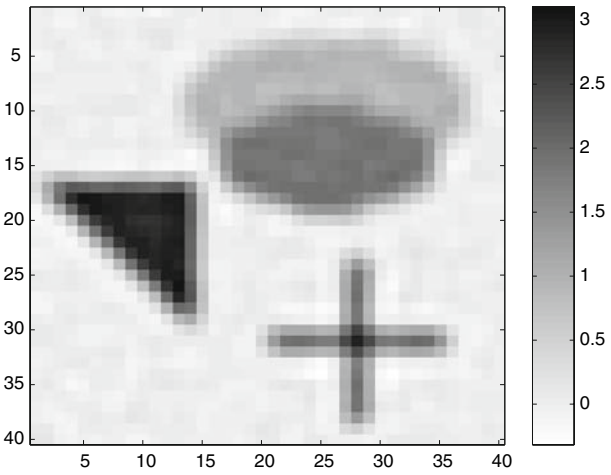

Fig. 14 Image deblurring example: point #3, $\tau = 127146$, rel. acc. = 38.07%

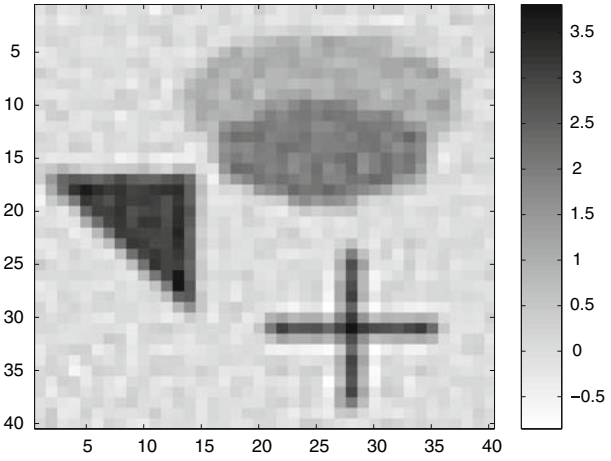

Fig. 15 Image deblurring example: point #4, $\tau = 137838$, rel. acc. = 31.82%

5.3 Comparison with CGLS

We compare our approach to the conjugate gradients based method for solving the least-squares problems CGLS. CGLS is one of the most robust regularization techniques that can handle very large problem instances. This method, described in [28] (see also [18]), applies conjugate gradients (CG) to the normal equations $G^T G x = G^T d$ along with an early termination criteria to obtain the regularized solution. The stopping condition is based on the discrepancy principle, i.e. the method terminates once the residual is smaller than some prescribed bound δ. Typically, the value of δ is based on the norm of the noise.

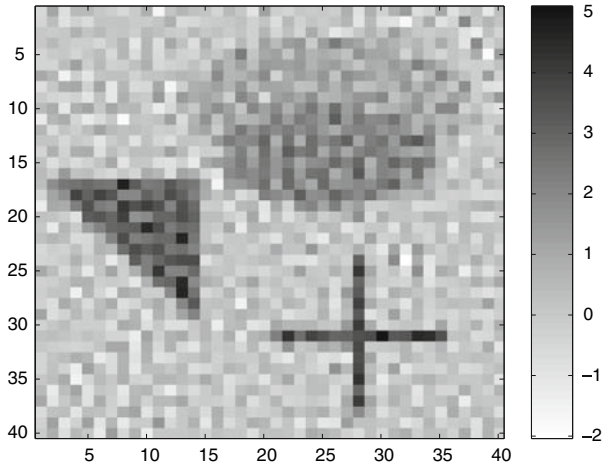

Fig. 16 Image deblurring example: point #5, $\|x\|_2 = 139212$, rel. acc. = 57.14%

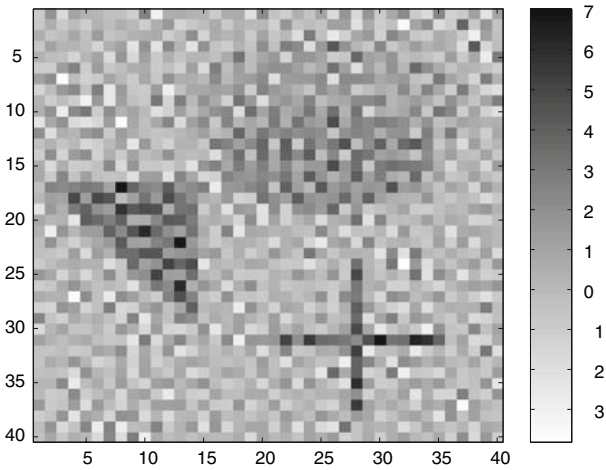

Fig. 17 Image deblurring example: point #6, $\|x\|_2 = 139345$, rel. acc. = 116.29%

We applied the CGLS algorithm on the data from the previous example supplying δ to be precisely the norm of the noise, i.e., $\delta = \|\eta\|_2$. In some sense this corresponds to the best case for CGLS. The results are presented in Table 2 and Fig. 19. The CGLS points are shown as circles above the L-curve. The CGLS solution is almost as good as the best Tikhonov solution. This result is not unusual and emphasizes the fact that the method was applied with exact knowledge of the noise. However, comparing both CGLS and RPTRS solutions to the true one (see Fig. 20), we see that both methods achieve practically the same accuracy.

The RPTRS algorithm, though, does not require a specific value of the norm of the noise. This is a big advantage in a sense that CGLS might perform very poorly

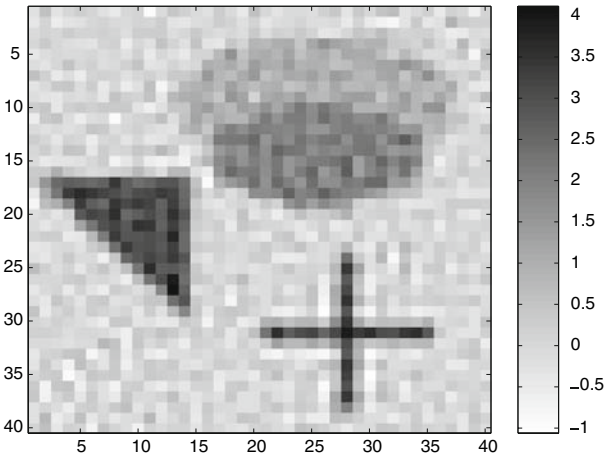

Fig. 18 Image deblurring example: RPTRS solution picture

Table 1 Data for points visited by the CGLS algorithm with $\delta = \|\eta\|_2$

##	$\|x\|_2$	$\|Gx - d\|_2$	Accuracy [%]
1	3.8162e+001	6.9804e+000	.57
2	3.9849e+001	3.9256e+000	.81
3	4.0593e+001	2.8676e+000	.99
4	4.1045e+001	2.3920e+000	.96
5	4.1406e+001	2.1105e+000	.55
6	4.1706e+001	1.9309e+000	.34

Table 2 Data for points visited by the RPTRS algorithm

##	$\|x\|_2$	$\|Gx - d\|_2$	Accuracy [%]	Time	t	λ
1	1.8573e+001	20010e+001	6539	2.794	652166	−9.8851e−001
2	2.8472e+001	11095e+001	4963	3.054	994155	−3.4166e−001
3	3.7079e+001	40222e+000	3807	3.014	127146	−7.7717e−002
4	4.1957e+001	15642e+000	3182	3.695	137838	−7.7959e−003
5	4.9732e+001	10570e+000	5714	6.509	139212	−5.3731e−004
6	6.8218e+001	7.9497e−001	11629	5.558	139345	−1.0426e−004
+1	4.2910e+001	14078e+000	3263	2.834	138490	−4.1666e−003
+2	5.3732e+001	9.7305e−001	7149	2.794	139269	−3.2078e−004
+3	4.3991e+001	12993e+000	3536	2.824	138832	−2.3520e−003

if supplied with slightly smaller (or larger) value of δ. Figure 21 illustrates this situation. Running CGLS with $\delta = 0.6\,\|\eta\|_2$ results in a larger number of iterations (31 comparing to 6 with $\delta = \|\eta\|_2$) and the computed solution is much worse now. This shows the importance of a robust stopping criteria that does not rely on the possibly uncertain data.

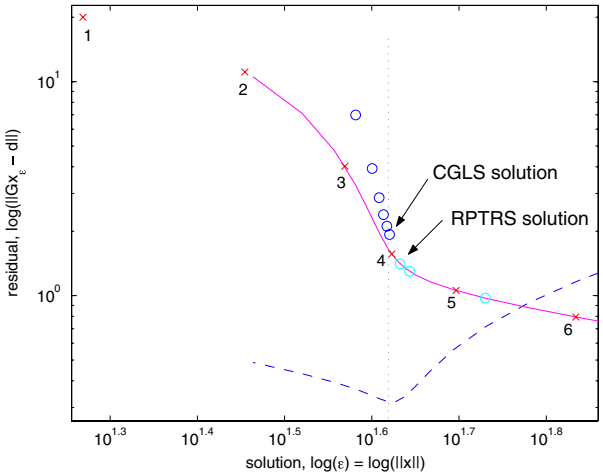

Fig. 19 Image deblurring example: corresponding L-curve with CGLS points

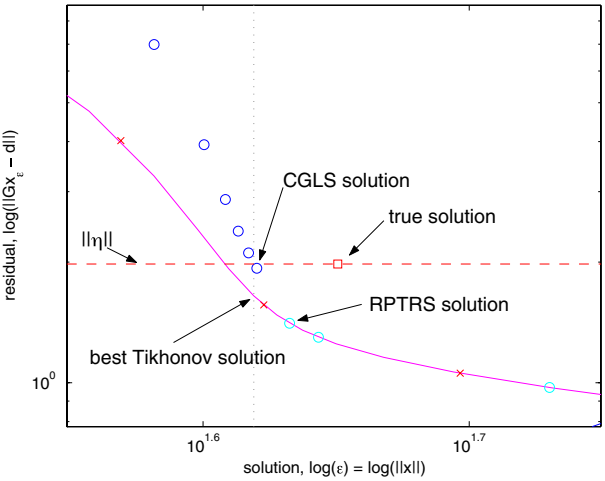

Fig. 20 Image deblurring example: CGLS, RPTRS, true, best Tikhonov solutions

The main advantage of the CGLS method is its speed. Each iteration of the algorithm requires only several matrix-vector multiplications, where only the original matrix G is used. This allows for solutions of problems that involve large sparse matrices which are never formed explicitly. At the same time, the RPTRS Algorithm can be viewed as a matrix-free iterative algorithm based on the Lanczos method that features conjugate gradients steps as well. This leads to a conclusion that combining both approaches may result in a better algorithm that can provide a reliable and a fast way to locate a regularized solution in the absence of any certain knowledge about the noise.

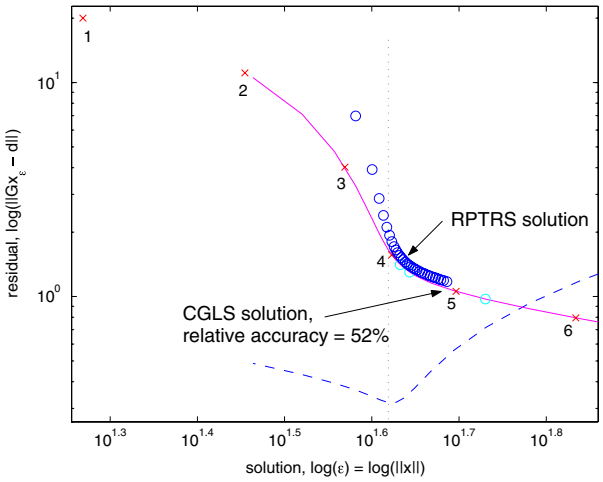

Fig. 21 Image deblurring example: CGLS with $\delta = 0.6\,\|\eta\|_2$, rel. acc. = 52%

6 Conclusion

We have applied ideas from the RW algorithm for TRS to efficiently find the point of maximum curvature on the L-curve. This provides a regularization procedure for ill-conditioned problems $Gx = d$. We have taken advantage of the fact that each iteration of the RW algorithm corresponds to a point on the L-curve. We implicitly change the trust region radius while applying the RW algorithm. The changes drive the algorithm to the *good* (*correct*) radius that corresponds to the elbow, the point of maximum curvature on the L-curve.

Acknowledgments The authors thank Professor Arkadi Nemirovski from Technion—Israel Institute of Technology, for helpful talks related to the CGLS method.

References

1. Aster, R., Borchers, B., Thurber, C.: Parameter Estimation and Inverse Problems. Academic, New York (2004)
2. Baumeister, J.: Stable solution of inverse problems. Advanced Lectures in Mathematics. Friedr. Vieweg & Sohn, Braunschweig (1987)
3. Calvetti, D., Hansen, P., Reichel, L.: L-curve curvature bounds via Lanczos bidiagonalization. Electron. Trans. Numer. Anal. **14**, 135–150 (electronic) (2002). Orthogonal polynomials, approximation theory, and harmonic analysis (Inzel, 2000)
4. Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**(3), 425–455 (1994)
5. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. J. Am. Stat. Assoc. **90**(432), 1200–1224 (1995)
6. Donoho, D.L., Johnstone, I.M., Kerkyacharian, G., Picard, D.: Wavelet shrinkage: asymptopia? J. R. Stat. Soc. Ser. B **57**, 301–337 (1995)
7. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT **17**, 134–145 (1977)

8. Fortin, C., Wolkowicz, H.: The trust region subproblem and semide nite programming. Optim. Methods Softw. **19**(1), 41–67 (2004). Special issue dedicated to Jochem Zowes 60th birthday, Guest Editors: Florian Jarre and Michal Kocvara
9. Gander, W.: On the linear least squares problem with a quadratic constraint. Technical Report, STAN-CS-78-697, Department of Computer Science, Stanford University, Stanford (1978). Habilitationsschrift ETH Zurich
10. Gay, D.: Computing optimal locally constrained steps. SIAM J. Sci. Stat. Comput. **2**, 186–197 (1981)
11. Golub, G., Van Loan, C.: Matrix computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
12. Golub, G.H., von Matt, U.: Tikhonov regularization for large scale problems. Technical Report, SCCM-97-03, Stanford University (1997)
13. Golub, G.H., von Matt, U.: Tikhonov regularization for large scale problems. In: Scienti c computing (Hong Kong, 1997), pp. 3–26. Springer, Singapore (1997)
14. Golub, G.H., von Matt, U.: Generalized cross-validation for large-scale problems. J. Comput. Graph. Stat. **6**(1), 1–34 (1997)
15. Hadamard, J.: Sur les problemes aux derivees partielles et leur signi cation physique. Princeton University Bulletin, pp. 49–52 (1902)
16. Hadamard, J.: Lectures on Cauchy's problem in linear partial differential equations. Dover Publications, New York (1953)
17. Hanke, M.: Conjugate gradient type methods for ill-posed problems. Pitman Research Notes in Mathematics Series, vol. 327. Longman Scienti c & Technical, Harlow (1995)
18. Hanke, M., Hansen, P.: Regularization methods for large-scale problems. Surv. Math. Ind. **3**(4), 253–315 (1993)
19. Hansen, P.: Analysis of discrete ill-posed problems by means of the l-curve. SIAM Rev. **34**(4), 561–580 (1992)
20. Hansen, P.: Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. Numer. Algorithms **6**(1–2), 1–35 (1994)
21. Hansen, P.: Rank-de cient and discrete ill-posed problems. SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia. Numerical aspects of linear inversion (1998)
22. Hansen, P.: The l-curve and its use in the numerical treatment of inverse problems. Technical report, Technical University of Denmark (1999). URL: http://www.imm.dtu.dk/documents/ftp/tr99/tr15_99.pdf
23. Hansen, P., O'leary, D.: The use of the l-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comput. **14**(6), 1487–1503 (1993)
24. Lancaster, P.: On eigenvalues of matrices dependent on a parameter. Numer. Math. **6**, 377–387 (1964)
25. Lancaster, P., Tismenetsky, M.: The theory of matrices, 2nd edn. Computer Science and Applied Mathematics. Academic, Orlando (1985)
26. von Matt, U.: Large constrained quadratic problems. PhD Thesis, Institute for Scienti c Computing, ETH, Zürich, Switzerland (1993)
27. Moré, J., Sorensen, D.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**, 553–572 (1983)
28. Natterer, F.: The Mathematics of Computerized Tomography. Wiley, New York (1986)
29. Nemirovskii, A.: The regularization properties of the adjoint gradient method in ill-posed problems. USSR Comput. Math. Math. Phys. **26**(2), 7–16 (1986)
30. Neumaier, A.: Solving ill-conditioned and singular linear systems: a tutorial on regularization. SIAM Rev. **40**(3), 636–666 (electronic) (1998)
31. Overton, M., Womersley, R.: Second derivatives for optimizing eigenvalues of symmetric matrices. SIAM J. Matrix Anal. Appl. **16**(3), 697–718 (1995)
32. Rendl, F., Wolkowicz, H.: A semide nite framework for trust region subproblems with applications to large scale minimization. Math. Program. **77**(2, Ser. B), 273–299 (1997)
33. Rojas, M., Santos, S., Sorensen, D.: A new matrix-free algorithm for the large-scale trust-region subproblem. SIAM J. Optim. **11**(3), 611–646 (electronic) (2000/01)
34. Rojas, M., Sorensen, D.: A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. SIAM J. Sci. Comput. **23**(6), 1842–1860 (electronic) (2002)
35. Rojas, M., Steihaug, T.: An interior-point trust-region-based method for large-scale non-negative regularization. Inverse Probl. **18**(5), 1291–1307 (2002)

36. Shaw, C. Jr.: Improvement of the resolution of an instrument by numerical solution of an integral equation. J. Math. Anal. Appl. **37**, 83–112 (1972)
37. Sorensen, D.: Minimization of a large-scale quadratic function subject to a spherical constraint. SIAM J. Optim. **7**(1), 141–161 (1997)
38. Stern, R., Wolkowicz, H.: Inde nite trust region subproblems and nonsymmetric eigenvalue perturbations. SIAM J. Optim. **5**(2), 286–313 (1995)
39. Stern, R., Ye, J.: Variational analysis of an extended eigenvalue problem. Linear Algebra Appl. **220**, 391–417 (1995)
40. Tikhonov, A.: Regularization of incorrectly posed problems. Sov. Math. **4**, 1624–1627 (1963)
41. Vanderbei, R.: The Amateur Astrophotographer. (2007, forthcoming)