Numerical Algorithms 6 (1994) 1–35

REGULARIZATION TOOLS: A Matlab package for analysis and solution of discrete ill-posed problems *

Per Christian Hansen

UNI•C (Danish Computing Center for Research and Education), Building 305, Technical University of Denmark, DK- 2800 Lyngby, Denmark

Received 7 October 1992; revised 22 March 1993
Communicated by C. Brezinski and M. Redivo Zaglia

The package REGULARIZATION TOOLS consists of 54 Matlab routines for analysis and solution of discrete ill-posed problems, i.e., systems of linear equations whose coefficient matrix has the properties that its condition number is very large, and its singular values decay gradually to zero. Such problems typically arise in connection with discretization of Fredholm integral equations of the first kind, and similar ill-posed problems. Some form of regularization is always required in order to compute a stabilized solution to discrete ill-posed problems. The purpose of REGULARIZATION TOOLS is to provide the user with easy-to-use routines, based on numerical robust and efficient algorithms, for doing experiments with regularization of discrete ill-posed problems. By means of this package, the user can experiment with different regularization strategies, compare them, and draw conclusions from these experiments that would otherwise require a major programming effort. For discrete ill-posed problems, which are indeed difficult to treat numerically, such an approach is certainly superior to a single black-box routine. This paper describes the underlying theory and gives an overview of the package; a complete manual is also available [1].

Keywords: Regularization, ill-posed problems, ill-conditioning, Matlab.

Subject classification: AMS(MOS) 65F30, 65F20.

1. Discrete ill-posed problems

The concept of ill-posed problems goes back to Hadamard in the beginning of this century, cf. e.g. [34]. Hadamard essentially defined a problem to be *ill-posed* if

* This work was supported by grants from Augustinus Fonden, Knud Højgaards Fond, and Civ. Ing. Frants Allings Legat.

[1] The package and the manual in PostScript form can be obtained by electronic mail via netlib. The e-mail address is netlib@research.att. com and the name of the library is NUMERALGO. Matlab Version 4.0 or later versions is required.

© J.C. Baltzer AG, Science Publishers

the solution is not unique or if it is not a continuous function of the data – i.e., if an arbitrary small perturbation of the data can cause an arbitrarily large perturbation of the solution. Hadamard believed that ill-posed problems were "artificial" in that they would not describe physical systems. He was wrong, however, and today there is a vast amount of literature on ill-posed problems arising in many areas of science and engineering, cf. e.g. [14–16,32,59,63,65,71,81].

The classical example of an ill-posed problem is a Fredholm integral equation of the first kind with a square integrable kernel [31],

$$\int_a^b K(s,t)f(t)\mathrm{d}t = g(s), \quad c \leqslant s \leqslant d\,, \tag{1}$$

where the right-hand side g and the kernel K are given, and where f is the unknown solution. If the solution f is perturbed by

$$\Delta f(t) = \epsilon \sin(2\pi p t), \quad p = 1, 2, \ldots, \ \epsilon = \text{constant}\,,$$

then the corresponding perturbation of the right-hand side g is given by

$$\Delta g(s) = \epsilon \int_a^b K(s,t)\sin(2\pi p t)\mathrm{d}t, \quad p = 1, 2, \ldots\,,$$

and due to the Riemann–Lebesgue lemma it follows that $\Delta g \to 0$ as $p \to \infty$ [31, p.2]. Hence, the ratio $\|\Delta f\|/\|\Delta g\|$ can become arbitrarily large by choosing the integer p large enough, thus showing that (1) is an ill-posed problem. In particular, this example illustrates that Fredholm integral equations of the first kind with square integrable kernels are extremely sensitive to high-frequency perturbations.

Strictly speaking, ill-posed problems must be infinite dimensional – otherwise the ratio $\|\Delta f\|/\|\Delta g\|$ stays bounded, although it may become very large. However, certain finite-dimensional discrete problems have properties very similar to those of ill-posed problems, such as being highly sensitive to high-frequency perturbations, and it is natural to associate the term *discrete ill-posed problems* with these problems. We can be more precise with this characterization for linear systems of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}^{n \times n}\,, \tag{2}$$

and linear least-squares problems

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2, \quad \boldsymbol{A} \in \mathbb{R}^{m \times n}, \ m > n\,. \tag{3}$$

We say that these are discrete ill-posed problems if both of the following criteria are satisfied:

(1) the singular values of $\boldsymbol{A}$ decay gradually to zero;
(2) the ratio between the largest and the smallest nonzero singular values is large.

Singular values are discussed in detail in section 3. Criterion 2 implies that the matrix $\boldsymbol{A}$ is ill- conditioned, i.e., that the solution is potentially very sensitive to per-

turbations; criterion 1 implies that there is no "nearby" problem with a well-conditioned coefficient matrix and with well-determined numerical rank.

The typical manifestations of discrete ill-posed problems are systems of linear equations and linear least-squares problems arising from discretization of ill-posed problems. E.g., if a Galerkin-type method [3] is used to discretize the Fredholm integral equation (1), then a problem of the form (2) or (3) arises – depending on the type of collocation method used – with the elements a_{ij} and b_i of the matrix A and the right-hand side $\boldsymbol{b}$ given by

$$a_{ij} = \int_a^b \int_c^d K(s,t)\phi_i(s)\psi_j(t)\,\mathrm{d}s\,\mathrm{d}t, \quad b_i = \int_c^d \phi_i(s)g(s)\,\mathrm{d}s, \tag{4}$$

where ϕ_i and ψ_j are the particular basis functions used in the Galerkin method. For such problems, the close relationship between the ill-posedness of the integral equation and the ill-conditioning of the matrix A are well understood [1,39,80]. In particular, it can be shown that the singular values of A decay in such a way that both criteria 1 and 2 above are satisfied.

An interesting and important aspect of discrete ill-posed problems is that the ill-conditioning of the problem does not mean that a meaningful approximate solution cannot be computed. Rather, the ill-conditioning implies that standard methods in numerical linear algebra [9,29] for solving (2) and (3), such as LU, Cholesky, or QR factorization, cannot be used in a straightforward manner to compute such a solution. Instead, more sophisticated methods must be applied in order to ensure the computation of a meaningful solution. This is the essential goal of regularization methods.

The package REGULARIZATION TOOLS provides a collection of easy-to-use Matlab routines for the numerical treatment of discrete ill-posed problems. The philosophy behind REGULARIZATION TOOLS is modularity and regularity between the routines. Many routines require the SVD of the coefficient matrix A – this is not necessarily the best approach in a given application, but it is certainly well suited for Matlab [56] and for this package.

The numerical treatment of integral equations in general is treated in standard references such as [4,5,13,17,18], and surveys of regularization theory can be found in, e.g., [7,10,31,32,46,51,53,67,75,81].

The package was developed in the period 1990–1993 and to some extent it reflects the author's own work. Dianne P. O'Leary and Martin Hanke helped with the iterative methods and Tommy Elfving helped with maximum entropy regularization. Lars Eldén – and his 1979 SIMULA package [21] with the same purpose as REGULARIZATION TOOLS – provided a great source of inspiration. The package was also inspired by a paper by Natterer [60] where a "numerical analyst's toolkit for ill-posed problems" is suggested. The Fortran programs by Drake [19], te Riele [68], and Wahba and her co-workers [6] also deserve to be mentioned here.

2. Regularization methods

The primary difficulty with the discrete ill-posed problems (2) and (3) is that they are essentially underdetermined due to the cluster of small singular values of A. Hence, it is necessary to incorporate further information about the desired solution in order to stabilize the problem and to single out a useful and stable solution. This is the purpose of *regularization.*

Although many types of additional information about the solution $\boldsymbol{x}$ are possible in principle, the dominating approach to regularization of discrete ill-posed problems is to require that the 2-norm – or an appropriate seminorm – of the solution be small. An initial estimate $\boldsymbol{x}^*$ of the solution may also be included in the side constraint. Hence, the side constraint involves minimization of the quantity

$$\Omega(\boldsymbol{x}) = \|L(\boldsymbol{x} - \boldsymbol{x}^*)\|_2 \,. \tag{5}$$

Here, the matrix L is typically either the identity matrix I_n or a $p \times n$ discrete approximation of the $(n-p)$th derivative operator, in which case L is a banded matrix with full row rank. In some cases it is more appropriate that the side constraint be a Sobolev norm of the form

$$\Omega(\boldsymbol{x})^2 = \alpha_0^2 \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + \sum_{i=1}^{q} \alpha_i^2 \|L_i(\boldsymbol{x} - \boldsymbol{x}^*)\|_2^2 \,,$$

where L_i approximates the ith derivative operator. Notice that this Ω can always be written in the form (5) by setting L equal to the Cholesky factor of the matrix $\alpha_0^2 I_n + \sum_{i=1}^{q} \alpha_i^2 L_i^{\mathrm{T}} L_i$. By means of the side constraint Ω one can therefore control the smoothness of the regularized solution.

When the constraint $\Omega(\boldsymbol{x})$ is introduced, one must give up the requirement that $A\boldsymbol{x}$ equals $\boldsymbol{b}$ in the linear system (2) and instead seek a solution that provides a fair balance between minimizing $\Omega(\boldsymbol{x})$ and minimizing the residual norm $\|A\boldsymbol{x} - \boldsymbol{b}\|_2$. The underlying idea is that a regularized solution with small (semi)norm and a suitably small residual norm is not too far from the desired, unknown solution to the unperturbed problem underlying the given problem. The same idea of course also applies to the least squares problem (3).

Undoubtedly, the most common and well-known form of regularization is the one known as *Tikhonov regularization* [64,69,70]. Here, the idea is to define the regularized solution $\boldsymbol{x}_\lambda$ as the minimizer of the following weighted combination of the residual norm and the side constraint

$$\boldsymbol{x}_\lambda = \arg\min \left\{ \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda^2 \|L(\boldsymbol{x} - \boldsymbol{x}^*)\|_2^2 \right\} , \tag{6}$$

where the *regularization parameter* λ controls the weight given to minimization of the side constraint relative to minimization of the residual norm. Clearly, a large λ (equivalent to a large amount of regularization) favors a small solution seminorm at the cost of a large residual norm, while a small λ (i.e., a small amount of regular-

ization) has the opposite effect. As we shall see in eq. (14), λ also controls the sensitivity of the regularized solution $\boldsymbol{x}_\lambda$ to perturbations in A and $\boldsymbol{b}$, and the perturbation bound is proportional to λ^{-1}. Thus, the regularization parameter λ is an important quantity which controls the properties of the regularized solution, and λ should therefore by chosen with care. In section 9 we return to numerical methods for actually computing λ.

We remark that an underlying assumption for the use of Tikhonov regularization in the form of eq. (6) is that the errors in the right-hand side are unbiased and that their covariance matrix is proportional to the identity matrix. If the latter condition is not satisfied one should incorporate the additional information and rescale the problem or use a regularized version of the general Gauss–Markov linear model:

$$\min\left\{ \|\boldsymbol{u}\|_2^2 + \lambda^2 \|L\boldsymbol{x}\|_2^2 \right\} \quad \text{subject to} \quad A\boldsymbol{x} + C\boldsymbol{u} = \boldsymbol{b}, \tag{7}$$

where C is the Cholesky factor of the covariance matrix. The latter approach using (7) must be used if the covariance matrix is rank deficient, i.e., if C is not a square matrix. For a discussion of this approach and a numerical algorithm for solving (7), cf. [82].

Besides Tikhonov regularization, there are many other regularization methods with properties that make them better suited to certain problems or certain computers. We return to these methods in sections 7 and 8, but first it is convenient to introduce some important numerical "tools" for analysis of discrete ill-posed problems in sections 3–5. As we shall demonstrate, getting insight into the discrete ill-posed problem is often at least as important as computing a solution, because the regularized solution should be computed with such care. Finally, in section 9 we shall describe some methods for choosing the regularization parameter.

3. SVD and generalized SVD

The superior numerical "tools" for analysis of discrete ill-posed problems are the *singular value decomposition* (SVD) of A and its generalization to two matrices, the *generalized single value decomposition* (GSVD) of the matrix pair (A, L) [29, §2.5.3 and §8.7.3]. The SVD reveals all the difficulties associated with the ill-conditioning of the matrix A while the GSVD of (A, L) yields important insight into the regularization problem involving both the coefficient matrix A and the regularization matrix L, such as in (6).

3.1. THE SINGULAR VALUE DECOMPOSITION

Let $A \in \mathbb{R}^{m \times n}$ be a rectangular matrix with $m \geq n$. Then the SVD of A is a decomposition of the form

$$A = U\Sigma V^{\mathrm{T}} = \sum_{i=1}^{n} u_i \sigma_i v_i^{\mathrm{T}}, \tag{8}$$

where $U = (u_1, \ldots, u_n)$ and $V = (v_1, \ldots, v_n)$ are matrices with orthonormal columns, $U^{\mathrm{T}}U = V^{\mathrm{T}}V = I_n$, and where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ has non-negative diagonal elements appearing in non-increasing order such that

$$\sigma_1 \geqslant \ldots \geqslant \sigma_n \geqslant 0. \tag{9}$$

The numbers σ_i are the *singular values* of A while the vectors u_i and v_i are the left and right singular vectors of A, respectively. The condition number of A is equal to the ratio σ_1/σ_n.

From the relations $A^{\mathrm{T}}A = V\Sigma^2 V^{\mathrm{T}}$ and $AA^{\mathrm{T}} = U\Sigma^2 U^{\mathrm{T}}$ we see that the SVD of A is strongly linked to the eigenvalue decompositions of the symmetric positive semi-definite matrices $A^{\mathrm{T}}A$ and AA^{T}. This shows that the SVD is unique for a given matrix A – except for singular vectors associated with multiple singular values. In connection with discrete ill-posed problems, two characteristic features of the SVD of A are very often found.

- The singular values σ_i decay gradually to zero with no particular gap in the spectrum. An increase of the dimensions of A will increase the number of small singular values.
- The left and right singular vectors u_i and v_i tend to have more sign changes in their elements as the index i increases, i.e., as σ_i decreases.

Although these features are found in many discrete ill-posed problems arising in practical applications, they are unfortunately very difficult – or perhaps impossible – to prove in general.

To see how the SVD gives insight into the ill-conditioning of A, consider the following relations which follows directly from eq. (8):

$$\left.\begin{aligned} Av_i &= \sigma_i u_i \\ \|Av_i\|_2 &= \sigma_i \end{aligned}\right\} \quad i = 1, \ldots, n. \tag{10}$$

We see that a small singular value σ_i, compared to $\|A\|_2 = \sigma_1$, means that there exists a certain linear combination of the columns of A, characterized by the elements of the right singular vector v_i, such that $\|Av_i\|_2 = \sigma_i$ is small. In other words, one or more small σ_i implies that A is nearly rank deficient, and the vectors v_i associated with the small σ_i are numerical null-vectors of A. From this and the characteristic features of A we conclude that the matrix in a discrete ill-posed problem is always highly ill-conditioned, and its numerical null-space is spanned by vectors with many sign changes.

The SVD also gives important insight into another aspect of discrete ill-posed problems, namely the smoothing effect typically associated with a square integrable kernel. Notice that as σ_i decreases, the singular vectors u_i and v_i become more and more oscillatory. Consider now the mapping Ax of an arbitrary vector x. Using the SVD, we get $x = \sum_{i=1}^{n} (v_i^{\mathrm{T}}x)v_i$ and

$$Ax = \sum_{i=1}^{n} \sigma_i (v_i^T x) u_i .$$

This clearly shows that due to the multiplication with the σ_i the high-frequency components of x are more damped in Ax than the low-frequency components. Moreover, the inverse problem, namely that of computing x from $Ax = b$ or $\min \|Ax - b\|_2$, must have the opposite effect: it amplifies the high-frequency oscillations in the right-hand side b.

3.2. THE GENERALIZED SINGULAR VALUE DECOMPOSITION

The GSVD of the matrix pair (A, L) is a generalization of the SVD of A in the sense that the generalized singular values of (A, L) are the square roots of the generalized eigenvalues of the matrix pair $(A^T A, L^T L)$. In order to keep our exposition simple, we assume that the dimensions of $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{p \times n}$ satisfy $m \geqslant n \geqslant p$, which is always the case in connection with discrete ill-posed problems. Then the GSVD is a decomposition of A and L in the form

$$A = U \begin{pmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{pmatrix} X^{-1}, \quad L = V(M, 0) X^{-1}, \tag{11}$$

where the columns of $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{p \times p}$ are orthonormal, $X \in \mathbb{R}^{n \times n}$ is nonsingular, and Σ and M are $p \times p$ diagonal matrices: $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, $M = \mathrm{diag}(\mu_1, \ldots, \mu_p)$. Moreover, the diagonal entries of Σ and M are non-negative and ordered such that

$$0 \leqslant \sigma_1 \leqslant \ldots \leqslant \sigma_p \leqslant 1, \quad 1 \geqslant \mu_1 \geqslant \ldots \geqslant \mu_p > 0 ,$$

and they are normalized such that

$$\sigma_i^2 + \mu_i^2 = 1, \quad i = 1, \ldots, p .$$

Then the *generalized singular values* γ_i of (A, L) are defined as the ratios

$$\gamma_i = \sigma_i / \mu_i, \quad i = 1, \ldots, p , \tag{12}$$

and they obviously appear in non-decreasing order. For historical reasons, this ordering is the opposite of the ordering of the ordinary singular values of A.

For $p < n$ the matrix $L \in \mathbb{R}^{p \times n}$ always has a nontrivial null-space $\mathcal{N}(L)$. E.g., if L is an approximation to the second derivative operator on a regular mesh, $L = \mathrm{tridiag}(1, -2, 1)$, then $\mathcal{N}(L)$ is spanned by the two vectors $(1, 1, \ldots, 1)^T$ and $(1, 2, \ldots, n)^T$. In the GSVD, the last $n - p$ columns x_i of the nonsingular matrix X satisfy

$$L x_i = 0, \quad i = p + 1, \ldots, n , \tag{13}$$

and they are therefore basis vectors for the null-space $\mathcal{N}(L)$.

There is a slight notational problem here because the matrices U, Σ, and V in the GSVD of (A, L) are *different* from the matrices with the same symbols in the SVD of A. However, in this presentation it will always be clear from the context which decomposition is used. When L is the identity matrix I_n, then the U and V of the GSVD are identical to the U and V of the SVD, and the generalized singular values of (A, I_n) are identical to the singular values of A – except for the ordering of the singular values and vectors.

In general, there is no connection between the generalized singular values/vectors and the ordinary singular values/vectors. For discrete ill-posed problems, though, we can actually say something about the SVD–GSVD connection because L is typically a reasonably well-conditioned matrix. When this is the case, then it can be shown that the matrix X in (11) is also well-conditioned. Hence, the diagonal matrix Σ must display the ill-conditioning of A, and since $\gamma_i = \sigma_i(1 - \sigma_i^2)^{1/2} \approx \sigma_i$ for small σ_i the generalized singular values must decay gradually to zero as the ordinary singular values do. Moreover, the oscillation properties (i.e., the increase in sign changes) of the right singular vectors carries over to the columns of X in the GSVD: the smaller the γ_i the more sign changes in $\boldsymbol{x}_i$. For more specific results, cf. [41].

As an immediate example of the use of GSVD in the analysis of discrete regularization problems, we mention the following perturbation bound for Tikhonov regularization derived in [40]. Let E and $\boldsymbol{e}$ denote the perturbations of A and $\boldsymbol{b}$, respectively, and let $\bar{\boldsymbol{x}}_\lambda$ denote the exact solution to the unperturbed problem; then the relative error in the perturbed solution $\boldsymbol{x}_\lambda$ satisfies

$$\frac{\|\boldsymbol{x}_\lambda - \bar{\boldsymbol{x}}_\lambda\|_2}{\|\bar{\boldsymbol{x}}_\lambda\|_2} \leqslant \frac{\|A\|_2\|X\|_2\lambda^{-1}}{1 - \|E\|_2\|X\|_2\lambda^{-1}} \times \left((1 + \operatorname{cond}(X)) \frac{\|E\|_2}{\|A\|_2} + \frac{\|\boldsymbol{e}\|_2}{\|\boldsymbol{b}_\lambda\|_2} + \|E\|_2\|X\|_2\lambda^{-1} \frac{\|\boldsymbol{r}_\lambda\|_2}{\|\boldsymbol{b}_\lambda\|_2} \right), \quad (14)$$

where we have defined $\boldsymbol{b}_\lambda = A\boldsymbol{x}_\lambda$ and $\boldsymbol{r}_\lambda = \boldsymbol{b} - \boldsymbol{b}_\lambda$. The important conclusion we can make from this relation is that for all reasonable λ the perturbation bound for the regularized solution $\boldsymbol{x}_\lambda$ is proportional to λ^{-1} and to the norm of the matrix X. The latter quantity is analyzed in [41] where it is shown that $\|X\|_2$ is approximately bounded by $\|L^\dagger\|_2$, i.e., by the inverse of the smallest singular value of L. Hence, in addition to controlling the smoothness of the regularized solution, λ and L also control its sensitivity to perturbations of A and $\boldsymbol{b}$.

The SVD and the GSVD are computed by means of routines `csvd` and `gsvd` in this package. The routine `bsvd` computes the SVD of a bidiagonal matrix.

4. The discrete Picard condition and filter factors

As we have seen in section 3, the integration in eq. (1) with a square integrable kernel K (1) has a smoothing effect on f. The opposite operation, namely, that of

solving the first kind Fredholm integral equation for f, therefore tends to amplify oscillations in the right-hand side g. Hence, if we require that the solution f be a square integrable solution with finite L_2-norm, then not all functions are valid as right-hand side g. Indeed, g must be sufficiently smooth to "survive" the inversion back to f. The mathematical formulation of this smoothness criterion on g – once the kernel K is given – is called the Picard condition [31, §1.2].

For discrete ill-posed problems there is, strictly speaking, no Picard condition because the norm of the solution is always bounded. Nevertheless, it makes sense to introduce a discrete Picard condition as follows. In a real-world application, the right-hand side $\boldsymbol{b}$ is always contaminated by various types of errors, such as measurement errors, approximation errors, and rounding errors. Hence, $\boldsymbol{b}$ can be written as

$$\boldsymbol{b} = \bar{\boldsymbol{b}} + \boldsymbol{e}, \tag{15}$$

where $\boldsymbol{e}$ are the errors, and $\bar{\boldsymbol{b}}$ is the unperturbed right-hand side. Both $\bar{\boldsymbol{b}}$ and the corresponding unperturbed solution $\bar{\boldsymbol{x}}$ represent the underlying unperturbed and unknown problem. Now, if we want to be able to compute a regularized solution $\boldsymbol{x}_{\mathrm{reg}}$ from the given right-hand side $\boldsymbol{b}$ such that $\boldsymbol{x}_{\mathrm{reg}}$ approximates the exact solution $\bar{\boldsymbol{x}}$, then it is shown in [44] that the corresponding exact right-hand side $\bar{\boldsymbol{b}}$ must satisfy a criterion very similar to the Picard condition:

THE DISCRETE PICARD CONDITION

The unperturbed right-hand side $\bar{\boldsymbol{b}}$ in a discrete ill-posed problem with regularization matrix L satisfies the discrete Picard condition if the Fourier coefficients $|\boldsymbol{u}_i^{\mathrm{T}}\bar{\boldsymbol{b}}|$ on the average decay to zero faster than the generalized singular values γ_i.

The discrete Picard condition is not as "artificial" as it may seem at first: it can be shown that if the underlying integral equation (1) satisfies the Picard condition, then the discrete ill-posed problem obtained by discretization of the integral equation satisfies the discrete Picard condition [39]. See also [75,76].

The main difficulty with discrete ill-posed problems is caused by the errors $\boldsymbol{e}$ in the given right-hand side $\boldsymbol{b}$ (15), because such errors typically tend to have components along all the left singular vectors $\boldsymbol{u}_i$. For example, if $\|\boldsymbol{e}\|_2 = \epsilon$ and if the elements of $\boldsymbol{e}$ are unbiased and uncorrelated, then the expected value of the Fourier coefficients of $\boldsymbol{e}$ satisfy

$$\mathcal{E}\left(|\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{e}|\right) = m^{-1/2}\epsilon, \quad i = 1, \ldots, n. \tag{16}$$

As a consequence, the Fourier coefficients $|\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}|$ of the perturbed right-hand side level off at approximately $m^{-1/2}\epsilon$ even if the unperturbed right-hand side $\bar{\boldsymbol{b}}$ satisfies the discrete Picard condition, because these Fourier coefficients are dominated by $|\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{e}|$ for large i.

Consider now the linear system (2) and the least squares problem (3), and assume for simplicity that A has no exact zero singular values. Using the SVD, it is easy to show that the solutions to both systems are given by the same equation:

$$x_{\mathrm{LSQ}} = \sum_{i=1}^{n} \frac{u_i^{\mathrm{T}} b}{\sigma_i} v_i \,. \tag{17}$$

This relation clearly illustrates the difficulties with the standard solution to (2) and (3). Since the Fourier coefficients $|u_i^{\mathrm{T}} b|$ corresponding to the smaller singular values σ_i do not decay as fast as the singular values – but rather tend to level off – the solution x_{LSQ} is dominated by the terms in the sum corresponding to the smallest σ_i. As a consequence, the solution x_{LSQ} has many sign changes and thus appears completely random.

With this analysis in mind, we can see that the purpose of a regularization method is to dampen or filter out the contributions to the solution corresponding to the small generalized singular values. Hence, we will require that a regularization method produces a regularized solution x_{reg} which, for $x^* = 0$, can be written as follows

$$x_{\mathrm{reg}} = \sum_{i=1}^{n} f_i \frac{u_i^{\mathrm{T}} b}{\sigma_i} v_i \qquad \text{if } L = I_n \,, \tag{18}$$

$$x_{\mathrm{reg}} = \sum_{i=1}^{p} f_i \frac{u_i^{\mathrm{T}} b}{\sigma_i} x_i + \sum_{i=p+1}^{n} (u_i^{\mathrm{T}} b) x_i \quad \text{if } L \neq I_n \,. \tag{19}$$

Here, the numbers f_i are *filter factors* for the particular regularization method. The filter factors must have the important property that as σ_i decreases, the corresponding f_i tends to zero in such a way that the contributions $(u_i^{\mathrm{T}} b/\sigma_i) x_i$ to the solution from the smaller σ_i are effectively filtered out. The difference between the various regularization methods lies essentially in the way that these filter factors f_i are defined. Hence, the filter factors play an important role in connection with regularization theory, and it is worthwhile to characterize the filter factors for the various regularization methods that we shall present below.

For Tikhonov regularization, which plays a central role in regularization theory, the filter factors are either $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ (for $L = I_n$) or $f_i = \gamma_i^2/(\gamma_i^2 + \lambda^2)$ (for $L \neq I_n$), and the filtering effectively sets in for $\sigma_i < \lambda$ and $\gamma_i < \lambda$, respectively. In particular, this shows that discrete ill-posed problems are essentially unregularized by Tikhonov's method for $\lambda < \sigma_n$ and $\lambda < \gamma_p$, respectively.

Filter factors for various regularization methods can be computed by means of routine `fil_fac` in this package, while routine `picard` plots the important quantities σ_i, $|u_i^{\mathrm{T}} b|$, and $|u_i^{\mathrm{T}} b|/\sigma_i$ if $L = I_n$, or γ_i, $|u_i^{\mathrm{T}} b|$, and $|u_i^{\mathrm{T}} b|/\gamma_i$ if $L \neq I_n$.

5. The L-curve

Perhaps the most convenient graphical tool for analysis of discrete ill-posed problems is the so-called *L-curve* which is a plot – for all valid regularization parameters – of the (semi)norm $\|Lx_{\mathrm{reg}}\|_2$ of the regularized solution versus the corresponding residual norm $\|Ax_{\mathrm{reg}} - b\|_2$. In this way, the L-curve clearly displays the compromise between minimization of these two quantities, which is the heart of any regularization method. The use of such plots in connection with ill-conditioned least squares problems goes back to Miller [57] and Lawson and Hanson [54].

For discrete ill-posed problems it turns out that the L-curve, when plotted in *log–log scale*, almost always has a characteristic L-shaped appearance (hence its name) with a distinct corner separating the vertical and the horizontal parts of the curve. To see why this is so, we notice that if $\bar{x}$ denotes the exact, unregularized solution corresponding to the exact right-hand side $\bar{b}$ in eq. (15), then the error $x_{\mathrm{reg}} - \bar{x}$ in the regularized solution consists of two components, namely, a perturbation error from the error e in the given right-hand side b, and a regularization error due to the regularization of the error-free component $\bar{b}$ in the right-hand side. The vertical part of the L-curve corresponds to solutions where $\|Lx_{\mathrm{reg}}\|_2$ is very sensitive to changes in the regularization parameter because the perturbation error e dominates x_{reg} and because e does not satisfy the discrete Picard condition. The horizontal part of the L-curve corresponds to solutions where it is the residual norm $\|Ax_{\mathrm{reg}} - b\|_2$ that is most sensitive to the regularization parameter because x_{reg} is dominated by the regularization error – as long as $\bar{b}$ satisfies the discrete Picard condition.

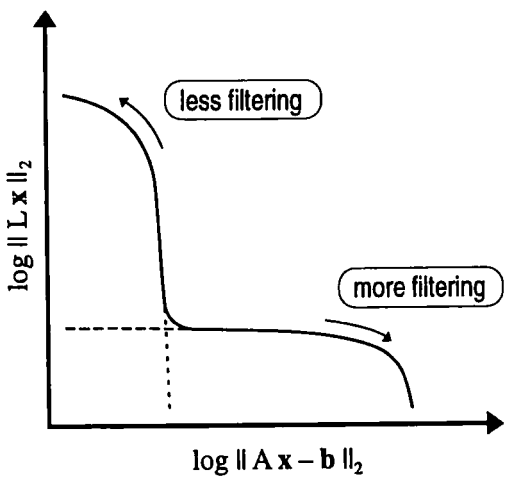

Fig. 1. The generic form of the L-curve.

We can substantiate this by means of the relations for the regularized solution x_{reg} in terms of the filter factors. For general-form regularization ($L \neq I_n$) eq. (19) yields the following expression for the error in x_{reg}:

$$x_{\mathrm{reg}} - \bar{x} = \left(\sum_{i=1}^{p} f_i \frac{u_i^{\mathrm{T}} e}{\sigma_i} x_i + \sum_{i=p+1}^{n} (u_i^{\mathrm{T}} e) x_i \right) + \sum_{i=1}^{p} (f_i - 1) \frac{u_i^{\mathrm{T}} \bar{b}}{\sigma_i} x_i . \qquad (20)$$

Here, the term in parentheses is the *perturbation error* due to the perturbation e, and the second term is the *regularization error* caused by regularization of the unperturbed component $\bar{b}$ of the right-hand side. When only little regularization is introduced, most of the filter factors f_i are approximately one and the error $x_{\mathrm{reg}} - \bar{x}$ is dominated by the perturbation error. On the other hand, with plenty of regularization most filter factors are small, $f_i \ll 1$, and $x_{\mathrm{reg}} - \bar{x}$ is dominated by the regularization error.

In [45,48], eq. (20) was used to analyze the relationship between the error in x_{reg} and the behavior of the L-curve. The result is that if $\bar{b}$ satisfies the discrete Picard condition, then the horizontal part of the curve corresponds to solutions where the regularization error dominates – i.e., where so much filtering is introduced that the solution stays very smooth and $\|Lx_{\mathrm{reg}}\|_2$ therefore only changes a little with the regularization parameter. In contrast, the vertical part of the L-curve corresponds to solutions that are dominated by the perturbation error, and due to the division by the small σ_i it is clear that $\|Lx_{\mathrm{reg}}\|_2$ varies dramatically with the regularization parameter while, simultaneously, the residual norm does not change much. Moreover, it is shown in [48] that a log–log scale emphasizes the different appearances of the vertical and the horizontal parts. In this way, the L-curve clearly displays the trade-off between minimizing the residual norm and the side constraint.

We note in passing that the L-curve is a continuous curve when the regularization parameter is continuous as in Tikhonov regularization. For regularization methods with a discrete regularization parameter, such as truncated SVD, we plot the L-curve as a finite set of points. How to make such a "discrete L-curve" continuous is discussed in [48]. In this reference, alternative norms for plotting the L-curve, depending on the regularization method, are also discussed.

The L-curve for Tikhonov regularization plays a central role in connection with regularization methods for discrete ill-posed problems because it divides the first quadrant into two regions. It is impossible to construct any solution that corresponds to a point below the Tikhonov L-curve; any regularized solution must lie on or above this curve. The solution computed by Tikhonov regularization is therefore optimal in the sense that for a given residual norm there does not exist a solution with smaller seminorm than the Tikhonov solution – and the same is true with the roles of the norms interchanged. A consequence of this is that one can compare other regularization methods with Tikhonov regularization by inspecting how close the L-curve for the alternative method is to the Tikhonov L-curve. If $\bar{b}$ satis-

fies the discrete Picard condition, then the two L-curves are close to each other and the solutions computed by the two regularization methods are similar [45].

For a given fixed right-hand side $\boldsymbol{b} = \bar{\boldsymbol{b}} + \boldsymbol{e}$, there is obviously an optimal regularization parameter that balances the perturbation error and the regularization error in $\boldsymbol{x}_{\text{reg}}$. An essential feature of the L-curve is that this optimal regularization parameter – defined in the above sense – is not far from the regularization parameter that corresponds to the L-curve's corner [45]. In other words, by locating the corner of the L-curve one can compute an approximation to the optimal regularization parameter and thus, in turn, compute a regularized solution with a good balance between the two error types. We return to this aspect in section 9; suffice it here to say that for continuous L-curves, a computationally convenient definition of the L-curve's corner is the point with maximum curvature in log–log scale.

In the REGULARIZATION TOOLS package, routine `l_curve` produces a log–log plot of the L-curve and – if required – also locates the corner and identifies the corresponding regularization parameter. Given a discrete set of values of $\|A\boldsymbol{x}_{\text{reg}} - \boldsymbol{b}\|_2$ and $\|L\boldsymbol{x}_{\text{reg}}\|_2$, routine `plot_lc` plots the corresponding L-curve, while routine `l_corner` locates the L-curve's corner.

6. Transformation to standard form

A regularization problem with side constraint $\Omega(\boldsymbol{x}) = \|L(\boldsymbol{x} - \boldsymbol{x}^*)\|_2$ (5) is said to be in *standard form* if the matrix L is the identity matrix I_n. In many applications, regularization in standard form is not the best choice, i.e., one should use some $L \neq I_n$ in the side constraint $\Omega(\boldsymbol{x})$. The proper choice of matrix L depends on the particular application, but often an approximation to the first or second derivative operator gives good results.

However, from a numerical point of view it is much simpler to treat problems in standard form, basically because only one matrix, A, is involved instead of the two matrices A and L. Hence, it is convenient to be able to transform a given regularization problem in general form into an equivalent one in standard form by means of numerically stable methods. For example, for Tikhonov regularization we want a numerically stable method for transforming the general-form problem (6) into the following standard-form problem

$$\min\left\{ \|\bar{A}\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}\|_2^2 + \lambda^2 \|\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^*\|_2^2 \right\}, \tag{21}$$

where the new matrix $\bar{A}$, the new right-hand side $\bar{\boldsymbol{b}}$, and the vector $\bar{\boldsymbol{x}}^*$ are derived from the original quantities, $A, L, \boldsymbol{b}$, and $\boldsymbol{x}^*$. Moreover, we also want a numerically stable scheme for transforming the solution $\bar{\boldsymbol{x}}_\lambda$ to (21) back to the general-form setting. Finally, we prefer a transformation that leads to a simple relationship between the SVD of $\bar{A}$ and the GSVD of (A, L), for then we have a perfect understanding of the relationship between the two regularization problems.

For the simple case where L is square and invertible, the transformation is obvious: $\bar{A} = AL^{-1}$, $\bar{b} = b$, $\bar{x}^* = Lx^*$, and the back-transformation simply becomes $x_\lambda = L^{-1}\bar{x}_\lambda$.

In most applications, however, the matrix L is not square, and the transformation becomes somewhat more involved than just a matrix inversion. It turns out that it is a good idea to distinguish between direct and iterative regularization methods – cf. the next two sections. For the direct methods we need to be able to compute the matrix $\bar{A}$ explicitly by standard methods such as the QR factorization. For the iterative methods, on the other hand, we merely need to be able to compute the matrix-vector product $\bar{A}\bar{x}$ efficiently. Below, we describe two methods for transformation to standard form which are suited for direct and iterative methods, respectively. We assume that the matrix $L \in \mathbb{R}^{p \times n}$ has full row rank, i.e., the rank of L is p.

6.1. TRANSFORMATION FOR DIRECT METHODS

The standard-form transformation for direct methods described here was developed by Eldén [22], and it is based on two QR factorizations. The description of this transformation is quite algorithmic, and it is summarized below (where, for convenience, the subscripts p, o, and q denote matrices with p, $n-p$, and $m-(n-p)$ columns, respectively). First, compute a QR factorization of L^T,

$$L^T = KR = (K_p, K_o)\begin{pmatrix} R_p \\ 0 \end{pmatrix}. \tag{22}$$

We remark that since L has full rank, its pseudoinverse is simply $L^\dagger = K_p R_p^{-T}$. Moreover, the columns of K_o are an orthonormal basis for the null space of L. Next, form the "skinny" matrix $AK_o \in \mathbb{R}^{m \times (n-p)}$ and compute its QR factorization,

$$AK_o = HT = (H_o, H_q)\begin{pmatrix} T_o \\ 0 \end{pmatrix}. \tag{23}$$

Then the transformed quantities $\bar{A}$ and $\bar{b}$ are given by the following identities

$$\bar{A} = H_q^T A L^\dagger = H_q^T A K_p R_p^{-T}, \quad \bar{b} = H_q^T b, \tag{24}$$

and we stress that the most efficient way to compute $\bar{A}$ and $\bar{b}$ is to apply the orthogonal transformations that make up K and H "on the fly" to A and b as the QR factorizations in (22) and (23) are computed. When (21) has been solved for $\bar{x}$, the transformation back to the general-form setting then takes the form

$$x = L^\dagger \bar{x} + K_o T_o^{-1} H_o^T (b - AL^\dagger \bar{x}). \tag{25}$$

The SVD of the matrix $\bar{A}$ is related to the GSVD of (A, L) as follows: let $\bar{A} = \bar{U}\bar{\Sigma}\bar{V}^T$ denote the SVD of $\bar{A}$, and let E_p denote the $p \times p$ exchange matrix

$E_p = \text{antidiag}(1, \ldots, 1)$. Also, let U, V, Σ, M, and X denote the GSVD matrices from eq. (11). Then

$$U = (H_q \bar{U} E_p, H_o), \quad \Sigma M^{-1} = E_p \bar{\Sigma} E_p, \tag{26}$$

$$V = \bar{V} E_p, \qquad X = \begin{pmatrix} M^{-1} V^{\mathrm{T}} L \\ H_o^{\mathrm{T}} A \end{pmatrix}^{-1}. \tag{27}$$

Moreover, the last $n-p$ columns of X are given by $K_o T_o^{-1}$. For proofs of (26)–(27) and an investigation of the accuracy of the GSVD matrices computed this way, cf. [41,43].

6.2. TRANSFORMATION FOR ITERATIVE METHODS

For the iterative methods the matrix $\bar{A}$ is never computed explicitly. Instead, one merely needs to be able to pre-multiply a vector with $\bar{A}$ and $\bar{A}^{\mathrm{T}}$ efficiently. If K_o is an orthonormal basis for the null space of L, e.g. computed by (22), then $y = L^{\dagger}\bar{x}$ and $\hat{y} = (L^{\dagger})^{\mathrm{T}} x$, both of which are used in the iterative procedures, can easily be computed by the following algorithms:

$$y \leftarrow \begin{pmatrix} I_{n-p} & 0 \\ L & \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \bar{x} \end{pmatrix}, \qquad x \leftarrow x - K_o K_o^{\mathrm{T}} x,$$

$$y \leftarrow y - K_o K_o^{\mathrm{T}} y, \qquad \begin{pmatrix} \hat{y} \\ z \end{pmatrix} \leftarrow \begin{pmatrix} L & \\ 0 & I_{n-p} \end{pmatrix}^{-\mathrm{T}} x. \tag{28}$$

By means of these simple algorithms, which are described in [8], the above standard-transformation method can also be used for iterative methods. Notice that a basis for the null space of L is required – often, the basis vectors can be computed explicitly, or they can be computed from L by, say, a rank revealing factorization [12].

The algorithm from section 6.1 can be reformulated in such a way that the pseudoinverse $L^{\dagger}$ is replaced by a weaker generalized inverse, using an idea from [23] (and later advocated in [35,37]). This reformulation has certain advantages for iterative methods, as we shall see in section 8.4. Define the *A-weighted generalized inverse of* L as follows

$$L_A^{\dagger} = X \begin{pmatrix} M^{-1} \\ 0 \end{pmatrix} V^{\mathrm{T}}, \tag{29}$$

where we emphasize that $L_A^{\dagger}$ is generally different from the pseudoinverse $L^{\dagger}$ when $p < n$. Also, define the vector

$$x_0 = \sum_{i=p+1}^{n} (u_i^{\mathrm{T}} b) x_i, \tag{30}$$

which is the unregularized component of the regularized solution x_{reg}, cf. eq. (19), i.e., x_0 is the component of x_{reg} that lies in the null space of L. Then the standard-form quantities $\bar{A}$ and $\bar{b}$ in the alternative version of the algorithm are defined as follows

$$\bar{A} = AL_A^{\dagger}, \quad \bar{b} = b - Ax_0 . \tag{31}$$

Moreover, the transformation back to the general-form setting takes the simple form

$$x = L_A^{\dagger}\bar{x} + x_0 . \tag{32}$$

This backtransformation is mathematically equivalent to the one in (25) since we have

$$L_A^{\dagger} = (I_n - K_o T_o^{-1} H_o^{\mathrm{T}} A)L^{\dagger} \quad \text{and} \quad x_0 = K_o T_o^{-1} H_o^{\mathrm{T}} b . \tag{33}$$

To use this alternative formulation of the standard-form transformation, we need to compute x_0 as well as the matrix-vector products $L_A^{\dagger}\bar{x}$ and $(L_A^{\dagger})^{\mathrm{T}}x$ efficiently. Given a basis W for $\mathcal{N}(L)$, the vector x_0 is computed by the following algorithm:

$$\begin{aligned} &T \leftarrow (A\ W)^{\dagger}, \\ &x_0 \leftarrow WTb . \end{aligned} \tag{34}$$

This algorithm involves $O(mn(n-p))$ operations. To compute $L_A^{\dagger}\bar{x}$ and $(L_A^{\dagger})^{\mathrm{T}}x$ efficiently (we emphasize that $L_A^{\dagger}$ is never computed explicitly), we need the matrix T from the above algorithm, and then $y = L_A^{\dagger}\bar{x}$ and $\hat{y} = (L_A^{\dagger})^{\mathrm{T}}x$ are computed by:

$$\begin{aligned} &y \leftarrow \begin{pmatrix} I_{n-p} & 0 \\ & L \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \bar{x} \end{pmatrix}, \qquad x \leftarrow x - (WTA)^{\mathrm{T}}x, \\ &y \leftarrow y - WTAy, \qquad \begin{pmatrix} \hat{y} \\ z \end{pmatrix} \leftarrow \begin{pmatrix} L & \\ 0 & I_{n-p} \end{pmatrix}^{-\mathrm{T}} x, \end{aligned} \tag{35}$$

where, for efficiency, the matrix product $TA \in \mathbb{R}^{(n-p)\times n}$ can be explicitly computed and saved. In the above formulas, W need not have orthonormal columns, although this is the best choice from a numerical point of view. For more details about these algorithms, cf. [37, section 4.3].

For the latter standard-form transformation there is an even simpler relation between the SVD of $\bar{A}$ and part of the GSVD of (A, L) than in section 6.1 because $AL_A^{\dagger} = \sum_{i=i}^{p} u_i \gamma_i v_i^{\mathrm{T}}$. I.e., except for the ordering, the GSVD quantities u_i, γ_i, and v_i are identical to the similar SVD quantities, and with the same notation as in eq. (26) we have

$$(u_1, \ldots, u_p) = \bar{U}E_p, \quad V = \bar{V}E_p, \quad \Sigma M^{-1} = E_p \bar{\Sigma} E_p . \tag{36}$$

This relation is very important in connection with the iterative regularization methods.

6.3. NORM RELATIONS ETC.

From the above relations (26) and (36) between the SVD of $\bar{A}$ and the GSVD of (A, L) we obtain the following very important relations between the norms related to the two regularization problems

$$\|L(\boldsymbol{x} - \boldsymbol{x}^*)\|_2 = \|\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^*\|_2, \quad \|A\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\bar{A}\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}\|_2, \tag{37}$$

where $\boldsymbol{x}$ denotes the solution obtained by transforming $\bar{\boldsymbol{x}}$ back to the general-form setting. These relations are very important in connection with methods for choosing the regularization parameter because they show that any parameter-choice strategy based on these norms will yield the *same* regularization parameter when applied to the general-form problem and the transformed standard-form problem.

Several routines are included in REGULARIZATION TOOLS for computations related to the standard-form transformations. First of all, the routines `stf_form` and `gen_form` perform both transformations to standard from and back to general form. These routines are mainly included for pedagogical reasons. Routine `pinit` computes the vector $\boldsymbol{x}_0$ in eq. (30) as well as the matrix TA by means of algorithm (34). Finally, the two routines `lsolve` and `ltsolve` compute $L_A^\dagger \bar{\boldsymbol{x}}$ and $(L_A^\dagger)^T \boldsymbol{x}$ by the algorithms in (35).

Regarding the matrix L, discrete approximations to derivative operators on a regular mesh can be computed by routine `get_l` which also provides a matrix W with orthonormal basis vectors for the null space of L.

7. Direct regularization methods

In this and the next section we shall briefly review the regularization methods for numerical treatment of discrete ill-posed problems included in the REGULARIZATION TOOLS package. Our aim is not to compare these and other methods, because that is outside the scope of this paper. In fact, very little has been done in this area, cf. [2,37,42,55]. This section focuses on methods which are essentially direct, i.e., methods where the solution is defined by a direct computation (which may still involve an iterative root-finding procedure, say), while regularization methods which are intrinsically iterative are treated in the next section.

7.1. TIKHONOV REGULARIZATION

Tikhonov's method is of course a direct method because the regularized solution $\boldsymbol{x}_\lambda$, defined by eq. (6), is the solution to the following least squares problem

$$\min \left\| \begin{pmatrix} A \\ \lambda L \end{pmatrix} x - \begin{pmatrix} b \\ \lambda L x^* \end{pmatrix} \right\|_2 , \tag{38}$$

and it is easy to see that x_λ is unique if the null spaces of A and L intersect trivially (as they usually do in practice). The most efficient algorithm for numerical treatment of Tikhonov's method for a general regularization matrix L consists of three steps [22]. First, the problem is transformed into standard form by means of eqs. (22)–(24) from section 6.1 ($\bar{A} = A$ if $L = I_n$), then the matrix $\bar{A}$ is transformed into a $p \times p$ upper bidiagonal matrix $\bar{B}$ by means of left and right orthogonal transformations,

$$\bar{A} = \bar{U} \bar{B} \bar{V}^T ,$$

and finally the resulting sparse problem with a banded $\bar{B}$ is solved for $\bar{V}^T \bar{x}_\lambda$ and the solution is transformed back to the original setting by means of (25).

In this package we take another approach to solving (38), namely, by using the filter factors and the GSVD explicitly (or the SVD, if $L = I_n$), cf. eqs. (18) and (19). This approach, which is implemented in routine `tikhonov`, is more suited to Matlab's coarse granularity. For pedagogical reasons, we also include a routine `bidiag` for bidiagonalization of a matrix.

7.2. LEAST SQUARES WITH A QUADRATIC CONSTRAINT

There are two other regularization methods which are almost equivalent to Tikhonov's method, and which can be treated numerically by essentially the same technique as mentioned above involving a transformation to standard form followed by bidiagonalization of the coefficient matrix. These two methods are the following least squares problems with a quadratic inequality constraint

$$\min \|Ax - b\|_2 \quad \text{subject to } \|L(x - x^*)\|_2 \leqslant \alpha , \tag{39}$$

$$\min \|L(x - x^*)\|_2 \quad \text{subject to } \|Ax - b\|_2 \leqslant \delta , \tag{40}$$

where α and δ are nonzero parameters each playing the role of regularization parameter in (39) and (40), respectively. The solution to both these problems is identical to x_λ from Tikhonov's method for suitably chosen values of λ that depend in a nonlinear way on α and δ. The solution to the first problem (39) is computed as follows: if $\|L(x_{\mathrm{LSQ}} - x_0)\|_2 \leqslant \alpha$ then $\lambda \leftarrow 0$ and $x_\lambda \leftarrow x_{\mathrm{LSQ}}$, else use an iterative scheme to solve

$$\min \|Ax_\lambda - b\|_2 \quad \text{subject to } \|L(x_\lambda - x^*)\|_2 = \alpha ,$$

for λ and x_λ. Similarly, the solution to the second problem (40) is computed as follows (where x_0 is given by eq. (30)): if $\|Ax_0 - b\|_2 \leqslant \delta$ then $\lambda \leftarrow \infty$ and $x_\lambda \leftarrow x_0$, else use an iterative scheme to solve

$$\min \|L(x_\lambda - x^*)\|_2 \quad \text{subject to } \|Ax_\lambda - b\|_2 = \delta ,$$

for λ and x_λ. In REGULARIZATION TOOLS, routines `lsqi` and `discrep` solve (39) and (40), respectively. The name "discrep" is related to the discrepancy principle for choosing the regularization parameter, cf. section 9. An efficient algorithm for solving (39) when A is large and sparse, based on Gauss quadrature and Lanczos bidiagonalization, is described in [30].

7.3. TSVD, MTSVD, AND TGSVD

A fundamental observation regarding the above-mentioned methods is that they circumvent the ill-conditioning of A by introducing a new problem (38) with a well-conditioned coefficient matrix $(A^T, \lambda L^T)^T$ with full rank. A different way to treat the ill-conditioning of A is to derive a new problem with a well-conditioned *rank deficient* coefficient matrix. A fundamental result about rank deficient matrices, which can be derived from the SVD of A, is that the closest rank-k approximation A_k to A – measured in the 2-norm – is obtained by truncating the SVD expansion in (8) at k, i.e., A_k is given by

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T, \quad k \leqslant n. \tag{41}$$

The truncated SVD (TSVD) [74,38,42] and the modified TSVD (MTSVD) [50] regularization methods are based on this observation in that one solves the problems

$$\min \|x\|_2 \quad \text{subject to} \quad \min \|A_k x - b\|_2, \tag{42}$$

$$\min \|Lx\|_2 \quad \text{subject to} \quad \min \|A_k x - b\|_2, \tag{43}$$

where A_k is the rank-k matrix in eq. (41). The solutions to these two problems are given by

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i \tag{44}$$

$$x_{L,k} = x_k - V_k (V_k L)^\dagger L x_k, \tag{45}$$

where $(V_k L)^\dagger$ is the pseudoinverse of $V_k L$, and $V_k \equiv (v_{k+1}, \ldots, v_n)$. In other words, the correction to x_k in (45) is the solution to the following least squares problem

$$\min \|(V_k L) z - L x_k\|_2 .$$

We note in passing that the TSVD solution x_k is the only regularized solution which has no component in the numerical null-space of A, spanned by the columns of V_k. All other regularized solutions, exemplified by the MTSVD solution $x_{L,k}$, have some component in A's numerical null-space in order to achieve the desired properties of the solution, as controlled by the matrix L.

As an alternative to the above-mentioned MTSVD method for general-form problems one can generalize the TSVD method to the GSVD setting [41,44]. The resulting method, truncated GSVD (TGSVD), is easiest to introduce via the standard-form transformation from section 6.2 with $\bar{A} = AL_A^{\dagger}$, $\bar{b} = b - Ax_0$, and $x = L_A^{\dagger}\bar{x} + x_0 \Rightarrow Lx = \bar{x}$. In analogy with the TSVD method we now introduce a rank-k approximation $\bar{A}_k$ to $\bar{A}$ via its SVD. Due to the SVD–GSVD relations between $\bar{A}$ and (A, L), computation of the matrix $\bar{A}_k$ is essentially a "truncated GSVD" because $\bar{A}_k = \sum_{i=p-k+1}^{p} u_i \gamma_i v_i^{\mathrm{T}}$. Then we define the truncated GSVD (TGSVD) solution as $\hat{x}_{L,k} = L_A^{\dagger}\bar{x}_k + x_0$, where $\bar{x}_k$ solves the problem

$$\min \|\bar{x}\|_2 \quad \text{subject to} \quad \min \|\bar{A}_k\bar{x} - \bar{b}\|_2 . \tag{46}$$

Definition (46) together with the GSVD of (A, L) then immediately lead to the following simple expression of the TGSVD solution

$$\hat{x}_{k,L} = \sum_{i=p-k+1}^{p} \frac{u_i^{\mathrm{T}} b}{\sigma_i} x_i + \sum_{i=p+1}^{n} (u_i^{\mathrm{T}} b) x_i , \tag{47}$$

where the last term is the component x_0 (30) in the null space of L. Defined this way, the TGSVD solution is a natural generalization of the TSVD solution x_k. The TGSVD method is also a generalization of TSVD because both x_k and $\hat{x}_{k,L}$ can be derived from the corresponding Tikhonov solutions (18) and (19) by substituting 0's and 1's for the Tikhonov filter factors f_i.

The TSVD, MTSVD, and TGSVD solutions are computed by the routines with the obvious names `tsvd`, `mtsvd`, and `tgsvd`.

7.4. DAMPED SVD/GSVD

A less know regularization method which is based on the SVD or the GSVD is the damped SVD/GSVD (damped SVD was introduced in [20], and our generalization to damped GSVD is obvious). Here, instead of using filter factors 0 and 1 as in TSVD and TGSVD, one introduces a smoother cut-off by means of filter factors f_i defined as

$$f_i = \frac{\sigma_i}{\sigma_i + \lambda} \quad (\text{for } L = I_n) \quad \text{and} \quad f_i = \frac{\sigma_i}{\sigma_i + \lambda\mu_i} \quad (\text{for } L \neq I_n) . \tag{48}$$

These filter factors decay slower than the Tikhonov filter factors and thus, in a sense, introduce less filtering. The damped SVD/GSVD solutions are computed by means of routine `dsvd`.

7.5. MAXIMUM ENTROPY REGULARIZATION

This regularization method is frequently used in image reconstruction and related applications where a solution with positive elements is sought. In maximum entropy regularization, the following nonlinear function is used as side constraint:

$$\Omega(\boldsymbol{x}) = \sum_{i=1}^{n} x_i \log(w_i x_i)\,, \tag{49}$$

where x_i are the positive elements of the vector $\boldsymbol{x}$, and $w_1, \ldots, w_n$ are n weights. Notice that $-\Omega(\boldsymbol{x})$ measures the entropy of $\boldsymbol{x}$, hence the name of this regularization method. The mathematical justification for this particular choice of $\Omega(\boldsymbol{x})$ is that it yields a solution $\boldsymbol{x}$ which is most objective, or maximally uncommitted, with respect to missing information in the right-hand side, cf. e.g. [66].

Maximum entropy regularization is implemented in REGULARIZATION TOOLS in the routine `maxent` which uses a nonlinear conjugate gradient algorithm [28, §4.1] with inexact line search to compute the regularized solution. The typical step in this method has the form

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &\leftarrow \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}\,, \\ \boldsymbol{p}^{(k+1)} &\leftarrow -\nabla F(\boldsymbol{x}^{(k+1)}) + \beta_k \boldsymbol{p}^{(k)}\,, \end{aligned} \tag{50}$$

in which F is the function to be minimized,

$$F(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda^2 \sum_{i=1}^{n} x_i \log(w_i x_i)\,,$$

and F's gradient is given by

$$\nabla F(\boldsymbol{x}) = 2A^{\mathrm{T}}(A\boldsymbol{x} - \boldsymbol{b}) + \lambda^2 \begin{pmatrix} 1 + \log(w_1 x_1) \\ \vdots \\ 1 + \log(w_n x_n) \end{pmatrix}.$$

In algorithm (50), the step-length parameter α_k minimizes $F(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)})$ with the constraint that all elements of $\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$ be positive, and it is computed by means of an inexact line search. Then β_k is computed by

$$\beta_k = (\nabla F(\boldsymbol{x}^{(k+1)}) - \nabla F(\boldsymbol{x}^{(k)}))^{\mathrm{T}} \nabla F(\boldsymbol{x}^{(k+1)}) / \|\nabla F(\boldsymbol{x}^{(k+1)})\|_2^2\,.$$

This choice of β_k has the potential advantage that it gives "automatic" restart to the steepest descent direction in case of slow convergence.

7.6. TRUNCATED TOTAL LEAST SQUARES

The last direct regularization method included in REGULARIZATION TOOLS is truncated total least squares (TTLS). For rank deficient matrices, total least squares [26] takes its basis in an SVD of the compound matrix $(A, \boldsymbol{b}) = \tilde{U}\tilde{\Sigma}\tilde{V}^{\mathrm{T}}$ with the matrix $\tilde{V} \in \mathbb{R}^{(n+1)\times(n+1)}$ partitioned such that

$$\tilde{V} = \begin{pmatrix} \tilde{V}_{11} & \tilde{V}_{12} \\ \tilde{V}_{21} & \tilde{V}_{22} \end{pmatrix}, \quad \tilde{V}_{11} \in \mathbb{R}^{n\times k}, \quad \tilde{V}_{22} \in \mathbb{R}^{1\times(n+1-k)}\,, \tag{51}$$

where k is the numerical rank of A. Then the TLS solution is given by

$$\bar{x}_k = -\tilde{V}_{12}\tilde{V}_{22}^{\dagger} = -\tilde{V}_{12}\tilde{V}_{22}^{\mathrm{T}}/\|\tilde{V}_{22}\|_2^2 . \tag{52}$$

The TLS solution is robust to the perturbations of A because inaccuracies in A are explicitly taken into account in the TLS method. Therefore, for discrete ill-posed problems with no gap in the singular value spectrum of A, it makes sense to define a truncated TLS solution by means of (52) where k then plays the role of the regularization parameter; see [27] for more details. The truncated TLS solution is computed by means of routine TTLS.

8. Iterative regularization methods

This section describes the iterative regularization methods included in REGULARIZATION TOOLS. We stress that our Matlab routines should be considered as model implementations; real implementations should incorporate any sparsity and/or structure of the matrix A. We shall first describe standard-form version of the methods and then describe the extension necessary for treating general-form problems. For more details about these and other iterative methods, cf. [37, chapters 6–7].

8.1. CONJUGATE GRADIENTS AND LSQR

The conjugate gradient (CG) algorithm is a well-known method for solving sparse systems of equations with a symmetric positive definite coefficient matrix. In connection with discrete ill-posed problems, it is an interesting fact that when the CG algorithm is applied to the unregularized normal equations $A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$ (implemented such that $A^{\mathrm{T}}A$ is not formed) then the low-frequency components of the solution tend to converge faster than the high-frequency components. Hence, the CG process has some inherent regularization effect where the number of iterations plays the role of the regularization parameter. The kth step of the CG process essentially has the form

$$\begin{aligned}
\beta_k &\leftarrow \|q^{(k-1)}\|_2^2/\|q^{(k-2)}\|_2^2 , \\
p^{(k)} &\leftarrow q^{(k-1)} + \beta_k p^{(k-1)} , \\
\alpha_k &\leftarrow \|q^{(k-1)}\|_2^2/\|Ap^{(k)}\|_2^2 , \\
x^{(k)} &\leftarrow x^{(k-1)} + \alpha_k p^{(k)} , \\
q^{(k)} &\leftarrow q^{(k-1)} - \alpha_k A^{\mathrm{T}}Ap^{(k)} ,
\end{aligned} \tag{53}$$

where $x^{(k)}$ is the approximation to x after k iterations, while $p^{(k)}$ and $q^{(k)}$ are two auxiliary iteration vectors of length n.

To explain this regularizing effect of the CG method, we introduce the Krylov subspace

$$\mathcal{K}_k(A^TA, A^Tb) = \text{span}\{A^Tb, A^TAA^Tb, \ldots, (A^TA)^{k-1}A^Tb\}$$

associated with the kth step of the CG algorithm applied to $A^TAx = A^Tb$. It is also convenient to introduce the Ritz polynomial P_k associated with step k:

$$P_k(\sigma) = \prod_{j=1}^{k} \frac{(\theta_j^{(k)})^2 - \sigma^2}{(\theta_j^{(k)})^2}. \tag{54}$$

Here, $(\theta_j^{(k)})^2$ are the Ritz values, i.e., the k eigenvalues of A^TA restricted to the Krylov subspace $\mathcal{K}_k(A^TA, A^Tb)$. The large Ritz values are approximations to some of the large eigenvalues σ_i^2 of the cross-product matrix A^TA. Then the filter factors associated with the solution after k steps of the CG algorithm are given by

$$f_i^{(k)} = 1 - P_k(\sigma_i), \quad i = 1, \ldots, k. \tag{55}$$

As k increases, and the Ritz values converge to some of the eigenvalues of A^TA, then for selected i and j we have $\theta_j^{(k)} \approx \sigma_i$. Moreover, as k increases these approximations improve while, simultaneously, more eigenvalues of A^TA are being approximated by the additional Ritz values.

Equations (54) and (55) for the CG filter factors shed light on the regularizing property of the CG method. After k iterations, if all the largest Ritz values $(\theta_j^{(k)})^2$ have converged to all the largest eigenvalues σ_i^2 of A^TA, then the corresponding $P_k(\sigma_i) \approx 0$ and the filter factors associated with these σ_i will therefore be close to one. On the other hand, for all those eigenvalues smaller than the smallest Ritz value, the corresponding filter factors satisfy

$$f_i^{(k)} = \sigma_i^2 \sum_{j=1}^{k} (\theta_j^{(k)})^{-2} + O\left(\sigma_i^4/(\theta_k^{(k)})^4\right),$$

showing that these filter factors decay like σ_i^2 for $\sigma_i < \theta_k(k)$.

From this analysis of the CG filter factors we see that the CG process indeed has a regularizing effect if the Ritz values converge to the eigenvalues of A^TA in their natural order, starting with the largest. When this is the case, we are sure that the CG algorithm is a regularizing process with the number of iterations k as the regularization parameter. Unfortunately, proving that the Ritz values actually converge in this order is a difficult task. For problems with a gap in the singular value spectrum of A it is proved in [49] that all the large eigenvalues of A^TA will be approximated by Ritz values before any of the small eigenvalues of A^TA get approximated. The case where the singular values of A decay gradually to zero with no gap in the spectrum is more difficult to analyze – but numerical examples and model problems [72,73] indicate that the desired convergence of the Ritz values actually holds as long as the discrete Picard condition is satisfied for the unperturbed component of the right-hand side and there is a good separation among the large singular values of A.

To put the CG method into the common framework of the previous section, we notice that the solution $x^{(k)}$ after k CG steps can be defined as

$$\min \|Ax - b\|_2 \quad \text{subject to } x \in \mathcal{K}_k(A^T A, A^T b)\,, \tag{56}$$

where $\mathcal{K}_k(A^T A, A^T b)$ is the Krylov subspace associated with the normal equations. Thus, we see that CG replaces the side constraint $\Omega(x) = \|x\|_2$ with the side constraint $x \in \mathcal{K}_k(A^T A, A^T b)$. Obviously, if the Ritz values converge as desired, then the Krylov subspace satisfies $\mathcal{K}_k(A^T A, A^T b) \approx \text{span}\{v_1, \ldots, v_k\}$ indicating that the CG solution $x^{(k)}$ is similar to, say, the TSVD solution x_k.

Even the best implementation of the normal-equation CG algorithm suffers from some loss of accuracy due to the implicit use of the cross-product matrix $A^T A$. An alternative iterative algorithm that avoids $A^T A$ completely is the algorithm LSQR [62]. This algorithm uses the Lanczos bidiagonalization algorithm [29, §9.3.4] to build up a lower bidiagonal matrix and, simultaneously, updates a QR factorization of this bidiagonal matrix. The QR factorization, in turn, is used to update the LSQR solution in each step. If B_k denotes the $(k+1) \times k$ bidiagonalization matrix generated in the kth step of LSQR, then the quantities $\theta_j^{(k)}$ in eq. (54) are the singular values of this B_k. Hence, the LSQR algorithm is mathematically equivalent to the normal-equation CG algorithm in that the kth iteration vectors $x^{(k)}$ in CG and LSQR are identical in exact arithmetic.

In real computations, the convergence of CG and LSQR is delayed due to the influence of the finite precision arithmetic, and the dimension of the subspace in which $x^{(k)}$ lies does not increase in each step. As a consequence, $x^{(k)}$ typically stays almost unchanged for a few steps, then changes to a new vector and stays unchanged again for some steps, etc. (The underlying phenomenon is related to CG and LSQR computing "ghost" eigenvalues and singular values, respectively, cf. [29, §9.2].) In LSQR, it is possible to store the so-called Lanczos vectors generated during the process and apply some reorthogonalization scheme to them, which prevents the above-mentioned delay – but in practice it is usually less computationally demanding just to use LSQR without any reorthogonalization.

There are several ways to implement the CG algorithm for the normal equations in a numerically stable fashion. The one used in the routine `cgls` in REGULARIZATION TOOLS is from [9, p. 560]. The implementation `lsqr` is identical to the original LSQR algorithm from [62].

Regarding the filter factors for CG and LSQR, we have found that the expression (55) using the Ritz polynomial is extremely sensitive to rounding errors. Instead, we compute the filter factors $f_i^{(k)}$ by means of numerically more robust recursions derived in [78] and [49]. Notice that the exact singular values of σ_i of A are required to compute the filter factors; hence this option is mainly of pedagogical interest.

8.2. BIDIAGONALIZATION WITH REGULARIZATION

It is possible to modify the LSQR algorithm and derive a hybrid between a direct and an iterative regularization algorithm. The idea is to use the above-mentioned Lanczos algorithm to build up the bidiagonal matrix B_k sequentially, and in each step to replace LSQR's QR factorization of B_k with a direct regularization scheme such as Tikhonov regularization or TSVD. These ideas are outlined in [8,61]; see also the discussion in [37, chapter 7]. The work involved in the direct regularization process is small compared to the work in the iterative process because of the bidiagonal form of B_k. Again, reorthogonalization of the Lanczos vectors is possible but rarely used in practice.

One rationale behind this "hybrid" algorithm is that if the number k of Lanczos bidiagonalization steps is so large that B_k becomes ill-conditioned and needs regularization – because the singular values $\theta_k^{(k)}$ of B_k start to approximate some of the smaller singular values of A – then hopefully all the *large* singular values of A are approximated by singular values of B_k. When this is the case, then we are ensured that the "hybrid" method computes a proper regularized solution, provided of course that the explicit regularization in each step properly filters out the influence of the small singular values.

The second, and perhaps most important, rationale behind the "hybrid" algorithm is that it requires a different stopping criterion which is not as dependent on choosing the correct k as the previously mentioned methods. Provided again that the explicit regularization scheme in each step is successful in filtering out the influence of the small singular values, then after a certain stage k the iteration vector $x^{(k)}$ of the "hybrid" algorithm will hardly change. This is so because all the components associated with the large singular values have been captured while the components associated with the small singular values are filtered out. Thus, the stopping criterion should now be based on the relative change in $x^{(k)}$. With this stopping criterion, we see that taking too many steps in the "hybrid" algorithm will not deteriorate the iteration vector, but merely increase the computational effort.

For convenience, REGULARIZATION TOOLS provides the routine `lanc_b` for computing the lower bidiagonal matrix B_k as well as the corresponding left and right Lanczos vectors, and a routine `bsvd` for computing the SVD of B_k. The user can then combine the necessary routines to form a specific "hybrid" algorithm.

8.3. THE ν-METHOD

Both CG and LSQR converge rather fast to a regularized solution with damped high-frequency components, and if the iterations are continued then the high-frequency components very soon start to dominate the iteration vector. For some methods for choosing the regularization parameter, i.e., the number of iterations k (such as the L-curve criterion described in section 9), this is a favorable property.

However, there are other circumstances in which it is more desirable to have an iterative scheme that converges slower and thus is less sensitive to the choice of k.

This is exactly the philosophy behind the ν-method [11] which is similar to the CG method except that the coefficients α_k and β_k used to update the iteration vectors in algorithm (53) are problem independent and depend only on the iteration number k and a prespecified constant ν satisfying $0 < \nu < 1$:

$$\alpha_k = 4\frac{(k+\nu)(k+\nu+\frac{1}{2})}{(k+2\nu)(k+2\nu+\frac{1}{2})}, \quad \beta_k = \frac{(k+\nu)(k+1)(k+\frac{1}{2})}{(k+2\nu)(k+2\nu+\frac{1}{2})(k+\nu+1)}. \tag{57}$$

(It can be shown that the corresponding filter factors can be expressed in terms of Jacobi polynomials.)

A slight inconvenience with the ν-method is that is requires the problem to be scaled such that $\|A\|_2$ is slightly less than one, otherwise the method either diverges or converges too slowly. A practical way to treat this difficulty [37] is to use the Lanczos bidiagonalization algorithm to compute a good approximation $\theta_1^{(k)} = \|B_k\|_2$ to $\|A\|_2$ and then rescale A and $\boldsymbol{b}$ by $0.99/\theta_1^{(k)}$. Usually a few Lanczos steps are sufficient. This initialization process can also be used to provide the ν-method with a good initial guess, namely, the iteration vector $\boldsymbol{x}^{(k)}$ after a few LSQR steps.

The routine nu in REGULARIZATION TOOLS implements the ν-method as described in [11] with the above-mentioned rescaling. The LSQR start-vector is not used, but can be supplied by the user if desired.

8.4. EXTENSION TO GENERAL-FORM PROBLEMS

So far we have described several iterative methods for treating regularization problems in standard form; we shall now briefly describe the necessary extension to these methods for treating problems in general form. The idea presented here is originally from [35,36]. From the discussion of the standard-form transformation for iterative methods in section 6.2, we see that essentially we must apply the above standard-form iterative methods to a transformed problem with $\bar{A}$ and $\bar{\boldsymbol{b}}$. I.e., according to (56) we must compute the solution $\bar{\boldsymbol{x}}^{(k)}$ to the problem

$$\min \|\bar{A}\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}\|_2 \quad \text{subject to } \bar{\boldsymbol{x}} \in \mathcal{K}_k(\bar{A}^{\mathrm{T}}\bar{A}, \bar{A}^{\mathrm{T}}\bar{\boldsymbol{b}}). \tag{58}$$

If we use the alternative formulation from section 6.2 with $\bar{A} = AL_A^{\dagger}$ and $\bar{\boldsymbol{b}} = \boldsymbol{b} - A\boldsymbol{x}_0$, then the standard-form transformation can be "built into" the iterative scheme. In this way, we work directly with $\boldsymbol{x}^{(k)}$ and avoid the back-transformation from $\bar{\boldsymbol{x}}^{(k)}$ to $\boldsymbol{x}^{(k)}$. To derive this technique, consider the side constraint in (58) which implies that there exist constants $\xi_0, \ldots, \xi_{k-1}$ such that

$$\bar{\boldsymbol{x}}^{(k)} = \sum_{i=0}^{k-1} \xi_i (\bar{A}^{\mathrm{T}}\bar{A})^i \bar{A}^{\mathrm{T}}\bar{\boldsymbol{b}}.$$

If we insert $\bar{A} = AL_A^\dagger$ and $\bar{b} = b - Ax_0$ into this relation, we obtain

$$\bar{x}^{(k)} = \sum_{i=0}^{k-1} \xi_i \Big((L_A^\dagger)^T A^T A L_A^\dagger \Big)^i (L_A^\dagger)^T A^T (b - Ax_0) .$$

Using eqs. (31) and (30) for $L_A^\dagger$ and x_0 together with the GSVD it is straightforward to show that $(L_A^\dagger)^T A^T A x_0 = 0$. Thus, by inserting the above expression for $\bar{x}^{(k)}$ into the back-transformation $x^{(k)} = L_A^\dagger \bar{x}^{(k)} + x_0$, we obtain

$$x^{(k)} = \sum_{i=0}^{k-1} \xi_i \Big(L_A^\dagger (L_A^\dagger)^T A^T A \Big)^i L_A^\dagger (L_A^\dagger)^T A^T b + x_0 . \tag{59}$$

From this relation we see that we can consider the matrix $L_A^\dagger (L_A^\dagger)^T$ a "preconditioner" for the iterative methods, and we stress that the purpose of the "preconditioner" is not to improve the condition number of the iteration matrix but rather to ensure that the "preconditioned" iteration vector $x^{(k)}$ lies in the correct subspace and thus minimizes $\|Lx^{(k)}\|_2$. Minimization of the correct residual norm is ensured by eq. (37).

"Preconditioning" is easy to build into CG, LSQR, and the ν-method by means of the algorithms in (35) from section 6.2. The special "preconditioned" versions are implemented as routines `pcgls`, `plsqr`, and `pnu` in REGULARIZATION TOOLS.

9. Methods for choosing the regularization parameter

No regularization package is complete without routines for computation of the regularization parameter. As we have already discussed in section 5, a good regularization parameter should yield a fair balance between the perturbation error and the regularization error in the regularized solution. Throughout the years a variety of parameter-choice strategies have been proposed. These methods can roughly be divided into two classes depending on their assumption about $\|e\|_2$, the norm of the perturbation of the right-hand side. The two classes can be characterized as follows:

(1) Methods based on knowledge, or a good estimate, of $\|e\|_2$.
(2) Methods that do not require $\|e\|_2$, but instead seek to extract the necessary information from the given right-hand side.

For many of these methods, the convergence rate for the solution as $\|e\|_2 \to 0$ has been analyzed [25,33,77]. Four parameter-choice routines are included in REGULARIZATION TOOLS, one from class 1 and three from class 2.

The only method belonging to class 1 is the *discrepancy principle* [58, §27] which, in all simplicity, amounts to choosing the regularization parameter such that the residual norm for the regularized solution satisfies

$$\|Ax_{\text{reg}} - b\|_2 = \|e\|_2 . \tag{60}$$

When a good estimate for $\|e\|_2$ is known, this method yields a good regularization parameter corresponding to a regularized solution immediately to the right of the L-curve's corner. Due to the steep part of the L-curve we see that an underestimate of $\|e\|_2$ is likely to produce an underregularized solution with a very large (semi)norm. On the other hand, an overestimate of $\|e\|_2$ produces an overregularized solution with too large regularization error.

The three methods from class 2 that we have included in REGULARIZATION TOOLS are the L-curve criterion, generalized cross-validation, and the quasi-optimality criterion. The *L-curve criterion* has already been discussed in connection with the introduction of the L-curve in section 5. Our implementation follows the description in [48] closely. For a continuous regularization parameter λ we compute the curvature of the curve

$$(\log \|Ax_\lambda - b\|_2, \log \|Lx_\lambda\|_2)$$

(with λ as its parameter) and seek the point with maximum curvature, which we then define as the L-curve's corner. When the regularization parameter is discrete, we approximate the discrete L-curve in log–log scale by a 2D spline curve, compute the point on the spline curve with maximum curvature, and define the corner of the discrete L-curve as that point which is closest to the corner of the spline curve.

Generalized cross-validation (GCV) is based on the philosophy that if an arbitrary element b_i of the right-hand side b is left out, then the corresponding regularized solution should predict this observation well, and the choice of regularization parameter should be independent of an orthogonal transformation of b; cf. [79, chapter 4] for more details. This leads to choosing the regularization parameter which minimizes the GCV function

$$G \equiv \frac{\|Ax_{\text{reg}} - b\|_2^2}{(\text{trace}(I_m - AA^I))^2} , \tag{61}$$

where A^I is a matrix which produces the regularized solution x_{reg} when multiplied with b, i.e., $x_{\text{reg}} = A^I b$. Note that G is defined for both continuous and discrete regularization parameters. The denominator in (61) can be computed in $O(n)$ operations if the bidiagonalization algorithm from section 7 is used [24]. Alternatively, the filter factors can be used to evaluate the denominator by means of the simple expression

$$\text{trace}(I_m - AA^I) = m - (n - p) - \sum_{i=1}^{p} f_i . \tag{62}$$

This is the approach used in routine `gcv`. In [45] it is illustrated that the GCV method indeed seeks to balance the perturbation and regularization errors and thus, in turn, is related to the corner of the L-curve.

The final method included in REGULARIZATION TOOLS is the *quasi-optimality criterion* [58, §27]. This method is, strictly speaking, only defined for a continuous regularization parameter λ and amounts to minimizing the function

$$Q \equiv \lambda \left\| \frac{\mathrm{d}\boldsymbol{x}_\lambda}{\mathrm{d}\lambda} \right\|_2 = \left(\sum_{i=1}^{p} \left(f_i(1-f_i) \frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}}{\gamma_i} \right)^2 \right)^{1/2}, \tag{63}$$

evaluated at the (generalized) singular values only [52]. As demonstrated in [45], under certain assumptions the approach also corresponds to finding a good balance between perturbation and regularization errors in $\boldsymbol{x}_\lambda$. For a discrete regularization parameter k, we use $\lambda = \gamma_k$ and the approximations

$$\left\| \frac{\mathrm{d}\boldsymbol{x}_\lambda}{\mathrm{d}\lambda} \right\|_2 \approx \frac{\|\Delta \boldsymbol{x}_k\|_2}{|\Delta\lambda|}, \quad \|\Delta \boldsymbol{x}_k\|_2 = \frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}}{\gamma_k}, \quad \Delta\lambda = \gamma_{k+1} - \gamma_k \approx \gamma_k$$

to obtain the expressions

$$Q \approx \frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}}{\sigma_k} \text{ if } L = I_n, \quad Q \approx \frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}}{\gamma_k} \text{ if } L \neq I_n. \tag{64}$$

The discrepancy principle is implemented in routine `discrep` described in section 7.2 in connection with direct regularization methods. The L-curve criterion is implemented in the two routines `l_curve` and `l_corner`, while GCV is provided by routine `gcv`. Finally, the quasi-optimally criterion is implemented in routine `quasiopt`.

10. Overview of REGULARIZATION TOOLS

We conclude this paper with a brief overview of the contents of REGULARIZATION TOOLS. The following six tables, grouped by subject area, list all the 54 Matlab routines in the package together with a short description of their purpose. More details about all the routines, as well as a tutorial introduction to the package, can be found in the manual [47].

The package is completely self-contained and needs no extra Matlab routines *except* for the following single exception: for discrete L-curves the routine `l_corner` requires ten routines from the Spline Toolbox, namely, `fnder`, `ppbrk`, `ppcut`, `ppmak`, `ppual`, `sp2pp`, `sorted`, `spbrk`, `spmak` and `sprpp`. If the Spline Toolbox is not available and `l_corner` is handling a discrete L-curve, then the routine returns with an error message. If the Spline Toolbox is unavailable, then the routine `l_curve` can still be used, but it cannot be used to locate the "corner" of a discrete L-curve, and in this situation it returns `reg_corner` = `NaN`.

REGULARIZATION ROUTINES

`cgls`	Computes the least squares solution based on k steps of the conjugate gradient algorithm.
`discrep`	Minimizes the solution (semi)norm subject to an upper bound on the residual norm (discrepancy principle).
`dsvd`	Computes a damped SVD/GSVD solution.
`lsqi`	Minimizes the residual norm subject to an upper bound on the (semi)norm of the solution.
`lsqr`	Computes the least squares solution based on k steps of the LSQR algorithm.
`maxent`	Computes the maximum entropy regularized solution.
`mtsvd`	Computes the modified TSVD solution.
`nu`	Computes the solution based on k steps of Brakhage's iterative ν-method.
`pcgls`	Same as `cgls`, but for general-form regularization.
`plsqr`	Same as `lsqr`, but for general-form regularization.
`pnu`	Same as `nu`, but for general-form regularization.
`tgsvd`	Computes the truncated GSVD solution.
`tikhonov`	Computes the Tikhonov regularized solution.
`tsvd`	Computes the truncated SVD solution.
`ttls`	Computes the truncated TLS solution.

ANALYSIS ROUTINES

`fil_fac`	Computes filter factors for some regularization methods.
`gcv`	Plots the GCV function and computes its minimum.
`lagrange`	Plots the Lagrange function $\|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2$ and its derivative.
`l_corner`	Locates the L-shaped corner of the L-curve.
`l_curve`	Computes the L-curve, plots it, and computes its corner.
`picard`	Plots the (generalized) singular values, the Fourier coefficients for the right-hand side, and a possibly smoothed curve of the solution's Fourier-coefficients.
`plot_lc`	Plots an L-curve.
`quasiopt`	Plots the quasi-optimality function and computes its minimum.

TEST PROBLEMS

`baart`	First kind Fredholm integral equation.
`deriv2`	Computation of second derivative.
`foxgood`	Severely ill-posed test problem.

`heat`	Inverse heat equation.
`ilaplace`	Inverse Laplace transformation.
`parallax`	Stellar parallax problem with real observations.
`phillips`	Phillips' "famous" test problem.
`shaw`	One-dimensional image restoration model.
`spikes`	Test problem with a "spiky" solution.
`ursell`	Integral equation with no square integrable solution.
`wing`	Test problem with a discontinuous solution.

STANDARD-FORM TRANSFORMATION

`gen_form`	Transforms a standard-form solution back into the general-form setting.
`std_form`	Transforms a general-form problem into one in standard form.

UTILITY ROUTINES

`bidiag`	Bidiagonalization of a matrix by Householder transformations.
`bsvd`	Computes the singular values, or the compact SVD, of a bidiagonal matrix stored in compact form.
`csd`	Computes the CS decomposition.
`csvd`	Computes the compact SVD of an $m \times n$ matrix.
`get_l`	Produces a $(n-d) \times n$ matrix which is the discrete approximation to the dth order derivative operator.
`gsvd`	Computes the generalized SVD of a matrix pair.
`lanc_b`	Performs k steps of the Lanczos bidiagonalization process with/without reorthogonalization.

AUXILIARY ROUTINES

`app_hh_l`	Applies a Householder transformation from the left.
`gen_hh`	Generates a Householder transformation.
`heb_new`	Newton–Raphson iteration with Hebden's rational approximation, used in `lsqi`.
`lsolve`	Inversion with A-weighted generalized inverse of L.
`ltsolve`	Inversion with transposed A-weighted generalized inverse of L.
`mgs`	Modified Gram–Schmidt orthonormalization.
`newton`	Newton–Raphson iteration, used in `discrep`.
`pinit`	Initialization for treating general-form problems.
`pythag`	Computes $\sqrt{a^2 + b^2}$.

`regudemo`	Tutorial introduction to REGULARIZATION TOOLS.
`spleval`	Computes points on a spline or spline curve.

References

[1] R.C. Allen, Jr., W.R. Boland, V. Faber and G.M. Wing, Singular values and condition numbers of Galerkin matrices arising from linear integral equations of the first kind, J. Math. Anal. Appl. 109 (1985) 564–590.

[2] R.S. Anderssen and P.M. Prenter, A formal comparison of methods proposed for the numerical solution of first kind integral equations, J. Austral. Math. Soc. (Series B) 22 (1981) 488–500.

[3] C.T.H. Baker, Expansion methods, chapter 7 in [18].

[4] C.T.H. Baker, *The Numerical Treatment of Integral Equations* (Clarendon Press, Oxford, 1977).

[5] C.T.H. Baker and G.F. Miller (eds.), *Treatment of Integral Equations by Numerical Methods* (Academic Press, New York, 1982).

[6] D.M. Bates, M.J. Lindstrom, G. Wahba and B.S. Yandell, GCVPACK – routines for generalized cross validation, Commun. Statist.-Simul. 16 (1987) 263–297.

[7] M. Bertero, C. De Mol and E.R. Pike, Linear inverse problems with discrete data: II. Stability and regularization, Inverse Problems 4 (1988) 573–594.

[8] Å. Björck, A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations, BIT 28 (1988) 659-670.

[9] Å. Björck, Least squares methods, in: *Handbook of Numerical Analysis*, Vol. 1, eds. P.G. Ciarlet and J.L. Lions (Elsevier, Amsterdam, 1990).

[10] Å. Björck and L. Eldén, Methods in numerical algebra for ill-posed problems, Report LiTH-MAT-R33-1979, Dept. of Mathematics, Linköping University (1979).

[11] H. Brakhage, On ill-posed problems and the method of conjugate gradients, in: *Inverse and Ill-Posed Problems*, eds. H.W. Engl and C.W. Groetsch (Academic Press, Boston, 1987).

[12] T.F. Chan and P.C. Hansen, Some applications of the rank revealing QR factorization, SIAM J. Sci. Stat. Comp. 13 (1992) 727-741.

[13] J.A. Cochran, *The Analysis of Linear Integral Equations* (McGraw–Hill, New York, 1972).

[14] D. Colton and R. Kress, *Integral Equation Methods for Scattering Theory* (Wiley, New York, 1983).

[15] L.J.D. Craig and J.C. Brown, *Inverse Problems in Astronomy* (Adam Hilger, Bristol, 1986).

[16] J.J.M. Cuppen, A numerical solution of the inverse problem of electrocardiology, Ph.D. Thesis, Dept. of Mathematics, Univ. of Amsterdam (1983).

[17] L.M. Delves and J.L. Mohamed, *Computational Methods for Integral Equations* (Cambridge University Press, Cambridge, 1985).

[18] L.M. Delves and J. Walsh (eds.), *Numerical Solution of Integral Equations* (Clarendon Press, Oxford, 1974).

[19] J.B. Drake, ARIES: a computer program for the solution of first kind integral equations with noisy data, Report K/CSD/TM43, Dept. of Computer Science, Oak Ridge National Laboratory (October 1983).

[20] M.P. Ekstrom and R.L. Rhodes, On the application of eigenvector expansions to numerical deconvolution, J. Comp. Phys. 14 (1974) 319–340.

[21] L. Eldén, A program for interactive regularization, Report LiTH-MAT-R-79-25, Dept. of Mathematics, Linköping University, Sweded (1979).

[22] L. Eldén, Algorithms for regularization of ill-conditioned least-squares problems, BIT 17 (1977) 134–145.

[23] L. Eldén, A weighted pseudoinverse, generalized singular values, and constrained least squares problems, BIT 22 (1982) 487–501.
[24] L. Eldén, A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems, BIT 24 (1984) 467–472.
[25] H.W. Engl and J. Gfrerer, A posteriori parameter choice for general regularization methods for solving linear ill-posed problems, Appl. Numer. Math. 4 (1988) 395–417.
[26] R.D. Fierro and J.R. Bunch, Collinearity and total least squares, Report, Dept. of Mathematics, Univ. of California, San Diego (1991), to appear in SIAM J. Matrix Anal. Appl.
[27] R.D. Fierro, P.C. Hansen and D.P. O'Leary, Regularization and total least squares, manuscript in preparation for SIAM J. Sci. Comp.
[28] R. Fletcher, *Practical Optimization Methods*. vol. 1, 2nd ed. (Wiley, Chichester, 1987).
[29] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed. (Johns Hopkins, Baltimore, 1989).
[30] G.H. Golub and U. von Matt, Quadratically constrained least squares and quadratic problems, Numer. Mathematik 59 (1991) 561-580.
[31] C.W. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind* (Pitman, Boston, 1984).
[32] C.W. Groetsch, *Inverse Problems in the Mathematical Sciences* (Vieweg Verlag, Wiesbaden), to be published.
[33] C.W. Groetsch and C.R. Vogel, Asymptotic theory of filtering for linear operator equations with discrete noisy data, Math. Comp. 49 (1987) 499–506.
[34] J. Hadamard, *Lectures on Cauchy's Problem in Linear Differential Equations* (Yale University Press, New Haven, 1923).
[35] M. Hanke, Regularization with differential operators. An iterative approach, J. Numer. Funct. Anal. Optim. 13 (1992) 523–540.
[36] M. Hanke, Iterative solution of underdetermined linear systems by transformation to standard form, submitted to J. Num. Lin. Alg. Appl.
[37] M. Hanke and P.C. Hansen, Regularization methods for large-scale problems, Report UNIC-92-04, UNI•C (August 1992), to appear in Surveys Math. Ind.
[38] P.C. Hansen, The truncated SVD as a method for regularization, BIT 27 (1987) 543–553.
[39] P.C. Hansen, Computation of the singular value expansion, Computing 40 (1988) 185–199.
[40] P.C. Hansen, Perturbation bounds for discrete Tikhonov regularization, Inverse Problems 5 (1989) L41–L44.
[41] P.C. Hansen, Regularization, GSVD and truncated GSVD, BIT 29 (1989) 491–504.
[42] P.C. Hansen, Truncated SVD solutions to discrete ill-posed problems with ill-determined numerical rank, SIAM J. Sci. Statist. Comp. 11 (1990) 503–518.
[43] P.C. Hansen, Relations between SVD and GSVD of discrete regularization problems in standard and general form, Lin. Alg. Appl. 141 (1990) 165–176.
[44] P.C. Hansen, The discrete Picard condition for discrete ill-posed problems, BIT 30 (1990) 658–672.
[45] P.C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, SIAM Rev. 34 (1992) 561–580.
[46] P.C. Hansen, Numerical tools for analysis and solution of Fredhold integral equations of the first kind, Inverse Problems 8 (1992) 849–872.
[47] P.C. Hansen, Regularization Tools, a Matlab package for analysis and solution of discrete ill-posed problems; Version 2.0 for Matlab 4.0, Report UNIC-92-03 (March 1993). Available in PostScript form by e-mail via netlib (netlib@research.att.com) from the library NUMERALGO.
[48] P.C. Hansen and D.P. O'Leary, The use of the L-curve in the regularization of discrete ill-posed problems, Report CS-TR-2781, Dept. of Computer Science, Univ. of Maryland (1991), to appear in SIAM J. Sci. Comp.

[49] P.C. Hansen, D.P. O'Leary and G.W. Stewart, Regularizing properties of conjugate gradient iterations, work in progress.
[50] P.C. Hansen, T. Sekii and H. Shibahashi, The modified truncated SVD method for regularization is general form, SIAM J. Sci. Statist. Comp. 13 (1992) 1142-1150.
[51] B. Hofmann, *Regularization for Applied Inverse and Ill-Posed Problems* (Teubner, Leipzig, 1986).
[52] T. Kitagawa, A deterministic approach to optimal regularization – the finite dimensional case, Japan J. Appl. Math. 4 (1987) 371–391.
[53] R. Kress, *Linear Integral Equations* (Springer, Berlin, 1989).
[54] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems* (Prentice–Hall, Englewood Cliffs, 1974).
[55] P. Linz, Uncertainty in the solution of linear operator equations, BIT 24 (1984) 92–101.
[56] *Matlab Reference Guide* (The MathWorks, MA, 1992).
[57] K. Miller, Least squares methods for ill-posed problems with a prescribed bound, SIAM J. Math. Anal. 1 (1970) 52–74.
[58] V.A. Morozov, *Methods for Solving Incorrectly Posed Problems* (Springer, New York, 1984).
[59] F. Natterer, *The Mathematics of Computerized Tomography* (Wiley, New York, 1986).
[60] F. Natterer, Numerical treatment of ill-posed problems, in [67].
[61] D.P. O'Leary and J.A. Simmons, A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems, SIAM J. Sci. Stat. Comp. 2 (1981) 474–489.
[62] C.C. Paige and M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1982) 43-71.
[63] R.L. Parker, Understanding inverse theory, Ann. Rev. Earth Planet Sci. 5 (1977), 35–64.
[64] D.L. Phillips, A technique for the numerical solution of certain integral equations of the first kind, J. ACM 9 (1962) 84–97.
[65] F. Santosa, Y.-H. Pao, W.W. Symes and C. Holland (eds.), *Inverse Problems of Acoustic and Elastic Waves* (SIAM, Philadelphia, 1984).
[66] C. Ray Smith and W.T. Grandy, Jr. (eds.), *Maximum-Entropy and Bayesian Methods in Inverse Problems* (Reidel, Boston, 1985).
[67] G. Talenti, *Inverse Problems*, Lecture Notes in Mathematics 1225 (Springer, Berlin, 1986).
[68] H.J.J. te Riele, A program for solving first kind Fredholm integral equations by means of regularization, Comp. Phys. Comm. 36 (1985) 423–432.
[69] A.N. Tikhonov, Solution of incorrectly formulated problems and the regularization method, Dokl. Adak. Nauk. SSSR 151 (1963) 501–504 [Soviet Math. Dokl. 4 (1963) 1035–1038].
[70] A.N. Tikhonov and V.Y. Arsenin, *Solutions of Ill-Posed Problems* (Winston, Washington, DC, 1977).
[71] A.N. Tikhonov and A.V. Goncharsky, *Ill-Posed Problems in the Natural Sciences* (MIR, Moscow, 1987).
[72] A. van der Sluis, The convergence behavior of conjugate gradients and Ritz values in various circumstances, in: *Iterative Methods in Linear Algebra*, eds. R. Bauwens and P. de Groen (North-Holland, Amsterdam, 1992).
[73] A. van der Sluis and H.A. van der Vorst, SIRT- and CG-type methods for iterative solution of sparse linear least-squares problems, Lin. Alg. Appl. 130 (1990) 257–302.
[74] J.M. Varah, On the numerical solution of ill-conditioned linear systems with applications to ill-posed problems, SIAM J. Numer. Anal. 10 (1973) 257–267.
[75] J.M. Varah, A practical examination of some numerical methods for linear discrete ill-posed problems, SIAM Rev. 21 (1979) 100–111.
[76] J.M. Varah, Pitfalls in the numerical solution of linear ill-posed problems, SIAM J. Sci. Statist. Comp. 4 (1983) 164–176.
[77] C.R. Vogel, Optimal choice of a truncation level for the standard SVD solution of linear first kind integral equations when data are noisy, SIAM J. Numer. Anal. 23 (1986) 109–117.

[78] C.R. Vogel, Solving ill-conditioned linear systems using the conjugate gradient method, Report, Dept. of Mathematical Sciences, Montana State University (1987).

[79] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59 (SIAM, Philidelphia, 1990).

[80] G.M. Wing, Condition numbers of matrices arising from the numerical solution of linear integral equations of the first kind, J. Integral Equations 9 (Suppl.) (1985) 191–204.

[81] G.M. Wing and J.D. Zahrt, *A Primer on Integral Equations of the First Kind* (SIAM, Philidelphia, 1991).

[82] H. Zha and P.C. Hansen, Regularization and the general Gauss–Markov linear model, Math. Comp. 55 (1990) 613–624.