Surv. Math. Ind. (1993) 3: 253–315

Surveys on Mathematics for Industry

Regularization methods for large-scale problems

Martin Hanke, Karlsruhe and Per Christian Hansen, Lyngby

Summary. Linear ill-posed problems give rise to ill-conditioned algebraic systems of linear equations. In this paper, we survey the theoretic and, in particular, the algorithmic treatment of such systems by means of regularization methods, focusing on methods that are specially suited for large-scale problems. The treatment is illustrated numerically by two examples, one from inverse helioseismology and one from image reconstruction.

AMS Subject Classifications: 65-02, 65R30.

Keywords and phrases: Ill-posed problems; Tikhonov regularization; Iterative regularization; Parameter-choice strategies; Inverse helioseismology; Image reconstruction.

1 Introduction

Back in 1923 Hadamard [41] introduced the concept of *well-posed* and *ill-posed problems*; to the latter class belong particularly those problems where the solution does not depend continuously on the given data. According to Hadamard's philosophy, ill-posed problems are actually ill posed, in the sense that the underlying model is wrong.

In the meantime, many applications in industry have led to the consideration of such ill-posed problems where the solution has a well-defined physical sense. Such applications arise quite naturally if one is interested in determining the internal structure of a physical system from the system's behavior; therefore, many of these applications are entitled *inverse problems* (in contrast to *direct problems* where the interest is in the system's behavior given the internal structure of the system). An interesting survey on inverse problems emphasizing both their origin in industrial applications and their mathematical treatment from a more theoretical point of view, has been given recently by Engl [28] in this journal. Our survey, on the other hand, focuses more on numerical techniques. We therefore prefer to sketch only a few examples, and refer to the extensive bibliography in [28] for further applications. Our examples in Sect. 2 may be seen as particular case studies that illustrate the different structures which may be met in practice.

It was in the early 60s of this century that mathematicians initiated a rigorous investigation of ill-posed problems. Thirty years have passed since Tikhonov introduced *regularization methods* in 1963, and today the theoretical basis for the solution of ill-posed problems is very well developed—at least for linear equations. Excellent treatments of this theory may be found in the books by Groetsch [39] and Louis [69], for instance.

In spite of this progress on the theoretical side, algorithmic aspects have not found comparable attention; the first sophisticated algorithmic developments go

back to the late 70s, and the few later developments are widely spread in the literature. It is striking that the only extensive survey on algorithmic details was written by Björck and Eldén [12] in 1979 for conference proceedings which, unfortunately, never appeared. To some extent, algorithmic details have been presented by Morozov [76], Natterer [82], Baumeister [6], and Björck [11]. The same situation is reflected on the software side: only a limited number of packages is available, e.g., by Drake [22], te Riele [100], Wahba and her co-workers [5], and the MATLAB package [50].

This lack of algorithmic development is especially disappointing, as more and more real-world problems need to be considered in higher dimensions, or require real-time solutions, so that the very straightforward, simple implementation of the basic algorithms no longer fulfills the storage or computer-time restrictions. This is particularly true in areas such as, e.g., medical imaging, seismic inversion, and signal processing. Such problems require much more sophisticated numerical tools.

The underlying idea of the present work is to put together in one common framework the more theoretical results of those authors preferring a purely mathematical "Theorem-Proof"-type language, as well as the computational results of those authors whose insight is often based on heuristic arguments and for which a mathematical foundation is out of reach. It is our strong belief that anybody working in this fascinating area of applied mathematics should have an understanding, not only of the mathematical reasoning behind ill-posed problems, but also of the more practical aspects that help to bridge the gap between theory and engineering requirements.

According to Linz [66: p. 93], "there are no general criteria by which different algorithms [for finding approximate solutions of ill-posed equations] can be compared". Nevertheless, an important aspect of our work is to try to compare different algorithms for computing regularized solutions of large-scale problems. We will do this on the basis of the theoretical (convergence) results and on the basis of efficiency of possible implementations. One outcome of our work (in agreement with Linz' statement above) is that there is no single method which is superior to everything else but that the optimal algorithm depends on a large variety of parameters. We mainly focus on the classical (direct) method of Tikhonov and the much more recent development of iterative regularization methods. We believe that the latter methods have a good chance to become much more important in connection with large-scale regularization problems. Trying to be as general as possible, we decided to leave aside other methods that nevertheless may well be important in specialized applications: to this category we count *truncated singular value decomposition* which is better known as *band-pass filtering* in all deconvolution-type problems [6, 69, 47], and the so-called *mollifier methods* (cf., e.g., [70]), a representative of which is the algorithm of Backus and Gilbert [2]. In spite of their potential importance, we also decided not to deal with any sort of a priori constraints such as inequality constraints, nonnegativity (including the idea of maximum entropy) or a priori known discontinuities.

We begin our discussion with a brief description of three case studies in Sect. 2. We are mainly concerned with two of these problems, namely *inverse helioseismology* and *image recovery*, which we have used for our numerical examples. We then

turn to a more generic treatment of Fredholm integral equations of the first kind in Sect. 3, in order to explain the meaning and consequences of ill-posedness. There, we also briefly discuss discretization issues, but only as much as is required for later developments.

The next four sections contain the algorithmic exposition. In Sect. 4 we introduce the generalized singular value decomposition (GSVD) of two-matrices and explain its intrinsic importance for regularization. Proper use of the GSVD is the first step, when transforming the discrete problem in more amenable form. A highly efficient implementation of Tikhonov regularization is then described in Sect. 5. The basic steps of this implementation are more or less standard by now, but in addition we also include two sections on the choice of adequate regularization parameters, a topic which is quite often skipped in the literature.

Section 6 explains the use of iterative methods for regularization. We describe the ν-methods that have been introduced recently [13], and the conjugate gradient (CG) method which, although classical, has rarely been considered a regularizing algorithm; we believe that CG is the most promising of these algorithms. In Sect. 6 we also discuss some stopping rules for the iterative methods.

Certain variants of the LSQR algorithm of Paige and Saunders [88]—an implementation of the CG method particularly suited for- ill-conditioned matrices—play a role in between Tikhonov's method and iterative methods; we describe these recently developed aspects in Sect. 7.

Finally, in Sect. 8 we illustrate the numerical regularization methods by applying them to the abovementioned problems from helioseismology and from image recovery. In particular, we illustrate the robustness and efficiency of various combinations of regularization methods and parameter-choice strategies.

2 Some specific ill-posed problems

In this chapter we consider three case studies for large-scale applications of ill-posed problems from science and engineering. We emphasize that these examples mainly serve to illustrate the different structures that may arise in applications. We hope that this will suffice as a motivation for the following detailed description of regularization methods. Other examples, with particular emphasis of industrial applications can be found in [28] and the references therein; see also [106] for applications considered in the Russian literature.

2.1 Inverse helioseismology

Our first example arises in astrophysics and is concerned with computation of the internal structure and dynamics of the sun. We shall give a somewhat detailed discussion of this problem, because we will return to it in Sect. 8.1.

The aim of *inverse helioseismology* is, from measurements of oscillations of the surface of the sun (which can be made from earth or—in the future—from satellites), to determine the internal structure and rotation of the sun, cf., e.g., [87]. This information, in turn, gives the astronomers a means for comparing their models of the sun's interior dynamics and temperature with actual observations.

For example, the hope is to solve the *neutrino problem*, namely that the measured neutrino flux from the sun is by a factor three smaller than that expected from some of the current solar models. Helioseismology is also a means for studying the *solar dynamo*, i.e., the convection of the plasma in which kinetic energy is converted to magnetic energy, and which is widely believed to be responsible for the sun's magnetic field.

Measurements are made today in a worldwide collaboration with many observatories throughout the earth, the reason being that uninterrupted observations of the sun (i.e., observations not interrupted during the night) are essential for the quality of the measured data. Once a set of observations has been made, it should therefore be considered as fixed; it is very difficult to make additional or refined measurements. Hence, in connection with the numerical solution of the inverse helioseismic problem, the aim of the regularization process is to "squeeze out" as much information as possible from this fixed set of data.

To simplify our discussion here, we assume that the internal rotation in the sun depends on depth only—and not on latitude. In this case, two quantum numbers n and l suffice to describe its eigen-oscillations. Let $\omega_{n,l}$ denote the eigen-oscillation frequencies, as a function of the two quantum numbers, that one would have observed if the sun were not rotating. These frequencies can be expressed in terms of spherical harmonics and their derivatives; we omit the details here and refer instead to Christensen-Dalsgaard et al. [16] and the references therein. However, the real sun is rotating, with a rotation frequency much lower than the frequencies of the observed modes, and as a consequence the observed frequencies are shifted to $\omega_{n,l} + \delta_{n,l}$. From the measurements of the sun, one can determine these frequency shifts $\delta_{n,l}$, caused by the rotation, very accurately, and the number of observable modes (i.e., combinations of n and l) is as many as a few thousands.

We note in passing that in the future, the astronomers wish to incorporate the dependence of the internal rotation on latitude as well. This will certainly be possible when satellite-based observations become available. However, it is outside the scope of the present discussion to go into these aspects.

We shall now give an expression that relates the frequency shifts to the internal solar rotation, which is the quantity that we are interested in resolving. Let $f(r)$ denote the exact internal solar rotation (angular velocity) at a distance r from the center, and let R denote the radius of the sun. Then the frequency shifts $\delta_{n,l}$ can be expressed as follows:

$$\delta_{n,l} = \int_0^R k_{n,l}(r) f(r)\, dr, \quad \{n, l\} \in \text{the observed mode set}. \tag{2.1}$$

Here, $k_{n,l}$ are functions that depend on n, l, and the underlying solar model. The particular version of $k_{n,l}$ that we shall use here has the form

$$k_{n,l} = \frac{((\xi_{n,l}(r) - \eta_{n,l}(r))^2 + (L^2 - 2)\eta_{n,l}(r)^2)\rho(r)r^2}{\int_0^R (\xi_{n,l}(r)^2 + L^2\eta_{n,l}(r)^2)\rho(r)r^2\, dr}.$$

Here, ρ is the density of the sun, and $L^2 = l(l+1)$ where $l = 1, 2, \ldots$. The functions $\xi_{n,l}$ and $\eta_{n,l}$ are too complicated to give in details here—they are eigenfunctions that can be determined for a given stellar model by solving the equations of nonradial stellar oscillations. An important feature of the functions $k_{n,l}$ is that they are all nonnegative, and they go to zero as r^{2l} for $r \to 0$. As a consequence, when Eq. (2.1) is solved for the unknown function f, then it is very difficult to recover information about the solution f near the center of the sun, because this information is "annihilated" by the integration with $k_{n,l}$. The information about the solution off the center is not so difficult to recover. The physical reason for this phenomenon is that none of the oscillations, consisting of pressure waves, that can be observed at the surface of the sun penetrate all the way to the center—hence, the observed data can rarely provide information about the solution near the sun's center.

Much effort is currently being put into solving model problems where a known solution f is specified, and the aim is to recover all features of f as accurately as possible. Work along this line is described in, e.g., [15, 16]. This is also the type of problem that we shall use later in Sect. 8 where we give some numerical examples.

2.2 *Computerized tomography*

Another source for ill-posed inverse problems is computerized tomography. The idea of tomography is to reconstruct a function f with compact support $\Omega \subset \mathbb{R}^2$, say, out of a system of line integrals of f over paths intersecting the region of interest, Ω. Below, we shall discuss two particular applications in more detail.

In transmission tomography of diagnostic radiology one is interested in the X-ray attenuation coefficients within a cross-section Ω of the human body. Characteristic values of the X-ray attenuation coefficients are associated with each type of tissue, and in this way one is able to detect the location of tumors or other anomalies inside the body.

We denote by $f(x, y)$ the X-ray attenuation coefficient at (x, y). The information on f is now obtained by scanning the body with relatively thin X-ray beams from a number of sources and by measuring the loss of intensity at detectors placed on the opposite side of the body. If Γ denotes one particular straight-line connection between one of the sources and one corresponding detector, then we obtain the following identity:

$$\exp\left(- \int_\Gamma f(x, y)\, ds \right) = E_d / E_s,$$

where E_s and E_d are the intensities of the X-ray beam at the source and detector, respectively, and s is the arc length along Γ.

We mention in passing that the above model is a rather simplified one. In practice the X-ray beam will rather be polychromatic than monochromatic, and one should also take beam scattering during the passage of the body into account. Finally, a more realistic model should include the nonlinear dependence of f on the X-ray's energy (beam hardening). We refer to [53, 81] for further details.

Incorporating all admissible source/detector constellations—which of course depend on the particular geometry of the scanner—we end up with a *system of*

linear integral equations for f,

$$\int_{\Gamma_i} f(x, y)\, ds = \gamma_i, \quad i = 1, 2, \ldots, m. \tag{2.2}$$

Here, m is the number of constellations, which also equals the dimension of the data vector $(\gamma_1, \ldots, \gamma_m)^T$. In other words we have a discrete sample of an integral transform of f which is known as the *Radon transform* [81]. The inversion of the Radon transform is an ill-posed problem. An additional difficulty is that in the above setting f is not uniquely determined by (2.2) due to the finite number of data that are available. This problem is usually handled by imposing side conditions on f, e.g., by restricting f to an appropriate finite-dimensional affine subspace.

In seismic borehole tomography, cf., e.g., [7], geophysicists are interested in the structure of a section of the earth's crust. To obtain data, seismic or electromagnetic waves are sent through the earth and their traveltimes are measured. Transmitters and receivers of these waves are located in separate vertical boreholes. Additional transmitters and/or receivers can be positioned on the surface of the earth between the boreholes. The goal of the experiment is to recover the seismic slowness distribution in the earth. According to a simple physical model the signals follow straight lines; of course, more complicated models include diffraction, and reflections at the layers of the earth.

The slowness distribution f is the reciprocal of the propagation velocity v which is identified with the structure of the layers in the earth. Thus, given the path Γ_i betwen a transmitter and a receiver, and the measured traveltime τ_i, we obtain the equation

$$\int_{\Gamma_i} \frac{1}{v(x, y, z)}\, ds = \tau_i.$$

Practical limitations of the experimental setting are the enormous costs and restrictions associated with the construction of boreholes. Only a restricted number of measured traveltimes τ_i is thus available, and these problems are therefore highly underdetermined.

We emphasize the structure within these tomographic applications. Every datum γ_i of the right-hand side of (2.2) only depends on a small portion of f, namely on its values along Γ_i. After careful discretization—a subject that we return to in the sequel—the coefficient matrix of the algebraic system of equations should therefore be a sparse matrix. This is in contrast to the helioseismology example where all values on the right-hand side will ultimately change when the solution f is altered on a set of positive Lebesgue measure; such problems will give dense coefficient matrices.

2.3 *Signal processing*

Many problems in signal processing also give rise to large-scale ill-posed problems. As a specific example of an ill-posed problem that arises in optics, we shall briefly review confocal scanning microscopy by which it is possible to produce images of fluorescent biological objects, cf., e.g., [8, 9]. In a confocal scanning microscope, the object is illuminated by incident light focused on the object by means of a lens (the illumination lens). The image of the object is formed by a second lens (the collector lens). The two lenses share the same focus, hence the name confocal microscopy. The confocal principle provides improved resolution and imaging fidelity compared to conventional light microscopy.

Assume that the point spread functions for the illumination and collector lenses are s_i and s_c, respectively, and let f denote the 2-D function that describes the object. Then the 2-D image g formed by the microscope is given by

$$g(x, y) = \int_{\Omega} s_c(x - x', y - y') s_i(x', y') f(x', y')\, dx'\, dy'. \tag{2.3}$$

Here, Ω is a region which covers the object that is being observed. It is this equation that one must solve in order to compute the unknown function f from the measured image g.

In the special case where the object is uniformly illuminated, the above relation becomes somewhat simpler because the function s_i drops out. In this case, the integral equation becomes a standard *deconvolution problem*. As a simple example, consider the case where the collector lens is simply an aperture of width d: then we have

$$s_i \equiv 1 \quad \text{and} \quad s_c(x, y) = d^2 \operatorname{sinc}(\pi dx) \operatorname{sinc}(\pi dy),$$

where, as usual,

$$\operatorname{sinc} t = \frac{\sin t}{t}.$$

It is interesting to note that exactly the same type of equation arises in many other applications, all of which are concerned with extrapolation of band-limited signals.

Another deconvolution problem arises in the imaging of astronomical objects. There, one common model is (2.3) with

$$s_i \equiv 1 \quad \text{and} \quad s_c(x, y) = \exp(-\gamma(x^2 + y^2)), \tag{2.4}$$

where γ is a constant, and where s_c is a very simple model for the point spread function due to atmospheric turbulences caused by random variations in the refractive index, cf., e.g., [57, p. 269].

Although deconvolution problems are easily solved analytically by means of Fourier transformations, straightforward numerical algorithms suffer from instabilities caused by noisy data. In image reconstruction problems, data noise has several components some of which are signal dependent, others are not. Signal dependent components may be due to nonlinearities within the recording process; signal independent components of the noise, on the other hand, are often modeled as Gaussian white noise covering a wide band of frequencies. Yet other types of data perturbations appear when the signal is discretized or sampled, but we shall not go into these details.

An important issue in connection with deconvolution problems is that the discrete systems derived from discretization of such problems have highly structured coefficient matrices with small displacement rank, cf. [58, 17], e.g., (block)-Toeplitz derived matrices or (block)-Hankel derived matrices. Whenever possible, it will be helpful to take advantage of this feature in the numerical treatment of the problem.

3 Theoretical background and basic philosophies

The aforementioned problems from Sect. 2 are particular instances of a Fredholm integral equation of the first kind, which we write in the generic form

$$\int_\Omega k(s,t)x(t)\,dt = g(s), \quad s\in\Omega\subset\mathbb{R}^q. \tag{3.1}$$

For the time being we restrict our attention to Eq. (3.1) with kernel $k(\cdot,\cdot)\in L^2(\Omega\times\Omega)$. Without loss of generality we hereby assume that s and t are restricted to the same subset $\Omega\subset\mathbb{R}^q$. Many linear ill-posed problems can be modeled by integral equations of this form. We therefore continue by studying intrinsic properties of these integral equations; we will clarify why integral equations of the first kind are ill-posed problems and discuss the consequences. It will become clear that sophisticated tools are required to resolve these equations.

Such tools usually incorporate some sort of regularization, and in Sect. 3.2 we will give a brief explanation of the underlying idea. We will there concentrate on the classical method of Tikhonov; we have chosen this approach because Tikhonov regularization has both statistical as well as analytical foundations and, above all, it is largely accepted as a regularization method because of its considerable numerical success in numerous applications.

By investigating Tikhonov regularization in the original, continuous setting of (3.1) we deduce all the required ingredients and suitable discretization procedures that will turn out to be important. Based on this insight we will then be ready to apply other—possibly more efficient and even more accurate—regularization methods to the discretized problem.

3.1. Compact operators in Hilbert spaces

In this section we briefly discuss general properties of integral equations of the first kind; the interested reader who is looking for a more rigorous treatment is referred to [97]. We will be working in the Hilbert space $L^2(\Omega)$ of real functions, and we shall introduce the linear operator $K: L^2(\Omega)\to L^2(\Omega)$ by

$$Kx = \int_\Omega k(\cdot,t)x(t)\,dt.$$

As k belongs to the space $L^2(\Omega\times\Omega)$, K is bounded, i.e., K is continuous with respect to the usual L^2 topology induced by

$$(x,y) = \int_\Omega x(t)y(t)\,dt, \quad \|x\| = (x,x)^{1/2}, \quad x,y\in L^2(\Omega).$$

The adjoint operator K^* of K is defined by

$$K^*y = \int_\Omega k(s,\cdot)y(s)\,ds.$$

Above all, K is a *compact* operator, that is, the image of every (in L^2) bounded sequence of functions has at least one converging subsequence; in other words, the range $\mathcal{R}(K)$ must be somewhat "smaller" than the entire space $L^2(\Omega)$—an observation that we will make more precise below.

Every compact operator between Hilbert spaces admits a *singular value expansion*: there is a non-increasing sequence of positive singular values $\sigma_j, j = 1, 2, \ldots,$ with corresponding singular functions u_j and v_j, such that

$$Kv_j = \sigma_j u_j, \quad K^*u_j = \sigma_j v_j, \quad j = 1,2,\ldots \tag{3.2}$$

In this paper we will restrict our attention to operators K where both K and K^* are injective. This is merely for the ease of presentation; the more general case can be handled analogously. With this assumption, we have infinitely many singular values converging to zero as j goes to infinity, and $\{v_j\}_{j=1,2,\ldots}$ is an *orthonormal basis* of $L^2(\Omega)$.

Although K is injective we emphasize that K is *not* onto (see the *Picard criterion* below), instead $\mathcal{R}(K)$ is merely dense in $L^2(\Omega)$; by a well-known principle of functional analysis, the inverse of K can then be no continuous operator. We shall clarify this by expanding the right-hand side g into the convergent series

$$g = \sum_{j=1}^{\infty} \eta_j u_j, \quad \eta_j = (u_j, g). \tag{3.3}$$

With this expansion at hand, we are ready to characterize $\mathcal{R}(K)$.

Picard condition: A function g given by (3.3) belongs to $\mathcal{R}(K)$ if and only if

$$\sum_{j=1}^{\infty} \left(\frac{\eta_j}{\sigma_j}\right)^2 < \infty.$$

For an arbitrary function $g\in L^2(\Omega)$, the coefficients η_j of (3.3) form a sequence in l^2, i.e., they will tend to zero somewhat faster than $j^{-1/2}$; but if g also satisfies $g\in\mathcal{R}(K)$ then the Picard condition states that the η_j converge to zero faster than $\sigma_j j^{-1/2}$. The latter requirement is, of course, more restrictive because $\sigma_j\to 0$ as $j\to\infty$.

If g of (3.3) belongs to $\mathcal{R}(K)$, then the solution x to (3.1) is given by

$$x = K^{-1}g = \sum_{j=1}^{\infty} \frac{\eta_j}{\sigma_j} v_j, \tag{3.4}$$

where the convergence of the expansion on the right-hand side is guaranteed by the Picard condition. We shall reserve the symbol K^{-1} for the operator—with domain $\mathcal{R}(K)$—which maps each element of $\mathcal{R}(K)$ onto its unique preimage under K.

If $g\notin\mathcal{R}(K)$, we may still approximate g by g_k obtained from truncating the expansion (3.3) after k terms,

$$g_k = \sum_{j=1}^{k} \eta_j u_j;$$

these g_k clearly fulfill the Picard condition for all $k = 1, 2, 3, \ldots,$ and we have

$$K^{-1}g_k = \sum_{j=1}^{k} \frac{\eta_j}{\sigma_j} v_j.$$

We conclude that as $k \to \infty$ we have $g_k \to g$, but

$$\| K^{-1} g_k \| \to \infty , \; k \to \infty .$$

It is for this lack of stability that integral equations of the first kind are considered *ill-posed problems.*

Unfortunately, in practical situations we will typically only have access to an approximate right-hand side g in (3.1), which is contaminated with measurement errors, approximation errors, and rounding errors:

$$g = g^{\text{exact}} + \varepsilon, \quad g^{\text{exact}} \in \mathcal{R}(K), \; \| \varepsilon \| / \| g^{\text{exact}} \| \text{ small}. \tag{3.5}$$

Here, g^{exact} denotes the unknown, exact right-hand side and ε denotes the perturbation. Hence, ideally it is $x^{\text{exact}} = K^{-1} g^{\text{exact}}$ that we want to compute. Of course, we cannot expect ε to satisfy the Picard condition, hence $g \notin \mathcal{R}(K)$. As illustrated in the example above, any naive approach which tries to compute $K^{-1} g$ instead of $K^{-1} g^{\text{exact}}$ will usually diverge or return a useless result with extremely large norm, *no matter how small the norm of the perturbation ε is.*

We mention in passing that there are also important examples of integral equations of the first kind where the associated operator K is *not compact*; convolution equations over $\Omega = \mathbb{R}^q$ (cf. Sect. 2.3) and the inverse Laplace transform constitute such examples. For these operators, compactness is lost because the countably many singular values are replaced by a continuum of spectral elements of K^*K which nevertheless accumulates at the origin. Thus, K^{-1} is still unbounded, and the above discussion more or less applies to these problems as well; we only have to replace sums by integrals and spectral expansion coefficients by spectral distribution functions. We do not elaborate on this any further but refer instead to [38]. What is important for us is that the numerical solution of these equations inherits the same difficulties as the solution of compact operator equations, and the same numerical tools are appropriate to approximate the unknown solution x^{exact}.

3.2 The basic rationale behind regularization

According to Tikhonov and Arsenin [103: sect. II.1] a given algorithm is called a *regularization method* for (3.1) if—in the presence of perturbations ε as in (3.5)—the algorithm determines approximations $x(\varepsilon)$ with the property that

$$x(\varepsilon) \to x^{\text{exact}}, \quad \varepsilon \to 0 .$$

Here, we shall be more specific: a regularization method will always be determined by the following quantities, which are to be elaborated on later:

(i) a *regularization operator* L, defining a (semi-) norm $\| L \cdot \|$ used to measure the "size" of the approximations,
(ii) a set of appropriate *filter functions* f_α or f_k where the *regularization parameter* α (or k) belongs to a suitable real parameter set, and
(iii) a rule for determining an adequate regularization parameter out of this set.

By means of the regularization operator L we will enforce the approximations to belong to affine subspaces of "well-behaved" functions. The role of the filter functions is then to dampen those components in the approximate solution which are especially affected by the noise perturbation ε. We leave a more precise discussion to the following sections.

Regularization may also be introduced as a generalization of that algorithm which is most widely accepted: after Tikhonov's fundamental work [101, 102] this method usually bears his name. We introduce it here to motivate the need for a regularization operator; this is necessary at this stage because it is convenient to incorporate the regularization operator already into the discretization step, see Sect. 3.3.

In Tikhonov's method the approximate solution x_α is defined as the unique minimizer of the quadratic functional

$$\| g - Kx \|^2 + \alpha \| Lx \|^2 \tag{3.6}$$

in the domain of L, which is usually assumed to be a dense subset of $L^2(\Omega)$; note that the uniqueness of x_α is ensured because we have assumed K to be injective. Equivalently, x_α may be introduced as the solution to the *regularized normal equations*

$$(K^*K + \alpha L^*L) x_\alpha = K^* g . \tag{3.7}$$

The parameter $\alpha > 0$ is called the *regularization parameter*. Note that L^*L is positive semidefinite so that the spectrum of the operator on the left-hand side of (3.7) is shifted in the positive direction; hence the solution of the regularized normal equations should be less susceptible to perturbations in the data g. In fact, L will typically be chosen such that Morozov's *complementation condition* [76: p. 3] is fulfilled, i.e.

$$\| Kx \| + \| Lx \| \geq \gamma \| x \| \tag{3.8}$$

for some $\gamma > 0$ and all x in the domain of L. In this case, $K^*K + \alpha L^*L$ is continuously invertible, hence (3.7) has a unique solution x_α depending continuously on g.

Instead of (3.6) one may as well consider the slightly more general functional

$$\| g - Kx \|^2 + \alpha \| L(x - x_{\text{guess}}) \|^2 , \tag{3.9}$$

if one has available a good guess for the exact solution x^{exact}. However, note that (3.9), in turn, can easily be reduced to (3.6) by replacing g by $g - Kx_{\text{guess}}$ in (3.6).

Since its introduction in 1963, many papers have dealt with Tikhonov regularization, suggesting reasonable regularization parameters or regularization operators, and analyzing the convergence of the minimizer x_α of (3.6) to x^{exact} for appropriate $\alpha = \alpha(\varepsilon)$ as ε tends to zero. The reader is encouraged to look at the original papers by Tikhonov [101, 102] for the very first fundamental results; a perfect exposition of what was the state of the art in the mid eighties can be found in [39]. More recently Louis [69] emphasized the concept of optimal information that may be recovered from perturbed data in operator equations, as considered previously by Miller [74] and Melkman and Micchelli [73].

The basic idea of Tikhonov regularization is nevertheless easy to describe. Minimizing the functional (3.6) means to search for some x_α, providing at the same time a small residual $g - Kx_\alpha$ and a moderate value of the penalty function $\|Lx_\alpha\|$. The way in which these two terms are balanced depends on the size of α. Note that if $g \notin \mathcal{R}(K)$ and $\alpha \to 0$, then $\|Lx_\alpha\| \to \infty$ if L and K satisfy the complementation condition (3.8); the rate of divergence will be moderate if the signal-to-noise ratio is sufficiently large.

In any event, if the regularization parameter is chosen too small, Eq. (3.7) is too close to the original ill-posed problem and we must expect instabilities; if α is too large, the problem we solve has only little connection with the original equation. Finding the *optimal parameter* is a tough problem, though, and we will return to that in Sect. 5.

As for the regularization operator L, it is most common to choose either the identity I or a linear differential operator $L \neq I$, defined on some dense Sobolev space $H^p(\Omega) \subset L^2(\Omega)$, $p \geq 0$; we mention that this choice of L usually implies the validity of the complementation condition. If $L = I$ then we shall speak of regularization *in standard form* opposed to regularization *in general form* if $L \neq I$.

We stress that the role of L must not be underestimated. Numerical computations, e.g. by Varah [112] have shown that a wrong choice of L can lead to completely erroneous results. Unfortunately, we are not aware of a general rule for choosing L. Analytical treatments of this problem can only be found in the literature in the special case where K and L commute; we refer, for example, to [103: chap. 5; 18; 80]. Franklin [30] chooses L on the basis of a stochastic reasoning.

As far as a general heuristic in concerned, there are at least two arguments recommending the use of differential operators L over the identity. First, roughly speaking, noisy components in the data will lead to rough oscillations in the reconstructions which will provoke large L-(semi)norms, $\|L\cdot\|$. The standard norm $\|\cdot\|$ need not be affected that much. Second, the regularized approximations will carry inherent smoothness properties, and the covergence to x^{exact} will be in a stronger topology if L is a differential operator, cf. [67]; this may even imply uniform convergence of x_α and some of its derivatives compared to convergence in the mean associated with $L = I$. Note however, that x^{exact} must belong to the domain of L, i.e., x^{exact} must be smooth, to legitimate the use of a differential operator L.

The slight disadvantage with differential operators is that they can have a non-trivial null space, $\mathcal{N}(L)$. As a matter of fact, L^*L in (3.7) will be semidefinite and $\|L\cdot\|$ is a seminorm, only. As we will see in the subsequent chapters this will always cause extra considerations in the numerical algorithms. One way to overcome this difficulty is to include boundary conditions for the approximate solutions: such treatments may be found for example in [39, 55, 67, 63]. As such an approach evidently requires appropriate a priori knowledge about the solution x^{exact} we shall not pursue it any further here.

The alternative approach, which is the one we will follow in this paper, consists—roughly speaking—in a separate treatment of $\mathcal{N}(L)$. We start by resolving the solution's component in $\mathcal{N}(L)$, and then we obtain the remaining component through a transformation into a standard form problem by employing a generalized inverse of L, see Sect. 4. We stress that the extra considerations for $L \neq I$ are not overwhelming and that the computational overhead with $L \neq I$, compared with $L = I$, is in fact negligible.

While Tikhonov's regularization method is widely considered to be the most promising algorithm for approximating x^{exact}, alternative methods have later been introduced to fix inconveniences associated with the original algorithm. Some of these alternatives, e.g. the *iterated Tikhonov method* (Sect. 5.2), have been designed with special regard on the restricted accuracy in the reconstruction of particular smooth functions, which is an inherent drawback of Tikhonov's approach. Other regularization methods, especially the iterative ones like *conjugate gradients* or the *v-methods* (Sect. 6), benefit from the simplicity of their algorithms and the potential reduction of the computer time required as compared to Tikhonov's method. In particular, for large-scale problems like those discussed in Sect. 2, this is a striking argument for seeking alternatives to Tikhnonov's method.

3.3 Discretization issues

To obtain a computer approximation to x^{exact}, we somehow have to discretize the original problem. Incidentally, there are various ways to reduce the integral equation to a finite-dimensional matrix equation: the most notable ones are *quadrature methods*, *collocation methods*, in particular *moment collocation*, and the *Galerkin method*, cf., e.g., [3, 63]. We do not want to elaborate on the different advantages of the respective methods; due to the ill-posedness of the continuous problem, it actually seems as if every problem—possibly even every right-hand side—requires its own sophisticated choice of discretization, cf. [118, 96]. Instead, we shall look at some general aspects of discretizations of (3.1), and we shall consider important numerical issues related to obtaining the discrete system of equations.

It is an important aspect of *projection methods*, like the Galerkin method, that they themselves have an inherent regularizing property, provided the subspaces are suitably chosen. This was observed first by Natterer [78] and has been analyzed further by various authors thereafter (cf., e.g., the treatments and references in [6, 39, 107, 63]). The conclusion is the following. If the discretization is too coarse, then the finite dimensional equation will be fairly well conditioned but its solution will suffer from a large discretization error; this is like using Tikhonov regularization with too large an α. If, on the other hand, the discretization level is too fine, then the influence of the small singular values of K is too significant, and the solution of the discrete problem resembles Tikhonov approximations with α too small. Somewhere in between there is an optimal level of discretization with the resulting approximation being competitive to the one obtained from optimal Tikhonov regularization of the continuous problem.

Unfortunately, for practical purposes this theory is difficult to apply. For one thing, the optimal discretization is hardly ever known in advance. Moreover, even if we were to hit this level of discretization by accident, it would still remain a good idea for the purpose of numerical stability to regularize the finite-dimensional

problem so that its condition number is improved. In practice, we therefore select a discretization too fine and then regularize the discrete equation *as if it were an ill-posed problem.* We emphasize that from a strictly mathematical point of view, the discrete problem is well-posed, as every nonsingular matrix automatically has a continuous inverse; however, due to the huge condition number of the matrix, we will observe similar phenomena as described in the sections before, when we start resolving the finite-dimensional system numerically. The algebraic linear system of equations is therefore often called a *discrete ill-posed problem* [48]. Plato and Vainikko [91] have analyzed this approach, connecting the level of discretization with appropriate regularization parameters.

Following the above argument we would first discretize the continuous equation and then regularize the finite-dimensional system. Another approach would be to first regularize and then somehow discretize the regularized problem; this is more promising as the (artificial) discretization step is performed as late as possible. Such a procedure can follow the lines of [79] (cf. also [39, 62, 29]), namely, minimizing the continuous Tikhonov functional (3.6) over a finite-dimensional subspace $V_n \subset L^2(\Omega)$. As is shown in [39: sect. 4.3], the algorithm by Marti [72] also belongs to this category. Alternatively, we can apply quadrature rules to discretize the second kind integro-differential equation (3.7); this has been analyzed in [40].

We do not elaborate on this any further; instead, we will illustrate the difference between the above two philosophies with a special case study in which—for simplicity—we use the identity as regularization operator. Our main points are to illustrate the difference between the algebraic systems arising from these discretizations, and to make clear that even if we start with regularization in standard form, the discrete regularization problem need not be in standard form.

We first consider the former philosophy of first discretizing and then regularizing. We choose the Galerkin method to discretize the integral equation without regularization. In particular (cf. [63: sect. 13.3]), given two n-dimensional subspaces V_n and W_n of $L^2(\Omega)$, respectively,

$$V_n = \operatorname{span}\{\phi_1, \ldots, \phi_n\} \subset L^2(\Omega), \quad W_n = \operatorname{span}\{\psi_1, \ldots, \psi_n\} \subset L^2(\Omega),$$

we are looking for an approximation x in V_n, i.e.,

$$x = \sum_{i=1}^{n} \xi_i \phi_i. \tag{3.10}$$

Note that in general the dimension of the two subspaces V_n and W_n may be different. The Galerkin conditions now characterize x of the form (3.10) as the unique solution to the finite-dimensional system

$$(\psi_j, Kx - g) = 0, \quad j = 1, 2, \ldots, n,$$

which leads to the matrix equation $M\mathbf{x} = \mathbf{g}$ for the vector $\mathbf{x} = (\xi_1, \ldots, \xi_n)$ of the coefficients in (3.10), with M and $\mathbf{g}$ given by

$$M = [(K\phi_i, \psi_j)]_{i,j=1}^{n}, \quad \mathbf{g} = [(g, \psi_j)]_{j=1}^{n}. \tag{3.11}$$

Applying Tikhonov regularization in standard form in the finite-dimensional

setting amounts to solving the regularized normal equations

$$(M^T M + \alpha I)\mathbf{x} = M^T \mathbf{g}.$$

In the special case where $\psi_j = K\phi_j$ the matrix M is symmetric, positive definite. Thus, regularization could instead be based on *Lavrentiev's method*, i.e., determining $\mathbf{x}$ from the equations

$$(M + \alpha I)\mathbf{x} = \mathbf{g}, \quad \text{where } M^T = M. \tag{3.12}$$

Consider next the alternative approach of regularizing first with $L = I$, and discretizing then. Following [39: sect. 4.2], we search for the unique minimum of the continuous Tikhonov functional (3.6) in V_n. Expanding x as in (3.10) leads to the system of linear equations

$$(M + \alpha N)\mathbf{x} = \mathbf{g}, \tag{3.13}$$

where, again, M and $\mathbf{g}$ are defined by (3.11) with $\psi_j = K\phi_j$, and where N is the Gramian matrix

$$N = [(\phi_i, \phi_j)]_{i,j=1}^{n}. \tag{3.14}$$

Note the difference between (3.12) and (3.13): it basically affects the norm in $\mathbb{R}^n$ with respect to which we regularize the finite dimensional system of equations.

We note in passing that the choice $\psi_j = K\phi_j$, which at first sight may seem reasonable, is not always a good choice. A more adequate choice is suggested in [62].

We thus summarize: even if we choose the regularization operator $L = I$, the discrete regularization problem need still not be in standard form. As we will see in the next chapter, discrete general-form problems can be handled with essentially the same amount of work as standard form problems—at least for matrices N that usually come up in real applications.

4 Reduction to standard form

In the remaining part of the paper we shall always reserve the letter A for the *coefficient matrix* of the discretized problem, and we shall denote by B the *discrete regularization operator.* The discrete data vector will be denoted by $\mathbf{b}$. For the ease of exposition we shall additionally assume throughout that A is a general real rectangular $m \times n$ matrix with a trivial null space, and that B is a general real matrix with dimensions $p \times n$. For our purpose, it is essentially no restriction to assume that these dimensions satisfy $m \geq n \geq p$. Moreover, we skip the special considerations that apply to square, nonnegative semidefinite matrices A and B.

Then, Tikhonov regularization amounts to minimizing the discrete functional

$$\|\mathbf{b} - A\mathbf{x}\|^2 + \alpha \|B\mathbf{x}\|^2,$$

where here, in the finite dimensional context, $\|\cdot\|$ always denotes the Euclidean norm, because then the minimizer $\mathbf{x}_\alpha$ is the (unique) solution of the discrete Euler equations

$$(A^T A + \alpha B^T B)\mathbf{x}_\alpha = A^T \mathbf{b}. \tag{4.1}$$

compare Sect. 3.3 above. There, we emphasized that for various reasons the appropriate discrete regularization problems to solve are the ones in *general form*, i.e., we face discrete regularization operators $B \neq I$. However, when it comes to *algorithmic methods* for treating the discrete regularization problems, it is much simpler to deal with standard-form problems where $B = I$, simply because there is essentially only one matrix involved, namely A—instead of having to deal with two matrices A and B. Therefore, it is convenient to be able to transform a discrete general-form regularization problem into one in standard form in a numerically appealing fashion.

As we shall demonstrate in this section, the work involved in the standard-form transformation is always an order of magnitude, i.e., a factor n, smaller than the work involved in the core regularization process. In other words, the overhead involved in the reduction to standard form is negligible. Hence, there are no computational reasons for not using the appropriate regularization matrix B whenever needed.

A brief note on our operation counts here and throughout the paper: as $p \approx n$ is always satisfied, and as m is rarely more than a small factor, say 1 to 5, greater than n, it is most convenient to express all operation counts with $O(\cdot)$-notation in terms of powers of n only, except for the leading term where we give the exact coefficient.

4.1 The generalized singular value decomposition

The most elegant way to introduce and discuss the transformation to standard form is via the generalized singular value decomposition (GSVD) of the matrix pair (A, B). With the above assumption about the dimensions m, n and p, the GSVD has the form (cf. [36])

$$A = U \begin{pmatrix} \operatorname{diag}(\rho_i) & 0 \\ 0 & I_{n-p} \\ 0 & 0 \end{pmatrix} W^{-1}, \quad B = V(\operatorname{diag}(\tau_i) \quad 0)\, W^{-1}. \tag{4.2}$$

Here, the matrices U, V, and W are given by

$$U = (\mathbf{u}_1, \ldots, \mathbf{u}_m) \in \mathbb{R}^{m \times m}$$

$$V = (\mathbf{v}_1, \ldots, \mathbf{v}_p) \in \mathbb{R}^{p \times p}$$

$$W = (\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbb{R}^{n \times n},$$

where the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are orthonormal (i.e., $U^T U = I_m$ and $V^T V = I_p$), while the vectors $\mathbf{w}_i$ are linearly independent (i.e., W is nonsingular). Moreover, $\operatorname{diag}(\rho_i)$ and $\operatorname{diag}(\tau_i)$ denote $p \times p$ diagonal matrices with positive diagonal entries $\rho_1, \ldots, \rho_p$ and $\tau_1, \ldots, \tau_p$, respectively, whose elements are usually normalized via $\rho_i^2 + \tau_i^2 = 1$, $i = 1, \ldots, p$. The generalized singular values of (A, B) are then defined as the ratios $\sigma_i = \rho_i / \tau_i$, and we shall assume them to appear in non-increasing order, i.e.,

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p > 0.$$

(Notice that some authors prefer the opposite ordering.) The last $n - p$ columns of

W are of particular interest since they form a basis for the null space of B, i.e.,

$$B\mathbf{w}_i = 0, \quad i = p+1, \ldots, n. \tag{4.3}$$

For later use we remark that although we have left some freedom in the choice of these vectors, they must be $A^T A$-orthogonal, i.e., they must satisfy

$$\mathbf{w}_i \perp A^T A \mathbf{w}_j, \quad p+1 \leq i, j \leq n, \quad i \neq j. \tag{4.4}$$

This is easy to see from (4.2).

The GSVD was originally introduced by Van Loan [109] for analysis of matrix pencils $A^T A + \lambda B^T B$, $\lambda \in \mathbb{R}$, and he showed that σ_i^2 are the generalized eigenvalues of $(A^T A, B^T B)$. A similar definition of a generalized singular value expansion in infinite dimensional Hilbert spaces for a compact operator K and a suitable operator L—i.e., K and L satisfying the complementation condition (3.8)—has not been introduced before recently [43]. The use of the GSVD in the analysis of discrete ill-posed problems goes back to Varah's paper [111] from 1979.

If, in particular, B is the identity I of $\mathbb{R}^n$, then the GSVD of (A, I) gives the well-known singular value decomposition (SVD) of A—up to a diagonal scaling of W—and then the numbers σ_i are just the ordinary singular values of A:

$$A = U \operatorname{diag}(\sigma_i) V^T \quad \text{when} \quad B = I. \tag{4.5}$$

If B is nonsingular then the GSVD of (A, B) reduces to the SVD of AB^{-1}; instead of the term GSVD one can therefore also find the term *quotient* SVD in the literature, cf., e.g., [20].

For general matrix pairs (A, B) very little can be said about the appearance of the GSVD. However, for discrete ill-posed problems arising from the discretization of Fredholm integral equations of the first kind, as well as many other ill-posed problems, the following three characteristic features of the GVSD are very often found:

(i) The generalized singular values σ_i decay towards 0, and the decay rate depends on the smoothness of the kernel k of (3.1); typically, we cannot expect a particular gap in the spectrum to separate large generalized singular values σ_i from those that may be judged numerically zero.

(ii) The singular vectors $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$ often have more sign changes in their elements as the corresponding σ_i decreases, i.e., the singular vectors corresponding to tiny singular values represent highly oscillating functions, whereas the singular vectors corresponding to larger singular values represent smooth functions. Also, the last $n - p$ columns of W usually have very few sign changes in their elements because they are basis vectors for the null space of B, cf. (4.3), i.e., they are related to functions in the null space of the differential operator L.

(iii) The "Fourier"-coefficients $\mathbf{u}_i^T \mathbf{b}^{\text{exact}}$ of the unperturbed right-hand side $\mathbf{b}^{\text{exact}}$ (obtained from discretizing g^{exact}) on the average decay faster to zero than the generalized singular values σ_i; this condition is the analog of the Picard condition and was therefore introduced as *discrete Picard condition* in [48].

We emphasize that especially the second of these features will be very difficult—or

perhaps impossible—to prove in most instances, but that it is still observed in many discrete ill-posed problems arising from practical applications.

A particularly nice feature of the GSVD is that it immediately lets us write down a formula for the regularized solution $\mathbf{x}_\alpha$ to Eq. (4.1). To demonstrate this, it is convenient to introduce the *Tikhonov filter functions*

$$\varphi_\alpha^{\mathrm{T}}(\lambda) = \frac{\lambda}{\lambda + \alpha}. \tag{4.6}$$

Then the Tikhonov approximation $\mathbf{x}_\alpha$ can be written as

$$\mathbf{x}_\alpha = \sum_{i=1}^{p} \varphi_\alpha^{\mathrm{T}}(\sigma_i^2) \frac{\mathbf{u}_i^T \mathbf{b}}{\rho_i} \mathbf{w}_i + \sum_{i=p+1}^{n} \mathbf{u}_i^T \mathbf{b}\, \mathbf{w}_i. \tag{4.7}$$

We observe that the regularized solution actually consists of two components. The last, unregularized, component, which is given by

$$\mathbf{x}_* = \sum_{i=p+1}^{n} \mathbf{u}_i^T \mathbf{b}\, \mathbf{w}_i, \tag{4.8}$$

is the component of the solution in the null space of B. It has very few sign changes, and is not affected by regularization. The first component in (4.7), on the other hand, depends via the filter factors $\varphi_\alpha^{\mathrm{T}}(\sigma_i^2)$ on the regularization parameter α.

From (4.6) it is obvious that

$$\varphi_\alpha^{\mathrm{T}}(\lambda) \le \frac{\lambda}{\alpha} \ll 1, \quad \lambda \ll \alpha, \tag{4.9}$$

while for fixed $\lambda = \sigma_i^2$,

$$\varphi_\alpha^{\mathrm{T}}(\sigma_i^2) = 1 - O(\alpha) \to 1, \quad \alpha \to 0.$$

Thus, with an appropriate choice of α, we can dampen the contributions to (4.7) arising from those indices i for which $\sigma_i^2 \ll \alpha$, where—due to the discrete Picard condition—the coefficients $\mathbf{u}_i^T \mathbf{b}$ are dominated by noise and magnified by large factors $1/\rho_i$. On the other hand, for those terms in (4.7) where i is small, $\sigma_i^2 \gg \alpha$, and the noise-component is negligible, the filter factors $\varphi_\alpha^{\mathrm{T}}(\sigma_i^2)$ are almost 1. We can vary α depending on the amount of noise: with little noise, we need less regularization, i.e. less filtering; with more noise, we need more regularization (cf. the discussion in Sect. 3.2).

To some extent, we have taken the extreme point of view in the heuristics above, that we have almost full reconstruction of those components in the solution corresponding to $\sigma_i^2 \gg \alpha$, almost complete dampening of components corresponding to $\sigma_i^2 \ll \alpha$, and that the remaining components in (4.7) can be neglected. This is of course a very simplifying point of view that cannot tell the whole story. A more sophisticated point of view is chosen by Gilliam et al. [32] who introduced the notion of an *effective rank* of the discrete problem by adding up the values of all filter factors $\varphi_{\alpha_*}^{\mathrm{T}}(\sigma_i^2)$ at $\alpha = \alpha_*$, where the value of α_* is obtained from a stochastical argument. According to [32] the effective rank defined this way, which is now a real number, represents the degrees of freedom inherent with the discretization A and the noise-level ε, but also with the exact solution $\mathbf{x}^{\mathrm{exact}}$.

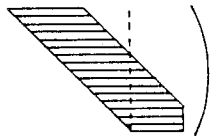
The GSVD of (A, B) is certainly of great theoretical interest for analyzing regularization problems, and it appears implicitly over and over again in this paper, especially in the transformation to standard form described in Section 4.3 below. However, we emphasize that the computation of the GSVD of (A, B) is quite demanding; the implementation by Bai and Demmel [4] requires about $2m^2n + 15n^3$ operations (where A is $m \times n$ and the factor before n^3 depends on the number of iterations required). Hence, in spite of the amount of insight that the GSVD gives into discrete ill-posed problems, it is mainly of computational interest for small- and medium-sized problems. We will rarely use the GSVD for the practical solution of large-scale regularization problems.

4.2 Preprocessing of the discrete system

In many applications, the quantities that the user has available are the matrices A and B, and the vector $\mathbf{b}$. In these situations, we stress that one should never form the cross-product matrices A^TA and B^TB, but instead work directly with A and B. The main reason for avoiding explicit formation of the cross-product matrices is the loss of information due to round-off as illustrated for example in [11: Sect. 6]. This is true for Tikhonov regularization as well as for iterative methods, where the matrix-vector products $A^TA\mathbf{x}$ should be computed as $A^T(A\mathbf{x})$. Moreover, the work involved in forming A^TA explicitly is often comparable with the overall work required for solving the regularization problem.

In some cases—for example in (3.13)—instead of B one has the cross-product matrix $N = B^TB$ at hand. However, for the subsequent algorithms it is always required that the discrete regularization operator B is known explicitly, and that it has the following structure:

$$B = \left(\quad \right) = (B_p \quad B_o), \tag{4.10}$$



that is, B is a banded, upper triangular $p \times n$ matrix, and B_p is its leading $p \times p$ submatrix. Moreover, we assume B_p to be nonsingular. In order to guarantee these assumptions, we shall comment on computing B in (4.10) in the case where only N, or some factorization of N, is available.

We remark that in most practical examples that we are aware of, B is a banded matrix because N is already banded. For example, by taking the basis functions in the Galerkin method to be a certain B-spline basis, the integrals in (3.14) will vanish for $|i - j|$ sufficiently large.

When N is given, then B may be computed by a Cholesky factorization of N. In many interesting cases the rank of N is $p < n$, which ideally calls for a Cholesky factorization with pivoting. However, we prefer to do the factorization without pivoting because pivoting will destroy the banded structure of N and B. Higham [54] has shown that it is possible to skip pivoting whenever N and its leading $p \times p$ principle submatrix are well conditioned, which in turn implies that B_p is well-conditioned. In regularization problems, an ill conditioned or rank deficient matrix

B_p always corresponds to a bad choice of discretization. Hence, it is within the responsibility of the user to select the basis functions for the finite dimensional subspace appropriately, so as to guarantee that B_p is (relatively) well conditioned. We remark that if we are working with spline bases, and if L is a differential operator, then this requirement will usually be fulfilled automatically.

Often we need not even perform the Cholesky decomposition of N, because we already have another factorization $N = C^T C$ at hand, at least implicitly. In this case, a more stable way of obtaining B is to compute the QR factorization $C = QR$, where as usual Q has orthonormal columns and R is upper triangular. We obtain $N = C^T C = R^T R$ and R is exactly what we need. In the rank-deficient case, similar arguments apply as before. For an illustration of the latter approach we refer to the treatment of example (3.13) by Neubauer [85] where he obtained a long but narrow matrix C from the quadrature rule with which the elements of N are computed. He also presents an efficient implementation of the QR factorization for this specific situation; a similar idea can be found in [52].

Whichever situation is actually met, we emphasize that the computation of B can usually be performed in only $O(n)$ flops because of the banded structure of N. As will become clear later, the computation of B will therefore be negligible extra work compared to the overall amount of work for solving (4.1).

As far as the matrix M is concerned, it depends on the regularization method that we use whether a decomposition $M = A^T A$ is required or not. If we use Tikhonov regularization, described in detail in the next chapter, then we explicitly need A because we must reduce this matrix to a bidiagonal matrix. On the other hand, in the iterative regularization schemes described in Sect. 6, only products like Mx are required, whereas A itself is never needed.

4.3 An algorithm for standard-form transformation

We are now searching for a standard-form problem defined in such a way that its solution immediately leads to $\mathbf{x}_z$ of (4.1). To distinguish the standard-form problem from the general form problem (4.1) we shall use bars in our notation, i.e., we are looking for a related problem of the form

$$(\bar{A}^T \bar{A} + \alpha I)\bar{\mathbf{x}} = \bar{A}^T \bar{\mathbf{b}}. \tag{4.11}$$

Also, let $\bar{\mathbf{x}}_z$ denote the solution to (4.11). The transformation that we are going to describe below was originally developed by Eldén [25]. Its usefulness in the context of regularization was later studied in [43]. The same concept implicitly appears in [19].

Consider first the simple case when B is square and invertible. In this case we can derive a standard-form problem (4.11) from (4.1) if we set $\bar{\mathbf{x}} = B\mathbf{x}$, $\bar{A} = AB^{-1}$, and $\bar{\mathbf{b}} = \mathbf{b}$, such that the back-transformation simply is $\mathbf{x}_z = B^{-1}\bar{\mathbf{x}}_z$.

In the general case when B is rectangular and has full row rank, i.e., $B \in \mathbb{R}^{p \times n}$ with rank $(B) = p < n$, we are still led to do the above transformation but now the situation is more complicated because B is not invertible. It is here that the GSVD of A and B plays an important role. The key step is to define the so-called *A-weighted generalized inverse of* B [25] by virtue of (4.2) as follows:

$$B_A^\dagger = W \begin{pmatrix} \operatorname{diag}(\tau_i^{-1}) \\ 0 \end{pmatrix} V^T, \tag{4.12}$$

and we emphasize that $B_A^\dagger$ is different from the Moore–Penrose pseudoinverse $B^\dagger$ in general. Define next the standard-form quantities $\bar{A} \in \mathbb{R}^{m \times p}$ and $\bar{\mathbf{b}} \in \mathbb{R}^m$ of (4.11) as

$$\bar{A} = AB_A^\dagger, \quad \bar{\mathbf{b}} = \mathbf{b} - A\mathbf{x}_*. \tag{4.13}$$

Then one can easily show by means of the GSVD (4.2) that $\mathbf{x}_z$ can be obtained from $\bar{\mathbf{x}}_z$ and $\mathbf{x}_*$ of (4.8) as follows:

$$\mathbf{x}_z = B_A^\dagger \bar{\mathbf{x}}_z + \mathbf{x}_*. \tag{4.14}$$

For the iterative methods, the matrix $\bar{A}$ is of course never formed explicitly as only matrix-vector products with $\bar{A}$ and $\bar{A}^T$ are required. Below, we describe how to compute these products efficiently. For Tikhonov regularization, where the entries of $\bar{A}$ are explicitly required, they can be computed one row at a time using the same algorithms described below. Hence, the transformation is applicable to iterative as well as direct regularization methods. It is especially suited for iterative methods because of the amenable form of the backtransformation which allows a "preconditioning"-type implementation of the iteration (cf. Sect. 6 for an explanation of this).

A crucial point of the implementation is the availability of a basis $W_a = (\mathbf{w}_{p+1}, \ldots, \mathbf{w}_n)$ for the null space of B, $\mathcal{N}(B)$, consisting of $n - p$ vectors which are $A^T A$-orthogonal, i.e., which satisfy (4.4). To compute W_a we need a basis W_o for $\mathcal{N}(B)$. Often, $\mathcal{N}(B)$ is known analytically in advance; e.g., if L is a differential operator then one can determine $\mathcal{N}(L)$ analytically and, with a bit of skill, one can then find appropriate discrete analogs of the functions spanning $\mathcal{N}(B)$. If $\mathcal{N}(B)$ is not known analytically, then W_o can be obtained either by a QR factorization of B^T, or we can make use of the special form (4.10) of B by computing

$$W_o = \begin{pmatrix} B_p^{-1} B_o \\ -I \end{pmatrix} \in \mathbb{R}^{n \times (n-p)}.$$

With the above assumptions, the computation of W_o does not exceed $O(n)$ operations. Given W_o, we can then compute W_a by means of the following algorithm, which is essentially a block Gram–Schmidt orthogonalization with respect to the inner product defined by (4.4):

Algorithm 4.1. (Compute W_a from W_o)

— $AW_o = QR$ (QR factorization with $Q \in \mathbb{R}^{m \times (n-p)}$)
— $W_a = W_o R^{-1}$

The computation of W_a can, of course, also be based on a Cholesky decomposition of $S = W_o^T A^T A W_o$ instead of the QR factorization in Algorithm 4.1; then we do not explicitly need the matrix A in case we are only given the cross-product matrix M.

Usually, the matrix S will be very small, hence we do not expect any loss of stability.

Now that W_a is available, we can comment on the practical implementation of (4.13), following [45]. The key idea is the following: from (4.12) we observe first that $B_A^\dagger \mathbf{z}$ is the unique solution to $B\mathbf{x} = \mathbf{z}$ in the subspace

$$\mathcal{R}(B_A^\dagger) = \operatorname{span}\{\mathbf{w}_1, \ldots, \mathbf{w}_p\}.$$

Given the matrix W_a it immediately follows from the definition (4.2) of the GSVD that

$$P = W_a(A^T A W_a)^T \tag{4.15}$$

is an oblique projector with range and null space as follows:

$$\mathcal{R}(P) = \operatorname{span}\{\mathbf{w}_{p+1}, \ldots, \mathbf{w}_n\} = \mathcal{N}(B),$$
$$\mathcal{N}(P) = \operatorname{span}\{\mathbf{w}_1, \ldots, \mathbf{w}_p\}.$$

Consequently, if $\hat{\mathbf{x}}$ is an arbitrary solution to the system $B\mathbf{x} = \mathbf{z}$, then we obtain the following expression for the matrix-vector product:

$$B_A^\dagger \mathbf{z} = (I - W_a(A^T A W_a)^T)\hat{\mathbf{x}}. \tag{4.16}$$

Using the special structure (4.10) of B that we have assumed before, then one solution $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n)^T$ is easily computed by putting first

$$\hat{x}_{p+1} = \ldots = \hat{x}_n = 0$$

and then performing one backward sweep through the rows of B to resolve $\hat{x}_p$ through $\hat{x}_1$ from $B_p\hat{\mathbf{x}} = \mathbf{z}$.

To save the expensive matrix-vector multiplications with A in (4.16) we compute them only once in the initial Gram–Schmidt process and store them in a matrix W_{aa},

$$W_{aa} = A^T A W_a = (A^T A \mathbf{w}_{p+1}, \ldots, A^T A \mathbf{w}_n),$$

and then rewrite (4.16) as

$$B_A^\dagger \mathbf{z} = (I - W_a W_{aa}^T)\hat{\mathbf{x}}. \tag{4.17}$$

In a similar fashion it is easy to derive formulas for the matrix-vector product $(B_A^\dagger)^T \mathbf{z}$, cf. [45]. The resulting algorithms for computing the two matrix-vector products are as follows, where the matrix inversions are implemented as backsubstitutions and therefore involve only $O(n)$ operations:

Algorithm 4.2. (Compute $\mathbf{y} = B_A^\dagger \mathbf{z}$; B has the form (4.10))

$$-\hat{\mathbf{x}} = \begin{pmatrix} B_p^{-1}\mathbf{z} \\ 0 \end{pmatrix}$$

$$-\mathbf{y} = \hat{\mathbf{x}} - W_a W_{aa}^T \hat{\mathbf{x}}$$

Algorithm 4.3. (Compute $\mathbf{y} = (B_A^\dagger)^T \mathbf{z}$; B has the form (4.10))

$$-\hat{\mathbf{z}} = \mathbf{z} - W_{aa} W_a^T \mathbf{z}$$
$$-\mathbf{y} = B_p^{-T}\hat{\mathbf{z}}$$

It remains to compute $\mathbf{x}_*$, which can also be done at the very beginning in $O(n)$ operations by means of the relation

$$\mathbf{x}_* = W_a(AW_a)^T \mathbf{b}. \tag{4.18}$$

We verify (4.18) by expanding $\mathbf{b}$ with respect to the orthonormal basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ of $\mathbb{R}^m$, i.e., $\mathbf{b} = \sum_{i=1}^m \mathbf{u}_i^T \mathbf{b}\, \mathbf{u}_i$. Recall from (4.2) that $A\mathbf{w}_i = \mathbf{u}_i$ for $p+1 \le i \le n$. Therefore,

$$W_a(AW_a)^T \mathbf{b} = \sum_{i=p+1}^{m} \mathbf{u}_i^T \mathbf{b}\, \mathbf{w}_i$$

which is $\mathbf{x}_*$ according to (4.8).

When Algorithm 4.3 is used to compute $\bar{A} = ((B_A^\dagger)^T A^T)^T$ in (4.13), then the standard-form transformation requires only $O(n^2)$ operations due to the structure of B. As we shall see in Sect. 5, Tikhonov's method when applied to standard-form problems requires $O(n^3)$ operations. Hence, the amount of work in the reduction to standard-form is negligible, compared to the other work in the Tikhonov regularization process.

For the iterative methods to be discussed in Sect. 6, the matrix $\bar{A}$ is never explicitly required—only matrix-vector products involving $\bar{A}^T\bar{A}$ are required. In this situation, the reduction to standard form manifests itself in the matrix-vector multiplications with $B_A^\dagger$ and $(B_A^\dagger)^T$, which require only $O(n)$ operations when implemented via Algorithms 4.2 and 4.3. The multiplications with A, on the other hand, require more work, typically $O(n^2)$ operations. Hence, also for the iterative methods the work involved in the reduction to standard form is a negligible fraction of the total amount of work, at least when A has no particular structure.

A nice property of the matrix $\bar{A}$ is that its SVD is strongly related to the GSVD of (A, B), which follows immediately from its definition. For example, the singular values of $\bar{A}$ are exactly the generalized singular values of (A, B). An immediate consequence of this SVD-GSVD relation is the following norm relation:

$$\|\bar{\mathbf{x}}_\lambda\|_2 = \|B\mathbf{x}_\lambda\|_2, \quad \|\bar{\mathbf{b}} - \bar{A}\bar{\mathbf{x}}_\lambda\|_2 = \|\mathbf{b} - A\mathbf{x}_\lambda\|_2. \tag{4.19}$$

Eldén [24] defined a related algorithm for transforming a general-form problem into one in standard form. This transformation is closely connected to the one described above, but its implementation is somewhat different as it is based on the Moore–Penrose pseudoinverse $B^\dagger$ of B instead of $B_A^\dagger$.

Nevertheless, similar techniques apply as described above (cf. [10]), and then it is only the backtransformation which is slightly more complicated than (4.14). Use of this alternative transformation is recommended when, for some reason, the structure of B is essentially different from (4.10); in this case one may prefer to take advantage of standard software for operating with the Moore–Penrose inverse of B. As it is this alternative transformation to standard form which is usually described in the literature, we omit further details and refer instead to [11, 24].

5 Tikhonov regularization

In this section we take a closer look at the numerical implementation of Tikhonov regularization for discrete ill-posed problems, and estimate the work and storage

requirements for this algorithm; the regularizing properties of Tikhonov's method have been analyzed in Sects. 3.2 and 4.1. We shall also briefly treat iterated Tikhonov regularization.

A considerable part of this chapter is devoted to methods for choosing appropriate regularization parameters, namely, the discrepancy principle and its modification due to Gfrerer and Raus, as well as the quasi-optimality criterion, the L-curve criterion and generalized cross-validation.

5.1 Implementation of Tikhonov regularization

The common approach to solving the general-form problem

$$(A^TA + \alpha B^TB)\mathbf{x}_\alpha = A^T\mathbf{b} \tag{5.1}$$

numerically is to treat it as a least squares problem of the form

$$\min \left\| \begin{pmatrix} A \\ \sqrt{\alpha}B \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\| \tag{5.2}$$

(this is easily proved by forming the normal equations for (5.2)). However, naively applying any orthogonal transformation, such as QR factorization or modified Gram–Schmidt orthonormalization, directly to the system (5.2) will lead to inefficiency because this will not take advantage of the sparsity of B. Instead, the special structure of the problem must be taken into consideration, and so should also the fact that in general we wish to solve (5.2) several times for various regularization parameters α in order to determine the optimal parameter.

The key to efficiency is to reduce the problem in (5.2) to an even sparser problem with a bidiagonal matrix in place of A on "top", and a diagonal matrix in place of B at the "bottom". To produce the "bottom" diagonal matrix αI in the general case where $B \neq I$ we transform the problem into one in standard form by means of the techniques described in the previous chapter. As we have shown in Sect. 4.3, we easily obtain $\mathbf{x}_\alpha$ from the solution $\bar{\mathbf{x}}_\alpha$ of

$$\min \left\| \begin{pmatrix} \bar{A} \\ \sqrt{\alpha}I \end{pmatrix} \bar{\mathbf{x}} - \begin{pmatrix} \bar{\mathbf{b}} \\ \mathbf{0} \end{pmatrix} \right\|. \tag{5.3}$$

As already mentioned previously, the standard-form transformation only involves $O(n^2)$ operations.

The matrix $\bar{A}$ can then be modified by orthogonal transformations from the left and from the right to bring it into more amenable form (note that orthogonal transformations do not alter the Euclidean norm). Following [24], we transform $\bar{A}$ into an upper bidiagonal matrix J,

$$\bar{A} = \bar{U} \begin{pmatrix} J \\ 0 \end{pmatrix} \bar{V}^T \tag{5.4}$$

where

$$\bar{U} \in \mathbb{R}^{m \times m}, \quad J \in \mathbb{R}^{p \times p}, \quad \bar{V} \in \mathbb{R}^{p \times p},$$

and $\bar{U}$ and $\bar{V}$ are orthogonal matrices. While J is computed explicitly, $\bar{U}$ and $\bar{V}$ are represented by series of orthogonal transformations, such as Householder or Givens transformations (cf. [36: sect. 5.4.3]), which are stored in appropriate arrays and later used when multiplications with $\bar{U}$ and $\bar{V}$ are needed. This step is the one that dominates the computations: the bidiagonalization of $\bar{A}$ requires $4/3\,n^3$ multiplications. We mention that this step of the algorithm coincides with the first part of the SVD algorithm of Golub and Kahan [35], and its details can be found in many books on numerical linear algebra, e.g., [11, 36].

The transformed problem now involves the new right-hand side $\mathbf{c}$ given by

$$\mathbf{c} = \bar{U}^T\bar{\mathbf{b}}, \quad \mathbf{c} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} \begin{matrix} \} p \\ \} m - p \end{matrix},$$

and requires the following transformation of variables

$$\bar{\mathbf{x}} = \bar{V}\xi, \quad \xi = \bar{V}^T\bar{\mathbf{x}}.$$

Note that all the above transformations are independent of the actual value of α, and are therefore computed once and for all.

For each α needed in the process of determining the optimal regularization parameter, we then only need to consider the following sparese linear least squares problem

$$\min \left\| \begin{pmatrix} J \\ \sqrt{\alpha}I \end{pmatrix} \xi - \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{0} \end{pmatrix} \right\|. \tag{5.5}$$

As demonstrated by Eldén [24], (5.5) can now be solved very efficiently with only $O(n)$ operations; *only these $O(n)$ operations depend on the actual value of the regularization parameter.*

The key idea is to compute the following QR factorization by means of $2p - 1$ Givens rotations

$$\begin{pmatrix} J \\ \sqrt{\alpha}I \end{pmatrix} = Q_\alpha \begin{pmatrix} T_\alpha \\ 0 \end{pmatrix}, \tag{5.6}$$

where T_α is an upper bidiagonal $p \times p$ matrix, and $Q_\alpha \in \mathbb{R}^{2p \times 2p}$ denotes the accumulated product of the Givens rotations. From (5.6) we can then recapture $\bar{\mathbf{x}}_\alpha$ as

$$\bar{\mathbf{x}}_\alpha = \bar{V}\xi_\alpha, \quad \xi_\alpha = (T_\alpha^{-1} \quad 0)Q_\alpha^T \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{0} \end{pmatrix}. \tag{5.7}$$

The storage requirements of the implementation (5.7) are as follows. If A is not needed, then $\bar{A}$ can overwrite A in the first step (5.3), and computing then $\bar{A} = AB_A^\dagger$ requires no extra storage since $B_A^\dagger$ is never computed explicitly. If the orthogonal transformations that make up $\bar{U}$ are applied to the vector $\bar{\mathbf{b}}$ "on the fly", then they need not be stored either (unless one wants to solve a new problem with a different right-hand side later). As can be seen from (5.7), only $\bar{V}$ is required at the very end of the algorithm to compute $\bar{\mathbf{x}}$ from ξ. The orthogonal transformations that make up $\bar{V}$, however, can overwrite $\bar{A}$. Summarizing, only $O(n)$ extra storage locations are required for storing J, T_α and the Givens transformations that make up Q_α.

Algorithm 5.1. (Tikhonov's method)

— Use Algorithms 4.1 and 4.3 to transform A and B to the standard-form coefficient matrix $\bar{A}$.
— Evaluate $\mathbf{x}_*$ of (4.8) and put $\bar{\mathbf{b}} = \mathbf{b} - A\mathbf{x}_*$.
— Compute the bidiagonalization (5.4) of $\bar{A}$.
— Select α and compute the QR factorization (5.6).
— Determine ξ_α from (5.7).
— Use some parameter choice rule to improve α (see Sects. 5.3 and 5.4 below), and go back to Step 4, if necessary.
— Finally, compute $\bar{\mathbf{x}}_\alpha = \bar{V}\xi_\alpha$, and determine

$$\mathbf{x}_\alpha = B_A^\dagger \bar{\mathbf{x}}_\alpha + \mathbf{x}_*$$

by means of Algorithm 4.2.

We stress that all methods for choosing α that we will present below require only $O(n)$ operations per trial whereas the final computation of $\bar{x}_\alpha$ takes $O(n^2)$ operations. This is so because the parameter search algorithms below can all be based on the following identities:

$$\|B\mathbf{x}_\alpha\| = \|\bar{\mathbf{x}}_\alpha\| = \|\xi_\alpha\|, \tag{5.8}$$

$$\|\mathbf{b} - A\mathbf{x}_\alpha\|^2 = \|\bar{\mathbf{b}} - \bar{A}\bar{\mathbf{x}}_\alpha\|^2 = \|\mathbf{c}_1 - J\xi_\alpha\|^2 + \|\mathbf{c}_2\|^2. \tag{5.9}$$

We mention at this point that for Tikhonov regularization it is very difficult to explicitly employ any particular structure of A or $\bar{A}$ (such as Hankel or Toeplitz). One exception that we are aware of arises from Volterra integral equations with stationary kernel, in which case appropriate discretization may give an upper triangular Toeplitz matrix A. This particular structure can be used to design an algorithm that requires only $O(n^2)$ operations in total for solving (5.2), cf. [26].

The difficulty with more general structured matrices is that any reduction along the lines considered above destroys the structure of the matrix; to really exploit such structures one will rather try different ideas starting directly from (5.2), or from the corresponding normal equations system (5.1). For the important special case where A and B are both Toeplitz matrices, Chun and Kailath [17] and Nagy [77] derived such algorithms that require only $O(n^2)$ operations to compute the regularized approximation $\mathbf{x}_\alpha$ for a particular value of α.

One way to exploit a potential sparsity structure in A is based on a much older idea: as the coefficient matrix of the damped normal equations system (5.1) is positive definite, one may use conjugate gradients to iteratively approximate the solution of (5.1). The rate of convergence, however, will usually be slow, as the condition number of this system will be of the order of $O(1/\alpha)$ which may be expected large. The efficiency of such an approach therefore depends on the availability of good preconditioners. For example, very efficient preconditioners have been suggested in [14] for the case where both A and B have a Toeplitz block structure with Toeplitz blocks; such systems arise naturally from higher dimensional convolution problems.

The shortcoming with all these alternative approaches is that the entire algorithm must be restarted once a new value of α has been chosen, and therefore the total amount of work is difficult to estimate.

5.2 Iterated Tikhonov regularization

Even with regularization, the exact solution of an ill-posed problem can only be reconstructed with limited accuracy in the presence of noisy data. From the analysis in [39: p. 42] we conclude that Tikhonov regularization can reach the optimal accuracy that the data allow, only if the exact solution is not a very smooth function.

In those cases where the Tikhonov approximation fails to have optimal accuracy, it is possible to improve it with the following *defect iteration* (e.g., [93]): compute

$$\mathbf{d}_\alpha = \mathbf{b} - A\mathbf{x}_\alpha,$$

and determine a new approximation $\mathbf{x}_\alpha^{\mathrm{II}}$ from

$$\bar{\mathbf{x}}_\alpha^{\mathrm{II}} = \bar{\mathbf{x}}_\alpha + \Delta\bar{\mathbf{x}}_\alpha, \tag{5.10}$$

where $\Delta\bar{\mathbf{x}}_\alpha$ solves the Tikhonov equation (5.3), but with $\mathbf{d}_\alpha$ instead of $\mathbf{b}$ in the right-hand side, or equivalently,

$$(\bar{A}^T\bar{A} + \alpha I)\Delta\bar{\mathbf{x}}_\alpha = \bar{A}^T\mathbf{d}_\alpha. \tag{5.11}$$

To obtain $\bar{\mathbf{x}}_\alpha^{\mathrm{II}}$ we simply replace $\mathbf{c}_1$ by $\mathbf{c}_1 - J\xi_\alpha$ in (5.7); note that this takes only $O(n)$ operations as $\bar{\mathbf{x}}_\alpha$ is not required explicitly.

Of course, there is no reason why one should stop this process with $\mathbf{x}_\alpha^{\mathrm{II}}$, and one may instead proceed to compute approximations $\mathbf{x}_\alpha^{\mathrm{III}}$, $\mathbf{x}_\alpha^{\mathrm{IV}}$, and so forth, in a similar fashion (even more general modifications are possible, e.g., [94]). For obvious reasons the resulting method is usually referred to as *iterated Tikhonov regularization*. Its most important property is that the set of solutions, which can be reconstructed with optimal accuracy, increases with every iteration, cf. [61]. In other words, if defect iteration has been repeated sufficiently many times, the resulting reconstruction will have reached an accuracy that cannot be improved significantly by any other method.

5.3 The choice of α: $\|\varepsilon\|$ known

While the implementation of Tikhonov's method only requires the use of standard techniques from numerical linear algebra, the user's skill is involved in the choice of the regularization parameter α.

We emphasize that no black-box procedures for choosing α are available yet, and most likely will never exist. Instead, there exist numerous heuristics for choosing α and we shall discuss some of them. We concentrate on those which make use of as little information as possible in order to provide as much generality as possible. Of course, the more information about the problem or the solution is

available, the better heuristics should exist. If only little information is at hand, we recommend to try different parameter choice strategies and then study the reconstructions on-line, if possible. In our presentation of parameter choice rules we will focus on *regularization problems in standard form* only, for this simplifies the notations enormously. As we have to reduce general-form problems to standard form anyway, this is no loss of generality. In case of a problem in general form we can observe from the identities (4.19) that in this way we base our parameter choice on the (semi-) norm $\|B\cdot\|$ in $\mathbb{R}^n$.

In this section we discuss two methods that rely on a fairly accurate estimate of $\|\varepsilon\|$, the Euclidean norm of the errors in the right-hand side $\mathbf{b}$. These methods are the well-known discrepancy principle (e.g., [39]), and a more recent method due to Raus [92] and Gfrerer [31]. Whenever such information is available, we highly recommended that it be used for choosing α, since it can be shown that under appropriate conditions on the exact solution, the aforementioned parameter choice rules will lead to essentially optimal values of α [39, 31]. On the other hand, if only a rough estimate of $\|\varepsilon\|$ is known, or if it is completely unknown, then one should rather use one of the alternative methods described in the next section.

The *discrepancy principle* goes back to Morozov [75], and it is used frequently in many applications— although we stress that it is not guaranteed to be as good as the Gfrerer/Raus-method described below. We include its discussion here mainly for historical reasons and for completeness.

The simple idea in the discrepancy principle is that one should choose the regularization parameter α such that the norm of the defect, $\|\mathbf{b} - A\mathbf{x}_\alpha\|$, equals the norm of the error term:

$$\phi_{\mathrm{M}}(\alpha) \equiv \|\mathbf{b} - A\mathbf{x}_\alpha\|^2 = \|\varepsilon\|^2. \tag{5.12}$$

This is a nonlinear equation for $\alpha = \alpha_{\mathrm{M}}$, and it can be solved by, e.g., Newton's method. It is not difficult to show, see e.g., [76: p. 227], that ϕ_{M} is a convex, increasing function of α; therefore, Newton's method for solving (5.12) will converge monotonically for every initial guess α_0 with $0 < \alpha_0 \le \alpha_{\mathrm{M}}$. However, to enhance stability we would prefer to employ only regularization parameters $\alpha \ge \alpha_{\mathrm{M}}$. We will show below, in connection with the Gfrerer/Raus-method, how this can be achieved.

The Gfrerer/Raus-method [31, 92] may be seen as an improved variant of the discrepancy principle. We shall briefly outline the underlying idea as developed by Gfrerer [31]. Let $\mathbf{x}^{\mathrm{exact}}$ denote the exact, unperturbed solution, and let $\mathbf{b}^{\mathrm{exact}} = A\mathbf{x}^{\mathrm{exact}}$ denote the corresponding right-hand side. Moreover, define the unperturbed regularized solution as

$$\mathbf{x}_\alpha^{\mathrm{exact}} = (A^TA + \alpha I)^{-1}A^T\mathbf{b}^{\mathrm{exact}}.$$

Then a standard error estimate (e.g., [39: Lemma 2.3.2]) yields

$$\begin{aligned}\|\mathbf{x}^{\mathrm{exact}} - \mathbf{x}_\alpha\|^2 &\le 2(\|\mathbf{x}_\alpha^{\mathrm{exact}} - \mathbf{x}_\alpha\|^2 + \|\mathbf{x}^{\mathrm{exact}} - \mathbf{x}_\alpha^{\mathrm{exact}}\|^2)\\ &\le 2\left(\tau\frac{\|\varepsilon\|^2}{\alpha} + \|\mathbf{x}^{\mathrm{exact}} - \mathbf{x}_\alpha^{\mathrm{exact}}\|^2\right)\end{aligned} \tag{5.13}$$

for any $\tau \ge 1$. Using

$$\frac{d}{d\alpha}\|\mathbf{x}^{\mathrm{exact}} - \mathbf{x}_\alpha^{\mathrm{exact}}\|^2 = 2\alpha(\mathbf{b}^{\mathrm{exact}})^T(AA^T + \alpha I)^{-3}\mathbf{b}^{\mathrm{exact}}, \tag{5.14}$$

we can then find the α which minimizes (5.13) as the solution to the nonlinear equation

$$2\alpha^3(\mathbf{b}^{\mathrm{exact}})^T(AA^T + \alpha I)^{-3}\mathbf{b}^{\mathrm{exact}} = \tau\|\varepsilon\|^2. \tag{5.15}$$

Gfrerer proceeds by inserting $\mathbf{b}$ for $\mathbf{b}^{\mathrm{exact}}$ in (5.15) and then solving for $\alpha = \alpha_{\mathrm{GR}}$. The resulting equation has a unique solution provided $\tau \ge 2$; in fact, Gfrerer recommends the choice $\tau = 2$, and hence obtains α_{GR} as the unique solution of

$$\phi_{\mathrm{GR}}(\alpha) \equiv \alpha^3\mathbf{b}^T(AA^T + \alpha I)^{-3}\mathbf{b} = \|\varepsilon\|^2. \tag{5.16}$$

It is interesting to note the following alternative expression, where $\mathbf{x}_\alpha^{\mathrm{II}}$ denotes the iterated Tikhonov approximation of (5.10):

$$\phi_{\mathrm{GR}}(\alpha) = (\mathbf{b} - A\mathbf{x}_\alpha)^T(\mathbf{b} - A\mathbf{x}_\alpha^{\mathrm{II}});$$

in particular, we emphasize the similarity to the discrepancy principle (5.12). The Gfrerer/Raus-method has been developed in the infinite dimensional context; King and Neubauer [62] slightly modified (5.16) for finite-dimensional approximations.

Newton's method applied to (5.16) will only converge locally, but global monotone convergence is guaranteed when Newton's iteration is applied instead to the function

$$\psi_{\mathrm{GR}}(\beta) = \phi_{\mathrm{GR}}\left(\frac{1}{\beta}\right) = \|\varepsilon\|^2, \quad \beta = \frac{1}{\alpha},$$

starting with $\alpha_0 \ge \alpha_{\mathrm{GR}}$ [76: p. 227]. This is particularly attractive because it allows the use of larger regularization parameters (the same trick applies to the discrepancy principle as well). Rewriting Newton's scheme in terms of the regularization parameter we obtain:

$$\alpha_{k+1} = \alpha_k - \alpha_k\frac{\phi_{\mathrm{GR}}(\alpha_k) - \|\varepsilon\|^2}{\alpha_k\phi'_{\mathrm{GR}}(\alpha_k) + \phi_{\mathrm{GR}}(\alpha_k) - \|\varepsilon\|^2}, \tag{5.17}$$

where

$$\phi'_{\mathrm{GR}}(\alpha) = 3\alpha^2\|(A^TA + \alpha I)^{-2}A^T\mathbf{b}\|^2 = 3\alpha^2\|(A^TA + \alpha I)^{-1}\mathbf{x}_\alpha\|^2.$$

We now turn to an efficient evaluation of (5.17). The key idea is the use of the identity

$$\alpha(AA^T + \alpha I)^{-1} = I - A(A^TA + \alpha I)^{-1}A^T \tag{5.18}$$

for the evaluation of ϕ_{GR} of (5.16). We omit the details here and summarize the final algorithm as follows:

Algorithm 5.2. (Gfrerer/Raus-method)

— Given α, compute the solution ξ_α of (5.5) with Algorithm 5.1, and put

$$z_\alpha = c_1 - J\xi_\alpha,$$

$$u_\alpha = z_\alpha - J(T_\alpha^{-1} \quad 0)Q_\alpha^T \begin{pmatrix} z_\alpha \\ 0 \end{pmatrix},$$

$$y_\alpha = J^T u_\alpha.$$

— Then, the following holds:

$$\phi_{GR}(\alpha) = z_\alpha^T u_\alpha + \|c_2\|^2,$$

$$\phi'_{GR}(\alpha) = 3\|y_\alpha\|^2/\alpha^2. \tag{5.19}$$

— If $\phi_{GR}(\alpha)$ is sufficiently close to $\|\varepsilon\|^2$, then stop; otherwise determine the next α from (5.17).

Note that Algorithm 5.2 requires only $O(n)$ operations.

5.4 The choice of α: $\|\varepsilon\|$ unknown

It is obvious that the performance of the algorithms in the previous section depends on the accuracy of the estimate for $\|\varepsilon\|$. If no good estimate is available, it will be better to use an ε-free parameter choice strategy. Several such strategies have been suggested, in the following we will discuss three of them, restricting again our attention to problems in standard form only.

The quasi-optimality criterion

The quasi-optimality criterion is based on the hypothesis of a plateau of $\|x_\alpha - x^{exact}\|$ near the optimal regularization parameter, and therefore $\alpha = \alpha_q$ is chosen so as to minimize

$$\phi_q(\alpha) = \left\| \alpha \frac{d}{d\alpha} x_\alpha \right\|^2, \quad \alpha > 0. \tag{5.20}$$

This method originates with Tikhonov and Glasko [104] in a slightly different form, and has been considered by numerous authors thereafter, especially in the Russian literature, e.g., [103: sect. II.6; 76: sect. 27].

It is easily verified that

$$\alpha \frac{d}{d\alpha} x_\alpha = x_\alpha - x_\alpha^{II},$$

where as in Sect. 5.2 x_α^{II} denotes the iterated Tikhonov approximation. Therefore, x_α^{II} is presumably a better reconstruction of the exact solution x^{exact}, and one is led to use a standard heuristic of numerical analysis,

$$x_\alpha - x^{exact} \approx x_\alpha - x_\alpha^{II},$$

hence we obtain another motivation for minimizing (5.20).

The function $\phi_q(\alpha)$ of (5.20) can be evaluated by means of the formula

$$\alpha \frac{d}{d\alpha} x_\alpha = -\alpha(A^TA + \alpha I)^{-1} x_\alpha.$$

Consequently, there is an interesting relationship between the Gfrerer/Raus-method, and the quasi-optimality criterion as

$$\phi_q(\alpha) = \frac{1}{3}\phi'_{GR}(\alpha).$$

In particular, we can make use of (5.19) in Algorithm 5.2 to evaluate (5.20).

When the quasi-optimality criterion is employed in finite dimensions (as we suggest it here), one has to be aware of the following source of potential confusion: the minimum of (5.20) is always attained at $\alpha = 0$ since $\phi_q(0) = 0$. Therefore, it is important to understand that this is an artificial feature which entirely stems from discretization; in the continuous setting (for which the method was originally designed) $\phi_q(\alpha)$ will diverge to infinity as $\alpha \to 0$ under quite general assumptions on the stochastic nature of the noise component in the right-hand side, cf. [105]. As a matter of fact, in finite dimensions $\phi_q(\alpha)$ has a large maximum in close vicinity of $\alpha = 0$. It is to the right of this maximum that we have to search for the minimum of ϕ_q.

Usually, the global minimum at $\alpha = 0$ will not cause any trouble at all as the aforementioned maximum will serve as a natural barrier for standard minimization routines when applied to minimizing (5.20). The user only has to take care of choosing the initial guess for α_q sufficiently large.

Generalized cross-validation

The parameter choice rule that is probably most widely used in practice is due to Wahba [115] (see also [34]): generalized cross-validation (GCV). The philosophy behind it comes from statistical estimation theory; a rigorous treatment can be found in [117]. Let α be fixed for the moment, and assume that we try to estimate one component b_{i_*} of the right-hand side vector $b = (b_1, \ldots, b_m)^T$ from the remaining $m - 1$ components in the following way:

— We first apply Tikhonov regularization with the chosen α to the modified system $A'x = b'$, which is obtained from $Ax = b$ by deleting the i_*th equation; let x_{α,i_*} denote the resulting approximation.
— Then, we use x_{α,i_*} to estimate $b_{i_*} \approx (Ax_{\alpha,i_*})_{i_*}$.

It can be expected that a good choice of α is one, for which the error of the above estimation—averaged with reasonable weights w_{i_*} over all possible values of $i_* \in \{1, \ldots, m\}$—becomes small. This is the basis for GCV, where α is chosen to be the minimizer of the function

$$\mathcal{V}(\alpha) = \frac{\|b - Ax_\alpha\|}{\operatorname{trace}[I - A(A^TA + \alpha I)^{-1}A^T]} = \frac{\|AA^T + \alpha I)^{-1} b\|}{\operatorname{trace}[(AA^T + \alpha I)^{-1}]}. \tag{5.21}$$

As shown in [34], the above functional results from a particular set of weights w_{i_*},

for which the values of $\mathcal{V}$ are independent of orthogonal transformations of the image space (including permutations of the data as a special case).

A particularly attractive feature of GCV is the availability of convergence results, e.g., by Wahba [115] and Lukas [71]: under not too restrictive assumptions the expected value of the optimal regularization parameter, and the expected value of the GCV-parameter, asymptotically coincide as the number of measured data goes to infinity. Lukas also considers the behavior of the expected error of the regularized solutions.

The disadvantage of the method, however, is a common observation, according to which the graph of the GCV functional $\mathcal{V}$ is very flat with many tiny wiggles near the optimal α which complicates a numerical black box determination of its global minimum. We stress that black box procedures for finding the appropriate minimum of (5.20) for the quasi-optimality criterion will hardly exist either, but the minimum of (5.20) is usually much easier to locate visually.

Eldén [27] has shown how to evaluate $\mathcal{V}(\alpha)$ efficiently on the basis of the transformations (5.4)–(5.7) of Sect. 5.1. Following [27] we observe from (5.18) that

$$\operatorname{trace}[I - A(A^TA + \alpha I)^{-1}A^T] = m - n + \alpha \operatorname{trace}[(T_\alpha^T T_\alpha)^{-1}]. \qquad (5.22)$$

The latter can be determined recursively: let

$$T_\alpha = \begin{pmatrix} t_{1,1} & t_{1,2} & & & 0 \\ & t_{2,2} & t_{2,3} & & \\ & & \ddots & \ddots & \\ & & & t_{n-1,n-1} & t_{n-1,n} \\ 0 & & & & t_{n,n} \end{pmatrix},$$

and denote by s_i^2, the squared norm of the ith row of T_α^{-1} for $i = 1, \ldots, n$. The quantities s_i^2 can be computed as follows:

$$s_n^2 = 1/t_{n,n}^2,$$
$$s_i^2 = (1 + t_{i,i+1}^2 s_{i+1}^2)/t_{i,i}^2, \quad i = n-1, n-2, \ldots, 1.$$

Then we can substitute

$$\operatorname{trace}[(T_\alpha^T T_\alpha)^{-1}] = s_1^2 + s_2^2 + \ldots + s_n^2$$

in (5.22), hence we obtain the denominator of (5.21). The numerator of (5.21) can be determined from (5.9). In this way the overall evaluation of $\mathcal{V}(\alpha)$ takes only $O(n)$ extra operations.

The L-curve criterion

The L-curve criterion differs from the previous parameter choice strategies in that it is rooted in a more heuristic approach, and no convergence results are available yet.

The L-curve criterion is based on a graph of the penalty term $\|\mathbf{x}_\alpha\|$ versus the residual norm $\|\mathbf{b} - A\mathbf{x}_\alpha\|$. Obviously, a convenient way to visualize the compromise between minimizing the penalty term and the residual norm, as a function of the regularization parameter α, is to plot $\|\mathbf{x}_\alpha\|$ versus $\|\mathbf{b} - A\mathbf{x}_\alpha\|$. The use of such plots goes back to Miller [74] and Lawson and Hanson [65], while the term "L-curve" for this plot was introduced much later in [49, 51], where the properties of the L-curve are studied in details. It is easy to show that $\|\mathbf{x}_\alpha\|$ is a monotonically decreasing function of $\|\mathbf{b} - A\mathbf{x}_\alpha\|$. More important, however, is the fact that when plotted in *log–log scale*, the L-curve will have a characteristic L-shaped appearance with a steep part, a flat part, and a distinct "corner" separating these two parts—hence its name. The only assumptions that are needed to show this, cf. [49], is that the unperturbed component of the right-hand side satisfies the discrete Picard condition and that the perturbation does not dominate the right-hand side. The flat part then corresponds to a solution $\mathbf{x}_\alpha$ that is dominated by regularization errors, i.e., the regularization parameter α is chosen too large and not all the available information in $\mathbf{b}$ is extracted. Moreover, the plateau of this part of the L-curve is at $\|\mathbf{x}_\alpha\| \approx \|\mathbf{x}^{\text{exact}}\|$. The vertical part, on the other hand, corresponds to a solution that is dominated by perturbation errors. The actual slope of this part of the curve depends only on the singular values of A, and the curve becomes almost vertical in the vicinity of $\alpha = 0$ where $\|\mathbf{b} - A\mathbf{x}_\alpha\|$ approaches its limit. The use of the log–log scale for plotting the L-curve is essential for distinguishing the two different parts of the curve, for then the slope of the L-curve is the relative change in $\|\mathbf{x}_\alpha\|$ divided by the relative change in $\|\mathbf{b} - A\mathbf{x}_\alpha\|$, and this ratio depends—roughly speaking—on which spectral component of $\mathbf{b} = \mathbf{b}^{\text{exact}} + \varepsilon$ dominates $\mathbf{x}_\alpha$. In lin-lin scale, the slope is the ratio between the absolute changes in $\|\mathbf{x}_\alpha\|$ and $\|\mathbf{b} - A\mathbf{x}_\alpha\|$ which is practically independent of $\mathbf{b}$, hence the lin-lin scale cannot be used to distinguish signal from noise. For more details, cf. [51].

According to the above reasoning, the corner of the L-curve which separates the vertical and the horizontal parts must correspond to a good balance between minimizing the regularization error and the perturbation error in the solution $\mathbf{x}_\alpha$. The key problem in the L-curve criterion for choosing α is therefore to seek a mathematical definition of the L-curve's corner and to use this as the criterion for choosing α.

The definition of the L-curve's corner that is suggested in [51] is the point on the L-curve—plotted in log–log scale—with maximum curvature. Let $\rho = \log\|\mathbf{b} - A\mathbf{x}_\alpha\|$ and $\eta = \log\|\mathbf{x}_\alpha\|$: then the curvature, as a function of the parameter α, is defined by

$$\kappa(\alpha) = \frac{\rho'\eta'' - \rho''\eta'}{((\rho')^2 + (\eta')^2)^{3/2}},$$

where $'$ denotes differentiation with respect to α. Any one-dimensional optimization routine can be used to compute α from this definition.

Although a rigorous analysis of the L-curve criterion is still lacking, we nevertheless have good experience with this method. In particular, numerical experiments in [49, 51] demonstrate that the L-curve criterion is as robust as GCV.

6 Iterative methods

When it comes to large-scale applications, a striking disadvantage of Tikhonov regularization becomes obvious: the amount of work required to compute the

regularized approximation is of the order $O(n^3)$ where n is the number of free parameters, i.e., the dimension of the subspace in which we are searching for an approximation of $\mathbf{x}^{\text{exact}}$.

Nowadays, as more and more people consider 2-D and 3-D problems like those of Sect. 2 that usually give rise to large structured or sparse coefficients matrices A, the above algorithms for Tikhonov regularization become impractical because they destroy the structure or sparsity of the matrix. Consequently, many engineers turned their attention to iterative alternatives that only require matrix-vector multiplications (which can be implemented very efficiently and which do not alter the matrix A). It was on the basis of their numerical experiences that one came to understand that iterative methods may also be used for regularization.

Below, we discuss how iterative methods may be employed to solve discrete regularization problems in general form (where in (4.1) the matrix $B \neq I$); the regularizing properties of the iterative methods are derived in terms of filtered GSVD-expansions of the iterates, similar to the exapnsion (4.7) for the Tikhonov approximations. It will become clear that the number of iterations plays the role of the regularization parameter.

We will then discuss the specific advantages of iterative algorithms in general; especially, we elaborate briefly on the case where A is a Toeplitz matrix. The efficiency of iterative regularization methods mainly depends on the number of iterations that are required: in this sense the conjugate gradient method and the ν-methods are optimal in that they regularize faster than other iterative methods proposed in the literature. These two methods are often more efficient (and sometimes more accurate) than Tikhonov's method.

Finally, we will treat stopping rules and also discuss their efficient implementation.

6.1 Preliminaries

To introduce some general ideas, consider for the moment the following basic iteration (which is associated with so many names in the different areas of applied mathematics) applied straight to the integral equation (3.1):

$$x_k = x_{k-1} + \omega K^*(g - Kx_{k-1}), \quad 0 < \omega < 2/\|K^*K\|, \quad k = 1, 2, \ldots \tag{6.1}$$

In 1951, Landweber [64] showed that $x_k \to K^{-1}g$ whenever $g \in \mathcal{R}(K)$. On the other hand, however, $\|x_k\| \to \infty$ if $g \notin \mathcal{R}(K)$ so that a straightforward application of (6.1) seems to be impossible in practical situations (recall the discussion in Sect. 3.1).

Nevertheless, many people observed in their numerical experiments that the approximations seem to converge in the beginning of the iteration, before they become worse and finally diverge; this phenomenon is often called *semiconvergence*, e.g., [81].

Expanding the iterates in terms of the singular value expansion provides an explanation for this phenomenon. Here, we replace K and g in (6.1) as in Sect. 4.3 by the discrete quantities $\bar{A} = AB_A^\dagger$ and $\bar{\mathbf{b}} = \mathbf{b} - A\mathbf{x}_*$ of the standard form equation (4.11) with $\mathbf{x}_*$ defined according to (4.8). We then consider the following discrete analog of (6.1), starting with $\bar{\mathbf{x}}_0 = 0$,

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}}_{k-1} + \omega\bar{A}^T(\bar{\mathbf{b}} - \bar{A}\bar{\mathbf{x}}_{k-1}), \quad k = 1, 2, \ldots \tag{6.2}$$

As in (4.14) we then get the approximation $\mathbf{x}_k$ after backtransformation, i.e.,

$$\mathbf{x}_k = B_A^\dagger \bar{\mathbf{x}}_k + \mathbf{x}_*.$$

Note that if we are given a discrete regularization problem in standard form ($B = I$) then all "bars" in (6.2) may simply be omitted. In any case, using the GSVD, we easily obtain by induction a representation similar to (4.7)

$$\mathbf{x}_k = \sum_{i=1}^{p} \varphi_k^L(\sigma_i^2)\frac{\mathbf{u}_i^T\mathbf{b}}{\rho_i}\mathbf{w}_i + \sum_{i=p+1}^{n} \mathbf{u}_i^T\mathbf{b}\,\mathbf{w}_i, \tag{6.3}$$

where we have introduced the *Landweber filter functions*

$$\varphi_k^L(\lambda) = 1 - (1 - \omega\lambda)^k. \tag{6.4}$$

From Bernoulli's inequality we deduce

$$0 \leq \varphi_k^L(\lambda) \leq \omega k\lambda \ll 1, \quad \lambda \ll 1/k, \tag{6.5}$$

while for fixed i, $\varphi_k^L(\sigma_i^2)$ converges to 1 with a geometric rate; the smaller the i—i.e., the larger the σ_i—the faster this convergence actually is.

In other words, contributions of $\mathbf{b}$ corresponding to small singular values of $\bar{A}$ are suppressed in the beginning of the iteration; components of the solution corresponding to the dominating singular values are approximated after a few iterations only. Recall that it is the influence of the small singular values' components that leads to the failure of conventional numerical algorithms; if the iteration is terminated just before their impact increases we obtain a regularized approximation of the exact solution; in other words, *the iteration index k takes the role of the regularization parameter α in Tikhonov's method.* Comparing (6.5) with the corresponding estimate (4.9) of the Tikhonov filter functions derived at the end of Sect. 4.1, we observe that, roughly, we can treat k as equivalent to $1/\alpha$.

By means of a careful study of the filter functions (6.4) it is possible to give a rigorous proof of the regularizing properties of Landweber's method, and to show that the regularized approximations obtained in this way have at least the same quality as those obtained by Tikhonov's method, cf. [39: p. 42; 42]; for the approximation of very smooth solutions to (3.1) the accuracy of Landweber's method will be better.

We will now outline an implementation of (6.2). We emphasize that we do this for illustrative purposes only: due to the large number of iterations that are usually required by Landweber's method, its use cannot be recommended in practice. However, the iterative methods to follow in Sects. 6.3 and 6.4 are implemented in exactly the same way.

First, rewriting (6.2) in terms of $\mathbf{x}_k$ gives

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \omega B_A^\dagger (B_A^\dagger)^T A^T(\mathbf{b} - A\mathbf{x}_{k-1}), \quad k = 1, 2, \ldots \tag{6.6}$$

with $\mathbf{x}_0 = \mathbf{x}_*$ as initial guess (it should be clear that it is no problem to incorporate some other "first guess" $\mathbf{x}_{\text{guess}}$ in exactly the same way as we already did it in Sect.

3.2, namely by appropriately modifying $\mathbf{b}$). From (6.6) we observe that regularization in general form amounts to a premultiplication of the normal equations systems $A^TAx = A^Tb$ by $B_A^\dagger(B_A^\dagger)^T$; in this way the iterates x_k are mapped into a suitable subspace.

We again stress the fact that it is not necessary to form $B_A^\dagger$ and $(B_A^\dagger)^T$ explicitly. All we need are two subroutines for computing $B_A^\dagger z$ and $(B_A^\dagger)^T y$, respectively, given vectors $z \in \mathbb{R}^p$ and $y \in \mathbb{R}^n$ occurring during the iteration. Such subroutines are described in Sect. 4.3; using them, each computation of $B_A^\dagger z$ and $(B_A^\dagger)^T y$ takes $O(n)$ operations only.

6.2 Efficiency considerations

Once we accept that iterative regularization methods on one side and Tikhonov's method on the other side can essentially achieve the same accuracy in real world applications, there are two major questions that have to be considered next:

(i) how can we find the particular regularization parameter (α or k) for which the respective method reaches its optimal accuracy, and
(ii) which of the methods is the less expensive?

Concerning the first question we refer to the stopping rules in Sects. 5.3, 5.4, and 6.5; below we only consider the second of these questions.

We shall do this on the basis of the operation counts that we have obtained so far. We ignore the discretization process assuming it to be independent of the regularization method; this is actually a simplification since it may be a more efficient approach to avoid a discretization of K in iterative methods and rather discretize Kx when computing Ax.

We already emphasized in Sect. 5 that the major part of work in Tikhonov regularization is the $4/3\,n^3$ multiplications that are required in the preliminary bidiagonalization step. Besides this, the search for the optimal regularization parameter, requiring only $O(n)$ operations per trial, is negligible if we assume that only a moderate number of parameters need to be considered.

The costs for iterative regularization methods accumulate completely differently and an estimate of the total amount of work is much more difficult. In fact, the implementation of iterative methods requires only little preprocessing, but to arrive at the optimal stopping index makes it necessary to perform all intermediate iterations. Therefore, the efficiency of an iterative method essentially depends on the magnitude of the optimal regularization parameter, which in turn depends on the particular iterative scheme, and also on the signal-to-noise ratio: the less noise there is, the more iterations are necessary to find the best approximation.

If the k_*th iterate x_{k_*} is the best regularized approximation then the iterative method requires k_* matrix-vector multiplies with A and A^T, respectively (and another few for computing W_{aa} and x_*), plus $O(k_* n)$ remaining operations for linear vector combinations, norm computations, etc.

If A is a dense and unstructured matrix—a very common situation for discretizations of integral equations—then the matrix-vector products can benefit from vector or parallel processors; this will give very good speedups because of the inherent parallelism within these operations (cf. [21]).

In some applications (cf. Sect. 2.2) A is a sparse matrix. In this case A can be stored with a sparse storage scheme and in a matrix-vector multiplication the nonzero elements of A are multiplied with the respective vector components only.

Finally, in other applications (cf. Sect. 2.3) the integral equation (3.1) is a convolution-type equation. Depending on the discretization, the matrix A can then have the form $A = D_1 T D_2$ with diagonal matrices D_1 and D_2 and a Toeplitz or (in 2-D applications) block-Toeplitz matrix T. Maintaining this factored form for A we can perform the matrix-vector multiplication with A in only $O(n \log n)$ instead of $O(n^2)$ operations by using fast Fourier transform techniques. As only few textbooks on numerical linear algebra mention this fact, we include more details for the reader's convenience. Put

$$\mathbf{t} = (t_0, t_{-1}, \ldots, t_{1-m}, 0, t_{n-1}, t_{n-2}, \ldots, t_1)^T \in \mathbb{R}^{m+n},$$

and let T be the $m \times n$ Toeplitz matrix generated by $\mathbf{t}$, i.e.,

$$T = \begin{pmatrix} t_0 & t_1 & \cdots & & t_{n-1} \\ t_{-1} & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & t_1 \\ & & \ddots & t_{-1} & t_0 \\ & & & & t_{-1} \\ \vdots & & & & \\ t_{1-m} & \cdots & & & t_{n-m} \end{pmatrix}.$$

Denote further by FFT and IFFT routines for the fast Fourier transform of a vector and its inverse, respectively. Then the matrix vector multiply Tx can be obtained as follows, assuming that $\hat{\mathbf{t}} = \text{FFT}(\mathbf{t})$ has already been computed once and for all:

Algorithm 6.1. (Compute Tx with Toeplitz T)

— Compute $\hat{x} = \text{FFT}\begin{pmatrix} x \\ 0 \end{pmatrix} \begin{matrix} \}n \\ \}m \end{matrix}$
— multiply componentwise $\hat{y} = \hat{x} \cdot \hat{t}$
— Compute $y = \text{IFFT}(\hat{y}) \in \mathbb{R}^{m+n}$
— the first m components of y now contain Tx.

We refer to Van Loan [110] for a justification of the above algorithm (in some situations, depending on the actual implementation of the FFT-routines, the respective components of y and Tx may differ by a multiplicative constant). In the above algorithm we haven't used that $m \geq n$ and it can therefore as well be used for matrix vector multiplies with the transpose of T. The generalization to block-Toeplitz matrices is straightforward.

We summarize, Tikhonov's method and iterative regularization methods are difficult to compare in general as the efficiency of the iterative methods mainly depends on the implementation of the matrix vector products with A and A^T, and

the number of iterations that are required. In the worst case where A is dense and unstructured, i.e., where the matrix-vector products amount to $O(n^2)$ operations, the optimal iteration number k_* should be somewhat smaller than n to be competitive to Tikhonov's method. In special cases the matrix-vector products are cheaper, hence significantly more iterations are admissible.

6.3 The ν-methods

For most real problems the number k_* in Landweber's method is much bigger than n and Tikhonov's method is much more efficient. Although improved by several authors thereafter (e.g., [12, 98, 95]), iterative algorithms for a long time remained too slow to be competitive to Tikhonov's method. Recently, Nemirovskii and Polyak [84] and Brakhage [13] developed two-step iterative schemes that do not share this advantage: the *ν-methods*.

The *ν-methods* as introduced in [13] constitute a one-parameter family of iterative methods; $\nu > 0$ is the free parameter to be chosen in advance, for $\nu = 0.5$ the method reduces to the *Chebyshev method* of Nemirovskii and Polyak [84]. A heuristic for choosing ν in practical applications is provided below.

We suggest the following implementation of the method which roughly consists of four steps: preprocessing of general form problems, rescaling of the problem under consideration, step zero of the iteration, and the iteration itself.

Algorithm 6.2. (ν-Method)

— Compute $\mathbf{x}_0 = \mathbf{x}_*$ and put $\mathbf{d}_* = \mathbf{b} - A\mathbf{x}_*$
— Rescale A and $\mathbf{b}$ (see the discussion below)
— Initialize

$$\alpha_0 = \frac{2\nu + 1}{2\nu + 1/2}$$

$$\mathbf{d}_0 = \mathbf{d}_*$$

$$\mathbf{z}_0 = 0$$

$$\Delta\mathbf{z}_1 = \alpha_0 \mathbf{d}_0$$

$$\Delta\mathbf{x}_1 = B_A^\dagger (B_A^\dagger)^T A^T \Delta\mathbf{z}_1 .$$

— Iterate for $k = 1, 2, \ldots$ until the termination criterion is fulfilled:

$$\mathbf{z}_k = \mathbf{z}_{k-1} + \Delta\mathbf{z}_k$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$$

$$\mathbf{d}_k = \mathbf{d}_{k-1} - A\Delta\mathbf{x}_k$$

$$\beta_k = \frac{1}{4} \frac{k(k - 1/2)}{(k + \nu - 1/2)(k + \nu)}$$

$$\alpha_k = 4 \frac{(k + \nu)(k + \nu + 1/2)}{(k + 2\nu)(k + 2\nu + 1/2)}$$

$$\Delta\mathbf{z}_{k+1} = \alpha_k(\mathbf{d}_k + \beta_k \Delta\mathbf{z}_k)$$

$$\Delta\mathbf{x}_{k+1} = B_A^\dagger (B_A^\dagger)^T A^T \Delta\mathbf{z}_{k+1} .$$

As usual, $\mathbf{x}_k$ is the kth approximation to $\mathbf{x}^{\text{exact}}$, related to $\mathbf{z}_k$ via

$$\mathbf{x}_k = B_A^\dagger (B_A^\dagger)^T A^T \mathbf{z}_k .$$

We call $\mathbf{d}_k$ the *defect* after k iterations,

$$\mathbf{d}_k = \mathbf{b} - A\mathbf{x}_k ,$$

to distinguish it from the *residual* $\mathbf{r}_k = A^T(\mathbf{b} - A\mathbf{x}_k) = A^T\mathbf{d}_k$ corresponding to the normal equations. The stopping criteria will typically involve the quantities

$$\|\mathbf{d}_k\| \quad \text{and/or} \quad \|\Delta\mathbf{z}_k\| = \|\mathbf{z}_k - \mathbf{z}_{k-1}\| ,$$

cf. Sect. 6.5. We emphasize that not all the above vectors need to be actually stored. Depending on the termination criterion it should suffice to keep only three of them in storage.

As with Landweber's method, the amount of work per iteration is dominated by one matrix-vector product with A and A^T, each, and one application of $B_A^\dagger$ and $(B_A^\dagger)^T$, respectively. The implementation of these subtasks is described in Sects. 6.1 and 6.2.

According to the GSVD (4.2), $\mathbf{x}_k$ may be written as

$$\mathbf{x}_k = \sum_{i=1}^{p} \varphi_k^\nu(\sigma_i^2) \frac{\mathbf{u}_i^T \mathbf{b}}{\rho_i} \mathbf{w}_i + \sum_{i=p+1}^{n} \mathbf{u}_i^T \mathbf{b}\ \mathbf{w}_i .$$

Here, the filter factors depend on the Jacobi-polynomials $P_k^{(\alpha,\beta)}$ (cf. [99]) with $\alpha = 2\nu - 1/2$ and $\beta = -1/2$:

$$\varphi_k^\nu(\lambda) = 1 - P_k^{(2\nu - 1/2,\ -1/2)}(1 - 2\lambda) / P_k^{(2\nu - 1/2,\ -1/2)}(1) . \tag{6.7}$$

This makes it possible to draw upon the many results on classical orthogonal polynomials for a theoretical analysis of the ν-methods; see [13, 42, 44] for more details. Especially, we quote from [44] that

$$0 \le \varphi_k^\nu(\lambda) \le 2k^2\lambda \ll 1, \quad \lambda \ll k^{-2},$$

while

$$\varphi_k^\nu(\lambda) \to 1, \quad \lambda k^2 \to \infty .$$

More precisely, for fixed i we have

$$|1 - \varphi_k^\nu(\sigma_i^2)| \sim k^{-2\nu} .$$

Comparing these estimates with (6.5) it follows that the ν-method will reduce the number of iterations to essentially the square root of the number of iterations required by Landweber's method, cf. [13, 42].

The rate of convergence and the optimal accuracy of the ν-methods improve as ν increases, but only up to some *saturation level* [13, 42]. In order to obtain optimal accuracy (and optimal speed) ν must be chosen above the saturation level. This

level, however, depends on properties of the solution which are (usually) unknown a priori; this suggests to fix ν as large as possible. Nevertheless, there are practical reasons opposing to this rule: the initial convergence of the ν-method is faster for smaller values of ν and this often dominates the overall performance until the stopping criterion is fulfilled. For this reason $0.5 \leq \nu \leq 1$ is usually the most promising parameter range. Still smaller ν should again be avoided: the initial rate of convergence does not improve much further, but striking oscillations in the error norm history make it difficult to find a robust stopping criterion.

We therefore trust in the following *heuristic for choosing* ν: if there is only little noise in the data, or if the solution is known to be very smooth, fix $\nu = 1$; otherwise, fix $\nu = 0.5$.

For a given problem, the approximations obtained with the ν-method with $\nu = 1$ and those from Tikhonov's method should have about the same quality according to the theory. In practice the latter are often inferior [42: sect. 11]. Instead, in most applications the ν-method with $\nu = 1/2$ and Tikhonov's method lead to comparable accuracy.

The most striking inconvenience of the ν-methods is their need for a *scaled problem* fulfilling

$$1 - \delta \leq \|AB_A^\dagger\| \leq 1. \tag{6.8}$$

Here, δ plays the role of a small threshold, as close to zero as possible; larger values of δ may lead to retarded convergence. If the upper bound in (6.8) is violated, the iteration diverges rapidly right from the beginning. Consequently, A and $\mathbf{b}$ have to be rescaled with some scaling factor γ in order to fulfill (6.8). If no such scaler γ is known in advance (the typical case), γ has to be determined numerically. This could be done by means of the power method but we rather suggest the use of a few steps of the LSQR-algorithm by Paige and Saunders [88]: both alternatives require about the same amount of work, but LSQR yields a good approximation $\mathbf{x}_{\text{guess}}$ for $\mathbf{x}^{\text{exact}}$ in addition to the γ-estimate. We describe LSQR in Sect. 7, and we therefore postpone the details to that chapter.

6.4 Conjugate gradients

The recursion coefficients α_k and β_k of the ν-method are independent of the problem under consideration; as a matter of fact the filter functions φ_k^ν in (6.7) are the same in every application. It was proved in [42] that among all iterative regularization schemes with the above property, no method can reach a comparable accuracy with significantly less iterations than the ν-method (with suitably chosen ν).

On the other hand, Brakhage has shown in [13] that the *conjugate gradient* method applied to the normal equations (CGNR) will always need less iterations to yield similar accuracy. According to [68] CGNR may even be dramatically faster under appropriate conditions.

A couple of equivalent implementations of CGNR are possible; here, we present the one that is generally assumed to be the most stable one, cf. Björck [11: p. 560]:

Algorithm 6.3. (CGNR-method)

— Generate $\mathbf{x}_0 = \mathbf{x}_*$ and put $\mathbf{d}_0 = \mathbf{b} - A\mathbf{x}_*$
— Initialize

$$\mathbf{r}_0 = A^T\mathbf{d}_0$$
$$\mathbf{q}_0 = (B_A^\dagger)^T\mathbf{r}_0$$
$$\mathbf{s}_1 = B_A^\dagger\mathbf{q}_0.$$

— Iterate for $k = 1, 2, \ldots$ until the termination criterion is fulfilled:

$$\alpha_k = \|\mathbf{q}_{k-1}\|^2/\|A\mathbf{s}_k\|^2$$
$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{s}_k$$
$$\mathbf{d}_k = \mathbf{d}_{k-1} - \alpha_k A\mathbf{s}_k$$
$$\mathbf{r}_k = A^T\mathbf{d}_k$$
$$\mathbf{q}_k = (B_A^\dagger)^T\mathbf{r}_k$$
$$\beta_k = \|\mathbf{q}_k\|^2/\|\mathbf{q}_{k-1}\|^2$$
$$\mathbf{s}_{k+1} = B_A^\dagger\mathbf{q}_k + \beta_k\mathbf{s}_k.$$

Again, $\mathbf{d}_k$ is the defect after k iterations:

$$\mathbf{d}_k = \mathbf{b} - A\mathbf{x}_k.$$

The vector $\mathbf{s}_k$ is called the kth *search direction*.

The details of the implementation of Algorithm 6.3 are the same as described in Sects. 6.1 and 6.2. Due to the inner products required for the evaluation of α_k and β_k, one CGNR iteration is slightly more expensive than one iteration of the ν-method; the difference, however, will be negligible in general—only if the matrix vector product with A is extremely cheap, the $O(n)$ extra operations can become important. No convergence condition like (6.8) is required for CGNR.

For the iterates $\mathbf{x}_k$ we have the filtered GSVD expansion,

$$\mathbf{x}_k = \sum_{i=1}^{p} \varphi_k^{\text{CG}}(\sigma_i^2)\frac{\mathbf{u}_i^T\mathbf{b}}{\rho_i}\mathbf{w}_i + \sum_{i=p+1}^{n} \mathbf{u}_i^T\mathbf{b}\,\mathbf{w}_i,$$

but to characterize the filter coefficients some more background material is needed.

The CGNR iterates are based on the orthogonal polynomials (also called *Ritz-polynomials*) p_k with respect to the discrete inner product

$$[\phi, \psi] \equiv \sum_{i=1}^{p} \phi(\sigma_i^2)\psi(\sigma_i^2)\sigma_i^2(\mathbf{u}_i^T\mathbf{b})^2; \tag{6.9}$$

the polynomial p_k has exact degree k and is uniquely defined through

$$p_k(0) = 1, \quad k = 0, 1, 2, \ldots$$

By means of p_k the filter functions φ_k^{CG} can now be expressed as

$$\varphi_k^{\text{CG}}(\lambda) = 1 - p_k(\lambda). \tag{6.10}$$

As a consequence of the orthogonality relation (6.9), p_k minimizes the discrete quadratic functional

$$F[\phi] \equiv \sum_{i=1}^{p} \phi^2(\sigma_i^2)(\mathbf{u}_i^T \mathbf{b})^2$$

among all polynomials ϕ_k of degree k fulfilling $\phi_k(0) = 1$. In other words, the larger the ith weight w_i of $F[\cdot]$,

$$w_i = (\mathbf{u}_i^T \mathbf{b})^2,$$

the smaller $|p_k(\sigma_i^2)|$ will be. According to the Picard condition and its discrete analog, cf. Sect. 4.1, we expect

$$w_i \sim \sigma_i^2 + \tilde{\varepsilon},$$

where the small number $\tilde{\varepsilon}$ is coming from the noisy perturbation in the right-hand side data. We conclude that w_i will be of the order of $\tilde{\varepsilon}$ for small generalized singular values, and much bigger for the larger generalized singular values. In other words, as the iteration proceeds, the orthogonal polynomials p_k will need to become small at the larger σ_i^2 first; hence the filter factors associated with these σ_i will be close to 1 after only a few iterations. As $p_k(0) = 1$, the filter factors corresponding to the smaller generalized singular values will still be close to zero, so that φ_k^{CG} fulfills precisely the filtering requirements we need.

A strict proof of the regularizing properties of conjugate gradients is extremely difficult. It took until 1986 that Nemirovskii [83] verified that the CGNR iterates reach about the same accuracy as Landweber's method before they will start to diverge. Plato [90] proved that CGNR can be made a regularizing algorithm in the sense of Tikhonov and Arsenin. Van der Sluis and Van der Vorst [108] studied the behavior of the filter factors for a set of model situations.

We mention that the polynomials p_k need not be small throughout the interval $(0, \sigma_1^2]$. It is sufficient for minimizing $F[\phi]$ to place zeroes of p_k (so-called *Ritz-values*) close to those σ_i for which the weights w_i are large. This is different for the ν-methods where the polynomials $1 - \varphi_k^{\nu}(\cdot)$ get small in a more uniform way, and this is actually the reason why CGNR can yield the desired accuracy significantly faster. We shall illustrate this further in two case studies.

Assume firstly that only a few, say, q, generalized singular values $\sigma_1, \ldots, \sigma_q$ are comparatively large while all other singular values are quite small. Such a situation will arise if the integral operator is *extremely ill-posed*, i.e., if its singular values decay very fast, say, like r^i for some $0 < r < 1$. Then the weights w_i, $q + 1 \leq i \leq n$, will not exceed the order of $\tilde{\varepsilon}$ and after only q iterations the filter factors φ_i^{CG} of the remaining terms ($i = 1, \ldots, q$) will be very close to 1, compare [68]. From that point onwards CGNR will essentially pick up the noise—and will diverge. In other words, after about q iterations only CGNR has reached its optimal accuracy.

Assume secondly that *all* the generalized singular values are equally well distributed over the entire interval $(0, \sigma_1^2]$. Such situations arise for instance in discrete convolution equations over $\mathbb{R}$, where K fails to be compact. Then, CGNR can no longer separate large and small singular values and the filter polynomials φ_k^{CG} will approximate the constant function 1 more uniformly. In this case, cf. [108], the convergence of CGNR cannot be much faster than the convergence of the ν-methods and the rate of convergence mainly depends on the smoothness of the solution, for this essentially determines the decay of the weights w_i, $i = 1, 2, \ldots, n$, cf., e.g., [69].

We finally emphasize that all the above statements refer to computations in exact arithmetic. In finite precision arithmetic it is well known that the Ritz-polynomials will soon lose their orthogonality. As a consequence the iterative process will eventually slow down, but the above discussions concerning the values of the filter factors is not much affected by this loss of orthogonality,cf. [36: sect. 9.2].

As we have seen, the optimal accuracy of CGNR is sometimes obtained after very few iterative steps, but it can also take significantly more steps. The optimal number of iterations, k_*, depends on several properties of the problem, some of which may not be clear in applications. We have seen further that in the first case study, the iterates will diverge from a very early stage of the iteration onwards. Eicke et al. [23] considered a highly informative example with two problems with arbitrarily close right-hand sides: they showed that for one of these problems CGNR reaches its optimal accuracy at some step k_*, whereas for the second problem CGNR already diverged after the same number of iterations. In other words, the optimal stopping index k_* is very sensitive to perturbations in the right-hand side.

We therefore believe that the merits of CGNR almost completely depend on the quality of the stopping rule employed, and this dependence is much more important here than it is for ν-methods or for Tikhonov's method. We mention the examples presented in [42: sect. 11] where three CGNR iterations lead to optimal accuracy and two further iterations bounce the error norm back up to where it was for $k = 1$. Thus, CGNR ultimately has to be stopped after three iterations in this particular instance; *no tolerance is permissible*. In Sect. 7.2 we will return to this problem when we describe a hybrid algorithm.

Finally, we briefly address *preconditioning issues*. In well-posed applications, conjugate gradients is usually employed with preconditioning, i.e., instead of $A\mathbf{x} = \mathbf{b}$, conjugate gradients is applied to the system

$$QA\mathbf{x} = Q\mathbf{b};$$

the nonsingular matrix Q is called the *preconditioner*. The idea of preconditioning is to improve the rate of convergence by improving spectral properties of the coefficient matrix, for example by clustering its singular values, or by reducing its condition number.

As we hopefully have made clear above, preconditioning is less important for ill-posed problems in general because the number of iterations for CGNR is often small, anyway. Nevertheless, there are applications where this is different, for instance in the aforementioned convolution problem where the spectra of K and A are well distributed in the entire interval $(0, \sigma_1^2]$.

To find suitable preconditioners for ill-posed problems is extremely more difficult than it is for well-posed problems: for example, to maintain the filtering properties of CGNR it is required that the right-hand side components corresponding to the very small singular values of A must remain in a subspace exclusively

spanned by left singular vectors associated with tiny singular values of QA. Roughly speaking, this means that Q should only precondition the "well-posed part" of A, i.e., cluster the larger singular values of A; Q *must not* precondition the "ill-posed part" of A but should rather act there like the identity.

Such a requirement will rarely be fulfilled by conventional preconditioners that have been suggested so far in the literature for well-posed problems. In the construction of those preconditioners one simply tries to exploit the properties of A as a matrix but not those of the associated mapping.

One exception is the use of circulant preconditioners for Toeplitz coefficient matrices: such circulant preconditioners have been adapted with success in [46] to meet the aforementioned requirements for certain ill-posed convolution problems.

A preconditioner for first kind integral equations which mimics multigrid has been suggested by King [59, 60]. Multigrid methods seek to first approximate the slowly varying components of the solution, and it is this property that may make such methods suited as preconditioners for ill-posed problems.

6.5 Stopping rules

Once we accept the sequential nature of iterative methods, we might as well take advantage of it: the user can study on-line adequate visualizations of the iterates as soon as they are computed, and simply halt the iteration when the approximations reach the desired quality—or when they become worse. This may actually be the most appropriate stopping rule in many practical applications, but it requires a good intuitive imagination of what to expect.

In other situations the user will need the computer's help to determine the optimal approximation, and this is what we are concerned with in the sequel. As we have emphasized above, the iteration index k may be considered as the regularization parameter of iterative methods, and therefore similar methods as in Sects. 5.3 and 5.4 apply for choosing k. As discussed there, parameter choice strategies naturally divide in two categories: rules which are based on knowledge of the norm of the errors, and rules which do not require such information.

If the error norm is known within reasonable accuracy, then we suggest to employ the *discrepancy principle* as a stopping rule. Let ε denote the noise component in $\mathbf{b}$, i.e., the discretization of ε in (3.5); the iteration process should then be stopped as soon as the defect $\mathbf{d}_k = \mathbf{b} - A\mathbf{x}_k$ for the first time drops below the noise level, i.e., when

$$\|\mathbf{b} - A\mathbf{x}_{k_D}\| \leq \|\varepsilon\| < \|\mathbf{b} - A\mathbf{x}_k\|, \quad 0 \leq k < k_D. \tag{6.11}$$

If (6.11) is satisfied already for $k_D = 0$, then one cannot expect to improve $\mathbf{x}_0$ by iteration; in this case the program should return $\mathbf{x}_0$ as regularized approximation.

The theoretical properties of the discrepancy principle were developed by Nemirovskii [83] and Plato [90] for CGNR, and in [42] for the ν-methods. These results indicate that with the discrepancy principle one will obtain approximations that are close to the optimal accuracy that we may expect (we refer to the aforementioned works for a precise statement). Actually, to obtain these theoretical results it is required that the right-hand side of (6.11) is slightly bigger than $\|\varepsilon\|$; on the other hand, other authors are led from their numerical results with discrete problems to rather decrease the right-hand side of (6.11) below $\|\varepsilon\|$, cf., e.g., [56: p. 129]; see also [42] for numerical experiments with the discrepancy principle as a stopping rule for iterative methods. We nevertheless emphasize that underestimation of the noise level may cause a failure of this stopping rule.

Concerning the second category of methods, we must consider CGNR and the ν-methods separately. Our experience with the *L-curve criterion*—cf. Sect. 8—is that this method works quite successfully for CGNR. Motivated by the numerical results in [49, 51] and by heuristic considerations, the L-curve criterion is likely to perform the better the more rapid the convergence of CGNR is superseded by divergence, because this will make the L-corner more pronounced; unfortunately we do not know how to justify this claim theoretically. The use of the L-curve is essentially the same as described for Tikhonov regularization in Sect. 5.4, except that now the L-curve is discrete. To define and computer the "corner" of such a graph, the following strategy is proposed in [51]:

Algorithm 6.4. (L-curve criterion for CGNR)

— Approximate the discrete L-curve by a fitting 2-D spline curve.
— Locate the corner of the spline curve.
— Determine the point on the discrete L-curve closest to the spline's corner.

Note that the defects $\mathbf{d}_k$ of the CGNR iterates are computed anyway in Algorithm 6.3.

Vogel [113] suggests to use generalized cross-validation (GCV) as a stopping rule for CGNR. However, there are several drawbacks concerned with this stopping rule. For one thing, it is not possible to compute the denominator of the GCV function for CGNR without explicit knowledge about the singular values of $\bar{A}$, which are usually unknown, and we know of no good aproximate scheme for computing this denominator. This makes the method quite impractical. Moreover, we fear that even if the GCV function could be computed exactly, then GCV may still not be suited as a stopping rule for CGNR because of the highly nonlinear dependence of the CGNR iterates on the right-hand side data $\mathbf{b}$.

Consider now the ν-methods in case we do not have an estimate for the noise-level. Unfortunately, the L-curve criterion does not get along well with the ν-methods because neither the residual norms nor the norms of the iterates of the ν-method depend on k monotonically. Therefore, the graph of the L-curve is usually full of wiggles and does not provide much help.

GCV is also difficult to employ as a stopping rule for the ν-methods. In this context (cf., e.g., [116]) GCV requires to monitor the sequence

$$\mathcal{G}_k = \frac{\|\bar{\mathbf{b}} - \bar{A}\bar{\mathbf{x}}_k\|}{\operatorname{trace}[I - \varphi_k^{\nu}(\bar{A}\bar{A}^T)]} \tag{6.12}$$

with the filter polynomials φ_k^{ν} of (6.7); unfortunately, the denominator of $\mathcal{G}_k$ cannot be computed unless we have precise knowledge of the singular values of $\bar{A}$.

To overcome this difficulty, there is a variant of GCV due to Girard [33] termed *Monte Carlo* GCV, which can be implemented at the prize of roughly twice the work

required for the iteration itself. The method is built on a second sequence of iterates $\tilde{\mathbf{x}}_k$ that are computed simultaneously with $\mathbf{x}_k$, starting from a *random right-hand side vector* $\tilde{\mathbf{b}}$ which should have similar statistical properties as ε. Expanding $\tilde{\mathbf{b}}$ in terms of the GSVD (4.2) of (A, B),

$$\tilde{\mathbf{b}} = \sum_{i=1}^{m} \mathbf{u}_i^T \tilde{\mathbf{b}}\, \mathbf{u}_i,$$

we obtain

$$\mathscr{T}_k := \tilde{\mathbf{b}}^T(\tilde{\mathbf{b}} - A\tilde{\mathbf{x}}_k) = \sum_{i=1}^{n} (1 - \varphi_k^{\nu}(\sigma_i^2))(\mathbf{u}_i^T \tilde{\mathbf{b}})^2. \tag{6.13}$$

From the analysis of Wahba [117] we conclude that the trace expression in the denominator of $\mathscr{G}_k$ acts as a statistical estimate for the noise amplification

$$\sum_{i=1}^{n} (1 - \varphi_k^{\nu}(\sigma_i^2))(\mathbf{u}_i^T \varepsilon)^2;$$

we readily observe that the same task will be fulfilled by the numbers $\mathscr{T}_k$ of (6.13) [33]. Therefore, if we compute $\tilde{\mathbf{x}}_k$ simultaneously with $\mathbf{x}_k$, then we can compute

$$\mathscr{G}_k^{\mathrm{MC}} = \frac{\|\mathbf{b} - A\mathbf{x}_k\|}{\mathscr{T}_k},$$

and stop the iteration as soon as $\mathscr{G}_k^{\mathrm{MC}}$ reaches its minimum. We emphasize, however, that our numerical experience with this stopping rule is only limited; the reason why we nevertheless include this algorithm is that today GCV is likely to be the most well known rule for choosing regularization parameters. Of course, Monte-Carlo GCV can be applied to CGNR in the same way as it can be applied to the ν-methods; but as we have mentioned before GCV and at least as much Monte-Carlo GCV can suffer severely from the nonlinearity of CGNR.

A second stopping rule which has been designed for the ν-methods in particular is the *update criterion* [44]. Here, a scalar comparison sequence Φ_k is computed during the iteration, serving as an approximation of the error norm,

$$\Phi_k \approx \|B(\mathbf{x}^{\mathrm{exact}} - \mathbf{x}_k)\|, \quad k = 1, 2, \ldots$$

The ν-method is then terminated when the comparison sequence reaches its minimum. In contrast to GCV or Monte-Carlo GCV, the computation of Φ_k does not require additional tools; instead, it is simply based on the norms of the *update vectors* $\Delta\mathbf{z}_k$ which are computed anyway in Algorithm 6.2: according to [44] the precise definition of Φ_k is

$$\Phi_0^2 = \frac{4\nu + 1}{4\nu + 2}\|\mathbf{z}_1\|^2$$

$$\Phi_k^2 = \gamma_k \Phi_{k-1}^2 + \delta_k \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \quad k = 1, 2, \ldots \tag{6.14}$$

with γ_k and δ_k defined by

$$\gamma_k = \frac{(k - \frac{1}{2})k^2(k + \nu - 1)}{(k + 1)(k + 2\nu - 1)(k + 2\nu - \frac{1}{2})(k + \nu)}$$

$$\delta_k = \frac{(k + 2\nu)(k + 2\nu + \frac{1}{2})}{4(k + 1)(k + \nu)(k + \nu + \frac{1}{2})}.$$

Consequently, the extra work required per iteration is essentially one inner product.

The update criterion can be justified by its theoretical properties and its numerical performance [44], but as any other stopping rule which ignores the magnitude of ε, the update criterion cannot lead to a regularizing algorithm in the strict sense of Tikhonov and Arsenin.

Note that neither $\mathscr{G}_k^{\mathrm{MC}}$ nor Φ_k need to have only one local minimum; therefore in practice, once a minimum has occurred one will nevertheless pursue the iteration somewhat further to come to the final decision whether to stop or not.

7 A hybrid method

In Sects. 5 and 6, we have described Tikhonov regularization and iterative regularization methods, and we have emphasized the different philosophies behind these methods and the resulting advantages. Some authors have tried to combine these ideas (in what we call *hybrid methods*) in order to exploit the particular advantages from both approaches.

Below, we review the LSQR algorithm for iterative bidiagonalization of the matrix $\bar{A}$, and discuss possible ways of incorporating regularization. Some of this material is taken from [86, 10].

7.1 *The LSQR algorithm*

We begin with a brief review of the LSQR method given by Paige and Saunders [88, 89]. Essentially, there are two different ways of interpreting their algorithm: one way is to consider LSQR as an iterative algorithm for bidiagonalizing an $m \times n$ matrix with orthogonal transformations (cf. [35]); the other way is to consider LSQR as an equivalent formulation of CGNR, which avoids *explicit and implicit* formation of the cross-product matrix $\bar{A}^T\bar{A}$. We discuss LSQR here for the standard-form problem with $\bar{A}$ and $\bar{\mathbf{b}}$.

The algorithm determines unit vectors $\bar{\mathbf{u}}_k$ and $\bar{\mathbf{v}}_k$, i.e.,

$$\|\bar{\mathbf{u}}_k\| = \|\bar{\mathbf{v}}_k\| = 1, \quad k = 1, 2, \ldots, \tag{7.1}$$

implicitly as follows. First, initialize

$$\beta_1 \bar{\mathbf{u}}_1 = \bar{\mathbf{b}}, \quad \alpha_1 \bar{\mathbf{v}}_1 = \bar{A}^T \bar{\mathbf{u}}_1,$$

and then, for $k = 2, 3, \ldots,$

$$\begin{aligned} \beta_k \bar{\mathbf{u}}_k &= \bar{A}\bar{\mathbf{v}}_{k-1} - \alpha_{k-1}\bar{\mathbf{u}}_{k-1} \\ \alpha_k \bar{\mathbf{v}}_k &= \bar{A}^T \bar{\mathbf{u}}_k - \beta_k \bar{\mathbf{v}}_{k-1}. \end{aligned} \tag{7.2}$$

Hereby, the positive numbers α_k and β_k, $k = 1, 2, \ldots,$ have to be chosen so as to satisfy (7.1). Of course, (7.2) is iterative in nature, and its implementation will follow the lines of Sect. 6.1.

The above recursion may be written compactly as follows:

$$\bar{A}\bar{V}_k = \bar{U}_{k+1} J_k, \quad k = 1, 2, \ldots \tag{7.3}$$

where the matrices

$$\bar{U}_k = (\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \ldots, \bar{\mathbf{u}}_k) \in \mathbb{R}^{m \times k} \quad \text{and} \quad \bar{V}_k = (\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \ldots, \bar{\mathbf{v}}_k) \in \mathbb{R}^{n \times k}$$

both have orthonormal columns, and span the *Krylov spaces*

$$\mathcal{R}(\bar{U}_k) = \operatorname{span}\{\bar{\mathbf{b}}, \bar{A}\bar{A}^T\bar{\mathbf{b}}, \ldots, (\bar{A}\bar{A}^T)^{k-1}\bar{\mathbf{b}}\}$$

$$\mathcal{R}(\bar{V}_k) = \operatorname{span}\{\bar{A}^T\bar{\mathbf{b}}, (\bar{A}^T\bar{A})\bar{A}^T\bar{\mathbf{b}}, \ldots, (\bar{A}^T\bar{A})^{k-1}\bar{A}^T\bar{\mathbf{b}}\}.$$

Moreover, J_k is a lowere bidiagonal matrix consisting of the recursion coefficients,

$$J_k = \begin{pmatrix} \alpha_1 & & & 0 \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ 0 & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{k+1 \times k}.$$

One can show that under the present assumptions the algorithm will not break down (that is, neither β_k nor α_k will become zero) unless the Krylov space spanned by $\bar{V}_k$ is large enough to contain a solution to the linear system $\bar{A}\bar{\mathbf{x}} = \bar{\mathbf{b}}$. When this is the case, the last row of J_k is zero, and we have obtained one bidiagonalization of $\bar{A}$, similar to the one in Sect. 5.1.

LSQR is closely related to the Lanczos process (cf. [36]), and the singular values of J_k (whose squares are the Ritz values) may serve as approximations to the singular values of $\bar{A}$. Recall that the Ritz values are precisely the zeros of the Ritz-polynomials which define the iterates of CGNR (cf. Sect. 6.4). Indeed, after k steps of the LSQR process one can easily obtain the kth CGNR iterate $\mathbf{x}_k$ by first solving the least squares problem

$$\text{minimize} \; \| J_k \bar{\mathbf{z}}_k - \beta_1 \mathbf{e}_1 \|, \tag{7.4}$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$ is the first Cartesian coordinate vector in $\mathbb{R}^{k+1}$, and then computing

$$\mathbf{x}_k = \mathbf{x}_* + B_A^\dagger \bar{V}_k \bar{\mathbf{z}}_k.$$

Paige and Saunders have shown how to solve (7.4) by means of a QR-decomposition of J_k that is updated efficiently from step to step. In this way, they obtained an alternative recursive implementation of CGNR, which is only slightly more expensive than the conventional implementation of CGNR, cf. Algorithm 6.3. On the other hand, it is widely accepted that LSQR is more stable as it completely avoids the matrix $\bar{A}^T\bar{A}$, but we are not aware of a thorough comparison of the accuracy and efficiency of CGNR versus LSQR for ill-posed problems.

We omit further details about LSQR, and refer to the original paper [88]. Instead we return to the *scaling problem* of the ν-methods (cf. Step 2 in Algorithm 6.2). From what we have said before, performing just a few, say k_0, steps of LSQR (in our experiments we never did more than three steps), we obtain an approximation $\tilde{\sigma}_1$ for the norm of $\bar{A}$:

$$\| \bar{A} \| \approx \tilde{\sigma}_1 = \| J_{k_0} \|.$$

We emphasize that the quality of this approximation will be similar to the one we would get from the power method, as $\bar{\mathbf{b}}$ usually has dominating components corresponding to the larger singular values of $\bar{A}$ (see also the discussion in Sect. 6.4).

Simultaneously, we can compute the k_0th iterate of CGNR cheaply from (7.4). Rescaling then A and $\mathbf{b}$ with some factor slightly smaller than $1/\tilde{\sigma}_1$ will usually suffice to guarantee (6.8), which is required for the convergence of the ν-methods. If we then also incorporate $\mathbf{x}_{k_0}$ as initial guess for the ν-method, the process of estimating $\tilde{\sigma}_1$ has been for free, essentially.

7.2 LSQR with regularization

Recall that in each step of the LSQR process we actually solve a least squares problem (7.4). When k increases, these systems will become ill-conditioned: as we have stated in the previous section, some of the singular values of J_k will approximate singular values of $\bar{A}$, and it may even happen that Ritz-values approximate tiny eigenvalues of $\bar{A}^T\bar{A}$ before all the large eigenvalues of $\bar{A}^T\bar{A}$ have been approximated. For this reason we may wish to combine LSQR with Tikhonov regularization, i.e., to regularize system (7.4). Accordingly, for some $\alpha > 0$, we would instead solve

$$\min \left\| \begin{pmatrix} J_k \\ \sqrt{\alpha} I \end{pmatrix} \bar{\mathbf{z}} - \begin{pmatrix} \beta_1 \mathbf{e}_1 \\ 0 \end{pmatrix} \right\| \tag{7.5}$$

for $\bar{\mathbf{z}} = \bar{\mathbf{z}}_{k,\alpha} \in \mathbb{R}^k$, say, and then put

$$\mathbf{x}_{k,\alpha} = \mathbf{x}_* + B_A^\dagger \bar{V}_k \bar{\mathbf{z}}_{k,\alpha}. \tag{7.6}$$

We call this *regularized LSQR*. Eq. (7.5) may be solved by means of Givens rotations along the lines of Sect. 5.1 with only $O(k)$ operations.

As follows from the discussion in [89], $\mathbf{x}_{k,\alpha}$ is the unique minimizer of the discrete Tikhonov functional

$$\| \mathbf{b} - A\mathbf{x} \|^2 + \alpha \| B\mathbf{x} \|^2$$

from the affine space

$$\mathbf{x}_* + \mathcal{R}(B_A^\dagger \bar{V}_k). \tag{7.7}$$

Moreover, $\mathbf{x}_{k,\alpha}$ is also the kth iterate of the conjugate gradient algorithm when applied to the damped normal equations system (5.1) as described at the end of Section 5.1. Consequently, for fixed α,

$$\mathbf{x}_{k,\alpha} \to \mathbf{x}_\alpha, \quad k \to \infty, \tag{7.8}$$

but the convergence will often be too slow for practical purposes.

The case where α is fixed is incorporated in the LSQR implementation in [89]. However, we stress that due to the special nature of the least squares problem (7.4) it is more natural to choose different *inner regularization parameters* $\alpha_k = \alpha(k)$ in each step of LSQR, in order to take into account the convergence properties of the

Ritz values. O'Leary and Simmons [86] and Björck [10] suggested such regularized versions of LSQR, as an iterative approach to regularization; instead of merely using Tikhonov regularization for stabilization of (7.4), they advocate to compute an SVD of the small $(k+1)\times k$ lower bidiagonal matrix J_k and then substitute any favorite filtered SVD expansion of $\bar{z}_k$ in (7.6), for example *truncated SVD*. We refer to the implementational details provided in [86, 10]. We mention in passing that a related approach using block power iterations is suggested by Vogel and Wade [114].

An interesting contribution to regularized LSQR was given recently by Golub and von Matt [37]. They make use of the spectral information on $\bar{A}$ acquired from k steps of LSQR to approximately determine the "correct" regularization parameter α (when interpreted as a Lagrange multiplier, see [37]), and then solve (7.5) for $\bar{z}_{k,\alpha}$. The drawback with their approach is the need for a reliable estimate of $\|x^{\text{exact}}\|$.

We emphasize that we face the problem of choosing two regularization parameters, k and α. For example, Björck [10] applies GCV within each iteration to determine the inner regularization parameters α_k, i.e. to minimize $\mathscr{V}_k(\alpha)$ according to (5.21) (we use the subscript k here to emphasize that the functional $\mathscr{V}$ is different in each step); Björck suggests to stop the iteration (i.e., to choose the *outer regularization parameter* k) when the minimal value of the kth GCV-function

$$\mathscr{V}_k(\alpha_k), \quad k = 1, 2, \ldots,$$

considered as function of k, attains its minimum.

We wish to exploit this idea somewhat further. If k is fixed, then the following analog of (7.8) holds:

$$x_{k,\alpha} \to x_k, \quad \alpha \to 0,$$

where, again, x_k is the kth CGNR iterate. In the beginning of the iteration, CGNR is perfectly stable and $x_k = x_{k,0}$ is probably the best approximation to the solution x that we can select from the affine space (7.7). When the optimal iterate x_{k_*} of CGNR is computed (from the LSQR process), then we would ideally want to stop the iteration with $k = k_*$, because x_{k_*} has optimal accuracy. The difficulty is that we are likely to fail to determine k_* exactly (cf. Sects. 6.4 and 6.5). However, if we continue iterating, then the CGNR iterates start to diverge; this is the point where we must incorporate regularization in (7.4) to overcome the divergence. Heuristically speaking, it should be possible to maintain the optimal accuracy in the subsequent *regularized* LSQR iterates x_{k,α_k} with a proper choice of α_k, because all information that is necessary to recover a good approximation is still maintained in the affine space (7.7). This is similar to what we have discussed in Section 3.3 concerning the regularizing properties of projection methods. Hence, another stopping criterion for the regularized LSQR process would be to stop when x_{k,α_k} does no longer change significantly.

Therefore, we may consider regularized LSQR (as suggested in [10]) as an approach to overcome the semiconvergence effect of CGNR; it remains an issue of future research to find a rigorous proof for the above heuristics, and to determine the most efficient ways of incorporating "inner regularization" to LSQR. So far, we can say that our numerical experiments have been quite promising, especially for those applications where CGNR attains its optimal approximation very early, and then diverges rapidly.

8 Numerical examples

In this final chapter we will illustrate the use of some of the techniques and algorithms presented in the previous chapters. In particular, we will illustrate the use of Tikhonov regularization, the ν-method, and the conjugate gradient method, by applying these methods to model problems with known exact solution. For each regularization method, we will apply several parameter choice strategies for choosing α and k, and we will compare the solutions obtained using these α, k with those corresponding to the optimal regularization parameters.

In order to give an idea about the robustness of the various parameter choice strategies, we will actually—for each combination of regularization method and parameter choice strategy—perform 100 numerical tests with the same exact solution and random perturbations with the same statistics added to the exact right-hand side. Such rigorous comparisons are rarely found in the literature.

All the numerical tests were performed in MATLAB on a Sun SPARC with machine precision $2.22\cdot 10^{-16}$ and IEEE arithmetic.

8.1 The model problem from inverse helioseismology

As our first model problem, we shall use the inverse helioseismic problem described in Sect. 2.1. The exact solution x^{exact} that we shall use has also been used by astrophysicists to obtain insight into this problem, cf. [15], and it is constructed so that it has some of the features that the real, unknown, solution can be expected to have, such as a discontinuous first derivative near the surface of the sun. As regularization operator we use the second derivative operator, which is the choice currently preferred by the astrophysicists. Also, following the approach taken by the astrophysicists, we use the approach "first regularize, then discretize", as far as possible. Note that the setting of this particular problem is already semi-discrete in that we are only given finitely many generalized moments of f, cf. (2.1).

We search for approximate solutions f out of the set of piecewise constant functions, where the particular decomposition of the interval $[0, R]$ that we use is non-equidistant in radius r, but instead equidistant in the so-called acoustical radius τ, defined as

$$\tau = \int_0^r \frac{d\rho}{c(\rho)}.$$

Here, $c(r)$ is the sound speed which depends on the radius r because the density and other characteristics of the sun depend on the depth. This discretization is much finer near the surface than at the center of the sun.

Let τ_j denote the midpoint of the j-th interval, $j = 1, \ldots, n$, and let f_j denote the value of the approximate solution in interval j. Then we build the unknown vector x as $x = (f_1, \ldots, f_n)^T$, i.e., the components of the solution vector are precisely the function values of the piecewise constant approximate solution f.

To discretize the right-hand side of (2.1) we simply use the midpoint rule, while we use central difference approximations to the values of $Lf = f''$ in $\tau = \tau_j$, $j = 2, \ldots, n-1$. The integrals required for the inner product for the evaluation of $\| Lf \|$ are again approximated by the midpoint rule. Thus, B of (4.1) is a tridiagonal $(n-2) \times n$ matrix of full rank—as in (4.10).

Our model problem is only moderately sized, namely we have $n = 100$ and $m = 212$ simulated observations, i.e. mode sets, cf. (2.1). The errors ε that we add to the exact right-hand side, generated as $\mathbf{b}^{\text{exact}} = A\mathbf{x}^{\text{exact}}$, are normally distributed with zero mean and standard deviation $s = 0.01$. This choice, which is a realistic noise level for today's observations, corresponds to a right-hand side measured in μHz, for which the elements of $\mathbf{b}$ lie in the range 0.3–0.5. The norm of $\mathbf{b}$ is $\| \mathbf{b} \| = 6.1$.

As we mentioned in Sect. 2.1, we cannot resolve the solution near the center of the sun because of the intrinsic properties of the kernels $k_{n,l}$ (which, in turn, reflect a fundamental physical property of the inversion problem). Hence—having in mind that the vector components of the computed solution are simply the function values of the piecewise constant approximation—when we compare the various regularized solutions with the exact solution $\mathbf{x}^{\text{exact}}$, then it makes only little sense to compare the elements of the solution vector with small indices, corresponding to the sun's center. Therefore, we "chop off" the first 12 elements, corresponding to a cut-off radius at $r \approx 0.25R$, and compute the relative error in the regularized solution as

$$\eta^{\text{reg}} = \| \mathbf{x}^{\text{exact}}(13:100) - \mathbf{x}^{\text{reg}}(13:100) \| / \| \mathbf{x}^{\text{exact}}(13:100) \| , \qquad (8.1)$$

where we have used MATLAB notation, and $\mathbf{x}^{\text{reg}}$ denotes a solution computed by any of the regularization methods in consideration. We believe that this is the fairest strategy for comparing these methods for this particular application. A more detailed discussion of the quality of the regularized solutions to this problem is outside the scope of the present paper.

We note in passing that it would perhaps be more natural to base the comparison on an error measure of the form $\| B(\mathbf{x}^{\text{exact}} - \mathbf{x}^{\text{reg}}) \|$, since it is indeed this error that the regularization methods seek to minimize. However, it is the error measure in (8.1) that is more interesting in practice.

First, we considered Tikhonov regularization, and we used the following five parameter choice criteria:

1. the discrepancy principle
2. the Gfrerer/Raus-method
3. the quasi-optimality criterion
4. generalized cross-validation
5. the L-curve criterion.

For the first two methods, we used the stochastic expectation value of $\| \varepsilon \| \approx \sqrt{ms} = \sqrt{212} \cdot 0.01 = 0.146$ to estimate $\| \varepsilon \|$. This estimate should be seen in relation to the actual values of our 100 noise samples in which $\| \varepsilon \|$ ranged between 0.130 and 0.161. The results of the 100 experiments are shown in Fig. 1 a: the error obtained with each of the parameter choice rules is compared with the optimal

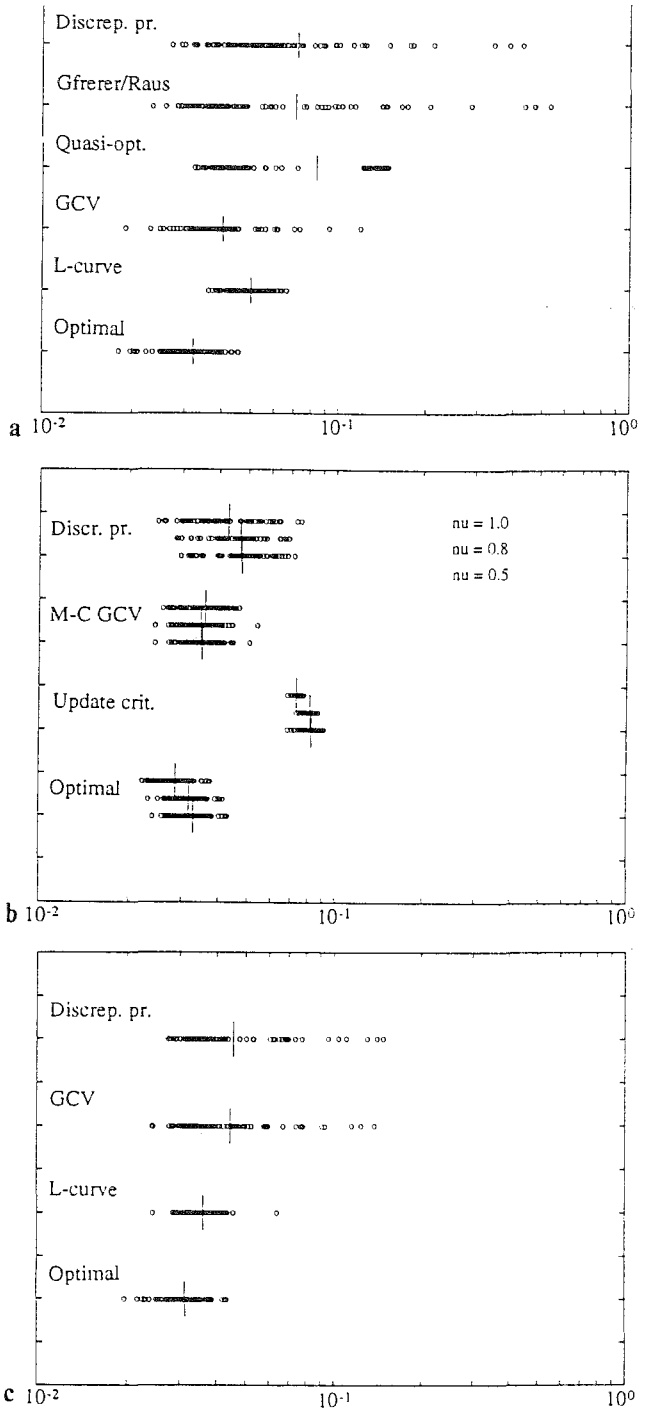


Fig. 1. Relative errors in the computed solutions for inverse helioseismology using **a** Tikhonov regularization with five different parameter-choice strategies, **b** the ν-method, and **c** CGNR regularization with three different parameter-choice strategies, respectively. The bottom line shows the optimal errors for the respective method

error, i.e., the error that minimizes η^{Tikhonov} over all possible α. The vertical line shown for each set of errors is the mean value of these errors.

In three cases out of the 100 tests, the discrepancy principle and the Gfrerer/Raus-method gave a completely wrong solution with a relative error greater than one. These three cases are not included in the results shown in the figure.

From Fig. 1 a we make several important observations. We see that for the discrepancy principle and the Gfrerer/Raus-method, it is very important to have a *good* estimate of the norm of the errors in the right-hand side; the reason why we can get quite large errors in the regularized solutions is that 0.146 is sometimes a too crude estimate for the optimal value, namely, the actual norm $\|\varepsilon\|$ (we recall that these methods would give guaranteed accuracy, if the estimate of the data-error is optimal). We also see that in approximately half of the tests using the quasioptimality criterion, this criterion does a good job in finding a good regularization parameter, but in the remaining half of the tests its choice is not so good. Both GCV and the L-curve criterion are the most robust methods and, in average, they also give more accurate results than the other parameter choice strategies. Finally, we see that for this model problem with normally distributed errors in the right-hand side, the GCV method yields solutions which are actually, in average, slightly more accurate than those obtained by the L-curve criterion.

Next, we carried out three sets of experiments with the ν-method, for ν equal to 1.0, 0.8, and 0.5, and with the zero vector as the initial vector for the iterations. We used the following three stopping rules:

1. the discrepancy principle
2. Monte-Carlo GCV
3. the update criterion.

The results are shown in Fig. 1 b. Again, the vertical lines show the mean values of the errors. With ν equal to 1.0 and 0.8, in average 19 iterations were required to reach the optimal solution, while about 15 iterations were required for $\nu = 0.5$. The optimal errors are slightly smaller for $\nu = 1.0$, but the difference between the results does not depend much on the parameter ν. The same is true for the actual errors corresponding to the above three parameter choice strategies. This reflects the fact that our model solution has a discontinuity and therefore lacks smoothness.

The best results are obtained by the Monte-Carlo GCV method, while the update criterion yields larger errors. This is because the update criterion notoriously leads to a too small number of iterations which, in turn, is caused by the fact that the update criterion seeks to approximate the error measured in the norm $\|B\cdot\|$, and not in the norm $\|\cdot\|$. Therefore, the wiggles in the approximations are overestimated, and the iteration is stopped too early. Recall that the costs of using the Monte-Carlo GCV method, as compared with the update criterion, are about twice as many operations because two systems are run "in parallel", cf. Sect. 6.5. For all three values of ν, the discrepancy principle failed in 35 out of the 100 tests because the norm of the defect never dropped below 0.146—hence, this method does not seem to be suited for the ν-method; at least for this model problem.

Finally, we consider the conjugate gradient method. The particular implementation of CGNR that we used here is based on the LSQR algorithm, and we have chosen the following four stopping rules:

1. the discrepancy principle
2. generalized cross-validation
3. the L-curve criterion.

In average, we found that the optimal solution required only four CGNR iterations. This illustrates how fast this method converges for some ill-posed problems.

The results for CGNR are shown in Fig. 1 c. Regarding the discrepancy principle, we found that in four cases the CGNR defect norm $\|\mathbf{b} - A\mathbf{x}_k\|$ never dropped below the estimate 0.146 for the norm of the right-hand side errors. Again, these four tests are not included in the figure.

The implementation of GCV that we use in which the GCV function is approximated as $\mathcal{G}_k \approx \|\mathbf{b} - A\mathbf{x}_k\|/(m-(n-p)-k)$, is not strictly correct because the denominator is only a rough approximation to the correct trace-term (which cannot be computed without explicitly knowing all the generalized singular values).

The L-curve criterion is by far the most robust parameter choice criterion for CGNR for this model problem, and its average error is not much larger than the average optimal error. One of the reasons for the L-curve's good performance is the fact that the L-curve's corner is very sharp and therefore easy to detect, due to the fast convergence of the CGNR iterations.

In conclusion, we find that for this model problem, all three regularization methods produce optimal solutions with an average relative error about $2\cdot 10^{-2}$, and that the best combinations of regularization method and parameter choice strategy are:

—Tikhonov regularization and GCV
—the ν-method and Monte-Carlo GCV
—the CGNR method and the L-curve criterion.

If we take into account the amout of work necessary to perform the three regularization methods, then CGNR—combined with the L-curve criterion—is our choice for this model problem because CGNR converges very fast and because the L-curve criterion is very robust.

8.2 The model problem from image reconstruction

Our second model problem is the 2D deconvolution problem (2.3) with $s_i \equiv 1$ and s_c as in (2.4); for our numerical tests we use $\gamma = 0.1$ and we try to reconstruct two peaks of high brightness, roughly simulating the image of, e.g., two stars.

This problem is of special interest because here the underlying (selfadjoint) operator K is not compact. Its spectrum is given by the closure of the values of the two-dimensional Fourier transform of s_c over $\mathbb{R}^2$. For $s_c \in L^1(\mathbb{R}^2)$ the spectrum of K is a continuum including in particular the origin $\lambda = 0$. Consequently K^{-1} is unbounded and the image reconstruction problem is ill-posed.

One of the most popular methods for computing an approximate solution of this problem is *Wiener filtering*, cf., e.g., [1, 57]. It is based on a statistical approach, and is equivalent to what we have called Tikhonov's method in Sect. 5. In Wiener filtering the operator L is constructed from the covariance operators for signal and noise, respectively, assuming them to be independent of each other.

For our discretization of (2.3) we follow the standard approach in the literature and represent the image by an equidistant mesh of $p \times p$ pixels. This is quite natural, indeed, when the image is measured by a CCD camera as is common in astronomy. For convenience, the digital image is stored in lexicographical order in a p^2-vector. The integrals are approximated by the rectangular quadrature rule. Hence, we face a collocation method with piecewise constant trial functions. For the matrix B in (5.1) we choose the identity matrix for simplicity. According to Wiener's approach this means that we incorporate no a priori knowledge about the exact image; as a consequence we must expect difficulties in the reconstruction of the sharp brightness contrasts in our test image (note that these "discontinuities" of the exact solution also prohibit the substitution of a differential operator for L).

We emphasize that the $p^2 \times p^2$ matrix A that we obtain is *doubly Toeplitz*, i.e., it consists of $p \times p$ dimensional Toeplitz blocks, these blocks being the same along each block-diagonal. Thus, matrix-vector products with A and A^T in the iterative regularization methods can be implemented with a generalization of Algorithm 6.1 on the basis of the two-dimensional FFT. For Wiener filtering (or equivalently, for Tikhonov regularization) the Toeplitz blocks are often approximated by circulant blocks thus allowing the use of FFTs for a very rapid approximate inversion of the damped normal equations (5.1). Even without such simplifying approximations a fast numerical algorithm for solving (5.1) or (5.2) was obtained by Chun and Kailath [17].

As in Sect. 8.1, our image reconstruction problem is only moderately sized, with $p = 16$ leading to a 256×256 coefficient matrix A. In the small 16×16 digital image, the two bright peaks are modeled as two 4×5 rectangles. The exact right-hand side is generated as $\mathbf{b}^{\text{exact}} = A\mathbf{x}^{\text{exact}}$ and then rescaled such that its maximum element is 1000. Its norm is then $\|\mathbf{b}\| = 5.7 \cdot 10^3$. Poisson noise is the most realistic approximation of actual data perturbations in today's CCD cameras; hence, we add Poisson noise ε with standard deviation $s = 9.51$ to $\mathbf{b}^{\text{exact}}$. This value of s was chosen to give approximately the same ratio between noise and signal as in Sect. 8.1. For Poisson noise, the expected value of $\|\varepsilon\|$ that we have used in our experiments is $\sqrt{p^2}\sqrt{s^2 + s} = 16 \cdot 10 = 160$ (where p^2 is the length of ε while $s^2 + s$ is the expected value of the square of each element of ε). We thus obtain $\|\varepsilon\|/\|\mathbf{b}^{\text{exact}}\| \approx 0.028$. The actual perturbation norms we observed in our 100 experiments ranged between 152 and 166. We applied the same regularization methods as in Sect. 8.1, except that we only used $\nu = 1$ in the ν-method. The relative error in the regularized solution for this model problem is simply

$$\eta^{\text{reg}} = \|\mathbf{x}^{\text{exact}} - \mathbf{x}^{\text{reg}}\|/\|\mathbf{x}^{\text{exact}}\| . \tag{8.2}$$

Our numerical results are shown in Fig. 2 a–c. The results for this model problem are somewhat different from those of the first model problem in Sect. 8.1. First of all, we notice that for the image reconstruction problem, the relative errors

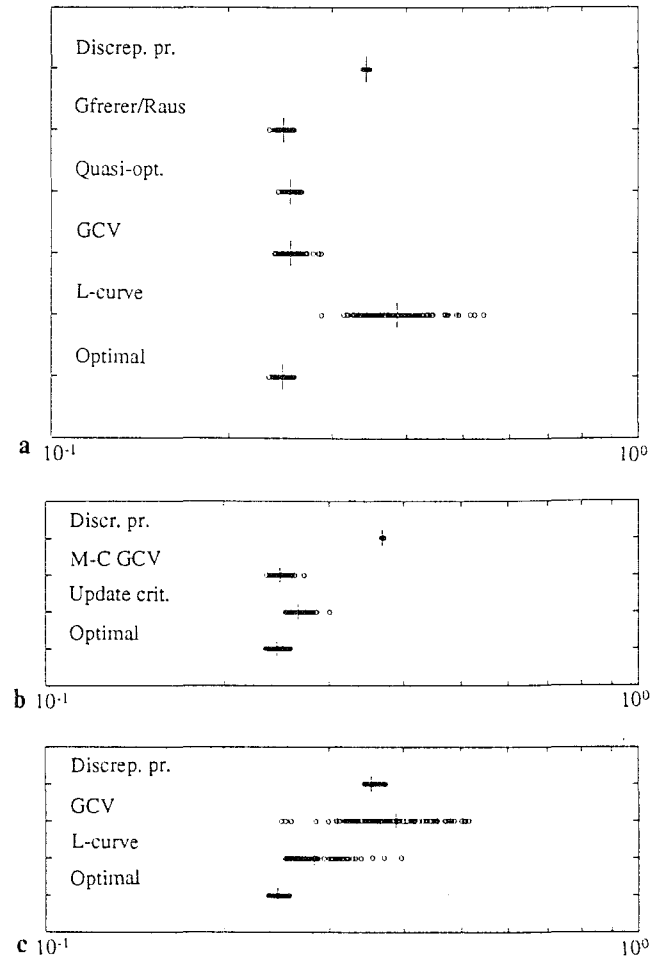


Fig. 2. Relative errors in the computed solutions for image reconstruction using a Tikhonov regularization, **b** the ν-method, and **c** CGNR regularization

η^{reg} are generally less sensitive to different noise samples, but that the magnitude of η^{reg} is worse by almost a factor of 10 compared to the results in Sect. 8.1. This is in agreement with the theory which predicts larger errors due to the sharp contrasts in the object to be reconstructed. For Tikhonov regularization, the best parameter-choice methods are now the Gfrerer/Raus-method, the quasi- optimality criterion, and GCV, while the discrepancy principle and the L-curve criterion consistently give larger errors in the regularized solution. Concerning the iterative methods, the situation is more similar to the previous model problem: Monte-Carlo GCV and the L-curve criterion are still optimal for the ν-method and CG, respectively; the update criterium performs better than in the first model problem.

The average number of iterations required in the iterative methods are listed in Table 1. We observe that for this model problem the ν-method requires only about

Table 1. Average number of iterations for the iterative regularization methods applied to the image restoration model problem

ν-Method regularization		CGNR regularization	
Discrepancy principle	12.0	Discrepancy principle	8.9
Monte-Carlo GCV	78.6	GCV	67.0
Update criterion	39.6	L-curve criterion	52.7
Optimal	74.7	Optimal	34.6

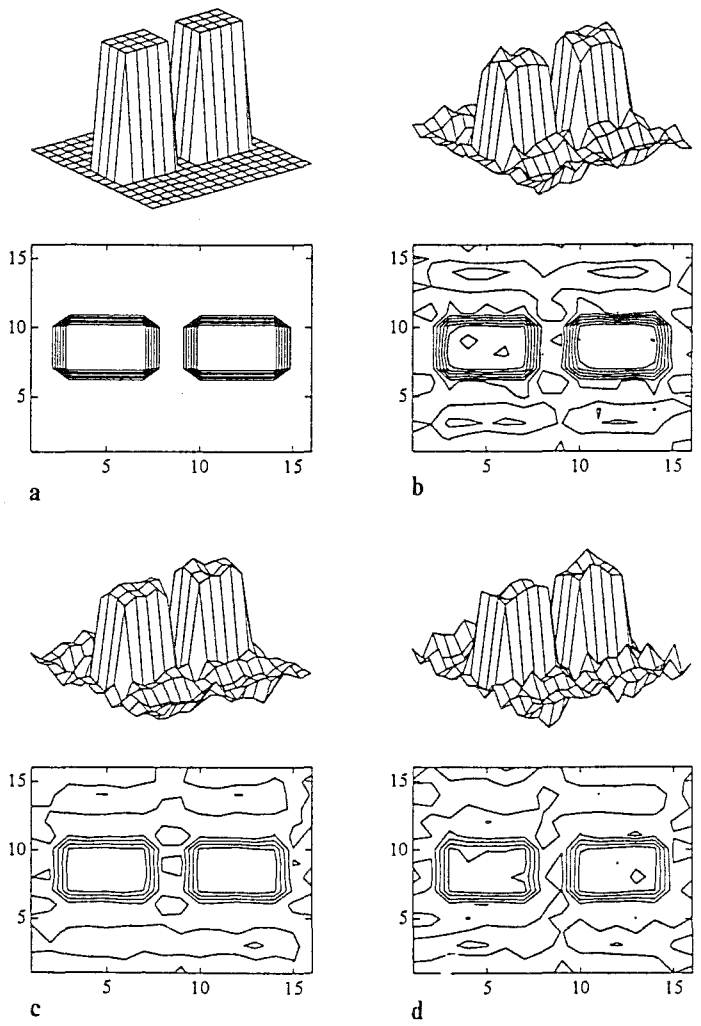


Fig. 3. Surface plots and contour plots of the exact solution (a) and three regularized solutions computed by Tikhonov's method with CGV, (b) by the ν method with $\nu = 1$ and Monte-Carlo CGV, (c) and by CGNR with the L-curve criterion (d)

twice as many iterations as CGNR to achieve approximately the same accuracy; recall that CGNR was significantly faster for the first problem. This reflects the fact that the underlying operator in Sect. 2.1 was compact while the spectrum of the operator in (2.3) is a continuum, cf. also the discussion in Sect. 6.4. The numbers in Table 1 show further that the discrepancy principle and the update criterion consequently lead to too few iterations, while GCV and the L-curve criterion consequently lead to too many iteration steps. Both too few and too many steps deteriorate the quality of the regularized solution. On the other hand, as in our first model problem, Monte-Carlo GCV terminates the ν-method very close to the optimal number of iterations.

The conclusion to be drawn from these results is that the combinations of regularization method and parameter-choice strategy found to be optimal for the helioseismology problem in Sect. 8.1 are also the optimal combinations for this image reconstruction problem. Moreover, the quasi-optimality criterion also works well for Tikhonov regularization. If a good estimate of $\|\varepsilon\|$ is at hand then the Gfrerer/Raus-method will also yield good results, while the discrepancy principle consistently yields bad results.

We conclude the discussion of the image reconstruction problem by giving Fig. 3 which shows the exact solution, together with three regularized solutions computed by the three combinations that we advocated above (having in mind that a good estimate of $\|\varepsilon\|$ is difficult to obtain). Both surface and contour plots are presented. As expected from the use of $B = I$, we cannot reconstruct the step functions in the image very well without introducing some oscillations in the regularized solutions. However, notice from the contour plots that the location of the steps are quite well reconstructed.

In conclusion, we wish to stress that our observations in this section are only valid for the particular model problems that we have studied. Things are very likely to be quite different for other ill-posed problems. What we have presented here are mainly examples that illustrate how difficult the "best" choice of regularization method really is. We also mention that the above conclusions may change in the future when more work with the "hybrid" LSQR algorithm has been done.

Acknowledgement

We are grateful to Jørgen Christensen-Dalsgaard and Jim Nagy for helpful discussions concerning the two model problems of Sect. 8. Thanks also go to Åke Björck, Andreas Neubauer, Robert Plato, and Curt Vogel for their careful reading of the first draft of this paper. This project was supported by the Danish Natural Science Research Council and by the Deutsche Forschungsgemeinschaft.

References

1. Andrews, H.C., Hunt, B.R.: Digital image restoration. Prentice-Hall, Englewood Cliffs (1977).
2. Backus, G., Gilbert, F.: Numerical applications of a formalism for geophysical inverse problems. Geophys. J. R. Astron. Soc. 13: 247–276 (1967).
3. Baker, C.T.H.: The numerical treatment of integral equations. Clarendon Press, Oxford (1977).
4. Bai, Z., Demmel, J.W.: Computing the generalized singular value decomposition. Dept. of Mathematics, University of Kentucky, Lexington, Report 91-09 (1991) (submitted to SIAM J. Sci. Stat. Comput.).
5. Bates, D.M., Lindstrom, M.J., Wahba, G., Yandell, B.S.: GCVPACK-routines for generalized cross validation. Commun. Statist. Simul. 16: 263–297 (1987).

6. Baumeister, J.: Stable solution of inverse problems. Vieweg, Braunschweig (1987).
7. Bednar, J.B., Lines, L.R., Stolt, R.H., Weglein, A.B.: Geophysical inversion. SIAM, Philadelphia (1992).
8. Bertero, M., Brianzi, P., Pike, E.R.: Super-resolution in confocal scanning microscopy. Inverse Problems 3: 195–212 (1987).
9. Bertero, M., De Mol, C., Pike, E.R.: Applied inverse problems in optics. In: Engl, H.W., Groetsch, C.W. (eds.): Inverse and ill-posed problems. Academic Press, Boston, pp. 291–313 (1987).
10. Björck, A.: A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. BIT 28: 659–670 (1988).
11. Björck, A.: Least squares methods. In: Ciarlet, P.G., Lions, J.L. (eds.): Handbook of numerical analysis, vol. I: finite difference methods. Solutions of equations in $\mathbb{R}^n$. Elsevier North-Holland, Amsterdam, pp. 465–652 (1990).
12. Björck, A., Eldén, L.: Methods in numerical analysis for ill-posed problems. Linköping University, Linköping, Tech. Report LiTH-MAT-R-33-1979 (1979).
13. Brakhage, H.: On ill-posed problems and the method of conjugate gradients. In, Engl, H.W., Groetsch, C.W. (eds.): Inverse and ill-posed problems. Academic Press, Boston, pp. 165–175 (1987).
14. Chan, R.H., Nagy, J.G., Plemmons, R.J.: FFT-based preconditioners for Toeplitz-block least squares problems. SIAM J. Numer. Anal. (to appear).
15. Christensen-Dalsgaard, J., Hansen, P.C., Thompson, M.J.: GSVD analysis of helioseismic inversions. Inst. for Theoretical Physics, University of California, Santa Barbara (1992) (submitted to Mon. Not. R. Astron. Soc.).
16. Christensen-Dalsgaard, J., Schou, J., Thompson, M.J.: A comparison of methods for inverting helioseismic data. Mon. Not. R. Astron. Soc. 242: 353–369 (1990).
17. Chun, J., Kailath, T.: Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices. In: Golub, G.H., Van Dooren, P. (eds.): Numerical linear algebra, digital signal processing and parallel algorithms. Springer, Berlin Heidelberg New York Tokyo, pp. 215–236 (1991).
18. Cullum, J.: The effective choice of the smoothing norm in regularization. Math. Comp. 33: 149–170 (1979).
19. Cuppen, J.J.M.: Regularization methods and parameter estimation methods for the solution of Fredholm integral equations of the first kind. In: te Riele, H.J.J. (ed.): Colloquium: Numerical treatment of integral equations. Mathematisch Centrum, Amsterdam, pp. 67–85 (1979).
20. De Moor, B., Zha, H.: A tree of generalizations of the ordinary singular value decomposition. Linear Algebra Appl. 147: 469–500 (1991).
21. Dongarra, J., Duff, I.S., Sorensen, D.C., Van der Vorst, H.A.: Solving linear systems on vector and shared memory computers. SIAM, Philadelphia (1991).
22. Drake, J.B.: ARIES: a computer program for the solution of first kind integral equations with noisy data. Dept. of Computer Science, Oak Ridge National Laboratory, Report K/CSD/TM-43, (1983).
23. Eicke, B., Louis, A.K., Plato, R.: The instability of some gradient methods for ill-posed problems. Numer. Math. 58: 129–134. (1990).
24. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT 17: 134–145 (1977).
25. Eldén, L.: A weighted pseudoinverse, generalized singular values, and constrained least squares problems. BIT 22: 487–502 (1982).
26. Eldén, L.: An efficient algorithm for the regularization of ill-conditioned least squares problems with triangular Toeplitz matrix. SIAM J. Sci. Statist. Comput. 5: 229–236 (1984).
27. Eldén, L.: A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems. BIT 24: 467–472 (1984).
28. Engl, H.W.: Regularization methods for the stable solution of inverse problems. Surv. Math. Ind. 3: 71–143 (1993).
29. Engl, H.W., Neubauer, A.: Convergence rates for Tikhonov regularization in finite-dimensional subspaces of Hilbert scales. Proc. Amer. Math. Soc. 102: 587–592 (1988).
30. Franklin, J.N.: Well-posed stochastic extensions of ill-posed linear problems. J. Math. Anal. Appl. 31: 682–716 (1970).
31. Gfrerer, H.: An a posteriori parameter choice for ordinary and iterated Tikhonov regularization of ill-posed problems leading to optimal convergence rates. Math. Comput. 49: 507–522 (1987).
32. Gilliam, D.S., Lund, J.R., Vogel, C.R.: Quantifying information content for ill-posed problems. Inverse Problems 6: 725–736 (1990).
33. Girard, D.A.: A fast 'Monte-Carlo cross-validation' procedure for large least squares problems with noisy data. Numer. Math. 56: 1–23 (1989).
34. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21: 215–223 (1979).
35. Golub, G.H., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. SIAM J. Numer. Anal. 2: 205–224 (1965).
36. Golub, G.H., Van Loan, C.F.: Matrix computations. The Johns Hopkins University Press, Baltimore (1989).
37. Golub, G.H., von Matt, U.: Quadratically constrained least-squares and quadratic problems. Numer. Math. 59: 561–580 (1991).
38. Groetsch, C.W.: Generalized inverses of linear operators. Marcel Dekker, New York (1977).
39. Groetsch, C.W.: The theory of Tikhonov regularization for Fredholm equations of the first kind. Pitman, Boston (1984).
40. Groetsch, C.W.: Convergence analysis of a-regularized degenerate kernel method for Fredholm integral equations of the first kind. Integral Equ. Operator Theory 13: 67–75 (1990).
41. Hadamard, J.: Lectures on Cauchy's problem in linear partial differential equations. Yale University Press, New Haven (1923).
42. Hanke, M.: Accelerated Landweber iterations for the solution of ill-posed equations. Numer. Math. 60: 341–373 (1991).
43. Hanke, M.: Regularization with differential operators: an iterative approach. Numer. Funct. Anal. Optim. 13: 523–540 (1992).
44. Hanke, M.: An ε-free a posteriori stopping rule for certain iterative regularization methods. SIAM J. Numer. Anal. 30 (to appear).
45. Hanke, M.: Iterative solution of underdetermined linear systems by transformation to standard form. (submitted).
46. Hanke, M., Nagy, J. G., Plemmons, R.J.: Preconditioned iterative regularization for ill-posed problems. In: Reichel, L., Ruttan, A., Varga, R.S. (eds.): Numerical linear algebra. de Gruyter, Berlin, pp. 141–163 (1993).
47. Hansen, P.C.: The truncated SVD as a method for regularization. BIT 27: 534–553 (1987).
48. Hansen, P.C.: The discrete Picard condition for discrete ill-posed problems. BIT 30: 658–672 (1990).
49. Hansen, P.C.: Analysis of discrete ill-posed problems by means of the L-curve. SIAM Rev. 34: 561–580 (1992).
50. Hansen, P.C.: Regularization tools, a Matlab package for analysis and solution of discrete ill-posed problems. Numer. Algorithms (to appear).
51. Hansen, P.C., O'Leary, D.P.: The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comput. (to appear).
52. Hegland, T.M.: Numerische Lösung von Fredholmschen Integralgleichungen erster Art bei ungenauen Daten. PhD thesis, ETH Zürich, Zürich (1988).
53. Herman, G.T.: Image reconstruction from projections. Academic Press, New York (1980).
54. Higham, N.J.: Analysis of the Cholesky decomposition of a semi-definite matrix. In: Cox, M.G. and Hammarling, S. (eds.): Reliable numerical computation. Clarendon Press, Oxford, pp. 161–185 (1990).
55. Hilgers, J.W.: On the equivalence of regularization and certain reproducing kernel Hilbert space approaches for solving first kind problems. SIAM J. Numer. Anal. 13: 172–184 (1976).
56. Hofmann, B.: Regularization for applied inverse and ill-posed problems. B.S.B. Teubner, Leipzig (1986).
57. Jain, A.K.: Fundamentals of digital image processing. Prentice-Hall, Englewood Cliffs (1989).
58. Kailath, T., Kung, S.-Y., Morf, M.: Displacement ranks of matrices and linear equations. J. Math. Anal. Appl. 68: 395–407 (1979).
59. King, J.T.: On the construction of preconditioners by subspace decomposition. J. Comput. Appl. Math. 29: 195–205 (1990).
60. King, J.T.: Multilevel algorithms for ill-posed problems. Numer. Math. 61: 311–334 (1992).
61. King, J.T., Chillingworth, D.: Approximation of generalized inverses by iterated regularization. Numer. Funct. Anal. Optim. 1: 499–513 (1979).
62. King, J. T., Neubauer, A.: A variant of finite-dimensional Tikhonov regularization with a-posteriori parameter choice. Computing 40: 91–109 (1988).
63. Kress, R.: Linear integral equations. Springer, Berlin Heidelberg New York Tokyo (1989).
64. Landweber, L.: An iteration formula for Fredholm integral equations of the first kind. Amer. J. Math. 73: 615–624 (1951).
65. Lawson, C.L., Hanson, R.J.: Solving least squares problems. Prentice-Hall, Englewood Cliffs (1974).
66. Linz, P.: Uncertainty in the solution of linear operator equations. BIT 24: 92–101 (1984).

67. Locker, J. Prenter, P.M.: Regularization with differential operators. I. General theory. J. Math. Anal. Appl. 74: 504–529 (1980).
68. Louis, A.K.: Convergence of the conjugate gradient method for compact operators. In: Engl, H.W., Groetsch, C.W. (eds.): Inverse and ill-posed problems. Academic Press, Boston, pp. 177–183 (1987).
69. Louis, A.K. Inverse und schlecht gestellte Probleme. B.G. Teubner, Stuttgart (1989).
70. Louis, A.K., Maass, P.: A mollifier method for linear operator equations of the first kind. Inverse Problems 6: 427–440 (1990).
71. Lukas, M.A.: Asymptotic optimality of generalized cross-validation for choosing the regularization parameter. (submitted).
72. Marti, J.T.: An algorithm for computing minimum norm solutions of Fredholm integral equations of the first kind. SIAM J. Numer. Anal. 15: 1071–1076 (1978).
73. Melkman, A.A., Micchelli, C.A.: Optimal estimation of linear operators in Hilbert spaces from inaccurate data. SIAM J. Numer. Anal. 16: 87–105 (1979).
74. Miller, K.: Least squares methods for ill-posed problems with a prescribed bound. SIAM J. Math. Anal. 1: 52–74 (1970).
75. Morozov, V.A.: On the solution of functional equations by the method of regularization. Soviet Math. Dokl. 7: 414–417 (1966).
76. Morozov, V.A.: Methods for solving incorrectly posed problems. Springer, New York Berlin Heidelberg (1984).
77. Nagy, J.G.: Fast inverse QR factorization for Toeplitz matrices. SIAM J. Sci. Statist. Comput. (to appear).
78. Natterer, F.: Regularisierung schlecht gestellter Probleme durch Projektionsverfahren. Numer. Math. 28: 329–341 (1977).
79. Natterer, F.: The finite element method for ill-posed problems. RAIRO Anal. Numér. 11: 271–278 (1977).
80. Natterer, F.: Error bounds for Tikhonov regularization in Hilbert scales. Appl. Anal. 18: 29–37 (1984).
81. Natterer, F.: The mathematics of computerized tomography. Wiley, Chichester (1986).
82. Natterer, F.: Numerical treatment of ill-posed problems. In: Talenti, G. (ed.): Inverse problems. Springer, New York Berlin Heidelberg Tokyo, pp. 165–175 (1987).
83. Nemirovskii, A.S.: The regularization properties of the adjoint gradient method in ill-posed problems. USSR Comput. Math. Math. Phys. 26 2: 7–16 (1986).
84. Nemirovskii, A.S., Polyak, B.T.: Iterative methods for solving linear ill-posed problems under precise information I. Eng. Cybernetics 22 3: 1–11 (1984).
85. Neubauer, A.: Numerical realization of an optimal discrepancy principle for Tikhonov-regularization in Hilbert scales. Computing 39: 43–55 (1987).
86. O'Leary, D.P., Simmons, J.A.: A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. SIAM J. Sci. Statist. Comput. 2: 474–489 (1981).
87. Osaki, Y., Shibahashi, H. (eds.): Progress of seismology of the sun and stars. Springer, Berlin Heidelberg New York Tokyo (1990).
88. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Software 8: 43–71 (1982).
89. Paige, C.C., Saunders, M.A.: Algorithm 583. LSQR: sparse linear equations and least squares problems. ACM Trans. Math. Software 8: 195–209 (1982).
90. Plato, R.: Optimal algorithms for linear ill-posed problems yield regularization methods. Numer. Funct. Anal. Optim. 11: 111–118 (1990).
91. Plato, R., Vainikko, G.: On the regularization of projection methods for solving ill-posed problems. Numer. Math. 57: 63–79 (1990).
92. Raus, T.: The principle of the residual in the solution of ill-posed problems with nonselfadjoint operator. Uchen. Zap. Tartu Gos. Univ. 715: 12–20 (1985) (in Russian).
93. Riley, J.D.: Solving systems of linear equations with a positive definite, symmetric, but possibly ill-conditioned matrix. Math. Tables Aids Comput. 9: 96–101 (1955).
94. Schock, E.: Implicite iterative methods for the approximate solution of ill-posed problems. Boll. UMI 7: 1171–1184 (1987).
95. Schock, E.: Semi-iterative methods for the approximate solution of ill-posed problems. Numer. Math. 50: 263–271 (1987).
96. Seidman, T.I.: Nonconvergence results for the application of least-squares estimation to ill-posed problems. J. Optim. Theory Appl. 30: 535–547 (1980).
97. Smithies, F.: Integral equations. Cambridge University Press, London (1962).
98. Strand, O.N.: Theory and methods related to the singular-function expansion and Landweber's iteration for integral equations of the first kind. SIAM J. Numer. Anal. 11: 798–825 (1974).
99. Szegö, G.: Orthogonal polynomials. American Mathematical Society, Providence (American Mathematical Society Colloquium Publications vol. 23) 1975.
100. Te Riele, J.J.J.: A program for solving first kind Fredholm integral equations by means of regularization. Comput. Phys. Comm. 36: 423–432 (1985).
101. Tikhonov, A.N.: Solution of incorrectly formulated problems and the regularization method. Soviet Math. Dokl. 4: 1035–1038 (1963).
102. Tikhonov, A.N.: Regularization of incorrectly posed problems. Soviet Math. Dokl. 4: 1624–1627 (1963).
103. Tikhonov, A.N., Arsenin, V.Y.: Solutions of ill-posed problems. Wiley, New York (1977).
104. Tikhonov, A.N., Glasko, V.B.: Use of the regularization method in non-linear problems. USSR Comput. Math. Math. Phys. 5 3: 93–107 (1965).
105. Tikhonov, A.N., Glasko, V.B., Kriksin, Y.A.: On the question of quasi-optimal choice of a regularized approximation. Soviet Math. Dokl. 20: 1036–1040 (1979).
106. Tikhonov, A.N., Goncharsky, A.V.: Ill-posed problems in the natural sciences. MIR, Moscow (1987).
107. Vainikko, G.M., Haemarik, U.: Projection methods and self-regularization in ill-posed problems. Soviet. Math. Iz. VUZ 29 10: 1–20 (1985).
108. Van der Sluis, A., Van der Vorst, H.A.: SIRT and CG type methods for the iterative solution of sparse linear least squares problems. Linear Algebra Appl. 130: 257–303 (1990).
109. Van Loan, C.F.: Generalizing the singular value decomposition. SIAM J. Numer. Anal. 13: 76–83 (1976).
110. Van Loan, C.F.: Computational frameworks for the fast Fourier transform. SIAM, Philadelphia (1992).
111. Varah, J.M.: A practical examination of some numerical methods for linear discrete ill-posed problems. SIAM Rev. 21: 100–111 (1979).
112. Varah, J.M.: Pitfalls in the numerical solution of linear ill-posed problems. SIAM J. Sci. Statist. Comput. 4: 164–176 (1983).
113. Vogel, C.R.: Solving ill-conditioned linear systems using the conjugate gradient method. Dept. of Mathematical Sciences, Montana State University, Bozeman, Manuscript (1987).
114. Vogel, C.R., Wade, J.G.: Iterative SVD-based methods for ill-posed problems. SIAM J. Sci. Comput. (to appear).
115. Wahba, G.: Practical approximate solutions to linear operator equations when the data are noisy. SIAM J. Numer. Anal. 14: 651–667 (1977).
116. Wahba, G.: Three topics in ill-posed problems. In: Engl, H.W., Groetsch, C.W. (eds.): Inverse and ill-posed problems. Academic Press, Boston, pp. 37–51 (1987).
117. Wahba, G.: Spline models for observational data. SIAM, Philadelphia (1990).
118. Wing, G.M.: A primer on integral equations of the first kind. SIAM, Philadelphia (1991).

Authors' addresses: Dr. M. Hanke, Institut für Praktische Mathematik, Universität Karlsruhe, Englerstrasse 2, D-76128 Karlsruhe, Federal Republic of Germany.—Prof. P.C. Hansen, The Danish Computing Center for Research and Education, Technical University of Denmark, Lyngby, Denmark.

Verleger: Springer-Verlag KG, Sachsenplatz 4-6, A-1201 Wien. – Herausgeber: Prof. Dipl.-Ing. Dr. H. Engl, Institut für Mathematik, Johannes Kepler Universität Linz, A-4040 Linz. – Redaktion: Altenbergerstraße 69, A-4040 Linz. – Hersteller: Satz und Umbruch: Macmillan India Ltd., Bangalore-25. Offsetdruck: Novographic, Ing. W. Schmid, Maurer Lange Gasse 64, A-1238 Wien. – Verlagsort: Wien. – Herstellungsort: Wien. – Printed in Austria.