5

Iterative regularization methods for linear problems

The iterative solution of linear systems of equations arising from the discretization of ill-posed problems is the method of choice when the dimension of the problem is so large that factorization of the matrix is either too time-consuming or too memory-demanding.

The ill-conditioning of the coefficient matrix for these linear systems is so extremely large that some sort of regularization is needed to guarantee that the computed solution is not dominated by errors in the data. In the framework of iterative methods, the regularizing effect is obtained by stopping the iteration prior to convergence to the solution of the linear system. This form of regularization is referred to as regularization by truncated iteration. The idea behind regularization by truncated iteration is that in the first few iteration steps, the iterated solution includes the components $[(\mathbf{u}_i^T \mathbf{y}^\delta)/\sigma_i]\,\mathbf{v}_i$ corresponding to the largest singular values and approaches a regularized solution. As the iteration continues, the iterated solution is dominated by amplified noise components and converges to some undesirable solution (often the least squares solution). This phenomenon is referred to as semi-convergence. In this context, it is apparent that the iteration index plays the role of the regularization parameter, and a stopping rule plays the role of a parameter choice method.

In this chapter we first review some classical iterative methods and then focus on the conjugate gradient method and a related algorithm based on Lanczos bidiagonalization. The classical iterative methods to be discussed include the Landweber iteration and semi-iterative methods.

5.1 Landweber iteration

The Landweber iteration is based on the transformation of the normal equation

$$\mathbf{K}^T \mathbf{K}\mathbf{x} = \mathbf{K}^T \mathbf{y}^\delta$$

into an equivalent fixed point equation

$$\mathbf{x} = \mathbf{x} + \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}\right),$$

that is

$$\mathbf{x}_k^\delta = \mathbf{x}_{k-1}^\delta + \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right), \quad k = 1, 2, \ldots. \tag{5.1}$$

The slight inconvenience with the Landweber iteration is that it requires the norm of $\mathbf{K}$ to be less than or equal to one, otherwise the method either diverges or converges too slowly. If this is not the case, we introduce a relaxation parameter χ, chosen as $0 < \chi \leq \|\mathbf{K}\|^{-1}$, to obtain

$$\mathbf{x}_k^\delta = \mathbf{x}_{k-1}^\delta + \chi^2 \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right), \quad k = 1, 2, \ldots.$$

This construction has the same effect as multiplying the equation $\mathbf{K}\mathbf{x} = \mathbf{y}^\delta$ by χ and iterating with (5.1). In the present analysis we assume that the problem has been scaled appropriately, so that $\|\mathbf{K}\| \leq 1$, and drop the relaxation parameter χ.

The initial guess $\mathbf{x}_0^\delta = \mathbf{x}_\mathsf{a}$ plays the same role as in Tikhonov regularization: it selects the particular solution which will be obtained in the case of ambiguity. The iterate $\mathbf{x}_k^\delta$ can be expressed non-recursively through

$$\mathbf{x}_k^\delta = \mathbf{M}^k \mathbf{x}_0^\delta + \sum_{l=0}^{k-1} \mathbf{M}^l \mathbf{K}^T \mathbf{y}^\delta, \tag{5.2}$$

where

$$\mathbf{M} = \mathbf{I}_n - \mathbf{K}^T\mathbf{K}.$$

This result can be proven by induction. For $k = 1$, there holds

$$\mathbf{x}_1^\delta = \mathbf{x}_0^\delta + \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_0^\delta\right) = \mathbf{M}\mathbf{x}_0^\delta + \mathbf{K}^T\mathbf{y}^\delta,$$

while under assumption (5.2), we obtain

$$\mathbf{x}_{k+1}^\delta = \mathbf{x}_k^\delta + \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_k^\delta\right) = \mathbf{M}\mathbf{x}_k^\delta + \mathbf{K}^T\mathbf{y}^\delta = \mathbf{M}^{k+1}\mathbf{x}_0^\delta + \sum_{l=0}^{k} \mathbf{M}^l \mathbf{K}^T \mathbf{y}^\delta.$$

To obtain more transparent results concerning the regularizing property of the Landweber iteration, we assume that $\mathbf{x}_0^\delta = \mathbf{0}$. Using the result

$$\mathbf{M}^l \mathbf{K}^T \mathbf{y}^\delta = \sum_{i=1}^{n} \left(1 - \sigma_i^2\right)^l \sigma_i \left(\mathbf{u}_i^T \mathbf{y}^\delta\right) \mathbf{v}_i, \quad l \geq 0,$$

where $(\sigma_i; \mathbf{v}_i, \mathbf{u}_i)$ is a singular system of $\mathbf{K}$, we deduce that the iterate $\mathbf{x}_k^\delta$ can be expressed as

$$\mathbf{x}_k^\delta = \sum_{i=1}^{n} \left[1 - \left(1 - \sigma_i^2\right)^k\right] \frac{1}{\sigma_i} \left(\mathbf{u}_i^T \mathbf{y}^\delta\right) \mathbf{v}_i, \tag{5.3}$$

and the regularized solution for the exact data vector $\mathbf{y}$ as

$$\mathbf{x}_k = \sum_{i=1}^{n} \left[1 - \left(1 - \sigma_i^2\right)^k\right] \frac{1}{\sigma_i} \left(\mathbf{u}_i^T \mathbf{y}\right) \mathbf{v}_i.$$

Accounting for the expression of the exact solution $\mathbf{x}^\dagger$,

$$\mathbf{x}^\dagger = \sum_{i=1}^{n} \frac{1}{\sigma_i} \left(\mathbf{u}_i^T \mathbf{y}\right) \mathbf{v}_i,$$

we find that the smoothing error norm is given by

$$\|\mathbf{e}_{\mathtt{s}k}\|^2 = \left\|\mathbf{x}^\dagger - \mathbf{x}_k\right\|^2 = \sum_{i=1}^{n} \left(1-\sigma_i^2\right)^{2k} \frac{1}{\sigma_i^2} \left(\mathbf{u}_i^T \mathbf{y}\right)^2. \tag{5.4}$$

Since by assumption $\|\mathbf{K}\| \leq 1$, it follows that $\sigma_i \leq 1$ for all $i = 1, \ldots, n$, and therefore, $\|\mathbf{e}_{\mathtt{s}k}\| \to 0$ as $k \to \infty$. On the other hand, the noise error norm

$$\|\mathbf{e}_{\mathtt{n}k}\|^2 = \left\|\mathbf{x}_k - \mathbf{x}_k^\delta\right\|^2 = \sum_{i=1}^{n} \left[1 - \left(1-\sigma_i^2\right)^k\right]^2 \frac{1}{\sigma_i^2} \left(\mathbf{u}_i^T \boldsymbol{\delta}\right)^2 \tag{5.5}$$

converges to

$$\left\|\mathbf{K}^\dagger \boldsymbol{\delta}\right\|^2 = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left(\mathbf{u}_i^T \boldsymbol{\delta}\right)^2$$

as $k \to \infty$. Since $\mathbf{K}$ possesses small singular values, the noise error is extremely large in this limit. The noise error can be estimated by using the inequality

$$\sup_{0 \leq x \leq 1} \frac{1 - \left(1-x^2\right)^k}{x} \leq \sqrt{k}, \quad k \geq 1,$$

and the result is

$$\|\mathbf{e}_{\mathtt{n}k}\|^2 \leq k\Delta^2. \tag{5.6}$$

From (5.4) and (5.6), we see that the smoothing error converges slowly to 0, while the noise error is of the same order of at most $\sqrt{k}\Delta$. For small values of k, the noise error is negligible and the iterate $\mathbf{x}_k^\delta$ seems to converge to the exact solution $\mathbf{x}^\dagger$. When $\sqrt{k}\Delta$ reaches the order of magnitude of the smoothing error, the noise error is no longer covered in $\mathbf{x}_k^\delta$ and the approximation changes to worse. This semi-convergent behavior requires a reliable stopping rule for detecting the transition from convergence to divergence.

The regularizing effect of the Landweber iteration is reflected by the filter factors of the computed solution. From (5.3), we infer that the kth iterate can be expressed as

$$\mathbf{x}_k^\delta = \sum_{i=1}^{n} f_k\left(\sigma_i^2\right) \frac{1}{\sigma_i} \left(\mathbf{u}_i^T \mathbf{y}^\delta\right) \mathbf{v}_i,$$

with the filter factors being given by

$$f_k\left(\sigma_i^2\right) = 1 - \left(1-\sigma_i^2\right)^k.$$

For $\sigma_i \ll 1$, we have $f_k\left(\sigma_i^2\right) \approx k\sigma_i^2$, while for $\sigma_i \approx 1$, there holds $f_k\left(\sigma_i^2\right) \approx 1$. Thus, for small values of k, the contributions of the small singular values to the solution are effectively filtered out, and when k increases, more components corresponding to small singular values are included in the solution. Therefore, an optimal value of k should reflect a trade-off between accuracy and stability.

5.2 Semi-iterative regularization methods

The major drawback of the Landweber iteration is its slow rate of convergence, this means, too many iterations are required to reduce the residual norm to the order of the noise level. More sophisticated methods have been developed on the basis of the so-called semi-iterative methods.

To introduce semi-iterative methods, we consider again the Landweber iteration and define the function $g_k(\lambda)$ in terms of the filter function

$$f_k(\lambda) = 1 - (1-\lambda)^k$$

by the relation

$$g_k(\lambda) = \frac{1}{\lambda} f_k(\lambda) = \frac{1}{\lambda}\left[1 - (1-\lambda)^k\right]. \tag{5.7}$$

In terms of g_k, the Landweber iterate reads as

$$\mathbf{x}_k^\delta = g_k\left(\mathbf{K}^T\mathbf{K}\right)\mathbf{K}^T\mathbf{y}^\delta, \tag{5.8}$$

where

$$g_k\left(\mathbf{K}^T\mathbf{K}\right) = \mathbf{V}\left[\operatorname{diag}\left(g_k\left(\sigma_i^2\right)\right)_{n\times n}\right]\mathbf{V}^T.$$

Evidently, $g_k(\lambda)$ is a polynomial of degree $k-1$, which converges pointwise to $1/\lambda$ on $(0,1]$ as $k \to \infty$. This property guarantees that in the noise-free case, the regularized solution converges to the exact solution, that is, $\lim_{k\to\infty}\left\|\mathbf{x}_k - \mathbf{x}^\dagger\right\| = 0$, where $\mathbf{x}_k = g_k\left(\mathbf{K}^T\mathbf{K}\right)\mathbf{K}^T\mathbf{y}$.

Any sequence of polynomials $\{g_k\}$, with g_k having the degree $k-1$, defines a semi-iterative method. The idea is that polynomials g_k different from the one given by (5.7) may converge faster to $1/\lambda$, and may thus lead to accelerated Landweber methods. In the case of semi-iterative methods, the polynomials g_k are called iteration polynomials, while the polynomials

$$r_k(\lambda) = 1 - \lambda g_k(\lambda)$$

are called residual polynomials. The residual polynomials are uniformly bounded on $[0,1]$ and converge pointwise to 0 on $(0,1]$ as $k \to \infty$. In addition, they are normalized in the sense that $r_k(0) = 1$.

If the residual polynomials form an orthogonal sequence with respect to some measure over $\mathbb{R}_+$, then they satisfy the three-term recurrence relation

$$r_k(\lambda) = r_{k-1}(\lambda) + \mu_k\left[r_{k-1}(\lambda) - r_{k-2}(\lambda)\right] - \omega_k\lambda r_{k-1}(\lambda), \quad k \geq 2. \tag{5.9}$$

By virtue of (5.9) and taking into account that

$$\mathbf{x}_k^\delta = \sum_{i=1}^{n}\left[1 - r_k\left(\sigma_i^2\right)\right]\frac{1}{\sigma_i}\left(\mathbf{u}_i^T\mathbf{y}^\delta\right)\mathbf{v}_i$$

and that

$$\mathbf{K}^T\left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right) = \sum_{i=1}^{n}\left[\sigma_i^2 r_{k-1}\left(\sigma_i^2\right)\right]\frac{1}{\sigma_i}\left(\mathbf{u}_i^T\mathbf{y}^\delta\right)\mathbf{v}_i,$$

we deduce that the iterates of the associated semi-iterative method satisfy the recurrence relation

$$\mathbf{x}_k^\delta = \mathbf{x}_{k-1}^\delta + \mu_k \left(\mathbf{x}_{k-1}^\delta - \mathbf{x}_{k-2}^\delta\right) + \omega_k \mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right), \quad k \geq 2. \tag{5.10}$$

Note that because the kth iterate does not depend only on the $(k-1)$th iterate, the iterative approach (5.10) is termed semi-iterative. As in the case of the Landweber iteration, $\mathbf{K}$ must be scaled so that $\|\mathbf{K}\| \leq 1$, and for this reason, systems of polynomials defined on the interval $[0, 1]$ have to be considered.

The Chebyshev method of Stiefel uses the residual polynomials (Rieder, 2003)

$$r_k(\lambda) = \frac{U_k(1 - 2\lambda)}{k+1},$$

where U_k are the Chebyshev polynomials of the second kind

$$U_k(\lambda) = \frac{\sin\left((k+1)\arccos\lambda\right)}{\sin\left(\arccos\lambda\right)}.$$

Due to the orthogonality of U_k in the interval $[-1, 1]$ with respect to the weight function $\sqrt{1-\lambda^2}$, it follows that the r_k are orthogonal in the interval $[0, 1]$ with respect to the weight function $\sqrt{\lambda/(1-\lambda)}$. The three-term recurrence relation reads as

$$\mathbf{x}_k^\delta = \frac{2k}{k+1}\mathbf{x}_{k-1}^\delta - \frac{k-1}{k+1}\mathbf{x}_{k-2}^\delta + \frac{4k}{k+1}\mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right), \quad k \geq 2,$$

with

$$\mathbf{x}_1^\delta = \mathbf{x}_0^\delta + 2\mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_0^\delta\right).$$

In the Chebyshev method of Nemirovskii and Polyak (1984), the residual polynomials are given by

$$r_k(\lambda) = \frac{(-1)^k T_{2k+1}\left(\sqrt{\lambda}\right)}{(2k+1)\sqrt{\lambda}},$$

where T_k are the Chebyshev polynomials of the first kind

$$T_k(\lambda) = \cos\left(k \arccos\lambda\right).$$

As before, the orthogonality of T_k in the interval $[-1, 1]$ with respect to the weight function $1/\sqrt{1-\lambda^2}$ implies the orthogonality of the r_k in the interval $[0, 1]$ with respect to the weight function $\sqrt{\lambda/(1-\lambda)}$. The recursion of the Chebyshev method of Nemirovskii and Polyak takes the form

$$\mathbf{x}_k^\delta = 2\frac{2k-1}{2k+1}\mathbf{x}_{k-1}^\delta - \frac{2k-3}{2k+1}\mathbf{x}_{k-2}^\delta + 4\frac{2k-1}{2k+1}\mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right), \quad k \geq 2,$$

with

$$\mathbf{x}_1^\delta = \frac{2}{3}\mathbf{x}_0^\delta + \frac{4}{3}\mathbf{K}^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_0^\delta\right).$$

The ν-method of Brakhage (1987) uses the residual polynomials

$$r_{\nu k}(\lambda) = \frac{P_k^{\left(2\nu-\frac{1}{2},-\frac{1}{2}\right)}(1-2\lambda)}{P_k^{\left(2\nu-\frac{1}{2},-\frac{1}{2}\right)}(1)},$$

where $P_k^{(\alpha,\beta)}$are the Jacobi polynomials. The parameter ν is fixed and is chosen as $0 < \nu < 1$. The orthogonality of the Jacobi polynomials in the interval $[-1,1]$ with respect to the weight function $(1-\lambda)^{\alpha}(1+\lambda)^{\beta}$, where $\alpha > -1$ and $\beta > -1$, yields the orthogonality of the residual polynomials in the interval $[0,1]$ with respect to the weight function $\lambda^{2\nu+1/2}(1-\lambda)^{-1/2}$. The three-term recurrence relation of the Jacobi polynomials leads to the following recursion of the ν-method

$$\mathbf{x}_k^{\delta} = \mathbf{x}_{k-1}^{\delta} + \mu_k\left(\mathbf{x}_{k-1}^{\delta} - \mathbf{x}_{k-2}^{\delta}\right) + \omega_k \mathbf{K}^T\left(\mathbf{y}^{\delta} - \mathbf{K}\mathbf{x}_{k-1}^{\delta}\right), \quad k \geq 2,$$

with

$$\mathbf{x}_1^{\delta} = \mathbf{x}_0^{\delta} + \omega_1 \mathbf{K}^T\left(\mathbf{y}^{\delta} - \mathbf{K}\mathbf{x}_0^{\delta}\right)$$

and

$$\mu_k = \frac{(k-1)(2k-3)(2k+2\nu-1)}{(k+2\nu-1)(2k+4\nu-1)(2k+2\nu-3)}, \quad k \geq 2,$$
$$\omega_k = 4\frac{(2k+2\nu-1)(k+\nu-1)}{(k+2\nu-1)(2k+4\nu-1)}, \quad k \geq 1.$$

5.3 Conjugate gradient method

Semi-iterative regularization methods are much more efficient than the classical Landweber iteration but require the scaling of $\mathbf{K}$. The conjugate gradient method due to Hestenes and Stiefel (1952) is scaling-free and is faster than any other semi-iterative method.

The conjugate gradient method is applied to the normal equation

$$\mathbf{K}^T\mathbf{K}\mathbf{x} = \mathbf{K}^T\mathbf{y}^{\delta}$$

of an ill-posed problem, in which case, the resulting algorithm is known as the conjugate gradient for normal equations (CGNR). In contrast to other iterative regularization methods, CGNR is not based on a fixed sequence of polynomials $\{g_k\}$ and $\{r_k\}$; these polynomials depend on the given right-hand side. This has the advantage of a greater flexibility of the method, but at the price of the iterates depending nonlinearly on the data,

$$\mathbf{x}_k^{\delta} = g_k\left(\mathbf{K}^T\mathbf{K}, \mathbf{y}^{\delta}\right)\mathbf{K}^T\mathbf{y}^{\delta}.$$

To formulate the CGNR method we first consider a preliminary definition. If $\mathbf{A}$ is a real $n \times n$ matrix and $\mathbf{x}$ is an element of $\mathbb{R}^n$, then the kth Krylov subspace $\mathcal{K}_k(\mathbf{x}, \mathbf{A})$ is defined as the linear space

$$\mathcal{K}_k(\mathbf{x}, \mathbf{A}) = \operatorname{span}\left\{\mathbf{x}, \mathbf{A}\mathbf{x}, \ldots, \mathbf{A}^{k-1}\mathbf{x}\right\}.$$

Using (5.8) and taking into account that g_k is a polynomial of degree $k-1$, we deduce that the kth iterate of any semi-iterative method belongs to the kth Krylov subspace

$$\mathcal{K}_k\left(\mathbf{K}^T\mathbf{y}^\delta, \mathbf{K}^T\mathbf{K}\right) = \text{span}\left\{\mathbf{K}^T\mathbf{y}^\delta, \left(\mathbf{K}^T\mathbf{K}\right)\mathbf{K}^T\mathbf{y}^\delta, \ldots, \left(\mathbf{K}^T\mathbf{K}\right)^{k-1}\mathbf{K}^T\mathbf{y}^\delta\right\}.$$

If rank $(\mathbf{K}) = r$, there holds

$$\left(\mathbf{K}^T\mathbf{K}\right)^{k-1}\mathbf{K}^T\mathbf{y}^\delta = \sum_{i=1}^{r}\sigma_i^{2(k-1)+1}\left(\mathbf{u}_i^T\mathbf{y}^\delta\right)\mathbf{v}_i, \;\; k \geq 1,$$

and we infer that

$$\mathcal{K}_k \subseteq \mathcal{N}\left(\mathbf{K}\right)^\perp = \text{span}\left\{\mathbf{v}_i\right\}_{i=\overline{1,r}}, \;\; k \geq 1, \tag{5.11}$$

where, for notation simplification, $\mathcal{K}_k$ stands for $\mathcal{K}_k\left(\mathbf{K}^T\mathbf{y}^\delta, \mathbf{K}^T\mathbf{K}\right)$.

The kth iterate of the CGNR method is defined as the minimizer of the residual norm in the corresponding Krylov subspace; assuming a zero initial guess, i.e., $\mathbf{x}_0^\delta = \mathbf{0}$, we have

$$\mathbf{x}_k^\delta = \arg\min_{\mathbf{x}_k \in \mathcal{K}_k}\left\|\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_k\right\|^2. \tag{5.12}$$

By virtue of (5.12) and the fact that the kth iterate of any semi-iterative belongs to $\mathcal{K}_k$, we may expect that CGNR requires the fewest iteration steps among all semi-iterative methods. Going further, we define the kth subspace

$$\mathcal{L}_k = \mathbf{K}\mathcal{K}_k = \left\{\mathbf{y}_k / \, \mathbf{y}_k = \mathbf{K}\mathbf{x}_k, \mathbf{x}_k \in \mathcal{K}_k\right\}, \tag{5.13}$$

and in view of (5.12), we consider the minimizer

$$\mathbf{y}_k^\delta = \arg\min_{\mathbf{y}_k \in \mathcal{L}_k}\left\|\mathbf{y}^\delta - \mathbf{y}_k\right\|. \tag{5.14}$$

The element $\mathbf{y}_k^\delta$ gives the best approximation of $\mathbf{y}^\delta$ among all elements of $\mathcal{L}_k$, that is,

$$\mathbf{y}_k^\delta = P_k\mathbf{y}^\delta, \tag{5.15}$$

where P_k is the orthogonal projection operator onto the (linear) subspace $\mathcal{L}_k$. The uniqueness of the orthogonal projection implies that $\mathbf{y}_k^\delta$ is uniquely determined and that

$$\mathbf{y}_k^\delta = \mathbf{K}\mathbf{x}_k^\delta. \tag{5.16}$$

If $\left\{\mathbf{u}_i\right\}_{i=\overline{1,k}}$ is an orthogonal basis of the (finite-dimensional) subspace $\mathcal{L}_k$, then $\mathbf{y}_k^\delta$ can be expressed as

$$\mathbf{y}_k^\delta = \sum_{i=1}^{k}\frac{\mathbf{u}_i^T\mathbf{y}^\delta}{\left\|\mathbf{u}_i\right\|^2}\mathbf{u}_i. \tag{5.17}$$

Let us now define the vectors

$$\mathbf{s}_k = \mathbf{K}^T\mathbf{r}_k^\delta, \;\; k \geq 0,$$

with $\mathbf{r}_0^\delta = \mathbf{y}^\delta$. As the residual vector at the kth iteration step,

$$\mathbf{r}_k^\delta = \mathbf{y}^\delta - \mathbf{y}_k^\delta = \left(\mathbf{I}_m - P_k\right)\mathbf{y}^\delta, \;\; k \geq 1, \tag{5.18}$$

is orthogonal to $\mathcal{L}_k$, the identity

$$\mathbf{s}_k^T \mathbf{x}_k = \left(\mathbf{K}^T \mathbf{r}_k^\delta\right)^T \mathbf{x}_k = \mathbf{r}_k^{\delta T} \mathbf{y}_k = 0, \tag{5.19}$$

which holds true for all $\mathbf{x}_k \in \mathcal{K}_k$ and $\mathbf{y}_k = \mathbf{K}\mathbf{x}_k \in \mathcal{L}_k$, yields

$$\mathbf{s}_k \perp \mathcal{K}_k, \;\; k \geq 1. \tag{5.20}$$

The finite-dimensional subspaces $\mathcal{K}_k$ and $\mathcal{L}_k$ can be characterized by appropriate orthogonal bases. For the kth Krylov subspace we note the following result: the system $\{\mathbf{s}_i\}_{i=\overline{0,k-1}}$ is an orthogonal basis of $\mathcal{K}_k$, that is,

$$\mathcal{K}_k = \operatorname{span}\{\mathbf{s}_i\}_{i=\overline{0,k-1}}, \;\; \mathbf{s}_i^T \mathbf{s}_j = \delta_{ij} \left\|\mathbf{s}_i\right\|^2, \;\; i,j = 0,\ldots,k-1. \tag{5.21}$$

This assertion can be proven by induction on k (Rieder, 2003). For $k = 1$, the result $\mathcal{K}_1 = \operatorname{span}\{\mathbf{s}_0\}$, with $\mathbf{s}_0 = \mathbf{K}^T \mathbf{y}^\delta$, is evidently true. Now, let us assume that (5.21) holds for k, i.e., $\mathcal{K}_k = \operatorname{span}\{\mathbf{s}_i\}_{i=\overline{0,k-1}}$, and let $\{\mathbf{u}_i\}_{i=\overline{1,k}}$ be an orthogonal basis of $\mathcal{L}_k$. As $\mathcal{L}_k = \mathbf{K}\mathcal{K}_k$, $\{\mathbf{u}_i\}_{i=\overline{1,k}}$ can be generated by orthogonalizing the set of vectors $\{\mathbf{K}\mathbf{s}_i\}_{i=\overline{0,k-1}}$. From (5.17), we have

$$\mathbf{y}_k^\delta = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{y}^\delta}{\left\|\mathbf{u}_i\right\|^2} \mathbf{u}_i = \mathbf{y}_{k-1}^\delta + \alpha_k \mathbf{u}_k, \;\; k \geq 1, \tag{5.22}$$

with $\mathbf{y}_0^\delta = \mathbf{0}$,

$$\mathbf{y}_{k-1}^\delta = P_{k-1}\mathbf{y}^\delta = \sum_{i=1}^{k-1} \frac{\mathbf{u}_i^T \mathbf{y}^\delta}{\left\|\mathbf{u}_i\right\|^2} \mathbf{u}_i,$$

and

$$\alpha_k = \frac{\mathbf{u}_k^T \mathbf{y}^\delta}{\left\|\mathbf{u}_k\right\|^2}. \tag{5.23}$$

Then, by (5.18) and (5.22), we obtain

$$\mathbf{r}_k^\delta = \mathbf{y}^\delta - \mathbf{y}_k^\delta = \left(\mathbf{y}^\delta - \mathbf{y}_{k-1}^\delta\right) - \alpha_k \mathbf{u}_k = \mathbf{r}_{k-1}^\delta - \alpha_k \mathbf{u}_k, \;\; k \geq 1, \tag{5.24}$$

and further,

$$\mathbf{s}_k = \mathbf{s}_{k-1} - \alpha_k \mathbf{K}^T \mathbf{u}_k, \;\; k \geq 1. \tag{5.25}$$

For $\mathbf{u}_k \in \mathcal{L}_k = \mathbf{K}\mathcal{K}_k$, there exists $\mathbf{v}_k \in \mathcal{K}_k$ such that $\mathbf{u}_k = \mathbf{K}\mathbf{v}_k$, and we deduce that

$$\mathbf{K}^T \mathbf{u}_k = \mathbf{K}^T \mathbf{K} \mathbf{v}_k \in \mathcal{K}_{k+1}. \tag{5.26}$$

Since by induction hypothesis $\mathbf{s}_{k-1} \in \mathcal{K}_k \subset \mathcal{K}_{k+1}$, (5.25) gives $\mathbf{s}_k \in \mathcal{K}_{k+1}$. This result together with the orthogonality relation (5.20) yields the (orthogonal) sum representation $\mathcal{K}_{k+1} = \mathcal{K}_k \oplus \operatorname{span}\{\mathbf{s}_k\}$, and the proof is finished. As $\dim(\mathcal{K}_k) = k$, $\dim(\mathcal{N}(\mathbf{K})^\perp) = r$, and $\mathcal{K}_k \subseteq \mathcal{N}(\mathbf{K})^\perp$, we find that for $k = r$, $\mathcal{K}_r = \mathcal{N}(\mathbf{K})^\perp$ and, in particular, that the CGNR iterate $\mathbf{x}_r^\delta = \arg\min_{\mathbf{x} \in \mathcal{N}(\mathbf{K})^\perp} \left\|\mathbf{y}^\delta - \mathbf{K}\mathbf{x}\right\|^2$ is the least squares minimal norm

solution of the equation $\mathbf{Kx} = \mathbf{y}^\delta$. Since $\mathbf{x}_r^\delta$ solves the normal equation $\mathbf{K}^T\mathbf{Kx} = \mathbf{K}^T\mathbf{y}^\delta$, we obtain

$$\mathbf{s}_r = \mathbf{K}^T\mathbf{r}_r^\delta = \mathbf{K}^T\left(\mathbf{y}^\delta - \mathbf{Kx}_r^\delta\right) = \mathbf{0}.$$

Thus, by the CGNR method we construct a sequence of iterates which approaches the least squares minimal norm solution, and we have to stop at some iteration step $k < r$ in order to obtain a reliable solution. The set of orthogonal vectors $\{\mathbf{u}_k\}_{k\geq 1}$ is generated by applying the Gram–Schmidt orthogonalization procedure to the set of vectors $\{\mathbf{Ks}_k\}_{k\geq 0}$, that is,

$$\begin{aligned}
&\mathbf{u}_1 = \mathbf{Ks}_0,\\
&\mathbf{u}_k = \mathbf{Ks}_{k-1} - \sum_{i=1}^{k-1} \frac{\mathbf{u}_i^T\mathbf{Ks}_{k-1}}{\|\mathbf{u}_i\|^2}\mathbf{u}_i,\ \ \mathbf{s}_{k-1} \neq \mathbf{0},\ \ k \geq 2.
\end{aligned} \tag{5.27}$$

The special form of the finite-dimensional subspaces $\mathcal{K}_k$ and $\mathcal{L}_k$ allows us to derive a recurrence relation for the orthogonal vectors $\mathbf{u}_k$. Since, for $k > 2$ and $i = 1, \ldots, k-2$, we have $\mathbf{s}_{k-1} \perp \mathcal{K}_{i+1} \subseteq \mathcal{K}_{k-1}$ and $\mathbf{K}^T\mathbf{u}_i \in \mathcal{K}_{i+1}$ (cf. (5.26)), we infer that

$$\mathbf{u}_i^T\mathbf{Ks}_{k-1} = \left(\mathbf{K}^T\mathbf{u}_i\right)^T\mathbf{s}_{k-1} = 0.$$

The basis vector $\mathbf{u}_k$ defined by (5.27) can then be expressed as

$$\mathbf{u}_k = \mathbf{Ks}_{k-1} + \beta_{k-1}\mathbf{u}_{k-1},\ \ k \geq 1, \tag{5.28}$$

with

$$\beta_{k-1} = -\frac{\mathbf{u}_{k-1}^T\mathbf{Ks}_{k-1}}{\|\mathbf{u}_{k-1}\|^2} \tag{5.29}$$

and the convention $\beta_0 = 0$. The first orthogonal vectors $\mathbf{s}_k$ and $\mathbf{u}_k$ are illustrated in Figure 5.1.

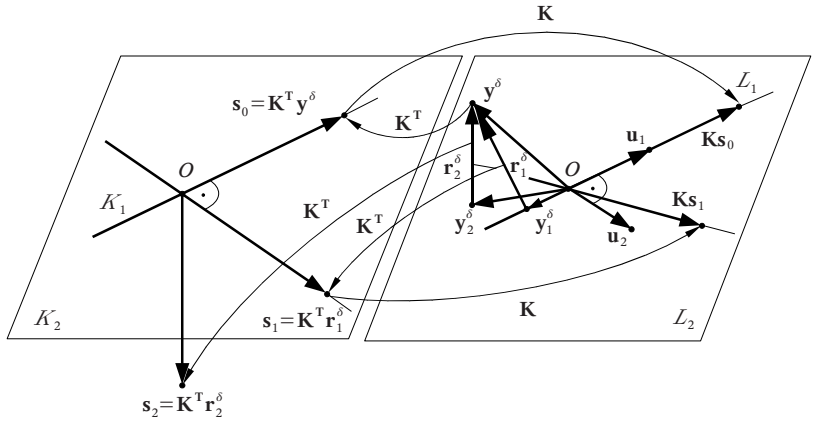

Fig. 5.1. The first orthogonal vectors $\mathbf{s}_k$ and $\mathbf{u}_k$. The construction is as follows: (1) $\mathbf{r}_0^\delta = \mathbf{y}^\delta \rightarrow \mathbf{s}_0 = \mathbf{K}^T\mathbf{r}_0^\delta$, $\mathcal{K}_1 = \text{span}\{\mathbf{s}_0\} \rightarrow \mathcal{L}_1 = \mathbf{K}\mathcal{K}_1$; (2) $\mathbf{r}_1^\delta = \mathbf{y}^\delta - P_{\mathcal{L}_1}\mathbf{y}^\delta \rightarrow \mathbf{s}_1 = \mathbf{K}^T\mathbf{r}_1^\delta$, $\mathcal{K}_2 = \text{span}\{\mathbf{s}_0, \mathbf{s}_1\} \rightarrow \mathcal{L}_2 = \mathbf{K}\mathcal{K}_2$; (3) $\mathbf{r}_2^\delta = \mathbf{y}^\delta - P_{\mathcal{L}_2}\mathbf{y}^\delta \rightarrow \mathbf{s}_2 = \mathbf{K}^T\mathbf{r}_2^\delta$, and so on.

The preimages $\mathbf{v}_k \in \mathcal{K}_k$ of the orthogonal vectors $\mathbf{u}_k \in \mathcal{L}_k$, already defined by

$$\mathbf{u}_k = \mathbf{K}\mathbf{v}_k, \tag{5.30}$$

satisfy the recurrence relation (cf. (5.11), (5.28) and (5.30))

$$\mathbf{v}_k = \mathbf{s}_{k-1} + \beta_{k-1}\mathbf{v}_{k-1}, \quad k \geq 1. \tag{5.31}$$

Besides that, the residual vector $\mathbf{r}_k^\delta$ can be computed recursively by using (5.24), while a recurrence relation for the iterates $\mathbf{x}_k^\delta$ can be obtained from (5.22) in conjunction with (5.11), (5.16) and (5.30); the result is

$$\mathbf{x}_k^\delta = \mathbf{x}_{k-1}^\delta + \alpha_k \mathbf{v}_k, \quad k \geq 1. \tag{5.32}$$

The coefficients α_k and β_k, defined by (5.23) and (5.29), respectively, can be computed efficiently as follows:

(1) For $k \geq 2$, we have $\mathbf{u}_k \perp \mathcal{L}_{k-1}$ and $\mathbf{K}\mathbf{x}_{k-1}^\delta \in \mathcal{L}_{k-1}$, and we find that $\mathbf{u}_k^T \mathbf{K}\mathbf{x}_{k-1}^\delta = 0$ for $k \geq 1$. Then, by (5.16), (5.18), (5.30), (5.31), and the orthogonality relation $\mathbf{s}_{k-1} \perp \mathbf{v}_{k-1} \in \mathcal{K}_{k-1}$, (5.23) yields

$$\begin{aligned} \alpha_k \|\mathbf{u}_k\|^2 &= \mathbf{u}_k^T \left(\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k-1}^\delta\right) \\ &= \left(\mathbf{K}\mathbf{v}_k\right)^T \mathbf{r}_{k-1}^\delta \\ &= \mathbf{v}_k^T \mathbf{s}_{k-1} \\ &= \|\mathbf{s}_{k-1}\|^2 + \beta_{k-1}\mathbf{v}_{k-1}^T \mathbf{s}_{k-1} \\ &= \|\mathbf{s}_{k-1}\|^2, \end{aligned}$$

and so,

$$\alpha_k = \frac{\|\mathbf{s}_{k-1}\|^2}{\|\mathbf{u}_k\|^2}, \quad k \geq 1.$$

(2) By (5.24) and the orthogonality relation $\mathbf{s}_k \perp \mathbf{s}_{k-1}$, we have

$$-\alpha_k \mathbf{u}_k^T \mathbf{K}\mathbf{s}_k = \left(\mathbf{r}_k^\delta - \mathbf{r}_{k-1}^\delta\right)^T \mathbf{K}\mathbf{s}_k = \left(\mathbf{s}_k - \mathbf{s}_{k-1}\right)^T \mathbf{s}_k = \|\mathbf{s}_k\|^2,$$

and (5.29) gives

$$\beta_k = \frac{\|\mathbf{s}_k\|^2}{\alpha_k \|\mathbf{u}_k\|^2} = \frac{\|\mathbf{s}_k\|^2}{\|\mathbf{s}_{k-1}\|^2}, \quad k \geq 1.$$

Collecting all results, we summarize the kth iteration step of the CGNR method as follows: given $\mathbf{x}_{k-1}^\delta$, $\mathbf{r}_{k-1}^\delta$, $\mathbf{s}_{k-1} \neq \mathbf{0}$ and $\mathbf{v}_k$, compute

$$\begin{aligned} \mathbf{u}_k &= \mathbf{K}\mathbf{v}_k, \\ \alpha_k &= \|\mathbf{s}_{k-1}\|^2 / \|\mathbf{u}_k\|^2, \\ \mathbf{x}_k^\delta &= \mathbf{x}_{k-1}^\delta + \alpha_k \mathbf{v}_k, \\ \mathbf{r}_k^\delta &= \mathbf{r}_{k-1}^\delta - \alpha_k \mathbf{u}_k, \\ \mathbf{s}_k &= \mathbf{K}^T \mathbf{r}_k^\delta, \\ \beta_k &= \|\mathbf{s}_k\|^2 / \|\mathbf{s}_{k-1}\|^2, \\ \mathbf{v}_{k+1} &= \mathbf{s}_k + \beta_k \mathbf{v}_k. \end{aligned}$$

Even the best implementation of the CGNR method suffers from some loss of accuracy due to the implicit use of the cross-product matrix $\mathbf{K}^T\mathbf{K}$. An alternative iterative method which avoids $\mathbf{K}^T\mathbf{K}$ completely is the LSQR algorithm of Paige and Saunders (1982). This method is based on the Lanczos bidiagonalization procedure of Golub and Kahan (1965) and is analytically equivalent to the CGNR method.

The Lanczos bidiagonalization algorithm is initialized with

$$\beta_1\bar{\mathbf{u}}_1 = \mathbf{y}^\delta, \;\; \alpha_1\bar{\mathbf{v}}_1 = \mathbf{K}^T\bar{\mathbf{u}}_1, \tag{5.33}$$

and the iteration step $k \geq 1$ has the form

$$\beta_{k+1}\bar{\mathbf{u}}_{k+1} = \mathbf{K}\bar{\mathbf{v}}_k - \alpha_k\bar{\mathbf{u}}_k, \tag{5.34}$$

$$\alpha_{k+1}\bar{\mathbf{v}}_{k+1} = \mathbf{K}^T\bar{\mathbf{u}}_{k+1} - \beta_{k+1}\bar{\mathbf{v}}_k. \tag{5.35}$$

The scalars $\alpha_k > 0$ and $\beta_k > 0$ are chosen such that

$$\|\bar{\mathbf{u}}_k\| = \|\bar{\mathbf{v}}_k\| = 1;$$

for example, the representation $\alpha_1\bar{\mathbf{v}}_1 = \mathbf{K}^T\bar{\mathbf{u}}_1$ assumes the calculations

$$\mathbf{v}_1 = \mathbf{K}^T\bar{\mathbf{u}}_1, \;\; \alpha_1 = \|\mathbf{v}_1\|, \;\; \bar{\mathbf{v}}_1 = (1/\alpha_1)\,\mathbf{v}_1.$$

Defining the dense matrices

$$\bar{\mathbf{U}}_{k+1} = [\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_{k+1}] \in \mathbb{R}^{m\times(k+1)}, \;\; \bar{\mathbf{V}}_k = [\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_k] \in \mathbb{R}^{n\times k},$$

and the bidiagonal matrix

$$\mathbf{B}_k = \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ \beta_2 & \alpha_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_k \\ 0 & 0 & \ldots & \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

we rewrite the recurrence relations (5.33)–(5.35) as

$$\beta_1\bar{\mathbf{U}}_{k+1}\mathbf{e}_1^{(k+1)} = \mathbf{y}^\delta, \tag{5.36}$$

$$\mathbf{K}\bar{\mathbf{V}}_k = \bar{\mathbf{U}}_{k+1}\mathbf{B}_k, \tag{5.37}$$

$$\mathbf{K}^T\bar{\mathbf{U}}_{k+1} = \bar{\mathbf{V}}_k\mathbf{B}_k^T + \alpha_{k+1}\bar{\mathbf{v}}_{k+1}\mathbf{e}_{k+1}^{(k+1)T}, \tag{5.38}$$

where $\mathbf{e}_j^{(k+1)}$ is the jth canonical vector in $\mathbb{R}^{k+1}$,

$$\left[\mathbf{e}_j^{(k+1)}\right]_i = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

The columns $\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_{k+1}$ of $\bar{\mathbf{U}}_{k+1}$ and $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_k$ of $\bar{\mathbf{V}}_k$ are called the left and the right Lanczos vectors, respectively. In exact arithmetics, $\bar{\mathbf{U}}_{k+1}$ and $\bar{\mathbf{V}}_k$ are orthogonal matrices, and we have

$$\bar{\mathbf{U}}_{k+1}^T\bar{\mathbf{U}}_{k+1} = \mathbf{I}_{k+1}, \;\; \bar{\mathbf{V}}_k^T\bar{\mathbf{V}}_k = \mathbf{I}_k.$$

As a result, $\mathbf{B}_k^T\mathbf{B}_k$ can be expressed as

$$\mathbf{B}_k^T\mathbf{B}_k = \bar{\mathbf{V}}_k^T\left(\mathbf{K}^T\mathbf{K}\right)\bar{\mathbf{V}}_k,$$

and we infer that

$$\left(\mathbf{B}_k^T\mathbf{B}_k\right)^j = \bar{\mathbf{V}}_k^T\left(\mathbf{K}^T\mathbf{K}\right)^j\bar{\mathbf{V}}_k,\ \ j\geq 1.$$

Using the relations

$$\mathbf{K}^T\mathbf{y}^\delta = \alpha\bar{\mathbf{v}}_1 = \alpha\bar{\mathbf{V}}_k\mathbf{e}_1^{(k)},\ \ \alpha = \left\|\mathbf{K}^T\mathbf{y}^\delta\right\|,$$

and

$$\left(\mathbf{K}^T\mathbf{K}\right)^j\mathbf{K}^T\mathbf{y}^\delta = \alpha\left(\mathbf{K}^T\mathbf{K}\right)^j\bar{\mathbf{v}}_1 = \alpha\left(\mathbf{K}^T\mathbf{K}\right)^j\bar{\mathbf{V}}_k\mathbf{e}_1^{(k)} = \alpha\bar{\mathbf{V}}_k\left(\mathbf{B}_k^T\mathbf{B}_k\right)^j\mathbf{e}_1^{(k)},$$

and setting

$$\mathtt{K}_k = \left[\mathbf{K}^T\mathbf{y}^\delta, \left(\mathbf{K}^T\mathbf{K}\right)\mathbf{K}^T\mathbf{y}^\delta, \ldots, \left(\mathbf{K}^T\mathbf{K}\right)^{k-1}\mathbf{K}^T\mathbf{y}^\delta\right]\in\mathbb{R}^{n\times k}$$

and

$$\mathbf{E}_k = \alpha\left[\mathbf{e}_1^{(k)}, \left(\mathbf{B}_k^T\mathbf{B}_k\right)\mathbf{e}_1^{(k)}, \ldots, \left(\mathbf{B}_k^T\mathbf{B}_k\right)^{k-1}\mathbf{e}_1^{(k)}\right]\in\mathbb{R}^{k\times k}$$

we find that

$$\mathtt{K}_k = \bar{\mathbf{V}}_k\mathbf{E}_k. \tag{5.39}$$

Thus, (5.39) resembles the QR factorization of the matrix $\mathtt{K}_k$, and as $\mathcal{R}\left(\mathtt{K}_k\right) = \mathcal{K}_k$, we deduce that $\{\bar{\mathbf{v}}_i\}_{i=\overline{1,k}}$ is an orthonormal basis of $\mathcal{K}_k$. Therefore, the LSQR method can be regarded as a method for constructing an orthonormal basis for the kth Krylov subspace $\mathcal{K}_k$. To solve the least squares problem

$$\min_{\mathbf{x}_k\in\text{span}\{\bar{\mathbf{v}}_i\}_{i=\overline{1,k}}}\left\|\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_k\right\|^2,$$

we proceed as follows. First, we set

$$\mathbf{x}_k = \bar{\mathbf{V}}_k\mathbf{z}_k,$$

for some $\mathbf{z}_k\in\mathbb{R}^k$. Then, we express the 'residual'

$$\mathbf{r}_k = \mathbf{y}^\delta - \mathbf{K}\mathbf{x}_k,$$

as (cf. (5.36) and (5.37))

$$\mathbf{r}_k = \bar{\mathbf{U}}_{k+1}\mathbf{t}_{k+1},$$

with

$$\mathbf{t}_{k+1} = \beta_1\mathbf{e}_1^{(k+1)} - \mathbf{B}_k\mathbf{z}_k.$$

As we want $\|\mathbf{r}_k\|^2$ to be small, and since $\bar{\mathbf{U}}_{k+1}$ is theoretically orthogonal, we minimize $\|\mathbf{t}_{k+1}\|^2$. Hence, in the kth iteration step of the LSQR method we solve the least squares problem

$$\min_{\mathbf{z}_k\in\mathbb{R}^k}\left\|\beta_1\mathbf{e}_1^{(k+1)} - \mathbf{B}_k\mathbf{z}_k\right\|^2. \tag{5.40}$$

If $\mathbf{z}_k^\delta$ is the least squares solution of (5.40), then the vector

$$\mathbf{x}_k^\delta = \bar{\mathbf{V}}_k \mathbf{z}_k^\delta = \beta_1 \bar{\mathbf{V}}_k \mathbf{B}_k^\dagger \mathbf{e}_1^{(k+1)},$$

which belongs to the kth Krylov subspace $\mathcal{K}_k = \text{span}\left\{\bar{\mathbf{v}}_i\right\}_{i=\overline{1,k}}$, is the iterate of the LSQR method. Computationally, the least squares problem (5.40) is solved by means of a QR factorization of $\mathbf{B}_k$, which is updated efficiently at each iteration step. The QR factorization then yields a simple recurrence relation for $\mathbf{x}_k^\delta$ in terms of $\mathbf{x}_{k-1}^\delta$, and neither $\bar{\mathbf{U}}_{k+1}$ nor $\bar{\mathbf{V}}_k$ need to be stored.

For discrete problems that do not require regularization, LSQR is likely to obtain more accurate results in fewer iteration steps as compared to CGNR (Paige and Saunders, 1982). However, for discrete ill-posed problems, where the iteration is stopped before convergence, both iterative methods yield results with comparable accuracies (Hansen, 1998).

In practice, the convergence of CGNR and LSQR is delayed due to the influence of finite precision arithmetic. Specifically, $\mathbf{x}_k^\delta$ stays almost unchanged for a few steps, then changes to a new vector and stays unchanged again for some steps, and so on. To prevent this delay and to simulate exact arithmetic, it is possible to incorporate some reorthogonalization techniques as for instance, the modified Gram–Schmidt algorithm or the Householder transformation. In LSQR we can orthogonalize the Lanczos vectors $\bar{\mathbf{u}}_i$ and $\bar{\mathbf{v}}_i$, while in CGNR we can orthogonalize the residual vectors $\mathbf{s}_i = \mathbf{K}^T \mathbf{r}_i^\delta$ (Hansen, 1998). The orthogonalization methods are illustrated in Algorithm 1.

For a deeper insight into the regularizing properties of the LSQR method, we consider the representation of the residual polynomial as given in Appendix F,

$$r_k(\lambda) = \prod_{j=1}^{k} \frac{\lambda_{k,j} - \lambda}{\lambda_{k,j}},$$

where

$$0 < \lambda_{k,k} < \lambda_{k,k-1} < \ldots < \lambda_{k,1},$$

are the eigenvalues of the matrix $\mathbf{B}_k^T \mathbf{B}_k$. The eigenvalues $\lambda_{k,j}$ are called Ritz values and for this reason, r_k is also known as the Ritz polynomial. The spectral filtering of the LSQR method is controlled by the convergence of the Ritz values to the eigenvalues of the matrix $\mathbf{K}^T \mathbf{K}$ (Hansen, 1998). This, in turn, is related to the number k of iteration steps. If, after k steps, a large eigenvalue σ_i^2 has been captured by the corresponding Ritz value $\lambda_{k,i}$, i.e., $\sigma_i^2 \approx \lambda_{k,i}$, then the corresponding filter factor is $f_k\left(\sigma_i^2\right) = 1 - r_k\left(\sigma_i^2\right) \approx 1$ (Appendix F). On the other hand, for an eigenvalue σ_i^2 much smaller than the smallest Ritz value, i.e., $\sigma_i^2 \ll \lambda_{k,k}$, the estimate

$$r_k\left(\sigma_i^2\right) = \prod_{j=1}^{k} \left(1 - \frac{\sigma_i^2}{\lambda_{k,j}}\right) \approx 1 - \sigma_i^2 \sum_{j=1}^{k} \frac{1}{\lambda_{k,j}},$$

yields

$$f_k\left(\sigma_i^2\right) \approx \sigma_i^2 \sum_{j=1}^{k} \frac{1}{\lambda_{k,j}},$$

Algorithm 1. Orthogonalization algorithms. (1) Modified Gram–Schmidt orthogonalization routine (MGSOrth): at the iteration step k, the new vector $\mathbf{p}$ is added to the set of orthonormal vectors stored in the columns of $\mathbf{P}$. (2) Householder orthogonalization routine (HOrth): at the iteration step k, the candidate vector $\mathbf{p}$ is transformed into a normalized vector $\bar{\mathbf{p}}$ orthogonal to the previous vectors; the vectors $\mathbf{v}_k$ and the scalars β_k, defining the reflection matrix $\mathbf{P}_k = \mathbf{I}_n - \beta_k \mathbf{v}_k \mathbf{v}_k^T$, are stored in the columns of the matrix $\mathbf{P}$ and in the array $\boldsymbol{\pi}$, respectively.

subroutine MGSOrth $(k, n, \mathbf{P};\ \mathbf{p})$
for $i = 1, k-1$ **do**
  $a \leftarrow \sum_{j=1}^{n} [\mathbf{p}]_j [\mathbf{P}]_{ji}$; {compute $\mathbf{p}^T [\mathbf{P}]_{\cdot i}$}
  for $j = 1, n$ **do** $[\mathbf{p}]_j \leftarrow [\mathbf{p}]_j - a [\mathbf{P}]_{ji}$; **end for**
end for

subroutine HOrth $(k, n, \boldsymbol{\pi}, \mathbf{P}, \mathbf{p};\ \bar{\mathbf{p}}, p_{\mathtt{nrm}}^{\mathtt{sgn}})$
{transformation $\mathbf{p} \leftarrow \mathbf{P}_{k-1}\mathbf{P}_{k-2}...\mathbf{P}_1\mathbf{p}$}
for $i = 1, k-1$ **do**
  $a \leftarrow \sum_{j=i}^{n} [\mathbf{p}]_j [\mathbf{P}]_{ji}$; {compute $[\mathbf{p}]_{i:n}^T [\mathbf{P}]_{i:n,i}$}
  for $j = i, n$ **do** $[\mathbf{p}]_j \leftarrow [\mathbf{p}]_j - a [\boldsymbol{\pi}]_i [\mathbf{P}]_{ji}$; **end for**
end for
{Householder reflection matrix $\mathbf{P}_k$}
$p \leftarrow \sqrt{\sum_{j=k}^{n} [\mathbf{p}]_j^2}$; $[\boldsymbol{\pi}]_k \leftarrow 1/\left(p^2 + |[\mathbf{p}]_k|\, p\right)$;
$[\mathbf{P}]_{kk} \leftarrow [\mathbf{p}]_k + \mathrm{sgn}\left([\mathbf{p}]_k\right) p$; **for** $j = k+1, n$ **do** $[\mathbf{P}]_{jk} \leftarrow [\mathbf{p}]_j$; **end for**
$p_{\mathtt{nrm}}^{\mathtt{sgn}} \leftarrow -\mathrm{sgn}\left([\mathbf{p}]_k\right) p$;
{transformation $\bar{\mathbf{p}} \leftarrow \mathbf{P}_1\mathbf{P}_2...\mathbf{P}_k\mathbf{e}_k$, where $\bar{\mathbf{p}}$ is normalized}
$\bar{\mathbf{p}} \leftarrow \mathbf{0}$, $[\bar{\mathbf{p}}]_k \leftarrow 1$;
for $i = k, 1, -1$ **do**
  $a \leftarrow \sum_{j=i}^{n} [\bar{\mathbf{p}}]_j [\mathbf{P}]_{ji}$; { $[\bar{\mathbf{p}}]_{i:n}^T [\mathbf{P}]_{i:n,i}$ }
  for $j = i, n$ **do** $[\bar{\mathbf{p}}]_j \leftarrow [\bar{\mathbf{p}}]_j - a [\boldsymbol{\pi}]_i [\mathbf{P}]_{ji}$; **end for**
end for

and we see that these filter factors decay like σ_i^2. Thus, if the Ritz values approximate the eigenvalues in natural order, starting from the largest, then the iteration index plays the role of the regularization parameter, and the filter factors behave like the Tikhonov filter factors.

5.4 Stopping rules and preconditioning

Stopping the iteration prior to the inclusion of amplified noise components in the solution is an important aspect of iterative regularization methods. Also relevant is the preconditioning of the system of equations in order to improve the convergence rate. These topics are discussed below.

5.4.1 Stopping rules

The most widespread stopping rule for iterative regularization methods is the discrepancy principle. According to the discrepancy principle, the algorithm is terminated with $k^\star$ when

$$\left\|\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_{k^\star}^\delta\right\|^2 \leq \tau\Delta^2 < \left\|\mathbf{y}^\delta - \mathbf{K}\mathbf{x}_k^\delta\right\|^2, \;\; 0 \leq k < k^\star. \tag{5.41}$$

In a semi-stochastic setting and for white noise with variance σ^2, the expected value of the noise $\mathcal{E}\{\|\boldsymbol{\delta}\|^2\} = m\sigma^2$ is used instead of the noise level Δ^2.

Error-free parameter choice methods can also be formulated as stopping rules. In this case we have to store each iterate together with the corresponding objective function, e.g., the generalized cross-validation function, and to perform a sufficient number of iteration steps in order to detect the minimum of the objective function. For iterative regularization methods, the use of the generalized cross-validation and the maximum likelihood estimation requires the knowledge of the influence matrix, which, in turn, requires the knowledge of the generalized inverse. This is a difficult task because neither a canonical decomposition of $\mathbf{K}$ nor the filter factors f_k are available (recall that iterative methods are preferred when a factorization of the matrix is infeasible).

More promising for iterative regularization methods is the use of the L-curve criterion. For the CGNR method, the monotonic behavior of both the solution norm $\left\|\mathbf{x}_k^\delta\right\|$ and the residual norm $\left\|\mathbf{r}_k^\delta\right\|$ recommends this approach. In the framework of Tikhonov regularization, the components of the L-curve are defined by some analytical formulas and the calculation of the curvature is straightforward. In the case of iterative methods, we are limited to knowing only a finite number of points on the L-curve (corresponding to different values of the iteration index). Unfortunately, these points are clustered giving fine-grained details that are not relevant for the determination of the corner. To eliminate this inconvenience, Hansen (1998) defined a differentiable smooth curve associated with the discrete points in such a way that fine-grained details are eliminated while the overall shape of the L-curve is maintained. The approximating curve is determined by fitting a cubic spline curve to the discrete points of the L-curve. Since a cubic spline curve does not have the desired local smoothing property, the following algorithm is employed:

(1) perform a local smoothing of the L-curve, that is, for each interior point $k = q + 1, \ldots, P - q$, where P is the number of discrete points of the L-curve and q is the half-width of the local smoothing interval, fit a polynomial of degree p to the points $k - q, \ldots, k + q$, and store the corresponding kth 'smoothed' point situated on the fitting polynomial;
(2) construct a cubic spline curve by using the smoothed points as control points;
(3) compute the corner of the spline curve by maximizing its curvature;
(4) select the point on the orginal discrete curve that is closest to the spline curve's corner.

Another method which couples a geometrical approach to identify the corner of the L-curve with some heuristics rules has been proposed by Rodriguez and Theis (2005). The main steps of this approach can be summarized as follows:

(1) compute the vectors $\mathbf{a}_k = [x_{k+1} - x_k, y_{k+1} - y_k]^T$, $k = 1, \ldots, P-1$, where $x_k = \log(\|\mathbf{r}_k^\delta\|^2)$ and $y_k = \log(\|\mathbf{x}_k^\delta\|^2)$;
(2) eliminate the clusters by deleting all the 'short' vectors;
(3) normalize the remaining V vectors;
(4) select the corner of the L-curve as that point which minimizes the scalar triple product between two successive vectors, i.e., $k^\star = \arg\min_{k=\overline{1,V-1}} w_k$, where $w_k = (\mathbf{a}_k \times \mathbf{a}_{k+1}) \cdot \mathbf{e}_3$, and $\mathbf{e}_3$ is the Cartesian unit vector codirectional with the z-axis.

5.4.2 Preconditioning

In general, the aim of preconditioning is to improve the convergence rate of iterative methods for solving large systems of equations. When preconditioning from the right, the linear system of equations

$$\mathbf{K}\mathbf{x} = \mathbf{y}^\delta, \tag{5.42}$$

is replaced by

$$\mathbf{K}\mathbf{M}\bar{\mathbf{x}} = \mathbf{y}^\delta, \;\; \mathbf{M}\bar{\mathbf{x}} = \mathbf{x},$$

with $\mathbf{M}$ being a nonsingular matrix. If (5.42) is solved by using an iterative method for normal equations, $\mathbf{M}$ should be chosen such that the condition number of $\mathbf{M}^T\mathbf{K}^T\mathbf{K}\mathbf{M}$ is smaller than that of $\mathbf{K}^T\mathbf{K}$. This spectral property then yields faster convergence for the iterative method.

For discrete ill-posed problems, the preconditioner should not be regarded as a convergence accelerator, but rather as an enhancer of solution quality, since convergence is never achieved. In fact, there is no point in improving the condition of $\mathbf{K}$ because only a part of the singular values contributes to the regularized solution (Hansen, 1998).

By right preconditioning we control the solution with a different norm as in the case of Tikhonov regularization with a regularization matrix $\mathbf{L}$. Therefore, there is no practical restriction to use a regularization matrix $\mathbf{L}$ in connection with iterative methods (Hanke and Hansen, 1993; Hansen, 1998). Regularization matrices, when used as right preconditioners, affect the solution of an iterative method in a similar way as they affect the solution of Tikhonov regularization. The system of equations preconditioned from the right by the nonsingular regularization matrix $\mathbf{L}$ then takes the form

$$\mathbf{K}\mathbf{L}^{-1}\bar{\mathbf{x}} = \mathbf{y}^\delta, \;\; \mathbf{L}^{-1}\bar{\mathbf{x}} = \mathbf{x}. \tag{5.43}$$

To obtain more insight into right preconditioning by regularization matrices, we recall that in the framework of Tikhonov regularization, we transformed a general-form problem (with $\mathbf{L} \neq \mathbf{I}_n$) into a standard-form problem (with $\mathbf{L} = \mathbf{I}_n$) by using the transformation $\bar{\mathbf{K}} = \mathbf{K}\mathbf{L}^{-1}$ and the back-transformation $\mathbf{x} = \mathbf{L}^{-1}\bar{\mathbf{x}}$. In terms of the standard-form variables, equation (5.43) expressed as

$$\bar{\mathbf{K}}\bar{\mathbf{x}} = \mathbf{y}^\delta, \;\; \mathbf{L}^{-1}\bar{\mathbf{x}} = \mathbf{x},$$

Algorithm 2. ν-algorithm with preconditioning. The control parameters of the algorithm are the maximum number of iterations $N_{\mathtt{iter}}$, the noise level Δ, and the tolerance τ. The notation $\|\mathbf{A}\|_{\mathtt{F}}$ stands for the Frobenius norm of the matrix $\mathbf{A}$.

$\chi \leftarrow 1/\left\|\mathbf{K}\mathbf{L}^{-1}\right\|_{\mathtt{F}}$; {relaxation parameter}
$\mathbf{x}^{\delta} \leftarrow \mathbf{0}$; $\mathbf{r}^{\delta} \leftarrow \chi\left(\mathbf{y}^{\delta} - \mathbf{K}\mathbf{x}^{\delta}\right)$;
{step $k = 1$}
$\omega \leftarrow \frac{4\nu+2}{4\nu+1}$;
$\mathbf{q} \leftarrow \omega\mathbf{r}^{\delta}$; $\mathbf{x}^{\delta} \leftarrow \mathbf{x}^{\delta} + \chi\left(\mathbf{L}^T\mathbf{L}\right)^{-1}\mathbf{K}^T\mathbf{q}$; $\mathbf{r}^{\delta} \leftarrow \chi\left(\mathbf{y}^{\delta} - \mathbf{K}\mathbf{x}^{\delta}\right)$;
if $\left\|\mathbf{r}^{\delta}\right\|^2 \leq \tau\chi^2\Delta^2$ **stop**; {residual smaller than the prescribed tolerance}
{steps $k \geq 2$}
for $k = 2, N_{\mathtt{iter}}$ **do**
$\omega \leftarrow 4\frac{(2k+2\nu-1)(k+\nu-1)}{(k+2\nu-1)(2k+4\nu-1)}$;
$\mu \leftarrow 0.25\frac{(k-1)(2k-3)}{(k+\nu-1)(2k+2\nu-3)}\omega$;
$\mathbf{q} \leftarrow \mu\mathbf{q} + \omega\mathbf{r}^{\delta}$; $\mathbf{x}^{\delta} \leftarrow \mathbf{x}^{\delta} + \chi\left(\mathbf{L}^T\mathbf{L}\right)^{-1}\mathbf{K}^T\mathbf{q}$; $\mathbf{r}^{\delta} \leftarrow \chi\left(\mathbf{y}^{\delta} - \mathbf{K}\mathbf{x}^{\delta}\right)$;
if $\left\|\mathbf{r}^{\delta}\right\|^2 \leq \tau\chi^2\Delta^2$ **exit**; {residual smaller than the prescribed tolerance}
end for

reveals that solving the right preconditioned system of equations is equivalent to solving the standard-form problem without preconditioning. In practice, the multiplication with $\mathbf{L}^{-1}$ is built into the iterative schemes, and the back-transformation is avoided. The ν-method, as well as the CGNR and the LSQR methods with preconditioning and using the discrepancy principle as stopping rule are outlined in Algorithms 2–4.

Algorithm 3. CGNR algorithm with preconditioning and reorthogonalization. The control parameters of the algorithm are the maximum number of iterations $N_{\texttt{iter}}$, the noise level Δ, the tolerance τ, and the logical variables $TypeOrth$. The values of $TypeOrth$ are as follows: 0 if no reorthogonalization is applied, 1 for Householder orthogonalization, and 2 for the modified Gram–Schmidt orthogonalization.

$\mathbf{x}^\delta \leftarrow \mathbf{0}$;
$\mathbf{r}^\delta \leftarrow \mathbf{y}^\delta - \mathbf{K}\mathbf{x}^\delta$;
if $TypeOrth \neq 0$ $\mathbf{S} \leftarrow \mathbf{0}$;
$\mathbf{q} \leftarrow \mathbf{K}^T\mathbf{r}^\delta$;
$\mathbf{s} \leftarrow \mathbf{L}^{-T}\mathbf{q}$;
{initialization of arrays $\mathbf{S}$ and $\boldsymbol{\sigma}$}
if $TypeOrth = 1$ **then**
    $\boldsymbol{\sigma} \leftarrow \mathbf{0}$; $s \leftarrow \|\mathbf{s}\|$; $[\boldsymbol{\sigma}]_1 \leftarrow 1/\left(s^2 + |[\mathbf{s}]_1|\, s\right)$;
    $[\mathbf{S}]_{11} \leftarrow [\mathbf{s}]_1 + \text{sgn}\left([\mathbf{s}]_1\right) s$;
    for $i = 2, n$ **do** $[\mathbf{S}]_{i1} \leftarrow [\mathbf{s}]_i$; **end for**
    $s_{\texttt{nrm}} \leftarrow -\text{sgn}\left([\mathbf{s}]_1\right) s$;
{initialization of array $\mathbf{S}$}
else if $TypeOrth = 2$ **then**
    $s_{\texttt{nrm}} \leftarrow \|\mathbf{s}\|$;
    for $i = 1, n$ **do** $[\mathbf{S}]_{i1} \leftarrow [\mathbf{s}]_i / s_{\texttt{nrm}}$; **end for**
else
    $s_{\texttt{nrm}} \leftarrow \|\mathbf{s}\|$;
end if
$\mathbf{v} \leftarrow \mathbf{L}^{-1}\mathbf{s}$;
for $k = 2, N_{\texttt{iter}}$ **do**
    $\mathbf{u} \leftarrow \mathbf{K}\mathbf{v}$;
    $\alpha \leftarrow s_{\texttt{nrm}}^2 / \|\mathbf{u}\|^2$;
    $\mathbf{x}^\delta \leftarrow \mathbf{x}^\delta + \alpha\mathbf{v}$;
    $\mathbf{r}^\delta \leftarrow \mathbf{r}^\delta - \alpha\mathbf{u}$;
    if $\left\|\mathbf{r}^\delta\right\|^2 \leq \tau\Delta^2$ **exit**; {residual smaller than the prescribed tolerance}
    $\mathbf{q} \leftarrow \mathbf{K}^T\mathbf{r}^\delta$;
    $\mathbf{s} \leftarrow \mathbf{L}^{-T}\mathbf{q}$;
    if $TypeOrth = 1$ **then**
        call HOrth $(k, n, \boldsymbol{\sigma}, \mathbf{S}, \mathbf{s};\ \bar{\mathbf{s}}, s_{\texttt{nrm1}})$; $\mathbf{s} \leftarrow s_{\texttt{nrm1}}\bar{\mathbf{s}}$;
    else if $TypeOrth = 2$ **then**
        call MGSOrth $(k, n, \mathbf{S};\ \mathbf{s})$; $s_{\texttt{nrm1}} \leftarrow \|\mathbf{s}\|$;
        for $i = 1, n$ **do** $[\mathbf{S}]_{ik} \leftarrow [\mathbf{s}]_i / s_{\texttt{nrm1}}$; **end for**
    else
        $s_{\texttt{nrm1}} \leftarrow \|\mathbf{s}\|$;
    end if
    $\beta \leftarrow s_{\texttt{nrm1}}^2 / s_{\texttt{nrm}}^2$;
    $s_{\texttt{nrm}} \leftarrow s_{\texttt{nrm1}}$;
    $\mathbf{v} \leftarrow \mathbf{L}^{-1}\mathbf{s} + \beta\mathbf{v}$;
end for

Algorithm 4. LSQR algorithm with preconditioning and reorthogonalization.

$\mathbf{x}^\delta \leftarrow \mathbf{0}$; **if** $TypeOrth \neq 0$ **then** $\mathbf{P} \leftarrow \mathbf{0}$; $\mathbf{Q} \leftarrow \mathbf{0}$; **end if**
if $TypeOrth = 1$ **then** {initialization of arrays $\mathbf{P}$ and $\boldsymbol{\pi}$}
  $\boldsymbol{\pi} \leftarrow \mathbf{0}$; $p \leftarrow \left\|\mathbf{y}^\delta\right\|$; $[\boldsymbol{\pi}]_1 \leftarrow 1/\left(p^2 + \left|\left[\mathbf{y}^\delta\right]_1\right| p\right)$;
  $[\mathbf{P}]_{11} \leftarrow \left[\mathbf{y}^\delta\right]_1 + \operatorname{sgn}\left(\left[\mathbf{y}^\delta\right]_1\right) p$; **for** $i = 2, m$ **do** $[\mathbf{P}]_{i1} \leftarrow \left[\mathbf{y}^\delta\right]_i$; **end for**
  $\beta \leftarrow -\operatorname{sgn}\left(\left[\mathbf{y}^\delta\right]_1\right) p$; $\bar{\mathbf{u}} \leftarrow (1/\beta)\, \mathbf{y}^\delta$;
else if $TypeOrth = 2$ **then** {initialization of array $\mathbf{P}$}
  $\beta \leftarrow \left\|\mathbf{y}^\delta\right\|$; $\bar{\mathbf{u}} \leftarrow (1/\beta)\, \mathbf{y}^\delta$; **for** $i = 1, m$ **do** $[\mathbf{P}]_{i1} \leftarrow [\bar{\mathbf{u}}]_i$; **end for**
else
  $\beta \leftarrow \left\|\mathbf{y}^\delta\right\|$; $\bar{\mathbf{u}} \leftarrow (1/\beta)\, \mathbf{y}^\delta$;
end if
$\mathbf{q} \leftarrow \mathbf{L}^{-T}\mathbf{K}^T\bar{\mathbf{u}}$;
if $TypeOrth = 1$ **then** {initialization of arrays $\mathbf{Q}$ and $\boldsymbol{\nu}$}
  $\boldsymbol{\nu} \leftarrow \mathbf{0}$; $q \leftarrow \|\mathbf{q}\|$; $[\boldsymbol{\nu}]_1 \leftarrow 1/\left(q^2 + |[\mathbf{q}]_1|\, q\right)$;
  $[\mathbf{Q}]_{11} \leftarrow [\mathbf{q}]_1 + \operatorname{sgn}\left([\mathbf{q}]_1\right) q$; **for** $i = 2, n$ **do** $[\mathbf{Q}]_{i1} \leftarrow [\mathbf{q}]_i$; **end for**
  $\alpha \leftarrow -\operatorname{sgn}\left([\mathbf{q}]_1\right) q$; $\bar{\mathbf{v}} \leftarrow (1/\alpha)\, \mathbf{q}$;
else if $TypeOrth = 2$ **then** {initialization of array $\mathbf{Q}$}
  $\alpha \leftarrow \|\mathbf{q}\|$; $\bar{\mathbf{v}} \leftarrow (1/\alpha)\, \mathbf{q}$; **for** $i = 1, n$ **do** $[\mathbf{Q}]_{i1} \leftarrow [\bar{\mathbf{v}}]_i$; **end for**
else
  $\alpha =\leftarrow \|\mathbf{q}\|$; $\bar{\mathbf{v}} \leftarrow (1/\alpha)\, \mathbf{q}$;
end if
$\mathbf{w} \leftarrow \mathbf{v}$; $\bar{\phi} \leftarrow \beta$; $\bar{\rho} \leftarrow \alpha$;
for $k = 2, N_{\mathtt{iter}}$ **do**
  $\mathbf{p} \leftarrow \mathbf{K}\mathbf{L}^{-1}\bar{\mathbf{v}} - \alpha\bar{\mathbf{u}}$;
  if $TypeOrth = 1$ **then**
    call HOrth $(k, m, \boldsymbol{\pi}, \mathbf{P}, \mathbf{p};\ \bar{\mathbf{u}}, \beta)$;
  else if $TypeOrth = 2$ **then**
    call MGSOrth $(k, m, \mathbf{P};\ \mathbf{p})$; $\beta \leftarrow \|\mathbf{p}\|$; $\bar{\mathbf{u}} \leftarrow (1/\beta)\, \mathbf{p}$;
  else
    $\beta \leftarrow \|\mathbf{p}\|$; $\bar{\mathbf{u}} \leftarrow (1/\beta)\, \mathbf{p}$;
  end if
  $\mathbf{q} \leftarrow \mathbf{L}^{-T}\mathbf{K}^T\bar{\mathbf{u}} - \beta\bar{\mathbf{v}}$;
  if $TypeOrth = 1$ **then**
    call HOrth $(k, n, \boldsymbol{\nu}, \mathbf{Q}, \mathbf{q};\ \bar{\mathbf{v}}, \alpha)$;
  else if $TypeOrth = 2$ **then**
    call MGSOrth $(k, n, \mathbf{Q};\ \mathbf{q})$; $\alpha \leftarrow \|\mathbf{q}\|$; $\bar{\mathbf{v}} \leftarrow (1/\alpha)\, \mathbf{q}$;
  else
    $\alpha \leftarrow \|\mathbf{q}\|$; $\bar{\mathbf{v}} \leftarrow (1/\alpha)\, \mathbf{q}$;
  end if
  if $TypeOrth = 2$ store $\bar{\mathbf{u}}$ in column k of $\mathbf{P}$ and $\bar{\mathbf{v}}$ in column k of $\mathbf{Q}$;
  $\rho \leftarrow \sqrt{\bar{\rho}^2 + \beta^2}$; $c \leftarrow \bar{\rho}/\rho$; $s \leftarrow \beta/\rho$; $\theta \leftarrow s\alpha$; $\bar{\rho} \leftarrow -c/\alpha$;
  $\phi \leftarrow c\bar{\phi}$; $\bar{\phi} \leftarrow s\bar{\phi}$; $\left\|\mathbf{r}^\delta\right\| \leftarrow \bar{\phi}$; $\mathbf{x}^\delta \leftarrow \mathbf{x}^\delta + (\phi/\rho)\, \mathbf{w}$; $\mathbf{w} \leftarrow \bar{\mathbf{v}} - (\theta/\rho)\, \mathbf{w}$;
  if $\left\|\mathbf{r}^\delta\right\|^2 \leq \tau\Delta^2$ **exit**; {residual smaller than the prescribed tolerance}
end for
$\mathbf{x}^\delta \leftarrow \mathbf{L}^{-1}\mathbf{x}^\delta$;

5.5 Numerical analysis

To analyze the performance of iterative regularization methods we consider the same retrieval scenario as in Chapter 3, but retrieve the O_3 profile together with the NO_2 profile in a spectral interval ranging from 520 to 580 nm. The atmosphere is discretized with a step of 1 km between 0 and 60 km, and a step of 5 km between 60 and 100 km. The number of unknowns of the inverse problem is $n = 100$. In our first simulation, we choose the discrepancy principle as stopping rule. As CGNR and LSQR yield identical results, only the CGNR results are reported here.

The solution errors for different values of the control parameter τ (cf. (5.41)) are illustrated in the left panel of Figure 5.2. The error curves possess a minimum for an optimal value of the control parameter: the smallest errors are $5.56 \cdot 10^{-2}$ for the ν-method, $5.20 \cdot 10^{-2}$ for CGNR without reorthogonalization and $5.02 \cdot 10^{-2}$ for CGNR with Householder orthogonalization. Note that the stepwise behavior of the error curves for the CGNR method is a consequence of the discrete nature of the stopping rule. The retrieved profiles are shown in the right panel of Figure 5.2, and a sensible superiority of CGNR with Householder orthogonalization can be observed in the lower part of the atmosphere.

Although the methods are of comparable accuracies, the convergence rates are completely different (Figure 5.3). To reduce the residual norm to the order of the noise level, 100 iteration steps are required by the ν-method, 50 by CGNR without reorthogonalization and 30 by CGNR with Householder orthogonalization.

The non-monotonic behavior of the residual curve in the case of the ν-method is apparent in the left panel of Figure 5.4, while the delay of CGNR without reorthogonalization

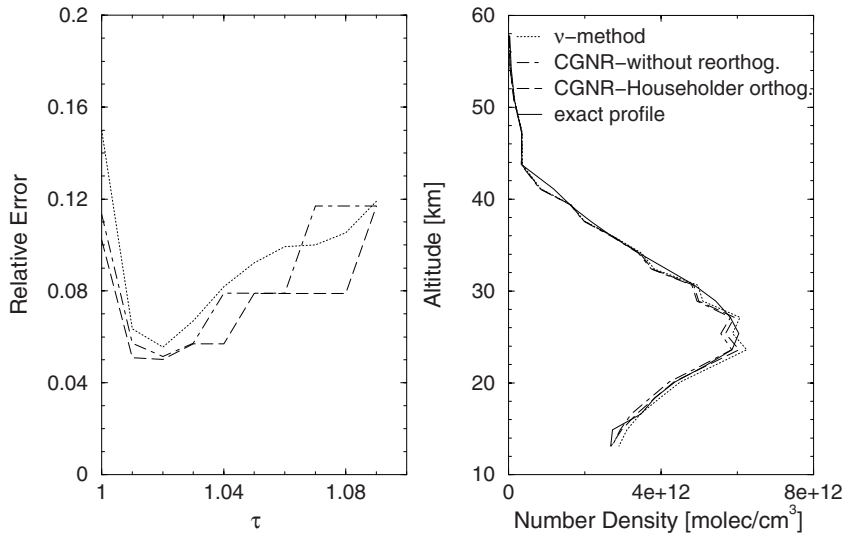

Fig. 5.2. Left: relative solution errors for different values of the control parameter τ. Right: retrieved profiles corresponding to the optimal values of τ. The results are computed with the ν-method ($\nu = 0.5$), CGNR without reorthogonalization, and CGNR with Householder orthogonalization.

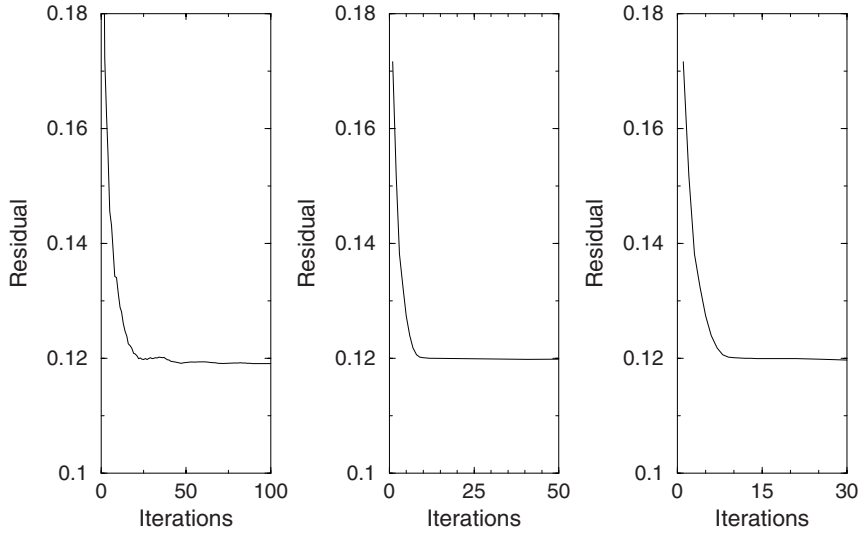

Fig. 5.3. Histories of the residual norm corresponding to the ν-method (left), CGNR without reorthogonalization (middle), and CGNR with Householder orthogonalization (right).

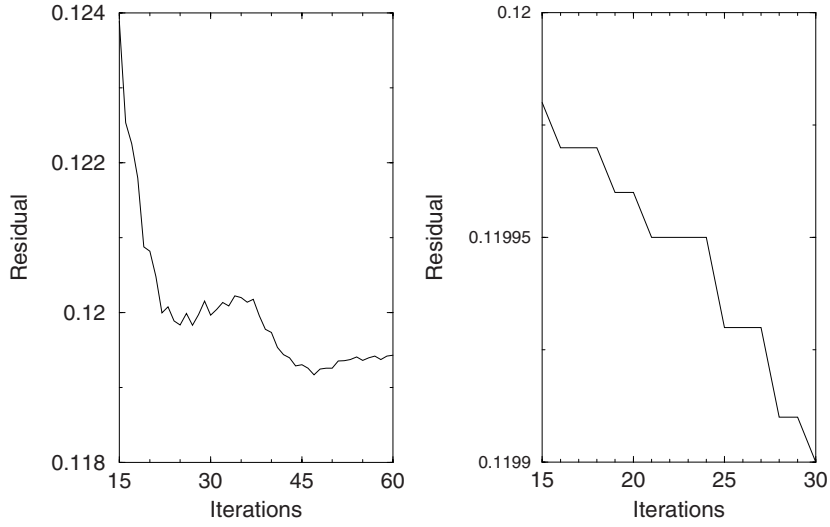

Fig. 5.4. Left: non-monotonic behavior of the residual curve corresponding to the ν-method. Right: delay of CGNR without reorthogonalization reflected in the residual curve.

(the iterate stays almost unchanged for a few steps) is evidenced in the right panel of Figure 5.4.

The discrete L-curve for the CGNR method illustrated in Figure 5.5 has a pronounced L-shape with a distinct corner. The inversion performance of CGNR with the L-curve method are slightly better than those of CGNR with the discrepancy principle; the retrieved profile in Figure 5.5 is characterized by a solution error of $4.52 \cdot 10^{-2}$.

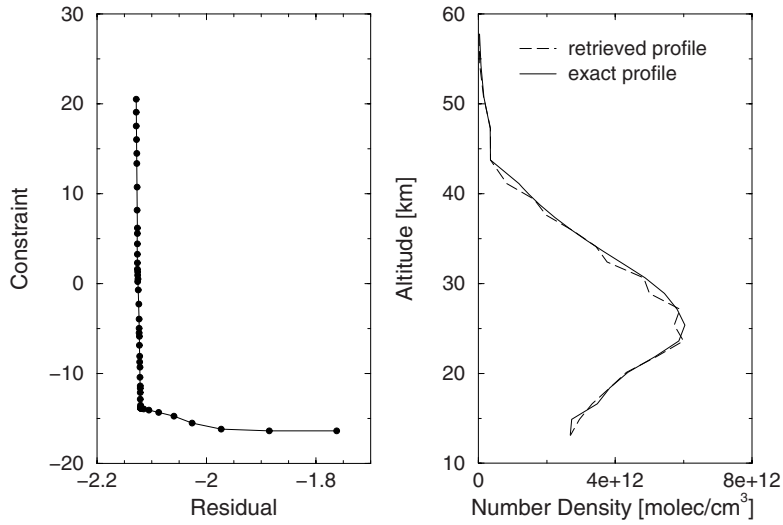

Fig. 5.5. Discrete L-curve for CGNR with Householder orthogonalization (left) and the corresponding retrieved profile (right).

5.6 Mathematical results and further reading

A deterministic analysis of the Landweber iteration and of semi-iterative methods equipped with the discrepancy principle as stopping rule is presented in the first part of Appendix E. For the source condition $\mathbf{x}^\dagger = \left(\mathbf{K}^T\mathbf{K}\right)^\mu \mathbf{z}$, with $\mu > 0$ and $\mathbf{z} \in \mathbb{R}^n$, the Landweber iteration is order-optimal for all $\mu > 0$, while the ν-method is order-optimal for $0 < \mu \leq \nu - 1/2$. Despite its optimal convergence rate, the Landweber iteration is rarely used in practice, since it usually requires far too many iteration steps until the stopping criterion (5.41) is met; the stopping index for the Landweber iteration is $k^\star = O(\Delta^{-2/(2\mu+1)})$, and the exponent $2/\left(2\mu+1\right)$ cannot be improved in general (Engl et al., 2000).

The convergence rate of the CGNR method using the discrepancy principle as stopping rule is derived in the second part of Appendix E. This method is order-optimal for $\mu > 0$, and so, no saturation effect occurs. In general, the number of iteration steps of the CGNR method is $k^\star = O(\Delta^{-1/(2\mu+1)})$, and in particular, we have

$$k^\star = O\left(\Delta^{-\frac{1}{(2\mu+1)(\beta+1)}}\right)$$

for the polynomial ill-posedness $\sigma_i = O(i^{-\beta})$ with $\beta > 0$, and

$$k^\star = O\left(\left|\log \Delta^{\frac{1}{2\mu+1}}\right|\right)$$

for the exponential ill-posedness $\sigma_i = O(q^i)$ with $q \in (0,1)$. In any case, the CGNR method requires significantly less iteration steps for the same order of accuracy than the Landweber iteration or the ν-method. A detailed analysis of conjugate gradient type methods for ill-posed problems can be found in Hanke (1995), while for a pertinent treatment of preconditioned iterative regularization methods we refer to Hanke et al. (1993).