Chapter 5
Systems of Inhomogeneous Linear Equations

Many problems in physics and especially computational physics involve systems of linear equations

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ \vdots \quad \vdots \quad \vdots & \\ a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{5.1}$$

or shortly in matrix form

$$A\mathbf{x} = \mathbf{b} \tag{5.2}$$

which arise e.g. from linearization of a general nonlinear problem like (Sect. 22.2)

$$0 = \begin{pmatrix} F_1(x_1 \ldots x_n) \\ \vdots \\ F_n(x_1 \ldots x_n) \end{pmatrix} = \begin{pmatrix} F_1(x_1^{(0)} \ldots x_n^{(0)}) \\ \vdots \\ F_n(x_1^{(0)} \ldots x_n^{(0)}) \end{pmatrix} + \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{pmatrix} \begin{pmatrix} x_1 - x_1^{(0)} \\ \vdots \\ x_n - x_n^{(0)} \end{pmatrix} + \ldots \tag{5.3}$$

or from discretization of differential equations like

$$\begin{aligned} 0 = \frac{\partial f}{\partial x} - g(x) &\to \begin{pmatrix} \vdots \\ \frac{f((j+1)\Delta x) - f(j\Delta x)}{\Delta x} - g(j\Delta x) \\ \vdots \end{pmatrix} \\ &= \begin{pmatrix} \ddots & & & & \\ & -\frac{1}{\Delta x} & \frac{1}{\Delta x} & & \\ & & -\frac{1}{\Delta x} & \frac{1}{\Delta x} & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ f_j \\ f_{j+1} \\ \vdots \end{pmatrix} - \begin{pmatrix} \vdots \\ g_j \\ g_{j+1} \\ \vdots \end{pmatrix}. \end{aligned} \tag{5.4}$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_5

If the matrix A is non singular and has full rank, (5.2) *can be formally solved by matrix inversion*

$$\mathbf{x} = A^{-1}\mathbf{b}. \tag{5.5}$$

If the matrix is singular or the number of equations smaller than the number of variables, a manifold of solutions exists which can be found efficiently by singular value decomposition (Sect. 11.2*). The general solution is given by a particular solution and the nullspace of A*

$$\mathbf{x} = \mathbf{x}_p + \mathbf{z} \; with \; A\mathbf{x}_p = \mathbf{b} \; and \; A\mathbf{z} = 0. \tag{5.6}$$

If the number of equations is larger than the number of variables there exists no unique solution. The "best possible solution" can be determined by minimizing the residual

$$|A\mathbf{x} - \mathbf{b}| = \min \tag{5.7}$$

which leads to a least squares problem (Sect. 11.1.1*).*

In the following we discuss several methods to solve non singular systems. If the dimension is not too large, direct methods like Gaussian elimination or QR decomposition are sufficient. Systems with a tridiagonal matrix are important for cubic spline interpolation and numerical second derivatives. They can be solved very efficiently with a specialized Gaussian elimination method. Practical applications often involve very large dimensions and require iterative methods. Stationary methods apply a simple iteration scheme repeatedly. The slow convergence of the methods by Jacobi and Gauss-Seidel can be improved with relaxation or overrelaxation. Non-stationary methods construct a sequence of improved approximations within a series of increasing subspaces of $\mathbb{R}^N$*. Modern Krylov-space methods minimize the residual* $\mathbf{r} = A\mathbf{x} - \mathbf{b}$ *within the sequence of Krylov-spaces* $K_n(A, \mathbf{r}^{(0)}) = \text{span}(\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, \ldots A^{n-1}\mathbf{r}^{(0)})$*. We discuss the conjugate gradients method (CG* [25]*) for symmetric positive definite matrices and the method of general minimal residuals (GMRES* [26]*) for non symmetric matrices. Other popular methods are the methods of bi-conjugate gradients (BiCG* [27] *BiCGSTAB* [28]*), conjugate residuals (CR* [29]*) , minimal residual (MINRES* [30]*), quasi-minimal residual (QMR* [31]*), the symmetric LQ-method (SYMMLQ* [32]*) and Lanzcos type product methods (LTPM* [33–35]*).*

5.1 Gaussian Elimination Method

A series of linear combinations of the equations transforms the matrix A into an upper triangular matrix. Start with subtracting a_{i1}/a_{11} times the first row from rows $2 \cdots n$

$$\begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}^{\mathbf{T}}{}_2 - l_{21}\mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T - l_{n1}\mathbf{a}_1^T \end{pmatrix} \tag{5.8}$$

which can be written as a multiplication

$$A^{(1)} = L_1 A \tag{5.9}$$

with the Frobenius matrix

$$L_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ -l_{31} & & 1 & \\ \vdots & & & \ddots \\ -l_{n1} & & & 1 \end{pmatrix} \qquad l_{i1} = \frac{a_{i1}}{a_{11}}. \tag{5.10}$$

The result has the form

$$A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n-1} & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n-1}^{(1)} & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \cdots & \cdots & a_{3n}^{(1)} \\ 0 & \vdots & & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & \cdots & a_{nn}^{(1)} \end{pmatrix}. \tag{5.11}$$

Now subtract $\frac{a_{i2}}{a_{22}}$ times the second row from rows $3 \cdots n$. This can be formulated as

$$A^{(2)} = L_2 A^{(1)} = L_2 L_1 A \tag{5.12}$$

with

$$L_2 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -l_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & -l_{n2} & & & 1 \end{pmatrix} \qquad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}. \tag{5.13}$$

The result is

$$A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}. \tag{5.14}$$

Continue until an upper triangular matrix results after n−1 steps:

$$A^{(n-1)} = L_{n-1} A^{(n-2)} \tag{5.15}$$

$$L_{n-1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -l_{n,n-1} & 1 \end{pmatrix} \qquad l_{n,n-1} = \frac{a_{n,n-1}^{(n-2)}}{a_{n-1,n-1}^{(n-2)}} \tag{5.16}$$

$$A^{(n-1)} = L_{n-1} L_{n-2} \cdots L_2 L_1 A = U \tag{5.17}$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}. \tag{5.18}$$

The transformed system of equations

$$U\mathbf{x} = \mathbf{y} \qquad \mathbf{y} = L_{n-1} L_{n-1} \cdots L_2 L_1 \mathbf{b} \tag{5.19}$$

can be solved easily by backward substitution:

$$x_n = \frac{1}{u_{nn}} y_n \tag{5.20}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}} \tag{5.21}$$

$$\vdots \tag{5.22}$$

Alternatively the matrices L_i can be inverted:

$$L_1^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & & 1 & \\ \vdots & & & \ddots \\ l_{n1} & & & 1 \end{pmatrix} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & l_{n,n-1} \; 1 \end{pmatrix}. \tag{5.23}$$

This gives

$$A = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} U. \tag{5.24}$$

The product of the inverted matrices is a lower triangular matrix:

$$L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & \vdots & & \ddots \\ l_{n1} & l_{n2} & & 1 \end{pmatrix}$$

$$\vdots$$

$$L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n-1,1} & l_{n-1,2} & \cdots & 1 \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} \; 1 \end{pmatrix}. \tag{5.25}$$

Hence the matrix A becomes decomposed into a product of a lower and an upper triangular matrix

$$A = LU \tag{5.26}$$

which can be used to solve the system of (5.2).

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b} \tag{5.27}$$

in two steps:

$$L\mathbf{y} = \mathbf{b} \tag{5.28}$$

which can be solved from the top

$$y_1 = b_1 \tag{5.29}$$

$$y_2 = b_2 - l_{21} y_1 \tag{5.30}$$

$$\vdots \tag{5.31}$$

and

$$U\mathbf{x} = \mathbf{y} \tag{5.32}$$

which can be solved from the bottom

$$x_n = \frac{1}{u_{nn}} y_n \tag{5.33}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}}. \tag{5.34}$$

$$\vdots \tag{5.35}$$

5.1.1 Pivoting

To improve numerical stability and to avoid division by zero pivoting is used. Most common is partial pivoting. In every step the order of the equations is changed in order to maximize the pivoting element $a_{k,k}$ in the denominator. This gives LU decomposition of the matrix PA where P is a permutation matrix. P is not needed explicitly. Instead an index vector is used which stores the new order of the equations

$$P \begin{pmatrix} 1 \\ \vdots \\ N \end{pmatrix} = \begin{pmatrix} i_1 \\ \vdots \\ i_N \end{pmatrix}. \tag{5.36}$$

Total pivoting exchanges rows and columns of A. This can be time consuming for larger matrices.

If the elements of the matrix are of different orders of magnitude it can be necessary to balance the matrix, for instance by normalizing all rows of A. This can be also achieved by selecting the maximum of

$$\frac{a_{ik}}{\sum_j |a_{ij}|} \tag{5.37}$$

as the pivoting element.

5.1.2 Direct LU Decomposition

LU decomposition can be also performed in a different order [36]. For symmetric positive definite matrices there exists the simpler and more efficient Cholesky method decomposes the matrix into the product LL^T of a lower triangular matrix and its transpose [37].

5.2 QR Decomposition

The Gaussian elimination method can become numerically unstable [38]. An alternative method to solve a system of linear equations uses the decomposition [39]

$$A = QR \tag{5.38}$$

with a unitary matrix $Q^\dagger Q = 1$ (an orthogonal matrix $Q^T Q = 1$ if A is real) and an upper right triangular matrix R. The system of linear equations (5.2) is simplified by multiplication with $Q^\dagger = Q^{-1}$

$$QR\mathbf{x} = A\mathbf{x} = \mathbf{b} \tag{5.39}$$

$$R\mathbf{x} = Q^\dagger \mathbf{b}. \tag{5.40}$$

Such a system with upper triangular matrix is easily solved (see 5.32).

5.2.1 QR Decomposition by Orthogonalization

Gram-Schmidt orthogonalization [2, 39] provides a simple way to perform a QR decomposition. It is used for symbolic calculations and also for least square fitting (11.1.2) but can become numerically unstable.

From the decomposition $A = QR$ we have

$$a_{ik} = \sum_{j=1}^{k} q_{ij} r_{jk} \tag{5.41}$$

$$\mathbf{a}_k = \sum_{j=1}^{k} r_{jk} \mathbf{q}_j \tag{5.42}$$

which gives the k-th column vector $\mathbf{a}_k$ of A as a linear combination of the orthonormal vectors $\mathbf{q}_1 \cdots \mathbf{q}_k$. Similarly $\mathbf{q}_k$ is a linear combination of the first k columns of A. With the help of the Gram-Schmidt method r_{jk} and $\mathbf{q}_j$ are calculated as follows:

$$r_{11} := |a_1| \tag{5.43}$$

$$\mathbf{q}_1 := \frac{\mathbf{a}_1}{r_{11}} \tag{5.44}$$

For $k = 2, \cdots n$:

$$r_{ik} := \mathbf{q}_i \mathbf{a}_k \quad i = 1 \cdots k-1 \tag{5.45}$$

$$\mathbf{b}_k := \mathbf{a}_k - r_{1k}\mathbf{q}_1 - \cdots r_{k-1,k}\mathbf{q}_{k-1} \tag{5.46}$$

$$r_{kk} := |\mathbf{b}_k| \tag{5.47}$$

$$\mathbf{q}_k := \frac{\mathbf{b}_k}{r_{kk}}. \tag{5.48}$$

Obviously now

$$\mathbf{a}_k = r_{kk}\mathbf{q}_k + r_{k-1,k}\mathbf{q}_{k-1} + \cdots r_{1k}\mathbf{q}_1 \tag{5.49}$$

since per definition

$$\mathbf{q}_i \mathbf{a}_k = r_{ik} \quad i = 1 \cdots k \tag{5.50}$$

and

$$r_{kk}^2 = |\mathbf{b}_k|^2 = |\mathbf{a}_k|^2 + r_{1k}^2 + \cdots r_{k-1,k}^2 - 2r_{1k}^2 - \cdots - 2r_{k-1,k}^2. \tag{5.51}$$

Hence

$$\mathbf{q}_k \mathbf{a}_k = \frac{1}{r_{kk}}(\mathbf{a}_k - r_{1k}\mathbf{q}_1 \cdots r_{k-1,k}\mathbf{q}_{k-1})\mathbf{a}_k = \frac{1}{r_{kk}}(|a_k|^2 - r_{1k}^2 \cdots - r_{k-1,k}^2) = r_{kk}. \tag{5.52}$$

Orthogonality gives

$$\mathbf{q}_i \mathbf{a}_k = 0 \quad i = k+1 \cdots n. \tag{5.53}$$

In matrix notation we have finally

$$A = (\mathbf{a}_1 \cdots \mathbf{a}_n) = (\mathbf{q}_1 \cdots \mathbf{q}_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}. \tag{5.54}$$

If the columns of A are almost linearly dependent, numerical stability can be improved by an additional orthogonalization step

$$\mathbf{b}_k \to \mathbf{b}_k - (\mathbf{q}_1 \mathbf{b}_k)\mathbf{q}_1 - \cdots (\mathbf{q}_{k-1}\mathbf{b}_k)\mathbf{q}_{k-1} \tag{5.55}$$

after (5.46) which can be iterated several times to improve the results [2, 40].

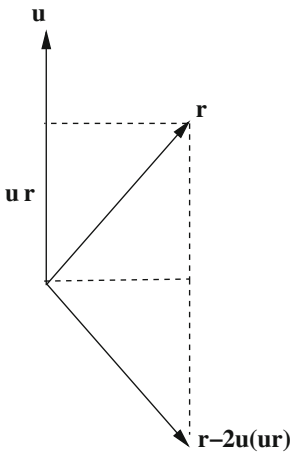

Fig. 5.1 (Householder transformation) Geometrically the Householder transformation (5.56) is a mirror operation with respect to a plane with normal vector **u**

5.2.2 *QR Decomposition by Householder Reflections*

Numerically stable algorithms use a series of transformations with unitary matrices, mostly Householder reflections (Fig. 5.1) [2][1] which have the form

$$P = P^T = 1 - 2\mathbf{u}\mathbf{u}^T \tag{5.56}$$

with a unit vector

$$|\mathbf{u}| = 1. \tag{5.57}$$

Obviously P is an orthogonal matrix since

$$P^T P = (1 - 2\mathbf{u}\mathbf{u}^T)(1 - 2\mathbf{u}\mathbf{u}^T) = 1 - 4\mathbf{u}\mathbf{u}^T + 4\mathbf{u}\mathbf{u}^T\mathbf{u}\mathbf{u}^T = 1. \tag{5.58}$$

In the first step we try to find a vector **u** such that the first column vector of A

$$\mathbf{a}_1 = \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix} \tag{5.59}$$

[1] Alternatively Givens rotations [39] can be employed which need slightly more floating point operations.

is transformed into a vector along the 1-axis

$$P\mathbf{a}_1 = \left(1 - 2\mathbf{u}\mathbf{u}^T\right)\mathbf{a}_1 = k\mathbf{e}_1 = \begin{pmatrix} k \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5.60}$$

Multiplication with the transpose vector gives

$$k^2 = (P\mathbf{a}_1)^T\, P\mathbf{a}_1 = \mathbf{a}_1^T P^T P\mathbf{a}_1 = |\mathbf{a}_1|^2 \tag{5.61}$$

and

$$k = \pm|\mathbf{a}_1| \tag{5.62}$$

can have both signs. From (5.60) we have

$$\mathbf{a}_1 - 2\mathbf{u}(\mathbf{u}\mathbf{a}_1) = k\mathbf{e}_1. \tag{5.63}$$

Multiplication with $\mathbf{a}_1^T$ gives

$$2(\mathbf{u}\mathbf{a}_1)^2 = |\mathbf{a}_1|^2 - k(\mathbf{a}_1\mathbf{e}_1) \tag{5.64}$$

and since

$$|\mathbf{a}_1 - k\mathbf{e}_1|^2 = |\mathbf{a}_1|^2 + k^2 - 2k(\mathbf{a}_1\mathbf{e}_1) = 2|\mathbf{a}_1|^2 - 2k(\mathbf{a}_1\mathbf{e}_1) \tag{5.65}$$

we have

$$2(\mathbf{u}\mathbf{a}_1)^2 = \frac{1}{2}|\mathbf{a}_1 - k\mathbf{e}_1|^2 \tag{5.66}$$

and from (5.63) we find

$$\mathbf{u} = \frac{\mathbf{a}_1 - k\mathbf{e}_1}{2\mathbf{u}\mathbf{a}_1} = \frac{\mathbf{a}_1 - k\mathbf{e}_1}{|\mathbf{a}_1 - k\mathbf{e}_1|}. \tag{5.67}$$

To avoid numerical extinction the sign of k is chosen such that

$$\sigma = \text{sign}(k) = -\text{sign}(a_{11}). \tag{5.68}$$

Then,

$$\mathbf{u} = \frac{1}{\sqrt{2(a_{11}^2 + \cdots + a_{n1}^2) + 2|a_{11}|\sqrt{a_{11}^2 + \cdots + a_{n1}^2}}} \begin{pmatrix} \text{sign}(a_{11})\left(|a_{11}| + \sqrt{a_{11}^2 + a_{21}^2 + \ldots a_{n1}^2}\right) \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \tag{5.69}$$

$$2\mathbf{u}\mathbf{u}^T\mathbf{a}_1 = \begin{pmatrix} \text{sign}(a_{11})\left(|a_{11}| + \sqrt{a_{11}^2 + a_{21}^2 + \ldots a_{n1}^2}\right) \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \times \frac{1}{(a_{11}^2 + \cdots + a_{n1}^2) + |a_{11}|\sqrt{a_{11}^2 + \cdots + a_{n1}^2}} \left(a_{11}^2 + |a_{11}|\sqrt{a_{11}^2 + \ldots a_{n1}^2} + a_{21}^2 + \cdots + a_{n1}^2\right) \tag{5.70}$$

and the Householder transformation of the first column vector of A gives

$$(1 - 2\mathbf{u}\mathbf{u}^T)\mathbf{a}_1 = \begin{pmatrix} -\text{sign}(a_{11})\sqrt{a_{11}^2 \ldots a_{n1}^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5.71}$$

Thus after the first step a matrix results of the form

$$A^{(1)} = P_1 A = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \ldots & a_{nn}^{(1)} \end{pmatrix}.$$

In the following (n-2) steps further Householder reflections are applied in the subspace $k \leq i, j \leq n$ to eliminate the elements

$$a_{k+1,k} \ldots a_{n,k}$$

of the $k - th$ row vector below the diagonal of the matrix:

$$A^{(k-1)} = P_{k-1} \dots P_1 A = \begin{pmatrix} a_{11}^{(1)} & \dots & a_{1,k-1}^{(1)} & a_{1,k}^{(1)} & \dots & a_{1,n}^{(1)} \\ 0 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & & a_{k-1,n}^{(k-1)} \\ \vdots & \vdots & 0 & a_{k,k}^{(k-1)} & & a_{k,n}^{(k-1)} \\ \vdots & \vdots & \vdots & a_{k+1,k}^{(k-1)} & & a_{k+1,n}^{(k-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} \end{pmatrix} \tag{5.72}$$

$$P_k = \begin{pmatrix} 1_1 & & & \\ & \ddots & & \\ & & 1_{k-1} & \\ & & & 1 - 2\mathbf{u}\mathbf{u}^T \end{pmatrix}.$$

Finally an upper triangular matrix results

$$A^{(n-1)} = (P_{n-1} \dots P_1)A = R = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1,n-1}^{(1)} & a_{1,n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2,n-1}^{(2)} & a_{2,n}^{(2)} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & 0 & \dots & 0 & a_{n,n}^{(n-1)} \end{pmatrix}. \tag{5.73}$$

If the orthogonal matrix Q is needed explicitly additional numerical operations are necessary to form the product

$$Q = (P_{n-1} \dots P_1)^T. \tag{5.74}$$

5.3 Linear Equations with Tridiagonal Matrix

Linear equations with the form

$$b_1 x_1 + c_1 x_2 = r_1 \tag{5.75}$$

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = r_i \quad i = 2 \cdots (n-1) \tag{5.76}$$

$$a_n x_{n-1} + b_n x_n = r_n \tag{5.77}$$

can be solved very efficiently with a specialized Gaussian elimination method.[2] They are important for cubic spline interpolation or second derivatives. We begin by eliminating a_2. To that end we multiply the first line with a_2/b_1 and subtract it from the first line. The result is the equation

$$\beta_2 x_2 + c_2 x_3 = \rho_2 \tag{5.78}$$

with the abbreviations

$$\beta_2 = b_2 - \frac{c_1 a_2}{b_1} \quad \rho_2 = r_2 - \frac{r_1 a_2}{b_1}. \tag{5.79}$$

We iterate this procedure

$$\beta_i x_i + c_i x_{i+1} = \rho_i \tag{5.80}$$

$$\beta_i = b_i - \frac{c_{i-1} a_i}{\beta_{i-1}} \quad \rho_i = r_i - \frac{\rho_{i-1} a_i}{\beta_{i-1}} \tag{5.81}$$

until we reach the n-th equation, which becomes simply

$$\beta_n x_n = \rho_n \tag{5.82}$$

$$\beta_n = b_n - \frac{c_{n-1} a_n}{\beta_{n-1}} \quad \rho_n = r_n - \frac{\rho_{n-1} a_n}{\beta_{n-1}}. \tag{5.83}$$

Now we immediately have

$$x_n = \frac{\rho_n}{\beta_n} \tag{5.84}$$

and backward substitution gives

$$x_{i-1} = \frac{\rho_{i-1} - c_{i-1} x_i}{\beta_{i-1}} \tag{5.85}$$

and finally

$$x_1 = \frac{r_1 - c_1 x_2}{\beta_2}. \tag{5.86}$$

This algorithm can be formulated as LU decomposition: Multiplication of the matrices

[2] This algorithm is only well behaved if the matrix is diagonal dominant $|b_i| > |a_i| + |c_i|$.

$$L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \qquad U = \begin{pmatrix} \beta_1 & c_1 & & & \\ & \beta_2 & c_2 & & \\ & & \beta_3 & c_3 & \\ & & & \ddots & \\ & & & & \beta_n \end{pmatrix} \tag{5.87}$$

gives

$$LU = \begin{pmatrix} \beta_1 & c_1 & & & & & \\ & \ddots & \ddots & & & & \\ & & \ddots & \ddots & & & \\ & & l_i\beta_{i-1} & (l_ic_{i-1}+\beta_i) & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & & \\ & & & & & l_n\beta_{n-1} & (l_nc_{n-1}+\beta_n) \end{pmatrix} \tag{5.88}$$

which coincides with the matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & & \\ a_2 & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{pmatrix} \tag{5.89}$$

if we choose

$$l_i = \frac{a_i}{\beta_{i-1}} \tag{5.90}$$

since then from (5.81)

$$b_i = \beta_i + l_i c_{i-1} \tag{5.91}$$

and

$$l_i\beta_{i-1} = a_i. \tag{5.92}$$

5.4 Cyclic Tridiagonal Systems

Periodic boundary conditions lead to a small perturbation of the tridiagonal matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & & a_1 \\ a_2 & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & & & a_n & b_n \end{pmatrix}. \tag{5.93}$$

The system of equations

$$A\mathbf{x} = \mathbf{r} \tag{5.94}$$

can be reduced to a tridiagonal system [41] with the help of the Sherman–Morrison formula [42], which states that if A_0 is an invertible matrix and $\mathbf{u}$, $\mathbf{v}$ are vectors and

$$1 + \mathbf{v}^T A_0^{-1} \mathbf{u} \neq 0 \tag{5.95}$$

then the inverse of the matrix[3]

$$A = A_0 + \mathbf{u}\mathbf{v}^T \tag{5.96}$$

is given by

$$A^{-1} = A_0^{-1} - \frac{A_0^{-1}\mathbf{u}\mathbf{v}^T A_0^{-1}}{1 + \mathbf{v}^T A_0^{-1}\mathbf{u}}. \tag{5.97}$$

We choose

$$\mathbf{u}\mathbf{v}^T = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ c_n \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 & \frac{a_1}{\alpha} \end{pmatrix} = \begin{pmatrix} \alpha & & & a_1 \\ & & & \\ & & & \\ c_n & & & \frac{a_1 c_n}{\alpha} \end{pmatrix}. \tag{5.98}$$

[3] Here $\mathbf{u}\mathbf{v}^T$ is the outer or matrix product of the two vectors.

Then

$$A_0 = A - \mathbf{u}\mathbf{v}^T = \begin{pmatrix} (b_1-\alpha) & c_1 & & & & 0 \\ a_2 & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & & & & & a_n & (b_n - \frac{a_1 c_n}{\alpha}) \end{pmatrix} \tag{5.99}$$

is tridiagonal. The free parameter α has to be chosen such that the diagonal elements do not become too small. We solve the system (5.94) by solving the two tridiagonal systems

$$\begin{aligned} A_0\mathbf{x}_0 &= \mathbf{r} \\ A_0\mathbf{q} &= \mathbf{u} \end{aligned} \tag{5.100}$$

and compute $\mathbf{x}$ from

$$\mathbf{x} = A^{-1}\mathbf{r} = A_0^{-1}\mathbf{r} - \frac{(A_0^{-1}\mathbf{u})\mathbf{v}^T(A_0^{-1}\mathbf{r})}{1+\mathbf{v}^T(A_0^{-1}\mathbf{u})} = \mathbf{x}_0 - \mathbf{q}\frac{\mathbf{v}^T\mathbf{x}_0}{1+\mathbf{v}^T\mathbf{q}}. \tag{5.101}$$

5.5 Linear Stationary Iteration

Discretized differential equations often lead to systems of equations with large sparse matrices, which have to be solved by iterative methods which, starting from an initial guess $\mathbf{x}_0$ (often simply $\mathbf{x}_0 = 0$ or $\mathbf{x}_0 = \mathbf{b}$) construct a sequence of improved vectors by the iteration

$$\mathbf{x}^{(n+1)} = \Phi(\mathbf{x}^{(n)}, \mathbf{b}) \tag{5.102}$$

which under certain conditions converges to the fixed point

$$\mathbf{x}_{FP} = \Phi(\mathbf{x}_{FP}, \mathbf{b}) \tag{5.103}$$

which solves the system of equations

$$A\mathbf{x} = \mathbf{b}. \tag{5.104}$$

A linear iterative method is called stationary, if it has the form

$$\mathbf{x}^{(n+1)} = B\mathbf{x}^{(n)} + C\mathbf{b} \tag{5.105}$$

where the matrices B (the so called iteration matrix) and C are constant and do not depend on the iteration count n. A fixed point of (5.105) solves (5.104) and hence the method is consistent, if

$$\mathbf{x} = B\mathbf{x} + C\mathbf{b} = B\mathbf{x} + CA\mathbf{x} \tag{5.106}$$

and hence

$$B = I - CA \tag{5.107}$$

which brings the iteration to the general form

$$\mathbf{x}^{(n+1)} = (I - CA)\mathbf{x}^{(n)} + C\mathbf{b} = \mathbf{x}^{(n)} - C(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.108}$$

$$\mathbf{r}^{(n+1)} = (1 - AC)\mathbf{r}^{(n)}. \tag{5.109}$$

5.5.1 Richardson-Iteration

The simplest of these methods uses $C = \omega I$ with a damping parameter ω. It iterates according to

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.110}$$

$$\mathbf{r}^{(n+1)} = (1 - \omega A)\mathbf{r}^{(n)}. \tag{5.111}$$

The Richardson iteration is not of much practical use. It serves as the prototype of a linear stationary method. To improve convergence (5.104) usually has to be preconditioned by multiplication with a suitable matrix P

$$PA\mathbf{x} = P\mathbf{b} \tag{5.112}$$

for which the Richardson iteration

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega P(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.113}$$

is of the general form (5.108).

5.5.2 Matrix Splitting Methods

Let us divide the matrix A into two (non singular) parts [2]

$$A = A_1 + A_2 \tag{5.114}$$

where A_1 can be easily inverted and rewrite (5.104) as

$$A_1\mathbf{x} = \mathbf{b} - A_2\mathbf{x} \tag{5.115}$$

which defines the iteration

$$\Phi(\mathbf{x}) = -A_1^{-1}A_2\mathbf{x} + A_1^{-1}\mathbf{b} \tag{5.116}$$

$$= -A_1^{-1}(A - A_1)\mathbf{x} + A_1^{-1}\mathbf{b} = \mathbf{x} - A_1^{-1}(A\mathbf{x} - \mathbf{b}). \tag{5.117}$$

5.5.3 Jacobi Method

Jacobi divides the matrix A into its diagonal and two triangular matrices [43]:

$$A = L + U + D. \tag{5.118}$$

For A_1 the diagonal part is chosen

$$A_1 = D \tag{5.119}$$

giving

$$\mathbf{x}^{(n+1)} = -D^{-1}(A - D)\mathbf{x}^{(n)} + D^{-1}\mathbf{b} \tag{5.120}$$

which reads explicitly

$$x_i^{(n+1)} = -\frac{1}{a_{ii}}\sum_{j\neq i} a_{ij}x_j^{(n)} + \frac{1}{a_{ii}}b_i. \tag{5.121}$$

This method is stable but converges rather slowly. Reduction of the error by a factor of 10^{-p} needs about $\frac{pN}{2}$ iterations. N grid points have to be evaluated in each iteration and the method scales with $O(N^2)$ [44].

5.5.4 Gauss-Seidel Method

With

$$A_1 = D + L \tag{5.122}$$

the iteration becomes

$$(D + L)\mathbf{x}^{(n+1)} = -U\mathbf{x}^{(n)} + \mathbf{b} \tag{5.123}$$

which has the form of a system of equations with triangular matrix [45]. It reads explicitly

$$\sum_{j \le i} a_{ij} x_j^{(n+1)} = -\sum_{j > i} a_{ij} x_j^{(n)} + b_i. \tag{5.124}$$

Forward substitution starting from x_1 gives

$$\begin{aligned}
i = 1: \quad & x_1^{(n+1)} = \frac{1}{a_{11}} \left(-\sum_{j \ge 2} a_{ij} x_j^{(n)} + b_1 \right) \\
i = 2: \quad & x_2^{(n+1)} = \frac{1}{a_{22}} \left(-a_{21} x_1^{(n+1)} - \sum_{j \ge 3} a_{ij} x_j^{(n)} + b_2 \right) \\
i = 3: \quad & x_3^{(n+1)} = \frac{1}{a_{33}} \left(-a_{31} x_1^{(n+1)} - a_{32} x_2^{(n+1)} - \sum_{j \ge 4} a_{ij} x_j^{(n)} + b_3 \right) \\
& \vdots \\
& x_i^{(n+1)} = \frac{1}{a_{ii}} \left(-\sum_{j < i} a_{ij} x_j^{(n+1)} - \sum_{j > i} a_{ij} x_j^{(n)} + b_i \right).
\end{aligned} \tag{5.125}$$

This looks very similar to the Jacobi method. But here all changes are made immediately. Convergence is slightly better (roughly a factor of 2) and the numerical effort is reduced [44].

5.5.5 Damping and Successive Over-relaxation

Convergence can be improved [44] by combining old and new values. Starting from the iteration

$$A_1\mathbf{x}^{(n+1)} = (A_1 - A)\mathbf{x}^{(n)} + \mathbf{b} \tag{5.126}$$

we form a linear combination with

$$D\mathbf{x}^{(n+1)} = D\mathbf{x}^{(n)} \tag{5.127}$$

giving the new iteration equation

$$((1-\omega)D + \omega A_1)\mathbf{x}^{(n+1)} = ((1-\omega)D + \omega A_1 - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b}. \tag{5.128}$$

In case of the Jacobi method with $D = A_1$ we have

$$D\mathbf{x}^{(n+1)} = (D - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} \tag{5.129}$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega D^{-1}(A\mathbf{x} - \mathbf{b}) \tag{5.130}$$

which can be also obtained directly from (5.113).

Explicitly,

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j\neq i} a_{ij}x_j^{(n)} + b_i\right). \tag{5.131}$$

The changes are damped ($0 < \omega < 1$) or exaggerated ($1 < \omega < 2$).

In case of the Gauss-Seidel method with $A_1 = D + L$ the new iteration[4] (5.128) is

$$(D + \omega L)\mathbf{x}^{(n+1)} = (D + \omega L - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} = (1-\omega)D\mathbf{x}^{(n)} - \omega U\mathbf{x}^{(n)} + \omega\mathbf{b} \tag{5.132}$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \left(\frac{1}{\omega}D + L\right)^{-1}(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.133}$$

or explicitly

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j<i} a_{ij}x_j^{(n+1)} - \sum_{j>i} a_{ij}x_j^{(n)} + \mathbf{b}\right). \tag{5.134}$$

[4] This is also known as the method of successive over-relaxation (SOR) and differs from the damped Gauss-Seidel method which follows from (5.113).

It can be shown that the successive over-relaxation method converges only for $0 < \omega < 2$. For optimal choice of ω about $\frac{1}{3}p\sqrt{N}$ iterations are needed to reduce the error by a factor of 10^{-p}. The order of the method is $O(N^{\frac{3}{2}})$ which is comparable to the most efficient matrix inversion methods [44].

5.6 Non Stationary Iterative Methods

Non stationary methods use a more general iteration

$$\mathbf{x}^{(n+1)} = \Phi_n(\mathbf{x}_n) \tag{5.135}$$

where the iteration function differs from step to step. The method can be formulated as a direction search

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda_n \mathbf{s}_n \tag{5.136}$$

with direction vectors $\mathbf{s}_n$ and step lengths λ_n. The residual

$$\mathbf{r}^{(n+1)} = A\mathbf{x}^{(n+1)} - \mathbf{b} = \mathbf{r}^{(n)} + \lambda_n A\mathbf{s}_n \tag{5.137}$$

is used as a measure of the remaining error since the exact solution $\mathbf{x}_{FP}$ together with the error vector

$$\mathbf{d}^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}_{FP} \tag{5.138}$$

are unknown. Both are, however, related by

$$A\mathbf{d}^{(n)} = A\mathbf{x}^{(n)} - A\mathbf{x}_{FP} = A\mathbf{x}^{(n)} - \mathbf{b} = \mathbf{r}^{(n)}. \tag{5.139}$$

5.6.1 Krylov Space Methods

Solution of the linear system

$$A\mathbf{x} = \mathbf{b} \tag{5.140}$$

can be formulated as a search for the minimum

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathbb{N}}} ||A\mathbf{x} - \mathbf{b}||. \tag{5.141}$$

General iterative methods look for the minimum residual in a subspace of $\mathbb{R}^{\mathbb{N}}$ which is increased at each step. The Richardson iteration, e.g. iterates

$$\begin{aligned}
&\mathbf{x}^{(n+1)} = (1-\omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} = \mathbf{x}^{(n)} - \omega\mathbf{r}^{(n)} \qquad (5.142)\\
&\mathbf{r}^{(n+1)} = A(\mathbf{x}^{(n)} - \omega\mathbf{r}^{(n)}) - \mathbf{b} = (1-\omega A)\mathbf{r}^{(n)}\\
&\mathbf{r}_0 = A\mathbf{x}_0 - \mathbf{b}\\
&\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \omega\mathbf{r}^{(0)}\\
&\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - \omega A\mathbf{r}^{(0)}\\
&\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \omega\mathbf{r}^{(1)} = \mathbf{x}^{(0)} - 2\omega\mathbf{r}^{(0)} + \omega^2 A\mathbf{r}^{(0)}\\
&\mathbf{r}^{(2)} = (1-\omega A)\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - 2\omega A\mathbf{r}^{(0)} + \omega^2 A^2\mathbf{r}^{(0)}\\
&\vdots
\end{aligned}$$

Obviously,

$$\mathbf{x}^{(n)} - \mathbf{x}_0 \in K_n(A, \mathbf{r}^{(0)}) \quad \mathbf{r}^{(n)} \in K_{n+1}(A, \mathbf{r}^{(0)})$$

with the definition of the n-th Krylov subspace[5]

$$K_n(A, \mathbf{r}^{(0)}) = \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots A^{n-1}\mathbf{r}^{(0)}\}. \qquad (5.143)$$

5.6.2 *Minimization Principle for Symmetric Positive Definite Systems*

If the matrix A is symmetric and positive definite, we consider the quadratic form defined by

$$h(\mathbf{x}) = h_0 - \mathbf{x}^T\mathbf{b} + \frac{1}{2}\mathbf{x}^T A\mathbf{x}. \qquad (5.144)$$

At a local minimum the gradient

$$\nabla h(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = \mathbf{r} \qquad (5.145)$$

is zero and therefore the minimum of h is also a solution of the linear system of equations

$$A\mathbf{x} = \mathbf{b}. \qquad (5.146)$$

[5]For the most common choice $\mathbf{x}_0 = 0$ we have $\mathbf{r}^{(0)} = -\mathbf{b}$ and $\mathbf{x}_0 + K_n(A, \mathbf{r}^{(0)}) = K_n(A, \mathbf{r}^{(0)}) = K_n(A, \mathbf{b})$.

5.6.3 Gradient Method

The simple Richardson iteration (Sect. 5.5.1) uses

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega \mathbf{r}^{(n)} \tag{5.147}$$

with a constant value of ω.

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \omega A \mathbf{r}^{(n)}. \tag{5.148}$$

Let us now optimize the step width along the direction of the gradient vector. From

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}\omega} |\mathbf{r}^{(n+1)}|^2 = \mathbf{r}^{(n)T}(1-\omega A)(1-\omega A)\mathbf{r}^{(n)} \\ &= \mathbf{r}^{(n)T}(-2A + 2\omega A^2)\mathbf{r}^{(n)} = -2\mathbf{r}^{(n)T} A \mathbf{r}^{(n)} + 2\omega |A\mathbf{r}^{(n)}|^2 \\ \mathbf{r} &= 2\mathbf{r}^{(n)T}(-1+\omega A) A \mathbf{r}^{(n)} \end{aligned} \tag{5.149}$$

we find the optimum value

$$\omega^{(n)} = \frac{\mathbf{r}^{(n)T} A \mathbf{r}^{(n)}}{|A\mathbf{r}^{(n)}|^2}. \tag{5.150}$$

The residuals[6]

$$\mathbf{r}_0 = -\mathbf{b} \tag{5.151}$$

$$\mathbf{r}^{(1)} = -\mathbf{b} + \omega^{(1)} A \mathbf{b} \tag{5.152}$$

$$\mathbf{r}^{(2)} = -\mathbf{b} + (\omega^{(1)} + \omega^{(2)}) A \mathbf{b} - \omega^{(2)} \omega^{(1)} A^2 \mathbf{b} \tag{5.153}$$

$$\vdots$$

etc. obviously are in the Krylov subspace

$$\mathbf{r}^{(n)} \in K_{n+1}(A, \mathbf{b}) \tag{5.154}$$

and so are the approximate solutions

$$\mathbf{x}^{(1)} = \omega^{(1)} \mathbf{b} \tag{5.155}$$

$$\mathbf{x}^{(2)} = (\omega^{(1)} + \omega^{(2)}) \mathbf{b} - \omega^{(2)} \omega^{(1)} A \mathbf{b} \tag{5.156}$$

$$\vdots$$

$$\mathbf{x}^{(n)} \in K_n(A, \mathbf{b}). \tag{5.157}$$

[6] We assume $\mathbf{x}_0 = 0$.

5.6.4 Conjugate Gradients Method

The gradient method gives an approximate solution

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(0)} - \omega^{(0)}\mathbf{r}^{(0)} - \omega^{(1)}\mathbf{r}^{(1)} \cdots - \omega^{(n)}\mathbf{r}^{(n)} \tag{5.158}$$

but the previously chosen $\omega^{(0)} \ldots \omega^{(n-1)}$are not optimal since the gradient vectors $\mathbf{r}^{(0)} \ldots \mathbf{r}^{(n)}$are not orthogonal. We want to optimize the solution within the space spanned by the gradients for which a new basis $\mathbf{s}^{(0)} \ldots \mathbf{s}^{(n)}$is introduced which will be determined later

$$K_{n+1} = \text{span}(\mathbf{r}^{(0)} \ldots \mathbf{r}^{(n)}) = \text{span}(\mathbf{s}^{(0)} \ldots \mathbf{s}^{(n)}). \tag{5.159}$$

Using $\mathbf{s}^{(n)}$as search direction the iteration becomes

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda^{(n)}\mathbf{s}^{(n)} \tag{5.160}$$

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \lambda^{(n)}A\mathbf{s}^{(n)}. \tag{5.161}$$

After n+1 steps

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(0)} + \sum \lambda^{(j)}A\mathbf{s}^{(\mathbf{j})} = A(\mathbf{d}^{(0)} + \sum \lambda^{(j)}\mathbf{s}^{(j)}). \tag{5.162}$$

Multiplication with $\mathbf{s}^{(m)}$ gives

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = \mathbf{s}^{(m)T}A\mathbf{d}^{(0)} + \sum \lambda^{(j)}\mathbf{s}^{(m)T}A\mathbf{s}^{(j)} \tag{5.163}$$

which, after introduction of the $A-$scalar product which is defined for a symmetric and positive definite matrix A as

$$(\mathbf{x}, \mathbf{y})_A = \mathbf{x}^T A\mathbf{y} \tag{5.164}$$

becomes

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = (\mathbf{s}^{(m)}, \mathbf{d}^{(0)})_A + \sum_{j=0}^{n} \lambda^{(j)}(\mathbf{s}^{(m)}, \mathbf{s}^{(j)})_A$$

which simplifies considerably if we assume A-orthogonality of the search directions

$$(\mathbf{s}^{(m)}, \mathbf{s}^{(j)}) = 0 \quad \text{for } m \neq j \tag{5.165}$$

because then

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = (\mathbf{s}^{(m)}, \mathbf{d}^{(0)})_A + \lambda^{(m)}(\mathbf{s}^{(m)}, \mathbf{s}^{(m)})_A = \mathbf{s}^{(m)T}\mathbf{r}^{(0)} + \lambda^{(m)}\mathbf{s}^{(m)T}A\mathbf{s}^{(m)}. \tag{5.166}$$

If we choose

$$\lambda^{(m)} = -\frac{\mathbf{s}^{(m)T}\mathbf{r}^{(0)}}{\mathbf{s}^{(m)T}A\mathbf{s}^{(m)}} \tag{5.167}$$

the projection of the new residual $\mathbf{r}^{(n+1)}$onto K_{n+1} vanishes, i.e. this is the optimal choice of the parameters $\lambda^{(0)} \ldots \lambda^{(n)}$.

Obviously the first search vector must have the direction of $\mathbf{r}^{(0)}$ to span the same one-dimensional space. Therefore we begin the iteration with

$$\mathbf{s}^{(0)} = \mathbf{r}^{(0)} \tag{5.168}$$

$$\lambda^{(0)} = -\frac{|\mathbf{r}^{(0)}|^2}{\mathbf{r}^{(0)T}A\mathbf{r}^{(0)}} \tag{5.169}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda^{(0)}\mathbf{s}^{(\mathbf{0})}. \tag{5.170}$$

For the next step we apply Gram-Schmidt orthogonalization

$$\mathbf{s}^{(1)} = \mathbf{r}^{(1)} - \mathbf{s}^{(0)}\frac{(\mathbf{r}^{(1)}, \mathbf{s}^{(0)})_A}{(\mathbf{s}^{(0)}, \mathbf{s}^{(0)})_A}. \tag{5.171}$$

For all further steps we have to orthogonalize $\mathbf{s}^{(n+1)}$with respect to all of $\mathbf{s}^{(n)} \ldots \mathbf{s}^{(0)}$. But, fortunately, the residual $\mathbf{r}^{(n+1)}$is already $A-$orthogonal to $\mathbf{s}^{(n-1)} \ldots \mathbf{s}^{(0)}$. This can be seen from (5.161)

$$\mathbf{r}^{(j+1)} - \mathbf{r}^{(j)} = \lambda^{(j)}A\mathbf{s}^{(j)} \tag{5.172}$$

$$(\mathbf{r}^{(n+1)}, \mathbf{s}^{(j)})_A = \mathbf{r}^{(n+1)T}A\mathbf{s}^{(j)} = \frac{1}{\lambda^{(j)}}\mathbf{r}^{(n+1)T}(\mathbf{r}^{(j+1)} - \mathbf{r}^{(j)}). \tag{5.173}$$

We already found, that $\mathbf{r}^{(n+1)}$ is orthogonal to K_{n+1} , hence to all $\mathbf{r}^{(n)}, \ldots \mathbf{r}^{(0)}$. Therefore we conclude

$$(\mathbf{r}^{(n+1)}, \mathbf{s}^{(j)})_A = 0 \quad \text{for } j+1 \leq n. \tag{5.174}$$

Finally we end up with the following procedure

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda^{(n)}\mathbf{s}^{(n)} \quad \text{with } \lambda^{(n)} = -\frac{\mathbf{s}^{(n)T}\mathbf{r}^{(0)}}{\mathbf{s}^{(n)T}A\mathbf{s}^{(n)}} \tag{5.175}$$

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \lambda^{(n)}A\mathbf{s}^{(n)} \tag{5.176}$$

$$\mathbf{s}^{(n+1)} = \mathbf{r}^{(n+1)} - \beta^{(n)}\mathbf{s}^{(n)} \quad \text{with } \beta^{(n)} = \frac{\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}}. \tag{5.177}$$

This method [25] solves a linear system without storing the matrix A itself. Only the product $A\mathbf{s}$ is needed. In principle the solution is reached after $N = \dim(A)$ steps, but due to rounding errors more steps can be necessary and the method has to be considered as an iterative one.

The expressions for λ and β can be brought to numerically more efficient form. From (5.162) we find

$$\mathbf{s}^{(n)T}\mathbf{r}^{(0)} = \mathbf{s}^{(n)T}\left(\mathbf{r}^{(n)} - \sum_{j=0}^{n-1} \lambda^{(j)} A\mathbf{s}^{(\mathbf{j})}\right). \tag{5.178}$$

But due to A-orthogonality of the search directions

$$\mathbf{s}^{(n)T}\mathbf{r}^{(0)} = \mathbf{s}^{(n)T}\mathbf{r}^{(n)} = \mathbf{r}^{(n)T}\mathbf{r}^{(n)} \tag{5.179}$$

which simplifies

$$\lambda^{(n)} = -\frac{\mathbf{r}^{(n)T}\mathbf{r}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}}. \tag{5.180}$$

Furthermore, from (5.176) and orthogonality of the residual vectors

$$\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)} = \lambda^{(n)}\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)} \tag{5.181}$$

from which

$$\begin{aligned}\beta^{(n)} &= \frac{\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}} = \frac{-\frac{1}{\lambda^{(n)}}\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}} \\ &= -\frac{\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)T}\mathbf{r}^{(n)}}.\end{aligned} \tag{5.182}$$

The conjugate gradients method is not useful for non symmetric systems. It can be applied to the normal equations (11.32)

$$A^T A\mathbf{x} = A^T\mathbf{b} \tag{5.183}$$

which, for a full-rank non singular matrix have the same solution. The condition number (Sect. 5.7), however, is

$$\text{cond}(A^T A) = (\text{cond}A)^2 \tag{5.184}$$

and the problem may be ill conditioned.

5.6.5 Non Symmetric Systems

The general minimum residual method (GMRES) searches directly for the minimum of $||A\mathbf{x}-\mathbf{b}||$ in the Krylov spaces of increasing order $K_n(A, \mathbf{r}^{(0)})$. To avoid problems with linear dependency, first an orthogonal basis

$$K_n(A, \mathbf{r}^{(0)}) = \text{span}(\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_n) \tag{5.185}$$

is constructed with Arnoldi's method, a variant of Gram-Schmidt orthogonalization. Starting from the normalized vector

$$\mathbf{q}_1 = \frac{\mathbf{r}^{(0)}}{|\mathbf{r}^{(0)}|} \tag{5.186}$$

the dimension is iteratively increased by orthogonalizing the vector $A\mathbf{q}_n$ with respect to $K_n(A, \mathbf{r}^{(0)})$[7]

$$\tilde{\mathbf{q}}_{n+1} = A\mathbf{q}_n - \sum_{j=1}^{n} (\mathbf{q}_j, \mathbf{q}_n)_A \mathbf{q}_j = A\mathbf{q}_n - \sum_{j=1}^{n} (\mathbf{q}_j^T A\mathbf{q}_n)\mathbf{q}_j = A\mathbf{q}_n - \sum_{j=1}^{n} h_{jn}\mathbf{q}_j \tag{5.187}$$

and normalizing this vector

$$h_{n+1,n} = |\tilde{\mathbf{q}}_{n+1}| \quad \mathbf{q}_{n+1} = \frac{\tilde{\mathbf{q}}_{n+1}}{h_{n+1,n}}. \tag{5.188}$$

Then

$$A\mathbf{q}_n = h_{n+1,n}\mathbf{q}_{n+1} + \sum_{j=1}^{n} h_{jn}\mathbf{q}_j \tag{5.189}$$

which explicitly reads

$$A\mathbf{q}_1 = h_{21}\mathbf{q}_2 + h_{11}\mathbf{q}_1 = (\mathbf{q}_1, \mathbf{q}_2)\begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} \tag{5.190}$$

$$A\mathbf{q}_2 = h_{32}\mathbf{q}_3 + h_{12}\mathbf{q}_1 + h_{22}\mathbf{q}_2 = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\begin{pmatrix} h_{12} \\ h_{22} \\ h_{32} \end{pmatrix} \tag{5.191}$$

$$\vdots$$

[7] $\mathbf{q}_2$ is a linear combination of $\mathbf{q}_1$ and $A\mathbf{q}_1$, $\mathbf{q}_3$ one of $\mathbf{q}_1$, $\mathbf{q}_2$ and $A\mathbf{q}_2$ hence also of $\mathbf{q}_1$, $A\mathbf{q}_1$, $A^2\mathbf{q}_1$ etc. which proves the validity of (5.185).

$$A\mathbf{q}_n = h_{n+1,n}\mathbf{q}_{n+1} + h_{1n}\mathbf{q}_1 + \ldots h_{nn}\mathbf{q}_n. \tag{5.192}$$

The new basis vectors are orthogonal[8] since

$$\mathbf{q}_1^T\tilde{\mathbf{q}}_2 = \mathbf{q}_1^T\left[A\mathbf{q}_1 - (\mathbf{q}_1^T A\mathbf{q}_1)\mathbf{q}_1\right] = (\mathbf{q}_1^T A\mathbf{q}_1)(1 - |\mathbf{q}_1|^2) = 0$$

and induction shows for $k = 1 \ldots n$

$$\mathbf{q}_k^T\tilde{\mathbf{q}}_{n+1} = \mathbf{q}_k^T A\mathbf{q}_n - \sum_{j=1}^{n}(\mathbf{q}_j^T A\mathbf{q}_n)(\mathbf{q}_k^T\mathbf{q}_j) = \mathbf{q}_k^T A\mathbf{q}_n - \sum_{j=1}^{n}(\mathbf{q}_j^T A\mathbf{q}_n)\delta_{k,j} = 0.$$

We collect the new basis vectors $\mathbf{q}_1 \ldots \mathbf{q}_n$ into a matrix

$$U_n = (\mathbf{q}_1, \ldots \mathbf{q}_n) \tag{5.193}$$

and obtain from (5.190) to (5.192)

$$AU_n = U_{n+1}H \tag{5.194}$$

with the $(n+1) \times n$ upper Hessenberg matrix

$$H = \begin{pmatrix} h_{11} & h_{12} & \ldots & & h_{1n} \\ h_{21} & h_{22} & & & h_{2n} \\ & h_{32} & \ddots & & \vdots \\ & & & \ddots & h_{nn} \\ & & & & h_{n+1,n} \end{pmatrix}. \tag{5.195}$$

Since

$$\mathbf{x}^{(n)} - \mathbf{x}_0 \in K_n(A, \mathbf{r}^{(0)}) \tag{5.196}$$

it can be written as a linear combination of $\mathbf{q}_1 \ldots \mathbf{q}_n$

$$\mathbf{x}^{(n)} - \mathbf{x}_0 = (\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v}. \tag{5.197}$$

The residual becomes

$$\begin{aligned}\mathbf{r}^{(n)} &= A(\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v} + A\mathbf{x}_0 - \mathbf{b} = A(\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v} + \mathbf{r}^{(0)} \\ &= U_{n+1}H\mathbf{v} + |\mathbf{r}^{(0)}|\mathbf{q}_1\end{aligned}$$

[8] If $\mathbf{q}_{n+1} = 0$ the algorithm has to stop and the Krylov space has the full dimension of the matrix.

$$= U_{n+1} \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right] \tag{5.198}$$

hence

$$|\mathbf{r}^{(n)}|^2 = \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right]^T U_{n+1}^T U_{n+1} \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right]. \tag{5.199}$$

But since

$$U_{n+1}^T U_{n+1} = \begin{pmatrix} \mathbf{q_1}^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} (\mathbf{q}_1 \dots \mathbf{q}_n) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \tag{5.200}$$

is a $n \times n$ unit matrix, we have to minimize

$$|\mathbf{r}^{(n)}| = \left| H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right| \tag{5.201}$$

which is a least square problem, since there are $n + 1$ equations for the n unknown components of $\mathbf{v}$. It can be solved efficiently with the help of QR decomposition (11.36)

$$H = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \tag{5.202}$$

with a $(n + 1) \times (n + 1)$ orthogonal matrix Q. The norm of the residual vector becomes

$$|\mathbf{r}^{(n)}| = \left| Q \begin{pmatrix} R \\ 0 \end{pmatrix} \mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} R\mathbf{v} \\ 0 \end{pmatrix} + |\mathbf{r}^{(0)}| Q^T \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right|. \tag{5.203}$$

Substituting

$$|\mathbf{r}^{(0)}|Q^T\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}=\begin{pmatrix}y_1\\\vdots\\y_n\\y_{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{y}\\y_{n+1}\end{pmatrix} \tag{5.204}$$

we have

$$|\mathbf{r}^{(n)}|=\sqrt{y_{n+1}^2+|R\mathbf{v}+\mathbf{y}|^2} \tag{5.205}$$

which is obviously minimized by solving the triangular system

$$R\mathbf{v}+\mathbf{y}=0. \tag{5.206}$$

The GMRES method usually has to be preconditioned (cf.5.112) to improve convergence. Often it is restarted after a small number (e.g.20) of iterations which avoids the necessity to store a large orthogonal basis.

5.7 Matrix Inversion

LU and QR decomposition can be also used to calculate the inverse of a non singular matrix

$$AA^{-1}=1. \tag{5.207}$$

The decomposition is performed once and then the column vectors of A^{-1} are calculated similar to (5.27)

$$L\left(UA^{-1}\right)=1 \tag{5.208}$$

or (5.40)

$$RA^{-1}=Q^{\dagger}. \tag{5.209}$$

Consider now a small variation of the right hand side of (5.2)

$$\mathbf{b}+\Delta\mathbf{b}. \tag{5.210}$$

Instead of

$$A^{-1}\mathbf{b}=\mathbf{x} \tag{5.211}$$

the resulting vector is

$$A^{-1}(\mathbf{b} + \Delta\mathbf{b}) = \mathbf{x} + \Delta\mathbf{x} \tag{5.212}$$

and the deviation can be measured by[9]

$$||\Delta\mathbf{x}|| = ||A^{-1}||\,||\Delta\mathbf{b}|| \tag{5.213}$$

and since

$$||A||\,||\mathbf{x}|| = ||\mathbf{b}|| \tag{5.214}$$

the relative error becomes

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}||} = ||A||\,||A^{-1}||\,\frac{||\Delta\mathbf{b}||}{||\mathbf{b}||}. \tag{5.215}$$

The relative error of $\mathbf{b}$ is multiplied by the condition number for inversion

$$\text{cond}(A) = ||A||\,||A^{-1}||. \tag{5.216}$$

Problem

Problem 5.1 (Comparison of different direct Solvers, Fig. 5.2) In this computer experiment we solve the system of equations

$$A\mathbf{x} = \mathbf{b} \tag{5.217}$$

with several methods:

- Gaussian elimination without pivoting (Sect. 5.1),
- Gaussian elimination with partial pivoting (Sect. 5.1.1),
- QR decomposition with Householder reflections (Sect. 5.2.2),
- QR decomposition with Gram-Schmidt orthogonalization (Sect. 5.2.1),
- QR decomposition with Gram-Schmidt orthogonalization with extra orthogonalization step (5.55).

The right hand side is chosen as

[9] The vector norm used here is not necessarily the Euclidean norm.

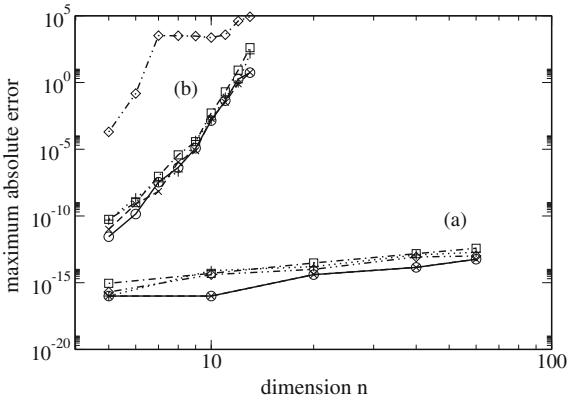

Fig. 5.2 (Comparison of different direct solvers) Gaussian elimination without (*circles*) and with (x) pivoting, QR decomposition with Householder reflections (*squares*), with Gram-Schmidt orthogonalization (*diamonds*) and including extra orthogonalization (+) are compared. The maximum difference $\max_{i=1\ldots n}(|x_i - x_i^{exact}|)$ increases only slightly with the dimension n for the well behaved matrix (5.224,a) but quite dramatically for the ill conditioned Hilbert matrix (5.226,b)

$$\mathbf{b} = A \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} \tag{5.218}$$

hence the exact solution is

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix}. \tag{5.219}$$

Several test matrices can be chosen:

- Gaussian elimination is theoretically unstable but is stable in many practical cases. The instability can be demonstrated with the example [38]

$$A = \begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}. \tag{5.220}$$

  No pivoting takes place in the LU decomposition of this matrix and the entries in the last column double in each step:

$$A^{(1)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ -1 & 1 & & 2 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & 2 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ & & 1 & 4 \\ & & \vdots & \ddots & \vdots \\ & & -1 & -1 & 4 \end{pmatrix} \ldots A^{(n-1)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ & & \ddots & & 4 \\ & & & \ddots & \vdots \\ & & & & 2^{n-1} \end{pmatrix}. \tag{5.221}$$

Since the machine precision is $\epsilon_M = 2^{-53}$ for double precision calculations we have to expect numerical inaccuracy for dimension $n > 53$.

- Especially well conditioned are matrices [46] which are symmetric

$$A_{ij} = A_{ji} \tag{5.222}$$

and also diagonal dominant

$$\sum_{j \neq i} |A_{ij}| < |A_{ii}|. \tag{5.223}$$

We use the matrix

$$A = \begin{pmatrix} n & 1 & \ldots & 1 & 1 \\ 1 & n & \ldots & 1 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & 1 & n & 1 \\ 1 & 1 & 1 & 1 & n \end{pmatrix} \tag{5.224}$$

which can be inverted explicitly by

$$A^{-1} = \begin{pmatrix} a & b & \ldots & b & b \\ b & a & & b & b \\ \vdots & & \ddots & & \\ b & b & b & a & b \\ b & b & b & b & a \end{pmatrix} \quad a = \frac{1}{n - \frac{1}{2}}, b = -\frac{1}{2n^2 - 3n + 1} \tag{5.225}$$

and has a condition number[10] which is proportional to the dimension n (Fig. 5.3).

- The Hilbert matrix [47, 48]

[10] Using the Frobenius norm $||A|| = \sqrt{\sum_{ij} A_{ij}^2}$.

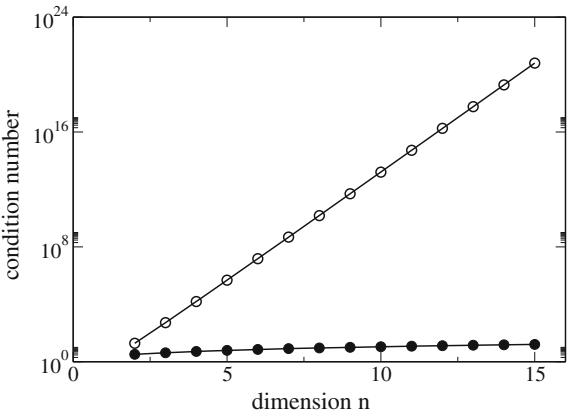

Fig. 5.3 (Condition numbers) The condition number $cond(A)$ increases only linearly with the dimension n for the well behaved matrix (5.224, *full circles*) but exponentially for the ill conditioned Hilbert matrix (5.226, *open circles*)

$$A = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & & \frac{1}{n+2} \\ \vdots & & & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{pmatrix} \tag{5.226}$$

is especially ill conditioned [49] even for moderate dimension. It is positive definite and therefore the inverse matrix exists and even can be written down explicitly [50]. Its column vectors are very close to linearly dependent and the condition number grows exponentially with its dimension (Fig. 5.3). Numerical errors are large for all methods compared (Fig. 5.2).

- random matrices

$$A_{ij} = \xi \in [-1, 1]. \tag{5.227}$$