Chapter 8
Preconditioning

8.1 The General Approach

Remark 8.1. Motivation and idea. It was seen in Chapter 7 that the number of iterations might depend on the condition number of the matrix. In order to reduce the number of iterations, one wants to replace the original linear system of equations (1.1) by an equivalent system whose system matrix has a smaller condition number. This strategy is called preconditioning.

The main idea of preconditioning consists in applying the iterative method to the equivalent system

$$M^{-1}A\underline{x} = M^{-1}\underline{b} \quad \text{(preconditioning from left)}$$

or

$$AM^{-1}\underline{y} = \underline{b},\ \underline{x} = M^{-1}\underline{y} \quad \text{(preconditioning from right).}$$

The non-singular matrix M is called preconditioner. This matrix should satisfy two requirements:

- The convergence of the iterative method for the system with the matrix $M^{-1}A$ or AM^{-1}, respectively, should be faster than for the original system with the matrix A. That means, M^{-1} should be a good approximation to A^{-1}.
- Linear systems with the matrix M should be solvable with low costs.

In general, one has to find a compromise between these two requirements.

Usually, left and right preconditioning lead to different methods which might behave sometimes quite differently. □

Remark 8.2. Some preconditioners. An easy way to construct preconditioners consists in starting with the decomposition $A = D + L + U$, see Section 3.2, and using parts of this decomposition which are easily invertible:

- $M = D$, diagonal preconditioner, Jacobi preconditioner,
- $M = D + L$, forward Gauss–Seidel preconditioner,

- $M = D + U$, backward Gauss–Seidel preconditioner,
- $M = (D + L)\, D^{-1}\, (D + U)$, symmetric Gauss–Seidel preconditioner.

Damped versions of the classical iterative schemes can be also used. A more advanced preconditioner will be presented in Section 8.3.

Note that M or M^{-1} do not need to be known explicitly. They can also stand for some numerical (iterative) method for solving linear systems of equations. Then, M^{-1} means that this method should be applied to a vector. □

Remark 8.3. Change in algorithms for general matrices if the preconditioner is applied. In algorithms for general matrices A, preconditioning from left consists in replacing A by $M^{-1}A$ and $\underline{r}^{(k)}$ by $M^{-1}\underline{r}^{(k)}$ in the algorithms. Then, e.g., GMRES computes the iterate

$$\underline{x}^{(k)} \in \underline{x}^{(0)} + K_k\left(M^{-1}\underline{r}^{(0)}, M^{-1}A\right)$$

such that $\left\|M^{-1}\underline{r}^{(k)}\right\|_2$ becomes minimal. □

8.2 Symmetric Matrices

Remark 8.4. A difficulty and its solution. A problem occurs if the matrix A is symmetric and the iterative method wants to exploit this property, e.g., using short recurrences, since in general neither $M^{-1}A$ nor AM^{-1} are symmetric. This problem can be solved by constructing the orthonormal basis of the Krylov subspace with respect to an appropriate inner product.

Let H be a Hilbert[1] space with the inner product $(\cdot,\cdot)_H$ and $\mathcal{L} : H \to H$ be a linear map. This map is called self-adjoint with respect to $(\cdot,\cdot)_H$ if

$$(\mathcal{L}v, w)_H = (v, \mathcal{L}w)_H \quad \forall\, v, w \in H.$$

In the case $H = \mathbb{R}^n$ equipped with the standard Cartesian basis and the Euclidean inner product $(\cdot,\cdot)$, a linear map, which is represented by a matrix A, is self-adjoint if

$$(A\underline{x}, \underline{y}) = (\underline{x}, A\underline{y}) \quad \forall\, \underline{x}, \underline{y} \in \mathbb{R}^n.$$

This condition is equivalent to A being symmetric.

If the preconditioner M is symmetric and positive definite, then

$$(\underline{x}, \underline{y})_M = (\underline{x}, M\underline{y}), \quad \forall\, \underline{x}, \underline{y} \in \mathbb{R}^n$$

[1] David Hilbert (1862 – 1943)

defines an inner product in $\mathbb{R}^n$. The induced norm is given by $\|\underline{x}\|_M = (\underline{x}, \underline{x})_M^{1/2}$.

Consider for the remainder of this section preconditioning from left. The matrix $M^{-1}A$ is self-adjoint with respect to this inner product since

$$(M^{-1}A\underline{x}, \underline{y})_M = (M^{-1}A\underline{x}, M\underline{y}) = (A\underline{x}, \underline{y}) = (\underline{x}, A\underline{y}) = (\underline{x}, M^{-1}A\underline{y})_M$$

for all $\underline{x}, \underline{y} \in \mathbb{R}^n$.

Now, one can generate an orthonormal basis with respect to the inner product $(\cdot, \cdot)_M$ of $K_k\left(M^{-1}\underline{r}^{(0)}, M^{-1}A\right)$ by an appropriate modification of the Lanczos algorithm. □

Algorithm 8.5. Preconditioned Lanczos algorithm for symmetric matrices. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, a symmetric positive definite matrix $M \in \mathbb{R}^{n\times n}$, and $\underline{r}^{(0)} \in \mathbb{R}^n$.

$\underline{z} = M^{-1}\underline{r}^{(0)}$
$\underline{q}_1 = \dfrac{\underline{z}}{(\underline{r}^{(0)}, \underline{z})^{1/2}}$
$\beta_0 = 0$
$\underline{q}_0 = \underline{0}$
`for` $j = 1 : k$
  $\underline{s} = A\underline{q}_j$
  $\underline{z} = M^{-1}\underline{s}$
  $\alpha_j = \left(\underline{s}, \underline{q}_j\right)$
  $\underline{z} = \underline{z} - \alpha_j\underline{q}_j - \beta_{j-1}\underline{q}_{j-1}$
  $\beta_j = (\underline{s}, \underline{z})^{1/2}$
  $\underline{q}_{j+1} = \underline{z}/\beta_j$
`endfor`

□

Remark 8.6. On the preconditioned Lanczos algorithm for symmetric matrices.

- The vector $\underline{z}$ is computed by solving $M\underline{z} = \underline{s}$.
- The matrix form of the preconditioned Lanczos algorithm is

$$M^{-1}AQ_k = Q_{k+1}H_k \text{ with } H_k = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ & & & & & \\ 0 & 0 & 0 & \cdots & \alpha_{k-1} & \beta_{k-1} \\ 0 & 0 & 0 & \cdots & \beta_{k-1} & \alpha_k \\ 0 & 0 & 0 & \cdots & 0 & \beta_k \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{8.1}$$

The columns of Q_{k+1} are orthogonal with respect to $(\cdot,\cdot)_M$

$$Q_{k+1}^T M Q_{k+1} = I \in \mathbb{R}^{(k+1)\times(k+1)}. \tag{8.2}$$

□

Remark 8.7. On the orthogonality condition for the preconditioned conjugate gradient method. The preconditioned conjugate gradient method (PCG) is one of the most important algorithms for solving linear systems of equations with symmetric and positive definite matrix. Besides the preconditioned Lanczos algorithm, one needs to implement the orthogonality condition of the residual with respect to $(\cdot,\cdot)_M$ with a short recurrence. Concretely, one has to construct

$$\underline{x}^{(k)} \in \underline{x}^{(0)} + K_k\left(M^{-1}\underline{r}^{(0)}, M^{-1}A\right) \tag{8.3}$$

such that $M^{-1}\underline{r}^{(k)} = M^{-1}\left(\underline{b} - A\underline{x}^{(k)}\right)$ is orthogonal to $K_k\left(M^{-1}\underline{r}^{(0)}, M^{-1}A\right)$ with respect to $(\cdot,\cdot)_M$

$$M^{-1}\underline{r}^{(k)} \perp_M K_k\left(M^{-1}\underline{r}^{(0)}, M^{-1}A\right) \quad \Longleftrightarrow \quad M^{-1}\underline{r}^{(k)} \perp_M Q_k.$$

Using the definition of $\underline{q}_1$, see Algorithm 8.5, lines 1 and 2, it is by (8.2)

$$\begin{aligned}\left(\underline{q}_k, \underline{r}^{(0)}\right) &= \left(\underline{q}_k, MM^{-1}\underline{r}^{(0)}\right) = \left\|M^{-1}\underline{r}^{(0)}\right\|_M \left(\underline{q}_k, M\underline{q}_1\right)\\ &= \left\|M^{-1}\underline{r}^{(0)}\right\|_M \delta_{1k},\end{aligned} \tag{8.4}$$

where δ_{ij} is the Kronecker symbol. Since by construction $\underline{x}^{(k)} = \underline{x}^{(0)} + Q_k\underline{y}_k$, one obtains with $\beta = \left\|M^{-1}\underline{r}^{(0)}\right\|_M$, (8.4), (8.1), and (8.2) the condition

$$\begin{aligned}\underline{0} &= \left(Q_k, M^{-1}\underline{r}^{(k)}\right)_M = \left(Q_k, MM^{-1}\underline{r}^{(k)}\right)\\ &= \left(Q_k, \underline{r}^{(k)}\right) = \left(Q_k, \underline{r}^{(0)} - AQ_k\underline{y}_k\right)\\ &= \beta\boldsymbol{e}_1 - Q_k^T A Q_k \underline{y}_k = \beta\boldsymbol{e}_1 - Q_k^T M Q_{k+1} H_k \underline{y}_k\\ &= \beta\boldsymbol{e}_1 - Q_k^T M\left[Q_k \underline{q}_{k+1}\right] H_k\underline{y}_k = \beta\boldsymbol{e}_1 - \left[I\underline{0}\right] H_k\underline{y}_k\\ &= \beta\boldsymbol{e}_1 - \tilde{H}_k\underline{y}_k,\end{aligned}$$

where $\tilde{H}_k \in \mathbb{R}^{k\times k}$ is the matrix consisting of the first k rows of H_k. Hence, $\underline{y}_k$ can be computed from $\tilde{H}_k$, from what follows that $\underline{x}^{(k)}$ can be computed with a short recurrence. Analogous calculations as in Section 6.2 lead finally to PCG. □

Algorithm 8.8. Preconditioned conjugate gradient (PCG). Given a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$, a right-hand side $\underline{b} \in \mathbb{R}^n$, an