Chapter 9

Preconditioning

9.1 The General Approach

Remark 9.1 *Idea.* The main idea of preconditioning consists in applying the iterative method for solving (1.1) not to the original system but to an equivalent system

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

or

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

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

- The convergence of the iterative method for the systems with the matrix $M^{-1}A$ or AM^{-1}, respectively, should be faster than 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. □

Remark 9.2 *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.

A more advanced preconditioner will be presented in Section 9.3.

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

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

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

9.2 Symmetric Matrices

Remark 9.4 *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

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

This is equivalent to A being symmetric. If the preconditioner M is symmetric and positive definite, then

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

defines an inner product in $\mathbb{R}^n$. The matrix $M^{-1}A$ is self-adjoint with respect to this inner product since

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

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. The induced norm is given by $\|\mathbf{x}\|_M = (\mathbf{x}, \mathbf{x})_M^{1/2}$.

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

Algorithm 9.5 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 $\mathbf{r}^{(0)} \in \mathbb{R}^n$.

1. $\mathbf{z} = M^{-1}\mathbf{r}^{(0)}$
2. $\mathbf{q}_1 = \dfrac{\mathbf{z}}{(\mathbf{r}^{(0)}, \mathbf{z})^{1/2}}$
3. $\beta_0 = 0$
4. $\mathbf{q}_0 = \mathbf{0}$
5. `for` $j = 1 : k$
6. $\quad \mathbf{s} = A\mathbf{q}_j$
7. $\quad \mathbf{z} = M^{-1}\mathbf{s}$
8. $\quad \alpha_j = (\mathbf{s}, \mathbf{q}_j)$
9. $\quad \mathbf{z} = \mathbf{z} - \alpha_j\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$
10. $\quad \beta_j = (\mathbf{s}, \mathbf{z})^{1/2}$
11. $\quad \mathbf{q}_{j+1} = \mathbf{z}/\beta_j$
12. `endfor`

□

Remark 9.6 *On the preconditioned Lanczos algorithm for symmetric matrices.* The vector $\mathbf{z}$ is computed by solving $M\mathbf{z} = \mathbf{s}$.

[1] David Hilbert (1862 – 1943)

The matrix form of the preconditioned Lanczos algorithm is

$$M^{-1}AQ_k = Q_{k+1}H_k \text{ with } H_k = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ & & & & & \\ 0 & 0 & 0 & \cdots & \alpha_{k-1} & \beta_{k-1} \\ 0 & 0 & 0 & \cdots & \beta_{k-1} & \alpha_k \\ 0 & 0 & 0 & \cdots & 0 & \beta_k \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{9.1}$$

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

$$Q_{k+1}^T M Q_{k+1} = I \in \mathbb{R}^{(k+1)\times(k+1)}. \tag{9.2}$$

□

Remark 9.7 *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

$$\mathbf{x}^{(k)} \in \mathbf{x}^{(0)} + K_k\left(M^{-1}\mathbf{r}^{(0)}, M^{-1}A\right) \tag{9.3}$$

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

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

Since by construction $\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + Q_k\mathbf{y}_k$, one obtains with (9.1) and (9.2) the condition

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

where $\tilde{H}_k \in \mathbb{R}^{k\times k}$ is the matrix consisting of the first k rows of H_k. Using line 2, the constant β is given by

$$\begin{aligned} \beta &= \left(\mathbf{q}_1, \mathbf{r}^{(0)}\right) = \frac{\left(M^{-1}\mathbf{r}^{(0)}, \mathbf{r}^{(0)}\right)}{\left(\mathbf{r}^{(0)}, M^{-1}\mathbf{r}^{(0)}\right)^{1/2}} = \left(\mathbf{r}^{(0)}, M^{-1}\mathbf{r}^{(0)}\right)^{1/2} \\ &= \left(M^{-1}\mathbf{r}^{(0)}, MM^{-1}\mathbf{r}^{(0)}\right)^{1/2} = \left\|M^{-1}\mathbf{r}^{(0)}\right\|_M. \end{aligned}$$

Analogous calculations as in Section 6.2 lead finally to PCG. □

Algorithm 9.8 Preconditioned conjugate gradient (PCG). Given a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$, a right hand side $\mathbf{b} \in \mathbb{R}^n$, an initial iterate $\mathbf{x}^{(0)} \in \mathbb{R}^n$, a tolerance $\varepsilon > 0$ and a symmetric positive definite preconditioner $M \in \mathbb{R}^{n\times n}$.

1. $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$

```
2.  solve  Mz_0 = r^(0)
3.  p_1 = z_0
4.  k = 0
5.  while (z_k, r^(k))^(1/2) > ε
6.     k = k + 1
7.     s = A p_k
8.     ν_k = (z_{k-1}, r^(k-1)) / (p_k, s)
9.     x^(k) = x^(k-1) + ν_k p_k
10.    r^(k) = r^(k-1) − ν_k s
11.    solve  M z_k = r^(k)
12.    μ_{k+1} = (z_k, r^(k)) / (z_{k-1}, r^(k-1))
13.    p_{k+1} = z_k + μ_{k+1} p_k
14. endwhile
```

□

Remark 9.9 *On PCG.* There exists also other ways to implement PCG. Compared with CG, one has to solve the linear system with the matrix M and one has to store one additional vector (five vectors altogether). □

Remark 9.10 *Preconditioners for PCG.* If PCG should be applied for solving $A\mathbf{x} = \mathbf{b}$ with A being symmetric and positive definite, also M has to be symmetric and positive definite. Among the preconditioners given in Remark 9.2 the Jacobi and the symmetric Gauss-Seidel preconditioner possess this property. For the Jacobi preconditioner, it follows from $\mathbf{x}^T A\mathbf{x} > 0$ that $\mathbf{x}^T D\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$. For the symmetric Gauss-Seidel preconditioner, one has

$$\mathbf{x}^T (D+L) D^{-1} (D+U)\mathbf{x} = \mathbf{x}^T (D+U)^T D^{-1} \underbrace{(D+U)\mathbf{x}}_{\mathbf{y}} = \mathbf{y}^T D^{-1}\mathbf{y} > 0.$$

In addition, multigrid methods and incomplete Cholesky[2] factorizations, see Remark 9.25, can be also used as preconditioners. □

Theorem 9.11 Estimate of the rate of convergence for s.p.d. matrices. *Let $A, M \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. Then, the k-th iterate of PCG fulfills*

$$\frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_A}{\left\|\mathbf{x} - \mathbf{x}^{(0)}\right\|_A} \leq 2 \left(\frac{\sqrt{\kappa_2\left(M^{-1/2} A M^{-1/2}\right)} - 1}{\sqrt{\kappa_2\left(M^{-1/2} A M^{-1/2}\right)} + 1} \right)^k .$$

Proof: The proof follows the lines of the proof of Theorem 7.6. The iterate of PCG minimizes the error in the norm $\|\cdot\|_A$ within all vectors of form (9.3). ■

Remark 9.12 *On the spectral condition number for the preconditioned system.* The matrices $M^{-1/2} A M^{-1/2}$ and $M^{-1}A$ are similar, i.e., there is a non-singular matrix S such that $M^{-1}A = S M^{-1/2} A M^{-1/2} S^{-1}$. Obviously, it is $S = M^{-1/2}$. Since $M^{-1/2} A M^{-1/2}$ is symmetric, cf. Remark 2.12, and similar matrices have the same eigenvalues, it follows that

$$\kappa_2\left(M^{-1/2} A M^{-1/2}\right) = \frac{\lambda_{\max}(M^{-1/2} A M^{-1/2})}{\lambda_{\min}(M^{-1/2} A M^{-1/2})} = \frac{\lambda_{\max}(M^{-1}A)}{\lambda_{\min}(M^{-1}A)}.$$

That means, if M is a good preconditioner, i.e. the ratio of the largest and smallest eigenvalue of $M^{-1}A$ is small, then the number of iterations are reduced by using PCG instead of CG. □

[2] André–Louis Cholesky (1875 – 1918)

9.3 Incomplete LU Decompositions

Remark 9.13 *Idea.* One drawback of the application of direct solvers for linear systems of equations with a sparse matrix is the additional fill-in that occurs if a decomposition of the matrix, like the LU decomposition, is computed. A main part of modern direct solvers for sparse linear systems, like `UMFPACK` Davis (2004) which is the package behind the backslash command in MATLAB, is a reordering of the unknowns such that the fill-in is reduced. In the context of preconditioning, the LU decomposition can be modified such that it respects the sparsity pattern or zero pattern of the matrix. □

Algorithm 9.14 Incomplete LU factorization (ILU). Given a matrix $A \in \mathbb{R}^{n\times n}$ and a zero pattern $P = \{(i,j) \ : \ i \neq j, 1 \leq i,j \leq n\}$.

```
1. for k = 1 : n − 1
2.    for i = k + 1 : n
3.      if (i, k) ∉ P
4.        a_ik = a_ik / a_kk
5.        for j = k + 1 : n
6.          if (i, j) ∉ P
7.            a_ij = a_ij − a_ik a_kj
8.          endif
9.        endfor
10.     endif
11.   endfor
12. endfor
```

□

Remark 9.15 *To Algorithm 9.14.*

- For $P = \emptyset$, Algorithm 9.14 is just a standard LU factorization of the matrix A. Then, Algorithm 9.14 replaces A by the factors L and U

$$u_{ij}, \text{ for } 1 \leq i \leq j \leq n, \text{ upper triangular matrix,}$$
$$l_{ij}, \text{ for } 1 \leq j < i \leq n, \text{ lower triangular matrix,}$$

  where the diagonal entries of L are all 1 and they are not stored. It holds $A = LU$.
- If $P \neq \emptyset$, then it is

$$A = LU - N \text{ with } 0 \neq N \in \mathbb{R}^{n\times n}.$$

  In this case, one needs extra memory to store the factors L and U.
- Usually, one calls the algorithm ILU if the zero pattern P is chosen to be the zero pattern of A. Sometimes, this algorithm is called also ILU(0).
- For using ILU as preconditioner, it is essential that the diagonal entries do not belong to P.

Now, properties of the ILU factorization of a matrix A are of interest. □

Definition 9.16 Non-negative matrix. A matrix $B \in \mathbb{R}^{n\times n}$ is called non-negative, if

$$b_{ij} \geq 0, \quad \forall\, i,j = 1,\ldots,n.$$

□

Theorem 9.17 Perron[3]–Frobenius theorem: Eigenvalue and eigenvector of a non-negative matrix. *Let $B \in \mathbb{R}^{n\times n}$ be a non-negative matrix. Then, the spectral radius $\rho(B) \in \mathbb{R}$ is an eigenvalue of B and there is a corresponding eigenvector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v} \neq \mathbf{0}$, with non-negative entries.*

Proof: See literature. ■

Lemma 9.18 Spectral radius of the difference of non-negative matrices. *Let $B \in \mathbb{R}^{n\times n}$ be a non-negative matrix and let $C \in \mathbb{R}^{n\times n}$ such that $C - B$ is a non-negative matrix, i.e., $b_{ij} \leq c_{ij}$ for $i, j = 1, \ldots, n$. Then $\rho(B) \leq \rho(C)$.*

Proof: See literature, e.g. (Saad, 2003, Section 1.10). ■

Definition 9.19 M-matrix. A matrix $A \in \mathbb{R}^{n\times n}$ is called M-matrix if it satisfies the following conditions

1. $a_{ij} \leq 0$ for $i, j = 1, \ldots, n$, $i \neq j$,
2. A is non-singular and A^{-1} is non-negative.

□

Lemma 9.20 Diagonal entries of an M-matrix. *If $A \in \mathbb{R}^{n\times n}$ is an M-matrix, then $a_{ii} > 0$, $i = 1, \ldots, n$.*

Proof: *exercise.* ■

Remark 9.21 *M-matrices.* M-matrices arise in some discretizations of partial differential equations. It is an important class of matrices with favorable mathematical properties. □

Lemma 9.22 M-matrices and Gauss algorithm. *Let $A \in \mathbb{R}^{n\times n}$ be an M-matrix and let $A^{(1)} \in \mathbb{R}^{n\times n}$ be the matrix that is obtained as result of the first step of the Gauss algorithm. Then, $A^{(1)}$ is also an M-matrix.*

Proof: The first step of the Gauss algorithm can be written in the following form

$$A^{(1)} = L^{(1)}A = \begin{pmatrix} 1 & & & & \\ -a_{21}/a_{11} & 1 & & & \\ -a_{31}/a_{11} & 0 & 1 & & \\ \vdots & \vdots & & \ddots & \\ -a_{n1}/a_{11} & 0 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \tag{9.4}$$

First property. For the off diagonal elements of $A^{(1)}$, it is

$$\begin{aligned} a_{1j}^{(1)} &= a_{1j} \leq 0, \quad j > 1, \\ a_{i1}^{(1)} &= 0, \quad i > 1, \\ a_{ij}^{(1)} &= \underbrace{a_{ij}}_{\leq 0} - \underbrace{\frac{a_{i1}}{a_{11}}}_{\leq 0} \underbrace{a_{1j}}_{\leq 0} \leq 0, \quad i, j > 1,\ i \neq j, \end{aligned}$$

since in the last line the first term is not positive and the second term is non-negative.

Second property. The matrix A is non-singular because it is an M-matrix and the matrix $L^{(1)}$ is also non-singular since $\det\left(L^{(1)}\right) = 1$. Thus, $A^{(1)}$ is non-singular since it is the product of two non-singular matrices.

[3] Oskar Perron (1880 - 1975)

Finally, one has to show that $\left(A^{(1)}\right)^{-1}$ is non-negative. To this end, consider the product of $\left(A^{(1)}\right)^{-1}$ with the Cartesian unit vectors $\mathbf{e}_j$, $j = 1, \ldots, n$. Using (9.4), one obtains

$$\left(A^{(1)}\right)^{-1} \mathbf{e}_1 = A^{-1} \left(L^{(1)}\right)^{-1} \mathbf{e}_1 = A^{-1}\mathbf{d}_1.$$

By the adjoint method, one has

$$\left(L^{(1)}\right)^{-1} = \frac{1}{\det\left((L^{(1)})\right)} \mathrm{adj}\left(\left(L^{(1)}\right)\right),$$

where the adjoint matrix is the transpose of the cofactor matrix. It can be shown that $\mathrm{adj}\left(\left(L^{(1)}\right)\right)$ has the same form as $L^{(1)}$ since all minors for zero entries of $L^{(1)}$ have a zero row or column and thus they vanish. A direct calculation of the first column of $\mathrm{adj}\left(\left(L^{(1)}\right)\right)$ shows that this column is non-negative. It follows that $\mathbf{d}_1$ is also non-negative *details left for exercise*. Analogously, one obtains

$$\left(A^{(1)}\right)^{-1} \mathbf{e}_j = A^{-1} \left(L^{(1)}\right)^{-1} \mathbf{e}_j = A^{-1}\mathbf{d}_j, \quad j = 2, \ldots, n,$$

with non-negative vectors $\mathbf{d}_j$. Since A is an M-matrix, all entries of the vectors $A^{-1}\mathbf{d}_j$, $j = 1, \ldots, n$, are non-negative, because all entries of A^{-1} are non-negative and all entries of $\mathbf{d}_j$ are non-negative. It follows that also the vectors $\left(A^{(1)}\right)^{-1} \mathbf{e}_j$, $j = 1, \ldots, n$, are non-negative. But these vectors are just the columns of $\left(A^{(1)}\right)^{-1}$. That means $\left(A^{(1)}\right)^{-1}$ is a non-negative matrix.

Altogether, it follows that $A^{(1)}$ is an M-matrix. ■

Lemma 9.23 Comparison criterion. *Let $A \in \mathbb{R}^{n\times n}$ be an M-matrix and let $B \in \mathbb{R}^{n\times n}$ be a matrix with*

$$\begin{array}{rcll} a_{ij} & \leq & b_{ij} & \forall\ i, j = 1, \ldots, n, \\ b_{ij} & \leq & 0 & \forall\ i, j = 1, \ldots, n, i \neq j. \end{array}$$

Then, B is also an M-matrix.

Proof: Decomposing A into the sum of strictly lower triangular part, diagonal matrix, and strictly upper triangular part

$$A = L_A + D_A + U_A,$$

one obtains

$$\begin{array}{rcl} -A^{-1}\left(L_A + U_A\right) & = & A^{-1}D_A\left(-D_A^{-1}\left(L_A + U_A\right)\right) \\ & = & \left(D_A^{-1}A\right)^{-1}\left(-D_A^{-1}\left(L_A + U_A\right)\right) \\ & = & \left(I + D_A^{-1}\left(L_A + U_A\right)\right)^{-1}\left(-D_A^{-1}\left(L_A + U_A\right)\right). \end{array}$$

Let μ be an eigenvalue of $-A^{-1}\left(L_A + U_A\right)$ and let $\mathbf{v}$ be a corresponding eigenvector, then it follows that

$$\begin{array}{rcll} \mu\mathbf{v} & = & \left(I + D_A^{-1}\left(L_A + U_A\right)\right)^{-1}\left(-D_A^{-1}\left(L_A + U_A\right)\right)\mathbf{v} & \Longleftrightarrow \\ \mu\left(I + D_A^{-1}\left(L_A + U_A\right)\right)\mathbf{v} & = & \left(-D_A^{-1}\left(L_A + U_A\right)\right)\mathbf{v} & \Longleftrightarrow \\ \mu\mathbf{v} & = & (1+\mu)\left(-D_A^{-1}\left(L_A + U_A\right)\right)\mathbf{v}. & \end{array}$$

Hence, the eigenvalues of $-D_A^{-1}\left(L_A + U_A\right)$ are

$$\lambda = \frac{\mu}{1+\mu}. \tag{9.5}$$

It follows that

$$\mu = \frac{\lambda}{1-\lambda}.$$

Since A^{-1} and D_A^{-1} are non-negative by Definition 9.19 and Lemma 9.20, and $-(L_A + U_A)$ is non-negative by Definition 9.19, too, one obtains that $-A^{-1}(L_A + U_A)$ and $-D_A^{-1}(L_A + U_A)$ are non-negative matrices. From Theorem 9.17 it follows that the spectral radii

$$\rho\left(-A^{-1}(L_A + U_A)\right) \text{ and } \rho\left(-D_A^{-1}(L_A + U_A)\right)$$

are eigenvalues of the respective matrices. Both eigenvalues are positive real numbers and the largest positive real eigenvalues of their matrices by the definition of the spectral radius, Definition 2.6. Since the correspondence (9.5) is strictly monoton for positive real numbers, it links in particular the largest eigenvalues and it follows that

$$\rho\left(-D_A^{-1}(L_A + U_A)\right) = \frac{\rho\left(-A^{-1}(L_A + U_A)\right)}{1 + \rho\left(-A^{-1}(L_A + U_A)\right)} < 1.$$

Now, it will be shown that B^{-1} is non-negative. By the assumption on the entries of the matrix B, one gets

$$\begin{aligned}
\underbrace{-D_A^{-1}(L_A + U_A)}_{\geq 0} + \underbrace{D_B^{-1}}_{>0,\leq D_A^{-1}} \underbrace{(L_B + U_B)}_{\leq 0} &\geq -D_A^{-1}(L_A + U_A) + D_A^{-1}(L_B + U_B) \\
&= D_A^{-1}(\underbrace{L_B - L_A}_{\geq 0} + \underbrace{U_B - U_A}_{\geq 0}) \geq 0.
\end{aligned}$$

It follows from Lemma 9.18 that

$$\rho\left(-D_B^{-1}(L_B + U_B)\right) \leq \rho\left(-D_A^{-1}(L_A + U_A)\right) < 1.$$

Since the spectral radius is less than 1, one obtains the following expansion with a convergent geometric series

$$\begin{aligned}
B^{-1} &= (L_B + D_B + U_B)^{-1} = \left(D_B\left(I + D_B^{-1}(L_B + U_B)\right)\right)^{-1} \\
&= \left(I + D_B^{-1}(L_B + U_B)\right)^{-1} D_B^{-1} \\
&= \sum_{i=0}^{\infty} \underbrace{\left(-D_B^{-1}(L_B + U_B)\right)^i}_{\geq 0} \underbrace{D_B^{-1}}_{>0}.
\end{aligned}$$

Hence, every term in the sum is a product of non-negative matrices such that the sum is also a non-negative matrix. ■

Theorem 9.24 Properties of an ILU decomposition of an M-matrix. *Let $A \in \mathbb{R}^{n\times n}$ be an M-matrix and let P a given zero pattern. Then, there is a lower triangular matrix L with ones on the main diagonal and an upper triangular matrix U such that $A = LU - N$ with*

$$\begin{aligned}
l_{ij} &= 0 \quad \text{for } (i,j) \in P, \\
u_{ij} &= 0 \quad \text{for } (i,j) \in P, \\
n_{ij} &\neq 0 \quad \text{for } (i,j) \notin P.
\end{aligned}$$

The matrices L^{-1} and N are non-negative.

Proof: The principal ILU decomposition is computed analogously to the Gauss elimination

1. $A^{(0)} = A$
2. `for` $k = 1 : n-1$
3. $\quad \tilde{A}^{(k)} = A^{(k-1)} + N^{(k)}$
4. $\quad A^{(k)} = L^{(k)}\tilde{A}^{(k)}$
5. `endfor`

In the k-th step of line 3, zero entries are generated in $\tilde{A}^{(k)}$ in the zero pattern of the k-th row and the k-th column, i.e. for $(k,j) \in P$ and $(i,k) \in P$. In line 4, the elimination step is applied to $\tilde{A}^{(k)}$. The elimination matrix has the form

$$L^{(k)} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\tilde{a}^{(k)}_{k+1,k}/a^{(k)}_{k,k} & 1 & \\ & & \vdots & & \ddots \\ & & -\tilde{a}^{(k)}_{n,k}/a^{(k)}_{k,k} & & \dots \quad 1 \end{pmatrix}.$$

The matrix $A^{(0)} = A$ is an M-matrix. Hence, the off-diagonal entries of $A^{(0)}$ are non-positive from what follows, see line 3, that $N^{(1)}$ is non-negative. The matrices $A^{(0)}$ and $\tilde{A}^{(1)}$ satisfy the assumptions of Lemma 9.23. It follows that $\tilde{A}^{(1)}$ is an M-matrix. All off-diagonal entries of $\tilde{A}^{(1)}$ are non-positive and $\tilde{a}_{11} > 0$. Thus, $L^{(1)}$ is non-negative. Analogously as in the proof of Lemma 9.22 one can show now that $A^{(1)}$ is an M-matrix.

It can be shown now by induction that $A^{(k)}$ and $\tilde{A}^{(k)}$ are M-matrices for $k = 1, \dots, n-1$, and $N^{(k)}$ and $L^{(k)}$ are non-negative$k = 1, \dots, n-1$.

By the form of the elimination matrix, the first k rows of $A^{(k)}$ and $\tilde{A}^{(k)}$ are the same. In particular, there are only non-zero entries in the first k rows of $A^{(k)}$ at entries which do not belong to the zero pattern. It follows that in the next step, the first k rows of $N^{(k+1)}$ are zero. From this form of the matrix $N^{(k+1)}$ and from the form of the elimination matrices, one obtains for $i < k+1$

$$L^{(i)} N^{(k+1)} = N^{(k+1)}.$$

With this relation, one gets

$$\begin{aligned} A^{(n-1)} &= L^{(n-1)} \tilde{A}^{(n-1)} \\ &= L^{(n-1)} \left(A^{(n-2)} + N^{(n-1)} \right) \\ &= \cdots \\ &= \left(\prod_{j=1}^{n-1} L^{(n-j)} \right) A^{(0)} + \sum_{i=1}^{n-1} \left(\prod_{j=1}^{n-i} L^{(n-j)} \right) N^{(i)} \\ &= \left(\prod_{j=1}^{n-1} L^{(n-j)} \right) A + \sum_{i=1}^{n-1} \left(\prod_{j=1}^{n-1} L^{(n-j)} \right) N^{(i)} \\ &= \left(\prod_{j=1}^{n-1} L^{(n-j)} \right) \left(A + \sum_{i=1}^{n-1} N^{(i)} \right) \\ &=: L^{-1} (A + N). \end{aligned}$$

Denoting $U = A^{(n-1)}$ yields $A = LU - N$. The matrix N is a sum of non-negative matrices such that it is non-negative. Similarly, the matrix L^{-1} is a product of non-negative matrices and hence it is non-negative, too. ∎

Remark 9.25 *ILU.*

- For a given zero pattern P, the ILU is uniqueliy determined.
- The application of ILU as preconditioner requires the solution of two sparse linear systems of equations with triangular matrices:
  1. solve the lower triangular system $L\mathbf{w} = \mathbf{r}$,
  2. solve the upper triangular system $U\mathbf{z} = \mathbf{w}$.
- The main costs of unpreconditioned iterative methods are the multiplication of the sparse matrix with a vector. If the zero pattern is appropriately given, then the costs for applying the ILU preconditioner are proportional to the costs of the matrix-vector multiplication. Very often, one takes the pattern of A, i.e.,

non-zero entries in L and U are allowed only for pairs of indicies for which A has a non-zero entry.

- If A is symmetric and positive definite, then one obtains (if the non-zero pattern is chosen to be symmetric) an incomplete Cholesky decomposition. The precondition matrix $M = LL^T$ is also symmetric and positive definite and it can be applied in the PCG algorithm.

□