Chapter 6

Semi-explicit prefactorization with implicit backward sweep

The methods discussed in chapter 5 are called explicit prefactorization methods because both forward and backward sweeps have an explicit form. This chapter describes a class of semi-explicit prefactorization algorithms with implicit backward sweep. These efficient iterative algorithms, introduced recently by the author under the name *modified SLOR algorithms* (or *MSLOR*) [91], are derived from the standard 1-line version of the SOR method, using a preliminary elimination.

Section 6.1 presents the matrix formulation of these algorithms; it also gives theoretical results in the form of a comparison theorem, proved using the nonnegative splitting theory of chapter 2. The implementation of MSLOR algorithms for several difference formulas in different mesh geometries is discussed in detail in section 6.2. Section 6.3 analyzes the convergence behavior of particular algorithms in numerical experiments performed for the test problems of section 3.5. These results show that modified algorithms provide solutions with an increased rate of convergence compared to the standard methods.

6.1 Matrix notation

We assume that in the solved linear system

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{c}, \tag{6.1}$$

the scalar decomposition of $\mathbf{A} = \mathbf{K} - \mathbf{L} - \mathbf{U}$ is defined in such a way that if

$$\mathbf{L} = \mathbf{L}_1 + \mathbf{L}_2 \quad \text{and} \quad \mathbf{U} = \mathbf{U}_1 + \mathbf{U}_2, \tag{6.2}$$

then with the notation

$$\overline{\mathbf{B}} = \mathbf{K} - \mathbf{L}_1 - \mathbf{U}_1, \quad \overline{\mathbf{L}} = \mathbf{L}_2 \quad \text{and} \quad \overline{\mathbf{U}} = \mathbf{U}_2 \tag{6.3}$$

and the assumed splitting

$$\overline{\mathbf{A}} = \overline{\mathbf{M}} - \overline{\mathbf{N}}, \tag{6.4}$$

we have the following definitions of 1-line methods, analyzed in detail in subsection 4.3.1.

The 1-line Jacobi:

$$\overline{\mathbf{M}}_J = \overline{\mathbf{B}}, \quad \overline{\mathbf{N}}_J = \mathbf{L}_2 + \mathbf{U}_2, \quad \overline{\mathcal{B}}_1 = \overline{\mathbf{M}}_J^{-1}\overline{\mathbf{N}}_J = \overline{\mathbf{B}}^{-1}\Big(\mathbf{L}_2 + \mathbf{U}_2\Big) \tag{6.5}$$

The 1-line forward Gauss-Seidel:

$$\begin{aligned} \overline{\mathbf{M}}_{fG} &= \overline{\mathbf{B}} - \mathbf{L}_2, \qquad \overline{\mathbf{N}}_{fG} = \mathbf{U}_2, \\ \overline{\mathcal{L}}_1^f &= \overline{\mathbf{M}}_{fG}^{-1}\overline{\mathbf{N}}_{fG} = \Big(\overline{\mathbf{B}} - \mathbf{L}_2\Big)^{-1}\mathbf{U}_2 = \Big(\mathbf{I} - \overline{\mathbf{B}}^{-1}\mathbf{L}_2\Big)^{-1}\overline{\mathbf{B}}^{-1}\mathbf{U}_2 \end{aligned} \tag{6.6a}$$

The 1-line backward Gauss-Seidel:

$$\begin{aligned} \overline{\mathbf{M}}_{bG} &= \overline{\mathbf{B}} - \mathbf{U}_2, \qquad \overline{\mathbf{N}}_{bG} = \mathbf{L}_2 \\ \overline{\mathcal{L}}_1^b &= \overline{\mathbf{M}}_{bG}^{-1}\overline{\mathbf{N}}_{bG} = \Big(\overline{\mathbf{B}} - \mathbf{U}_1\Big)^{-1}\mathbf{L}_2 = \Big(\mathbf{I} - \overline{\mathbf{B}}^{-1}\mathbf{U}_2\Big)^{-1}\overline{\mathbf{B}}^{-1}\mathbf{L}_2. \end{aligned} \tag{6.6b}$$

In 1-line orderings, the $\overline{\mathbf{B}}$ are always tridiagonal matrices, so their inverses can be replaced by factoring $\overline{\mathbf{B}}$ as follows:

$$\overline{\mathbf{B}} = \mathbf{K} - \mathbf{L}_1 - \mathbf{U}_1 = \Big(\mathbf{I} - \mathbf{L}_1\mathbf{P}^{-1}\Big)\mathbf{P}\Big(\mathbf{I} - \mathbf{P}^{-1}\mathbf{U}_1\Big), \tag{6.7}$$

where $\mathbf{P}$ is assumed to be a nonsingular diagonal matrix, computed using the implicit relation

$$\mathbf{P} = \mathbf{K} - \mathbf{L}_1\mathbf{P}^{-1}\mathbf{U}_1. \tag{6.8}$$

Such a computation of $\mathbf{P}$ is possible because the strictly lower triangular matrix $\mathbf{L}_1$ has nonzero entries only on one subdiagonal, and the strictly upper triangular matrix $\mathbf{U}_1$ has nonzero entries only on the symmetric superdiagonal. This means that the product $\mathbf{L}_1\mathbf{U}_1$ is a diagonal matrix with the first entry on the diagonal equal to zero. This property allows us to compute all entries of $\mathbf{P}$ because $p_{1,1} = k_{1,1}$, and when we compute $p_{i,i}$ for $i > 1$, only the entry $p_{i-1,i-1}$ contributes to $\mathbf{L}_1\mathbf{P}^{-1}\mathbf{U}_1$.

For the 1-line Jacobi splitting, the equation $\mathbf{A}\boldsymbol{\phi} = \mathbf{c}$ is solved by

$$\overline{\mathbf{B}}\boldsymbol{\phi}^{(t+1)} = \Big(\mathbf{L}_2 + \mathbf{U}_2\Big)\boldsymbol{\phi}^{(t)} + \mathbf{c} \tag{6.9}$$

or, equivalently, by

$$\Big(\mathbf{I} - \mathbf{L}_1\mathbf{P}^{-1}\Big)\mathbf{P}\Big(\mathbf{I} - \mathbf{P}^{-1}\mathbf{U}_1\Big)\boldsymbol{\phi}^{(t+1)} = \Big(\mathbf{L}_2 + \mathbf{U}_2\Big)\boldsymbol{\phi}^{(t)} + \mathbf{c}. \tag{6.10}$$

The above equation can be written in the form

$$\Big[\mathbf{I} - \mathbf{P}^{-1}\mathbf{U}_1\Big]\boldsymbol{\phi}^{(t+1)} = \mathbf{P}^{-1}\boldsymbol{\gamma}^{(t+1)}, \tag{6.11}$$

where

$$\boldsymbol{\gamma}^{(t+1)} = \Big[\mathbf{I} - \mathbf{L}_1\mathbf{P}^{-1}\Big]^{-1}\Big[(\mathbf{L}_2 + \mathbf{U}_2)\boldsymbol{\phi}^{(t)} + \mathbf{c}\Big], \tag{6.12}$$