Chapter 7

Semi-explicit prefactorization with implicit forward sweep (OLA)

In this chapter we consider semi-explicit prefactorization methods with implicit forward sweep, also known as *OLA algorithms*

Prefactorization methods for solving difference equations oriented in mesh structures were first published in the form of recursive formulas with unknowns corresponding to mesh point indices (see for example [12, 66]).

The idea of prefactorization consists in postulating the formula for the backward sweep solution and substituting it into suitable unknowns, coupled by a given difference equation. After reordering the obtained equation, the process of substitution of the backward sweep formula into unknowns, coupled by the difference equation or new ones appearing in the reordered equation, may be continued, with simultaneous reordering of the newly obtained equations. Equating the coefficients of the backward sweep formula with those of the equation obtained after final reordering, allows us to define recursive formulas for the elimination sweep and coefficients.

For example, in the case of the explicit prefactorization (AGA) methods discussed in chapter 5, when one is working in rectangular geometry with the domain depicted in figure 3.4.1, the unknown at the mesh point (m, n) is coupled in a postulated backward sweep with the unknowns at the mesh points $(m, n+1)$, $(m, n+2), \ldots, (m+1, n-2)$, $(m+1, n-1)$.

In the case of semi-explicit prefactorization methods with implicit backward sweep described in chapter 6 (also called *modified line algorithms*), in a postulated backward sweep the unknown at the mesh point (m, n) is coupled with those at the mesh points $\ldots, (m, n-2)$, $(m, n-1)$, $(m, n+1)$, $(m, n+2)$, $\ldots$ and $(m+1, n)$ for a 5-point formula, and additionally with $(m+1, n-1)$ and $(m+1, n+1)$ for a higher-order 9-point formula.

Semi-explicit prefactorization methods with implicit *forward* sweep were developed more recently [90, 91]. In these OLA methods, in a postulated *backward* sweep the unknown at mesh point (m, n) is coupled only with those on the mesh line $m + 1$, that is, with unknowns corresponding to $\ldots, (m+1, n-2)$, $(m+1, n-1)$, $(m+1, n)$, $(m+1, n+1)$, $(m+1, n+2), \ldots$.

A priori formulation of these methods in matrix notation is impossible; it can be done instead by interpreting coefficients in derived recursive formulas in mesh structures.

Section 7.1 discusses the use of particular OLA methods when solving a system of two-dimensional difference equations, indexed by mesh points for different geometries of a mesh structure. The matrix formulation of these algorithms is presented in section 7.2. In section 7.3 we study the performance of these algorithms, when they are used to solve linear systems of the type arising from the difference approximation of both self-adjoint and non-self-adjoint two-dimensional elliptic partial differential equations. Special attention is paid to determining the optimum relaxation parameters providing the maximum rate of convergence.

7.1 Implementation in mesh structures

The main topic of this section is the use of OLA methods (semi-explicit prefactorization methods with implicit forward sweep) for solving two-dimensional difference equations in different geometries of mesh structure.

Particular OLA-(l, r) algorithms are created by postulating the formula for the backward sweep, involving only mesh points lying on the mesh line $m+1$, where the point n is always included, $l \geq 0$ denotes the number of mesh points at indices $n-1, n-2, \ldots, n-l$, and $r \geq 0$ denotes the number of mesh points at indices $n+1, n+2, \ldots, n+r$ included in the mesh coupling of the backward sweep formula.

7.1.1 Rectangular geometry

Rectangular geometry, with the domain Ω_h shown in figure 3.4.1, is represented by finite-difference equations based on the 5-point and 9-point formulas (3.64) and (3.91) as well as the 9-point formula (3.116) in the reduced system.

Let us consider the 5-point formula (3.64), with the mesh point coupling shown in figure 7.1.1, i.e.,

$$k_n^m \phi_n^m = c_n^m + e_n^m \phi_n^{m-1} + l_n^m \phi_{n-1}^m + w_n^m \phi_{n+1}^m + u_n^m \phi_n^{m+1}, \tag{7.1}$$

where $u_n^m \neq 1$, $1 \leq m \leq M$, and $1 \leq n \leq N$.

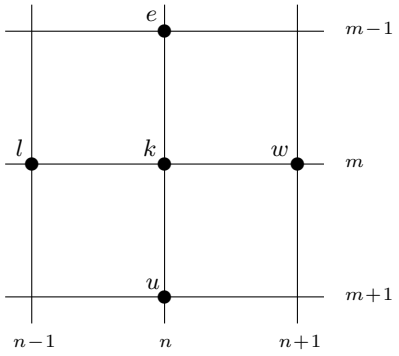

FIGURE 7.1.1.

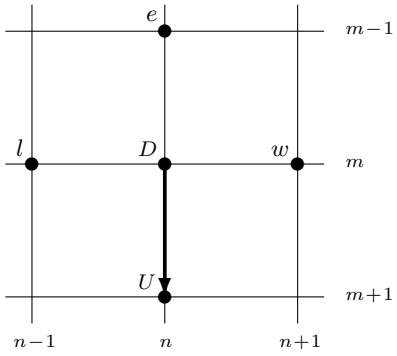

FIGURE 7.1.2.

The RecOLA-(0,0) algorithm

The RecOLA-(0,0) algorithm, that is, RecOLA-(l,r) with $l=r=0$, is the simplest semi-explicit prefactorization method with implicit forward sweep. We postulate the solution in the form

$$\phi_n^m = \frac{\beta_n^m + \psi_n^m + U_n^m \phi_n^{m+1}}{D_n^m}, \tag{7.2}$$

with the mesh point coupling shown in figure 7.1.2. Rewriting the above equation at mesh points $(m-1,n)$, $(m,n-1)$, and $(m,n+1)$ i.e.,

$$\phi_n^{m-1} = \frac{\beta_n^{m-1} + \psi_n^{m-1} + U_n^{m-1}\phi_n^m}{D_n^{m-1}}, \quad \phi_{n-1}^m = \frac{\beta_{n-1}^m + \psi_{n-1}^m + U_{n-1}^m \phi_{n-1}^{m+1}}{D_{n-1}^m}$$

$$\text{and} \quad \phi_{n+1}^m = \frac{\beta_{n+1}^m + \psi_{n+1}^m + U_{n+1}^m \phi_{n+1}^{m+1}}{D_{n+1}^m},$$

and substituting it into (7.1), we get

$$\begin{aligned}\left(k_n^m - E_n^m U_n^{m-1}\right)\phi_n^m = c_n^m + E_n^m\left(\beta_n^{m-1} + \psi_n^{m-1}\right) + L_n^m\left(\beta_{n-1}^m + \psi_{n-1}^m + U_{n-1}^m\phi_{n-1}^{m+1}\right) \\ + W_n^m\left(\beta_{n+1}^m + \psi_{n+1}^m + U_{n+1}^m\phi_{n+1}^{m+1}\right) + u_n^m\phi_n^{m+1},\end{aligned} \tag{7.3}$$

where

$$E_n^m = \frac{e_n^m}{D_n^{m-1}}, \quad L_n^m = \frac{l_n^m}{D_{n-1}^m}, \quad \text{and} \quad W_n^m = \frac{w_n^m}{D_{n+1}^m}. \tag{7.4}$$

After equating coefficients in (7.2) and (7.3), we have

$$D_n^m = k_n^m - E_n^m U_n^{m-1} \quad \text{and} \quad U_n^m = u_n^m, \tag{7.5}$$

$$\beta_n^m = c_n^m + E_n^m\left(\beta_n^{m-1} + \psi_n^{m-1}\right) + L_n^m\beta_{n-1}^m + W_n^m\beta_{n+1}^m \tag{7.6}$$

and

$$\psi_n^m = L_n^m\left(\psi_{n-1}^m + U_{n-1}^m\phi_{n-1}^{m+1}\right) + W_n^m\left(\psi_{n+1}^m + U_{n+1}^m\phi_{n+1}^{m+1}\right). \tag{7.7}$$

Since the two last equations are 3-point formulas *with regard to the mesh line* m, these equations can be solved by postulating their backward solutions, similar to the case of the 1-line Gauss-Seidel algorithm discussed in subsection 4.3.1. In the case of (7.6), we can postulate for each mesh line m the backward solution

$$\beta_n^m = \frac{\gamma_n^m + W_n^m\beta_{n+1}^m}{P_n^m}. \tag{7.8}$$

Rewriting the above equation at the mesh point $(m,n-1)$ and substituting it into (7.6), gives the forward elimination equation

$$\gamma_n^m = c_n^m + E_n^m\left(\beta_n^{m-1} + \psi_n^{m-1}\right) + Q_n^m\gamma_{n-1}^m, \tag{7.9}$$

where

$$Q_n^m = \frac{L_n^m}{P_{n-1}^m} \quad \text{and} \quad P_n^m = 1 - Q_n^m W_{n-1}^m \tag{7.10}$$