JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 76 (1996) 13–30

Recursive interpolation algorithm: a formalism for solving systems of linear equations—I. Direct methods

A. Messaoudi

Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA-M3, Université des Sciences et Technologies de Lille, F-59655 Villeneuve d'Ascq Cedex, France

Received 5 July 1995; revised 2 February 1996

Abstract

This paper presents a simple unifying algorithm for solving systems of linear equations. Solving a system of linear equations will be interpreted as an interpolation problem. This new approach leads us to a general algorithm called the recursive interpolation algorithm RIA, which includes the direct methods and some of the iterative methods. A version of the RIA with pivoting strategy will be given. We will also show how to choose two free sets of parameters in the RIA for recovering known direct methods. Other choices of these parameters yield some new methods.

Keywords: Schur complements; Sylvester's identity; Recursive interpolation algorithm; Projector; Direct methods

AMS classification: AMS(MOS) 65F10

1. Introduction

The first contribution is to present a unified approach to the majority of the existing algorithms for solving systems of linear equations. They are embedded in a general class of algorithms, the RIA where they correspond to particular choices of two free parameters. The RIA contains essentially all possible algorithms with the following property: they can solve, in exact arithmetic, a linear system starting from an arbitrary point and in a number of iterations no greater than the number of equations. The majority of the direct and iterative methods proposed in the literature have this property and fall therefore into the RIA. The second contribution of this paper lies in that new formulations of classical algorithms may be computationally attractive and compete with classical formulations. The Part I of this paper is organized as follows: in Section 2 we recall the Schur complements and the Sylvester identity, we also give the formulation and briefly the construction of the RIA and prove some of its properties. We study also an important particular case of this algorithm and the version of the RIA with a pivoting strategy will be also given. Section 3 is concerned with application of the RIA to systems of linear equations. We show how to choose the free parameters in the RIA for

PII S0377-0427(96)00066-0

recovering known direct methods. We give also some new algorithms. The interested reader may look at other choices of these free sets of parameters for finding additional new methods.

2. Recursive interpolation algorithm: RIA

In [3] Brezinski proposed two algorithms, called the *recursive interpolation algoritm* (RIA) and the *recursive projection algoritm* (RPA). These algorithms have been applied for implementing some vector sequence transformations, which can be expressed as a ratio of two determinants [3, 4, 7, 8, 13, 15]. They are connected to other methods used in numerical analysis [5, 7]. These algorithms have also been applied for implementing some vector extrapolation methods for solving systems of linear and nonlinear equations [13, 14, 21]. In this section we will give the formulation of the RIA and recall some of its properties which have been studied in [14, 17], and we will use the projectors for deriving some new properties. We will also give the version with pivoting strategy of the RIA. A particular class of this algorithm will be also studied. For constructing the RIA we need some properties of the Schur complement and the vector Sylvester identity. We recall them briefly, for more details see [4, 5, 9, 11, 18].

2.1. Schur complements and Sylvester's identity

First let us recall the definition of the Schur complement [5, 9, 11, 18] and give some of its properties [18].

Definition 2.1. Let M be a matrix partitioned into four blocks

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{2.1}$$

where the submatrix D is assumed to be square and nonsingular. The Schur complement of D in M, denoted by (M/D), is defined by

$$(M/D) = A - BD^{-1}C. \tag{2.2}$$

Let us now give some properties of the Schur complements. It is easy to show the following properties.

Property 2.1. *Let us assume that the matrix D is nonsingular; then*

$$\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix} \Big/ D\right) = \left(\begin{bmatrix} D & C \\ B & A \end{bmatrix} \Big/ D\right) = \left(\begin{bmatrix} B & A \\ D & C \end{bmatrix} \Big/ D\right) = \left(\begin{bmatrix} C & D \\ A & B \end{bmatrix} \Big/ D\right). \tag{2.3}$$

Property 2.2. *Assuming that the matrix D is nonsingular and E is a matrix such that the product EA is well defined, then*

$$\left(\begin{bmatrix} EA & EB \\ C & D \end{bmatrix} \Big/ D\right) = E\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix} \Big/ D\right). \tag{2.4}$$

Now we will give the Sylvester identity [4, 18].

Property 2.3 (*The Sylvester identity*). *Let M be the matrix defined by* (2.1) *and K be the matrix partitioned as follows*:

$$K = \begin{bmatrix} E & F & G \\ H & A & B \\ L & C & D \end{bmatrix}.$$

If the matrices A and M are nonsingular, then we have

$$\begin{aligned}(K/M) &= ((K/A)/(M/A)) \\ &= \left(\begin{bmatrix} E & F \\ H & A \end{bmatrix} \Big/ A\right) - \left(\begin{bmatrix} F & G \\ A & B \end{bmatrix} \Big/ A\right)(M/A)^{-1}\left(\begin{bmatrix} H & A \\ L & C \end{bmatrix} \Big/ A\right).\end{aligned} \tag{2.5}$$

We will use the Schur complement and the Sylvester identity for obtaining the RIA.

2.2. Formulation of the RIA

Let $x, u_1, \ldots, u_n, z_1, \ldots, z_n$ be elements of $\mathbb{R}^n$. We denote by $\langle \cdot, \cdot \rangle$ the usual inner product in $\mathbb{R}^n$, by $\| \cdot \|$ the corresponding norm and by $| \cdot |$ the determinant. We assume that the vectors $u_1, \ldots, u_n$ are linearly independent. Now we define a vector interpolation problem as follows:

Let x_0 be an arbitrary vector of $\mathbb{R}^n$. For $k \leqslant n$ find the vector x_k of $\mathbb{R}^n$ such that

$$x_k = x_0 + \sum_{i=1}^{k} \alpha_i u_i \tag{2.6}$$

and for $j = 1, \ldots, k$

$$\langle z_j, x_k \rangle = \langle z_j, x \rangle. \tag{2.7}$$

We will show briefly how to solve this problem, for more details see [3–5]. The relation (2.7) can be written explicitly as follows:

$$\begin{bmatrix} \langle z_1, u_1 \rangle & \cdots & \langle z_1, u_k \rangle \\ \cdots & \cdots & \cdots \\ \langle z_k, u_1 \rangle & \cdots & \langle z_k, u_k \rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \cdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \langle z_1, x - x_0 \rangle \\ \cdots \\ \langle z_k, x - x_0 \rangle \end{bmatrix}. \tag{2.8}$$

Denoting by D_k the matrix of the linear system (2.8) and assuming that this matrix is nonsingular we get from (2.6)

$$x_k = x_0 + [u_1, \ldots, u_k] D_k^{-1} \begin{bmatrix} \langle z_1, x - x_0 \rangle \\ \cdots \\ \langle z_k, x - x_0 \rangle \end{bmatrix}. \tag{2.9}$$

From (2.9) we see that x_k can be expressed as a Schur complement (2.2),

$$x_k = -\left(\begin{bmatrix} -x_0 & u_1 & \cdots & u_k \\ \langle z_1, x - x_0 \rangle & \langle z_1, u_1 \rangle & \cdots & \langle z_1, u_k \rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle z_k, x - x_0 \rangle & \langle z_k, u_1 \rangle & \ldots & \langle z_k, u_k \rangle \end{bmatrix} \Big/ D_k\right). \tag{2.10}$$

For computing x_k recursively we need the following.

Definition 2.2. D_k is said to be a strongly nonsingular matrix if $|D_m| \neq 0$ for $m = 1,\dots,k$.

Now we assume that D_k is a strongly nonsingular matrix. We set for $m = 1,\dots,k$

$$x_m = -\left(\begin{bmatrix} -x_0 & u_1 & \cdots & u_m \\ \langle z_1, x - x_0\rangle & \langle z_1, u_1\rangle & \cdots & \langle z_1, u_m\rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle z_m, x - x_0\rangle & \langle z_m, u_1\rangle & \cdots & \langle z_m, u_m\rangle \end{bmatrix} \Big/ D_m\right),$$

and for $m = 1,\dots,k-1$ and for $i > m$, we set

$$g_{m,i} = \left(\begin{bmatrix} u_i & u_1 & \cdots & u_m \\ \langle z_1, u_i\rangle & \langle z_1, u_1\rangle & \cdots & \langle z_1, u_m\rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle z_m, u_i\rangle & \langle z_m, u_1\rangle & \cdots & \langle z_m, u_m\rangle \end{bmatrix} \Big/ D_m\right). \tag{2.11}$$

So applying the vector Sylvester identity (2.5) to x_m, and properties of the Schur complements (2.3) and (2.4) we get

$$x_m = x_{m-1} + \frac{\langle z_m, x - x_{m-1}\rangle}{\langle z_m, g_{m-1,m}\rangle} g_{m-1,m}, \tag{2.12}$$

where $g_{m-1,m}$ is the vector defined by (2.11), which can be computed by using the same Sylvester identity

$$\begin{aligned} & g_{0,m} = u_m, \\ & \textit{for } i = 1,\dots,m-1 \\ & \quad g_{i,m} = g_{i-1,m} - \frac{\langle z_i, g_{i-1,m}\rangle}{\langle z_i, g_{i-1,i}\rangle} g_{i-1,i}, \\ & \textit{end } i. \end{aligned} \tag{2.13}$$

From (2.12) and (2.13) we get the *recursive interpolation algorithm* (RIA) which is described as follows:

Algorithm 1: RIA

$$\begin{aligned} & x_0 \textit{ is an arbitrary vector}, \quad g_{0,1} = u_1, \\ & \textit{for } m = 1,\dots,k, \\ & \quad x_m = x_{m-1} + \frac{\langle z_m, x - x_{m-1}\rangle}{\langle z_m, g_{m-1,m}\rangle} g_{m-1,m}, \\ & \quad g_{0,m+1} = u_{m+1}, \\ & \quad \textit{for } i = 1,\dots,m, \\ & \qquad g_{i,m+1} = g_{i-1,m+1} - \frac{\langle z_i, g_{i-1,m+1}\rangle}{\langle z_i, g_{i-1,i}\rangle} g_{i-1,i}, \\ & \quad \textit{end } i, \\ & \textit{end } m. \end{aligned}$$

Remark that the RIA, obtained with two free sets of parameters z_m and u_m, is well defined (no break-down) if and only if $\langle z_m, g_{m-1,m}\rangle \neq 0$ for $m = 1,\ldots,k$. We will see that this condition is satisfied if and only if D_k is a strongly nonsingular matrix. Now we will give some of its properties.

2.3. Some properties of the RIA

We will recall some properties of the RIA, for more details see [17], and we will use projectors for giving some new properties. We will also study an important particular case of the RIA.

Proposition 2.1. *If D_k is a strongly nonsingular matrix, then we have*

(1) $\langle z_m, g_{m-1,m}\rangle = |D_m|/|D_{m-1}|$, *for* $m = 1,\ldots,k$, *with* $|D_0| = 1$.
(2) *The* RIA *is well defined (i.e. no break-down).*
(3) $\langle z_m, g_{j,i}\rangle = 0$, *for* $i > j \geqslant m$.
(4) $\langle z_m, x_i\rangle = \langle z_m, x\rangle$, *for* $i = 1,\ldots,k$ *and* $m = 1,\ldots,i$.
(5) $g_{j,i}$ *is a linear combination of the vectors* $u_1,\ldots,u_j,u_i$.
(6) *The vectors* $g_{0,1}, g_{1,2},\ldots,g_{m-1,m}$ *generated by the* RIA *are linearly independent.*
(7) *There exists* $k_0 \leqslant n$ *such that* $x_{k_0} = x$.

Remark 2.1. From (5) and (6) of Proposition 2.1 we see that the vectors $\{u_1,\ldots,u_m\}$ and $\{g_{0,1},\ldots,g_{m-1,m}\}$ generate the same subspace. And from (2.13) we have

$$g_{i,m} = u_m - \sum_{j=1}^{i} \frac{\langle z_j, g_{j-1,m}\rangle}{\langle z_j, g_{j-1,j}\rangle} g_{j-1,j} \tag{2.14}$$

The process used in the RIA for computing the vectors $g_{m-1,m}$ for $m = 1,\ldots,n$ can be interpreted as a process for constructing a new basis of $\mathbb{R}^n$ from the old basis u_m. The vectors $g_{m-1,m}$ generated by the RIA can be normalized to have length 1.

Now we will give some new properties of the RIA. We set for $m = 1,\ldots,k$

$$U_m = [u_1,\ldots,u_m], \tag{2.15}$$

$$Z_m = [z_1,\ldots,z_m], \tag{2.16}$$

$$Q_m = U_m[Z_m^{\mathrm{T}} U_m]^{-1} Z_m^{\mathrm{T}} \tag{2.17}$$

where Z^{T} is the adjoint of the matrix Z.

Remark 2.2. It is easy to see that for $m = 1,\ldots,k$ we have

$$D_m = Z_m^{\mathrm{T}} U_m, \tag{2.18}$$

$$x_m = x_0 + Q_m(x - x_0), \tag{2.19}$$

$$g_{m,i} = (I - Q_m)u_i, \quad \text{for } i > m. \tag{2.20}$$

Note that the matrices $D_m = Z_m^T U_m$ for $m = 1, \ldots, k$ do not have particular structure; they are only nonsingular.

Proposition 2.2. *If D_k is a strongly nonsingular matrix, then we have*
(1) *Q_m is an oblique projector along the column space of $Z_m^\perp$ on the column space of U_m.*
(2) $(I - Q_m)^2 = I - Q_m$.
(3) $Q_m Q_i = Q_i Q_m = Q_i$, *if* $m \geqslant i$.

Proof. $Q_m^2 = Q_m$ and $(I - Q_m)^2 = I - Q_m$ are given immediately by using (2.17). Let us remark that $Q_m U_m = U_m$ and $Q_m^T Z_m = Z_m$ so the result (1) of the proposition follows. (3) is proved by using (1). □

Now let us set for $m = 1, \ldots, k$

$$G_m = [g_{0,1}, \ldots, g_{m-1,m}], \tag{2.21}$$

$$Q'_m = G_m [Z_m^T G_m]^{-1} Z_m^T. \tag{2.22}$$

Proposition 2.3. *If D_k is a strongly nonsingular matrix, then we have*
(1) *$Z_m^T G_m$ is a lower triangular matrix.*
(2) *Q'_m is an oblique projector along the column space of $Z_m^\perp$ on the column space of G_m.*
(3) $(I - Q'_m)^2 = I - Q'_m$.
(4) $Q'_m Q'_i = Q'_i Q'_m = Q'_i$, *if* $m \geqslant i$.

Proof. Item (1) of the proposition follows from (3) of Proposition 2.1. The other points use the same proof of Proposition 2.2. □

Proposition 2.4. *If D_k is a strongly nonsingular matrix, then we have for $m = 1, \ldots, k$*

$$Q'_m = Q_m, \tag{2.23}$$

$$x_m = x_0 + Q'_m (x - x_0), \tag{2.24}$$

$$g_{m,i} = (I - Q'_m) u_i \quad \text{for } i > m. \tag{2.25}$$

Proof. Eq. (2.23) of this proposition will be proved by induction. For $m = 1$ we have $Q'_1 = Q_1$ because $g_{0,1} = u_1$. Assume now that (2.23) is true for $m - 1$ with $m \geqslant 2$, we will prove it for m. First let us consider for $m = 2, \ldots, k$ the matrix $Z_m^T G_m$ partitioned as follows:

$$Z_m^T G_m = \begin{bmatrix} Z_{m-1}^T G_{m-1} & 0 \\ z_m^T G_{m-1} & \langle z_m, g_{m-1,m} \rangle \end{bmatrix}.$$

Then we get

$$[Z_m^T G_m]^{-1} = \begin{bmatrix} [Z_{m-1}^T G_{m-1}]^{-1} & 0 \\ -\dfrac{z_m^T G_{m-1} [Z_{m-1}^T G_{m-1}]^{-1}}{\langle z_m, g_{m-1,m} \rangle} & \dfrac{1}{\langle z_m, g_{m-1,m} \rangle} \end{bmatrix}. \tag{2.26}$$

We have from (2.22) and (2.26)

$$
\begin{aligned}
Q'_m &= G_m[Z_m^{\mathrm{T}}G_m]^{-1}Z_m^{\mathrm{T}} \\
&= [G_{m-1}, g_{m-1,m}] \begin{bmatrix} [Z_{m-1}^{\mathrm{T}}G_{m-1}]^{-1} & 0 \\ -\dfrac{z_m^{\mathrm{T}}G_{m-1}[Z_{m-1}^{\mathrm{T}}G_{m-1}]^{-1}}{\langle z_m, g_{m-1,m}\rangle} & \dfrac{1}{\langle z_m, g_{m-1,m}\rangle} \end{bmatrix} \begin{bmatrix} Z_{m-1}^{\mathrm{T}} \\ z_m^{\mathrm{T}} \end{bmatrix} \\
&= Q'_{m-1} + \frac{g_{m-1,m}z_m^{\mathrm{T}}}{\langle z_m, g_{m-1,m}\rangle}(I - Q'_{m-1}),
\end{aligned}
$$

and for Q_m we have

$$
D_m^{-1} = [Z_m^{\mathrm{T}}U_m]^{-1} = \begin{bmatrix} Z_{m-1}^{\mathrm{T}}U_{m-1} & Z_{m-1}^{\mathrm{T}}u_m \\ z_m^{\mathrm{T}}U_{m-1} & \langle z_m, u_m\rangle \end{bmatrix}^{-1}
$$

$$
= \begin{bmatrix} D_{m-1}^{-1} + D_{m-1}^{-1}Z_{m-1}^{\mathrm{T}}u_m(D_m/D_{m-1})^{-1}z_m^{\mathrm{T}}U_{m-1}D_{m-1}^{-1} & -D_{m-1}^{-1}Z_{m-1}^{\mathrm{T}}u_m(D_m/D_{m-1})^{-1} \\ -(D_m/D_{m-1})^{-1}z_m^{\mathrm{T}}U_{m-1}D_{m-1}^{-1} & (D_m/D_{m-1})^{-1} \end{bmatrix},
$$

using (2.20) we remark that

$$
\begin{aligned}
(D_m/D_{m-1}) &= \langle z_m, u_m\rangle - z_m^{\mathrm{T}}U_{m-1}D_{m-1}^{-1}Z_{m-1}^{\mathrm{T}}u_m \\
&= \langle z_m, (I - Q_{m-1})u_m\rangle \\
&= \langle z_m, g_{m-1,m}\rangle,
\end{aligned} \tag{2.27}
$$

using (2.20), (2.26), (2.27) and the fact that $Q'_{m-1} = Q_{m-1}$, we get

$$
\begin{aligned}
Q_m &= U_m[Z_m^{\mathrm{T}}U_m]^{-1}Z_m^{\mathrm{T}} = [U_{m-1}, u_m]D_m^{-1}\begin{bmatrix} Z_{m-1}^{\mathrm{T}} \\ z_m^{\mathrm{T}} \end{bmatrix} \\
&= Q_{m-1} + (D_m/D_{m-1})^{-1}(I - Q_{m-1})u_m z_m^{\mathrm{T}}(I - Q_{m-1}) \\
&= Q'_{m-1} + \frac{g_{m-1,m}z_m^{\mathrm{T}}}{\langle z_m, g_{m-1,m}\rangle}(I - Q'_{m-1}) \\
&= Q'_m.
\end{aligned}
$$

Eqs. (2.24) and (2.25) of the proposition follow immediatly from (2.19), (2.20) and (2.23). □

Many methods for solving systems of linear equations are particular cases of the RIA; see 3 and Part II [19]. An important particular case of the RIA is defined by choosing $z_m = Hg_{m-1,m}$, where H is assumed to be a symmetric matrix. For this case we give some important properties. Let $u_1, \ldots, u_n$

be linearly independent vectors of $\mathbb{R}^n$, and let Q'_m be the matrix defined by (2.22). Then, for this particular case, we have the following result.

Proposition 2.5. *If for $m=1,\dots,k$, we choose $z_m = Hg_{m-1,m}$, where H is assumed to be a symmetric matrix, and if D_k is a strongly nonsingular matrix, then we have*

(1) $Q_m'^{\mathrm{T}} H = HQ'_m$.
(2) $\langle z_m, x - x_{m-1}\rangle = \langle z_m, x - x_0\rangle = \langle Hu_m, x - x_{m-1}\rangle$.
(3) $\langle z_m, g_{m-1,i}\rangle = \langle Hu_m, g_{m-1,i}\rangle = \langle z_m, u_i\rangle$, *for* $i \geqslant m$.
(4) $Z_m^{\mathrm{T}} U_m$ *is an upper triangular matrix.*
(5) $Z_m^{\mathrm{T}} G_m$ *is a diagonal matrix.*

Proof. As H is a symmetric matrix and $Z_m = HG_m$ then item (1) of the proposition follows immediatly. For proving (2) and (3) we use the fact that $x_m = x_0 + Q'_m(x - x_0)$, $g_{m,i} = (I - Q'_m)u_i$ and that Q'_m is a projector. For proving (4) we use (3) of Proposition 2.2 and (2.20). (5) of the proposition follows immediatly from the fact that $Z_m^{\mathrm{T}} G_m = G_m^{\mathrm{T}} HG_m$ is a symmetric and lower triangular matrix. □

Remark 2.3. From (2) and (3) of Proposition 2.5 we see that the choice $z_m = Hg_{m-1,m}$ or $z_m = Hu_m$ leads to mathematically equivalent algorithms.

In the formulation of the RIA we have assumed that D_k is a strongly nonsingular matrix (i.e. no break-down), in general this condition is not satisfied and the RIA will fail. For avoiding this difficulty we use a pivoting strategy. This version will be denoted by PRIA, it can be described as follows:

Algorithm 2: PRIA

1. x_0 *is an arbitrary vector,* $g_{0,1} = u_1$,
2. *for* $m = 1,\dots,k$, $j = 0$,
3. *if* $\langle z_m, g_{m-1,m}\rangle \neq 0$,

$$x_m = x_{m-1} + \frac{\langle z_m, x - x_{m-1}\rangle}{\langle z_m, g_{m-1,m}\rangle} g_{m-1,m},$$

$g_{0,m+1} = u_{m+1}$,
for $i = 1,\dots,m$,

$$g_{i,m+1} = g_{i-1,m+1} - \frac{\langle z_i, g_{i-1,m+1}\rangle}{\langle z_i, g_{i-1,i}\rangle} g_{i-1,i},$$

end i,
otherwise,
$j = j + 1$, $v = z_{m+j}$, $z_{m+j} = z_{m+j-1}$, $z_{m+j-1} = z_m$, $z_m = v$, *go to* 3,
end m.

Note that the PRIA is obtained by pivoting the mth row of D_k with the $(m+j)$th row.

3. Application: direct methods

Let us consider the linear system

$$Ax = b, \tag{3.1}$$

where $A = (a_{i,j})$ is an $n \times n$ real nonsingular matrix and $b = (b_1, \ldots, b_n)^T$ is a given vector of $\mathbb{R}^n$. x is the unique exact solution of (3.1). If, in the RIA, we choose z_m and u_m such that $\langle z_m, x - x_{m-1} \rangle$ is known, then from (7) of Proposition 2.1 we know that there exists $k_0 \leqslant n$ such that $x_{k_0} = x$. For the mth iterate vector x_m we define the associate residual vector, $r_m = b - Ax_m$, given by

$$r_m = r_{m-1} - \frac{\langle z_m, x - x_{m-1} \rangle}{\langle z_m, g_{m-1,m} \rangle} A g_{m-1,m}. \tag{3.2}$$

For applying the RIA we have to solve two problems:

P1. The RIA must be well defined (i.e. D_k is a strongly nonsingular matrix).

P2. z_m must be choosen such that $\langle z_m, x - x_{m-1} \rangle$ is known.

The majority of the methods proposed in the literature, for solving (3.1), have the following property: they can solve, in exact arithmetic, the linear system (3.1) in a number of iterations no greater than n. There are two classes of these methods: direct methods and iterative methods. The direct methods will be defined from the RIA by choosing z_m directly from the system (3.1). If problem P2 is also solved for this choice we call such a method explicit direct method. We find in this class the Bordering method [6, 9], the Reinforcement method [9, 10], the Sloboda method [22], the Huang method [12], a class of the unscaled ABS algorithms [1], and others. A special case, which is a generalization of the Purcell method, see [14], will be also studied. If we use the lower triangular matrix $Z_n^T G_n$ (see Proposition 2.3) for obtaining the solution of the system (3.1), this method will be called implicit direct method. We find in this class the implicit Sloboda method, the implicit Huang method, the LU-factorization [9, 11, 16], the QR-factorization [9, 11] and others. The iterative methods will be obtained from the RIA by choosing z_m depending on the preceding iterations. Now we will specify the choice of u_m and z_m to obtain these methods and we will also give a new formulation for some of them. Some of the iterative methods obtained from the RIA will be studied in Part II [19]. We also give other choices with the goal to obtain new methods. For $m = 1, \ldots, n$, we set $a_m = (a_{m,1}, \ldots, a_{m,n})^T$ and $\tilde{a}_m = (a_{1,m}, \ldots, a_{n,m})^T$.

3.1. Explicit direct methods

We mean by explicit direct method any algorithm obtained from the RIA by choosing z_m directly from the system (3.1) such that x_m can be computed from x_{m-1}. We will recall some well known methods which are particular cases of the RIA. We will give also some new algorithms.

3.1.1. The Sloboda method

This method is a general one [22] which contains the Bordering method, the Reinforcement method and the Huang method. It is described as follows:

Algorithm 3: The Sloboda method

$x_0^{(0)}, x_0^{(1)}, \ldots, x_0^{(n)}$, *are arbitrary vectors,*

for $m = 1, \ldots, n$,

$v_{m-1}^{(m)} = x_{m-1}^{(m)} - x_{m-1}^{(m-1)}$,

For $i = m, \ldots, n$,

$$x_m^{(i)} = x_{m-1}^{(i)} + \frac{b_m - \langle a_m, x_{m-1}^{(i)} \rangle}{\langle a_m, v_{m-1}^{(m)} \rangle} v_{m-1}^{(m)},$$

end i,
end m.

It has been proved in [22] that $x_n^{(n)}=x$, and if we set for $i=1,\dots,n-1$ and $m=i+1,\dots,n$,

$$v_i^{(m)}=x_i^{(m)}-x_i^{(i)}, \tag{3.3}$$

then we get

$$v_i^{(m)}=v_{i-1}^{(m)}-\frac{\langle a_i,v_{i-1}^{(m)}\rangle}{\langle a_i,v_{i-1}^{(i)}\rangle}v_{i-1}^{(i)}. \tag{3.4}$$

Now we will show that the Sloboda method is a particular case of the RIA, and we will give the connection between this method and other methods. From (3.4) and the auxiliary vectors $g_{i,m}$ used in the RIA it is clear that we have $g_{i,m}=v_i^{(m)}$, by choosing $z_m=a_m$ we also have $g_{0,m}=u_m=v_0^{(m)}=x_0^{(m)}-x_0^{(0)}$ for $m=1,\dots,n$. Thus, we have the following result.

Theorem 3.1. *If we choose, in the* RIA, $x_0=x_0^{(0)}$, $z_m=a_m$ *and* $u_m=x_0^{(m)}-x_0^{(0)}$ *for* $m=1,\dots,n$, *then we get the Sloboda method.*

Proof. From the above discussion it is sufficient to show that $x_m=x_m^{(m)}$. We have from the Sloboda method and (3.3)

$$\begin{aligned}
x_m^{(m)}&=x_{m-1}^{(m)}+\frac{b_m-\langle a_m,x_{m-1}^{(m)}\rangle}{\langle a_m,v_{m-1}^{(m)}\rangle}v_{m-1}^{(m)}\\
&=x_{m-1}^{(m-1)}+v_{m-1}^{(m)}+\frac{\langle a_m,x\rangle-\langle a_m,x_{m-1}^{(m-1)}+v_{m-1}^{(m)}\rangle}{\langle a_m,v_{m-1}^{(m)}\rangle}v_{m-1}^{(m)}\\
&=x_{m-1}+\frac{\langle a_m,x\rangle-\langle a_m,x_{m-1}\rangle}{\langle a_m,g_{m-1,m}\rangle}g_{m-1,m}\\
&=x_m. \qquad \square
\end{aligned}$$

Then the associate algorithm, which is a modified version of the Sloboda method, is given as follows:

Algorithm 4: The modified Sloboda method

x_0 *is an arbitrary vector,* $g_{0,1}=x_0^{(1)}-x_0$,
for $m=1,\dots,n$,
$$x_m=x_{m-1}+\frac{b_m-\langle a_m,x_{m-1}\rangle}{\langle a_m,g_{m-1,m}\rangle}g_{m-1,m},$$
$g_{0,m+1}=x_0^{(m+1)}-x_0$,
for $i=1,\dots,m$,
$$g_{i,m+1}=g_{i-1,m+1}+\frac{\langle a_i,g_{i-1,m+1}\rangle}{\langle a_i,g_{i-1,i}\rangle}g_{i-1,i},$$
end i,
end m.

The x_m's and the $g_{m,m+1}$'s can be updated at the cost one extra vector ($g_{m,m+1}$) and one extra scalar ($\langle a_m, g_{m-1,m}\rangle$) in memory and $(4m+6)n$ operations at each iteration.

Remark 3.1. If we choose in the modified Sloboda method
1. $x_0 = 0$ and $x_0^{(m)} = e_m$, then we get the *Bordering method.*
2. $x_0 = b$ and $x_0^{(m)} = e_m + b$, then we get the *Reinforcement method.*
3. $x_0 = 0$ and $x_0^{(m)} = a_m$, then we get the *Huang method.*

3.1.2. The unscaled ABS algorithms

We will recall the unscaled ABS algorithms [1] and consider a particular class of this method. We show also that this class is a particular case of the RIA. The unscaled ABS algorithms are described as follows.

Algorithm 5: The unscaled ABS algorithms

$\bar{x}_0$ is an arbitrary vector, H_0 is an arbitrary nonsingular matrix,

for $m = 1, \ldots,$ until convergence do,

$$p_m = H_{m-1}^{\mathrm{T}} v_m,\ v_m \text{ is choosen such that } \langle p_m, a_m\rangle \neq 0,$$

$$\bar{x}_m = \bar{x}_{m-1} + \frac{b_m - \langle a_m, \bar{x}_{m-1}\rangle}{\langle a_m, p_m\rangle} p_m,$$

$$H_m = H_{m-1} - H_{m-1} a_m w_m^t H_{m-1},\ w_m \text{ is choosen such that } \langle w_m, H_{m-1} a_m\rangle = 1,$$

end m.

We are interested by the following particular choice of w_m:

$$w_m = \frac{v_m}{\langle a_m, p_m\rangle}. \tag{3.5}$$

This particular choice of w_m satisfies the condition $\langle w_m, H_{m-1} a_m\rangle = 1$. Now we will give the choice of z_m and u_m in the RIA for giving the unscaled ABS algorithm with w_m defined by (3.5). Let us remark that

$$H_{m-1} = H_0 - \sum_{j=1}^{m-1} H_{j-1} \frac{a_j v_j^{\mathrm{T}}}{\langle a_j, p_j\rangle} H_{j-1},$$

and

$$\begin{aligned} p_m &= H_{m-1}^{\mathrm{T}} v_m \\ &= H_0^{\mathrm{T}} v_m - \sum_{j=1}^{m-1} H_{j-1}^{\mathrm{T}} \frac{v_j a_j^{\mathrm{T}}}{\langle a_j, p_j\rangle} H_{j-1}^{\mathrm{T}} v_m \\ &= H_0^{\mathrm{T}} v_m - \sum_{j=1}^{m-1} \frac{\langle a_j, H_{j-1}^{\mathrm{T}} v_m\rangle}{\langle a_j, p_j\rangle} p_j. \end{aligned}$$

Then we get the following result.

Theorem 3.2. *Let w_m be given by* (3.5). *If we choose, in the* RIA, $x_0 = \bar{x}_0$, $z_m = a_m$ *and* $u_m = H_0^T v_m$, *then we get the corresponding class of* unscaled ABS algorithm.

Proof. From the above discussion it is sufficient to show that $g_{j,m} = H_j^T v_m$. We have $g_{0,m} = u_m = H_0^T v_m$. Now we assume that $g_{j-1,m} = H_{j-1}^T v_m$ and we will prove it for j. We have

$$\begin{aligned} g_{j,m} &= g_{j-1,m} - \frac{\langle z_j, g_{j-1,m}\rangle}{\langle z_j, g_{j-1,j}\rangle} g_{j-1,j} \\ &= H_{j-1}^T v_m - \frac{\langle a_j, H_{j-1}^T v_m\rangle}{\langle a_j, H_{j-1}^T v_j\rangle} H_{j-1}^T v_j \\ &= \left(H_{j-1}^T - H_{j-1}^T \frac{v_j a_j^T}{\langle a_j, p_j\rangle} H_{j-1}^T \right) v_m \\ &= H_j^T v_m. \qquad \square \end{aligned}$$

Then the associate algorithm, which is a modified version of the unscaled ABS algorithm, is given as follows.

Algorithm 6: The modified unscaled ABS algorithm

x_0 is an arbitrary vector, H_0 is an arbitrary nonsingular matrix,
$g_{0,1} = H_0^T v_1$, *v_1 is choosen such that* $\langle g_{0,1}, a_1\rangle \neq 0$, $m = 0$,

1. $m = m + 1$ *until convergence do,*

$$x_m = x_{m-1} + \frac{b_m - \langle a_m, x_{m-1}\rangle}{\langle a_m, g_{m-1,m}\rangle} g_{m-1,m},$$

v_{m+1} *is choosen,*

2. $g_{0,m+1} = H_0^T v_{m+1}$,

for $i = 1, \ldots, m$,

$$g_{i,m+1} = g_{i-1,m+1} - \frac{\langle a_i, g_{i-1,m+1}\rangle}{\langle a_i, g_{i-1,i}\rangle} g_{i-1,i},$$

end i,
if $\langle a_{m+1}, g_{m,m+1}\rangle \neq 0$ *go to* 1,
otherwise change v_{m+1} *and go to* 2.

The x_m's and the $g_{m,m+1}$'s can be updated at the cost one extra vector ($g_{m,m+1}$) and one extra scalar ($\langle a_m, g_{m-1,m}\rangle$) in memory and $2n^2 + (4m + 6)n$ operations at each iteration without changing v_{m+1}.

3.1.3. A special case

It has been proved in [13, 14] that if we choose $\tilde{z}_m = (a_m^T, -b_m)^T \in \mathbb{R}^{n+1}$ and $\tilde{u}_m = e_m \in \mathbb{R}^{n+1}$ and if we apply the following process:

$\tilde{g}_{0,1} = \tilde{u}_1$,
for $m = 1, \ldots, n$
$\tilde{g}_{0,m+1} = \tilde{u}_{m+1}$,

for $i = 1,\dots,m$

$$\tilde{g}_{i,m+1} = \tilde{g}_{i-1,m+1} - \frac{\langle \tilde{z}_i, \tilde{g}_{i-1,m+1}\rangle}{\langle \tilde{z}_i, \tilde{g}_{i-1,i}\rangle}\tilde{g}_{i-1,i}, \tag{3.6}$$

end i,

end m,

then we get $\tilde{g}_{n,n+1} = (x^{\mathrm{T}}, 1)^{\mathrm{T}}$, which is equivalent to the Purcell method. We will give a general case of this method and show how to choose $\tilde{z}_m$ and $\tilde{u}_m$ for obtaining the exact solution of (3.1). Let us consider for $m = 1,\dots,n$, $\tilde{z}_m = (z_m^{\mathrm{T}}, -v_m)^{\mathrm{T}}$, $\tilde{u}_m = (u_m^{\mathrm{T}}, -w_m)^{\mathrm{T}}$ and $\tilde{u}_{n+1} = (q^{\mathrm{T}}, s)^{\mathrm{T}}$ where v_m, w_m and s are scalars and consider the process (3.6) for having $\tilde{g}_{n,n+1}$. Invoking (2.20) we have

$$\tilde{g}_{n,n+1} = \tilde{u}_{n+1} - \tilde{Q}_n\tilde{u}_{n+1},$$

with $\tilde{Q}_n = \tilde{U}_n[\tilde{Z}_n^{\mathrm{T}}\tilde{U}_n]^{-1}\tilde{Z}_n^{\mathrm{T}}$, where

$$\tilde{Z}_n = [\tilde{z}_1,\dots,\tilde{z}_n] = \begin{bmatrix} Z_n \\ -v^{\mathrm{T}} \end{bmatrix},$$

$$\tilde{U}_n = [\tilde{u}_1,\dots,\tilde{u}_n] = \begin{bmatrix} U_n \\ -w^{\mathrm{T}} \end{bmatrix},$$

$v = (v_1,\dots,v_n)^{\mathrm{T}}$ and $w = (w_1,\dots,w_n)^{\mathrm{T}}$. Now assuming that $Z_n^{\mathrm{T}}U_n$ and $\tilde{Z}_n^{\mathrm{T}}\tilde{U}_n$ are nonsingular matrices and applying the Scherman–Morrisson formula [9, 10] we get

$$\begin{aligned}[\tilde{Z}_n^{\mathrm{T}}\tilde{U}_n]^{-1} &= [Z_n^{\mathrm{T}}U_n + vw^{\mathrm{T}}]^{-1} \\ &= [Z_n^{\mathrm{T}}U_n]^{-1} - \frac{[Z_n^{\mathrm{T}}U_n]^{-1}vw^{\mathrm{T}}[Z_n^{\mathrm{T}}U_n]^{-1}}{1 + w^{\mathrm{T}}[Z_n^{\mathrm{T}}U_n]^{-1}v}.\end{aligned}$$

Then using (2.18), $\tilde{Q}_n$ will be as follows:

$$\begin{aligned}\tilde{Q}_n &= \begin{bmatrix} U_n \\ -w^{\mathrm{T}} \end{bmatrix}[D_n + vw^{\mathrm{T}}]^{-1}[Z_n^{\mathrm{T}}, -v] \\ &= \begin{bmatrix} U_nD_n^{-1}Z_n^{\mathrm{T}} - \dfrac{U_nD_n^{-1}vw^{\mathrm{T}}D_n^{-1}Z_n^{\mathrm{T}}}{1 + w^{\mathrm{T}}D_n^{-1}v} & -U_nD_n^{-1}v + \dfrac{U_nD_n^{-1}vw^{\mathrm{T}}D_n^{-1}v}{1 + w^{\mathrm{T}}D_n^{-1}v} \\ -w^{\mathrm{T}}D_n^{-1}Z_n^{\mathrm{T}} + \dfrac{w^{\mathrm{T}}D_n^{-1}vw^{\mathrm{T}}D_n^{-1}Z_n^{\mathrm{T}}}{1 + w^{\mathrm{T}}D_n^{-1}v} & w^{\mathrm{T}}D_n^{-1}v - \dfrac{w^{\mathrm{T}}D_n^{-1}vw^{\mathrm{T}}D_n^{-1}v}{1 + w^{\mathrm{T}}D_n^{-1}v} \end{bmatrix}.\end{aligned}$$

Assuming that the matrices Z_n and U_n are nonsingular then we get

$$\tilde{g}_{n,n+1} = \begin{bmatrix} q - U_n D_n^{-1} Z_n^{\mathrm{T}} q + \dfrac{s}{1 + w^{\mathrm{T}} D_n^{-1} v} Z_n^{-\mathrm{T}} v \\ \\ \dfrac{1}{1 + w^{\mathrm{T}} D_n^{-1} v}(s + w^{\mathrm{T}} D_n^{-1} Z_n^{\mathrm{T}} q) \end{bmatrix} = \begin{bmatrix} \dfrac{s}{1 + w^{\mathrm{T}} D_n^{-1} v} Z_n^{-\mathrm{T}} v \\ \\ \dfrac{1}{1 + w^{\mathrm{T}} D_n^{-1} v}(s + w^{\mathrm{T}} U_n^{-1} q) \end{bmatrix}. \tag{3.7}$$

Remark 3.2. If we choose in (3.7)

1. $v = b$, $w = q = 0$, $s = 1$, $z_m = a_m$ and $u_m = e_m$, then we get the *Purcell method.*

2. $v = b$, $w = 0, z_m = a_m$, $u_m = e_m$, $s \neq 0$ and q an arbitrary vector, then we get $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$. The *Bordering* and the *Reinforcement* methods are included in this case.

3. $v = b$, $w = 0, z_m = u_m = a_m$, $s \neq 0$ and q an arbitrary vector, then we get the *Huang method,* and $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$.

4. $v = b$, $w = 0, z_m = a_m$, $s \neq 0$, q and u_m are arbitrary vectors, then we get the *Sloboda method,* and $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$.

3.1.4. Some new explicit direct methods

We now propose other choices of z_m and u_m in the RIA for obtaining some new explicit direct methods. We also give the condition for their existence. We recall that the RIA is well defined if and only if $D_n = Z_n^{\mathrm{T}} U_n$ is a strongly nonsingular matrix.

(a) $z_m = a_m$, and

$u_m = \tilde{a}_m$, $D_n = A^2$, or

$u_m = A a_m$, $D_n = A^2 A^{\mathrm{T}}$, or

$u_m = A \tilde{a}_m$, $D_n = A^3$, or

$u_m = A^{\mathrm{T}} a_m$, $D_n = A A^{\mathrm{T}} A$, or

$u_m = A^{\mathrm{T}} \tilde{a}_m$, $D_n = A A^{\mathrm{T}} A^{\mathrm{T}}$,

if we choose, in (3.7), $v = b$, $w = 0$, $s \neq 0$, and q an arbitrary vector, then we get $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$.

(b) $z_m = A^{\mathrm{T}} a_m$, and

$u_m = e_m$, $D_n = A^2$, or

$u_m = a_m$, $D_n = A^2 A^{\mathrm{T}}$, or

$u_m = \tilde{a}_m$, $D_n = A^3$,

if we choose, in (3.7), $v = Ab$, $w = 0$, $s \neq 0$, and q an arbitrary vector, then we get $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$.

(c) $z_m = A^{\mathrm{T}}\tilde{a}_m$, and
$u_m = e_m$, $D_n = A^{\mathrm{T}}A$, or
$u_m = a_m$, $D_n = A^{\mathrm{T}}AA^{\mathrm{T}}$, or
$u_m = \tilde{a}_m$, $D_n = A^{\mathrm{T}}A^2$,
if we choose, in (3.7), $v = A^{\mathrm{T}}b$, $w = 0$, $s \neq 0$, and q an arbitrary vector, then we get $\tilde{g}_{n,n+1} = (s \cdot x^{\mathrm{T}}, s)^{\mathrm{T}}$.

3.2. Implicit direct methods

We mean by implicit direct method any procedure obtained from the RIA by choosing z_m directly from the system (3.1) and using the lower triangular matrix $Z_n^{\mathrm{T}}G_n$ for obtaining the solution of (3.1). We will give the implicit direct methods corresponding to the explicit direct methods reviewed above. We give also the LU-factorization and the QR-factorization.

3.2.1. The implicit Sloboda method

As shown in Theorem 3.1, the explicit Sloboda method is obtained from the RIA with the choice $z_m = a_m$ and $u_m = x_0^{(m)} - x_0$ (i.e. u_m is arbitrary). The associate implicit Sloboda method with this choice is given as follows. Invoking Proposition 2.3, $Z_n^{\mathrm{T}}G_n = T$ is a lower triangular matrix, and with this choice $Z_n^{\mathrm{T}} = A$, then solving $Ax = b$ is equivalent to solve $AG_n y = Ty = b$ with $x = G_n y$. Remark that the implicit Bordering method and the implicit Reinforcement method are obtained from the implicit Sloboda method with the choice $u_m = e_m$. For this choice G_n is an upper unit triangular matrix. Remark also that the implicit Huang method is obtained from the implicit Sloboda method with the choice $u_m = a_m$. For this choice $G_n^{\mathrm{T}}G_n$ is a diagonal matrix [14].

3.2.2. The LU-factorization

It is well known that for any strongly nonsingular matrix A, there exist a lower unit triangular matrix L and an upper triangular matrix U such that $A = LU$. The factors L and U are unique [9, 11, 16]. This factorization of A can be obtained by the auxiliary vectors $g_{j,m}$ used in the RIA, with the choice $z_m = \tilde{a}_m$ and $u_m = e_m$. For this choice and from (5) of Proposition 2.1, G_n becomes an upper unit triangular matrix, and from (1) of Proposition 2.3, $Z_n^{\mathrm{T}}G_n$ becomes a lower triangular matrix. We have $Z_n^{\mathrm{T}}G_n = A^{\mathrm{T}}G_n = T$, then $[Z_n^{\mathrm{T}}G_n]^{\mathrm{T}} = G_n^{\mathrm{T}}A = T^{\mathrm{T}}$ is an upper triangular matrix and G_n^{T} is a lower unit triangular matrix. From the uniqueness of the LU-factorization, we get $G_n^{\mathrm{T}} = L^{-1}$ and $T^{\mathrm{T}} = U$, and solving $Ax = b$ is equivalent to solving $G_n^{\mathrm{T}}Ax = T^{\mathrm{T}}x = G_n^{\mathrm{T}}b$. Remark that if A is symmetric and positive definite the LDL^{T} Choleski factorization of A can be obtained with the choice $G_n^{\mathrm{T}} = L^{-1}$ and $T^{\mathrm{T}} = DL^{\mathrm{T}}$. From this discussion we have the following result.

Theorem 3.3. *If we choose, in* (2.13), $z_m = \tilde{a}_m$ *and* $u_m = e_m$, *then we get the* LU-factorization *of A with* $L = G_n^{\mathrm{T}}$ *and* $U = [Z_n^{\mathrm{T}}G_n]^{\mathrm{T}} = G_n^{\mathrm{T}}A$; *and if A is symmetric and positive definite, then we get the* LDL^{T}-Choleski factorization *of A with* $L = G_n^{\mathrm{T}}$ *and* $DL^{\mathrm{T}} = [Z_n^{\mathrm{T}}G_n]^{\mathrm{T}}$.

3.2.3. The QR-factorization

It is well known that for any nonsingular matrix $A = [\tilde{a}_1, \ldots, \tilde{a}_n]$, there exist an orthogonal matrix $Q = [q_1, \ldots, q_n]$ (i.e. $Q^{\mathrm{T}}Q$ is diagonal) and an upper unit triangular matrix $R = (r_{i,j})$ such that $A = QR$

[9, 11]. This factorization can be obtained by the Gram–Schmidt process [9, 11], which is described as follows:

$$q_1 = \tilde{a}_1,$$

for $i = 2, \ldots, n,$

$$q_i = \tilde{a}_i - \sum_{j=1}^{i-1} \frac{\langle q_j, \tilde{a}_i \rangle}{\langle q_j, q_j \rangle} q_j,$$

end i.

From this process we can see that $r_{i,i} = 1$, $r_{i,j} = \langle q_i, \tilde{a}_j \rangle / \langle q_i, q_i \rangle$ for $j > i$ and $r_{i,j} = 0$ otherwise. This factorization can be obtained by using the auxiliary vectors $g_{j,m}$ used in the RIA with the choice $z_m = q_m$ and $u_m = \tilde{a}_m$. We have the following result.

Theorem 3.4. *If we choose, in* (2.13), $z_m = q_m$ *and* $u_m = \tilde{a}_m$, *then we get the* QR-factorization *of* A *with* $Q = G_n$ *and* $R = [Z_n^{\mathrm{T}} G_n]^{\mathrm{T}}$.

Proof. First let us prove that $g_{m-1,m} = q_m$ for $m = 1, \ldots, n$. We have $g_{0,1} = u_1 = \tilde{a}_1 = q_1$. Assume now that it is true for $m \geqslant 1$, then from (2.25) we have $g_{m,m+1} = (I - Q_m')u_{m+1}$ with Q_m' a diagolal matrix because $z_i = g_{i-1,i}$, and the property follows. From Proposition 2.5 $Z_n^{\mathrm{T}} G_n = G_n^{\mathrm{T}} G_n$ is a diagonal matrix and $Z_n^{\mathrm{T}} U_n = G_n^{\mathrm{T}} A$ is an upper triangular matrix. Then the result follows. □

Remark that from Proposition 2.5 and Remark 2.3 the choice $z_m = u_m = \tilde{a}_m$ gives also the QR-factorization of A. The version given by Theorem 3.4 is called the modified Gram–Schmidt process [2, 20] and it is as follows:

$$g_{0,1} = \tilde{a}_1,$$

for $m = 1, \ldots, n-1,$

$$g_{0,m+1} = \tilde{a}_{m+1},$$

for $i = 1, \ldots, m,$

$$g_{i,m+1} = g_{i-1,m+1} - \frac{\langle g_{i-1,i}, g_{i-1,m+1} \rangle}{\langle g_{i-1,i}, g_{i-1,i} \rangle} g_{i-1,i},$$

end i,

end m.

Invoking Proposition 2.5 and Remark 2.3 we have other equivalent formulations of the Gram–Schmidt process. We will give an important one which is given with the choice $z_m = g_{m-1,m}$, $u_m = \tilde{a}_m$ and by using the process given in the Remark 2.1.

Algorithm 7: The normalized modified Gram–Schmidt process

$$g_{0,1} = \tilde{a}_1 / ||\tilde{a}_m||,$$

for $m = 1, \ldots, n-1,$

Table 1

Method	Choice of x_0	Choice of z_m	Choice of u_m
Bordering	0	a_m	e_m
Reinforcement	b	a_m	e_m
Huang	0	a_m	a_m
Sloboda	$x_0^{(0)}$	a_m	$x_0^{(m)} - x_0^{(0)}$
Unscaled ABS	$\bar{x}_0$	a_m	$H_0^{\mathrm{T}} v_m$
LU-factorization	–	$\tilde{a}_m$	e_m
QR-factorization	–	q_m or $\tilde{a}_m$	$\tilde{a}_m$

$$
\begin{aligned}
&\quad g_{0,m+1} = \tilde{a}_{m+1},\\
&\quad \textit{for } i = 1,\ldots,m,\\
&\qquad g_{i,m+1} = g_{i-1,m+1} - \langle g_{i-1,i}, g_{i-1,m+1}\rangle g_{i-1,i},\\
&\quad \textit{end } i,\\
&\quad g_{m,m+1} = g_{m,m+1}/\|g_{m,m+1}\|,\\
&\textit{end } m.
\end{aligned}
$$

Remark that G_n obtained from this algorithm is an orthonormal matrix (i.e. $G_n^{\mathrm{T}} G_n = I$) and solving $Ax = b$ is equivalent to solve $G_n^{\mathrm{T}} Ax = G_n^{\mathrm{T}} b$ where $G_n^{\mathrm{T}} A$ is an upper triangular matrix. In Table 1 we summarize the results about the identification of the RIA with the various methods discussed in this paper.

We remark that for the bordering method the $n - m$ last components of the iterate vector x_m are zero, and for the reinforcement method the $n - m$ last components of the iterate vector x_m are equal to the $n - m$ last components of b. For this reason these methods are not interesting for a large system.

In Part II [19] of this paper we will give the connection between the RIA and some iterative methods. We will show that this formalism gives us new formulations of these methods, which can be computationally attractive and compete with classical formulations.

Acknowledgements

I am grateful to the Professors C. Brezinski, K. Jbilou and H. Sadok for their helpful comments. I also would like to thank the referees for their helpful comments and valuable suggestions.

References

[1] J. Abaffy and E. Spedicato, *ABS Projection Algorithms: Mathematical Techniques for Linear and Nonlinear Equations* (Halsted Press, Chichester, 1989).

[2] A. Bjorck and C.C. Paige, Loss recapture of orthogonality in the modified Gram–Schmidt algorithms, *SIAM J. Matrix Anal. Appl.* **13** (1992) 176–190.

[3] C. Brezinski, Recursive interpolation, extrapolation and projection, *J. Comput. Appl. Math.* **9** (1983) 369–376.

[4] C. Brezinski, Some determinantal identities in a vector space, with applications, in: H. Werner and H.J. Bunger, Eds., *Padé Approximation and its Applications*, Bad-Honnef, 1983, Lecture Notes in Math., Vol. 1071 (Springer, Berlin, 1984) 1–11.

[5] C. Brezinski, Other manifestations of the Schur complement, *Linear Algebra Appl.* **111** (1988) 231–247.

[6] C. Brezinski, Bordering methods and progressive forms for sequence transformations, *Zastosow Math.* **20** (1990) 435–443.

[7] C. Brezinski, *Biorthogonality and its Applications to Numerical Analysis* (Marcel Dekker, New York, 1992).

[8] C. Brezinski and M. Redivo Zaglia, *Extrapolation Methods, Theory and Practice* (North-Holland, Amsterdam, 1991).

[9] D.K. Faddeev and V.N. Faddeeva, *Computational Methods for Linear Algebra* (Freeman, San Francisco, 1963).

[10] D. Goldfarb, Modification methods for inverting matrices and solving systems of linear algebraic equations, *Math. Comput.* **26** (1972) 829–852.

[11] G.H. Golub and C.F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore, 1983).

[12] H.Y. Huang, A direct method for the general solution of a system of linear equations, *J. Optim. Theory Appl.* **16** (1975) 429–445.

[13] K. Jbilou, Méthodes d' extrapolation et de projection, application aux suites de vecteurs, Thèse Univ. de Lille 1, 1988.

[14] K. Jbilou, Recursive projection and interpolation algorithms. Application for solving linear systems, Publication ANO 303, Univ. de Lille 1, 1993.

[15] A. Lembarki, Méthodes de projection et extensions: étude théorique et pratique, Thèse Univ. de Lille 1, 1984.

[16] A. Messaoudi, Sur la décomposition incomplète de certaines classes de matrices; algorithmes itératifs associés, Thèse Univ. de Franche Comté, 1983.

[17] A. Messaoudi, Some properties of the recursive projection and interpolation algorithms, *IMA J. Numer. Anal.* **15** (1995) 307–318.

[18] A. Messaoudi, Schur complements and variants of the quotient property with applications, Publication ANO 328, Univ. de Lille, submitted.

[19] A. Messaoudi, Recursive interpolation algorithm: a formalism for solving systems of linear equations – II: Iterative Methods, *J. Comput, Appl. Math.*, submitted.

[20] A. Ruhe, Numerical aspects of Gram–Schmidt orthogonalization of vectors, *Linear Algebra Appl.* **52/53** (1983) 591–601.

[21] H. Sadok, Méthodes de projection pour les systèmes linéaires et non linéaires, Thèse Univ. de Lille 1, 1994.

[22] F. Sloboda, A parallel projection method for linear algebraic systems, *Aplikace Mate. Svazek* **23** (1978) 185–198.