7. (Bi)conjugate gradient squared algorithms. In 1984 Sonneveld [65] proposed a
new polynomial acceleration method for solving linear systems of equations which soon proved
to be very successful in practice. It is closely related to the biconjugate gradient algorithm
(BIOMIN), has typically a better or at least similar convergence behavior, and has the often quite
important advantage that the adjoint matrix $A^H$ is not needed. Its $n$th residual polynomial
is the square $\rho_n^2$ of the $n$th residual polynomial $\rho_n$ of the three normalized Lanczos solvers
discussed in Sects. 2 and 3. In view of this connection Sonneveld called his method the conjugate
gradient squared (CGS) algorithm; we call it here more distinctly BIORTHOMIN Squared or,
briefly, BIOMIN$^2$. (Aimed at nonsymmetric problems, this method is in fact related to the BCG
method, not to the generalized CG method.)

For the derivation we start from the recurrences (4.3), choosing $\gamma_n := -\varphi_n$ and substituting
$\omega_n := 1/\varphi_n$:

$$\rho_{n+1} := \rho_n - \omega_n \zeta \sigma_n, \quad \sigma_{n+1} := \rho_{n+1} - \psi_{n+1} \sigma_n.$$  \hfill (7.1)

(In this section we use the sloppy notation $\zeta \sigma_n$ instead of $\zeta \sigma_n(\zeta)$.) Multiplying (7.1) by $\sigma_n$ and
(7.2) by $\rho_{n+1}$ yields

$$\rho_{n+1} \sigma_n = \rho_n \sigma_n - \omega_n \zeta \sigma_n^2, \quad \rho_{n+1} \sigma_{n+1} = \rho_{n+1}^2 - \psi_{n+1} \rho_{n+1} \sigma_n.$$  \hfill (7.3)

Next, squaring both sides of (7.1) and (7.2) and utilizing (7.3) and (7.4), respectively, leads to

$$\rho_{n+1}^2 = \rho_n^2 - 2 \omega_n \zeta \rho_n \sigma_n + \omega_n^2 \zeta^2 \sigma_n^2, \quad \sigma_{n+1}^2 = \rho_{n+1}^2 - \omega_n \zeta (\rho_n \sigma_n + \rho_{n+1} \sigma_n)$$  \hfill (7.5a)

and

$$\sigma_{n+1}^2 = \rho_{n+1}^2 - 2 \psi_{n+1} \rho_{n+1} \sigma_n + \psi_{n+1}^2 \sigma_n^2.$$  \hfill (7.5b)

The point is that Eqs. (7.4), (7.5b), (7.6) and (7.3) are (in this order) a system of recurrence
relations for the four polynomial sequences \{\rho_n \sigma_n\}, \{\rho_n \sigma_{n-1}\}, \{\rho_n^2\}, \{\sigma_n^2\}. From the formulas
(4.41) it follows that the recurrence coefficients can be computed from the values which the
functional $\Phi$ (defined by (4.10)) takes at the polynomials $\zeta \sigma_n^2, \rho_{n+1}^2$ and $\rho_n^2$. (Recall that
$\gamma_n = -\varphi_n$.) Here we need to express these values in terms of the new Krylov space vectors

$$x_n := \rho_n^2(A)x_0, \quad u_n := \rho_n(A)\sigma_n(A)x_0, \quad p_n := \sigma_n^2(A)x_0, \quad q_n := \rho_n(A)\sigma_{n-1}(A)x_0$$  \hfill (7.7)

and their inner products with an additional vector $y_0$, which is now only used for these inner
products and can, e.g., be chosen equal to $x_0$, as in the paper of Sonneveld [65]. Finally, it is
important that one can additionally compute a vector sequence \{z_n\} with the property that
$x_n$ is the residual at $z_n$. One obtains:

Algorithm 9 (BIOMIN$^2$). For solving $Ax = b$ choose an initial approximation $z_0 \in \mathbb{C}^N$
and set $x_0 := u_0 := p_0 := b - Ax_0$, $q_0 := o \in \mathbb{C}^N$, $\psi_0 := 0$. Choose $y_0 \in \mathbb{C}^N$ such that
$\delta_0 := <y_0, x_0 > B \neq 0$. Then compute for $n = 0, 1, \ldots$

$$\varphi_n := <y_0, Ap_n > B / \delta_n.$$  \hfill (7.8a)
if \( \varphi_n = 0 \), set \( \nu := n \) and stop; otherwise compute

\[
(7.8b) \quad \omega_n := \frac{1}{\varphi_n}, \\
(7.8c) \quad q_{n+1} := u_n - Ap_n \omega_n, \\
(7.8d) \quad x_{n+1} := x_n - A(u_n + q_{n+1}) \omega_n, \\
(7.8e) \quad z_{n+1} := z_n + (u_n + q_{n+1}) \omega_n, \\
(7.8f) \quad \delta_{n+1} := \langle y_0, x_{n+1} \rangle > B, \\
(7.8g) \quad \psi_{n+1} := -\delta_{n+1}/\delta_n, \\
(7.8h) \quad u_{n+1} := x_{n+1} - q_{n+1} \psi_{n+1}, \\
(7.8i) \quad p_{n+1} := u_{n+1} - q_{n+1} \psi_{n+1} + p_n \psi_{n+1}^2;
\]

if \( \delta_{n+1} = 0 \), set \( \nu := n + 1 \) and stop; otherwise proceed with the next step.

The recurrences for the vectors \( q_n, u_n, x_n \), and \( p_n \) are direct translations of Eqs. (7.3)-(7.6), and the formulas for \( \varphi_n, \delta_n, \) and \( \psi_n \) are in view of the definition (4.10) of \( \Phi \) identical with (4.41). Finally, the recurrence for \( z_n \) is chosen so that \( x_n = b - Az_n (\forall n) \) by (7.8d), (7.8e) and by induction one gets \( x_{n+1} = x_n - A(z_n - z_{n+1}) = b - Az_{n+1} \).

From our derivation of this algorithm it is clear that the following holds:

**Theorem 7.1.** If \( \text{BIOMIN} \) and \( \text{BIOMIN}^2 \) are started with the same \( z_0 \) and \( y_0 \), the breakdown index \( \nu \), the recurrence coefficients \( \varphi_n, \psi_n \), and the inner products \( \delta_n \) are for both methods the same. The residual polynomials are \( \rho_n \) and \( \rho_n^2 \), respectively.

Although theoretically, if convergence were defined by \( x_n = o \) (exactly), the two algorithms would converge or break down at the same step, it is evident that in practice, where convergence is defined by \( ||x_n|| \leq \varepsilon, \) \( \text{BIOMIN}^2 \) converges faster than \( \text{BIOMIN} \), since \( ||\rho_n^2(A)|| \leq ||\rho_n(A)||^2 < ||\rho_n(A)|| \) if the latter is smaller than 1.

At this point one may ask whether in analogy to the other versions of the (bi)conjugate gradient approach there exist also other versions of the (bi)conjugate gradient squared approach. To derive such methods, which one should naturally call \( \text{BIORES}^2 \) and \( \text{BIODIR}^2 \), we first need separate recurrences for the polynomials \( \rho_n^2 \) and \( \sigma_n^2 \). We start with those for \( \text{BIORES}^2 \). By multiplying (4.4) with \( \rho_n \) and by squaring (4.4) we obtain, respectively,

\[
(7.9) \quad \gamma_n \rho_n \rho_{n+1} = (\zeta - \alpha_n) \rho_n^2 - \beta_n \rho_{n-1}^2 \rho_n, \\
(7.10) \quad \gamma_n^2 \rho_n^2 = (\zeta - \alpha_n)^2 \rho_n^2 - 2(\zeta - \alpha_n) \beta_n \rho_{n-1}^2 \rho_n + \beta_n^2 \rho_{n-1}^2.
\]

where (7.9) has already been used to simplify (7.10). These two relations can be used to generate the two sequences \( \{\rho_{n-1}\rho_n\} \) and \( \{\rho_n^2\} \) recursively. The coefficients \( \alpha_n, \beta_n \) are given by (4.42); the parameters \( \gamma_n \) can either be chosen freely \( (\neq 0) \), if one aims at an unnormalized version of \( \text{BIORES}^2 \), or are given by \( \gamma_n := -\alpha_n - \beta_n \) in the case of normalized \( \text{BIORES}^2 \), since this condition is equivalent to \( \rho_n(0) = 1 \), which implies \( \rho_n^2(0) = 1 \); the latter is the consistency condition for the residual polynomial \( \rho_n^2 \).

Hence, a method for generating

\[
(7.11) \quad x_n := \rho_n^2(A)x_0, \quad t_n := \rho_{n-1}(A)x_0
\]

can be based on (7.9), (7.10) and (4.42):

\[
(7.12) \quad t_{n+1} := [Ax_n - x_n \alpha_n - t_n \beta_n]/\gamma_n,
\]
Multiplying \( \beta_n \) with \( \gamma_n \) yields a recursive formula for \( \beta_{n+1} \) and \( \gamma_{n+1} \) where

\[
x_{n+1} := [A t_{n+1} \gamma_n - A \beta_n - (t_{n+1} \gamma_n - t_n \beta_n) \alpha_n + x_{n-1} \beta_n^2] / \gamma_n,
\]

However, it remains to find a way to compute the sequence \( \{z_n\} \) of approximants with the property that

\[
x_n = b - Az_n,
\]

in the case of normalized BiORES\(^2\), or, more generally,

\[
x_n = b \rho_n^2 - Az_n,
\]

where \( \rho_n := \rho_n(0) \) as before. Assuming that this is possible we conclude from (7.13) and (2.32) that

\[
A z_{n+1} \gamma_n^2 = (b \rho_{n+1}^2 - x_{n+1}) \gamma_n^2
\]

\[
= b(\rho_n^2 \rho_n - \alpha_n \gamma_n \rho_n + \alpha_n \beta_n \rho_n + \beta_n - A(t_{n+1} \gamma_n - t_n \beta_n)) + A \gamma_n - x_{n-1} \beta_n^2
\]

\[
= A z_n - A \gamma_n \beta_n - A s_{n+1} \alpha_n \gamma_n + A s_n \alpha_n \beta_n - A t_{n+1} \gamma_n + A t_n \beta_n,
\]

where

\[
z_n := A^{-1}(b \rho_n^2 - x_n), \quad s_n := A^{-1}(b \rho_{n-1} \rho_n - t_n),
\]

i.e.,

\[
x_n = b \rho_n^2 - Az_n, \quad t_n = b \rho_{n-1} \rho_n - A s_n.
\]

Multiplying (7.16) with \( A^{-1} \) yields a recursive formula for \( z_{n+1} \). Likewise, using (7.13) and (2.32) we get

\[
A s_{n+1} \gamma_n = b \rho_n \rho_{n+1} \gamma_n - t_{n+1} \gamma_n - A x_n + x_{n+1} \beta_n - A \gamma_n
\]

\[
= -b(\rho_n^2 \alpha_n + \rho_{n-1} \rho_n \beta_n) - A x_n + x_{n+1} \beta_n - A \gamma_n
\]

\[
= -A x_n - A s_n \beta_n - A x_n
\]

If we set \( \rho_{-1} := 0, \rho_0 := 1 \), then (7.18) holds for \( x_0 := b - A z_0, t_0 := s_0 := 0 \), and the recurrence can be started with these initial values.

**Algorithm 10 (Unnormalized BiORES\(^2\)).** For solving \( Az = b \) choose an initial approximation \( z_0 \in \mathbb{C} \) and set \( x_0 := b - A z_0, s_0 := t_0 := \alpha \in \mathbb{C}^N, \beta := 0, \rho := 1 \). Choose \( y_0 \in \mathbb{C}^N \) such that \( \delta := < y_0, x_0 > B \neq 0 \). Then compute for \( n = 0, 1, \ldots \), with arbitrary \( \gamma_n \neq 0 \),

\[
(7.20a) \quad \alpha_n := < y_0, A x_n > B / \delta_n,
\]

\[
(7.20b) \quad \beta_n := < y_0, A t_n > B / \delta_n = \gamma_n / \delta_n \quad (if \ n > 0),
\]

\[
(7.20c) \quad \rho_{n+1} := - (\alpha_n \rho_n + \beta_n \rho_{n-1}) / \gamma_n,
\]

\[
(7.20d) \quad t_{n+1} := [A x_n - x_n \alpha_n - t_n \beta_n] / \gamma_n,
\]

\[
(7.20e) \quad x_{n+1} := [A t_{n+1} \gamma_n - A t_n \beta_n - (t_{n+1} \gamma_n - t_n \beta_n) \alpha_n + x_{n-1} \beta_n^2] / \gamma_n,
\]

\[
(7.20f) \quad s_{n+1} := -(x_n \alpha_n + s_n \beta_n + x_n / \gamma_n),
\]

\[
(7.20g) \quad z_{n+1} := (z_n \beta_n^2 - s_n \alpha_n \gamma_n - s_n \alpha_n \beta_n - t_{n+1} \gamma_n + t_n \beta_n) / \gamma_n^2,
\]

\[
(7.20h) \quad \delta_{n+1} := < y_0, x_{n+1} > B.
\]

50
If \( x_{n+1} = 0 \) and \( \rho_{n+1} \neq 0 \), the process terminates and \( z_{n+1}/\rho_{n+1} \) is the solution; if \( x_{n+1} = 0 \) and \( \rho_{n+1} = 0 \), then \( A \) is singular, unless \( z_{n+1} = 0 \) also, which is then a useless result; if \( x_{n+1} \neq 0 \) but \( \delta_{n+1} = 0 \) the algorithm breaks down. In each case we set \( \nu := n + 1 \).

**Algorithm 11 (Normalized Biores\(^2\)).** Modify Algorithm 10 by choosing \( \gamma_n := -\alpha_n - \beta_n \). If \( \gamma_n = 0 \), the algorithm breaks also down, and we set \( \nu := n \).

According to the derivation of these two algorithms, \( \alpha_n, \beta_n \), and \( \delta_n \) computed here are the same as those in Biores. Each step requires two applications of the operator \( A \), i.e., two multiplications of \( A \) with a vector, namely \( Ax_n \) and \( A\delta_{n+1} \); this is comparable to the two matrix-vector multiplications (with \( A \) and \( A^H \)) in BIOMIN, but in many applications, in particular on vector computers, it is an advantage that the multiplication with \( A^H \) is replaced now by one with \( A \). The breakdown conditions are also the same as those of the respective version of Biores. Hence, in view of Theorem 3.5, the following holds:

**Theorem 7.2.** If normalized Biores and normalized Biores\(^2\) are started with the same \( z_0 \) and \( y_0 \), then the breakdown index \( \nu \), the recurrence coefficients \( \alpha_n, \beta_n \) (and thus \( \gamma_n := -\alpha_n - \beta_n \)), and the inner products \( \delta_n \) are the same for both algorithms. The breakdown index is also the same as in BIOMIN and Biores\(^2\). Moreover, if normalized Biores\(^2\) and Biores\(^2\) are started the same way, they produce the same iterates \( z_n \) and thus also the same residuals \( x_n = b - Ax_n \) and the same residual polynomials \( \rho_n^2 \).

Likewise, if unnormalized Biores and unnormalized Biores\(^2\) are started with the same \( z_0 \) and \( y_0 \), and if the same constants \( \gamma_n \) are used in both algorithms, then the breakdown index \( \nu \), the recurrence coefficients \( \alpha_n, \beta_n \), and the inner products \( \delta_n \) are the same for both algorithms. The iterates \( z_n \) and the vectors \( x_n \) are related by (7.14) and (7.15), respectively.

In particular, \( \nu \geq \nu \) holds, and for \( n < \nu \) the iterates produced by unnormalized Biores\(^2\) are except for the scaling the same as those generated by normalized Biores\(^2\).

Let us now turn to Biodir\(^2\). By

\[
\sigma_n' := \sigma_n/\Omega_n, \quad \text{where } \Omega_n := \frac{\Gamma_n'}{\Gamma_n} = \frac{\gamma_0' \gamma_1' \cdots \gamma_{n-1}'}{\gamma_0 \gamma_1 \cdots \gamma_{n-1}} = \frac{\omega_n'}{\omega_n}
\]

(cf. (3.24), (3.27) and Theorem 4.1), we denote the rescaled version of the polynomials \( \sigma_n \) which corresponds to the vectors \( u_n' \) of Biodir\(^2\), as opposed to the vectors \( u_n \) generated by BIOMIN. Recall that by this renormalization it was possible to proceed with Biodir even when \( u_n' = 0 \), in which case also \( \omega_n = 0 \) and thus the definition \( \gamma_n' := \gamma_n := -1/\omega_n \), which is implicitly used in BIOMIN, is not feasible. Note that \( \Omega_n \) is nonvanishing and finite as long as \( \gamma_{n-1} := -\alpha_{n-1} - \beta_{n-1} \neq 0 \), i.e., for \( n \leq \nu' \), cf. Theorem 3.5.

From the analogy between the BO and the BC algorithm it is clear that recurrence formulas for generating \( (\sigma_n')^2 \) along with \( \sigma_n' \sigma_{n+1}' \) differ from (7.9) and (7.10) for generating \( \rho_n^2 \) and \( \rho_n \rho_{n+1} \) only in that \( \alpha_n, \beta_n, \) and \( \gamma_n \) are replaced by \( \alpha_n', \beta_n', \) and \( \gamma_n' \). Hence, recurrence formulas for

\[
p_n := (\sigma_n')^2(A)x_0, \quad w_n := \sigma_n'(A)\sigma_n'(A)x_0
\]

analogous to those for \( x_n \) and \( t_n \) in (7.12) and (7.13) are readily found.

It remains to express \( x_n := \rho_n^2(A)x_0 \) and \( z_n \) (such that \( x_n = b - Ax_n \)) in terms of \( \{p_n\} \) and \( \{w_n\} \). Inserting in (7.5a) \( \rho_n \) according to (7.2), \( \sigma_n \) according to (7.21) and \( \omega_n \) according to \( \omega_n = \omega_n'/\Omega_n \) yields

\[
\rho_{n+1}^2 = \rho_n^2 - \omega_n'z_n[2\psi_n\Omega_n - \sigma_n'\sigma_n' + 2\Omega_n(\sigma_n')^2 + \omega_n'(\sigma_n')^2]
\]

(7.23)
Dropping the factor $\zeta$ in front of the bracket leads to the recurrence formula for the approximants $z_n$. In view of $\gamma_n = -\varphi_m = -1/\omega_m$ and (3.27) one sees further that

$$\Omega_n = \frac{\Gamma_{n-1}^2 \gamma_{n-1}}{\Gamma_{n-1} \gamma_{n-1}} = \frac{\omega_{m-1}^2 \gamma_{n-1}}{\omega_{m-1} \gamma_{n-1}} = -\omega_{m-1}^2 \gamma_{n-1}.$$  

Moreover, according to (7.21), (3.1a), (3.20) and (3.24), $\omega'_n = \omega_n \Omega_n = \omega_n / \varphi_n = \delta_n / (\delta_n \Omega_n)$. Altogether, we get the following first version of BIOMIN$^2$:

**Algorithm 12 (BIODIR$^2$).** For solving $Ax = b$ choose an initial approximation $z_0 \in \mathbb{C}$ and set $p_0 := x_0 := b - Az_0$, $w_0 := \alpha \in \mathbb{C}^N$, $\beta_0 := 0$, $\Omega_0 := 1$, $\Omega_{-1} := \psi_0 := 0$. Choose $v_0 \in \mathbb{C}^N$ such that $\delta_0 := < v_0, x_0 >_B \neq 0$ and $\delta_0 := < A^T v_0, x_0 >_B \neq 0$. Then compute for $n = 0, 1, \ldots$, with arbitrary $\gamma'_n \neq 0$,

(7.24a) \hspace{1cm} \omega'_n := \delta_n / (\delta'_n \Omega_n), \\
(7.24b) \hspace{1cm} \psi_n := -\delta_n / \delta_{n-1} \quad (if \ n > 0), \\
(7.24c) \hspace{1cm} \alpha_n := < A^T v_0, Ap_n >_B / \delta'_n, \\
(7.24d) \hspace{1cm} \beta'_n := < A^T v_0, Ap_n >_B \gamma'_n / \delta'_n = \gamma'_n - \delta'_n / \delta'_n = (if \ n > 0), \\
(7.24e) \hspace{1cm} w_{n+1} := \{ Ap_n - p_n \alpha'_n - w_n \beta'_n \gamma'_n, \\
(7.24f) \hspace{1cm} h_n := [w_n (2 \psi_n \Omega_n) + p_n (2 \Omega_n) + Ap_n \omega'_n] \omega'_n, \\
(7.24g) \hspace{1cm} z_{n+1} := z_n + h_n, \\
(7.24h) \hspace{1cm} x_{n+1} := x_n - [Aw_n (2 \psi_n \Omega_n - \beta'_n) + Ap_n (2 \Omega_n + \alpha'_n) + A p_{n+1} \gamma'_n \omega'_n, \\
(7.24i) \hspace{1cm} \delta_{n+1} := < v_0, x_{n+1} >_B = \delta_n - (2 \Omega_n + \alpha'_n \omega'_n) \delta'_n, \\
(7.24j) \hspace{1cm} p_{n+1} := [Aw_n + \gamma'_n - Aw_n + \beta'_n + (w_{n+1} \gamma'_n - w_n \beta'_n \alpha'_n + p_{n-1} \beta'_n^2) / (\gamma'_n)^2, \\
(7.24k) \hspace{1cm} \delta'_n := < A^T v_0, p_{n+1} >_B, \\
(7.24l) \hspace{1cm} \Omega_{n+1} := -\gamma'_n \omega'_n. \\

If $x_{n+1} = 0$, the process terminates and $z_{n+1}$ is the solution; if $x_{n+1} \neq 0$ but $\delta'_{n+1} = 0$ or $\delta_n = 0$, the algorithm breaks down. In each case we set $\nu := n + 1$.

It is in particular possible to choose $\gamma'_n := -1 / \omega'_n$ (n = 0, \ldots, $\nu - 1$), so that $\Omega_n = 1$ (n = 0, \ldots, $\nu$).

These formulas are obtained from the foregoing as follows: (7.24a) and (7.24l) have just been derived; (7.24b) follows from (3.8f) or (4.41c) since $\gamma_n = -\varphi_n$ here; (7.24c), (7.24d), (7.24k) follow from (4.43); the first part of (7.24i) corresponds to (4.42c), the second part of it can be derived from (7.23) by applying $\Phi$ and making use of the orthogonality of $\{ \sigma'_n \}$ with respect to $\Phi_1$; (7.24e) is the analogue of (7.9) for $w_n := \sigma'_n (A) \sigma'_n (A) x_0$; likewise, (7.24j) is the analogue of (7.10) for $p_n := (\sigma'_n)^2 (A) x_0$; (7.24f) and (7.24h) implement (7.23); the second part of (7.24h) is obtained by inserting $Ap_n$ according to (7.24e); finally, (7.24g) is linked to (7.24h) by $x_n = b - Az_n$.

From Theorem 3.5 it can be seen that Algorithm 12 has the same breakdown behavior as BIOMIN$^2$. (Of course, this is already indicated by our choice of $\nu$ as breakdown index.) If one aims at the more general applicability of BIODIR, where only $\delta_n = 0$ but not $\delta'_{n+1} = 0$ cause a breakdown, then one has to avoid the occurence of $\psi_n$ and has to choose $\gamma'_n \neq -1 / \omega'_n$ at least whenever $\omega'_n = 0$ (or, in practice, close to 0). The reduced applicability of BIODIR$^2$ is caused by the fact that (7.2) and, hence, (7.23) and (7.24f)-(7.24h) do not hold when $\omega'_n = 0$.

We start again from (7.5a), modifying it by using $\omega'_n$ from BIODIR instead of $\omega_n$ from BIOMIN and replacing $\sigma_n$ accordingly by $\sigma'_n := \sigma_n \omega_n / \omega'_n$ as before, but now we do not replace
\( \rho_n \) according to (7.2):

\[
\rho_{n+1}^2 = \rho_n^2 - 2\psi_n \omega_n \sigma_n' + (\omega_n')^2 (\sigma_n')^2.
\]

By multiplying (7.1) with \( \gamma_n \sigma_{n+1} \), by making use of the recurrence formula for \( \sigma_n' \) (i.e., (4.5) with \( \gamma_n \) replaced by \( \gamma_n' \) and \( \sigma_n \) replaced by \( \sigma_n' \)), and by inserting (7.1) we get additionally:

\[
\begin{align*}
\gamma_n \rho_{n+1} \sigma_{n+1}' &= \gamma_n \rho_n \sigma_n' - \gamma_n' \omega_n \sigma_n' \sigma_{n+1}' \\
&= (\omega_n')^2 (\sigma_n')^2 - \gamma_n' \omega_n \sigma_n' \sigma_{n+1}',
\end{align*}
\]

which allows us to “update” \( \rho_n \sigma_n' \). Setting

\[
\begin{align*}
\omega_n' &= (\omega_n')^2 (\sigma_n')^2 = \rho_n (A) \sigma_n' (A) x_0
\end{align*}
\]

and noting that (3.22a) translates now into \( \omega_n' := \psi_n, u_n > B / \delta_n' \), we get a second version of BioDr:

**Algorithm 13 (Biodir\(^2\)).** For solving \( Ax = b \) choose an initial approximation \( z_0 \in \mathbb{C} \) and set \( p_0 := u_0 := x_0 := b - Ax_0, w_0 := 0 \in \mathbb{C}^N, \beta_0' := 0 \). Choose \( v_0' \in \mathbb{C}^N \) such that \( \delta_0' := \psi_n > B / \delta_n' \). Then compute for \( n = 0, 1, \ldots \), with arbitrary \( \gamma_n' \neq 0 \),

\[
\begin{align*}
\omega_n' &= \psi_n, u_n > B / \delta_n', \\
\alpha_n' &= (\omega_n')^2 (\sigma_n')^2, \\
\beta_n' &= (\omega_n')^2 (\sigma_n')^2, \\
w_{n+1} &= (Ap_n - p_n \alpha_n - w_m \beta_n') / \gamma_n', \\
h_n' &= u_n (2\omega_n') - Ap_n (\omega_n')^2, \\
z_{n+1} &= z_n + h_n', \\
x_{n+1} &= x_n - Ah\]

\[
\begin{align*}
\rho_{n+1} &= x_n - [Au_n (2\omega_n') + [Aw_n \beta_n' + Ap_n \alpha_n + Az_m + \gamma_n'] (\omega_n')^2, \\
p_{n+1} &= [Az_{n+1} \gamma_n' - Aw_n \beta_n' - (w_m + \gamma_n') (\omega_n')^2, \\
u_{n+1} &= [Au_n - u_n \alpha_n - w_m \beta_n' + Az_m + \gamma_n'] (\omega_n')^2, \\
\delta_{n+1} &= (\omega_n')^2 (\sigma_n')^2 > B .
\end{align*}
\]

If \( x_{n+1} = \alpha \), the process terminates and \( z_{n+1} = \) the solution; if \( x_{n+1} \neq \alpha \), but \( \delta_{n+1} = 0 \), the algorithm breaks down. In both cases we set \( n := n + 1 \).

As in BioDr \( \rho_{n+1} > 0 \) is here the only breakdown condition, hence this algorithm is more generally applicable than BioMin\(^2\) and BioDr\(^2\). However, in each step three applications of the operator \( A \) are required now, namely \( Ap_n, Au_n \) and \( Aw_{n+1} \), while BioMin\(^2\) and BioDr\(^2\) need only two, \( Ap_n \) and \( A(u_n + q_{n+1}) \), and \( Ap_n \) and \( Aw_{n+1} \), respectively.

Fortunately, it is easy to switch between BioDr\(^2\) and BioDr\(^2\). Therefore, one can work with the less costly BioDr\(^2\) as long as \( \gamma_n \) remains bounded, i.e., \( |\delta_{n-1}| > \epsilon \). If this condition is no longer fulfilled, we recall from (7.2) and (7.21) that

\[
(\sigma_{m+1}')^2 = \sigma_{m+1}' \rho_{m+1} - \psi_{m+1} \Omega_m \sigma_m' \sigma_{m+1}',
\]

which after an index shift translates into

\[
u_m = p_m + w_m \psi_m \Omega_m - 1 ,
\]
and allows us to compute the vectors $u_{n-1}$ and $u_n$, which are needed to proceed with $\text{BIODIR}^2$. Once $|\delta_n \omega_n^* \Omega_n| = |\gamma_n^{*} \theta_n \omega_n^* \omega_n^*| > \epsilon$ there, it follows that $|\delta_n| = |\delta_n \Omega_n \omega_n^*| > \epsilon$, and one can switch back to $\text{BIODIR}^2$ in the next step.

Similarly, one can switch between $\text{BIOMIN}^2$ and $\text{BIODIR}^2$. In one direction, from $\text{BIODIR}^2$ to $\text{BIOMIN}^2$, one has just to scale down $p_n$ and $u_n$. For the other direction, $w_n$ is needs to be computed, which is possible by solving (7.30) for $w_m$. 