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 nth residual polynomial is the square ρ_n^2 of the nth residual polynomial ρ_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 $\omega_n := 1/\varphi_n$:

$$\rho_{n+1} := \rho_n - \omega_n \zeta \sigma_n \,, \tag{7.1}$$

$$\sigma_{n+1} := \rho_{n+1} - \psi_{n+1} \sigma_n \,. \tag{7.2}$$

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

$$\rho_{n+1}\sigma_n = \rho_n\sigma_n - \omega_n \zeta \sigma_n^2 \,, \tag{7.3}$$

$$\rho_{n+1}\sigma_{n+1} = \rho_{n+1}^2 - \psi_{n+1}\rho_{n+1}\sigma_n \,. \tag{7.4}$$

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

$$\rho_{n+1}^2 = \rho_n^2 - 2\omega_n\zeta\rho_n\sigma_n + \omega_n^2\zeta^2\sigma_n^2 \tag{7.5a}$$

$$= \rho_n^2 - \omega_n\zeta(\rho_n\sigma_n + \rho_{n+1}\sigma_n) \tag{7.5b}$$

and

$$\begin{aligned} \sigma_{n+1}^2 &= \rho_{n+1}^2 - 2\psi_{n+1}\rho_{n+1}\sigma_n + \psi_{n+1}^2\sigma_n^2 \\ &= \rho_{n+1}\sigma_{n+1} - \psi_{n+1}\rho_{n+1}\sigma_n + \psi_{n+1}^2\sigma_n^2 \,. \end{aligned} \tag{7.6}$$

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 Φ (defined by (4.10)) takes at the polynomials $\zeta\sigma_n^2$, ρ_{n+1}^2 and ρ_n^2. (Recall that $\gamma_n = -\varphi_n$.) Here we need to express these values in terms of the new Krylov space vectors

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

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 $Az = b$ choose an initial approximation $z_0 \in \mathbf{C}^N$ and set $x_0 := u_0 := p_0 := b - Az_0$, $q_0 := o \in \mathbf{C}^N$, $\psi_0 := 0$. Choose $y_0 \in \mathbf{C}^N$ such that $\delta_0 := < y_0, x_0 >_B \neq 0$. Then compute for $n = 0, 1, \ldots$*

$$\varphi_n := < y_0, Ap_n >_B / \delta_n \,; \tag{7.8a}$$

if $\varphi_n = 0$, *set* $\dot\nu := n$ *and stop; otherwise compute*

$$\omega_n := 1/\varphi_n\,, \tag{7.8b}$$

$$q_{n+1} := u_n - Ap_n\omega_n\,, \tag{7.8c}$$

$$x_{n+1} := x_n - A(u_n + q_{n+1})\,\omega_n\,, \tag{7.8d}$$

$$z_{n+1} := z_n + (u_n + q_{n+1})\,\omega_n\,, \tag{7.8e}$$

$$\delta_{n+1} := < y_0, x_{n+1} >_B\,, \tag{7.8f}$$

$$\psi_{n+1} := -\delta_{n+1}/\delta_n\,, \tag{7.8g}$$

$$u_{n+1} := x_{n+1} - q_{n+1}\psi_{n+1}\,, \tag{7.8h}$$

$$p_{n+1} := u_{n+1} - q_{n+1}\psi_{n+1} + p_n\psi_{n+1}^2\,; \tag{7.8i}$$

if $\delta_{n+1} = 0$, *set* $\dot\nu := n+1$ *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 φ_n, δ_{n+1}, and ψ_{n+1} are in view of the definition (4.10) of Φ 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* BIOMIN *and* BIOMIN^2 *are started with the same* z_0 *and* y_0, *the breakdown index* $\dot\nu$, *the recurrence coefficients* φ_n, ψ_n, *and the inner products* δ_n *are for both methods the same. The residual polynomials are* ρ_n *and* ρ_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 $||x_n|| \le \varepsilon$, BIOMIN^2 converges faster than BIOMIN, since $||\rho_n^2(A)|| \le ||\rho_n(A)||^2 < ||\rho_n(A)||$ 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 BIORES^2 and BIODIR^2, we first need separate recurrences for the polynomials ρ_n^2 and σ_n^2. We start with those for BIORES^2: By multiplying (4.4) with ρ_n and by squaring (4.4) we obtain, respectively,

$$\gamma_n\rho_n\rho_{n+1} = (\zeta - \alpha_n)\rho_n^2 - \beta_n\rho_{n-1}\rho_n\,, \tag{7.9}$$

$$\gamma_n^2\rho_{n+1}^2 = (\zeta - \alpha_n)^2\rho_n^2 - 2(\zeta - \alpha_n)\beta_n\rho_{n-1}\rho_n + \beta_n^2\rho_{n-1}^2$$

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

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 α_n, β_n are given by (4.42); the parameters γ_n can either be chosen freely ($\neq 0$), if one aims at an unnormalized version of BIORES^2, or are given by $\gamma_n := -\alpha_n - \beta_n$ in the case of normalized 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 ρ_n^2.

Hence, a method for generating

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

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

$$t_{n+1} := [Ax_n - x_n\alpha_n - t_n\beta_n]/\gamma_n\,, \tag{7.12}$$

$$x_{n+1} := [At_{n+1}\gamma_n - At_n\beta_n - (t_{n+1}\gamma_n - t_n\beta_n)\alpha_n + x_{n-1}\beta_n^2]/\gamma_n^2\,, \tag{7.13}$$

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

$$x_n = b - Az_n \tag{7.14}$$

in the case of normalized BIORES^2, or, more generally,

$$x_n = b\dot\rho_n^2 - Az_n\,, \tag{7.15}$$

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

$$\begin{aligned} Az_{n+1}\gamma_n^2 &= (b\dot\rho_{n+1}^2 - x_{n+1})\gamma_n^2 \\ &= b(\beta_n^2\dot\rho_{n-1}^2 - \alpha_n\gamma_n\dot\rho_n\dot\rho_{n+1} + \alpha_n\beta_n\dot\rho_n\dot\rho_{n-1}) \\ &\quad -A(t_{n+1}\gamma_n - t_n\beta_n) + (t_{n+1}\gamma_n - t_n\beta_n)\alpha_n - x_{n-1}\beta_n^2 \\ &= Az_{n-1}\beta_n^2 - As_{n+1}\alpha_n\gamma_n + As_n\alpha_n\beta_n - At_{n+1}\gamma_n + At_n\beta_n\,, \end{aligned} \tag{7.16}$$

where

$$z_n := A^{-1}(b\dot\rho_n^2 - x_n), \quad s_n := A^{-1}(b\dot\rho_{n-1}\dot\rho_n - t_n)\,, \tag{7.17}$$

i.e.,

$$x_n = b\dot\rho_n^2 - Az_n, \quad t_n = b\dot\rho_{n-1}\dot\rho_n - As_n\,. \tag{7.18}$$

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

$$\begin{aligned} As_{n+1}\gamma_n &= b\dot\rho_n\dot\rho_{n+1}\gamma_n - t_{n+1}\gamma_n \\ &= -b(\dot\rho_n^2\alpha_n + \dot\rho_{n-1}\dot\rho_n\beta_n) - Ax_n + x_n\alpha_n + t_n\beta_n \\ &= -Az_n\alpha_n - As_n\beta_n - Ax_n \end{aligned} \tag{7.19}$$

If we set $\dot\rho_{-1} := 0$, $\dot\rho_0 := 1$, then (7.18) holds for $x_0 := b - Az_0$, $t_0 := s_0 := 0$, and the recurrence can be started with these initial values.

ALGORITHM 10 (UNNORMALIZED BIORES^2). *For solving $Az = b$ choose an initial approximation $z_0 \in \mathbf{C}$ and set $x_0 := b - Az_0$, $s_0 := t_0 := o \in \mathbf{C}^N$, $\beta_0 := 0$, $\dot\rho := 1$. Choose $y_0 \in \mathbf{C}^N$ such that $\delta_0 := < y_0, x_0 >_B \neq 0$. Then compute for $n = 0, 1, \ldots$, with arbitrary $\gamma_n \neq 0$,*

$$\alpha_n := < y_0, Ax_n >_B /\delta_n\,, \tag{7.20a}$$

$$\beta_n := < y_0, At_n >_B /\delta_{n-1} = \gamma_{n-1}\delta_n/\delta_{n-1} \quad (\textit{if } n > 0), \tag{7.20b}$$

$$\dot\rho_{n+1} := -(\alpha_n\dot\rho_n + \beta_n\dot\rho_{n-1})/\gamma_n\,, \tag{7.20c}$$

$$t_{n+1} := [Ax_n - x_n\alpha_n - t_n\beta_n]/\gamma_n\,, \tag{7.20d}$$

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

$$s_{n+1} := -(z_n\alpha_n + s_n\beta_n + x_n)/\gamma_n\,, \tag{7.20f}$$

$$z_{n+1} := (z_{n-1}\beta_n^2 - s_{n+1}\alpha_n\gamma_n - s_n\alpha_n\beta_n - t_{n+1}\gamma_n + t_n\beta_n)/\gamma_n^2\,, \tag{7.20g}$$

$$\delta_{n+1} := < y_0, x_{n+1} >_B\,. \tag{7.20h}$$

If $x_{n+1} = o$ and $\dot\rho_{n+1} \neq 0$, the process terminates and $z_{n+1}/\dot\rho_{n+1}$ is the solution; if $x_{n+1} = o$ and $\dot\rho_{n+1} = 0$, then A is singular, unless $z_{n+1} = o$ also, which is then a useless result; if $x_{n+1} \neq o$ 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 $\dot\nu := n$.*

According to the derivation of these two algorithms, α_n, β_n, and δ_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 At_{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 $\dot\nu$, the recurrence coefficients α_n, β_n (and thus $\gamma_n := -\alpha_n - \beta_n$), and the inner products δ_n are the same for both algorithms. The breakdown index is also the same as in BIOMIN and BIOMIN^2. Moreover, if normalized BIORES^2 and BIOMIN^2 are started the same way, they produce the same iterates z_n and thus also the same residuals $x_n = b - Az_n$ and the same residual polynomials ρ_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 γ_n are used in both algorithms, then the breakdown index ν, the recurrence coefficients α_n, β_n, and the inner products δ_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, $\nu \geq \dot\nu$ holds, and for $n < \dot\nu$ 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

$$\sigma'_n := \sigma_n/\Omega_n\,, \quad \text{where } \Omega_n := \frac{\Gamma'_n}{\Gamma_n} = \frac{\gamma'_0\gamma'_1\cdots\gamma'_{n-1}}{\gamma_0\gamma_1\cdots\gamma_{n-1}} = \frac{\omega'_n}{\omega_n} \tag{7.21}$$

(cf. (3.24), (3.27) and Theorem 4.1), we denote the rescaled version of the polynomials σ_n which corresponds to the vectors u'_n of BIODIR, as opposed to the vectors u_n generated by BIOMIN. Recall that by this renormalization it was possible to proceed with BIODIR even when $\omega'_n = 0$, 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 Ω_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 $(\sigma'_{n+1})^2$ along with $\sigma'_n\sigma'_{n+1}$ differ from (7.9) and (7.10) for generating ρ_{n+1}^2 and $\rho_n\rho_{n+1}$ only in that α_n, β_n and γ_n are replaced by α'_n, β'_n and γ'_n. Hence, recurrence formulas for

$$p_n := (\sigma'_n)^2(A)x_0, \quad w_n := \sigma'_{n-1}(A)\sigma'_n(A)x_0 \tag{7.22}$$

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 $x_n = b - Az_n$) in terms of $\{p_n\}$ and $\{w_n\}$. Inserting in (7.5a) ρ_n according to (7.2), σ_n according to (7.21) and ω_n according to $\omega_n = \omega'_n/\Omega_n$ yields

$$\rho_{n+1}^2 = \rho_n^2 - \omega'_n\zeta[2\psi_n\Omega_{n-1}\sigma'_{n-1}\sigma'_n + 2\Omega_n(\sigma'_n)^2 + \omega'_n\zeta(\sigma'_n)^2]\,. \tag{7.23}$$

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

$$\Omega_n = \frac{\Gamma'_{n-1}\gamma'_{n-1}}{\Gamma_{n-1}\gamma_{n-1}} = \frac{\omega'_{n-1}\gamma'_{n-1}}{\omega_{n-1}\gamma_{n-1}} = -\omega'_{n-1}\gamma'_{n-1}\,.$$

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* BIODIR^2:

ALGORITHM 12 (BIODIR$^2_1$). *For solving $Az = b$ choose an initial approximation $z_0 \in \mathbf{C}$ and set $p_0 := x_0 := b - Az_0$, $w_0 := o \in \mathbf{C}^N$, $\beta'_0 := 0$, $\Omega_0 := 1$, $\Omega_{-1} := \psi_0 := 0$. Choose $v_0 \in \mathbf{C}^N$ such that $\delta_0 := < v_0, x_0 >_B \neq 0$ and $\delta'_0 := < A^H v_0, x_0 >_B \neq 0$. Then compute for $n = 0, 1, \ldots$, with arbitrary $\gamma'_n \neq 0$,*

$$\omega'_n := \delta_n/(\delta'_n\Omega_n)\,, \tag{7.24a}$$

$$\psi_n := -\delta_n/\delta_{n-1} \quad (\textit{if } n > 0)\,, \tag{7.24b}$$

$$\alpha'_n := < A^H v_0, Ap_n >_B /\delta'_n\,, \tag{7.24c}$$

$$\beta'_n := < A^H v_0, p_n >_B \gamma'_{n-1}/\delta'_{n-1} = \gamma'_{n-1}\delta'_n/\delta'_{n-1} \quad (\textit{if } n > 0)\,, \tag{7.24d}$$

$$w_{n+1} := [Ap_n - p_n\alpha'_n - w_n\beta'_n]/\gamma'_n\,, \tag{7.24e}$$

$$h_n := [w_n(2\psi_n\Omega_{n-1}) + p_n(2\Omega_n) + Ap_n\omega'_n]\,\omega'_n\,, \tag{7.24f}$$

$$z_{n+1} := z_n + h_n\,, \tag{7.24g}$$

$$\begin{aligned} x_{n+1} &:= x_n - Ah_n \\ &= x_n - [Aw_n(2\psi_n\Omega_{n-1} + \beta'_n) + Ap_n(2\Omega_n + \alpha'_n) + Aw_{n+1}\gamma'_n]\,\omega'_n\,, \end{aligned} \tag{7.24h}$$

$$\delta_{n+1} := < v_0, x_{n+1} >_B = \delta_n - (2\Omega_n + \alpha'_n\omega'_n)\,\delta'_n\omega'_n\,, \tag{7.24i}$$

$$p_{n+1} := [Aw_{n+1}\gamma'_n - Aw_n\beta'_n - (w_{n+1}\gamma'_n - w_n\beta'_n)\alpha'_n + p_{n-1}(\beta'_n)^2]/(\gamma'_n)^2\,, \tag{7.24j}$$

$$\delta'_{n+1} := < A^H v_0, p_{n+1} >_B\,, \tag{7.24k}$$

$$\Omega_{n+1} := -\gamma'_n\omega'_n\,. \tag{7.24l}$$

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

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

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 Φ and making use of the orthogonality of $\{\sigma'_m\}$ with respect to Φ_1; (7.24e) is the analogue of (7.9) for $w_n := \sigma'_{n-1}(A)\sigma'_n(A)x_0$; 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 $\dot\nu$ 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 ψ_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$^2_1$ 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 ω'_n from BIODIR instead of ω_n from BIOMIN and replacing σ_n accordingly by $\sigma'_n := \sigma_n\omega_n/\omega'_n$ as before, but now we do not replace

ρ_n according to (7.2):

$$\rho_{n+1}^2 := \rho_n^2 - 2\omega_n'\zeta\rho_n\sigma_n' + (\omega_n')^2\zeta^2(\sigma_n')^2 \,. \tag{7.25}$$

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

$$\begin{aligned}
\gamma_n'\rho_{n+1}\sigma_{n+1}' &= \gamma_n'\rho_n\sigma_{n+1}' - \gamma_n'\omega_n'\zeta\sigma_n'\sigma_{n+1}' \\
&= \zeta\rho_n\sigma_n' - \alpha_n'\rho_n\sigma_n' - \beta_n'\rho_n\sigma_{n-1}' - \gamma_n'\omega_n'\zeta\sigma_n'\sigma_{n+1}' \qquad (7.26a)\\
&= \zeta\rho_n\sigma_n' - \alpha_n'\rho_n\sigma_n' - \beta_n'\rho_{n-1}\sigma_{n-1}' + \beta_n'\omega_{n-1}'\zeta(\sigma_{n-1}')^2 - \gamma_n'\omega_n'\zeta\sigma_n'\sigma_{n+1}' \,, \qquad (7.26b)
\end{aligned}$$

which allows us to "update" $\rho_n\sigma_n'$. Setting

$$u_n := \rho_n(A)\sigma_n'(A)x_0 \tag{7.27}$$

and noting that (3.22a) translates now into $\omega_n' := < v_0', u_n >_B / \delta_n'$, we get a *second version of* BIODIR:

ALGORITHM 13 (BIODIR$_2^2$). *For solving $Az = b$ choose an initial approximation $z_0 \in \mathbf{C}$ and set $p_0 := u_0 := x_0 := b - Az_0$, $w_0 := o \in \mathbf{C}^N$, $\beta_0' := 0$. Choose $v_0' \in \mathbf{C}^N$ such that $\delta_0' := < A^H v_0, p_0 >_B \neq 0$. Then compute for $n = 0, 1, \ldots$, with arbitrary $\gamma_n' \neq 0$,*

$$\begin{aligned}
\omega_n' &:= < v_0', u_n >_B / \delta_n' \,, && (7.28a)\\
\alpha_n' &:= < A^H v_0', Ap_n >_B / \delta_n' \,, && (7.28b)\\
\beta_n' &:= < A^H v_0', p_n >_B \gamma_{n-1}'/\delta_{n-1}' = \gamma_{n-1}'\delta_n'/\delta_{n-1}' \quad (\text{if } n > 0) \,, && (7.28c)\\
w_{n+1} &:= [Ap_n - p_n\alpha_n' - w_n\beta_n']/\gamma_n' \,, && (7.28d)\\
h_n &:= u_n(2\omega_n') - Ap_n(\omega_n')^2 \,, && (7.28e)\\
z_{n+1} &:= z_n + h_n \,, && (7.28f)\\
x_{n+1} &:= x_n - Ah_n \\
&= x_n - [Au_n(2\omega_n') + [Aw_n\beta_n' + Ap_n\alpha_n' + Aw_{n+1}\gamma_n'](\omega_n')^2 \,, && (7.28g)\\
p_{n+1} &:= [Aw_{n+1}\gamma_n' - Aw_n\beta_n' - (w_{n+1}\gamma_n' - w_n\beta_n')\alpha_n' + p_{n-1}(\beta_n')^2]/(\gamma_n')^2 \,, && (7.28h)\\
u_{n+1} &:= [Au_n - u_n\alpha_n' - u_{n-1}\beta_n' + Ap_{n-1}\beta_n'\omega_{n-1}']/\gamma_n' - Aw_{n+1}\omega_n' \,, && (7.28i)\\
\delta_{n+1}' &:= < A^H v_0', p_{n+1} >_B \,. && (7.28j)
\end{aligned}$$

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

As in BIODIR $\delta_{n+1}' = 0$ is here the only breakdown condition, hence this algorithm is more generally applicable than BIOMIN^2 and BIODIR$_1^2$. 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 BIODIR$_1^2$ 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 BIODIR$_1^2$ and BIODIR$_2^2$. Therefore, one can work with the less costly BIODIR$_1^2$as long as ψ_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

$$(\sigma_{m+1}')^2 = \sigma_{m+1}'\rho_{m+1} - \psi_{m+1}\Omega_m\sigma_m'\sigma_{m+1}' \,, \tag{7.29}$$

which after an index shift translates into

$$u_m = p_m + w_m\psi_m\Omega n - 1 \,, \tag{7.30}$$

and allows us to compute the vectors u_{n-1} and u_n, which are needed to proceed with BIODIR$_2^2$. Once $|\delta'_n\omega'_n\Omega_n| = |\gamma'_{n-1}\delta'_n\omega'_{n-1}\omega'_n| > \epsilon$ there, it follows that $|\delta_n| = |\delta'_n\Omega_n\omega'_n| > \epsilon$, and one can switch back to BIODIR$_1^2$ in the next step.

Similarly, one can switch between BIOMIN^2 and BIODIR$_2^2$. In one direction, from BIODIR$_2^2$to BIOMIN^2, one has just to scale down p_n and u_n. For the other direction, w_n is needs to be computet, which is possible by solving (7.30) for w_m.