Electronic Transactions on Numerical Analysis.
Volume 36, pp. 126-148, 2010.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

# IDR EXPLAINED*

MARTIN H. GUTKNECHT†

*Dedicated to Richard S. Varga on the occasion of his 80th birthday.*

**Abstract.** The Induced Dimension Reduction (IDR) method is a Krylov space method for solving linear systems that was developed by Peter Sonneveld around 1979. It was noticed by only a few people, and mainly as the forerunner of Bi-CGSTAB, which was introduced a decade later. In 2007, Sonneveld and van Gijzen reconsidered IDR and generalized it to IDR$(s)$, claiming that IDR$(1) \approx$ IDR is equally fast but preferable to the closely related Bi-CGSTAB, and that IDR$(s)$ with $s > 1$ may be much faster than Bi-CGSTAB. It also turned out that when $s > 1$, IDR$(s)$ is related to ML$(s)$BiCGSTAB of Yeung and Chan, and that there is quite some flexibility in the IDR approach. This approach differs completely from traditional approaches to Krylov space methods, and therefore it requires an extra effort to get familiar with it and to understand the connections as well as the differences to better-known Krylov space methods. This expository paper aims to provide some help in this and to make the method understandable even to non-experts. After presenting the history of IDR and related methods, we summarize some of the basic facts on Krylov space methods. Then we present the original IDR$(s)$ in detail and put it into perspective with other methods. Specifically, we analyze the differences between the IDR method published in 1980, IDR$(1)$, and Bi-CGSTAB. At the end of the paper, we discuss a recently proposed ingenious variant of IDR$(s)$ whose residuals fulfill extra orthogonality conditions. There we dwell on details that have been left out in the publications of van Gijzen and Sonneveld.

**Key words.** Krylov space method, iterative method, induced dimension reduction, IDR, CGS, Bi-CGSTAB, ML$(k)$BiCGSTAB, large nonsymmetric linear system

**1. History.** The Induced Dimension Reduction (IDR) method was introduced by Wesseling and Sonneveld from Delft University at a symposium of the International Union of Theoretical and Applied Mechanics in September 1979. In the proceedings it is covered on just three and a half pages of a twenty-page paper [40], and it is explicitly attributed to the second author. It was labeled as a Lanczos-type method for nonsymmetric linear systems which does not require the transpose of the matrix. The term Lanczos-type method meant that the new method was related to the biconjugate gradient (BiCG) method of Lanczos [17], which had been revived and reformulated by Fletcher [4] four years before. Up to that time there had been little interest in Lanczos's approach, despite the fact that it was very closely related to the widely used conjugate gradient method [16, 21]. Popular alternative Krylov space solvers for nonsymmetric systems were methods like Vinsome's OrthoMin[1] [39] (now often referred to as GCR) and its variants OrthoDir and OrthoRes [42], as well as similar methods introduced by Axelsson [2]. GMRes [22] was still five years away. Also popular were parameter-dependent Krylov space methods, such as Chebyshev iteration, and parameter-dependent iterative methods based on matrix splitting, such as SOR.

The IDR method received hardly any attention, probably because it was neither presented at a conference of the core numerical analysis community nor published in a widely read journal. Moreover, Sonneveld's approach to designing a Krylov space solver was very unusual and, even for experts, hard to fully understand. Although the method was defined clearly and uniquely under certain regularity conditions in [40], some of the details, and in particular the proof of the connection to BiCG, were left for publication elsewhere. The basic result on this

*Received December 28, 2008. Accepted for publication October 16, 2009. Published online February 22, 2010. Recommended by M. Eiermann.

†Seminar for Applied Mathematics, ETH Zurich, CH-8092 Zurich, Switzerland (mhg@math.ethz.ch). Work done while the author was visiting the TU Berlin, supported by the DFG Forschungszentrum MATHEON and the Mercator Visiting Professorship Program of the DFG.

[1] We write the acronyms for the various methods in a unified way, which sometimes differs from the one in the original publication.

connection is that the IDR residual polynomials are of the form

$$\rho_n^{\mathsf{IDR}}(t) = \begin{cases} \Omega_j(t)\rho_j(t) & \text{if } n = 2j, \\ \Omega_j(t)\widehat{\rho}_{j+1}(t) & \text{if } n = 2j+1, \end{cases} \tag{1.1}$$

where $\Omega_0(t) :\equiv 1$, $\Omega_j(t) :\equiv (1-\omega_1 t)\cdots(1-\omega_j t)$, and where $\rho_j$ denotes the $j$th BiCG residual polynomial, which is often referred to as a Lanczos polynomial, scaled such that $\rho_j(0) = 1$, while $\widehat{\rho}_{j+1}$ denotes another residual polynomial, which has degree $j+1$. A new linear factor $(1-\omega_{j+1}t)$ is appended to $\Omega_j$ in every other step, and it was suggested to choose it such that the norm of the new IDR residual is minimized among those that lie on a certain straight line. This is a widely used type of minimization step. For example, it is also found in the conjugate residual [34] method, but there it leads to a global minimum solution (for a symmetric positive definite matrix). And such a minimization is a key ingredient of BiCGStab. The publication [40] only mentioned that the first line of (1.1) had been proven in the case where the matrix $\mathbf{A}$ of the (real) linear system of equations $\mathbf{Ax} = \mathbf{b}$ is symmetric positive definite.

In 1984, Sonneveld introduced another Lanczos-type method: the Conjugate Gradient Squared (CGS) method [29]. It is based on residual polynomials that are the squares of the BiCG residual polynomials:[2]

$$\rho_n^{\mathsf{CGS}}(t) = \rho_n^2(t). \tag{1.2}$$

Note that the indexing of the residual polynomials and residuals is different in IDR and CGS: in the former, the degree increases by one when the index grows by one; in the latter, the degree increases by two.

The CGS paper [29] was received by the SIAM Journal on Scientific and Statistical Computing (SISSC) on April 24, 1984, but it took nearly five years to get published in revised and extended form [31]. Nevertheless, the method was accepted quickly by numerical analysts and engineers. In typical cases it converges nearly twice as fast as BiCG, though often in a very erratic manner. Although the idea and derivation of CGS are ingenious, they are easy to understand: starting from the standard recursions for the BiCG residual polynomials, one just derives recursions for their squares by defining additional suitable products of pairs of polynomials.

Yet another similar method was presented at the Householder Symposium in Tylösand in June 1990 by van der Vorst, then still also at Delft University. The title of his talk and the corresponding paper coauthored by Sonneveld and submitted to SISSC on May 21, 1990, was *"CGSTAB: A more smoothly converging variant of CG-S"* [37]. As part of the revision process the title was later changed into *"Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems"*, and Sonneveld resigned as a coauthor [36]. In this paper, van der Vorst started from the first formula in (1.1), now written as

$$\rho_n^{\mathsf{BiCGSTAB}}(t) = \Omega_n(t)\rho_n(t). \tag{1.3}$$

Thus, he adopted the indexing from CGS, and he also adopted from CGS the derivation based on directly finding a recursion for these residual polynomials; that is, he abstained from using the recursions imposed by the IDR approach.

[2]Therefore the name Biconjugate Gradient Squared (BiCGS) method would also make sense. However Sonneveld's view was that CGS is derived from a CG-type algorithm for building up a set of orthogonal (or formally orthogonal) polynomials [30]. Only after the recursions are mapped into a Krylov space that is embedded in an inner product space does the notion of biorthogonality make sense.

In the following years, BiCGStab was generalized to BiCGStab2 [12] and BiCGStab($\ell$) [24, 27], where the polynomial $\Omega_n$ is built up from factors of degree 2 and $\ell$, respectively, whose coefficients are determined by a two– or an $\ell$–dimensional residual norm minimization, respectively. This allows a better approximation of complex eigenvalues and yields typically faster convergence at the price of higher complexity, but only slightly higher computational cost. Nevertheless, the simple original BiCGStab became the method of choice for most users who apply a Lanczos-type method for solving a nonsymmetric linear system.

Due to the structure of the residual polynomials, CGS and the methods of the BiCGStab family are often referred to as Lanczos-type product methods (LTPMs) [13] or as hybrid BiCG methods [26].[3]

A new dimension came into play when, in 1997, Yeung and Chan submitted the paper *ML($k$)BiCGSTAB: a BiCGSTAB variant based on multiple Lanczos starting vectors* to the renamed SIAM Journal on Scientific Computing (SISC) [41]. In this paper, they first introduced with ML($k$)BiCG a version of BiCG where the left Krylov space (generated by $\mathbf{A}^\mathsf{T}$ from an arbitrary shadow residual $\widetilde{\mathbf{r}}_0$), which is used for the oblique projection of $\mathbf{r}_0$, is replaced by a block Krylov space generated from a matrix $\widetilde{\mathbf{R}}_0$ with $k$ columns. Then they generalized the transition from BiCG to BiCGStab to the new situation. This led to very complicated formulas, which were then meticulously modified to get a simpler and efficient code. The method was shown to converge amazingly well for a large number of fairly ill-conditioned examples, handled mostly without preconditioning and with rather large $k$, $25 \leq k \leq 100$. However, the large $k$ meant high memory consumption and considerable computational cost.[4] Although the essence of the paper is well explained and easy to understand, the complexity of the formulas and, paradoxically, the treatment of all the details must have kept people from reading the paper and applying the method — despite the very promising numerical results. The authors were aware of the connection to nonsymmetric block Lanczos methods [1, 3, 5, 6, 8, 19], but while these are based on generalizing the Lanczos three-term recursions, Yeung and Chan generalized the two-term recursions of BiCG, as was done before by Simoncini [23]. This was partly the reason for the complex formulas.

In [40], Wesseling and Sonneveld had announced a further publication on IDR to be in preparation, but only in 2007 was such a paper submitted, again to SISC; see [32] and, for the final version, [33]. In the sequel of an enquiry by Jens-Peter Zemke [43] in May 2006, Sonneveld and van Gijzen reconsidered IDR and generalized it to IDR($s$), where the original method is included as the case $s = 1$ (except for a small but interesting detail). They also clarify the relation of their method to BiCGStab (when $s = 1$) and, in the final version, to ML($s$)BiCGStab (when $s > 1$), which they did not know when submitting the paper. It turns out that the even-indexed IDR(1) residuals are (up to roundoff effects) exactly the BiCGStab residuals, and that likewise every $(s+1)$th IDR($s$) residual is, up to the possibly different choice of the parameters $\omega_j$, a ML($s$)BiCGStab residual. However, the way these residuals are constructed differs, and the "intermediate" residuals do not exist in BiCGStab and differ in ML($s$)BiCGStab, respectively. The paper also features numerical examples that are relevant in practice and demonstrate the power of the method even for small values of $s$ where the cost per step in $n$ is small.

In a follow-up publication, Sleijpen, Sonneveld, and van Gijzen [25] introduced partly different notation and tried to explain IDR($s$) and its connection to BiCGStab from a somewhat different viewpoint, but this author prefers the presentation in [33]. They also introduce

[3] Note that many other authors have introduced other classes of "hybrid" iterative methods for linear systems.

[4] For one example, the matrix ORSIRR1, the dependence on $k$ was investigated for small $k$, where dramatic improvements can be noticed already for $2 \leq k \leq 10$; see Figure 3(b) of [41].

methods similar to ML($s$)BiCG and ML($s$)BiCGStab.

In the recent publication [38], van Gijzen and Sonneveld introduce yet another, very ingenious algorithm that fits into the IDR framework and leads to an elegant code. It uses recursions that are quite different from those of the original IDR($s$) of [33] and produces "intermediate residuals" satisfying additional orthogonality conditions that lead to shorter recurrence relations for some $n$ and improved memory management. The authors do not introduce a separate name for this new algorithm, but in some of the figures they refer to it as IDR($s$) Bi-ortho. We will discuss this IDR variant in Section 5 and use the shorter acronym IDR($s$)BiO.

**2. From BiCG to ML($k$)BiCGStab.** In this section we review some basic facts on Krylov space solvers, putting special emphasis on the biconjugate gradient (BiCG) method, and then look at the transition from BiCG to BiCGStab. We also have a quick look at Yeung and Chan's [41] generalization of these methods to ML($k$)BiCG and ML($k$)BiCGStab, respectively, which feature multiple initial left (or shadow) residuals.

**2.1. Krylov space solvers based on projection.** Given a nonsingular linear system $\mathbf{A}\mathbf{x} = \mathbf{b} \in \mathbb{C}^N$ and an initial approximation $\mathbf{x}_0$ along with its residual $\mathbf{r}_0 :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_0$, a Krylov space solver constructs recursively approximate solutions $\mathbf{x}_n$ (often referred to as iterates) such that

$$\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n,$$

where

$$\mathcal{K}_n :\equiv \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) :\equiv \mathsf{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{n-1}\mathbf{r}_0\}$$

is the $n$th Krylov subspace generated by $\mathbf{A}$ from $\mathbf{r}_0$.[5] Note that we allow complex data now.

Two of the basic theoretical facts in this setting are: (i) There is a minimal $\nu$ such that $\mathcal{K}_\nu$ is invariant. (ii) For the solution $\mathbf{x}_\star$, it holds that $\mathbf{x}_\star \in \mathbf{x}_0 + \mathcal{K}_\nu$, and $\nu$ is the minimal index for which this is true. So, if we choose $\mathbf{x}_n$ well, the solver terminates in $\nu$ steps. In particular, it suffices to choose the iterates so that the corresponding residuals $\mathbf{r}_n$ are linearly independent unless zero. In practice, $\nu$ is typically large (close to $N$), and therefore this finite termination property is irrelevant.

The true aim is to find $\mathbf{x}_n$ very close to $\mathbf{x}_\star$ in few (or at least not very many) steps. Because of the limited computer memory, it is important to find solvers that allow us to compute $\mathbf{x}_n$ with short recursions. The restriction $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n$ implies that

$$\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n \subseteq \mathcal{K}_{n+1}. \tag{2.1}$$

Most methods produce residuals that have a component in the "new part of the space", that is $\mathbf{r}_n \notin \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-1}$; in other methods, there may occur exceptional situations with $\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-1}$, which implies that the residuals are linearly dependent at this moment.

We will refer to spaces of the form $\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n$ or the form $\mathbf{x}_0 + \mathcal{K}_n$ as affine Krylov subspaces.

Since the goal is a small $\mathbf{r}_n$, we need to approximate $\mathbf{r}_0$ by elements from $\mathbf{A}\mathcal{K}_n$. For example, $\|\mathbf{r}_n\|$ is minimal if we choose $\mathbf{r}_n$ as the perpendicular from $\mathbf{r}_0$ to its orthogonal projection into $\mathbf{A}\mathcal{K}_n$. This is the basis of the conjugate residual (CR) method [34] for Hermitian systems and its various generalizations for the non-Hermitian case, such as GCR and

[5] $\mathcal{K}_n(\mathbf{B}, \mathbf{y})$ denotes in this paper a Krylov subspace generated by $\mathbf{B}$ from $\mathbf{y}$, while $\mathcal{K}_n$ without an argument is an abbreviation for $\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$.

GMRES. For these methods, we have

$$\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n, \qquad \mathbf{r}_n \perp \mathbf{A}\mathcal{K}_n.$$

Some Krylov space solvers are based on other orthogonal or oblique projections. In particular, for the biconjugate gradient (BICG) method [18, 4], which is of special importance here, we have

$$\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n, \qquad \mathbf{r}_n \perp \widetilde{\mathcal{K}}_n :\equiv \mathcal{K}_n(\mathbf{A}^\star, \widetilde{\mathbf{r}}_0). \tag{2.2}$$

Here, the initial shadow residual $\widetilde{\mathbf{r}}_0$ can be chosen arbitrarily; preferably, it should be in arbitrary position with respect to an eigenbasis of $\mathbf{A}^\star$.

The most often used recursions for BICG are the coupled two-term or BIOMIN recursions, which can be written as follows:

$$\begin{aligned}
\alpha_n &:= \delta_n/\delta'_n, && \text{(2.3a)}\\
\mathbf{r}_{n+1} &:= \mathbf{r}_n - \alpha_n \mathbf{A}\mathbf{v}_n, && \text{(2.3b)}\\
\widetilde{\mathbf{r}}_{n+1} &:= \widetilde{\mathbf{r}}_n - \overline{\alpha_n}\mathbf{A}^\star\widetilde{\mathbf{v}}_n, && \text{(2.3c)}\\
\mathbf{x}_{n+1} &:= \mathbf{x}_n + \alpha_n \mathbf{v}_n, && \text{(2.3d)}\\
\delta_{n+1} &:= \langle \widetilde{\mathbf{r}}_{n+1}, \mathbf{r}_{n+1}\rangle, && \text{(2.3e)}\\
\beta_n &:= \delta_{n+1}/\delta_n, && \text{(2.3f)}\\
\mathbf{v}_{n+1} &:= \mathbf{r}_{n+1} + \beta_n \mathbf{v}_n, && \text{(2.3g)}\\
\widetilde{\mathbf{v}}_{n+1} &:= \widetilde{\mathbf{r}}_{n+1} + \overline{\beta_n}\widetilde{\mathbf{v}}_n, && \text{(2.3h)}\\
\delta'_{n+1} &:= \langle \widetilde{\mathbf{v}}_{n+1}, \mathbf{A}\mathbf{v}_{n+1}\rangle. && \text{(2.3i)}
\end{aligned}$$

In addition to the residuals $\mathbf{r}_n$ and iterates $\mathbf{x}_n$, three other sets of vectors are constructed: shadow residuals (or left-hand side Lanczos vectors) $\widetilde{\mathbf{r}}_n \in \widetilde{\mathbf{r}}_0 + \mathbf{A}^\star\widetilde{\mathcal{K}}_n$, search directions $\mathbf{v}_n \in \mathbf{v}_0 + \mathbf{A}^\star\mathcal{K}_n$, and shadow search directions $\widetilde{\mathbf{v}}_n \in \widetilde{\mathbf{v}}_0 + \mathbf{A}^\star\widetilde{\mathcal{K}}_n$. All these vectors are updated by coupled two-term recursions.

**2.2. Residual polynomials and Lanczos-type product methods.** Let $\mathcal{P}_n$ denote the space of polynomials of degree at most $n$, and let $\mathcal{P}_n^\circ :\equiv \{\rho \in \mathcal{P}_n\,;\, \rho(0) = 1\}$. The inclusion (2.1) implies that one can associate $\mathbf{r}_n$ with a residual polynomial $\rho_n \in \mathcal{P}_n^\circ$ such that $\mathbf{r}_n = \rho_n(\mathbf{A})\mathbf{r}_0$. Roughly, $\|\mathbf{r}_n\|$ is small if $|\rho_n(t)|$ is small at those eigenvalues of $\mathbf{A}$ that are "active" when $\mathbf{r}_0$ is written in terms of the eigenbasis of $\mathbf{A}$. (This statement needs to be modified when $\mathbf{A}$ has no eigenbasis or an ill-conditioned one.) This observation motivates derivations of Krylov space methods via real (when $\mathbf{A}$ is Hermitian) or complex (when $\mathbf{A}$ is non-Hermitian) approximation problems. It must also have motivated Sonneveld's CGS method [31], where, as noted in (1.2), the residual polynomials are the squares of the BICG residual polynomials $\rho_n$. Clearly, whenever $|\rho_n(t)| \ll 1$ at an eigenvalue, $|\rho_n^2(t)|$ is even much smaller there. But often the residual norm of BICG oscillates wildly as a function of $n$, and then the residual norm of CGS oscillates even more. Avoiding or at least damping this was the motivation for van der Vorst [36] for the choice (1.3) in BICGSTAB. Recall that there, at step $n$, $\omega_n$ is chosen to minimize the residual norm on a straight line.

Let $\overline{\rho}_n$ denote the polynomial obtained from $\rho_n$ by complex conjugation of the coefficients. Then, in BICG, we have $\mathbf{r}_n = \rho_n(\mathbf{A})\mathbf{r}_0$ and $\widetilde{\mathbf{r}}_n = \overline{\rho}_n(\mathbf{A}^\star)\widetilde{\mathbf{r}}_0$. In addition, the search direction polynomials $\sigma_n \in \mathcal{P}_n \backslash \mathcal{P}_{n-1}$ associated with the search directions $\mathbf{v}_n$ play an important role: since $\mathbf{v}_0 = \mathbf{r}_0$ and $\widetilde{\mathbf{v}}_0 = \widetilde{\mathbf{r}}_0$, we have $\mathbf{v}_n = \sigma_n(\mathbf{A})\mathbf{r}_0$ and $\widetilde{\mathbf{v}}_n = \overline{\sigma}_n(\mathbf{A}^\star)\widetilde{\mathbf{r}}_0$.

**ETNA**
**Kent State University**
**http://etna.math.kent.edu**



Hence, associated with the recursions (2.3b), (2.3c), (2.3g), and (2.3h) there are the underlying coupled polynomial recursions

$$\rho_{n+1}(t) := \rho_n(t) - \alpha_n t \sigma_n(t), \qquad \sigma_{n+1}(t) := \rho_{n+1}(t) + \beta_n \sigma_n(t). \tag{2.4}$$

These polynomial recursions are fundamental for deriving CGS and BICGSTAB, and also, as we will see in Section 4, for understanding the difference between IDR$(1)$ and BICGSTAB.

For CGS it is easy to derive from (2.4) four coupled recursions for the polynomials $\rho_n^{\mathsf{CGS}} :\equiv \rho_n^2$, $\sigma_n^2$, $\rho_n\sigma_n$, and $\rho_n\sigma_{n-1}$, which can then be translated into recursions for elements of $\mathcal{K}_\nu$. Likewise, for BICGSTAB one combines (2.4) with the trivial recursion $\Omega_{n+1}(t) = (1-\omega_n\, t)\Omega_n(t)$ to derive three recursions for the three products $\rho_n^{\mathsf{BiCGSTAB}} :\equiv \rho_n\Omega_n$, $\rho_n\Omega_{n-1}$, and $\sigma_n\Omega_n$. In both cases alternative recursions exist, too.[6]

**2.3. Multiple initial shadow residuals.** Yeung and Chan [41] generalized BICG by replacing the left Krylov subspaces $\widetilde{\mathcal{K}}_n$ by block Krylov subspaces, which are a sum of Krylov spaces for the same matrix $\mathbf{A}^\star$ but with several different initial shadow residuals, stored as the columns of an $N \times s$ matrix $\widetilde{\mathbf{R}}_0$. Yeung and Chan called the resulting method the ML$(s)$BICG method (except that they used $k$ instead of $s$). The residuals whose index is a multiple of $s$ satisfy

$$\mathbf{r}_{sj} \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{sj}, \qquad \mathbf{r}_{sj} \perp \mathcal{K}_j(\mathbf{A}^\star, \widetilde{\mathbf{R}}_0) :\equiv \sum_{i=1}^{s} \mathcal{K}_j(\mathbf{A}^\star, \widetilde{\mathbf{r}}_0^{(i)}). \tag{2.5}$$

For the others, with index $n = sj + \ell$, where $1 < \ell < s$, we have analogously

$$\begin{aligned} &\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n, \\ &\mathbf{r}_n \perp \mathcal{K}_{j;\ell}(\mathbf{A}^\star, \widetilde{\mathbf{R}}_0) :\equiv \sum_{i=1}^{\ell} \mathcal{K}_{j+1}(\mathbf{A}^\star, \widetilde{\mathbf{r}}_0^{(i)}) + \sum_{i=\ell+1}^{s} \mathcal{K}_j(\mathbf{A}^\star, \widetilde{\mathbf{r}}_0^{(i)}). \end{aligned} \tag{2.6}$$

These residuals could also be constructed by using a variant of the nonsymmetric block Lanczos method, where the block size of the left block Krylov space is $s$, while that of the right (block) Krylov space is just one, that is, the right space is an ordinary Krylov space; see [1, 3, 5, 6, 8, 19] for ways to construct bases for these spaces with short recursions. Yeung and Chan rather generalize block BICG, described before by O'Leary [20] and Simoncini [23], but the latter authors assumed the same block size in the left-hand and the right-hand side Krylov spaces. Unfortunately, in theory and practice there can (and ultimately do) occur problems that must be addressed by block size reduction (deflation). Moreover there may occur Lanczos breakdowns and pivot breakdowns; see, e.g., [13] for a discussion of the breakdowns of BICG. Deflation and breakdowns have not been addressed in [41].

Yeung and Chan [41] continued by applying to ML$(s)$BICG the same transformation that turns BICG into BICGSTAB. Unfortunately, this lead to rather complicated formulas, which they were able to simplify and economize somewhat by algebraic manipulations. The resulting algorithm, called ML$(s)$BICGSTAB, was shown to be very effective for a large number of rather ill-conditioned test matrices.

Under the titles Bi-CG and Bi-CGSTAB, Sleijpen, Sonneveld, and van Gijzen [25] sketched two methods that are in spirit the same as ML$(s)$BICG and ML$(s)$BICGSTAB,

[6] In [11] four sets of equivalent recursions for CGS derived from the BIORES and BIODIR recursions of BICG are given; however, they are all more complicated than the original CGS recursions, and therefore more costly in work and storage.

but in detail differ considerably. First, they are not using the equivalent of Lanczos's coupled two-term recursions; second, their "intermediate" residuals satisfy only a block biorthogonality, and not the stricter requirement of (2.6) that determines the residuals of ML($s$)BICG uniquely.

**3. IDR basics.** In this section, we review basic facts about the IDR($s$) method, following essentially the presentation in [33]. One aspect that we stress more explicitly than Sonneveld and van Gijzen is that IDR($s$) is a Krylov space method, and therefore the residuals lie in an affine space that is embedded in a Krylov subspace. We also try to give more realistic figures, although we will see that they still do not reflect the whole truth.

**3.1. The IDR Theorem.** The IDR approach is based on a finite series of nested linear subspaces $\mathcal{G}_j$ of diminishing dimension with the property that for some increasing index sequence $\{n_j\}$ the residuals $\mathbf{r}_n$ with $n \geq n_j$ all lie in $\mathcal{G}_j$. Of course, all residuals lie in the invariant Krylov space $\mathcal{K}_\nu :\equiv \mathcal{K}_\nu(\mathbf{A}, \mathbf{r}_0)$; therefore, we can start with $n_0 :\equiv 0$ and $\mathcal{G}_0 :\equiv \mathcal{K}_\nu$. The other spaces $\mathcal{G}_j$ are defined by the recursion

$$\mathcal{G}_j :\equiv (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}). \tag{3.1}$$

Here, $\mathcal{S}$ is a prescribed linear subspace of codimension $s \ll N$, and the constants $\omega_j \neq 0$ will be suitably chosen to boost convergence. Let us denote the dimension of $\mathcal{G}_j$ by $d_j$. Clearly, $\mathcal{G}_{j-1} \cap \mathcal{S}$ can be represented by $N - d_{j-1} + s$ linear equations, and it is likely that these are linearly independent. However, as pointed out in [33], we cannot conclude easily that linear independence is here a generic property (valid for almost all problems if data are chosen randomly) since $\mathcal{G}_{j-1}$ actually depends on $\mathcal{S}$. But, typically, $\mathcal{G}_{j-1} \cap \mathcal{S}$ and its image $\mathcal{G}_j$ have dimension $d_j = d_{j-1} - s$. We will mostly take it for granted that this and other regularity assumptions are satisfied, and we will refer to this as the *regular case*.[7] One can see, however, by analogy to the behavior of the related ML($s$)BICG and ML($s$)BICGSTAB methods, that we cannot expect that $d_j = d_{j-1} - s$ remains true for $j$ up to $N/s$ if $s > 1$.

The IDR Theorem, given next, states two properties that are not apparent: the spaces $\mathcal{G}_j$ are nested; and, under mild assumptions on $\mathcal{S}$, the inclusion is strict, i.e., $\mathcal{G}_j \subsetneq \mathcal{G}_{j-1}$. For an illustration, see Figure 3.1.

THEOREM 3.1 (IDR Theorem [40, 33]). *Assume that $\mathcal{S} \cap \mathcal{G}_0$ contains no eigenvector of $\mathbf{A}$. Then*

$$\mathcal{G}_j \subsetneq \mathcal{G}_{j-1} \textit{ unless } \mathcal{G}_{j-1} = \{\mathbf{o}\}.$$

For the proof, see [40, 33]. As a consequence of the strict inclusions, $\mathcal{G}_j = \{\mathbf{o}\}$ for some $j \leq N$, say, $j \equiv: J$. However, the bound $J \leq N$ that follows from the IDR Theorem leads to a strong overestimation of the finite termination index, for the simple reason that termination is characterized by $d_J = 0$, which we can expect for $J$ of the size $N/s$.

Sonneveld and van Gijzen [33] also provide the Extended IDR Theorem, the main result of which is that the difference $d_j - d_{j+1}$ is monotonically non-increasing:

$$0 \leq d_j - d_{j+1} \leq d_{j+1} - d_j \leq s.$$

Alternatively, this result could also be concluded from the connection to ML($s$)BICG.

Of course, neither $\mathcal{G}_0$ nor the other spaces $\mathcal{G}_j$ are known in advance in the sense that we know a basis for them. In theory, the IDR algorithm would provide these bases if we

[7] The authors of [33] refer to it as the *generic case*, although this may not be fully consistent with the common usage of the word *generic*.

**ETNA**
Kent State University
http://etna.math.kent.edu



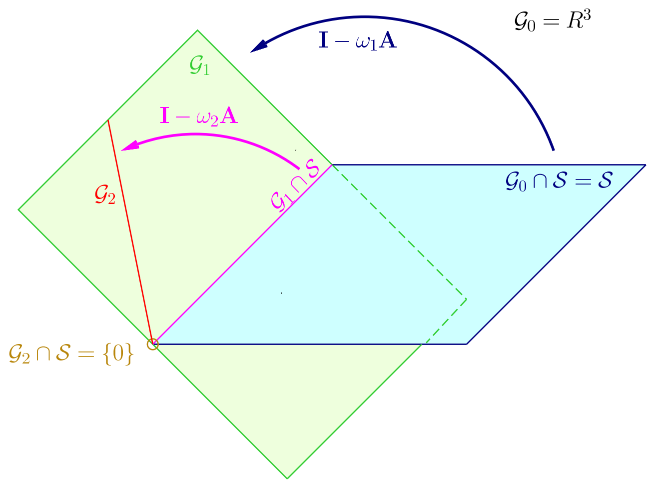


FIGURE 3.1. *Case* $s = 1$*: The spaces* $\mathbb{R}^3 = \mathcal{K}_3 = \mathcal{G}_0 \supsetneq \mathcal{G}_1 \subsetneq \mathcal{G}_2$.

continued it until the exact solution of $\mathbf{Ax} = \mathbf{b}$ was found, that is, until $\mathbf{r}_\nu = \mathbf{o}$ was attained; but this is not feasible unless $\nu$ is very small, so that the linear system can be solved exactly in a few steps.

IDR constructs typically only $s + 1$ residuals in $\mathcal{G}_j$ before turning to the next subspace $\mathcal{G}_{j+1} \subsetneq \mathcal{G}_j$. To accommodate exceptional situations we introduce a monotonically growing index sequence $\{n_j\}$ defined implicitly by[8]

$$\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n), \qquad n \geq n_j. \tag{3.2}$$

In the normal case, $n_j = (s+1)j$, so $n_{j+1} - n_j = s+1$, for reasons we will see in a moment; but $n_{j+1} - n_j > s+1$ may occur in exceptional situations. In the simplest case $s = 1$, which is when IDR is closely related to BiCGStab, two new residuals are computed for each $j$. This is depicted in Figure 3.2. The details of the construction are discussed next.

**3.2. Recursions.** The recursion for the residuals builds upon the recursion (3.1) for the spaces $\mathcal{G}_j$:

$$\mathbf{r}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n, \qquad \mathbf{v}_n \in \mathcal{G}_{j-1} \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n). \tag{3.3}$$

We suppose here that, for all $n$, $\mathbf{v}_n$ lies in the affine Krylov subspace spanned by $\mathbf{r}_1, \ldots, \mathbf{r}_n$ and shifted by $\mathbf{r}_0$. This means that $\mathbf{v}_n$ and $\mathbf{r}_{n+1}$ have "maximum degree" in the sense that $\mathbf{v}_n \notin \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-1}$ and $\mathbf{r}_{n+1} \notin \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n$. There may be situations where the latter assumptions do not hold, and, according to [33], there are in the IDR framework ways to recover from such situations, but we will not treat that here. There is some vague analogy to look-ahead Lanczos [7] or look-ahead block Lanczos [1] in such recovery procedures.

To construct vectors or "points" $\mathbf{r}_{n+1} \in \mathcal{G}_j$, we need vectors $\mathbf{v}_n \in \mathcal{G}_{j-1} \cap \mathcal{S}$, and the first time we construct a point in $\mathcal{G}_j$, we can choose $\omega_j$. To construct $\mathbf{v}_n$ in $\mathcal{G}_{j-1} \cap \mathcal{S}$ we need to intersect an $s$-dimensional affine subspace of $\mathcal{G}_{j-1}$ (in theory it could be represented by $N - s$ inhomogeneous linear equations) with the subspace $\mathcal{S}$ represented by $s$ homogeneous linear equations that we may write as $\mathbf{P}^\star \mathbf{v}_n = \mathbf{o}$ with an $N \times s$ matrix $\mathbf{P}$ whose columns form

[8] It is conceivable that the inclusion in (3.2) holds by chance for some $n < n_j$ too, but for $n \geq n_j$ the inclusion will be forced by construction.

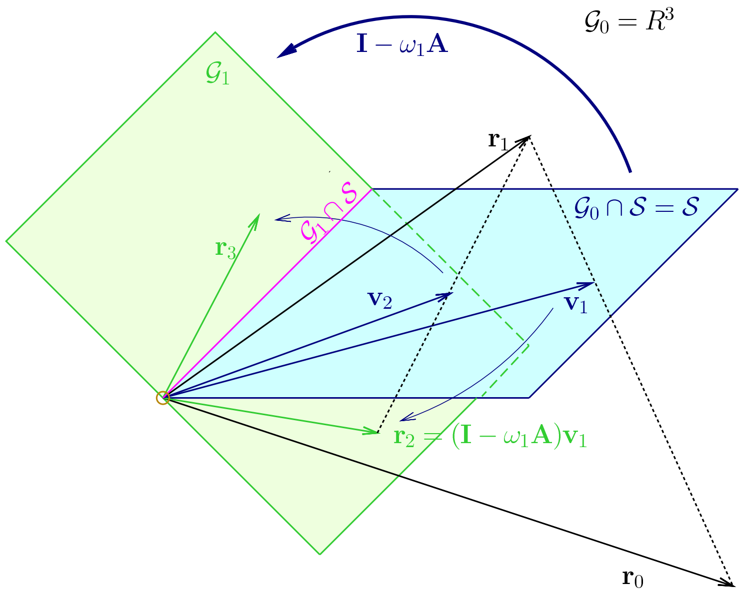


FIGURE 3.2. *Case $s = 1$: The first two steps: construction of $\mathbf{r}_2$ and $\mathbf{r}_3$ (for details see §4.1).*

a basis of $\mathcal{S}^\perp$. In theory, we would end up with $N$ inhomogeneous linear equation for $\mathbf{v}_n$ that will usually have a unique solution, but this is of course not feasible in practice. Instead, we represent the $s$-dimensional affine subspace of $\mathcal{G}_{j-1}$ directly by an affine combination (a linear combination whose coefficients sum up to one) of $s+1$ points in $\mathcal{G}_{j-1}$. The natural choice for these $s+1$ points are the last computed residuals $\mathbf{r}_{n-s}, \ldots, \mathbf{r}_n$. Here we see why we need $n_{j+1} - n_j \geq s+1$. A neat way to take the condition of an affine combination into account is to introduce the differences of the residual vectors, and that is what Sonneveld and van Gijzen do:

$$\mathbf{v}_n := \mathbf{r}_n - \sum_{i=1}^{\iota(n)} \gamma_i^{(n)} \Delta \mathbf{r}_{n-i} = \mathbf{r}_n - \Delta \mathbf{R}_n \, \mathbf{c}_n, \tag{3.4}$$

where

$$\begin{aligned} s \leq \iota(n) &\leq n - n_{j-1}, \\ \Delta \mathbf{r}_n &:\equiv \mathbf{r}_{n+1} - \mathbf{r}_n, \\ \Delta \mathbf{R}_n &:\equiv \begin{bmatrix} \Delta \mathbf{r}_{n-1} & \ldots & \Delta \mathbf{r}_{n-\iota(n)} \end{bmatrix}, \\ \mathbf{c}_n &:\equiv \begin{bmatrix} \gamma_1^{(n)} & \ldots & \gamma_{\iota(n)}^{(n)} \end{bmatrix}^\mathsf{T}. \end{aligned}$$

The restriction $\iota(n) \leq n - n_{j-1}$ ensures that $\Delta \mathbf{r}_{n-i} \in \mathcal{G}_{j-1}$. Usually, $\iota(n) = s$, but, again, there may be exceptional situations not covered here where one needs to choose a larger $\iota(n)$. Note that using the differences of the residuals leads to a $\mathbf{v}_n$ whose polynomial representation automatically inherits from $\mathbf{r}_n$ the value 1 at zero. Therefore, indeed $\mathbf{v}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n$, and we may view $\mathbf{v}_n$ as a residual, so there is $\mathbf{x}'_n \in \mathbf{x}_0 + \mathcal{K}_n$ such that $\mathbf{v}_n = \mathbf{b} - \mathbf{A}\mathbf{x}'_n$.

To enforce $\mathbf{v}_n \in \mathcal{S}$ we enforce[9] $\mathbf{v}_n \perp \mathcal{S}^\perp = \mathcal{R}(\mathbf{P})$, that is, $\mathbf{P}^\star \mathbf{v}_n = \mathbf{o}$. This means that the term $\Delta \mathbf{R}_n \, \mathbf{c}_n$ in (3.4) must be the oblique projection of $\mathbf{r}_n$ into $\mathcal{R}(\Delta \mathbf{R}_n)$ along

[9] $\mathcal{R}(\mathbf{P})$ denotes the range of $\mathbf{P}$.

ETNA
Kent State University
http://etna.math.kent.edu



$\mathcal{S}$. In order that this projection is uniquely defined, we need $\mathbf{P}^\star \Delta\mathbf{R}_n$ to be nonsingular, in particular $\iota(n) = s$ to make the matrix square. Then,

$$\mathbf{c}_n :\equiv \left(\mathbf{P}^\star \Delta\mathbf{R}_n\right)^{-1} \mathbf{P}^\star \mathbf{r}_n. \tag{3.5}$$

Otherwise, when $\iota(n) > s$, we might choose $\mathbf{c}_n$ as the minimum norm solution of an underdetermined least squares problem.

For the initial phase, that is, for constructing $\mathbf{r}_1, \ldots, \mathbf{r}_s$, we may apply a fairly arbitrary starting procedure, e.g., GMRES.

We need not just one point $\mathbf{r}_{n+1} \in \mathcal{G}_j$ but at least $s+1$ of them before we can continue to the next space $\mathcal{G}_{j+1}$. At first one might expect to need $2s+1$ points in $\mathcal{G}_{j-1}$ to repeat the above construction $s+1$ times. However, this is not the case. Because $\mathcal{G}_j \subset \mathcal{G}_{j-1}$, the just-constructed $\mathbf{r}_{n+1} \in \mathcal{G}_j$ also qualifies as a point of $\mathcal{G}_{j-1}$ and can be used when we replace $n$ by $n+1$ in the above construction.[10] So, $s+1$ points in $\mathcal{G}_{j-1} \backslash \mathcal{G}_j$ will usually be enough. However, we cannot fully exclude degenerate situations, where the last $s+1$ points constructed in $\mathcal{G}_{j-1}$ do not span an $s$-dimensional affine space and therefore $\Delta\mathbf{R}_n$ is singular (or nearly so). A careful implementation of the method will need to address such situations, which are also reflected by a zero (or absolutely small) coefficient in the $t^n$ term of the polynomial representation of some of the vectors $\mathbf{v}_n$.

For the case $s = 1$, the first two steps, from given $\mathbf{r}_0$ and $\mathbf{r}_1$ to $\mathbf{r}_2$ and $\mathbf{r}_3$, are shown in Figure 3.2. Then the construction is continued till $\mathbf{v}_5 = \mathbf{r}_6 = \mathbf{o}$ in Figure 3.3. However, our figures show actually one of the "exceptional situations" we just referred to: the constructed residuals are not all linearly independent. In fact, since the figures show a construction in $\mathbb{R}^3$ (i.e., $N = 3$), linearly independent residuals would mean convergence in at most three steps, that is, $\mathbf{r}_3 = \mathbf{o}$.

We could avoid using $\mathbf{v}_n$ by inserting (3.4) in (3.3):

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \Delta\mathbf{R}_n \mathbf{c}_n - \omega_j \mathbf{A}(\mathbf{r}_n - \Delta\mathbf{R}_n \mathbf{c}_n). \tag{3.6}$$

[10] The IDR$(s)$ variant of Section 5 will differ in the choice of points used in (3.4).

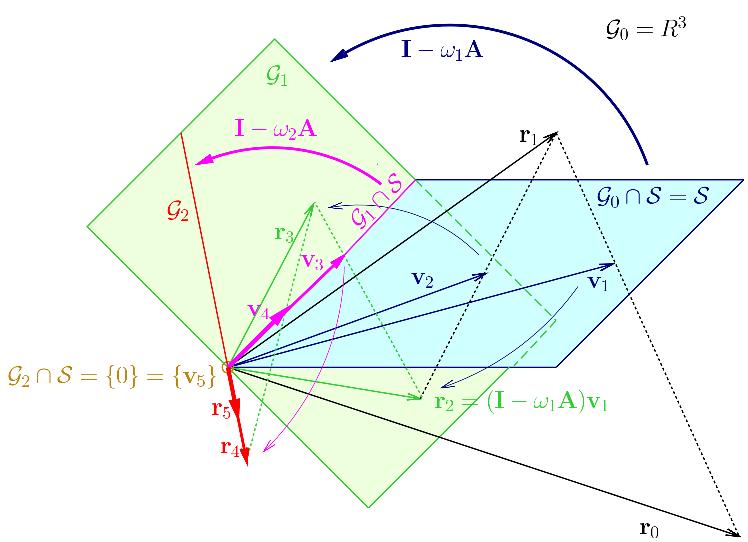


FIGURE 3.3. *Case $s = 1$: Construction of $\mathbf{r}_4$ and $\mathbf{r}_5$. Termination with $\mathbf{v}_5 = \mathbf{o}$ (for details see §4.1).*

This formula illustrates that IDR($s$) differs considerably from most commonly used Krylov space solvers such as CG, BiCG, or GCR. The difference is that in (3.6) not only $\mathbf{r}_n$ is multiplied by $\mathbf{A}$, but also $\mathbf{r}_{n-s}, \ldots, \mathbf{r}_{n-1}$. This means that, in the terminology of [10], IDR($s$) is a $(s+1, s+1)$-step method, while, e.g., CG, BiCG are $(2,1)$-step methods, OrthoMin($k$) is a $(k,1)$-step method, and the untruncated GCR is a $(\infty, 1)$-step method.

As mentioned before, $\omega_j$ can only be chosen when we construct the first point in $\mathcal{G}_j$, that is, $\mathbf{r}_{n+1}$ with $n+1 = n_j$. The formula $\mathbf{r}_{n+1} = (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n$ suggests that we choose $\omega_j$ so that $\|\mathbf{r}_{n+1}\|$ is minimal among all $\mathbf{r}$ of the form $\mathbf{r} = (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n$, that is, we choose it such that $\mathbf{r}_{n+1} \perp \mathbf{A}\mathbf{v}_n$:

$$\omega_j :\equiv \frac{\langle \mathbf{A}\mathbf{v}_n, \mathbf{v}_n \rangle}{\|\mathbf{A}\mathbf{v}_n\|^2}. \tag{3.7}$$

Note that this value of $\omega_j$ may turn out to be zero or close to zero. As in BiCGStab, this is a source of breakdown or instability, but it is easily cured by choosing another value that does not minimize the residual locally.

Finally, we need to address the fact that it does not suffice to construct residuals $\mathbf{r}_n$ and $\mathbf{v}_n$, but that we also need the corresponding approximate solution $\mathbf{x}_n$ and $\mathbf{x}'_n \in \mathbf{x}_0 + \mathcal{K}_n$. It is readily verified that

$$\mathbf{v}_n := \mathbf{r}_n - \Delta\mathbf{R}_n\, \mathbf{c}_n \quad \Longleftrightarrow \quad \mathbf{x}'_n := \mathbf{x}_n - \Delta\mathbf{X}_n\, \mathbf{c}_n, \tag{3.8}$$

$$\mathbf{r}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n \quad \Longleftrightarrow \quad \mathbf{x}_{n+1} := \omega_j \mathbf{v}_n + \mathbf{x}'_n. \tag{3.9}$$

There are several ways to rearrange these four recursions and to combine them with the iterate-residual relationships; see [25]. Also in the "prototype algorithm" of [33] a different, but equivalent set of recursions is used. It includes the analog of (3.6) for $\mathbf{x}_{n+1}$,

$$\mathbf{x}_{n+1} := \mathbf{x}_n - \Delta\mathbf{X}_n \mathbf{c}_n + \omega_j(\mathbf{r}_n - \Delta\mathbf{R}_n \mathbf{c}_n) \tag{3.10}$$

and the relation $\Delta\mathbf{r}_n = -\mathbf{A}\Delta\mathbf{x}_n$.

Note that here, as in any competitive set of recursions, the major cost of computing $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n$ consists of $n+1$ matrix-vector products (MVs) with $\mathbf{A}$. Regarding memory, one needs just to store $s$ columns of each $\mathbf{P}$, $\Delta\mathbf{X}_n$, and $\Delta\mathbf{R}_n$, plus a few single $N$-vectors.

**3.3. Characterization by orthogonality.** Clearly, the dimension of $\mathcal{G}_j$ gets reduced due to taking the intersection with the $(N-s)$–dimensional space $\mathcal{S}$. This dimension reduction is viewed as the basic force behind IDR and gave the method its name. However, dimension reduction in Krylov space solvers is not at all a unique feature of IDR. In fact, projection based methods can be understood in a similar way. For example, the characterization (2.2) of the BiCG residuals could be written as

$$\mathbf{r}_n \in \mathcal{L}_n^{\perp} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)\,,$$

where $\mathcal{L}_n = \widetilde{\mathcal{K}}_n = \mathcal{K}_n(\mathbf{A}^\star, \widetilde{\mathbf{r}}_0)$, and for CR, GCR, and GMRes the same is true with $\mathcal{L}_n = \mathbf{A}\mathcal{K}_n$. What is different in IDR is that $\mathcal{G}_j$ is not an orthogonal complement of a Krylov subspace. However, due to the form of the recursion for $\{\mathcal{G}_j\}$, $\mathcal{G}_j$ turns out to be the image of an orthogonal complement of a Krylov subspace. This result is implicit in Subsection 5.1 of [33] and has been explicitly formulated in [25]:

$$\mathcal{G}_j = \left\{ \Omega_j(\mathbf{A})\mathbf{w} \;\middle|\; \mathbf{w} \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P}) \right\} = \Omega_j(\mathbf{A})\, [\mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})]^{\perp}\,. \tag{3.11}$$

Here, as before, $\Omega_j(t) :\equiv (1-\omega_1 t)\cdots(1-\omega_j t) \in \mathcal{P}_j^\circ$, and $\mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})$ is the $j$th block Krylov subspace generated by $\mathbf{A}^\star$ from the $s$ columns of $\mathbf{P}$, which are assumed to be a basis of $\mathcal{S}^\perp$. Note that when we choose $\widetilde{\mathbf{R}}_0 = \mathbf{P}$, this block Krylov subspace is the same as in ML($k$)BiCG; see (2.5). Note also that the larger $s$, the larger is $\mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})$, and thus the smaller is $\mathcal{G}_j$.

To prove (3.11), let us repeat here the argument from [33] that is linked to the recursions that we have just discussed and provides further insight. (The induction proof of (3.11) in [25] is different.) We start from (3.1) and (3.3) and the observation that $\mathbf{v}_n \in \mathcal{G}_{j-1}$ must likewise be of the form

$$\begin{aligned}
\mathbf{v}_n &= (\mathbf{I} - \omega_{j-1}\mathbf{A})\,\mathbf{v}_n', & \mathbf{v}_n' &\in \mathcal{G}_{j-2} \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-1}),\\
\mathbf{v}_n' &= (\mathbf{I} - \omega_{j-2}\mathbf{A})\,\mathbf{v}_n'', & \mathbf{v}_n'' &\in \mathcal{G}_{j-3} \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-2}),\\
&\ \ \vdots & &\ \ \vdots\\
\mathbf{v}_n^{(j-2)} &= (\mathbf{I} - \omega_1\mathbf{A})\,\mathbf{w}_{n+1}, & \mathbf{w}_{n+1} &\in \mathcal{G}_0 \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j+1}).
\end{aligned}$$

Starting at the bottom, we can also write (with $\Omega_0 \equiv 1$):

$$\begin{aligned}
\mathbf{w}_{n+1} &= \Omega_0(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{G}_0 \cap \mathcal{S},\\
\mathbf{v}_n^{(j-2)} &= \Omega_1(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{G}_1 \cap \mathcal{S},\\
&\ \ \vdots\\
\mathbf{v}_n' &= \Omega_{j-2}(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{G}_{j-2} \cap \mathcal{S},\\
\mathbf{v}_n &= \Omega_{j-1}(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{G}_{j-1} \cap \mathcal{S},\\
\mathbf{r}_{n+1} &= \Omega_j(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{G}_j.
\end{aligned}$$

Since $\{\Omega_k\}_{k=0}^{j-1}$ is a basis of $\mathcal{P}_{j-1}$ we see that

$$\Omega(\mathbf{A})\mathbf{w}_{n+1} \in \mathcal{S} \quad (\forall \Omega \in \mathcal{P}_{j-1}),$$

that is, $\mathbf{P}^\star\Omega(\mathbf{A})\mathbf{w}_{n+1} = \mathbf{o} \in \mathbb{C}^s$ or, in other words, $\mathbf{w}_{n+1} \perp \overline{\Omega}(\mathbf{A}^\star)\mathbf{P}$, $\forall\overline{\Omega} \in \mathcal{P}_{j-1}$, or $\mathbf{w}_{n+1} \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})$. In summary, we conclude that any $\mathbf{r}_{n+1} \in \mathcal{G}_j$ is of the form

$$\mathbf{r}_{n+1} = \Omega_j(\mathbf{A})\mathbf{w}_{n+1}, \quad \mathbf{w}_{n+1} \in \mathcal{G}_0 \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j+1}), \quad \mathbf{w}_{n+1} \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P}). \tag{3.12}$$

This proves (3.11). For simplicity, we may replace $n+1$ by $n$ here and, for our records, write any $\mathbf{r}_n \in \mathcal{G}_j$ ($n = n_j, \ldots, n_{j+1} - 1$) as

$$\mathbf{r}_n = \Omega_j(\mathbf{A})\mathbf{w}_n, \qquad \mathbf{w}_n \in \mathcal{G}_0 \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j}), \qquad \mathbf{w}_n \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P}). \tag{3.13}$$

Note that for $n = n_j - 1$ and $n = n_j$, the polynomials associated with $\mathbf{w}_n$ have the same degree: $\mathbf{w}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n_j - j}$. (That is why we chose $n$ as index for $\mathbf{w}_n$, although this is not the degree of the associated polynomial.)

In the generic case, for fixed $j$, we will construct $n_{j+1} - n_j = s+1$ linearly independent vectors $\mathbf{w}_n$ that provide $s+1$ linearly independent vectors $\mathbf{r}_n$ (with $n_j \le n < n_{j+1}$). So, as long as we stay in the generic case, $n_j = j\,(s+1)$.

Moreover, generically, for $n = n_j = j\,(s+1)$ where $\mathbf{w}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{js}$ and $\mathbf{w}_n \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})$ with $\dim \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P}) = js = \dim \mathbf{A}\mathcal{K}_{js}$, there is a *unique* $\mathbf{w}_n$ satisfying (3.13), since it can be characterized as the solution of a linear system with a $js \times js$ matrix that can be assumed to be nonsingular in the generic case:

THEOREM 3.2 ([33]). *Assume that $n_j = j\,(s+1)$, $j = 1, 2, \ldots, J$, and that the iterates $\mathbf{x}_n$ and residuals $\mathbf{r}_n$ of IDR($s$) are for $n \le n_J$ uniquely constructible by the recursions* (3.3) *and* (3.4) *with the choice* $\iota(n) = s$ *and the coefficients* $\mathbf{c}_n$ *from* (3.5)*. Then, for* $j \le J$*, the residuals* $\mathbf{w}_{n_j}$ *and* $\mathbf{r}_{n_j}$ *are uniquely characterized by the conditions* (3.13)*.*

COROLLARY 3.3. *Under the assumptions of Theorem 3.2, and if the same parameters* $\omega_j$ $(1 \le j \le J)$ *have been chosen, the IDR*$(s)$ *iterates* $\mathbf{x}_{n_j}$ *and residuals* $\mathbf{r}_{n_j}$ *are identical to the iterates* $\mathbf{x}_j$ *and residuals* $\mathbf{r}_j$ *of* BICGSTAB *(if* $s = 1$*) or* ML$(s)$BICGSTAB *(if* $s > 1$*), respectively.*

But the $s$ other vectors $\mathbf{w}_n$ (with $n_j < n < n_{j+1}$), and thus also the corresponding residuals $\mathbf{r}_n$, are not uniquely determined by (3.13). We cannot expect that they appear in BICGSTAB or ML$(s)$BICGSTAB, and, in fact, they usually do not.

**4. The case $s = 1$ and the comparison with BICGSTAB.** If $s = 1$, the subspace $\mathcal{S}$ is a hyperplane determined by a single vector $\mathbf{p} \perp \mathcal{S}$. So the matrix $\mathbf{P}$ consists of the single column $\mathbf{p}$ now. By Corollary 3.3, when $s = 1$, every other set of vectors $\{\mathbf{w}_n, \mathbf{r}_n, \mathbf{v}_{n-1}, \mathbf{x}_n, \dots\}$ (with $n$ even) is uniquely determined up to the choice of the parameters $\omega_j$. If the latter are chosen as in BICGSTAB (and they usually are), and if $\widetilde{\mathbf{r}}_0 := \mathbf{p}$ in BICGSTAB, then

$$\mathbf{r}_{2j} = \mathbf{r}_j^{\text{BiCGSTAB}}, \qquad \mathbf{x}_{2j} = \mathbf{x}_j^{\text{BiCGSTAB}}, \qquad \mathbf{w}_{2j} = \mathbf{r}_j^{\text{BICG}}. \tag{4.1}$$

So there remains the question whether and how BICGSTAB and IDR$(1)$ differ. In order to answer this question, we will look at the polynomial recursions that mirror the recursions for the Krylov space vectors generated by the two methods.

**4.1. Recursions and orthogonality properties of IDR$(1)$.** When $s = 1$, the recursions (3.8) and (3.9) of IDR$(s)$ simplify to

$$\begin{aligned} \mathbf{v}_n &:= \mathbf{r}_n - \gamma_n(\mathbf{r}_n - \mathbf{r}_{n-1}), \qquad & \mathbf{x}'_n &:= \mathbf{x}_n - \gamma_n(\mathbf{x}_n - \mathbf{x}_{n-1}), \\ \mathbf{r}_{n+1} &:= (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n, & \mathbf{x}_{n+1} &:= \mathbf{x}'_n + \omega_j \mathbf{v}_n, \end{aligned} \tag{4.2}$$

where $n \ge 1$, $j = \lfloor (n+1)/2 \rfloor$. The first line can be written

$$\mathbf{v}_n := (1 - \gamma_n)\mathbf{r}_n + \gamma_n \mathbf{r}_{n-1}, \quad \mathbf{x}'_n := (1 - \gamma_n)\mathbf{x}_n + \gamma_n \mathbf{x}_{n-1},$$

to emphasize that $\mathbf{v}_n$ lies on the straight line through $\mathbf{r}_n$ and $\mathbf{r}_{n-1}$, and likewise, $\mathbf{x}'_n$ lies on the line through $\mathbf{x}_n$ and $\mathbf{x}_{n-1}$. By (3.5), $\gamma_n :\equiv \gamma_1^{(n)} = \langle \mathbf{p}, \mathbf{r}_n \rangle \,/\, \langle \mathbf{p}, \Delta \mathbf{r}_{n-1} \rangle$ is chosen such that $\mathbf{v}_n \in \mathcal{S}$, that is, $\mathbf{v}_n \perp \mathbf{p}$. This is illustrated in the Figures 3.2 and 3.3. The parameter $\omega_j$ is usually chosen to make $\mathbf{r}_{2j}$ as short as possible; this means that $\mathbf{r}_{2j}$ is orthogonal to $\mathbf{A}\mathbf{v}_{2j-1}$; see (3.7). (This property is not taken into account in the figures.)

From (4.1) and (3.13) we know that

$$\mathbf{w}_{2j} = \mathbf{r}_j^{\text{BICG}} = \rho_j(\mathbf{A})\mathbf{r}_0 \perp \widetilde{\mathcal{K}}_j, \tag{4.3}$$

where $\rho_j$ is still the $j$th Lanczos polynomial, and where now $\widetilde{\mathcal{K}}_j :\equiv \mathcal{K}_j(\mathbf{A}^\star, \mathbf{p})$. According to (3.13), $\mathbf{w}_{2j+1}$ is represented by a polynomial $\widehat{\rho}_{j+1} \in \mathcal{P}^\circ_{j+1}$ and

$$\mathbf{w}_{2j+1} = \widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 \perp \widetilde{\mathcal{K}}_j. \tag{4.4}$$

So, since $\mathbf{r}_n = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} = \Omega_j(\mathbf{A})\mathbf{w}_n$, we have

$$\begin{aligned} \mathbf{r}_n &= \Omega_j(\mathbf{A})\mathbf{w}_n = \begin{cases} \Omega_j(\mathbf{A})\rho_j(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j, \\ \Omega_j(\mathbf{A})\widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j+1, \end{cases} \\ \mathbf{v}_n &= \Omega_{j-1}(\mathbf{A})\mathbf{w}_{n+1} = \begin{cases} \Omega_{j-1}(\mathbf{A})\rho_j(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j-1, \\ \Omega_{j-1}(\mathbf{A})\widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j. \end{cases} \end{aligned} \tag{4.5}$$

Inserting these formulas into $\mathbf{v}_n = (1-\gamma_n)\mathbf{r}_n + \gamma_n \mathbf{r}_{n-1}$ we get, after a short calculation, for $n = 2j+1$ and $n = 2j$, respectively,

$$\begin{aligned} \rho_{j+1}(t) &:= (1-\gamma_{2j+1})\,\widehat{\rho}_{j+1}(t) + \gamma_{2j+1}\,\rho_j(t) \qquad (j=0,1,2,\dots), \\ \widehat{\rho}_{j+1}(t) &:= (1-\gamma_{2j})\,(1-\omega_j t)\,\rho_j(t) + \gamma_{2j}\,\widehat{\rho}_j(t) \qquad (j=1,2,\dots). \end{aligned} \tag{4.6}$$

**4.2. Comparison with the recursions and orthogonality properties of BiCGStab.** The Lanczos (residual) polynomials $\rho_j$ and the BiCG search direction polynomials $\sigma_j$ are formal orthogonal polynomials (FOPs) in the sense that, for $i \neq j$,

$$\rho_i \perp \rho_j \quad\Longleftrightarrow\quad \langle \overline{\rho}_i(\mathbf{A}^\star)\widetilde{\mathbf{r}}_0\,,\, \rho_j(\mathbf{A})\mathbf{r}_0\rangle = 0 \quad\Longleftrightarrow\quad \left\langle \widetilde{\mathbf{r}}_i^{\,\mathsf{BICG}}\,,\, \mathbf{r}_j^{\mathsf{BICG}} \right\rangle = 0,$$

$$\sigma_i \perp_t \sigma_j \quad\Longleftrightarrow\quad \langle \overline{\sigma}_i(\mathbf{A}^\star)\widetilde{\mathbf{r}}_0, \mathbf{A}\,\sigma_j(\mathbf{A})\mathbf{r}_0\rangle = 0 \quad\Longleftrightarrow\quad \left\langle \widetilde{\mathbf{v}}_i^{\,\mathsf{BICG}}, \mathbf{A}\mathbf{v}_j^{\mathsf{BICG}} \right\rangle = 0,$$

where $\mathbf{v}_j^{\mathsf{BICG}}$ and $\widetilde{\mathbf{v}}_i^{\,\mathsf{BICG}}$ are the search directions and the "shadow" search directions, respectively, that appeared in the recursions (2.3a)–(2.3i). Since $\{\rho_0, \dots, \rho_{j-1}\}$ and $\{\sigma_0, \dots, \sigma_{j-1}\}$ both span $\mathcal{P}_{j-1}$, we actually have

$$\rho_j \perp \mathcal{P}_{j-1}, \quad \sigma_j \perp_t \mathcal{P}_{j-1}, \quad\Longleftrightarrow\quad \mathbf{r}_j^{\mathsf{BICG}} \perp \widetilde{\mathcal{K}}_j, \quad \mathbf{v}_j^{\mathsf{BICG}} \perp_{\mathbf{A}} \widetilde{\mathcal{K}}_j.$$

Here, $\perp_{\mathbf{A}}$ denotes formal orthogonality with respect to the formal $\mathbf{A}$–inner product $\langle .,.\rangle_{\mathbf{A}}$. In summary, the basic BiCG recursions (2.3b), (2.3c), (2.3g), and (2.3h) upon which BiCGStab builds too, are mirrored by the following recursions for $\rho_j$ and $\sigma_j$:

$$\underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} := \underbrace{\rho_j(t)}_{\perp \mathcal{P}_{j-1}} - \alpha_j \underbrace{t\sigma_j(t)}_{\perp \mathcal{P}_{j-1}}, \qquad \underbrace{\sigma_{j+1}(t)}_{\perp t\, \mathcal{P}_j} := \underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} + \beta_j \underbrace{\sigma_j(t)}_{\perp t\, \mathcal{P}_{j-1}}. \tag{4.7}$$

Here, both $\alpha_j$ and $\beta_j$ are chosen so that the new polynomials $\rho_{j+1}$ and $\sigma_{j+1}$ feature the indicated orthogonality properties, by which they are uniquely determined up to a scalar multiple.

In contrast, in IDR$(1)$, by (4.6), (4.3), and (4.4),

$$\underbrace{\widehat{\rho}_{j+1}(t)}_{\perp \mathcal{P}_{j-1}} := (1-\gamma_{2j})\, \underbrace{(1-\omega_j t)\,\rho_j(t)}_{\perp \mathcal{P}_{j-2}} + \gamma_{2j}\, \underbrace{\widehat{\rho}_j(t)}_{\perp \mathcal{P}_{j-2}},$$

$$\underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} := (1-\gamma_{2j+1})\, \underbrace{\widehat{\rho}_{j+1}(t)}_{\perp \mathcal{P}_{j-1}} + \gamma_{2j+1}\, \underbrace{\rho_j(t)}_{\perp \mathcal{P}_{j-1}}.$$

Comparing these recursions for $(\rho_j, \widehat{\rho}_j)$ with (4.7) we easily see that

$$(1-\gamma_{2j+1})\,(\widehat{\rho}_{j+1}(t) - \rho_j(t)) = -\alpha_j\, t\, \sigma_j(t).$$

So,

$$\widehat{\rho}_{j+1}(t) = \rho_j(t) - \frac{\alpha_j}{1-\gamma_{2j+1}}\, t\, \sigma_j(t), \tag{4.8}$$

or, after multiplication by $\Omega_j(t)$ and translation into the Krylov space,

$$\mathbf{r}_{2j+1} = \mathbf{r}_{2j} - \frac{\alpha_j}{(1-\gamma_{2j+1})}\, \mathbf{A}\mathbf{s}_j^{\mathsf{BiCGSTAB}}, \qquad \text{where } \mathbf{s}_j^{\mathsf{BiCGSTAB}} :\equiv \Omega_j(\mathbf{A})\mathbf{v}_j^{\mathsf{BICG}}. \tag{4.9}$$

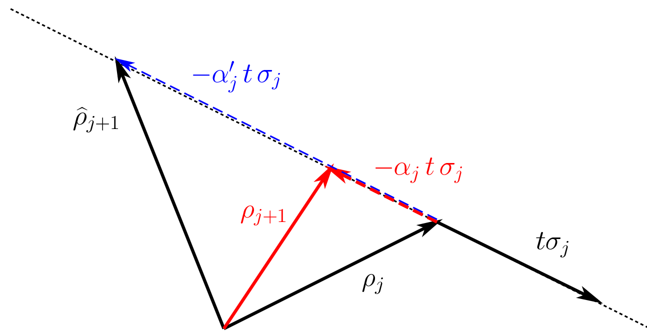


FIGURE 4.1. *Case $s = 1$: The connection between* BICGSTAB *and IDR*$(1)$.

This formula expresses the odd-indexed IDR$(1)$ residuals $\mathbf{r}_{2j+1}$ in terms of quantities from BICGSTAB and the IDR coefficient $\gamma_{2j+1}$. We illustrate the connection in Figure 4.1. While BICGSTAB implicitly constructs $\rho_{j+1}$ by enforcing a biorthogonality condition on a polynomial that lies on the line determined by $\rho_j$ and $t\,\sigma_j$, IDR$(1)$ first generates by the second recursion in (4.6) the polynomial $\widehat{\rho}_{j+1}$ that lies on that line and then also enforces this condition.

Let us finally note that the parameter $\omega_j$, which is in (3.7) chosen to make $\mathbf{r}_{2j}$ orthogonal to $\mathbf{A}\mathbf{v}_{2j-1}$, is indeed the same in IDR$(1)$ and BICGSTAB, since $\mathbf{v}_{2j-1}$ is the same in both methods.

**4.3. How does the original IDR differ from IDR$(1)$?** In contrast to IDR$(1)$ of [33], where (4.2) holds for all $n > 1$, for $n$ odd, the original IDR of [40] used the recursions

$$\begin{aligned} \mathbf{v}_n &:= \mathbf{r}_n - \gamma'_n(\mathbf{r}_{n-1} - \mathbf{r}_{n-2}), & \mathbf{x}'_n &:= \mathbf{x}_n - \gamma'_n(\mathbf{x}_{n-1} - \mathbf{x}_{n-2}), \\ \mathbf{r}_{n+1} &:= (\mathbf{I} - \omega_j \mathbf{A})\,\mathbf{v}_n, & \mathbf{x}_{n+1} &:= \mathbf{x}'_n + \omega_j \mathbf{v}_n, \end{aligned} \tag{4.10}$$

with $\gamma'_n :\equiv \langle \mathbf{p}, \mathbf{r}_n\rangle \,/\, \langle \mathbf{p}, \Delta\mathbf{r}_{n-2}\rangle$. So, here, when computing the "intermediate iterate" $\mathbf{x}'_n$, one modifies $\mathbf{x}_n$ by a step in the same direction as has been used in the previous step for modifying $\mathbf{x}_{n-1}$.

Moreover, in contrast to what we have stated here, the new IDR$(s)$ of [33] computes the residual differences actually as $\Delta\mathbf{r}_n = -\mathbf{A}\Delta\mathbf{x}_n$. This couples the recursions for $\mathbf{x}_n$ and $\mathbf{r}_n$ more tightly, and thus reduces the gap between the recursively computed residual and the true residual. This gap is known to be closely linked to the attainable accuracy that can be achieved with a Krylov space solver; see [9, 15, 27].

The IDR Theorem still applies, and still $\mathbf{x}_{2j} = \mathbf{x}_j^{\text{BiCGSTAB}}$. This follows from the fact that the arguments of Subsection 3.3 are still applicable.

**5. IDR$(s)$ with locally biorthogonal residuals.** Recently, van Gijzen and Sonneveld [38] came up with a new version of IDR$(s)$, in which, in the regular case assumed throughout this section, each of $s$ consecutive "intermediate" residuals $\mathbf{r}_{n_j+k}$ is orthogonal to a growing subset of the $s$ prescribed columns $\mathbf{p}_k$ of $\mathbf{P}$:

$$\mathbf{r}_{n_j+k} \perp \{\mathbf{p}_1, \ldots, \mathbf{p}_k\}, \quad k = 1, \ldots, s. \tag{5.1}$$

For clarity, we will call this algorithm IDR$(s)$BIO here.[11]

[11] Actually, the sets $\{\mathbf{r}_{n_j+k}\}$ and $\{\mathbf{p}_i\}$ are not biorthogonal, but by a triangular linear transformation we could replace the basis $\{\mathbf{p}_i\}$ of $\mathcal{S}^\perp$ by $\{\mathbf{p}'_i\}$ so that $\{\mathbf{r}_{n_j+k}\}$ and $\{\mathbf{p}'_i\}$ are biorthogonal. However, the transformation would depend on $j$.

IDR($s$)BIO still fits into the IDR framework in the sense that the IDR Theorem 3.1 and the orthogonality result of Theorem 3.2 as well as its Corollary 3.3 still apply. One important aspect where it differs from the original IDR($s$) is in the ansatz for recursively constructing $\mathbf{r}_{n_j+k}$ from previous residuals: while the original version uses in the formula (3.4) for $\mathbf{v}_n$ the latest $s$ residual differences (i.e., the choice $\iota(n) = s$) for all $n$, in IDR($s$)BIO that sum involves the $s$ residual differences

$$\begin{aligned}
\Delta\mathbf{r}_{n_{j-1}} \equiv \Delta\mathbf{r}_{n_j-s-1} \quad &\in \mathcal{G}_{j-1} \cap \mathbf{A}\mathcal{K}_{n_{j-1}+1}, \\
&\vdots \\
\Delta\mathbf{r}_{n_{j-1}+s-1} \equiv \Delta\mathbf{r}_{n_j-2} \quad &\in \mathcal{G}_{j-1} \cap \mathbf{A}\mathcal{K}_{n_j-1},
\end{aligned}$$

none of which relates to a residual that lies in $\mathcal{G}_j$ already. So, in the case $s = 1$, there is an analogy to the original IDR of Sonneveld [40]; see (4.10). Additionally, these residual differences are actually replaced by another set of $s$ vectors

$$\begin{aligned}
\mathbf{g}_{n_{j-1}} \equiv \mathbf{g}_{n_j-s-1} \quad &\in \mathcal{G}_{j-1} \cap \mathbf{A}\mathcal{K}_{n_{j-1}+1}, \\
&\vdots \\
\mathbf{g}_{n_{j-1}+s-1} \equiv \mathbf{g}_{n_j-2} \quad &\in \mathcal{G}_{j-1} \cap \mathbf{A}\mathcal{K}_{n_j-1},
\end{aligned}$$

that are multiples of the residual differences and thus also orthogonal to a growing subset of the $s$ prescribed columns $\mathbf{p}_k$:

$$\mathbf{g}_{n_{j-1}+k} \perp \{\mathbf{p}_1, \ldots, \mathbf{p}_k\}, \quad k = 0, \ldots, s-1. \tag{5.2}$$

However, these residual differences are not defined as before, but undergo a linear transformation to impose (5.2). Note that in (5.2), the range of the index $k$ is shifted by 1; so for $k = 0$, the orthogonality condition is empty.

To construct preliminary vectors in $\mathcal{G}_j$, we now define, for $n = n_j + k$ ($k = 0, \ldots, s$), vectors $\mathbf{v}_n \in \mathcal{S}$ by the ansatz

$$\mathbf{v}_n := \mathbf{r}_n - \sum_{i=1}^{s} \gamma_i^{(n)} \mathbf{g}_{n_{j-1}+i-1} = \mathbf{r}_n - \mathbf{G}_{j-1}\mathbf{c}_n, \tag{5.3}$$

where

$$\mathbf{G}_{j-1} :\equiv \left[\begin{array}{ccc} \mathbf{g}_{n_{j-1}} & \ldots & \mathbf{g}_{n_j-1} \end{array}\right], \qquad \mathbf{c}_n :\equiv \left[\begin{array}{ccc} \gamma_1^{(n)} & \ldots & \gamma_s^{(n)} \end{array}\right].$$

$\mathbf{c}_n$ is determined by the condition $\mathbf{v}_n \perp \mathcal{R}(\mathbf{P})$. So we have, as in (3.4) and (3.5),

$$\mathbf{c}_n := \left(\mathbf{P}^\star \mathbf{G}_{j-1}\right)^{-1} \mathbf{P}^\star \mathbf{r}_n, \qquad \mathbf{v}_n := \mathbf{r}_n - \mathbf{G}_{j-1}\,\mathbf{c}_n. \tag{5.4}$$

Here, the projection along $\mathcal{S}$ is on $\mathcal{R}(\mathbf{G}_{j-1})$, that is, on a space that only depends on $j-1$ and therefore is the same for $s+1$ values of $n$. Consequently, the matrix $\mathbf{P}^\star\,\mathbf{G}_{j-1}$ in the $s+1$ linear systems for $\mathbf{c}_n$ is also the same. (However, the systems cannot be solved at once, because the vector $\mathbf{r}_n$ in the right-hand side $\mathbf{P}^\star\mathbf{r}_n$ results from the previous system.)

The elegance of IDR($s$)BIO comes from special features that result from the imposed orthogonality conditions (5.1) and (5.2). Due to (5.2), the matrix

$$\mathbf{M}_{j-1} \equiv \{\mu_{i,k'}^{(j-1)}\}_{i,k'=1}^{s} \equiv \left[\begin{array}{ccc} \mathbf{m}_{n_{j-1}} & \ldots & \mathbf{m}_{n_{j-1}+s-1} \end{array}\right] :\equiv \mathbf{P}^\star\,\mathbf{G}_{j-1} \tag{5.5}$$

is lower triangular, and due (5.1) the matrix with the $s$ right-hand sides $\mathbf{P}^\star \mathbf{r}_n$ $(n = n_j, \ldots, n_j + s - 1)$,

$$\mathbf{F}_j \equiv \{\phi_{i,k'}^{(j)}\}_{i,k=1}^{s} \equiv \begin{bmatrix} \mathbf{f}_{n_j} & \ldots & \mathbf{f}_{n_j+s-1} \end{bmatrix} :\equiv \mathbf{P}^\star \begin{bmatrix} \mathbf{r}_{n_j} & \ldots & \mathbf{r}_{n_j+s-1} \end{bmatrix} \tag{5.6}$$

is lower triangular, too. Consequently, the matrix with the $s$ solutions $\mathbf{c}_n$ of $\mathbf{M}_j \mathbf{c}_n = \mathbf{f}_n$ for $n = n_j, \ldots, n_j + s - 1$,

$$\mathbf{C}_j \equiv \{\gamma_i^{(n_j+k'-1)}\}_{i,k'=1}^{s} \equiv \begin{bmatrix} \mathbf{c}_{n_j} & \ldots & \mathbf{c}_{n_j+s-1} \end{bmatrix} :\equiv \mathbf{M}_j^{-1} \mathbf{F}_j \tag{5.7}$$

is also lower triangular. So its $k'$th column $\mathbf{c}_{n_j+k'-1}$ only depends on the $(s-k'+1)$th trailing principal submatrix (of order $s - k' + 1$) of $\mathbf{M}_{j-1}$, whereas its first $k' - 1$ entries are zero. In other words, the possibly nonzero entries of $\mathbf{c}_{n_j+k'-1}$ result from a $(s - k') \times (s - k')$ linear system. This means in particular that the recursion (5.3) becomes shorter while $k$ increases: for[12] $n = n_j + k$, $k = 0, \ldots, s - 1$,

$$\mathbf{v}_n := \mathbf{r}_n - \sum_{i=k+1}^{s} \gamma_i^{(n)} \mathbf{g}_{n_{j-1}+i-1} = \mathbf{r}_n - \mathbf{G}_{j-1} \mathbf{c}_n. \tag{5.8}$$

This not only reduces the computational cost, but it allows us to overwrite $\mathbf{G}_{j-1}$ by $\mathbf{G}_j$ and $\mathbf{M}_j$ by $\mathbf{M}_{j+1}$ inside the loop over $k$; for details, see the pseudocode in [38].

We still need to explain how we find $s + 1$ residuals $\mathbf{r}_{n_j+k} \in \mathcal{G}_j$ $(k = 0, \ldots, s)$ so that (5.1) holds for the last $s$ of them, and how we construct a new set of $s$ vectors $\mathbf{g}_{n_j+k} \in \mathcal{G}_j$ $(k = 0, \ldots, s - 1)$ satisfying (5.2). We may use the orthogonality conditions (5.1) with $j$ replaced by $j - 1$ and (5.2) as induction assumption. For the initialization ($j = 0$), such sets can be constructed by a one-sided Lanczos process that combines the generation of a basis for $\mathcal{K}_{s+1}$ with orthogonalization with respect to the columns of $\mathbf{P}$. Of course, at the same time, approximate solutions $\mathbf{x}_1, \ldots \mathbf{x}_s$ need to be constructed, too, but they are obtained using essentially the same recursions.

Among the $s + 1$ residuals in $\mathcal{G}_{j-1}$ satisfying the orthogonality condition, the last one, $\mathbf{r}_{n_j-1}$, is orthogonal to all columns of $\mathbf{P}$, whence $\mathbf{r}_{n_j-1} \in \mathcal{S}$. So, in accordance with (5.8) for $k = s$, where the sum is empty, we can choose $\mathbf{v}_{n_j-1} = \mathbf{r}_{n_j-1}$, and thus

$$\mathbf{r}_{n_j} := (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{r}_{n_j-1}. \tag{5.9}$$

Next, for $n = n_j + k > n_j$, any $\mathbf{v}_n$ obtained from (5.8) lies in $\mathcal{G}_{j-1} \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$. So, by the recursive definition of $\mathcal{G}_j$,

$$\widetilde{\mathbf{r}}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_n = \mathbf{r}_n - \mathbf{G}_{j-1} \mathbf{c}_n - \omega_j \mathbf{A} \mathbf{v}_n$$

is a tentative residual in $\mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n+1})$. Since $\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$,

$$\widetilde{\mathbf{g}}_n :\equiv \mathbf{r}_n - \widetilde{\mathbf{r}}_{n+1} = \mathbf{G}_{j-1} \mathbf{c}_n + \omega_j \mathbf{A} \mathbf{v}_n \quad \in \mathcal{G}_j \cap \mathbf{A}\mathcal{K}_{n+1}, \tag{5.10}$$

too; but in order to serve as a column of $\mathbf{G}_j$, it needs to be replaced by $\mathbf{g}_n$ satisfying the orthogonality condition

$$\mathbf{g}_n \perp \{\mathbf{p}_1, \ldots, \mathbf{p}_k\}, \quad n = n_j + k, \quad k = 0, \ldots, s - 1. \tag{5.11}$$

[12]Note that in the presentation of this and other formulas in [38] the notation $v_{n+k}$ with $n = n_{j+1} - 1$ $(k = 1, \ldots, s)$ is used, while here $n = n_j + k$ $(k = 0, \ldots, s - 1)$ and $k' = k + 1$.

This can be achieved by applying a Gram-Schmidt-like process: recursively, the projection of $\widetilde{\mathbf{g}}_n \equiv \widetilde{\mathbf{g}}_{n_j+k}$ on the span of $\mathbf{g}_{n_j}, \ldots, \mathbf{g}_{n-1}$ along the span of $\mathbf{p}_1, \ldots, \mathbf{p}_k$ is subtracted from $\widetilde{\mathbf{g}}_n$ to yield $\mathbf{g}_n \equiv \mathbf{g}_{n_j+k} \in \mathcal{G}_j \cap \mathbf{A}\mathcal{K}_{n+1}$. This can be expressed as follows: for $k = 0$ the condition (5.11) is empty, so $\mathbf{g}_{n_j} := \widetilde{\mathbf{g}}_{n_j}$; then, for $n = n_j + k,\ k = 0, \ldots, s-1$,

$$\mathbf{g}_n := \widetilde{\mathbf{g}}_n - \sum_{i=1}^{k} \alpha_{i,k}^{(j)} \mathbf{g}_{n_j+i-1} = \widetilde{\mathbf{g}}_n - \begin{bmatrix} \mathbf{g}_{n_j} & \ldots & \mathbf{g}_{n-1} \end{bmatrix} \mathbf{a}_k^{(j)}, \tag{5.12}$$

where

$$\mathbf{a}_k^{(j)} :\equiv \left( \begin{bmatrix} \mathbf{p}_1 & \ldots \mathbf{p}_k \end{bmatrix}^\star \begin{bmatrix} \mathbf{g}_{n_j} & \ldots & \mathbf{g}_{n-1} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{p}_1 & \ldots \mathbf{p}_k \end{bmatrix}^\star \widetilde{\mathbf{g}}_n \in \mathbb{C}^k. \tag{5.13}$$

Here, the $k \times k$ matrix $\begin{bmatrix} \mathbf{p}_1 & \ldots \mathbf{p}_k \end{bmatrix}^\star \begin{bmatrix} \mathbf{g}_{n_j} & \ldots & \mathbf{g}_{n-1} \end{bmatrix}$ is the $k$th leading principal submatrix of $\mathbf{P}^\star \mathbf{G}_j = \mathbf{M}_j$, and thus it is lower triangular. Therefore, $\mathbf{a}_k^{(j)}$ can be found by forward substitution. Of course, the diagonal elements $\mu_{i,i}^{(j)} = \mathbf{p}_i^\star \mathbf{g}_{n_j-1+i}$ need to be nonzero. Otherwise the process breaks down and the orthogonality condition (5.11) cannot be satisfied for some $n$. For the whole block this step is summarized by

$$\widetilde{\mathbf{G}}_j :\equiv \begin{bmatrix} \widetilde{\mathbf{g}}_{n_j} & \ldots & \widetilde{\mathbf{g}}_{n_j+s-1} \end{bmatrix} = \mathbf{G}_j \mathbf{A}_j^\triangledown \tag{5.14}$$

with $\mathbf{A}_j^\triangledown$ unit upper triangular and $\mathbf{P}^\star \mathbf{G}_j = \mathbf{M}_j$ lower triangular. Above the diagonal $\mathbf{A}_j^\triangledown$ contains the $s-1$ coefficients vectors $\mathbf{a}_1^{(j)}$ (in column 2) to $\mathbf{a}_{s-1}^{(j)}$ (in the last column). Hence, $\mathbf{M}_j \mathbf{A}_j^\triangledown = (\mathbf{P}^\star \mathbf{G}_j) \mathbf{A}_j^\triangledown$ is an LU decomposition of $\widetilde{\mathbf{M}}_j :\equiv \mathbf{P}^\star \widetilde{\mathbf{G}}_j$.

Amazingly, when we replace this classical Gram-Schmidt-like process by a modified Gram-Schmidt-like process as suggested in [38], there is no need to solve triangular linear systems.

In matrix notation the first sweep of the modified Gram-Schmidt-like process can be summarized as

$$\mathbf{G}_j^{(1)} :\equiv \widetilde{\mathbf{G}}_j \mathbf{B}_j^{(1)},$$

where the upper triangular matrix $\mathbf{B}_j^{(1)}$ is given by

$$\mathbf{B}_j^{(1)} :\equiv \begin{bmatrix} 1 & -\beta_{12}^{(j)} & -\beta_{13}^{(j)} & \ldots & -\beta_{1s}^{(j)} \\ & 1 & 0 & \ldots & 0 \\ & & \ddots & & \vdots \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix}, \qquad \beta_{1,k+1}^{(j)} :\equiv \frac{\langle \mathbf{p}_1, \widetilde{\mathbf{g}}_{n_j+k} \rangle}{\langle \mathbf{p}_1, \widetilde{\mathbf{g}}_{n_j} \rangle}$$

(with $k = 1, \ldots, s-1$), and has the effect that, by subtracting a multiple of the first column $\widetilde{\mathbf{g}}_{n_j}$ of $\widetilde{\mathbf{G}}_j$, columns 2 to $s$ of $\widetilde{\mathbf{G}}_j$ are transformed into columns of $\mathbf{G}_j^{(1)}$ that are orthogonal to $\mathbf{p}_1$. Then, for $\ell = 2, \ldots, s-1$, in further analogous sweeps, by subtracting a multiple of the $\ell$th column $\mathbf{g}_{n_j+\ell-1}^{(\ell-1)}$ of $\mathbf{G}_j^{(\ell-1)}$, columns $\ell+1$ to $s$ of $\mathbf{G}_j^{(\ell-1)}$ are transformed into columns of $\mathbf{G}_j^{(\ell)}$ that are additionally orthogonal to $\mathbf{p}_\ell$:

$$\mathbf{G}_j^{(\ell)} :\equiv \mathbf{G}_j^{(\ell-1)} \mathbf{B}_j^{(\ell)}, \qquad \ell = 2, \ldots, s-1,$$

where $\mathbf{B}_j^{(\ell)}$ for $\ell = 2, \ldots, s-1$ is given by

$$\mathbf{B}_j^{(\ell)} :\equiv \begin{bmatrix} 1 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\ & \ddots & & & & & \\ & & 1 & -\beta_{\ell,\ell+1}^{(j)} & \ldots & \ldots & -\beta_{\ell,s}^{(j)} \\ & & & 1 & 0 & \ldots & 0 \\ & & & & \ddots & & \vdots \\ & & & & & 1 & 0 \\ & & & & & & 1 \end{bmatrix}, \qquad \beta_{\ell,k+1}^{(j)} :\equiv \frac{\left\langle \mathbf{p}_\ell, \mathbf{g}_{n_j+k}^{(\ell-1)} \right\rangle}{\left\langle \mathbf{p}_\ell, \mathbf{g}_{n_j+\ell-1}^{(\ell-1)} \right\rangle}$$

(with $k = \ell, \ldots, s-1$). Ultimately, the columns of

$$\mathbf{G}_j :\equiv \mathbf{G}_j^{(s)} = \widetilde{\mathbf{G}}_j \mathbf{B}_j^{(1)} \mathbf{B}_j^{(2)} \cdots \mathbf{B}_j^{(s-1)} \tag{5.15}$$

satisfy the orthogonality condition (5.11), and thus are identical to those obtained by (5.12). Moreover, the comparison with (5.14) reveals that

$$\mathbf{A}_j^{\triangledown} = \left( \mathbf{B}_j^{(1)} \mathbf{B}_j^{(2)} \cdots \mathbf{B}_j^{(s-1)} \right)^{-1} = \widehat{\mathbf{B}}_j^{(s-1)} \widehat{\mathbf{B}}_j^{(s-2)} \cdots \widehat{\mathbf{B}}_j^{(1)}, \tag{5.16}$$

where $\widehat{\mathbf{B}}_j^{(\ell)} :\equiv (\mathbf{B}_j^{(\ell)})^{-1}$ is obtained by replacing in $\mathbf{B}_j^{(\ell)}$ the coefficients $-\beta_{\ell,k+1}^{(j)}$ by $+\beta_{\ell,k+1}^{(j)}$. In view of the special structure of the matrices $\widehat{\mathbf{B}}_j^{(\ell)}$, one can conclude from (5.16) that $\beta_{\ell,k+1}^{(j)} = \alpha_{\ell,k}^{(j)}$. Of course, the matrices $\widetilde{\mathbf{G}}_j, \mathbf{G}_j^{(1)}, \ldots, \mathbf{G}_j^{(s-1)}, \mathbf{G}_j$ can all be stored in the same place.

Finally, since by induction (within a block) both $\mathbf{r}_n$ and $\mathbf{g}_n$ are orthogonal to $\mathbf{p}_1, \ldots, \mathbf{p}_k$, and moreover, $\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n$ and $\mathbf{g}_n \in \mathbf{A}\mathcal{K}_{n+1}$, we get $\mathbf{r}_{n+1} \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n+1}$ satisfying (5.1) according to

$$\mathbf{r}_{n+1} :\equiv \mathbf{r}_n - \frac{\phi_{k+1,k+1}^{(j)}}{\mu_{k+1,k+1}^{(j)}} \mathbf{g}_n \qquad (n = n_j + k;\ k = 0, \ldots, s-1), \tag{5.17}$$

where $\phi_{k,k}^{(j)}$ and $\mu_{k,k}^{(j)}$ are diagonal elements of $\mathbf{F}_j$ and $\mathbf{M}_j$, which enforce that $\mathbf{r}_{n+1} \perp \mathbf{p}_{k+1}$, as can be checked easily.

As has been noted in [38] and can be seen by premultiplying (5.17) with $\mathbf{P}^\star$, the columns of $\mathbf{F}_j$ can be updated in an elegant way, too:

$$\mathbf{f}_{n+1} :\equiv \mathbf{f}_n - \frac{\phi_{k+1,k+1}^{(j)}}{\mu_{k+1,k+1}^{(j)}} \mathbf{m}_{n+1} \qquad (n = n_j + k,\ k = 0, \ldots, s-1). \tag{5.18}$$

So far, we have concentrated on the possibility of constructing efficiently residuals satisfying the orthogonality properties (5.1), but we still need to give formulas for the recursive computation of the approximate solutions $\mathbf{x}_n$; and due to the relation $\Delta\mathbf{r}_n = -\mathbf{A}\Delta\mathbf{x}_n$, these formulas will lead to other options for updating the residuals.

In general, update formulas for $\mathbf{x}_n$ are fairly easily obtained from those for $\mathbf{r}_n$, and here this is true also. Let us define

$$\begin{aligned} \widetilde{\mathbf{u}}_n &:\equiv \mathbf{A}^{-1} \widetilde{\mathbf{g}}_n, \\ \mathbf{u}_n &:\equiv \mathbf{A}^{-1} \mathbf{g}_n, \\ \mathbf{U}_{j-1} &:\equiv \begin{bmatrix} \mathbf{u}_{n_{j-1}} & \ldots & \mathbf{u}_{n_j - 1} \end{bmatrix} = \mathbf{A}^{-1} \mathbf{G}_{j-1}. \end{aligned}$$

Then, from (5.10), we get

$$\widetilde{\mathbf{u}}_n := \mathbf{U}_{j-1}\mathbf{c}_n + \omega_j \mathbf{v}_n \quad \in \mathcal{K}_{n+1}, \tag{5.19}$$

which allows us to replace (5.10) by

$$\widetilde{\mathbf{g}}_n := \mathbf{A}\widetilde{\mathbf{u}}_n. \tag{5.20}$$

This helps to couple the updates of $\mathbf{x}_n$ and $\mathbf{r}_n$ and thus to avoid a fast growth of the residual gap mentioned before. Moreover, (5.12) translates into

$$\mathbf{u}_n := \widetilde{\mathbf{u}}_n - \sum_{i=1}^{k} \alpha_{i,k}^{(j)} \mathbf{u}_{n_j+i-1} \qquad (n = n_j + k,\ k = 1, \ldots, s-1). \tag{5.21}$$

Finally, from (5.17) we get

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \frac{\phi_{k+1,k+1}^{(j)}}{\mu_{k+1,k+1}^{(j)}} \mathbf{u}_n \qquad (n = n_j + k,\ k = 0, \ldots, s-1). \tag{5.22}$$

These are the formulas on which the IDR pseudocode in [38] is based. But what makes this formulation so ingenious is the fact that it minimizes memory usage by systematically overwriting data that is no longer used. On the other hand, this makes the code harder to understand.

**6. Comments and conclusions.** The various IDR algorithms may still be not as well understood as other algorithms that are directly derived from the Lanczos process (be it symmetric or non-symmetric), but their close relationship to Lanczos-based algorithms certainly helps us to understand them.

**6.1. The case $s = 1$.** This case is easy, because in exact arithmetic the even-indexed IDR$(1)$ iterates and residuals are exactly the BiCGStab iterates and residuals. However, as we have seen, the recursions are not the same, and therefore, it is possible that IDR$(1)$ is more stable than BiCGStab or vice versa. The odd-numbered IDR$(1)$ iterates and residuals have no counterpart in the original BiCGStab algorithm.

The existence of all BiCGStab iterates, i.e., all even IDR$(1)$ iterates, requires that all BiCG residuals exists, and therefore any serious Lanczos breakdown and any so-called pivot breakdown cause BiCGStab and IDR$(1)$ to break down unless extra precautions against such breakdowns have been implemented. This follows from the fact that the BiOMin version and the BiORes version of BiCG break down at the same times: the existence of the residuals implies the existence of the coupled two-term recursions [13]. Additionally, the choice of the parameters $\omega_j$ is the same in BiCGStab and IDR$(1)$, so a breakdown due to $\omega_j = 0$ will occur at the same time in both methods; but in both it is also easy to fix. It is conceivable that there are further causes for breakdown in IDR$(1)$. On the other hand, the recovery procedure in case of a breakdown seems to be much simpler in IDR$(1)$; but so far, it seems to be less understood and less documented than for BiCGStab [14].

While IDR$(1)$ and BiCGStab produce in exact arithmetic essentially the same results based on a common mathematical background, they are clearly different algorithms obtained by different approaches.[13]

[13] From Proposition 5.1 in [25] one may get the impression that BiCGStab and IDR$(1)$ are nearly identical. But the authors do not compare the original BiCGStab recursions with the original IDR$(1)$ recursions.

**6.2. The case $s > 1$.** The relation BICG $\rightsquigarrow$ BICGSTAB $\sim$ IDR$(1)$ is matched by the relation ML$(s)$BICG $\rightsquigarrow$ ML$(s)$BICGSTAB $\sim$ IDR$(s)$, but the similarity between ML$(s)$BICGSTAB and IDR$(s)$ is weaker than between BICGSTAB and IDR$(1)$. In IDR$(s)$ there is some freedom in choosing the $s$ "intermediate" iterates and residuals because, unlike in ML$(s)$BICGSTAB, the vectors $\mathbf{w}_n$ in (3.13) need not satisfy the strict condition (2.6) of an "intermediate" ML$(s)$BICG residual, but only the weaker block orthogonality condition $\mathbf{w}_n \perp \mathcal{K}_j(\mathbf{A}^\star, \mathbf{P})$ of (3.13). With the IDR approach, such vectors can be obtained with much simpler recursions, which, in addition, allow considerable flexibility that may enable us to overcome breakdowns.

The many published numerical examples of IDR$(s)$ [25, 33, 38] show the fast convergence of this method and the superiority of the choice $s > 1$. Due to the careful choice of the numerical examples and the restriction to small values of $s$, where the method is less costly than for large $s$, the numerical results are more relevant than those of Yeung and Chan [41], who applied their ML$(s)$BICGSTAB with large $s$ and mostly without preconditioning to rather ill-conditioned test matrices. Heuristically, it is plausible that these methods are particularly effective for such examples. In BICG, the basis of $\widetilde{\mathcal{K}}_n$ tends to be somewhat ill-conditioned if $\mathbf{A}$ is far from Hermitian. Moreover, its construction is prone to loss of biorthogonality, which can be expected to be stronger when $A$ is ill-conditioned and $n$ is large. When constructing in ML$(s)$BICG a basis of the block Krylov subspace $\mathcal{K}_j(\mathbf{A}^\star, \widetilde{\mathbf{R}}_0)$, we can start with $s$ orthonormal basis vectors and work with the dimension $j$ of order $n/s$ of each single Krylov subspace. Then, for the same value of $n$, we can expect the basis of the block Krylov subspace to be better conditioned. A similar improvement of the condition of the basis can be expected when we compare BICGSTAB to ML$(s)$BICGSTAB or IDR$(s)$, and it seems to be relevant also in preconditioned problems that are not extremely ill-conditioned.

The effectiveness of this improvement due to changing the dual ("left") space and using a better conditioned basis is quite surprising. The discovery of this effect is due to Yeung and Chan [41], but the equally strong improvement of IDR$(s)$ over IDR$(1)$, which seems to rely partly on the same effect, was discovered independently a decade later by Sonneveld and van Gijzen. Additionally, the left-hand side block Krylov space that is implicitly used by both ML$(s)$BICGSTAB and IDR$(s)$ seems to be more effective in capturing the spectrum. By choosing in IDR$(s)$ this space (i.e., the matrix $\mathbf{P}$) appropriately — and perhaps even adaptively — depending on some knowledge on the spectrum of $\mathbf{A}$, one may be able to further speed up convergence.

The other fundamental fact is that in the framework of Lanczos-type product methods, multiple left projections can reduce the MV count. By the reduction of the MV count we understand a smaller ratio between the search space dimension $n$ ($n_j \leq n < n_{j+1}$) and the number $js$ of orthogonality conditions satisfied by $\mathbf{w}_n$. For $n = n_j = j(s+1)$, this ratio is $1 + \frac{1}{s}$, while for CGS and BICGSTAB it is 2. This also applies both to ML$(s)$BICGSTAB and IDR$(s)$, but not to ML$(s)$BICG, where building up the left block Krylov space costs $s$ MVs per value of $j$, while ML$(s)$BICGSTAB and IDR$(s)$ achieve the same effect with just one MV. Therefore, ML$(s)$BICG is not competitive with ML$(s)$BICGSTAB or IDR$(s)$, except perhaps in situations where the Lanczos-type product methods fail due to roundoff problems. (It is well known that the recursion coefficients produced by BICGSTAB are usually less accurate than the same coefficients produced by BICG, and there are problems where BICGSTAB fails to converge, while BICG succeeds.) For this reason, Yeung and Chan [41] introduced ML$(s)$BICG only as a tool for deriving ML$(s)$BICGSTAB.

IDR$(s)$ also inherits from BICGSTAB the disadvantage that for a problem with real-valued data, the parameters $\omega_j$ are all real-valued (when chosen in the standard way), and therefore the zeros $\omega_i^{-1}$ of $\Omega_j$ cannot approximate complex eigenvalues of $\mathbf{A}$ well. This problem has been addressed by BiCGStab2 [12] and later by BiCGStab($\ell$) [24, 27] by building up $\Omega_j$ from polynomial factors of degree 2 and $\ell$, respectively. An adaptation of IDR($s$) to include this idea in an efficient way is not so straightforward and requires a modifiction to the framework. This topic is addressed in [28] and [35], which were published while this paper was reviewed.

**Acknowledgment.** The author acknowledges the advice of and the discussions with many colleagues, in particular Michael Eiermann, Klaus Gärtner, Martin van Gijzen, Miroslav Rozložník, Valeria Simoncini, Peter Sonneveld, Man-Chung Yeung, and Jens-Peter Zemke. He is also indebted to the Technical University Berlin and the DFG Research Center MATHEON for hosting him.

REFERENCES

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, *A Lanczos-type method for multiple starting vectors*, Math. Comp., 69 (2000), pp. 1577–1601.
[2] O. Axelsson, *Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations*, Linear Algebra Appl., 29 (1980), pp. 1–16.
[3] Z. Bai, D. Day, and Q. Ye, *ABLE: an adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082.
[4] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Dundee, 1975, G. A. Watson, ed., vol. 506 of Lecture Notes in Mathematics, Springer, Berlin, 1976, pp. 73–89.
[5] R. W. Freund, *Band Lanczos method (Section 4.6)*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 80–88.
[6] ———, *Band Lanczos method (Section 7.10)*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 205–216.
[7] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.
[8] R. W. Freund and M. Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., 254 (1997), pp. 119–157.
[9] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.
[10] M. H. Gutknecht, *Stationary and almost stationary iterative $(k,l)$-step methods for linear and nonlinear systems of equations*, Numer. Math., 56 (1989), pp. 179–213.
[11] ———, *The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, and the qd algorithm*, in Preliminary Proceedings of the Copper Mountain Conference on Iterative Methods, April 1990. http://www.sam.math.ethz.ch/~mhg/pub/CopperMtn90.ps.gz and http://www.sam.math.ethz.ch/~mhg/pub/CopperMtn90-7.ps.gz.
[12] ———, *Variants of BiCGStab for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033.
[13] ———, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numerica, 6 (1997), pp. 271–397.
[14] M. H. Gutknecht and K. J. Ressel, *Look-ahead procedures for Lanczos-type product methods based on three-term Lanczos recurrences*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1051–1078.
[15] M. H. Gutknecht and Z. Strakoš, *Accuracy of two three-term and three two-term recurrences for Krylov space solvers*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 213–229.
[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bureau Standards, 49 (1952), pp. 409–435.
[17] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bureau Standards, 45 (1950), pp. 255–281.
[18] ———, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bureau Standards, 49 (1952), pp. 33–53.
[19] D. Loher, *Reliable nonsymmetric block Lanczos algorithms*, PhD thesis, Diss. no. 16337, ETH Zurich, Zurich, Switzerland, 2006.
[20] D. P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.
[21] J. K. Reid, *On the method of conjugate gradients for the solution of large sparse systems of linear equations*,

in Large Sparse Sets of Linear Equations, Proceedings of the Oxford Conference of the Institute of Mathematics and its Applications held in April, 1970, J. K. Reid, ed., Academic Press, London, 1971, pp. 231–254.

[22] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[23] V. SIMONCINI, *A stabilized QMR version of block BICG*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 419–434.

[24] G. L. G. SLEIJPEN AND D. R. FOKKEMA, *BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electronic Trans. Numer. Anal., 1 (1993), pp. 11–32. http://etna.math.kent.edu/vol.1.1993/pp11-32.dir/.

[25] G. L. G. SLEIJPEN, P. SONNEVELD, AND M. B. VAN GIJZEN, *Bi-CGSTAB as an induced dimension reduction method*, Report 08-07, Department of Applied Mathematical Analysis, Delft University of Technology, 2008.

[26] G. L. G. SLEIJPEN AND H. A. VAN DER VORST, *An overview of approaches for the stable computation of hybrid BiCG methods*, Appl. Numer. Math., 19 (1995), pp. 235–254.

[27] G. L. G. SLEIJPEN, H. A. VAN DER VORST, AND D. R. FOKKEMA, *BiCGstab(l) and other hybrid Bi-CG methods*, Numer. Algorithms, 7 (1994), pp. 75–109.

[28] G. L. G. SLEIJPEN AND M. B. VAN GIJZEN, *Exploiting BiCGstab($\ell$) strategies to induce dimension reduction*, Report 09-02, Department of Applied Mathematical Analysis, Delft University of Technology, 2009.

[29] P. SONNEVELD, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, Report 84-16, Department of Mathematics and Informatics, Delft University of Technology, 1984.

[30] P. SONNEVELD, private communication, 2009.

[31] ———, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

[32] P. SONNEVELD AND M. B. VAN GIJZEN, *IDR($s$): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, Report 07-07, Department of Applied Mathematical Analysis, Delft University of Technology, 2007.

[33] P. SONNEVELD AND M. B. VAN GIJZEN, *IDR($s$): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Comput., 31 (2008), pp. 1035–1062.

[34] E. STIEFEL, *Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme*, Comm. Math. Helv., 29 (1955), pp. 157–179.

[35] M. TANIO AND M. SUGIHARA, *GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials*, Tech. Report METR 2009-16, Department of Mathematical Informatics, Graduate School of information Science and Technology, University of Tokyo, April 2009.

[36] H. A. VAN DER VORST, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[37] H. A. VAN DER VORST AND P. SONNEVELD, *CGSTAB, a more smoothly converging variant of CG-S*, Report 90-50, Department of Mathematics and Informatics, Delft University of Technology, 1990.

[38] M. B. VAN GIJZEN AND P. SONNEVELD, *An elegant IDR($s$) variant that efficiently exploits bi-orthogonality properties*, Report 08-21, Department of Applied Mathematical Analysis, Delft University of Technology, 2008.

[39] P. K. W. VINSOME, *ORTHOMIN—an iterative method for solving sparse sets of simultaneous linear equations*, in Proc. Fourth SPE Symposium on Reservoir Simulation, Los Angeles, 1976, pp. 149–160.

[40] P. WESSELING AND P. SONNEVELD, *Numerical experiments with a multiple grid and a preconditioned Lanczos type method*, in Approximation methods for Navier-Stokes problems (Proc. Sympos., Univ. Paderborn, Paderborn, 1979), vol. 771 of Lecture Notes in Math., Springer, Berlin, 1980, pp. 543–562.

[41] M.-C. YEUNG AND T. F. CHAN, *ML($k$)BiCGSTAB: a BiCGSTAB variant based on multiple Lanczos starting vectors*, SIAM J. Sci. Comput., 21 (1999), pp. 1263–1290.

[42] D. M. YOUNG AND K. C. JEA, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.

[43] J.-P. ZEMKE, private communication, 2006.