DELFT UNIVERSITY OF TECHNOLOGY

REPORT 10-21

Interpreting IDR(s) as a Deflation Method

T. P. Collignon, G. L. G. Sleijpen, and M. B. van Gijzen

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2010

Interpreting IDR(s) as a deflation method

T. P. Collignon[*], G. L. G. Sleijpen[†], and M. B. van Gijzen [‡]

October 2010

Abstract

In this paper the IDR(s) method is interpreted in the context of deflation methods. It is shown that IDR(s) can be seen as a Richardson iteration preconditioned by a variable deflation–type preconditioner. The main result of this paper is the *IDR projection theorem*, which relates the spectrum of the deflated system in each IDR(s) cycle to all previous cycles. The theorem shows that this so–called *active spectrum* becomes increasingly more *clustered*. This clustering property may serve as an intuitive explanation for the excellent convergence properties of IDR(s). These remarkable spectral properties exist whilst using a deflation subspace matrix of fixed rank. Variants of explicitly deflated IDR(s) are compared to IDR(s) in which the IDR deflation subspace matrix is augmented with "traditional" deflation vectors. The theoretical results are illustrated by numerical experiments.

Key words. iterative methods, numerical linear algebra, nonsymmetric linear systems, IDR(s), deflation.

[*]Delft Institute of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, the Netherlands. e-mail: t.p.collignon@tudelft.nl

[†]Mathematical Institute, Utrecht University, Budapestlaan 6, De Uithof, Utrecht, the Netherlands. e-mail: g.l.g.sleijpen@uu.nl

[‡]Delft Institute of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, the Netherlands. e-mail: m.b.vangijzen@tudelft.nl

1 Introduction

The recently proposed induced dimension reduction (IDR) method [21] and its variants are short recurrence Krylov subspace methods for iteratively solving large nonsymmetric linear systems

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{C}^{n\times n}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{C}^n. \tag{1}$$

In [7, 16, 21] it is shown that for $s = 1$, IDR(s) is mathematically equivalent to the ubiquitous Bi–CGSTAB [25] method. For important classes of problems and for relatively small values of $s > 1$ (e.g., $s = 4$ or 6), the IDR(s) algorithms outperform Bi–CGSTAB, see for example [21, §6] and [28].

The IDR(s) method has attracted considerable attention and we will give a short overview of some IDR(s) related papers. The IDR approach to solving nonsymmetric linear systems is quite nonstandard and in [7, 16] the connections between the IDR(s) method and more traditional Krylov subspace methods are explained. Several algorithmic variations are proposed and analysed in for example [12, 13, 27, 28], while [3, 26] aims to optimise IDR(s) methods in a parallel and Grid computing context. In [18], the strengths of BiCGstab(ℓ) [19] and IDR(s) are combined, resulting in the superior IDRstab method. A related approach is used in [24]. In [14] the IDR(s) method is interpreted as a Petrov–Galerkin method and in [8] the IDR approach is used for eigenvalues computations. More recently, an in–depth convergence analysis of IDR(s) using statistical arguments was performed in [20]. Several highly interesting results related to the algebra of induced dimension reduction are discussed in [17].

In this paper it is shown in Sect. 2 that the IDR(s) method can be viewed as a so–called *adapted deflation method* [23, §2.3.3]. The method can be seen as a Richardson iteration preconditioned by a variable deflation–type preconditioner, where the preconditioner is updated in each cycle with new spectral information. This interpretation leads to more insight into the structure and the convergence properties of IDR(s) methods. In particular, it leads to the *IDR projection theorem* given in Sect. 3 (Theorem 3.3), which relates the spectrum of the deflated system of a particular cycle to the spectra of the deflated systems of all the previous cycles.

An IDR(s) cycle consists of $s + 1$ (preconditioned) matrix–vector multiplications and in the kth cycle a new smoothing parameter ω_k can be chosen. These parameters play the same role as the ω's in the Bi-CGSTAB method: to obtain smoother convergence behaviour. Note that for Bi-CGSTAB we have $s = 1$ and for this method the ω is therefore updated every other matrix–vector multiplication step.

Assume that $\omega_k \neq 0$ and put $\mu_k \equiv \omega_k^{-1}$. The IDR projection theorem states that in the kth cycle, s eigenvalues of $\mathbf{A}$ are shifted to μ_k. Moreover, each $\nu \in \{\mu_0, \mu_1, \ldots, \mu_{k-1}\}$ is an eigenvalue of the deflated system in cycle k each with geometric multiplicity s. This implies that the spectra of the deflated systems becomes increasingly *clustered* for increasing k. Quite remarkably, this is accomplished using a deflation subspace matrix of fixed rank equal to s. The deflation subspace built in cycle k retains spectral information from all previous IDR(s) cycles. It is argued that the effectiveness of IDR(s) methods comes

from the clustering of the spectra of the deflated systems. Possible consequences of this interpretation are discussed in this paper. With the exception of certain extreme cases, these results also suggest that the value of μ_k itself plays a relatively small role in the convergence process, especially for larger s.

It can be shown that in the generic case, IDR(s) methods compute the exact solution in exact arithmetic within n/s cycles [21, §3]. In this case the spectrum of the deflated system in the final cycle solely consists of n/s eigenvalues each with geometric multiplicity s.

Deflation for iterative methods has been investigated by many authors, see for example [5, 6, 10, 11, 22]. A typical deflation procedure consists of three steps: identifying some particular subspace that hampers convergence, finding a suitable approximation to this space, and removing the influence of this space on the iteration process. Usually, this subspace is the eigenspace corresponding to eigenvalues of $\mathbf{A}$ that are somehow "undesirable".

In Sect. 4 two variants of explicitly deflated IDR(s) methods are compared to IDR(s) in which the IDR deflation matrices are augmented with traditional deflation vectors. Spectral comparisons between the deflated systems of these three approaches are performed.

The IDR(s) method adaptively constructs a deflation–type preconditioner. This is not a new concept: for example, methods such as described in [2, 4, 1] *explicitly* construct deflation vectors based on spectral information gathered by the Arnoldi process during iterations of restarted GMRES(m). The goal there is to approximate invariant subspaces associated with a *specific* set of eigenvalues (e.g., eigenvalues that are small in magnitude). These methods differ in how the approximate invariant subspaces are constructed and how they are incorporated in the iteration process. In general, the dimension of the invariant subspace grows during the iterative process. In order to limit memory cost, the dimension of the invariant subspace has to be fixed, which can reduce the effectiveness of the preconditioner.

In contrast, IDR(s) constructs the deflation preconditioner *implicitly*. Furthermore, the deflation subspace in IDR(s) is unrelated to any specific spectral components of $\mathbf{A}$. Nevertheless, new spectral information seems to be continuously injected in the iteration process, all the while keeping the dimension of the deflation subspace fixed. In this sense IDR(s) can be seen as an efficient deflation–type method.

This paper shows that interpreting IDR(s) as a deflation method has resulted in new insights into the structure of IDR(s) methods and Sect. 5 lists the main conclusions.

Preliminaries

The following notational conventions, terminology, and definitions will be used in this paper.

Notation 1.1. Given $\mathbf{A} \in \mathbb{C}^{n\times n}$, $\text{span}(\mathbf{A})$ denotes the range of $\mathbf{A}$, $\mathcal{N}(\mathbf{A})$ the nullspace of $\mathbf{A}$, $\sigma(\mathbf{A})$ the set of eigenvalues of $\mathbf{A}$, $\text{rank}\,\mathbf{A}$ the rank of $\mathbf{A}$, and $\mathbf{A}^*$ the adjoint of $\mathbf{A}$. The Krylov subspace of order k generated by a matrix $\mathbf{A}$ and a vector $\mathbf{v}$ is denoted by $\mathcal{K}_k(\mathbf{A}, \mathbf{v}) \equiv \text{span}(\mathbf{v}, \mathbf{A}\mathbf{v}, \ldots, \mathbf{A}^{k-1}\mathbf{v})$.

Notation 1.2. If $\mathcal{V}$ is a linear subspace of $\mathbb{C}^n$, then $\dim \mathcal{V}$ is its dimension and an n-vector $\mathbf{v}$ is *orthogonal* to $\mathcal{V}$, $\mathbf{v} \perp \mathcal{V}$, if $\mathbf{v}$ is orthogonal to all $\mathbf{w} \in \mathcal{V}$. The space of all n-vectors $\mathbf{v}$ that are orthogonal to $\mathcal{V}$ is denoted by $\mathcal{V}^\perp$.

If $\widetilde{\mathbf{R}}, \mathbf{V}_1, \ldots, \mathbf{V}_k$ are matrices with column vectors of size n, then $\mathrm{span}(\mathbf{V}_1, \ldots, \mathbf{V}_k)$ is the subspace of $\mathbb{C}^n$ spanned by all columns of all $\mathbf{V}_j$. We put $\mathbf{V}_1, \ldots, \mathbf{V}_k \perp \widetilde{\mathbf{R}}$ if $\mathrm{span}(\mathbf{V}_1, \ldots, \mathbf{V}_k) \subset \mathrm{span}(\widetilde{\mathbf{R}})^\perp$. Then we say that the $\mathbf{V}_1, \ldots, \mathbf{V}_k$ are *orthogonal to* $\widetilde{\mathbf{R}}$. $[\mathbf{V}_1, \ldots, \mathbf{V}_k]$ is the matrix with column vectors the columns of the matrices $\mathbf{V}_j$. We identify n-vectors $\mathbf{v}$ and $n \times 1$ matrices $[\mathbf{v}]$. We call the columns of $\mathbf{V}$ a basis of $\mathrm{span}(\mathbf{V}) \subset \mathbb{C}^n$.

Notation 1.3. If not specified otherwise, the norm $||\cdot||$ denotes the 2–norm. The notation $\mathbf{0}_{n,s}$ denotes the all–zero $n \times s$ matrix.

An MV is a matrix-vector multiplication $\mathbf{A}\mathbf{v}$, where $\mathbf{v}$ is an n-vector. The multiplication $\mathbf{A}\mathbf{V}$ with $\mathbf{V} \in \mathbb{C}^{n \times s}$ requires s MVs. The index k refers to the kth IDR(s) cycle. When not specified otherwise, we assume that the coefficient matrix $\mathbf{A}$ is nonsingular.

2 Relation between IDR and deflation

In Sect. 2.1 IDR methods are discussed, while Sect. 2.2 presents the structure of adapted deflation methods. Section 2.3 unites these two concepts by showing how the IDR method can be interpreted as a deflation method. Lastly, Sect. 2.4 contains some remarks related to IDR *algorithms*.

In the following, let $\widetilde{\mathbf{R}}$ be an $n \times s$ matrix; $\widetilde{\mathbf{R}}$ is the s-dimensional *initial shadow residual* or *IDR test matrix*. It is assumed that the value of s is fixed during the iteration process.

2.1 IDR methods

Induced dimension reduction (IDR) methods iteratively construct residuals in a sequence $(\mathcal{G}_k)$ of shrinking subspaces: in each cycle k, we start from a residual in $\mathcal{G}_k$ and construct a residual in $\mathcal{G}_{k+1}$. Ultimately, the residual is forced in the zero–dimensional subspace $\mathcal{G}_k = \{\mathbf{0}\}$ for some $k \leq n$. These IDR subspaces are recursively defined as follows:

Definition 2.1. Let $\mathcal{G}_0$ be a linear subspace of $\mathbb{C}^n$ such that $\mathbf{A}\mathcal{G}_0 \subset \mathcal{G}_0$ (for example, the *full* Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{v}) \equiv \mathrm{span}\{\mathbf{A}^k\mathbf{v} \mid k = 0, 1, \ldots\}$ for some $\mathbf{v} \in \mathbb{C}^n$). For a sequence (μ_k) in $\mathbb{C}$, let the sequence $(\mathcal{G}_k)$ of *IDR subspaces* be defined by

$$\mathcal{G}_k \equiv (\mathbf{A} - \mu_k \mathbf{I})(\mathcal{G}'_{k-1}), \quad \text{where} \quad \mathcal{G}'_{k-1} \equiv \mathcal{G}_{k-1} \cap \widetilde{\mathbf{R}}^\perp \quad (k \in \mathbb{N}). \tag{2}$$

The following result states that the sequence $(\mathcal{G}_k)$ of IDR subspaces forms a strict chain of nested linear subspaces. For a proof, see [16, 21].

Theorem 2.2 (IDR)**.** *With $\mathcal{G}_k$ as in Def. 2.1, we have for $k, \ell \in \mathbb{N}_0$*

$$\mathcal{G}_k \subset \mathcal{G}_\ell \quad (\ell \leq k).$$

If $\mathbf{A}$ has no eigenvector in $\widetilde{\mathbf{R}}^\perp$, then $\mathcal{G}_k = \mathcal{G}_\ell$ $(\ell < k)$ if and only if $\mathcal{G}_k = \{\mathbf{0}\}$. □

The IDR subspaces $\mathcal{G}_k$ are a special case of a wider class of spaces called *Sonneveld subspaces* defined below.

Definition 2.3 (cf. [18])**.** Let $\mathcal{G}_0$ be as in Def. 2.1. For a polynomial P of exact degree k, the *Sonneveld subspace* $\mathcal{S}(P, \mathbf{A}, \widetilde{\mathbf{R}})$ is defined by

$$\mathcal{S}(P, \mathbf{A}, \widetilde{\mathbf{R}}) \equiv \{P(\mathbf{A})\mathbf{v} \mid \mathbf{v} \in \mathcal{G}_0, \mathbf{v} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})\}, \tag{3}$$

where

$$\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}}) \equiv \left\{ \sum_{j=0}^{k-1} (\mathbf{A}^*)^j \widetilde{\mathbf{R}} \gamma_j \mid \gamma_j \in \mathbb{C}^s \right\} \tag{4}$$

is the (block) *Krylov subspace* of order k (generated by $\mathbf{A}^*$ and $\widetilde{\mathbf{R}}$). Note that the dimension of the Sonneveld subspace $\mathcal{S}(P, \mathbf{A}, \widetilde{\mathbf{R}})$ is in general (*generic case*) equal to $n - ks$. Also, note that

$$\mathbf{v} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}}) \qquad \Leftrightarrow \qquad \widetilde{\mathbf{R}}^* \mathbf{A}^j \mathbf{v} = 0 \quad \text{for all} \quad j < k. \tag{5}$$

The following result explicitly relates the Sonneveld subspaces to the IDR subspaces. For a proof and additional interesting results on Sonneveld subspaces, see [16].

Theorem 2.4. *Let $P_k(\lambda) \equiv \prod_{j=1}^{k}(\lambda - \mu_j)$. With $\mathcal{G}_k$ as in Def. 2.1, we have*

$$\mathcal{G}_k = \mathcal{S}(P_k, \mathbf{A}, \widetilde{\mathbf{R}}). \tag{6}$$

The first residual that arrives in an IDR subspace $\mathcal{G}_k$ is called a *primary* residual. This terminology is consistent with the literature on IDR(s) methods. In addition, we introduce here the term *secondary* residual, which is also always formed and which lives in the subspace $\mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$. For observations on the uniqueness of these residuals, see [17, 21]. We call any other residuals that may be constructed during a cycle *auxiliary* residuals.

Iterative algorithms based on the IDR theorem (i.e., Theorem 2.2) essentially consist of three key steps, which constitute the kth cycle of an IDR method (i.e., $s+1$ MVs). Let $k \in \mathbb{N}_0$. Given a full rank $n \times s$ matrix $\mathbf{V}_k$ with columns in $\mathcal{G}_k$ and a primary residual $\mathbf{r}_k \in \mathcal{G}_k$, we have:

(i) *The projection step*: the secondary residual $\mathbf{r}'_k$ is formed in $\mathcal{G}'_k \equiv \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$ using an oblique projection involving $\mathbf{V}_k$. This step consists of s MVs.

(ii) *The dimension reduction step*: given the secondary residual $\mathbf{r}'_k \in \mathcal{G}'_k$ and after selecting a scalar $\omega_{k+1} \equiv 1/\mu_{k+1}$, the next primary residual $\mathbf{r}_{k+1}$ in the lower dimensional subspace $\mathcal{G}_{k+1} \subset \mathcal{G}_k$ is computed as $\mathbf{r}_{k+1} = (\mathbf{I} - \omega_{k+1}\mathbf{A})\mathbf{r}'_k$. This step consists of 1 MV.

(iii) *The search matrix step*: a full rank $n \times s$ matrix $\mathbf{V}_{k+1}$ with columns in $\mathcal{G}_{k+1}$ of the form $\mathbf{A}\mathbf{U}_{k+1}$ with the so–called *search matrix* $\mathbf{U}_{k+1}$ explicitly available is constructed. This matrix is used for the next projection step.

To summarise, one full cycle of IDR(s) consists of $s+1$ MVs:

$$\begin{array}{c} \text{primary residual} \\ \mathbf{r}_k \in \mathcal{G}_k \end{array} \xrightarrow{s \text{ MVs}} \begin{array}{c} \text{secondary residual} \\ \mathbf{r}'_k \in \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp} \end{array} \xrightarrow{1 \text{ MV}} \begin{array}{c} \text{next primary residual} \\ \mathbf{r}_{k+1} \in \mathcal{G}_{k+1} \end{array} . \tag{7}$$

Following the discussion of [21, §4], three fundamental choices can be distinguished when deriving practical IDR(s) algorithms. These choices are directly related to the three steps given above. Not only do these choices influence the numerical stability and efficiency of the resulting IDR(s) algorithm, they can also drastically affect parallel scalability. These choices are:

(i) *Choosing the IDR test matrix.* In IDR(s) algorithms the computation of most of the inner products pertains to the columns of $\widetilde{\mathbf{R}}$. To reduce the amount of computational work (and communication) involving the IDR test matrix, sparse column vectors for $\widetilde{\mathbf{R}}$ may be used. A possible disadvantage is that such an approach can have an adverse effect on robustness. For a study on using sparse column vectors for $\widetilde{\mathbf{R}}$ and other methods of minimising communication in parallel IDR(s) algorithms, see [3].

(ii) *Selecting ω_{k+1} in the dimension reduction step.* In the dimension reduction step the value of ω_{k+1} can be chosen freely. Similar to other short–recurrence methods such as Bi–CGSTAB, smoother convergence may be achieved by choosing ω_{k+1} in such a way that the next residual is minimised in norm. For certain problems the (standard) linear minimal residual step causes break–down of the iteration process and sophisticated repair techniques such as used in Bi–CGSTAB(ℓ) are required [15, 19]. In Bi–CGSTAB(ℓ) stabilisation polynomials of degree ℓ are used and recently this technique has been combined with IDR(s) [18].

(iii) *Constructing vectors for the space $\mathcal{G}_{k+1}$ in the search matrix step.* Vectors for $\mathcal{G}_{k+1}$ can be generated by a number of ways. For example, using GCR–type methods [27]. Also, vectors for $\mathcal{G}_{k+1}$ can be made to satisfy (one–sided) bi–orthogonality relations with the columns of $\widetilde{\mathbf{R}}$ [28]. Furthermore, one can either derive variants that *only* compute primary and secondary residuals (e.g., [18]) or variants that also construct auxiliary residuals (e.g., [27, 28]). For more details on this type of freedom, see [17].

Note that different strategies for the last choice result in mathematically equivalent IDR(s) methods, while different strategies for the first and second choice result in fundamentally different iterative processes.

2.2 Deflation methods

In this section the structure of a particular deflation–type method called *adapted* deflation is presented. The discussion follows that of [23, §2]. We will first look at general deflation methods.

2.2.1 General deflation

We start with some terminology (cf. Def. 2.5) and a related result (cf. Lemma 2.6).

Definition 2.5 (cf. [5, 10, 17, 23])**.** Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be nonsingular and let $\mathbf{U}, \widetilde{\mathbf{R}} \in \mathbb{C}^{n\times s}$ be suitable *deflation subspace matrices* of full rank. Then the oblique projections (or *deflation matrices*) $\Pi, \widehat{\Pi} \in \mathbb{C}^{n\times n}$, the *correction matrix* (or *coarse grid correction*) $\mathbf{Q} \in \mathbb{C}^{n\times n}$, and the invertible (*Galerkin*) matrix $\mathbf{E} \in \mathbb{C}^{s\times s}$ are defined as

$$\Pi \equiv \mathbf{I} - \mathbf{AQ} \quad \text{and} \quad \widehat{\Pi} \equiv \mathbf{I} - \mathbf{QA}, \quad \text{where} \quad \mathbf{Q} \equiv \mathbf{U}\mathbf{E}^{-1}\widetilde{\mathbf{R}}^* \quad \text{and} \quad \mathbf{E} \equiv \widetilde{\mathbf{R}}^*\mathbf{AU}. \tag{8}$$

In the next section we will sometimes relate an operator or matrix with a specific IDR cycle k by adding a subscript k, e.g., $\mathbf{U}_k$ and $\mathbf{Q}_k$. Analogously, if a projection belongs to an IDR cycle k, we write e.g., Π_k.

Lemma 2.6 (cf. [5, 23])**.** *Note that the following properties hold for arbitrary full–rank matrices* $\mathbf{A}, \mathbf{U}$*, and* $\widetilde{\mathbf{R}}$*. In particular, we do not assume that the columns of* $\mathbf{U}$ *or* $\widetilde{\mathbf{R}}$ *are (approximations of) eigenvectors. Let* $j, k \in \mathbb{N}_0$*. Then*

(i) $\Pi^2 = \Pi, \widehat{\Pi}^2 = \widehat{\Pi}, (\mathbf{AQ})^2 = \mathbf{AQ}, (\mathbf{QA})^2 = \mathbf{QA}$

(ii) $\Pi\mathbf{A} = \mathbf{A}\widehat{\Pi} = \Pi\mathbf{A}\widehat{\Pi}$

(iii) $\widehat{\Pi}\mathbf{U} = \Pi\mathbf{AU} = \mathbf{0}_{n,s}$

(iv) $\widetilde{\mathbf{R}}^*\Pi = \widetilde{\mathbf{R}}^*\mathbf{A}\widehat{\Pi} = \mathbf{0}_{s,n}$

(v) $\mathbf{AQ} = \mathbf{I} - \Pi, \mathbf{QA} = \mathbf{I} - \widehat{\Pi}$

(vi) $\mathbf{QAU} = \mathbf{U}, \mathbf{QAQ} = \mathbf{Q}$

(vii) $\Pi\mathbf{Q} = \mathbf{Q}\Pi = \Pi\mathbf{AQ} = \mathbf{AQ}\Pi = \mathbf{QA}\widehat{\Pi} = \mathbf{Q}\Pi\mathbf{A} = \mathbf{0}_{n,n}$

(viii) $(\mathbf{I} - \widehat{\Pi})\mathbf{x} = \mathbf{Qb}$

(ix) $\widetilde{\mathbf{R}}^*\mathbf{AQ} = \widetilde{\mathbf{R}}^*$

Proof. We only show property (iii). The other properties can be derived similarly. For (iii), note that $\Pi\mathbf{AU} = \mathbf{AU} - \mathbf{AU}(\widetilde{\mathbf{R}}^*\mathbf{AU})^{-1}\widetilde{\mathbf{R}}^*\mathbf{AU} = \mathbf{AU} - \mathbf{AU} = \mathbf{0}_{n,s}$. □

Note 2.7. The following observations can be made.

- The operator Π is an oblique projection along $\mathrm{span}(\mathbf{AU})$ onto the orthogonal complement of $\widetilde{\mathbf{R}}$ (or onto $\mathrm{span}(\widetilde{\mathbf{R}})^\perp = \mathcal{N}(\widetilde{\mathbf{R}}^*)$). The operator $\mathbf{AQ}$ is an oblique projection along the orthogonal complement of $\widetilde{\mathbf{R}}$ onto $\mathrm{span}(\mathbf{AU})$.

- The operator $\widehat{\Pi}$ is an oblique projection along $\mathrm{span}(\mathbf{U})$ onto $\mathrm{span}(\mathbf{A}^*\widetilde{\mathbf{R}})^\perp = \mathcal{N}(\widetilde{\mathbf{R}}^*\mathbf{A})$. The operator $\mathbf{QA}$ is an oblique projection along $\mathrm{span}(\mathbf{A}^*\widetilde{\mathbf{R}})^\perp$ onto $\mathrm{span}(\mathbf{U})$.
- $\Pi\mathbf{v} \perp \widetilde{\mathbf{R}}$ for all $\mathbf{v}$, $\mathbf{A}\widehat{\Pi}\mathbf{v} \perp \widetilde{\mathbf{R}}$ for all $\mathbf{v}$, $\Pi\mathbf{v} \in \mathcal{N}(\widetilde{\mathbf{R}}^*)$ for all $\mathbf{v}$, $\mathbf{U} \in \mathcal{N}(\Pi\mathbf{A})$, $\mathbf{U} \in \mathcal{N}(\widehat{\Pi})$, $\Pi\mathbf{A} \in \mathcal{N}(\widetilde{\mathbf{R}}^*)$.
- Both projections Π and $\widehat{\Pi}$ have s zero and $n-s$ unit eigenvalues, since $\Pi\mathbf{AU} = 0_{n,s}$ and $\Pi^2\mathbf{Y} = \Pi\mathbf{Y}$ for full rank $\mathbf{Y} \in \mathbb{R}^{n\times(n-s)}$ satisfying $\mathrm{span}(\mathbf{Y}) = \mathrm{span}(\widetilde{\mathbf{R}})^\perp$. Vice versa, both the projections $\mathbf{AQ}$ and $\mathbf{QA}$ have $n-s$ zero eigenvalues and s unit eigenvalues.
- Note that $\dim\mathrm{span}(\Pi) = \dim\mathcal{N}(\mathbf{AQ}) = \dim\mathrm{span}(\widetilde{\mathbf{R}})^\perp = \dim\mathcal{N}(\widetilde{\mathbf{R}}^*) = n-s$.
- Note that $\dim\mathcal{N}(\Pi) = \dim\mathrm{span}(\mathbf{AQ}) = \dim\mathrm{span}(\mathbf{AU}) = s$.
- Note that $\dim\mathrm{span}(\Pi) + \dim\mathcal{N}(\Pi) = (n-s) + s = n = \mathrm{rank}\,\Pi$.

2.2.2 Adapted deflation methods

In an adapted deflation method [23, §2.3.3], two different preconditioners $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{C}^{n\times n}$ are combined as follows. Let $k \in \mathbb{N}_0$. Given $\mathbf{x}_0$, consider the two–step stationary iterative method (cf. predictor/corrector–type method)

$$\left\{\begin{array}{lcl} \mathbf{x}'_k &=& \mathbf{x}_k + \mathbf{C}_1(\mathbf{b} - \mathbf{A}\mathbf{x}_k); \\ \mathbf{x}_{k+1} &=& \mathbf{x}'_k + \mathbf{C}_2(\mathbf{b} - \mathbf{A}\mathbf{x}'_k), \end{array}\right. \tag{9}$$

which can be combined to obtain

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{P}(\mathbf{b} - \mathbf{A}\mathbf{x}_k), \tag{10}$$

where

$$\mathbf{P} = \mathbf{C}_1 + \mathbf{C}_2 - \mathbf{C}_2\mathbf{A}\mathbf{C}_1. \tag{11}$$

Using Def. 2.5, let $\mathbf{C}_1 \equiv \mathbf{Q}$ and $\mathbf{C}_2 \equiv \mathbf{B}^{-1}$ where $\mathbf{B}$ is an arbitrary (but fixed) matrix. Then $\mathbf{P} = \mathbf{B}^{-1}\Pi + \mathbf{Q}$ and the deflated linear system can be written as

$$\mathbf{PAx} = \mathbf{Pb}, \tag{12}$$

which can be solved using an iterative method. This particular deflation variant is called the *adapted deflation variant 1* (A–DEF1) and is described in [23, §2.3.3]. A common approach is to take $\mathbf{U}$ equal to $\widetilde{\mathbf{R}}$, consisting of s eigenvectors belonging to the smallest eigenvalues in norm. Additionally, $\mathbf{B}^{-1}$ is a traditional preconditioner such as an Incomplete LU factorisation. As a result, these s eigenvalues of $\mathbf{A}$ are then shifted to one in $\mathbf{PA}$, removing their influence from the iteration process (assuming that an appropriate scaling has taken place).

Note that premultiplying (9) with $-\mathbf{A}$ and adding $\mathbf{b}$ gives the corresponding recursions for the residuals

$$\left\{ \begin{array}{lcl} \mathbf{r}'_k & = & (\mathbf{I} - \mathbf{A}\mathbf{C}_1)\mathbf{r}_k; \\ \mathbf{r}_{k+1} & = & (\mathbf{I} - \mathbf{A}\mathbf{C}_2)\mathbf{r}'_k. \end{array} \right. \tag{13}$$

In the following, we mainly focus on the recursions for the residual. The corresponding recursions for the iterate can normally speaking be constructed easily.

2.3 Interpreting IDR(s) as a deflation method

In this section, the three steps of an IDR(s) method given in Sect. 2.1 will be discussed in the context of the adapted deflation variant given in Sect. 2.2. For reference purposes, Alg. 2.1 lists a complete IDR(s) algorithm for solving a system $\mathbf{A}\mathbf{x} = \mathbf{b}$ by generating primary and secondary residuals with corresponding iterates, illustrating the three distinct phases of IDR(s). Note that this variant is purely intended for illustrative purposes and that it should not be used to solve linear systems in practice. Also, this variant does not produce auxiliary residuals. In the discussion below we sometimes specifically refer to line numbers in Alg. 2.1. The projection step and the dimension reduction step are discussed together. After that, the search matrix step is discussed separately.

2.3.1 Projection step and dimension reduction step

Prop. 2.8 below shows that the IDR(s) method can be seen as a combination of two very specific operators $\mathbf{C}_1$ and $\mathbf{C}_2$. It explains how to move from a primary residual $\mathbf{r}_k$ to the next primary residual $\mathbf{r}_{k+1}$.

To form the secondary residual $\mathbf{r}'_k$ in $\mathcal{G}'_k \equiv \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$ from the primary residual in $\mathcal{G}_k$, we need a full rank $n \times s$ matrix with columns also in $\mathcal{G}_k$. In order to be able to update the approximate solution associated with $\mathbf{r}_k$ this $n \times s$ matrix has to be of the form $\mathbf{A}\mathbf{U}_k$ with $\mathbf{U}_k$ explicitly available: the update for the approximate solution is a linear combination of the columns of $\mathbf{U}_k$. This is also expressed in Prop. 2.8 (cf. (15) and line 9 of Alg. 2.1). We call an $n \times s$ matrix $\mathbf{U}_k$ a *search matrix* if it has full rank and $\mathrm{span}(\mathbf{A}\mathbf{U}_k) \subset \mathcal{G}_k$.

Proposition 2.8. *Consider a $k \in \mathbb{N}_0$. Assume $\mathbf{r}_k \in \mathcal{G}_k$ with $\mathbf{r}_k$ a residual, $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$, with approximate solution $\mathbf{x}_k$, where $\mathbf{x}_k$ is explicitly available. Also, assume* $\mathrm{span}(\mathbf{A}\mathbf{U}_k) \subset \mathcal{G}_k$, *with $\mathbf{U}_k$ explicitly available. Select a scalar $\mu_{k+1} \neq 0$ and put $\omega_{k+1} \equiv 1/\mu_{k+1}$. Let $\mathbf{B} \in \mathbb{C}^{n\times n}$ be a "traditional" preconditioning matrix. In the projections $\Pi, \widehat{\Pi}$ and in the operator $\mathbf{Q}_k$ of Def. 2.5, let $\widetilde{\mathbf{R}}$ be the IDR test matrix and let $\mathbf{U} = \mathbf{U}_k$. In* (13)*, let $\mathbf{C}_1 \equiv \mathbf{Q}_k$ (cf. a correction matrix) and $\mathbf{C}_2 \equiv \omega_{k+1}\mathbf{B}^{-1}$ (cf. modified preconditioned Richardson, smoothing step). In particular, assume that* $\mathrm{span}(\mathbf{A}\mathbf{Q}_k) \subset \mathcal{G}_k$. *The recursions for computing the primary and secondary residuals of a* single *IDR cycle are (cf.* (13)*)*

$$\left\{ \begin{array}{lcll} \mathbf{r}'_k & = & (\mathbf{I} - \mathbf{A}\mathbf{Q}_k)\mathbf{r}_k; & \textit{(see line 10 of Alg. 2.1)} \\ \mathbf{r}_{k+1} & = & (\mathbf{I} - \omega_{k+1}\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'_k, & \textit{(see line 30 of Alg. 2.1)} \end{array} \right. \tag{14}$$

Algorithm 2.1 IDR(s) as a deflation method: generating a sequence of IDR subspaces, $\text{span}(\mathbf{r}_k, \mathbf{V}_k) \subset \mathcal{G}_k$.

INPUT: $\mathbf{A} \in \mathbb{C}^{n\times n}; \mathbf{x}_0, \mathbf{b} \in \mathbb{C}^n; \widetilde{\mathbf{R}} \in \mathbb{C}^{n\times s}$; preconditioner $\mathbf{B} \in \mathbb{C}^{n\times n}$; tolerance tol
OUTPUT: Approximate solution $\mathbf{x}$ such that $||\mathbf{b} - \mathbf{A}\mathbf{x}|| \leq \text{tol}$

// Initiation
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
Select an $n \times s$ matrix $\mathbf{V}_0 \equiv \mathbf{A}\mathbf{U}_0$ with $\mathbf{U}_0$ explicitly available such that $[\mathbf{r}_0, \mathbf{V}_0]$ spans the Krylov subspace $\mathcal{K}_{s+1}(\mathbf{A}, \mathbf{r}_0)$
// Loop over nested $\mathcal{G}_k$ spaces
for $k = 1, 2, \ldots$ **do**
    *// **(i) Projection step:***
    // Generate unique secondary residual $\mathbf{r}'_{k-1} \in \mathcal{G}'_{k-1}$ with corresponding iterate $\mathbf{x}'_{k-1}$
    let $\Pi_{k-1} \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}_{k-1}$ and $\widehat{\Pi}_{k-1} \equiv \mathbf{I} - \mathbf{Q}_{k-1}\mathbf{A}$ where $\mathbf{Q}_{k-1} \equiv \mathbf{U}_{k-1}(\widetilde{\mathbf{R}}^*\mathbf{V}_{k-1})^{-1}\widetilde{\mathbf{R}}^*$
    $\mathbf{x}'_{k-1} = \mathbf{x}_{k-1} + \mathbf{Q}_{k-1}\mathbf{r}_{k-1}$
    $\mathbf{r}'_{k-1} = \Pi_{k-1}\mathbf{r}_{k-1}$
    if $||\mathbf{r}'_{k-1}|| \leq \text{tol}$ **then** break **end if**
    *// **(ii) Search matrix step:***
    // generate a basis $\mathbf{U}'_{k-1}$ of $\mathcal{K}_s(\widehat{\Pi}_{k-1}\mathbf{B}^{-1}\mathbf{A}, \widehat{\Pi}_{k-1}\mathbf{B}^{-1}\mathbf{r}'_{k-1})$
    // generate a basis $\mathbf{V}'_{k-1}$ of $\mathcal{K}_s(\Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}, \Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}\mathbf{r}'_{k-1})$
    // for example, a power basis:
    let $\mathbf{v}'_0 \equiv \mathbf{r}'_{k-1}$
    for $i = 1$ to s **do**
        $\mathbf{u}'_i = \widehat{\Pi}_{k-1}\mathbf{B}^{-1}\mathbf{v}'_{i-1}$
        $\mathbf{v}'_i = \mathbf{A}\mathbf{u}'_i = \Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}\mathbf{v}'_{i-1} \in \mathcal{G}'_{k-1} \equiv \mathcal{G}_{k-1} \cap \widetilde{\mathbf{R}}^{\perp}$
    end for
    let $\mathbf{U}'_{k-1} \equiv [\mathbf{u}'_1, \ldots, \mathbf{u}'_s]$ and $\mathbf{V}'_{k-1} \equiv [\mathbf{v}'_1, \ldots, \mathbf{v}'_s]$
    // Entering $\mathcal{G}_k$
    *// **(iii) Dimension reduction step:***
    Select a scalar ω_k, e.g., $\omega_k = \arg\min_\omega ||(\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'_{k-1}||$
    // Compute search matrix $\mathbf{U}_k$ for next IDR projection step with $\mathbf{V}_k \in \mathcal{G}_k$
    $\mathbf{U}_k = \mathbf{U}'_{k-1} - \omega_k\mathbf{B}^{-1}\mathbf{V}'_{k-1}$
    $\mathbf{V}_k = (\mathbf{I} - \omega_k\mathbf{A}\mathbf{B}^{-1})\mathbf{V}'_{k-1}$
    // Compute next primary residual $\mathbf{r}_k \in \mathcal{G}_k$ and corresponding iterate $\mathbf{x}_k$
    $\mathbf{x}_k = \mathbf{x}'_{k-1} + \omega_k\mathbf{B}^{-1}\mathbf{r}'_{k-1}$
    $\mathbf{r}_k = (\mathbf{I} - \omega_k\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'_{k-1}$
end for

where the secondary residual $\mathbf{r}'_k = \Pi_k\mathbf{r}_k$ in $\mathcal{G}'_k \equiv \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$ (i.e., step (i), the projection step, *cf.* (8)*) and the next primary residual $\mathbf{r}_{k+1}$ in $\mathcal{G}_{k+1}$ (i.e., step (ii), the* dimension reduction

step*). The corresponding recursions for the iterates are*

$$\begin{cases} \mathbf{x}'_k &= \mathbf{x}_k + \mathbf{Q}_k\mathbf{r}_k; & \textit{(see line 9 of Alg. 2.1)} \\ \mathbf{x}_{k+1} &= \mathbf{x}'_k + \omega_{k+1}\mathbf{B}^{-1}\mathbf{r}'_k. & \textit{(see line 29 of Alg. 2.1)} \end{cases} \tag{15}$$

Note that the update for $\mathbf{x}'_k$ is a linear combination of the columns of $\mathbf{U}_k$. The resulting operator $\mathbf{P}_k$ is (cf. (11)*)*

$$\mathbf{P}_k = \mathbf{Q}_k + \omega_{k+1}\mathbf{B}^{-1} - \omega_{k+1}\mathbf{B}^{-1}\mathbf{A}\mathbf{Q}_k = \omega_{k+1}\mathbf{B}^{-1}\Pi_k + \mathbf{Q}_k. \tag{16}$$

The complete recursions for the residual and iterate are

$$\begin{cases} \mathbf{r}_{k+1} &= \mathbf{r}_k - \mathbf{A}\mathbf{P}_k\mathbf{r}_k; \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{P}_k(\mathbf{b} - \mathbf{A}\mathbf{x}_k). \end{cases} \tag{17}$$

Proof. Use Def. 2.1 and Def. 2.5. □

This proposition also shows that the IDR(s) method can be seen as a Richardson iteration with a varying deflation–type preconditioner. In particular, the operator $\mathbf{P}_k$ is analogous to the A–DEF1 deflation method described above and in [23, §2.3.3]. Also, note that $\mathbf{P}_k^2 \neq \mathbf{P}_k$ and that $\mathbf{P}_k\mathbf{A}$ is non–symmetric even if $\mathbf{A}$ is symmetric. The matrix $\Pi_k\mathbf{A}$ is singular, but the matrix $\mathbf{P}_k\mathbf{A}$ is not. It is shown in [21] that the primary residuals $\mathbf{r}_k$ (and therefore also $\mathbf{r}'_k$) are unique. If there is no risk of ambiguity (that is, if it is clear within which IDR subspace k we are operating or if it is irrelevant), we will sometimes drop the index k of for example $\mathbf{P}_k$ and Π_k.

Note 2.9. The error–propagation operator (or iteration matrix) belonging to IDR(s) is

$$\mathbf{I} - \mathbf{P}_k\mathbf{A} = \mathbf{I} - \omega_{k+1}\mathbf{B}^{-1}\Pi_k\mathbf{A} - \mathbf{Q}_k\mathbf{A} \tag{18}$$

$$= \widehat{\Pi}_k - \omega_{k+1}\mathbf{B}^{-1}\mathbf{A}\widehat{\Pi}_k \tag{19}$$

$$= (\mathbf{I} - \omega_{k+1}\mathbf{B}^{-1}\mathbf{A})\widehat{\Pi}_k, \tag{20}$$

where $(\mathbf{I} - \omega_{k+1}\mathbf{B}^{-1}\mathbf{A})$ can be seen as a (post–)smoother and $\widehat{\Pi}_k$ a coarse–grid correction operation [23]. Note that the iteration matrix changes in each cycle k.

Note 2.10. In IDR(s), a single Richardson step is applied in each cycle k to the (deflated and nonsingular) system (cf. (17))

$$\mathbf{P}_k\mathbf{A}\mathbf{x}_k = \mathbf{P}_k\mathbf{b}. \tag{21}$$

This step consists of $s + 1$ (preconditioned) MVs.

The role of the preconditioner $\mathbf{B}^{-1}$ in Alg. 2.1 may not be entirely apparent so we will discuss this in some detail. In Fig. 1 the three steps of introducing a (right) preconditioner $\mathbf{B}^{-1}$ into the IDR(s) method from Alg. 2.1 are shown. We only show a single IDR(s) cycle and we drop the index k, starting with a primary residual $\mathbf{r}$, generating the secondary residual $\mathbf{r}'$ and finally the next primary residual $\mathbf{r}''$.

$\mathbf{y}' = \mathbf{y} + \mathbf{Q}\mathbf{r}$
$\mathbf{r}' = \Pi\mathbf{r}$
for $i = 1$ to s **do**
  $\mathbf{u}' = \widehat{\Pi}\mathbf{v}'$
  $\mathbf{v}' = \mathbf{A}\mathbf{B}^{-1}\mathbf{u}'$
end for
$\mathbf{y}'' = \mathbf{y}' + \omega\mathbf{r}'$
$\mathbf{r}'' = (\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'$

$\mathbf{B}^{-1}\mathbf{y}' = \mathbf{B}^{-1}\mathbf{y} + \mathbf{B}^{-1}\mathbf{Q}\mathbf{r}$
$\mathbf{r}' = \Pi\mathbf{r}$
for $i = 1$ to s **do**
  $\mathbf{B}^{-1}\mathbf{u}' = \mathbf{B}^{-1}\widehat{\Pi}\mathbf{v}'$
  $\mathbf{v}' = \mathbf{A}\mathbf{B}^{-1}\mathbf{u}'$
end for
$\mathbf{B}^{-1}\mathbf{y}'' = \mathbf{B}^{-1}\mathbf{y}' + \omega\mathbf{B}^{-1}\mathbf{r}'$
$\mathbf{r}'' = (\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'$

$\mathbf{x}' = \mathbf{x} + \underline{\mathbf{Q}}\mathbf{r}$
$\mathbf{r}' = \Pi\mathbf{r}$
for $i = 1$ to s **do**
  $\underline{\mathbf{u}}' = \underline{\widehat{\Pi}}\mathbf{B}^{-1}\mathbf{v}'$
  $\mathbf{v}' = \mathbf{A}\underline{\mathbf{u}}'$
end for
$\mathbf{x}'' = \mathbf{x}' + \omega\mathbf{B}^{-1}\mathbf{r}'$
$\mathbf{r}'' = (\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'$

Figure 1: Left: solve $\mathbf{A}\mathbf{B}^{-1}\mathbf{y} = \mathbf{x}$. Middle: premultiply $\mathbf{y}$ and $\mathbf{u}$ by $\mathbf{B}^{-1}$. Right: substitute $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$.

In the left part of Fig. 1, the IDR(s) method from Alg. 2.1 is applied to the preconditioned system $\mathbf{A}\mathbf{B}^{-1}\mathbf{y} = \mathbf{b}$ with $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$ as the final solution. In the middle part the vectors $\mathbf{y}$ and $\mathbf{u}$ are premultiplied by $\mathbf{B}^{-1}$ and again we have the final solution $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$. Finally, in the right part we set $\underline{\mathbf{u}}' = \mathbf{B}^{-1}\mathbf{u}'$ and the transformed iterates $\mathbf{y}$ are scaled back to the iterates $\mathbf{x}$ of the original system $\mathbf{A}\mathbf{x} = \mathbf{b}$. The notation $\underline{\widehat{\Pi}}$ underlines the fact that this projection uses the preconditioned search matrix $\underline{\mathbf{U}} \equiv \mathbf{B}^{-1}\mathbf{U}$. The same holds for the operator $\underline{\mathbf{Q}}$. In the listing of Alg. 2.1, this distinction in notation is not explicitly made.

Note that in the right part of Fig. 1, the projection $\underline{\widehat{\Pi}}$ is operating within the original solution space ($\mathbf{x}$), while Π operates within the transformed solution space ($\mathbf{y}$).

For ease of notation, we will often take $\mathbf{B} = \mathbf{I}$ in the remainder of this paper.

2.3.2 Search matrices

In the search matrix step (i.e., step (iii)), we need to generate a full rank $n \times s$ matrix with columns in $\mathcal{G}_{k+1}$ of the form $\mathbf{A}\mathbf{U}_{k+1}$ with $\mathbf{U}_{k+1}$ explicitly available. We have to construct this matrix from vectors in $\mathcal{G}_k$. In this IDR subspace, we only have the subspace $\text{span}(\mathbf{r}_k, \mathbf{A}\mathbf{U}_k)$ available with $\mathbf{U}_k$ an $n \times s$ matrix. The vector $\mathbf{y}$ for which $\mathbf{A}\mathbf{y} = \mathbf{r}_k$ will not be available (because that would require to solve a linear system with $\mathbf{A}$). Therefore, the vector $\mathbf{r}_k$ is not helpful. However, in the computation of $\mathbf{r}_{k+1} \in \mathcal{G}_{k+1}$, we also computed $\mathbf{A}\mathbf{r}'_k$, which also belongs to $\mathcal{G}_k$ (use Theorem 2.2, now with $\mu_k = 0$). Hence, we can use vectors from $\text{span}(\mathbf{A}\mathbf{r}'_k, \mathbf{A}\mathbf{U}_k)$.

Using the projection $\Pi_k, \widehat{\Pi}_k$ from Prop. 2.8 we have in particular: $\Pi_k(\mathbf{A}\mathbf{v})$ is in $\mathcal{G}'_k$ if $\mathbf{A}\mathbf{v} \in \mathcal{G}_k$ and $\Pi_k(\mathbf{A}\mathbf{v})$ is of the form $\mathbf{A}\mathbf{u}$ with $\mathbf{u}$ explicitly available: $\mathbf{u} = \widehat{\Pi}_k\mathbf{v}$. Moreover, $\Pi_k(\mathbf{A}\mathbf{v}) = \mathbf{A}\mathbf{v} - \mathbf{A}\mathbf{u}\beta$ with $\beta \equiv \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{v}$ and for the same β we have that $\widehat{\Pi}_k\mathbf{v} = \mathbf{v} - \mathbf{U}\beta$: the projection $\widehat{\Pi}_k$ requires vector updates, but no additional inner products.

With $\mathbf{s} \in \mathbb{C}^n$ and $\widetilde{\mathbf{U}}$ a full rank $n \times s$ matrix such that $[\mathbf{s}, \widetilde{\mathbf{U}}]$ spans an $s+1$ dimensional subspace of $\text{span}(\mathbf{r}'_k, \mathbf{U}_k)$, we can use $\widetilde{\mathbf{U}}$ and $\mathbf{A}\widetilde{\mathbf{U}}$ in the projections Π_k and $\widehat{\Pi}_k$ of (8) and take $\mathbf{s}$ to start the construction of $\mathbf{U}_{k+1}$. However, for ease of notation, we formulate Prop. 2.12 for $\widetilde{\mathbf{U}} = \mathbf{U}_k$ and we suggest to use $\mathbf{s} = \mathbf{r}'_k$.

Lemma 2.11. *If* $\mathbf{s} \in \mathcal{G}'_k$, *then* $\Pi_k \mathbf{A}\mathbf{s} = \mathbf{A}\widehat{\Pi}_k \mathbf{s} \in \mathcal{G}'_k$.

Proof. From Theorem 2.4 we learn that $\mathbf{A}\mathbf{s} \in \mathbf{A}(\mathcal{G}'_k) \subset \mathcal{G}_k$, whence $\Pi_k \mathbf{A}\mathbf{s} \in \mathcal{G}'_k$. $\Pi_k \mathbf{A}\mathbf{s} = \mathbf{A}\widehat{\Pi}_k \mathbf{s}$ follows from Lemma 2.6:(ii). □

Proposition 2.12. $\mathbf{A}\widehat{\Pi}_k \mathbf{r}'_k \in \mathcal{G}'_k$. *Let* $\mathbf{u}'_1 \in \mathbb{C}^n$ *(for instance,* $\mathbf{u}'_1 = \widehat{\Pi}_k \mathbf{r}'_k$, *see line 16 of Alg. 2.1).*
If $\mathbf{A}\mathbf{u}'_1 \in \mathcal{G}'_k$, *and* $\mathbf{U}'_k$ *is a matrix with columns in* $\mathcal{K}_s(\widehat{\Pi}_k \mathbf{A}, \mathbf{u}'_1)$, *then*

$$\operatorname{span}(\mathbf{A}\mathbf{U}'_k) \subset \mathcal{K}_s(\Pi_k \mathbf{A}, \mathbf{A}\mathbf{u}'_1) \subset \mathcal{G}'_k \quad \textit{and} \quad \operatorname{span}(\mathbf{A}(\mu_{k+1}\mathbf{I} - \mathbf{A})\mathbf{U}'_k) \subset \mathcal{G}_{k+1}.$$

Proof. The first claim follows by combining Prop. 2.8 and Lemma 2.11. The inclusion $\mathcal{K}_s(\Pi_k \mathbf{A}, \mathbf{A}\mathbf{u}'_1) \subset \mathcal{G}'_k$ follows from Lemma 2.11 and an induction argument. Further, $\mathbf{A}\mathcal{K}_s(\widehat{\Pi}_k \mathbf{A}, \mathbf{u}'_1) = \mathcal{K}_s(\mathbf{A}\widehat{\Pi}_k, \mathbf{A}\mathbf{u}'_1) = \mathcal{K}_s(\Pi_k \mathbf{A}, \mathbf{A}\mathbf{u}'_1)$ (use Lemma 2.6:(ii)). □

The proposition tells us that any set of s linearly independent vectors in $\mathcal{K}_s(\widehat{\Pi}_k \mathbf{A}, \mathbf{u}'_1)$ forms a matrix $\mathbf{U}'_k$ for which the columns of $\mathbf{A}(\mathbf{I} - \omega\mathbf{A})\mathbf{U}'_k$ are in $\mathcal{G}_{k+1}$: $\mathbf{U}_{k+1} \equiv (\mathbf{I} - \omega\mathbf{A})\mathbf{U}'_k$ forms an appropriate search matrix for the next IDR step (see line 26 of Alg. 2.1).

In Alg. 2.1, a power basis for the Krylov subspaces is generated. For larger values of s (as $s > 4$) a more stable basis might be required. The choice of such a basis and its efficient computation is considered in detail in [17].

2.4 IDR algorithms

In this section we will make some general remarks about IDR(s) algorithms. Algorithm 2.1 describes a way to recursively generate $s+1$ dimensional subspaces of $\mathcal{G}_k$:

Proposition 2.13. *If* $[\mathbf{r}_j, \mathbf{V}_j]$ *has rank* $s+1$ *for* $j \leq k$, *then* $\operatorname{span}(\mathbf{r}_k, \mathbf{V}_k) \subset \mathcal{G}_k$.

Proof. Clearly, $\operatorname{span}(\mathbf{r}_0, \mathbf{V}_0) \subset \mathcal{G}_0$. Assume that $\operatorname{span}(\mathbf{r}_{k-1}, \mathbf{V}_{k-1}) \subset \mathcal{G}_{k-1}$. Then $\Pi_{k-1}\mathbf{r}_{k-1} \in \mathcal{G}'_{k-1}$ and, by Lemma 2.11, $\mathcal{K}_{s+1}(\Pi_{k-1}\mathbf{A}, \Pi_{k-1}\mathbf{r}_{k-1}) \subset \mathcal{G}'_{k-1}$. Hence, $\operatorname{span}(\mathbf{r}_k, \mathbf{V}_k) \subset (\mathbf{A} - \mu_k \mathbf{I})\mathcal{G}'_{k-1} = \mathcal{G}_k$. □

The listing of Alg. 2.1 is somewhat different from more "common" listings of IDR(s) algorithms. To give a concrete example, Alg. 2.2 shows such a traditional listing of an IDR(s) algorithm using the deflation language of this paper. For comparison purposes, the preconditioner $\mathbf{B}$ is explicitly added. Like Alg. 2.1, this listing is one of the most basic formulations of an IDR(s) algorithm and it is not intended for practical applications. It was chosen to facilitate comparing with Alg. 2.1. For efficient and stable variants of IDR(s) algorithms, see [3, 28]. We will now discuss the differences and similarities between Alg. 2.1 and Alg. 2.2.

In Alg. 2.2, the vectors $\mathbf{g}_i$ are directly lifted to the (primary) IDR subspace $\mathcal{G}_k$, while Alg. 2.1 explicitly generates s (secondary) vectors $\mathbf{v}'_i$ in the (secondary) IDR subspace $\mathcal{G}_{k-1} \cap \widetilde{\mathbf{R}}^{\perp}$. The main goal of traditional listings such as Alg. 2.2 is to generate linearly independent vectors (often denoted by $\mathbf{g}$ as in Alg. 2.2) in the IDR subspace $\mathcal{G}_k$ of unknown (possibly large) dimension. These vectors *do not form a basis* of the subspace $\mathcal{G}_k$. In contrast,

Algorithm 2.2 IDR(s) as a deflation method: a "traditional" listing.

INPUT: $\mathbf{A} \in \mathbb{C}^{n\times n}; \mathbf{x}_0, \mathbf{b} \in \mathbb{C}^n; \widetilde{\mathbf{R}} \in \mathbb{C}^{n\times s}$; preconditioner $\mathbf{B} \in \mathbb{C}^{n\times n}$; tolerance tol
OUTPUT: Approximate solution $\mathbf{x}$ such that $||\mathbf{b} - \mathbf{A}\mathbf{x}|| \leq$ tol

1: *// Initiation*
2: $\Pi_{-1} = \mathbf{I}; \mathbf{Q}_{-1} = \mathbf{0}; \omega_0 = 1$
3: Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
4: *// Loop over nested $\mathcal{G}_k$ spaces*
5: **for** $k = 0, 1, \ldots$ **do**
6: *// Compute s independent vectors $\mathbf{g}$ in $\mathcal{G}_k$ using vectors from $\mathcal{G}_{k-1}$*
7: **let** $\Pi_{k-1} = \mathbf{I} - \mathbf{A}\mathbf{Q}_{k-1}$ where $\mathbf{Q}_{k-1} = \mathbf{U}_{k-1}(\widetilde{\mathbf{R}}^*\mathbf{G}_{k-1})^{-1}\widetilde{\mathbf{R}}^*$
8: **let** $\mathbf{P}_{k-1} = \omega_k\mathbf{B}^{-1}\Pi_{k-1} + \mathbf{Q}_{k-1}$
9: **let** $\mathbf{g}_0 = \mathbf{r}_k$ *// (cf. line 16 of Alg. 2.1)*
10: **for** $i = 1$ to s **do**
11: $\mathbf{u}_i = \mathbf{P}_{k-1}\mathbf{g}_{i-1}$ *// (cf. eq. (16))*
12: $\mathbf{g}_i = \mathbf{A}\mathbf{u}_i \in \mathcal{G}_k$ *// (cf. line 19 of Alg. 2.1)*
13: **end for**
14: **let** $\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_s]$ and $\mathbf{G}_k = [\mathbf{g}_1, \ldots, \mathbf{g}_s]$
15: **let** $\Pi_k = \mathbf{I} - \mathbf{A}\mathbf{Q}_k$ where $\mathbf{Q}_k = \mathbf{U}_k(\widetilde{\mathbf{R}}^*\mathbf{G}_k)^{-1}\widetilde{\mathbf{R}}^*$
16: $\mathbf{x}'_k = \mathbf{x}_k + \mathbf{Q}_k\mathbf{r}_k$ *// (cf. line 9 of Alg. 2.1)*
17: $\mathbf{r}'_k = \Pi_k\mathbf{r}_k$ *// (cf. line 10 of Alg. 2.1)*
18: *// Entering $\mathcal{G}_{k+1}$*
19: $\omega_{k+1} = \arg\min_\omega ||(\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'_k||$ *// (cf. line 24 of Alg. 2.1)*
20: *// Compute next primary residual $\mathbf{r}_{k+1} \in \mathcal{G}_{k+1}$ and corresponding iterate $\mathbf{x}_{k+1}$*
21: $\mathbf{x}_{k+1} = \mathbf{x}'_k + \omega_{k+1}\mathbf{B}^{-1}\mathbf{r}'_k$ *// (cf. line 29 of Alg. 2.1)*
22: $\mathbf{r}_{k+1} = (\mathbf{I} - \omega_{k+1}\mathbf{A}\mathbf{B}^{-1})\mathbf{r}'_k$ *// (cf. line 30 of Alg. 2.1)*
23: **if** $||\mathbf{r}_{k+1}|| \leq$ tol **then** break **end if** *// (cf. line 11 of Alg. 2.1)*
24: **end for**

the vectors $\mathbf{v}'$ are explicitly generated to form a basis of $\mathcal{K}_s(\Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}, \Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}\mathbf{r}'_{k-1})$. To underline these distinctions, we use different letters for these sets of vectors.

However, it can be shown that in both listings we are in fact generating vectors $\mathbf{u}_i$ for a basis of the same Krylov subspace. In particular, the columns of $\mathbf{U}_a \equiv \mathbf{U}_k$ in line 26 of Alg. 2.1 form a basis of the Krylov subspace $(\mathbf{I}-\omega_k\mathbf{B}^{-1}\mathbf{A})\mathcal{K}_s(\widehat{\Pi}_{k-1}\mathbf{B}^{-1}\mathbf{A}, \widehat{\Pi}_{k-1}\mathbf{B}^{-1}\Pi_{k-1}\mathbf{r}_{k-1})$, while the columns of $\mathbf{U}_b \equiv \mathbf{U}_k$ in line 14 of Alg. 2.2 form a basis of the Krylov subspace $\mathcal{K}_s(\mathbf{P}_{k-1}\mathbf{A}, \mathbf{P}_{k-1}\mathbf{r}_k)$. The following proposition shows that these subspaces are identical.

Proposition 2.14. *Let $\mathbf{P}, \Pi,$ and $\widehat{\Pi}$ be as in Prop 2.8 and let $\mathbf{B} = \mathbf{I}$. Then we have*

$$\mathcal{K}_s(\mathbf{P}_{k-1}\mathbf{A}, \mathbf{P}_{k-1}\mathbf{r}_k) = (\mathbf{I} - \omega_k\mathbf{A})\mathcal{K}_s(\widehat{\Pi}_{k-1}\mathbf{A}, \widehat{\Pi}_{k-1}\Pi_{k-1}\mathbf{r}_{k-1}). \tag{22}$$

Proof. Using Lemma 2.6 and Prop. 2.8, we have that

$$\begin{aligned}
\mathbf{P}_{k-1}\mathbf{r}_k &= (\omega_k\Pi_{k-1} + \mathbf{Q}_{k-1})(\mathbf{I} - \omega_k\mathbf{A})\Pi_{k-1}\mathbf{r}_{k-1} \\
&= \omega_k\Pi_{k-1}^2\mathbf{r}_{k-1} - \omega_k^2\Pi_{k-1}\mathbf{A}\Pi_{k-1}\mathbf{r}_{k-1} + \mathbf{Q}_{k-1}\Pi_{k-1}\mathbf{r}_{k-1} - \omega_k\mathbf{Q}_{k-1}\mathbf{A}\Pi_{k-1}\mathbf{r}_{k-1} \\
&= \omega_k(\mathbf{I} - \omega_k\Pi_{k-1}\mathbf{A} - \mathbf{Q}_{k-1}\mathbf{A})\Pi_{k-1}\mathbf{r}_{k-1} \\
&= \omega_k(\widehat{\Pi}_{k-1} - \omega_k\mathbf{A}\widehat{\Pi}_{k-1})\Pi_{k-1}\mathbf{r}_{k-1} \\
&= \omega_k(\mathbf{I} - \omega_k\mathbf{A})\widehat{\Pi}_{k-1}\Pi_{k-1}\mathbf{r}_{k-1}.
\end{aligned}$$

Using these arguments, it can be shown that for $\mathbf{U}_b$ of Alg. 2.2 we have

$$\begin{aligned}
\mathrm{span}(\mathbf{U}_b) &= \mathcal{K}_s(\mathbf{P}_{k-1}\mathbf{A}, \mathbf{P}_{k-1}\mathbf{r}_k) \\
&= \mathcal{K}_s((\omega_k\Pi_{k-1} + \mathbf{Q}_{k-1})\mathbf{A}, \omega_k(\mathbf{I} - \omega_k\mathbf{A})\widehat{\Pi}_{k-1}\Pi_{k-1}\mathbf{r}_{k-1}) \\
&= (\mathbf{I} - \omega_k\mathbf{A})\mathcal{K}_s(\widehat{\Pi}_{k-1}\mathbf{A}, \widehat{\Pi}_{k-1}\Pi_{k-1}\mathbf{r}_{k-1}) \\
&= \mathrm{span}(\mathbf{U}_a),
\end{aligned}$$

for $\mathbf{U}_a$ of Alg. 2.1. Therefore, both sets of vectors span the same Krylov subspace, which proves the proposition. □

Note that the secondary residual $\mathbf{r}'_k$ is explicitly generated in both listings, since it is needed to perform the dimension reduction step.

In practical applications, when the IDR(s) iterative process begins to exhibit its superlinear convergence property, the largest drops in residual norms seem to occur for the secondary residuals $\mathbf{r}'_k$, so we therefore propose to check convergence using these residuals (see line 11 of Alg. 2.1).

For both completeness and illustrative purposes, we reproduce in Alg. 2.3 the IDR(s)-biortho method from [28] using the deflation language of this paper. Note that this variant computes auxiliary residuals.

3 The IDR projection theorem

In the previous section it has been shown that the IDR(s) method can be seen as a Richardson iteration preconditioned by a variable deflation–type preconditioner. Therefore, it makes sense to investigate the spectra of the sequence of deflated systems, i.e.,

$$\sigma(\mathbf{P}_k\mathbf{A}), \qquad k = 0, 1, \ldots. \tag{23}$$

The system $\mathbf{P}_k\mathbf{A}$ is called the kth *active system* of the iteration process.

In Sect. 3.1 the spectrum of the active system is investigated for a *single* IDR(s) cycle. Section 3.2 contains the main result of this paper, which relates the spectra of the active systems of *multiple* IDR(s) cycles. We will show that in IDR(s) the spectrum of $\mathbf{P}_k\mathbf{A}$ become increasingly more *clustered* with increasing k. Also, the spectrum of $\mathbf{P}_k\mathbf{A}$ is related to the active spectra of all the previous cycles $0, \ldots, k-1$. Section. 3.3 contains numerical examples to illustrate the IDR projection theorem. In Sect. 3.4 some possible interpretations of the IDR projection theorem are discussed.

Algorithm 2.3 IDR(s) as a deflation method: the IDR(s)-biortho variant from [28].

INPUT: $\mathbf{A} \in \mathbb{C}^{n\times n}; \mathbf{x}, \mathbf{b} \in \mathbb{C}^n; \widetilde{\mathbf{R}} \in \mathbb{C}^{n\times s}$; preconditioner $\mathbf{B} \in \mathbb{C}^{n\times n}$; tolerance tol
OUTPUT: Approximate solution $\mathbf{x}$ such that $||\mathbf{b} - \mathbf{A}\mathbf{x}|| \leq \varepsilon$

$\mathbf{Q}_{-1} = \mathbf{0} \in \mathbb{C}^{n\times n}; \omega_0 = 1$
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
for $k = 0, 1, \ldots$ **do**
    // Compute s independent vectors $\mathbf{g}$ in $\mathcal{G}_k$ using vectors from $\mathcal{G}_{k-1}$
    for $i = 1$ to s **do**
        let $\Pi_{k-1} \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}_{k-1}$ where $\mathbf{Q}_{k-1} \equiv \mathbf{U}_{k-1}(\widetilde{\mathbf{R}}^*\mathbf{G}_{k-1})^{-1}\widetilde{\mathbf{R}}^*$ and $\widetilde{\mathbf{R}} \equiv [\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_s]$
        let $\mathbf{P}_{k-1} \equiv \omega_k \mathbf{B}^{-1}\Pi_{k-1} + \mathbf{Q}_{k-1}$
        $\widehat{\mathbf{u}}_i = \mathbf{P}_{k-1}\mathbf{r}$
        $\widehat{\mathbf{g}}_i = \mathbf{A}\widehat{\mathbf{u}}_i$ $\in \mathcal{G}_k$
        let $\mathbf{G} \equiv [\mathbf{g}_1, \ldots, \mathbf{g}_{i-1}]$, $\mathbf{U} \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_{i-1}]$, and $\widetilde{\mathbf{R}} \equiv [\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_{i-1}]$ for $i > 1$
        let $\Pi \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}$ and $\widehat{\Pi} \equiv \mathbf{I} - \mathbf{Q}\mathbf{A}$ where $\mathbf{Q} \equiv \mathbf{U}(\widetilde{\mathbf{R}}^*\mathbf{G})^{-1}\widetilde{\mathbf{R}}^*$
        $\mathbf{u}_i = \widehat{\Pi}\widehat{\mathbf{u}}_i$ $\perp_{\mathbf{A}} \widetilde{\mathbf{R}}$
        $\mathbf{g}_i = \Pi\widehat{\mathbf{g}}_i$ $\in \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$
        let $\mathbf{G} \equiv [\mathbf{g}_1, \ldots, \mathbf{g}_i]$, $\mathbf{U} \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_i]$, and $\widetilde{\mathbf{R}} \equiv [\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_i]$
        let $\Pi \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}$ and $\widehat{\Pi} \equiv \mathbf{I} - \mathbf{Q}\mathbf{A}$ where $\mathbf{Q} \equiv \mathbf{U}(\widetilde{\mathbf{R}}^*\mathbf{G})^{-1}\widetilde{\mathbf{R}}^*$
        $\mathbf{x} = \mathbf{x} + \mathbf{Q}\mathbf{r}$
        $\mathbf{r} = \Pi\mathbf{r}$ $\in \mathcal{G}_k \cap \widetilde{\mathbf{R}}^{\perp}$
    end for
    // Search matrix $\mathbf{U}_k$ for next IDR projection step and $\mathbf{G}_k$ with columns in $\mathcal{G}_k$:
    let $\mathbf{G}_k \equiv \mathbf{G}$ and $\mathbf{U}_k \equiv \mathbf{U}$
    // Entering $\mathcal{G}_{k+1}$
    $\widetilde{\mathbf{v}} = \mathbf{B}^{-1}\mathbf{r}$
    $\mathbf{t} = \mathbf{A}\widetilde{\mathbf{v}}$
    $\omega_{k+1} = \arg\min_\omega ||(\mathbf{I} - \omega\mathbf{A}\mathbf{B}^{-1})\mathbf{r}||$
    // Compute next primary residual $\mathbf{r} \in \mathcal{G}_{k+1}$ and corresponding iterate $\mathbf{x}$
    $\mathbf{x} = \mathbf{x} + \omega_{k+1}\widetilde{\mathbf{v}}$
    $\mathbf{r} = \mathbf{r} - \omega_{k+1}\mathbf{t}$
    if $||\mathbf{r}|| \leq$ tol **then** break **end if**
end for

3.1 Single IDR(s) cycle

In the following, we partly follow [5] and [23]. We will first show some properties of the active system of a *single* IDR(s) cycle k. Therefore, the index k is dropped in this section.

Lemma 3.1 (cf. Lemma 2.5 [5])**.** *Let $\mathbf{A}$ and $\mathbf{B}$ be nonsingular. For all full ranked rectangular matrices $\mathbf{U}$ and $\widetilde{\mathbf{R}}$, $\mathbf{B}^{-1}\Pi\mathbf{A}$ and $\mathbf{P}\mathbf{A}$ have s eigenvalues equal to 0 and 1, respectively.*

Proof. From Lemma 2.6 it follows that $\mathbf{B}^{-1}\Pi\mathbf{A}\mathbf{U} = \mathbf{0}_{n,s}$ and $(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}\mathbf{U} = \mathbf{U}$. Additionally, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_s]$ are the eigenvectors of $\mathbf{B}^{-1}\Pi\mathbf{A}$ and $(\mathbf{B}^{-1}\Pi+\mathbf{Q})\mathbf{A}$ associated with the eigenvalues 0 and 1, respectively. □

Theorem 3.2 (cf. [23], cf. Theorem 2.8 [5])**.** *For all cycles we have*

$$\sigma(\mathbf{B}^{-1}\Pi\mathbf{A}) = \{0\} \cup \{\lambda'_{s+1}, \ldots, \lambda'_n\} \tag{24}$$

$$\Leftrightarrow$$

$$\sigma((\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}) = \{1\} \cup \{\lambda'_{s+1}, \ldots, \lambda'_n\}, \tag{25}$$

where the eigenvalues 0 *and* 1 *both have multiplicity* s*.*

Proof. For the implication $\Leftarrow$, see [5, Theorem 2.8]. For the implication $\Rightarrow$, note that for $\mathbf{u}_i$ with $i = 1, \ldots, s$ where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_s]$ we have $(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}\mathbf{U} = \mathbf{U}$ and $\mathbf{B}^{-1}\Pi\mathbf{A}\mathbf{U} = \mathbf{0}_{n,s}$, so the eigenvectors $\mathbf{U}$ of $(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}$ corresponding to the unit eigenvalues are the same as those corresponding to the zero eigenvalues of $\mathbf{B}^{-1}\Pi\mathbf{A}$.

For $i = s+1, \ldots, n$, suppose that the eigenvectors $\{\mathbf{v}_i\}$ satisfy $\mathbf{B}^{-1}\Pi\mathbf{A}\widehat{\Pi}\mathbf{v}_i = \lambda_i\widehat{\Pi}\mathbf{v}_i$ with corresponding eigenvalues $\{\lambda_i\}$. Then (using (vii) from Lem. 2.6)

$$(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}\widehat{\Pi}\mathbf{v}_i = \mathbf{B}^{-1}\Pi\mathbf{A}\widehat{\Pi}\mathbf{v}_i + \mathbf{Q}\mathbf{A}\widehat{\Pi}\mathbf{v}_i \tag{26}$$

$$= \lambda_i\widehat{\Pi}\mathbf{v}_i. \tag{27}$$

So the eigenvalues of $(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}$ are the same as the eigenvalues of $\mathbf{B}^{-1}\Pi\mathbf{A}$, with eigenvectors $\Pi\mathbf{v}_i$. □

In other words, $(\mathbf{B}^{-1}\Pi + \mathbf{Q})\mathbf{A}$ has eigenvectors $\widehat{\Pi}\mathbf{v}_i$, with $\widehat{\Pi}\mathbf{v}_i$ the eigenvectors of $\mathbf{B}^{-1}\Pi\mathbf{A}$.

3.2 Main result: IDR projection theorem

Since the spectra of $\mathbf{P}_k\mathbf{A}$ and $\Pi_k\mathbf{A}$ merely differ in the sense that the zero eigenvalues of $\Pi_k\mathbf{A}$ are shifted to one in $\mathbf{P}_k\mathbf{A}$ (cf. Theorem 3.2), we will mainly focus on the spectrum of $\Pi_k\mathbf{A}$ from now on. We first state the main result of this paper: the IDR projection theorem. It describes the complete spectrum of the active IDR(s) systems $\Pi_k\mathbf{A}$, relating the spectrum of the active system of cycle k to all the previous cycles.

Theorem 3.3 (IDR Projection Theorem)**.** *Let* $k \in \mathbb{N}_0$ *and let* $\mathbf{W}$ *be an* $n \times s$ *matrix such that* $\mathbf{W} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})$*. The eigenvalues of* $\Pi_k\mathbf{A}$ *are in*

$$\sigma(\Pi_k\mathbf{A}) = \{0\} \cup \{\lambda \mid P(\lambda) = 0\} \cup \{\lambda \mid \det(\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}) = 0\}, \tag{28}$$

where the polynomial $P(\lambda) \equiv \prod_{j=1}^{k}(\lambda - \mu_j)$ *as defined in Theorem 2.4. The zero eigenvalue and the zeros* μ_j *of the polynomial* P *are all eigenvalues of* $\Pi_k\mathbf{A}$ *that have geometric multiplicity at least* s*.*

Before we give the proof of the IDR projection theorem, we will give some preliminary results.

Let $\mathbf{V}$ be an $n \times s$ matrix such that $\widetilde{\mathbf{R}}^*\mathbf{V}$ is non-singular. Define (cf. Def. 2.5)

$$\Pi_k \equiv \mathbf{I} - \mathbf{V}\mathbf{E}^{-1}\widetilde{\mathbf{R}}^* \quad \text{where} \quad \mathbf{E} \equiv \widetilde{\mathbf{R}}^*\mathbf{V}. \tag{29}$$

Note that

$$\widetilde{\mathbf{R}}^*\Pi_k = \mathbf{0}. \tag{30}$$

In particular, we have $\mathbf{V} \equiv \mathbf{A}\mathbf{U}_k$ in IDR (cf. Alg. 2.1 and Prop 2.13).

Proposition 3.4. *Assume* $\mathrm{span}(\mathbf{V}) \subset \mathcal{G}_k$. *Then each* $\nu \in \{\mu_1, \ldots, \mu_k\}$ *is an eigenvalue of* $\Pi_k\mathbf{A}$ *with geometric multiplicity at least* s.

Proof. The equality in (6) tells us that $\mathbf{V} = (\mathbf{A} - \nu\mathbf{I})\mathbf{W}$ for some $n \times s$ matrix $\mathbf{W} \perp \widetilde{\mathbf{R}}$. Hence

$$(\Pi_k\mathbf{A} - \nu\mathbf{I})\mathbf{W} = (\mathbf{A} - \nu\mathbf{I})\mathbf{W}(\mathbf{I} - \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{W}).$$

Since $\mathbf{E} = \widetilde{\mathbf{R}}^*\mathbf{V} = \widetilde{\mathbf{R}}^*(\mathbf{A} - \nu\mathbf{I})\mathbf{W} = \widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{W}$, we see that $\mathbf{W}$ is in the kernel of $\Pi_k\mathbf{A} - \nu\mathbf{I}$, that is, $\mathrm{span}(\mathbf{W})$ consists of eigenvectors of $\Pi_k\mathbf{A}$ with eigenvalue ν. □

Again using the equality in (6), we know that there exists a $\mathbf{W} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})$ such that $\mathbf{V} = P(\mathbf{A})\mathbf{W}$ for some polynomial P of exact degree k. In the following, let $\mathbf{W}$ be such a matrix and assume $\mathbf{E} \equiv \widetilde{\mathbf{R}}^*\mathbf{V}$ is non–singular.

Proposition 3.5. *Assume* $\lambda \in \mathbb{C}$ *is not an eigenvalue of* $\mathbf{A}$ *and* $\lambda P(\lambda) \neq 0$. *Then, a non-trivial vector* $\mathbf{x}$ *is an eigenvector of* $\Pi_k\mathbf{A}$ *with eigenvalue* λ *if and only if* $\mathbf{x} = (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha$ *for some* $\alpha \in \mathbb{C}^s$, $\alpha \neq 0$ *such that* $\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}\alpha = 0$.

Proof. Assume $\Pi_k\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. Since $\Pi_k\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x} - \mathbf{V}\mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{x}$, we see that $\mathbf{A}\mathbf{x} - \lambda\mathbf{x} = \mathbf{V}\alpha$ for some $\alpha \in \mathbb{C}^s$ (actually, $\alpha = \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{x}$). The scalar λ is not an eigenvalue of $\mathbf{A}$. Therefore, $\mathbf{x} = (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha$. Hence,

$$\alpha = \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{A}(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha = \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\mathbf{V}\alpha + \mathbf{E}^{-1}\widetilde{\mathbf{R}}^*\lambda(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha.$$

Since $\mathbf{E} = \widetilde{\mathbf{R}}^*\mathbf{V}$, this is equivalent to $0 = \lambda\mathbf{E}^{-1}\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha$. Therefore,

$$0 = \widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha. \tag{31}$$

Here, we used that, by assumption $\lambda \neq 0$. Conversely, if (31) holds, then it is easy to check that $\mathbf{x} = (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha$ defines an eigenvector of $\Pi_k\mathbf{A}$.

Note that $(\mathbf{A} - \lambda\mathbf{I})^{-1}(\mathbf{A} - \mu\mathbf{I}) = \mathbf{I} + (\lambda - \mu)(\mathbf{A} - \lambda\mathbf{I})^{-1}$. Since, $P(\zeta) = \gamma\Pi_{j=1}^k(\zeta - \mu_j)$ ($\zeta \in \mathbb{C}$) for certain scalars $\mu_j \in \mathbb{C}$ (the zeros of P) and a scalar $\gamma \in \mathbb{C}$ (a scaling factor), and $\mathbf{W} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})$, we see that $\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}P(\mathbf{A})\mathbf{W} = P(\lambda)\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}$. Hence, (31) holds if and only if $P(\lambda) = 0$ or $\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}\alpha = 0$. □

Note 3.6. Note that the above proof relies on the fact that $\mathbf{W} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})$.

Note 3.7. Note that the above arguments also prove Prop. 3.4: Because, if $\lambda = \mu_i$ for some zero μ_i of P, then, for any $\alpha \in \mathbb{C}^s$, we have that $P(\lambda)\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}\alpha = 0$ and any $\mathbf{x} \equiv (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{V}\alpha$ is an eigenvector of $\Pi_k\mathbf{A}$.

We can now give a proof of Theorem 3.3:

Proof. [Proof of IDR Projection Theorem] Use (iii) of Lemma 2.6 and use Proposition 3.5. □

The proposition belows shows that in IDR the latter set of (28) is independent of P:

Proposition 3.8. *In IDR,* $\mathbf{W}$ *depends on* k *but is independent of* P*, that is, independent of* $\mu_1, \ldots, \mu_k$*.*

Proof. The projection Π_k applied to some vector forms vectors of the form $P_k(\mathbf{A})\mathbf{s} - P_k(\mathbf{A})\mathbf{W}\alpha = P_k(\mathbf{A})(\mathbf{s} - \mathbf{W}\alpha) \perp \widetilde{\mathbf{R}}$. If $\mathbf{s}, \mathbf{W} \perp \mathcal{K}_{k-1}(\mathbf{A}^*, \widetilde{\mathbf{R}})$, then the orthogonality condition from Π_k is equivalent to requiring $\mathbf{s} - \mathbf{W}\alpha \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}})$. In particular, the vector $\mathbf{s} - \mathbf{W}\alpha$ is independent of P if $\mathbf{s}$ and $\mathbf{W}$ are independent of P.

Following the inductive construction of the $\mathbf{V}$ matrices in IDR (as we saw in Theorem 2.2), the above argument proves the proposition. □

3.3 Numerical examples

To illustrate the clustering effect of IDR(s) algorithms, we will inspect the spectra of $\mathbf{A}$ and $\Pi_k\mathbf{A}$ of each cycle k while solving a small test problem using the IDR algorithm from Alg. 2.1. In other words, we take $\mathbf{V} = \mathbf{A}\mathbf{U}_k$ in the projection Π from (29). We will use either a minimum residual strategy for computing ω_k or set $\omega_k = 1$ in each cycle $k > 0$.

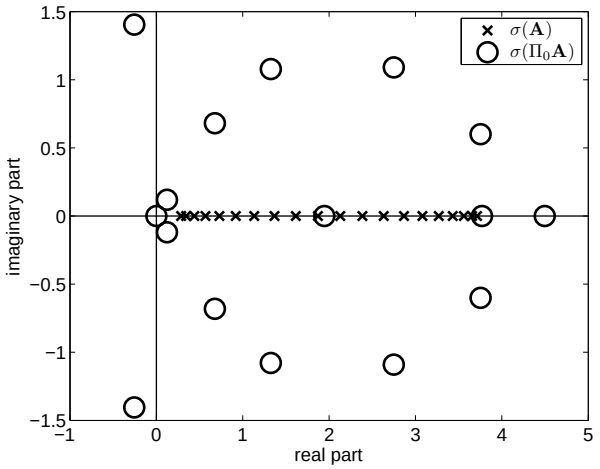
The test problem is a finite difference discretisation of 1D convection–diffusion problem using central differences for the first derivative. The system has size $n = 20$ and the mesh Péclet number p_h is equal to $1/2$. To more precise, the diagonal elements of the coefficient matrix $\mathbf{A}$ are equal to 2, while on the subdiagonal and superdiagonal the values are equal to $-1 - p_h$ and $-1 + p_h$, respectively. The right–hand side vector is equal to $\mathbf{b} = [1 + p_h, 0, \ldots, 0, 1 - p_h]^\top$. Also, we put $s = 5$, so the iteration converges in $n/s = 20/5 = 4$ cycles in exact arithmetic and we set $\mathbf{x}_0 = \mathbf{0}$.

Figure 2 shows the spectra of the active systems if a minimum residual strategy for computing ω_k for $k > 0$ is used. Put $\mu_0 = 0$. This gives $\mu_k \in \{0\} \cup \{2.9, 2.5, 2.2\}$ for $k = 0, 1, 2, 3$, resulting in the final clustered spectrum shown in Fig. 2(d), which solely consists of four eigenvalues each with multiplicity $s = 5$.

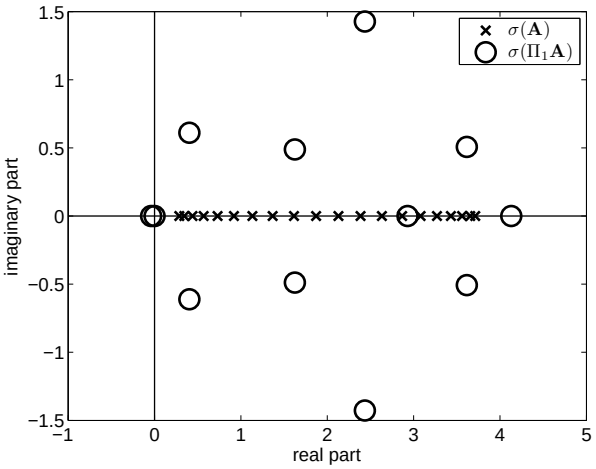
Shown in Fig. 3 are the spectra of the active systems for all four cycles if $\mu_k = 1$ for $k = 1, 2, 3$. As a result, the active system has $10, 15$, and 20 unit eigenvalues in cycle $1, 2$, and 3, respectively.

These results also indicate that the (total) spectrum seems to increasingly converge towards the values of μ_k with each cycle.

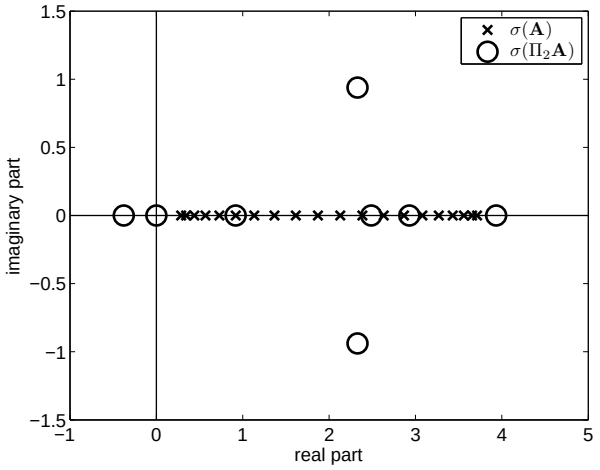
Also, it can be observed from these experiments that the "non–clustered" part of the spectrum is independent of the choices for $\mu_1, \ldots, \mu_k$, as indicated by Proposition 3.8.



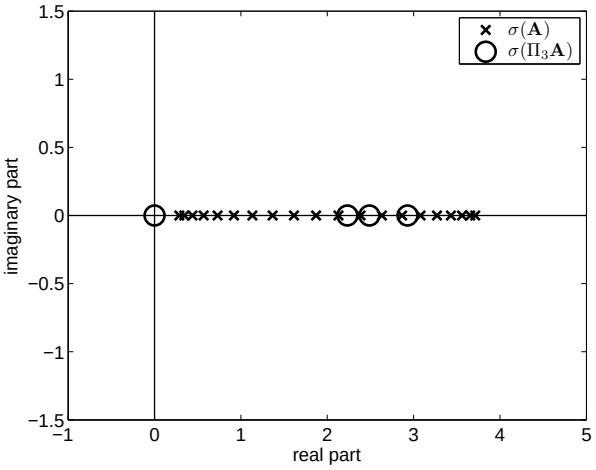
(a) Cycle 0, $\mu_k \in \{0\}$.



(b) Cycle 1, $\mu_k \in \{0\} \cup \{2.9\}$.



(c) Cycle 2, $\mu_k \in \{0\} \cup \{2.9, 2.5\}$.



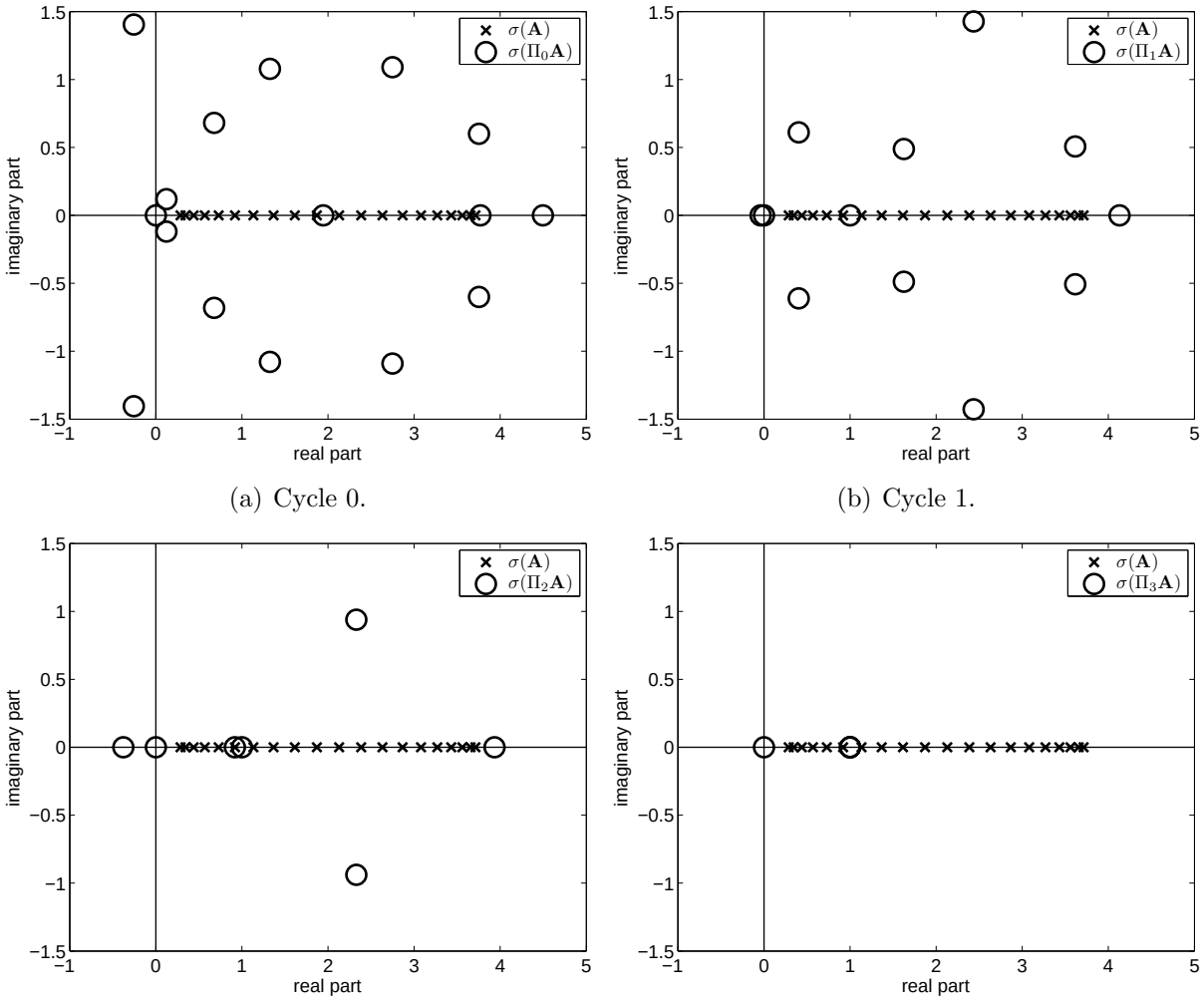
(d) Cycle 3 (converged), $\mu_k \in \{0\} \cup \{2.9, 2.5, 2.2\}$.

Figure 2: $\mathbf{B} = \mathbf{I}, n = 20, s = 5$, four cycles in total, $\mu_k \in \{0\} \cup \{2.9, 2.5, 2.2\}$ for $k = 0, 1, 2, 3$.

3.4 Discussion

In each IDR(s) cycle k, s *additional* eigenvalues of $\mathbf{P}_k\mathbf{A}$ are shifted to μ_k and the matrix $\mathbf{P}_k\mathbf{A}$ (and therefore $\Pi_k\mathbf{A}$) has $k+1$ eigenvalues of geometric multiplicity s. The IDR projection theorem holds independently of the way a basis for $\mathcal{K}_s(\Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}, \Pi_{k-1}\mathbf{A}\mathbf{B}^{-1}\mathbf{r}'_{k-1})$ is computed.

In standard deflation methods, the deflation subspace matrices $\mathbf{U}$ and $\widetilde{\mathbf{R}}$ are often equal to each other and consists of (approximate) eigenvectors belonging to eigenvalues of $\mathbf{A}$ that are small in norm. Also, the matrix $\mathbf{B}$ is a traditional preconditioner that deals with the extremes of the spectrum. In IDR(s), the space span($\mathbf{U}_k$) is not related to any *specific* components of the spectrum of $\mathbf{A}$. What is important is that s new eigenvalues of $\mathbf{P}_k\mathbf{A}$



(a) Cycle 0.

(b) Cycle 1.

(c) Cycle 2.

(d) Cycle 3 (converged).

Figure 3: $\mathbf{B} = \mathbf{I}, n = 20, s = 5$, four cycles in total, $\mu_k \in \{0\} \cup \{1\}$ for all k.

are deflated in each cycle. In this context, the ideal role of the preconditioner $\mathbf{B}$ is less clear.

In the multigrid context (for elliptic problems), the matrix $\mathbf{B}^{-1}$ should act as a smoothing (or relaxation) step that eliminates the high–frequency errors. Interestingly, in IDR(s) the operator $\mathbf{C}_2 = \omega_{k+1}\mathbf{B}^{-1}$ acts similarly, "smoothing" the new primary residual $\mathbf{r}_{k+1}$ in norm.

During the IDR(s) iteration process, new spectral components of $\mathbf{P}_k\mathbf{A}$ are continuously projected out of the residual, while retaining spectral information from all previous cycles.

Note that if we take $s = n$, then $\sigma(\Pi_0) = \sigma(\Pi_0\mathbf{A}) = \{0\}$ and $\sigma(\mathbf{P}_0\mathbf{A}) = \{1\}$, which means that the iteration terminates within a single cycle.

4 Explicitly deflated IDR(s)

Similar to other Krylov subspace methods, IDR(s) methods can be explicitly preconditioned with deflation methods. In Sect. 4.1 a comparison is made between two variants of explicitly deflated IDR(s) and IDR(s) where the IDR deflation matrics are augmented with traditional deflation vectors. These results gives rise to possible good choices of $\widetilde{\mathbf{R}}$ and ω_k, which are discussed in Sect. 4.2. Numerical experiments that illustrate the theoretical results are given in Sect. 4.3.

4.1 Deflation vs. augmentation

4.1.1 Deflation

In standard deflation methods, the deflation matrices are defined as follows (cf. Definition 2.5)

$$\Pi^{\text{def}} \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}^{\text{def}} \quad \text{and} \quad \widehat{\Pi}^{\text{def}} \equiv \mathbf{I} - \mathbf{Q}^{\text{def}}\mathbf{A} \quad \text{where} \quad \mathbf{Q}^{\text{def}} \equiv \mathbf{Z}(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}\mathbf{Z}^* \tag{32}$$

where $\mathbf{Z} \in \mathbb{C}^{n\times t}$ is a deflation–subspace matrix of full rank and $\mathbf{E}^{\text{def}} \equiv \mathbf{Z}^*\mathbf{A}\mathbf{Z}$ is assumed to be invertible. Note that we have (cf. Lemma 2.6)

$$\Pi^{\text{def}}\mathbf{A}\mathbf{Z} = \mathbf{0}_{n,t} \quad \text{and} \quad \mathbf{Q}^{\text{def}}\mathbf{A}\mathbf{Z} = \mathbf{Z}. \tag{33}$$

To distinguish the standard deflation projection from the IDR projection, a superscript is added. Using this notation, the IDR(s) operator is written as (cf. (16))

$$\mathbf{P}_k^{\text{idr}} = \omega_{k+1}\Pi_k^{\text{idr}} + \mathbf{Q}_k^{\text{idr}}, \tag{34}$$

where Π_k^{idr} and $\mathbf{Q}_k^{\text{idr}}$ are the same as Π_k and $\mathbf{Q}_k$ in Prop 2.8.

Two variants of so–called "explicitly deflated" IDR(s) will be considered. The first one is based on the DEF1 variant [23, §2.3.2] where IDR(s) is used to solve the deflated system

$$\Pi^{\text{def}}\mathbf{A}\mathbf{x}' = \Pi^{\text{def}}\mathbf{b}. \tag{35}$$

The solution $\mathbf{x}'$ to the system (35) is related to the solution $\mathbf{x}$ of the original system $\mathbf{A}\mathbf{x} = \mathbf{b}$ as follows:

$$\mathbf{x} = \mathbf{Q}^{\text{def}}\mathbf{r}_0 + \widehat{\Pi}^{\text{def}}\mathbf{x}', \tag{36}$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. Also, note that the system (35) is singular, since $\Pi^{\text{def}}\mathbf{A}\mathbf{Z} = \mathbf{0}_{n,t}$. A singular system can still be solved as long as it is consistent, i.e., $\mathbf{b} \in \text{span}(\mathbf{A})$, see [9]. This is true for our case, since the same projection is applied to both sides of (35).

The second deflation method is based on the A–DEF1 variant [23, §2.3.3], where IDR(s) is applied to the deflated system

$$\mathbf{P}^{\text{adef1}}\mathbf{A}\mathbf{x} = \mathbf{P}^{\text{adef1}}\mathbf{b}, \tag{37}$$

where $\mathbf{P}^{\text{adef1}} = \Pi^{\text{def}} + \mathbf{Q}^{\text{def}}$ (cf. Sect. 2.2). The only difference between A–DEF1 and DEF1 is that the zero eigenvalues of $\Pi^{\text{def}}\mathbf{A}$ are shifted to one in $\mathbf{P}^{\text{adef1}}\mathbf{A}$.

The main motivation behind using an adapted deflation method such as A–DEF1 instead of DEF1 is as follows. It is known that perturbations (roundoff errors, perturbed starting vectors, inaccurate preconditioning solves, inaccurate Galerkin solves) can transform the zero eigenvalues of $\Pi^{\text{def}}\mathbf{A}$ into near–zero eigenvalues, making them potentially harmful to the convergence process. In an adapted deflation method, the corresponding near–unit eigenvalues are harmless.

4.1.2 Augmentation

We have shown that IDR(s) itself can be seen as an adapted deflation method. Therefore, a natural way to combine deflation–type preconditioners with IDR(s) is to augment the deflation subspace matrices $\mathbf{U}$ and $\widetilde{\mathbf{R}}$ with the deflation subspace matrix $\mathbf{Z}$ as follows:

$$\underline{\mathbf{U}}_k \equiv [\mathbf{U}_k \ \mathbf{Z}] \quad \text{and} \quad \underline{\widetilde{\mathbf{R}}} \equiv [\widetilde{\mathbf{R}} \ \mathbf{Z}]. \tag{38}$$

The IDR(s) deflation matrices are then

$$\Pi_k^{\text{idr}'} \equiv \mathbf{I} - \mathbf{A}\mathbf{Q}_k^{\text{idr}'}, \quad \text{where} \quad \mathbf{Q}_k^{\text{idr}'} \equiv \underline{\mathbf{U}}_k \underline{\mathbf{E}}_k^{-1} \underline{\widetilde{\mathbf{R}}}^* \quad \text{and} \quad \underline{\mathbf{E}}_k \equiv \underline{\widetilde{\mathbf{R}}}^* \mathbf{A} \underline{\mathbf{U}}_k, \tag{39}$$

with dimensions

$$|\underline{\mathbf{U}}_k| = |\underline{\widetilde{\mathbf{R}}}| = n \times (s+t) \quad \text{and} \quad |\underline{\mathbf{E}}_k| = (s+t) \times (s+t). \tag{40}$$

The corresponding augmented IDR(s) operator is

$$\mathbf{P}_k^{\text{idr}'} = \omega_{k+1} \Pi_k^{\text{idr}'} + \mathbf{Q}_k^{\text{idr}'}. \tag{41}$$

We call this approach "augmented IDR(s)".

4.1.3 Comparisons

Since IDR(s) is analogous to the A–DEF1 method, we will compare augmented IDR(s) to IDR(s) explicitly deflated with A–DEF1. For completeness, comparisons are also made with the DEF1 method.

Unless stated otherwise, no assumptions are made on the columns of $\mathbf{Z}$ or $\widetilde{\mathbf{R}}$. However, in some cases we make one of the following assumptions:

Assumption 1. The matrix $\widetilde{\mathbf{R}}$ is orthogonal to the matrix $\mathbf{Z}$.

Assumption 2. The matrix $\widetilde{\mathbf{R}}$ is orthogonal to the matrix $\mathbf{AZ}$.

In the following, let (cf. Theorem 3.3)

$$\Lambda_k \equiv \{\lambda \mid P_k(\lambda) = 0\} \cup \{\lambda \mid \det(\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}) = 0\} \tag{42}$$

Proposition 4.1. *We distinguish between three deflation–type iterative processes of IDR(s):*

(i) Augmented IDR(s): We have for arbitrary $\mathbf{Z}$

$$\sigma\left(\Pi^{\mathrm{idr}'}\mathbf{A}\right) = \{0\} \cup \Lambda_k \tag{43}$$

where the zero eigenvalue has geometric multiplicity $s+t$ *and*

$$\sigma\left(\left[\Pi^{\mathrm{idr}'} + \mathbf{Q}^{\mathrm{idr}'}\right]\mathbf{A}\right) = \{1\} \cup \Lambda_k \tag{44}$$

where the unit eigenvalue also has geometric multiplicity $s+t$

(ii) IDR(s)-DEF1: Let $\overline{\mathbf{A}}_1 = \Pi^{\mathrm{def}}\mathbf{A}$*. We have for arbitrary* $\mathbf{Z}$

$$\sigma\left(\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_1\right) \equiv \sigma\left((\mathbf{I} - \overline{\mathbf{A}}_1\mathbf{U}(\widetilde{\mathbf{R}}^*\overline{\mathbf{A}}_1\mathbf{U})^{-1}\widetilde{\mathbf{R}}^*)\overline{\mathbf{A}}_1\right) = \{0\} \cup \Lambda_k \tag{45}$$

where the zero eigenvalue has geometric multiplicity $s+t$ *and*

$$\sigma\left(\left[\overline{\Pi}^{\mathrm{idr}} + \overline{\mathbf{Q}}^{\mathrm{idr}}\right]\overline{\mathbf{A}}_1\right) = \{0\} \cup \{1\} \cup \Lambda_k \tag{46}$$

where the zero eigenvalue has geometric multiplicity t *and the unit eigenvalue has geometric multiplicity* s*. The overlines of the operators signify the fact that they employ vectors based on the projected system.*

(iii) IDR(s)-ADEF1: Let $\overline{\mathbf{A}}_2 = (\Pi^{\mathrm{def}} + \mathbf{Q}^{\mathrm{def}})\mathbf{A}$*. Given Assumption 1, we have*

$$\sigma\left(\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_2\right) = \{0\} \cup \{1\} \cup \Lambda_k \tag{47}$$

where the zero eigenvalue has geometric multiplicity s *and the unit eigenvalue has geometric multiplicity* t*. Also, we have*

$$\sigma\left(\left[\overline{\Pi}^{\mathrm{idr}} + \overline{\mathbf{Q}}^{\mathrm{idr}}\right]\overline{\mathbf{A}}_2\right) = \{1\} \cup \Lambda_k \tag{48}$$

where the unit eigenvalue has geometric multiplicity $s+t$*.*

Proof. We have using Theorem 3.2:

(i) Augmented IDR(s): For (43) we have

$$\Pi^{\mathrm{idr}'}\mathbf{A}\underline{\mathbf{U}} = \mathbf{0}_{n,s+t} \tag{49}$$

and for (44) we have

$$\mathbf{Q}^{\mathrm{idr}'}\mathbf{A}\underline{\mathbf{U}} = \underline{\mathbf{U}}_{n,s+t} \tag{50}$$

(ii) IDR(s)-DEF1: For (45), we have

$$\overline{\mathbf{A}}_1\mathbf{Z} = \mathbf{0}_{n,t} \tag{51}$$

$$\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_1\mathbf{U} = \mathbf{0}_{n,s} \tag{52}$$

and for (46) we have

$$\overline{\mathbf{Q}}^{\mathrm{idr}}\overline{\mathbf{A}}_1\mathbf{U} = \mathbf{U}(\widetilde{\mathbf{R}}^*\Pi^{\mathrm{def}}\mathbf{A}\mathbf{U})^{-1}\widetilde{\mathbf{R}}^*)\Pi^{\mathrm{def}}\mathbf{A}\mathbf{U} \tag{53}$$

$$= \mathbf{U}_{n,s} \tag{54}$$

(iii) IDR(s)-ADEF1: Assumption 1 implies that $\widetilde{\mathbf{R}}^*\mathbf{Z} = \mathbf{0}_{s,t}$, so for (47) we have

$$\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_2\mathbf{U} = \mathbf{0}_{n,s} \tag{55}$$

$$\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_2\mathbf{Z} = \overline{\Pi}^{\mathrm{idr}}(\Pi^{\mathrm{def}} + \mathbf{Q}^{\mathrm{def}})\mathbf{A}\mathbf{Z} \tag{56}$$

$$= (\mathbf{I} - \overline{\mathbf{A}}_2\mathbf{U}(\widetilde{\mathbf{R}}^*\overline{\mathbf{A}}_2\mathbf{U})^{-1}\widetilde{\mathbf{R}}^*)\mathbf{Z} \tag{57}$$

$$= \mathbf{Z}_{n,t} \tag{58}$$

For (48) we have

$$\overline{\mathbf{Q}}^{\mathrm{idr}}\overline{\mathbf{A}}_2\mathbf{U} = \mathbf{U}(\widetilde{\mathbf{R}}^*(\Pi^{\mathrm{def}} + \mathbf{Q}^{\mathrm{def}})\mathbf{A}\mathbf{U})^{-1}\widetilde{\mathbf{R}}^*)(\Pi^{\mathrm{def}} + \mathbf{Q}^{\mathrm{def}})\mathbf{A}\mathbf{U} \tag{59}$$

$$= \mathbf{U}_{n,s} \tag{60}$$

This concludes the proof. □

Proposition 4.1 implies that in the generic case and in exact arithmetic all three variants compute the exact solution in at most $\frac{n-t}{s}$ IDR cycles.

In iterative process (iii), IDR(s) is applied to a system with $n-t+1$ distinct eigenvalues. If we do not make Assumption 1, the exact solution for this case is computed in at most $\lceil\frac{n-t+1}{s}\rceil$ cycles. Another possibility is to use $t+1$ deflation vectors for $\mathbf{Z}$.

Ideally, the deflation vectors in $\mathbf{Z}$ approximate the eigenspace corresponding to the unfavourable eigenvalues of $\mathbf{A}$, e.g., eigenvalues small in magnitude. Depending on the deflation technique, these eigenvalues will be shifted to zero or one, removing them from the IDR(s) iteration process. In this way, techniques from domain decomposition and deflation can be easily used in IDR.

Note that augmenting the matrices $\mathbf{U}$ and $\widetilde{\mathbf{R}}$ with deflation vectors is different from increasing s in "standard" IDR(s). Also, if in IDR(s) we would set $\mathbf{U} = \widetilde{\mathbf{R}} = \mathbf{Z}$, we obtain a standard Richardson iteration deflated with A–DEF1.

For most applications we have $t \gg s$, which would make the augmented Galerkin matrix too big to solve directly. To avoid this problem, note that in augmented IDR(s) the augmented Galerkin matrix $\underline{\mathbf{E}}_k$ has the following form:

$$\underline{\mathbf{E}}_k = \begin{bmatrix}\widetilde{\mathbf{R}}^*\\ \mathbf{Z}^*\end{bmatrix}\mathbf{A}\begin{bmatrix}\mathbf{U}_k & \mathbf{Z}\end{bmatrix} = \begin{bmatrix}\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k & \widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{Z}\\ \mathbf{Z}^*\mathbf{A}\mathbf{U}_k & \mathbf{Z}^*\mathbf{A}\mathbf{Z}\end{bmatrix} = \begin{bmatrix}\mathbf{E}_{11} & \mathbf{E}_{12}\\ \mathbf{E}_{21} & \mathbf{E}_{22}\end{bmatrix} = \left\{\begin{array}{cc} |s\times s| & |s\times t|\\ |t\times s| & |t\times t|\end{array}\right\}. \tag{61}$$

If $\mathbf{Z}$ consists of subdomain deflation vectors, then $\mathbf{Z}^*\mathbf{A}\mathbf{Z}$ is a diagonal band matrix. To compute $\mathbf{x} = \underline{\mathbf{E}}_k^{-1}\mathbf{b}$ for some $\mathbf{b}$, one can make use of the Schur complement as follows

$$(\mathbf{E}_{11} - \mathbf{E}_{12}\mathbf{E}_{22}^{-1}\mathbf{E}_{21})\mathbf{x}_1 = \mathbf{b}_1 - \mathbf{E}_{12}\mathbf{E}_{22}^{-1}\mathbf{b}_2 \qquad |\mathbf{x}_1| = |\mathbf{b}_1| = s \tag{62}$$

$$\mathbf{E}_{22}\mathbf{x}_2 = \mathbf{b}_2 - \mathbf{E}_{21}\mathbf{x}_1 \qquad |\mathbf{x}_2| = |\mathbf{b}_2| = t \tag{63}$$

Note that there are *three* instances where we have to solve systems involving $\mathbf{E}_{22}$. Also, we have

$$\mathbf{E}_{11} - \mathbf{E}_{12}\mathbf{E}_{22}^{-1}\mathbf{E}_{21} = \widetilde{\mathbf{R}}^*\mathbf{A}(\mathbf{I} - \mathbf{Z}(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}\mathbf{Z}^*\mathbf{A})\mathbf{U}_k \tag{64}$$

$$= \widetilde{\mathbf{R}}^*\mathbf{A}\widehat{\Pi}^{\mathrm{def}}\mathbf{U}_k \tag{65}$$

$$= \widetilde{\mathbf{R}}^*\Pi^{\mathrm{def}}\mathbf{A}\mathbf{U}_k \tag{66}$$

which is exactly the (deflated) Galerkin system of IDR(s)-DEF1.

Let $\mathbf{b}_1 = \widetilde{\mathbf{R}}^*\mathbf{A}$ and $\mathbf{b}_2 = \mathbf{Z}^*\mathbf{A}$. Then we have (omitting the subscript k)

$$\Pi^{\mathrm{idr}'}\mathbf{A} = (\mathbf{I} - \mathbf{A}\underline{\mathbf{U}}\,\underline{\mathbf{E}}^{-1}\underline{\widetilde{\mathbf{R}}}^*)\mathbf{A} \tag{67}$$

$$= \mathbf{A} - \mathbf{A}(\mathbf{U}\mathbf{x}_1 + \mathbf{Z}\mathbf{x}_2) \tag{68}$$

$$= (\mathbf{I} - \Pi^{\mathrm{def}}\mathbf{A}\mathbf{U}(\widetilde{\mathbf{R}}^*\Pi^{\mathrm{def}}\mathbf{A}\mathbf{U})^{-1}\widetilde{\mathbf{R}}^*)(\mathbf{I} - \mathbf{A}\mathbf{Z}(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}\mathbf{Z}^*)\mathbf{A} \tag{69}$$

$$= \overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_1 \tag{70}$$

This implies that the spectrum of $\Pi^{\mathrm{idr}'}\mathbf{A}$ is the same as the spectrum of $\overline{\Pi}^{\mathrm{idr}}\overline{\mathbf{A}}_1$, which is in accordance with Proposition 4.1, i.e., properties (43) and (45). However, it is expected that in practical applications the IDR(s) processes will behave differently, since in augmented IDR(s) $s+t$ eigenvalues of the active systems are shifted to one, while in IDR(s)-DEF1 t eigenvalues remain zero.

That is, perturbations (roundoff errors, perturbed starting vectors, inaccurate preconditioning solves, inaccurate Galerkin solves) can transform the zero eigenvalues of the active systems in IDR(s)-DEF1 into near–zero eigenvalues, which may result in numerical instabilities. This suggests that augmented IDR(s) will be numerically more stable than IDR(s)-DEF1. A similar argument can be used to show that IDR(s)-ADEF1 is numerically more stable than IDR(s)-DEF1.

Some of the advantages of using augmented IDR(s) as opposed to IDR(s)-DEF1 or IDR(s)-ADEF1 are:

- Possible increased numerical stability, since unfavourable eigenvalues are shifted to one instead of zero.
- It is a natural way of introducing deflation–type preconditioner into IDR(s) algorithms.
- Given a specific choice of $\widetilde{\mathbf{R}}$ (see Sect. 4.2), the two by two block augmented Galerkin system (61) can be inverted efficiently.

Algorithm 4.1 Computation of $\Pi^{\text{idr}'}\mathbf{y}$

1: $\mathbf{b}_1 = \widetilde{\mathbf{R}}^*\mathbf{y}$
2: $\mathbf{b}_2 = \mathbf{Z}^*\mathbf{y}$
3: **Solve** $(\mathbf{Z}^*\mathbf{A}\mathbf{Z})\mathbf{b}_3 = \mathbf{b}_2$
4: $\mathbf{b}_4 = \widetilde{\mathbf{R}}^*(\mathbf{A}\mathbf{Z})\mathbf{b}_3$
5: $\mathbf{b}_5 = \mathbf{b}_1 - \mathbf{b}_4$
6: **Solve** $(\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k - \widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{Z}(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}\mathbf{Z}^*\mathbf{A}\mathbf{U}_k)\mathbf{x}_1 = \mathbf{b}_5$
7: $\mathbf{b}_6 = \mathbf{Z}^*(\mathbf{A}\mathbf{U}_k)\mathbf{x}_1$
8: $\mathbf{b}_7 = \mathbf{b}_2 - \mathbf{b}_6$
9: **Solve** $(\mathbf{Z}^*\mathbf{A}\mathbf{Z})\mathbf{x}_2 = \mathbf{b}_7$
10: $\mathbf{y}_1 = \mathbf{A}\mathbf{U}_k\mathbf{x}_1$
11: $\mathbf{y}_2 = \mathbf{A}\mathbf{Z}\mathbf{x}_2$
12: $\Pi^{\text{idr}'}\mathbf{y} = \mathbf{y} - \mathbf{y}_1 - \mathbf{y}_2$

	theory		implementation		
method	$\Pi^{\text{def}}\mathbf{y}$	$\mathbf{Q}^{\text{def}}\mathbf{y}$	IP/MV	AXPY	GSS
IDR(s)-DEF1	1	0	2	1	1
IDR(s)-ADEF1	1	1	3	1	1
augmented IDR(s)	n/a	n/a	3	1	1

Table 1: Extra computational cost per MV compared to "standard" IDR(s).

- Because IDR(s)-ADEF1 is applied to a system with $n-t+1$ distinct eigenvalues, it converges in more cycles than augmented IDR(s) in exact arithmetic (i.e., $(n-t+1)/s$ cycles instead of $(n-t)/s$) if no special assumptions are made.

To apply the operator $\Pi^{\text{idr}'}\mathbf{y}$ to some vector $\mathbf{y}$, we need to perform the steps shown in Alg. 4.1. For efficiency, the following (small) matrices can be computed/factored and stored beforehand:

$$(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}, \quad \mathbf{A}\mathbf{Z}, \quad \text{and} \quad \widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{Z}(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}. \tag{71}$$

Note that the matrices $\mathbf{Z}^*\mathbf{A}\mathbf{U}_k$ and $\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k$ have to be recomputed in each cycle k, but they can be reused within a cycle. The matrix $\mathbf{A}\mathbf{U}_k$ is readily available and does not require additional MVs.

In practical algorithms, the operation $\mathbf{Q}^{\text{idr}'}\mathbf{y}$ also has to be computed. This can be done efficiently by reusing quantities from $\Pi^{\text{idr}'}\mathbf{y}$, similar to the A-DEF1 method.

4.2 Possible good choices for $\widetilde{\mathbf{R}}$ and ω_k

4.2.1 Choice of $\widetilde{\mathbf{R}}$

Inspection of the Galerkin matrix $\underline{\mathbf{E}}_k$ from (61) belonging to augmented IDR(s) shows that the submatrices $\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k$ and $\mathbf{Z}^*\mathbf{A}\mathbf{U}_k$ have to be recomputed each cycle. The non–

Algorithm 4.2 Computation of $\Pi^{\text{idr}'}\mathbf{y}$ with special choice for $\widetilde{\mathbf{R}}$.

1: $\mathbf{b}_1 = \widetilde{\mathbf{R}}^*\mathbf{y}$
2: $\mathbf{b}_2 = \mathbf{Z}^*\mathbf{y}$ *// IP/MV #1*
3: **Solve** $(\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k)\mathbf{x}_1 = \mathbf{b}_1$
4: $\mathbf{b}_3 = (\mathbf{Z}^*\mathbf{A}\mathbf{U}_k)\mathbf{x}_1$
5: $\mathbf{b}_4 = \mathbf{b}_2 - \mathbf{b}_3$
6: **Solve** $(\mathbf{Z}^*\mathbf{A}\mathbf{Z})\mathbf{x}_2 = \mathbf{b}_4$ *// GSS #1*
7: $\mathbf{y}_1 = (\mathbf{A}\mathbf{U}_k)\mathbf{x}_1$
8: $\mathbf{y}_2 = (\mathbf{A}\mathbf{Z})\mathbf{x}_2$ *// IP/MV #2*
9: $\Pi^{\text{idr}'}\mathbf{y} = \mathbf{y} - \mathbf{y}_1 - \mathbf{y}_2$ *// AXPY #1*

zero matrix $\mathbf{Z}^*\mathbf{A}\mathbf{Z}$ is fixed. However, the matrix $\widetilde{\mathbf{R}}$ can be chosen such that $\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{Z} = \mathbf{0}$ using an orthogonal projection such as (cf. Assumption 2):

$$\widetilde{\mathbf{R}}' = (\mathbf{I} - \mathbf{A}\mathbf{Z}((\mathbf{A}\mathbf{Z})^*\mathbf{A}\mathbf{Z})^{-1}(\mathbf{A}\mathbf{Z})^*)\widetilde{\mathbf{R}} \quad \Leftrightarrow \quad \widetilde{\mathbf{R}}' \perp \mathbf{A}\mathbf{Z}. \tag{72}$$

The resulting Galerkin matrix is then lower block diagonal:

$$\underline{\mathbf{E}}_k = \begin{bmatrix} \mathbf{E}_{11} & \varnothing \\ \mathbf{E}_{21} & \mathbf{E}_{22} \end{bmatrix}, \tag{73}$$

which simplifies the computation of $\mathbf{x} = \underline{\mathbf{E}}_k^{-1}\mathbf{b}$. Using forward substition, we can compute

$$\mathbf{E}_{11}\mathbf{x}_1 = \mathbf{b}_1 \tag{74}$$
$$\mathbf{E}_{22}\mathbf{x}_2 = \mathbf{b}_2 - \mathbf{E}_{21}\mathbf{x}_1. \tag{75}$$

Note that $|\mathbf{x}_1| = s$ and that $|\mathbf{x}_2| = t$. The corresponding computation of $\Pi^{\text{idr}'}\mathbf{y}$ using this choice of $\widetilde{\mathbf{R}}$ is given in Alg. 4.2.

As before, for the computation of $\mathbf{Q}^{\text{idr}'}\mathbf{y}$ quantities from $\Pi^{\text{idr}'}\mathbf{y}$ can be reused. Also, the matrices $(\mathbf{Z}^*\mathbf{A}\mathbf{Z})^{-1}$ and $\mathbf{A}\mathbf{Z}$ can be computed and stored beforehand. The matrices $\mathbf{Z}^*\mathbf{A}\mathbf{U}_k$ and $(\widetilde{\mathbf{R}}^*\mathbf{A}\mathbf{U}_k)^{-1}$ can be computed at the start of cycle k and reused during a cycle. Also, the matrix $\mathbf{A}\mathbf{U}_k$ is readily available and does not require additional MVs.

Tab. 1 lists the *additional* computational cost of the three deflation approaches compared to a non–deflated IDR(s) method. The term GSS denotes a "Galerkin System Solve" such as computing the solution to $(\mathbf{Z}^*\mathbf{A}\mathbf{Z})\mathbf{x} = \mathbf{y}$ for some $\mathbf{y}$. Also, depending on the type of deflation vectors, an operation involving $\mathbf{Z}$ either counts as a MV or as an inner product (IP). Note that the third IP/MV for augmented IDR(s) is included in the application of $\mathbf{Q}^{\text{idr}'}$.

4.2.2 Choice of μ_k

Generally speaking, the goal of a preconditioner is to cluster the spectrum of the preconditioned system around one (after an appropriate scaling). Choosing $\mu_k = 1$ for all k would

be the most effective spectral choice, since in this case the spectrum of the active system will become increasingly more clustered around one. That is, we have (in exact arithmetic) for $k \in \mathbb{N}_0$:

$$\sigma(\mathbf{P}_k\mathbf{A}) = \{1\} \cup \{\lambda \mid \det(\widetilde{\mathbf{R}}^*(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{W}) = 0\} \tag{76}$$

where the eigenvalue 1 has geometric multiplicity $(k+1)s$ and

$$\sigma(\mathbf{P}_m\mathbf{A}) = \{1\} \tag{77}$$

where $m = n/s - 1$ and where the eigenvalue 1 has geometric multiplicity n. However, if at the same time the remaining eigenvalues of an active system are located close to zero, convergence of the iteration process within that cycle may be hampered. Numerical experiments seem to indicate that even though this can happen in some particular cycle, this will generally not be the case in the next cycle.

Continuing this line of reasoning, it can be argued that the *value* of μ_k does not significantly affect the convergence process, in particular for large s. The key property of IDR(s) methods is that the spectrum of the active system becomes increasingly more *clustered*. The *location* of the clustered spectrum (i.e., the values of μ_k) is less important. Nevertheless, using a near–zero μ_k should be avoided, since it could result in a badly conditioned active system.

For smaller s, the clustering property is much less pronounced and choosing an appropriate value of μ_k will be more important. This effect is observed in many experiments, see for example [21, §6.4] and [20]. For a detailed mathematical analysis of the influence of the factors μ_k on the IDR iteration process, see [20].

4.3 Numerical examples

In the following experiments, the matrix $\mathbf{Z}$ consists of random and orthogonalised vectors. These experiments are for illustrative purposes only and the test problem is the 1D convection–diffusion problem from Sect. 3.3 with $n = 25$ and $s = 5$. We use the IDR method from Alg. 2.1 and set $\omega_k = 1$ for all k. The finite convergence behaviour of the following five iteration processes will be compared:

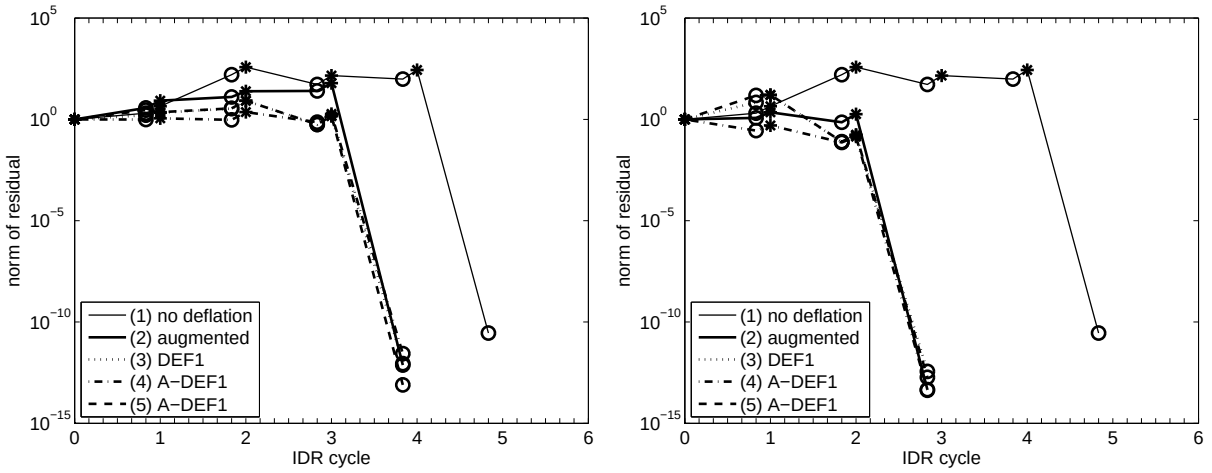
(1) Non–deflated IDR(s) (i.e., $t = 0$).

(2) Augmented IDR(s) with Assumption 2 in order to efficiently invert the two by two block system (73), see Sect. 4.2.

(3) IDR(s)-DEF1.

(4) IDR(s)-ADEF1 with an additional deflation vector for $\mathbf{Z}$.

(5) IDR(s)-ADEF1 with Assumption 1.



(a) Let $\omega_k = 1$ for all k and $t = s$.

(b) Let $\omega_k = 1$ for all k and $t = 2s$.

Figure 4: Residual norms for IDR(s), $n = 25, s = 5, \mathbf{B} = \mathbf{I}$, showing primary "$*$" and secondary "$\circ$" residuals.

Shown in Fig. 4(a) and Fig. 4(b) are the logarithms of the norms of the primary and secondary residuals using $t = s$ and $t = 2s$ deflation vectors for $\mathbf{Z}$, respectively. The ticks on the horizontal axis represent the number of MVs. Note that one IDR cycle consists of $s+1$ MVs: computing a secondary residual $\mathbf{r}'_{k-1}$ ("$\circ$") involves s MVs, while the computing the next primary residual $\mathbf{r}_k$ ("$*$") uses one MV (see also Sect. 2.1). For iteration process (4), an additional deflation vector is used and the following observations can be made:

- Since the iteration matrices of the four iterative processes are different (and hence the "initiation" matrix $\mathbf{V}_0$), the residual norms are different.
- When convergence occurs the drop in residual norm can be observed for the secondary residuals.
- According to the theory, the four iteration processes (2)–(5) should converge within $(n-t)/s = (25-t)/5$ cycles, i.e., within four and three cycles for $t = s$ and $t = 2s$, respectively. The non–deflated process converges within five cycles. This is in accordance with the numerical results.
- If we do not make Assumption 1, then iteration process (5) converges in at most $(n-t+1)/s$ IDR cycles.

5 Conclusions

By interpreting IDR(s) as a deflation method, interesting properties of the IDR(s) method have been revealed. Firstly, this has lead to the IDR projection theorem, which shows that

the spectrum of the deflated systems in IDR(s) become increasingly more clustered. This can be seen as an intuitive explanation for the excellent convergence properties of IDR(s).

Based on this interpretation, one cycle of IDR(s) can be seen as consisting of three key steps: constructing a unique primary residual, constructing the unique secondary residual, and constructing vectors for a basis of a very specific Krylov subspace.

It also shows that the IDR(s) method is an instantiation of a specific deflation method: a so–called adapted deflation method. The deflation subspace matrix in IDR(s) is updated in each cycle with new information while retaining information from all previous IDR(s) cycles.

Lastly, this interpretation allows for the efficient inclusion of standard deflation–type preconditioners into IDR(s) methods.

Acknowledgments

The work of the first author was financially supported by the Delft Centre for Computational Science and Engineering (DCSE) within the framework of the DCSE project entitled "*Development of an Immersed Boundary Method, Implemented on Cluster and Grid Computers*". The authors wish to thank Jok Tang, Peter Sonneveld, and Kees Vuik for valuable discussions and remarks.

References

[1] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel. Adaptively preconditioned GMRES algorithms. *SIAM Journal on Scientific Computing*, 20(1):243–269, 1998.

[2] Kevin Burrage and Jocelyne Erhel. On the performance of various adaptive preconditioned GMRES strategies. *Numerical linear algebra with applications*, 5(2):101–121, March/April 1998.

[3] Tijmen P. Collignon and Martin B. van Gijzen. Minimising synchronisation in IDR(s). *Numerical Linear Algebra with Applications*, 2010. (accepted for publication).

[4] Jocelyne Erhel, Kevin Burrage, and Bert Pohl. Restarted GMRES preconditioned by deflation. *Journal of Computational and Applied Mathematics*, 69(2):303–318, 1996.

[5] Yogi A. Erlangga and Reinhard Nabben. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(2):684–699, 2008.

[6] J. Frank and C. Vuik. On the construction of deflation–based preconditioners. *SIAM Journal on Scientific Computing*, 23(2):442–462, 2001.

[7] Martin H. Gutknecht. IDR Explained. *Electronic Transactions on Numerical Analysis*, 36:126–148, 2010.

[8] Martin H. Gutknecht and Jens-Peter M. Zemke. Eigenvalue computations based on IDR. Technical report, Seminar für Angewandte Mathematik, ETH Zürich, SAM, ETH Zürich, Switzerland, 2010. Research Report No. 2010–13.

[9] Ilse C. F. Ipsen and Carl D. Meyer. The idea behind Krylov methods. *American Mathematical Monthly*, 105:889–899, 1998.

[10] R. Nabben and C. Vuik. A comparison of deflation and coarse grid correction applied to porous media flow. *SIAM Journal on Numerical Analysis*, 42(4):1631–1647, 2004.

[11] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM Journal on Scientific Computing*, 27(5):1742–1759, 2006.

[12] Yusuke Onoue, Seiji Fujino, and Norimasa Nakashima. Improved IDR(s) method for gaining very accurate solutions. *World Academy of Science, Engineering and Technology*, 55:520–525, 2009.

[13] Yusuke Onoue, Seiji Fujino, and Norimasa Nakashima. An overview of a family of new iterative methods based on IDR theorem and its estimation. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong*, pages 2129–2134, 2009.

[14] Valeria Simoncini and Daniel B. Szyld. Interpreting IDR as a Petrov–Galerkin method. *SIAM Journal on Scientific Computing*, 32(4):1898–1912, 2010.

[15] G. L. G. Sleijpen and Henk A. van der Vorst. Maintaining convergence properties of BiCGstab methods in finite precision arithmetic. *Numerical Algorithms*, 10(3–4):203–223, 1995.

[16] Gerard L. G. Sleijpen, Peter Sonneveld, and Martin B. van Gijzen. Bi–CGSTAB as an induced dimension reduction method. *Applied Numerical Mathematics*, In Press, Corrected Proof, 2009.

[17] Gerard L. G. Sleijpen and Martin B. van Gijzen. The algebra for induced dimension reduction. In *The Proceedings of the 2nd Kyoto–Forum on Krylov Subspace Methods, Kyoto University, Kyoto, Japan*, March 2010.

[18] Gerard L. G. Sleijpen and Martin B. van Gijzen. Exploiting BiCGstab(ℓ) strategies to induce dimension reduction. *SIAM Journal on Scientific Computing*, 32(5):2687–2709, 2010.

[19] G.L.G. Sleijpen and D.R. Fokkema. BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Transactions on Numerical Analysis*, 1:11–32, 1993.

[20] Peter Sonneveld. On the convergence behaviour of IDR(s). Technical report, Delft University of Technology, Delft, the Netherlands, 2010. DUT report 10–08.

[21] Peter Sonneveld and Martin B. van Gijzen. IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 31(2):1035–1062, 2008.

[22] J. M. Tang, S. P. MacLachlan, R. Nabben, and C. Vuik. A comparison of two–level preconditioners based on multigrid and deflation. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1715–1739, 2010.

[23] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of Scientific Computing*, 39(3):340–370, 2009.

[24] Masaaki Tanio and Masaaki Sugihara. GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials. *Journal of Computational and Applied Mathematics*, 235(3):765–784, 2010.

[25] H. A. van der Vorst. Bi–CGSTAB: A fast and smoothly converging variant of Bi–CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[26] Martin B. van Gijzen and Tijmen P. Collignon. Exploiting the flexibility of IDR(s) for Grid computing. In *The Proceedings of the 2nd Kyoto–Forum on Krylov Subspace Methods, Kyoto University, Kyoto, Japan*, March 2010.

[27] Martin B. van Gijzen and Peter Sonneveld. An IDR(s) variant with minimal intermediate residual norms. In *The Proceedings of the International Kyoto–Forum on Krylov Subspace methods, Kyoto University, Kyoto, Japan*, pages 85–92, 2008.

[28] Martin B. van Gijzen and Peter Sonneveld. An elegant IDR(s) variant that efficiently exploits bi–orthogonality properties. Technical report, Delft University of Technology, Delft, the Netherlands, 2010. DUT report 10–16 (revised version of DUT report 08–21).