Journal of Computational and Applied Mathematics 345 (2019) 33–47

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Accelerating the Induced Dimension Reduction method using spectral information

R. Astudillo [a,*], J.M. de Gier [b], M.B. van Gijzen [a]

[a] Delft Institute of Applied Mathematics, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE, Delft, The Netherlands
[b] TNO Technical Sciences, Distributed Sensor Systems, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands

ARTICLE INFO

Article history:
Received 20 June 2017
Received in revised form 21 March 2018

Keywords:
Induced Dimension Reduction method
System of linear equations
Sequence of systems of linear equation
Eigenvalues and eigenvectors

ABSTRACT

The Induced Dimension Reduction method (IDR(s)) (Sonneveld and van Gijzen, 2008) is a short-recurrences Krylov method to solve systems of linear equations. In this work, we accelerate this method using spectral information. We construct a Hessenberg relation from the IDR(s) residual recurrences formulas, from which we approximate the eigenvalues and eigenvectors. Using the Ritz values, we propose a self-contained variant of the Ritz-IDR(s) method (Simoncini and Szyld, 2010) for solving a system of linear equations. In addition, the Ritz vectors are used to speed-up IDR(s) for the solution of sequence of systems of linear equations.

1. Introduction

In this work, we are interested in accelerating the convergence of the Induced Dimension Reduction method (IDR(s)) [1] to solve a system of linear equations

$$A\mathbf{x} = \mathbf{b}, \qquad \text{with } A \in \mathbb{C}^{n\times n} \text{ and } \mathbf{b} \in \mathbb{C}^n, \tag{1}$$

and also to solve sequences of systems of linear equations

$$A\mathbf{x}^{(i)} = \mathbf{b}^{(i)}, \qquad \text{with } A \in \mathbb{C}^{n\times n} \text{ and } \mathbf{b}^{(i)} \in \mathbb{C}^n, \text{ for } i = 1,\ 2,\ \ldots,\ p. \tag{2}$$

The vectors $\mathbf{x},\ \mathbf{x}^{(1)},\ \ldots,\ \mathbf{x}^{(p)}$ represent the unknowns in $\mathbb{C}^n$, and we only consider the case when the coefficient matrix A is a non-Hermitian and non-singular matrix.

IDR(s) is a Krylov subspace method which has been proved to be effective for solving large and sparse systems of linear equations. Both theoretical and practical aspects of the IDR(s) have been studied in different works, e.g., [2–5], and [6] among others. Simoncini and Szyld reformulate IDR(s) as a Petrov–Galerkin method in [2]. The authors prove that in IDR(s) the subspace of constraints or left space is a block rational Krylov subspace. Based on this connection with the rational subspaces, they propose to use the Ritz values to accelerate the convergence of IDR(s). This idea originates Ritz-IDR(s), which is an effective IDR(s) variant to solve systems of linear equations (1) where the spectrum is highly complex.

To obtain a subset of the Ritz values, Ritz-IDR(s) requires a preceding call to an external sparse eigensolver routine, for example the Arnoldi method [7] or Bi-Lanczos method [8]. In the first part of this paper, we present a self-contained version

* Corresponding author.
E-mail addresses: R.A.Astudillo@tudelft.nl (R. Astudillo), jan.degier@tno.nl (J.M. de Gier), M.B.vanGijzen@tudelft.nl (M.B. van Gijzen).

https://doi.org/10.1016/j.cam.2018.06.014
0377-0427/

of the Ritz-IDR(s), i. e., a Ritz-IDR(s) variant that does not require an external call to an eigensolver routine. We compute the upper Hessenberg matrix H_m from a Hessenberg relation as

$$AW_m = W_m H_m + \mathbf{f}\mathbf{e}_m^T, \tag{3}$$

during the first iterations of IDR(s). Then, we obtain the Ritz values from the matrix H_m, and use them as input parameters of the subsequent iterations of IDR(s).

In the second part of this paper, we apply IDR(s) to solve sequences of systems of linear equations (2). We only consider the case when the coefficient matrix does not change and the right-hand side vectors $\{\mathbf{b}^{(i)}\}_{i=1}^{p}$ are not available simultaneously. These kind of problems arises naturally from the discretization of linear time-dependent differential equations and the solution of systems of non-linear equations using modified Newton-type methods with constant Jacobian matrix.

Subspace recycling is a common technique to accelerate the Krylov method (see for example [9,10], and [11] among others). This process consists of approximating invariant subspaces or calculating a "good" Krylov subspace basis and use this information to save matrix–vector products at the solution of the system of linear equations. For methods as GMRES [12] and GCR [13] the recycling idea has been incorporated to accelerate the solution of a single linear system of equations in [9] and [10] respectively. In the case of solving sequences of systems of linear equations, these methods have been adapted in [14] and [15]. Also, other Krylov methods have been adapted to solve sequences of systems of linear equations, for example BiCG in [16], GMRES(m) in [15], and IDRstab in [17].

GCROT [14] and GMRES are long-recurrences methods, with an optimal residual minimization property, but also these methods can be expensive in terms of memory and CPU consumption. For this reason, we propose an IDR(s) variant, that is a short-recurrences and memory-limited method to solve (2). First, we show how to obtain Ritz values and Ritz vectors from IDR(s) for solving a system of linear equations. Second, we present how to enrich the searching subspace of IDR(s) with the Ritz vectors. Finally, we apply IDR(s) with the Ritz vectors to solve sequences of linear equations as a main application of this enrichment.

This document is organized as follows. A review of IDR(s) and its recurrences is presented in the second section. In Section 3, we present an IDR(s) variant to solve system of linear equations. We present how to obtain an underlying Hessenberg relation from the IDR(s) residual recurrences. This allows us to find approximation to the eigenvalues of the coefficient matrix involved. These eigenvalues approximations are used to accelerate the IDR(s) method. Section 3.1 shows the numerical examples related to the solution of system of linear equations. In section 4, we explain how to add the Ritz vectors to the initial searching space of IDR(s) to save computational effort. As a main application of this idea, we apply IDR(s) to solve a sequence of system of linear equations. Using the Hessenberg relation deduced in Section 3, we compute a set of Ritz vectors during the solution of the first system of linear equation. These Ritz vectors are used to accelerate the subsequent systems of linear equations. Numerical experiments for the solution of a sequence of systems of linear equations using IDR(s) are presented in Section 4.2. In Section 5, we present the general conclusions and remarks.

2. Review on IDR(s)

In this section, we first review the recurrence formulas of IDR(s) for solving a system of linear equations, and then we discuss the work of Simoncini and Szyld in [2].

The Induced Dimension Reduction method is based on the following theorem.

Theorem 1 (*IDR(s) Theorem*). *Let A be any matrix in $\mathbb{C}^{n\times n}$, let $\mathbf{v}_0$ be any nonzero vector in $\mathbb{C}^n$, and let $\mathcal{G}_0$ be the full Krylov subspace $\mathcal{K}_n(A, \mathbf{v}_0)$. Let $\mathcal{S}$ be any (proper) subspace of $\mathbb{C}^n$ such that $\mathcal{S}$ and $\mathcal{G}_0$ do not share a nontrivial invariant subspace of A, and define the sequence $\mathcal{G}_j$, $j = 1, 2, \ldots$ as*

$$\mathcal{G}_j \equiv (I - \omega_j A)(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where ω_j's are nonzero scalars. Then

1. *$\mathcal{G}_{j+1} \subset \mathcal{G}_j$, for $j \geq 0$ and*
2. *dimension$(\mathcal{G}_{j+1}) <$ dimension$(\mathcal{G}_j)$ unless $\mathcal{G}_j = \{\mathbf{0}\}$.*

Proof. See [1]. □

The main idea of the IDR(s) method is to create approximation vectors $\mathbf{x}_m$ such that their corresponding residual vectors $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$ belong to the nested and shrinking subspaces $\mathcal{G}_j$. IDR(s) creates $s + 1$ residual vectors in $\mathcal{G}_j$, and uses those vectors for the creation of the $s + 1$ subsequent residuals in $\mathcal{G}_{j+1}$. This process is repeated iteratively until convergence.

Our implementation of IDR(s) is based on IDR(s) with biorthogonal residuals (see [4]). In practice, this variant has proved to be more stable, and is also slightly less expensive. Next, we present the recurrences used by this IDR(s) variant. For sake of simplicity, we introduce new notation. The subspace $\mathcal{S}$ is represented by the left null space of some full-rank $n \times s$ matrix $P = [\mathbf{p}_1, \ldots \mathbf{p}_s]$ (called shadow space). The superindex of a vector or a scalar represents the number of subspace $\mathcal{G}_j$ where the current residual belongs. The subindex represents the position in the sequence of intermediate residuals. For $\mathbf{r}_k^{(j)}$ represents the kth residual in $\mathcal{G}_j$. The first residual vectors in $\mathcal{G}_{j+1}$ and its respective approximation are

$$\mathbf{x}_0^{(j+1)} = \mathbf{x}_s^{(j)} + \omega_{j+1}\mathbf{r}_s^{(j)},$$

and

$$\mathbf{r}_0^{(j+1)} = (I - \omega_{j+1}A)\mathbf{r}_s^{(j)}, \tag{4}$$

and the recurrences to create the intermediate residuals in $\mathcal{G}_{j+1}$, are

$$\mathbf{x}_k^{(j+1)} = \mathbf{x}_{k-1}^{(j+1)} + \beta_k^{(j+1)}\mathbf{u}_k^{(j+1)},$$

and

$$\mathbf{r}_k^{(j+1)} = \mathbf{r}_{k-1}^{(j+1)} - \beta_k^{(j+1)}\mathbf{g}_k^{(j+1)}, \quad \text{for } k = 1,\ 2,\ \ldots,\ s. \tag{5}$$

The scalar $\beta_k^{(j+1)}$ is selected such that

$$\langle \mathbf{r}_k^{(j+1)}, \mathbf{p}_k \rangle = 0. \tag{6}$$

The direction vectors are defined as

$$\mathbf{u}_k^{(j+1)} = \hat{\mathbf{u}}_k^{(j+1)} - \sum_{i=1}^{k-1} \alpha_i^{(j+1)}\mathbf{u}_i^{(j+1)}, \tag{7}$$

and

$$\mathbf{g}_k^{(j+1)} = \hat{\mathbf{g}}_k^{(j+1)} - \sum_{i=1}^{k-1} \alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)}, \tag{8}$$

where the vector $\hat{\mathbf{u}}_k^{(j+1)}$ and $\hat{\mathbf{g}}_k^{(j+1)}$ are

$$\hat{\mathbf{g}}_k^{(j+1)} = A\hat{\mathbf{u}}_k^{(j+1)}, \tag{9}$$

$$\hat{\mathbf{u}}_k^{(j+1)} = \omega_{j+1}\left(\mathbf{r}_{k-1}^{(j+1)} - \sum_{i=s-k}^{s} \gamma_i^{(j+1)}\mathbf{g}_i^{(j)}\right) + \sum_{i=s-k}^{s} \gamma_i^{(j+1)}\mathbf{u}_i^{(j)}. \tag{10}$$

The scalars $\{\alpha_i^{(j+1)}\}_{i=1}^{k-1}$ in (7) and (8) are selected, such that

$$\langle \mathbf{g}_k^{(j+1)}, \mathbf{p}_i \rangle = 0 \quad \text{for } i = 1, \ldots, k-1, \tag{11}$$

and the scalars $\{\gamma_i^{(j+1)}\}_{i=k}^{s}$ in (10) are selected as

$$\left\langle \mathbf{r}_{k-1}^{(j+1)} - \sum_{i=k}^{s} \gamma_i^{(j+1)}\mathbf{g}_i^{(j)}, \mathbf{p}_j \right\rangle = 0. \tag{12}$$

The conditions (6), (11), and (12) not only ensure that the residual $\mathbf{r}_k^{(j+1)}$ belongs to $\mathcal{G}_{j+1}$, but also, that the residual $\mathbf{r}_k^{(j+1)}$ is orthogonal to the vectors $\mathbf{p}_1,\ \mathbf{p}_2,\ \ldots,\ \mathbf{p}_k$ for $k = 1,\ 2,\ \ldots,\ s$.

An important property needed for the deduction of the IDR(s)-Hessenberg relation to be presented in Section 3, is that for any IDR(s) variant a residual in $\mathcal{G}_j$ can be also written as

$$\mathbf{r}_k^{(j)} = \Omega_j(A)\Psi(A)_{s\times j+k}\mathbf{r}_0, \tag{13}$$

where

$$\Omega_j(t) = \prod_{i=0}^{j}(1 - \omega_i t), \quad \omega_i \neq 0, \quad i = 1, \ldots, j, \tag{14}$$

$\Omega_0(t) = 1$, and $\Psi_m(t)$ is a multi-Lanczos-type polynomial [18] of order m, that uses $s+2$ terms recurrences such that $\Psi_0 = 1$ (see Section 5 in [1]). When the first residual vector is created in $\mathcal{G}_{j+1}$, the polynomial $\Omega_j(A)$ increases by one degree. Then, the degree of the polynomial $\Psi_m(A)$ is increased by one for each matrix–vector product during the creation of the others intermediate residuals.

2.1. IDR(s) as a Petrov–Galerkin method and Ritz-IDR(s)

As we mention in the introduction of this work, Simoncini and Szyld showed that IDR(s) can be viewed as a Petrov–Galerkin method in [2]. Particularly IDR(s) finds the approximation $\mathbf{x}_{k+1}$ in the right or searching subspace $\mathbf{x}_0 + \mathcal{K}_{k+1}(A, \mathbf{r}_0)$, by imposing the condition that $\mathbf{r}_{k+1}$ is orthogonal to the subspace $\mathcal{W}_j$, defined as

$$\mathcal{W}_j = \Omega_j(A^H)^{-1}\mathcal{K}_j(A^H, P), \tag{15}$$

where $\Omega_j(A)$ is the polynomial defined in (14), and $\mathcal{K}_j(A^H, P)$ is the block Krylov subspace of order j, associated with the matrix A and the block P.

This link between IDR(s) and the rational block subspaces leads to the development of the variant Ritz-IDR(s). The authors in [2] argue that selecting the scalars ω_j as the inverse of Ritz values of the coefficient matrix A, is a good choice for the creation of the left space $\mathcal{W}_j$. This selection enriches the left subspace with information about the associated eigencomponents. This would damp the eigenvector components from the residual vector in a quick way, which leads to a faster convergence. The Ritz values required are computed with a call to an eigensolver routine as the Arnoldi method. Note that Ritz-IDR(s) might require complex arithmetics even when the coefficient matrix and right-hand size vector are real, in the case when complex Ritz values are encountered.

In the following section we present how to obtain an Hessenberg relation from the IDR(s) recurrences. Using this Hessenberg relation, we can obtain approximations to the eigenvalues of the coefficient matrix, and in this form we obtain a self-contained variant of the Ritz-IDR(s). To distinguish it, we label our proposed algorithm as SC-Ritz-IDR(s).

3. Part 1: Accelerating IDR(s) using the Ritz values

IDR(s) has been previously used to obtain spectral information of a matrix. In [19], the authors adapt IDR(s) to solve the eigenvalue problem, and they obtain the matrices $\hat{H}_m$ and T_m from a generalized Hessenberg relation

$$AW_mT_m = W_m\hat{H}_m + \hat{\mathbf{f}}\mathbf{e}_m^T.$$

where $W_m \in \mathbb{C}^{n\times m}$ (not explicitly available) represents a Krylov subspace basis for $\mathcal{K}(A, \mathbf{w}_1)$, T_m is an s-banded, upper triangular matrix; $\hat{H}$ is an s-banded, upper Hessenberg matrix, and $\hat{\mathbf{f}} \in \mathbb{C}^n$. The approximation of the eigenvalues of A are obtained from the eigenvalue pencil $(\hat{H}_m,\ T_m)$. In [20], the authors create a standard Hessenberg relation

$$AW_m = W_mH_m + \mathbf{f}\mathbf{e}_m^T, \tag{16}$$

where $W_m \in \mathbb{C}^{n\times m}$, and H_m is a Hessenberg matrix. This matrix H_m has the same eigenvalues as the matrix pencil $(\hat{H}_m,\ T_m)$.

The mentioned works [19] and [20] target specifically the eigenvalue/eigenvector approximation problem. Next, we describe how to obtain a matrix H_m part of a standard Hessenberg relation (16) from the underlying IDR(s)-recurrences used to solve systems of linear equation. This allows us to obtain the solution of a system of linear equation, and at the same time obtain approximations to the eigenvalues of the coefficient matrix. Particularly, we use this spectral information as is suggested in [2], and we proposed a Ritz-IDR(s) variant labeled as SC-Ritz-IDR(s).

To derive this Hessenberg matrix, let us consider the IDR(s) relations described in Section 2. Substituting (8)–(10) in (5), we obtain

$$\begin{aligned}
\frac{\mathbf{r}_{k-1}^{(j+1)} - \mathbf{r}_k^{(j+1)}}{\beta_k^{(j+1)}} &= \hat{\mathbf{g}}_k^{(j+1)} - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= A\hat{\mathbf{u}}_k^{(j+1)} - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= A\left[\omega_{j+1}\left(\mathbf{r}_{k-1}^{(j+1)} - \sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{g}_i^{(j)}\right) + \sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{u}_i^{(j)}\right] - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= \omega_{j+1}A\mathbf{r}_{k-1}^{(j+1)} - \omega_{j+1}A\sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{g}_i^{(j)} + \sum_{i=s-k}^{s}\gamma_i^{(j+1)}A\mathbf{u}_i^{(j)} - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= \omega_{j+1}A\mathbf{r}_{k-1}^{(j+1)} - \omega_{j+1}A\sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{g}_i^{(j)} + \sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{g}_i^{(j)} - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= \omega_{j+1}A\mathbf{r}_{k-1}^{(j+1)} + (I - \omega_{j+1}A)\sum_{i=s-k}^{s}\gamma_i^{(j+1)}\mathbf{g}_i^{(j)} - \sum_{i=1}^{k-1}\alpha_i^{(j+1)}\mathbf{g}_i^{(j+1)} \\
&= \omega_{j+1}A\mathbf{r}_{k-1}^{(j+1)} + (I - \omega_{j+1}A)\sum_{i=s-k}^{s}\frac{\gamma_i^{(j+1)}}{\beta_i^{(j)}}(\mathbf{r}_{i-1}^{(j)} - \mathbf{r}_i^{(j)}) - \sum_{i=1}^{k-1}\frac{\alpha_i^{(j+1)}}{\beta_i^{(j+1)}}(\mathbf{r}_{i-1}^{(j+1)} - \mathbf{r}_i^{(j+1)}).
\end{aligned}$$

From the equations above, we obtain the following relation

$$\begin{aligned}\omega_{j+1}A\mathbf{r}_{k-1}^{(j+1)} &= \frac{\mathbf{r}_{k-1}^{(j+1)}-\mathbf{r}_{k}^{(j+1)}}{\beta_k^{(j+1)}} - (I-\omega_{j+1}A)\sum_{i=s-k}^{s}\frac{\gamma_i^{(j+1)}}{\beta_i^{(j)}}(\mathbf{r}_{i-1}^{(j)}-\mathbf{r}_i^{(j)}) \\ &+ \sum_{i=1}^{k-1}\frac{\alpha_i^{(j+1)}}{\beta_i^{(j+1)}}(\mathbf{r}_{i-1}^{(j+1)}-\mathbf{r}_i^{(j+1)}).\end{aligned} \tag{17}$$

Using (13), we obtain that each vector in $\mathcal{G}_j$ can be written as

$$\mathbf{r}_i^{(j)} = \Omega_j(A)\hat{\mathbf{r}}_i^{(j)} \qquad \text{for } i=0,\ \ldots,s, \tag{18}$$

and equivalently, any residuals in $\mathcal{G}_{j+1}$ can be written as

$$\mathbf{r}_i^{(j+1)} = \Omega_{j+1}(A)\hat{\mathbf{r}}_i^{(j+1)} \qquad \text{for } i=0,\ \ldots,s. \tag{19}$$

Taking into account (19) and (18), we can multiply (17) by $\Omega_{j+1}(A)^{-1}$ and obtain

$$\begin{aligned}\omega_{j+1}A\hat{\mathbf{r}}_{k-1}^{(j+1)} &= \frac{\hat{\mathbf{r}}_{k-1}^{(j+1)}-\hat{\mathbf{r}}_{k}^{(j+1)}}{\beta_k^{(j+1)}} - \sum_{i=s-k}^{s}\frac{\gamma_i^{(j+1)}}{\beta_i^{(j)}}(\hat{\mathbf{r}}_{i-1}^{(j)}-\hat{\mathbf{r}}_i^{(j)}) \\ &+ \sum_{i=1}^{k-1}\frac{\alpha_i^{(j+1)}}{\beta_i^{(j+1)}}(\hat{\mathbf{r}}_{i-1}^{(j+1)}-\hat{\mathbf{r}}_i^{(j+1)}).\end{aligned} \tag{20}$$

The set of vectors $\hat{\mathbf{r}}_i$ represents the Krylov basis associated with the polynomial $\Psi(A)$. In fact, one can see that the basis grows with the degree of the polynomial $\Psi(A)$. Substituting (18) and (19) in (4), we obtain that

$$\hat{\mathbf{r}}_0^{(j+1)} = \hat{\mathbf{r}}_s^{(j)}. \tag{21}$$

This implies that every $s+1$ matrix–vector products, IDR(s) creates s new vectors basis $\hat{\mathbf{r}}_i$. Using (21), we can rewrite (20) as

$$\begin{aligned}\omega_{j+1}A\hat{\mathbf{r}}_{k-1}^{(j+1)} &= -\frac{\gamma_{s-k}^{(j+1)}}{\beta_{s-k}^{(j+1)}}\hat{\mathbf{r}}_{s-k-1}^{(j)} - \sum_{i=s-k}^{s-1}\left(\frac{\gamma_{i+1}^{(j+1)}}{\beta_{i+1}^{(j)}}-\frac{\gamma_i^{(j+1)}}{\beta_i^{(j)}}\right)\hat{\mathbf{r}}_i^{(j)} \\ &+ \left(\frac{\gamma_s^{(j+1)}}{\beta_s^{(j+1)}}+\frac{\alpha_1^{(j+1)}}{\beta_1^{(j+1)}}\right)\hat{\mathbf{r}}_s^{(j)} + \sum_{i=1}^{k-1}\left(\frac{\alpha_{i+1}^{(j+1)}}{\beta_{i+1}^{(j+1)}}-\frac{\alpha_i^{(j+1)}}{\beta_i^{(j+1)}}\right)\hat{\mathbf{r}}_i^{(j+1)} \\ &- \frac{1}{\beta_k^{(j+1)}}\hat{\mathbf{r}}_k^{(j+1)}.\end{aligned} \tag{22}$$

One can see in (22) that the vector $A\hat{\mathbf{r}}_{k-1}^{(j+1)}$ is a linear combination of the vectors $\{\hat{\mathbf{r}}_i^{(j)}\}_{i=s-k-1}^{s}$ and $\{\hat{\mathbf{r}}_i^{(j+1)}\}_{i=1}^{k}$. This defines a Hessenberg relation of the form

$$A\hat{R}_{\bar{m}} = \hat{R}_{\bar{m}+1}\bar{H}_{\bar{m}}, \tag{23}$$

where $\bar{m}$ is the number of intermediate residuals created by IDR(s), and $\hat{R}_{\bar{m}}$ is a Krylov subspace basis defined as

$$\hat{R}_{\bar{m}} = [\hat{\mathbf{r}}_0^{(0)},\ldots,\hat{\mathbf{r}}_s^{(0)},\hat{\mathbf{r}}_1^{(1)},\ldots,\hat{\mathbf{r}}_s^{(1)},\ldots,\hat{\mathbf{r}}_1^{(j)},\ldots,\hat{\mathbf{r}}_s^{(j)},\hat{\mathbf{r}}_1^{(j+1)},\ldots,\hat{\mathbf{r}}_k^{(j+1)}]_{n\times\bar{m}}. \tag{24}$$

The vectors $\hat{\mathbf{r}}_i$ are not constructed explicitly, however, it is easy to see that

$$\hat{\mathbf{r}}_0^{(0)} = \mathbf{r}_0. \tag{25}$$

The matrix $H_{\bar{m}}$ is an upper and $s+1$ banded Hessenberg matrix whose columns are defined as

$$\mathbf{H}_\ell = \begin{bmatrix} 0 \\ \vdots \\ h_{\ell-s,\ell} \\ \vdots \\ h_{\ell+1,\ell} \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{C}^{\bar{m}+1},$$

where

$$\begin{bmatrix} h_{\ell-s,\ell} \\ h_{\ell-s+1,\ell} \\ \vdots \\ h_{\ell-s+m,\ell} \\ h_{\ell-s+m+1,\ell} \\ \vdots \\ h_{\ell+1,\ell} \end{bmatrix} = \frac{1}{\omega_{j+1}} \begin{bmatrix} -\dfrac{\gamma_{s-k}^{(j+1)}}{\beta_{s-k}^{(j+1)}} \\ \dfrac{\gamma_{s-k+1}^{(j+1)}}{\beta_{s-k+1}^{(j)}} - \dfrac{\gamma_{s-k}^{(j+1)}}{\beta_{s-k}^{(j)}} \\ \vdots \\ \dfrac{\gamma_{s}^{(j+1)}}{\beta_{s}^{(j)}} + \dfrac{\alpha_{1}^{(j+1)}}{\beta_{1}^{(j+1)}} \\ \dfrac{\alpha_{2}^{(j+1)}}{\beta_{2}^{(j+1)}} - \dfrac{\alpha_{1}^{(j+1)}}{\beta_{1}^{(j+1)}} \\ \vdots \\ -1/\beta_{k}^{(j+1)} \end{bmatrix} \in \mathbb{C}^{s+2},$$

Our implementation of SC-Ritz-IDR(s) is based on the IDR(s) with biorthogonal residuals. The memory consumption of SC-Ritz-IDR(s) is similar to that of IDR(s) (see Section 3.5 in [4]). The sets of coefficients $\{\alpha_i\}_{i=1}^{s}$, $\{\gamma_i\}_{i=1}^{s}$, and $\{\beta_i\}_{i=1}^{s}$, used in SC-Ritz-IDR(s), are stored in three extra vectors of size s. Algorithm 1 shows an implementation of SC-Ritz-IDR(s).

3.1. Numerical experiments

To illustrate the numerical behavior of the proposed algorithm, we repeat all the experiments presented in [2]. We compare our proposed variant SC-Ritz-IDR(s) with IDR(s), Ritz-IDR(s) and full GMRES. All the experiments are performed in Matlab 2015a running on a 64 bit GNU/Debian Linux computer with 32 GB of RAM. The right-hand side vector $\mathbf{b} = \hat{\mathbf{b}}/\|\hat{\mathbf{b}}\|$ with $\hat{\mathbf{b}} = \mathbf{1}$, and the initial vector is $\mathbf{x}_0 = \mathbf{0}$. As stopping criterion, we use

$$\frac{\|\mathbf{b} - A\mathbf{x}_k\|}{\|\mathbf{b}\|} < \epsilon,$$

with $\epsilon = 10^{-10}$.

For Ritz-IDR(s) and SC-Ritz-IDR(s), we use as parameter

$$\omega_j = \frac{1}{\lambda_i}, \tag{26}$$

where λ_i is an eigenvalue of the matrix $H_{\bar{m}}$. We select $\bar{m} = 20$ and the 15 smallest magnitude eigenvalues. For Ritz-IDR(s), the matrix $H_{\bar{m}}$ is obtained with a preliminary call to the Arnoldi method. In the case of SC-Ritz-IDR(s) the matrix $H_{\bar{m}}$ is computed as is explained in Section 3. Before the creation of the matrix $H_{\bar{m}}$, SC-Ritz-IDR(s) uses the converge maintenance strategy, proposed in [4], to select the first ω_j parameters.

3.1.1. Convection–diffusion–reaction equation examples

The linear systems of equations used in the next three examples are based on the finite difference discretization of the simple convection–diffusion–reaction model problem

$$-\epsilon \triangle u + \mathbf{v}^T \nabla u + \rho u = f, \qquad \text{in } \Omega = [0,\ 1]^d \tag{27}$$

with $d = 2$ or $d = 3$, and homogeneous Dirichlet boundary conditions on $\partial\Omega$. Particularly, it is known that IDR(s) with $s > 1$ outperforms BiCGStab [21] when the $\|\mathbf{v}\| \gg \epsilon$ (see for example [22] and [23]).

Example 1. In this example the coefficient matrix A is given by the finite difference discretization of (27) for the 2D case. The physical parameters used are $\epsilon = 1, \mathbf{v} = [4,\ 0]^T$, and $\rho = 400$. We discretize the domain Ω using 21 points in each direction. Fig. 1(a) shows the convergence of the norm of the residual for the matrix A of order 400 generated with the parameters described. Ritz-IDR(s) and SC-Ritz-IDR(s) do not show any improvement over IDR(s). However, using a convection-dominated taking 41 points in each direction and $\epsilon = 1, \mathbf{v} = [80,\ 0]^T$, and $\rho = 1600$, we can see in Fig. 1(b) a better performance of Ritz-IDR(s) and SC-Ritz-IDR(s) over IDR(s).

Example 2. We consider two matrices of order 8000 from the discretization of the 3D problem (27) with $\epsilon = 1$, $\mathbf{v} = \beta[1,\ 1,\ 1]$, and $\rho = 0$. First using $\beta = 100$, we can see in Fig. 2(a) a similar behavior between the IDR(s) variants.

Algorithm 1 IDR(s) accelerated with Ritz values

procedure IDR(A, $\mathbf{b}$, s, tol, $\mathbf{x}_0$)
  Input: $A \in \mathbb{C}^{n\times n}$, $\mathbf{b} \in \mathbb{C}^n$, $s \in \mathbb{N}^+$, $tol \in (0,\ 1)$, $\mathbf{x}_0 \in \mathbb{C}^n$.
  $\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - A\mathbf{x}$
  P a random matrix in $\mathbb{C}^{n\times s}$.
  $G = 0 \in \mathbb{C}^{n\times s}$, $U = 0 \in \mathbb{C}^{n\times s}$
  $M = I_s \in \mathbb{C}^{s\times s}$.
  $\omega = 1.0$, $\ell = 0$, $H_{\bar{m}} = 0 \in \mathbb{C}^{\bar{m}+1\times\bar{m}}$, $\mathbf{c} = \mathbf{0}$, $\alpha = \mathbf{0}$, $\beta = \mathbf{0} \in \mathbb{C}^s$.
  while $\|\mathbf{r}\| \le tol \times \|\mathbf{b}\|$ **do** ▷ Loop over $\mathcal{G}_j$ spaces
    $\mathbf{f} = P^H\mathbf{r}$
    for $k = 1$ to s **do** ▷ Compute s independent vectors $\mathbf{g}_k$ in $\mathcal{G}_j$ space
      Solve $\mathbf{c}$ from $M\mathbf{c} = \mathbf{f}$, $(\gamma_1, \ldots, \gamma_s)^H = \mathbf{c}$ ▷ Note that $M = P^HG$
      $\mathbf{v} = \mathbf{r} - \sum_{i=k}^{s} \gamma_i\mathbf{g}_i$
      $\mathbf{v} = B^{-1}\mathbf{v}$ ▷ Preconditioning operation
      $\mathbf{u}_k = \omega\mathbf{v} + \sum_{i=k}^{s} \gamma_i\mathbf{g}_i$
      $\mathbf{g}_k = A\mathbf{u}_k$
      for $i = 1$ to $k-1$ **do** ▷ Make $\mathbf{g}_k$ orthogonal to P
        $\alpha_i = \langle \mathbf{g}_k, \mathbf{p}_i\rangle / \mu_{i,i}$
        $\mathbf{g}_k = \mathbf{g}_k - \alpha_i\mathbf{g}_i$
        $\mathbf{u}_k = \mathbf{u}_k - \alpha_i\mathbf{u}_i$
      end for
      $\mu_{i,k} = \langle \mathbf{g}_k, \mathbf{p}_i\rangle$, $M_{i,k} = \mu_{i,k}$, for $i = k, \ldots, s$ ▷ Update M
      $\beta_k = \phi_k / \mu_{k,k}$ ▷ Now $\langle \mathbf{r}, \mathbf{p}_i\rangle = 0$ for $i = 1, \ldots, k$
      $\mathbf{r} = \mathbf{r} - \beta_k\mathbf{g}_k$
      $\mathbf{x} = \mathbf{x} + \beta_k\mathbf{u}_k$
      if $k + 1 \le s$ **then**
        $\mathbf{f}_i = 0$ for $i = 1, \ldots, k$
        $\mathbf{f}_i = \mathbf{f}_i - \beta_k M_{i,k}$ for $i = k+1, \ldots, s$
      end if
      $\ell = \ell + 1$
      if $\ell \le \bar{m}$ **then**
        $H_{\ell-s:\ell-k,\ell} = \mathbf{c}_{k:s} / \beta_{k:s}$
        $H_{\ell-k+1:\ell-1,\ell} = \alpha_{1:k-1} / \beta_{1:k-1}$
        $H_{\ell,\ell} = 1.0 / \beta_k$
        $H_{\ell-s+1:\ell+1,\ell} = H_{\ell-s+1:\ell+1,\ell} + H_{\ell-s:\ell,\ell}$
        $H_{\ell-s:\ell+1,\ell} = H_{\ell-s:\ell+1,\ell} / \omega$
      end if
      Overwrite kth columns of G and U by $\mathbf{g}_k$ and $\mathbf{u}_k$ respectively.
    end for ▷ Entering $\mathcal{G}_{j+1}$
    $\mathbf{v} = B^{-1}\mathbf{r}$ ▷ Preconditioning operation
    $\mathbf{t} = A\mathbf{v}$
    if $\ell \le \bar{m}$ **then** ▷ Select new ω
      ω is selected using the converge maintenance strategy [4].
    else
      ω is selected using the spectral information provided by $H_{\bar{m}}$.
    end if
    $\mathbf{r} = \mathbf{r} - \omega\mathbf{t}$
    $\mathbf{x} = \mathbf{x} + \omega\mathbf{v}$
  end while
  Return $\mathbf{x}$ and $H_{\bar{m}}$ (if required).
end procedure

However, Ritz-IDR(s) and SC-Ritz-IDR(s) are clearly superior with respect to the IDR(s) when the parameter β is increased to 500 (see Fig. 2(b)).

Example 3. The coefficient matrix used in this example is the unsymmetric matrix of order 8000 that comes from the finite difference discretization of the 3D (27), with parameter $\epsilon = 1$, $\rho = 0$, and $\mathbf{v} = [0,\ 0,\ 1000]^T$. As in part (b) of previous

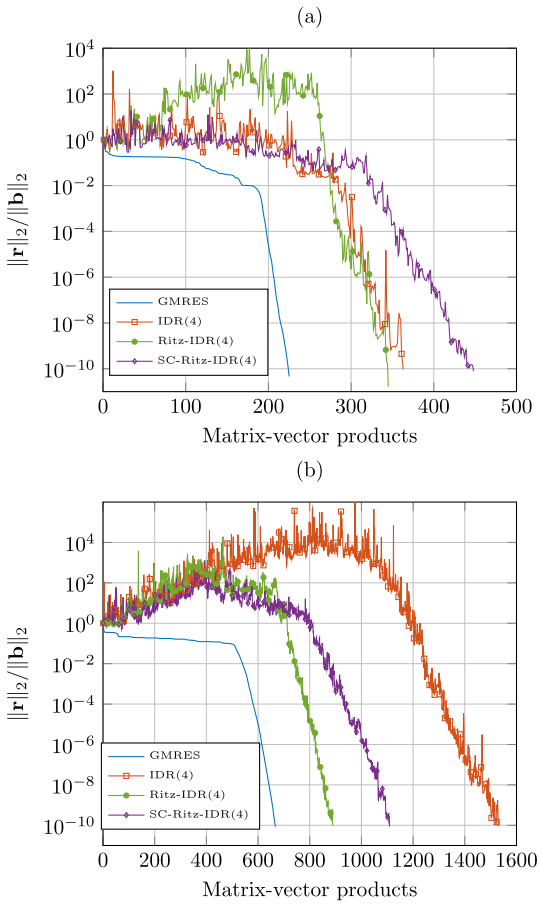

Fig. 1. (Example 1) Evolution of the residual norm of full GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4). (a) Diffusion-dominated example. (b) Convection-dominated example.

example, IDR(4) does not converge for the maximum number of iterations allowed, while Ritz-IDR(4) and SC-Ritz-IDR(4) converge using almost the same number of matrix–vector products (see Fig. 3).

3.1.2. Examples from matrix market

The matrices used in the next two examples are part of the Matrix Market collection [24].

Example 4. We consider the highly indefinite matrix Sherman5 of order 3312. As is reported in [2], Ritz-IDR(s) diverges for this example. SC-Ritz-IDR(s) exhibits a similar behavior. On the other hand, Fig. 4 shows that both Ritz-IDR(s) variants converge using the Incomplete LU factorization of the matrix $A + I$ as preconditioner with threshold tolerate 10^{-2}. In this example, IDR(s) and its variant behave similarly in term of matrix–vector products required.

Example 5. In this example, we consider the linear system of equations ADD20 which arises from computer component design. In this example, we stop the algorithms when the relative residual norm is less than 1×10^{-8}. As is proposed in [2], we also consider 20 Leja points located in the interval where the 20 real Ritz values are located. The Leja points are computed using the algorithm proposed in [25]. Fig. 5 shows a similar behavior between all the IDR(s) variants.

We include Table 1 where we compare the execution times required by the different methods for the solution of each numerical example. The numerical results show a similar behavior between Ritz-IDR(s) and SC-Ritz-IDR(s) in terms of matrix–vector products and convergence. Moreover, their computational requirements are virtually the same. The only difference is that SC-Ritz-IDR(s) requires storing a Hessenberg matrix of size $\bar{m}$. The main advantage of SC-Ritz-IDR(s) is that it computes the Ritz-values "on-the-fly". Therefore, unlike in Ritz-IDR(s), the time overhead of a call to an external eigensolver is avoided in SC-Ritz-IDR(s).

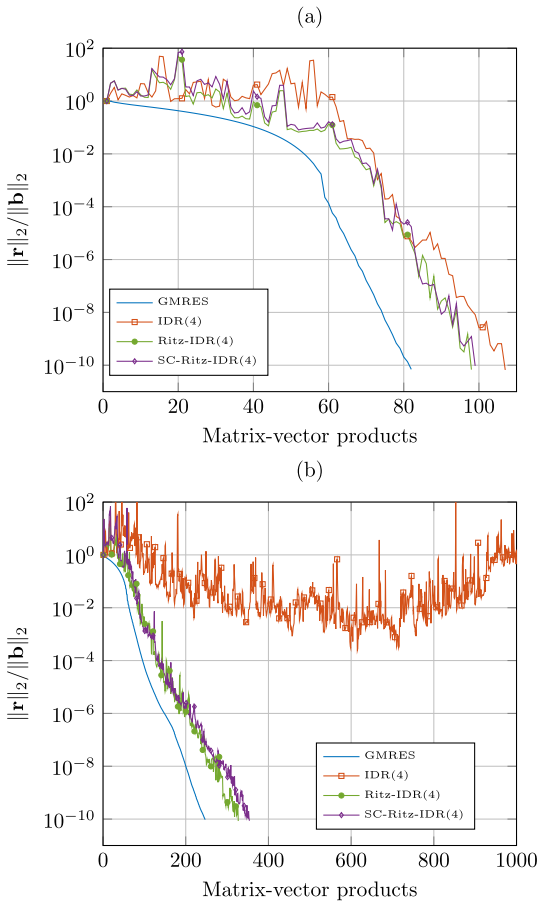

Fig. 2. (Example 2) Evolution of the residual norm of full GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4). (a) Diffusion-dominated example. (b) Convection-dominated example.

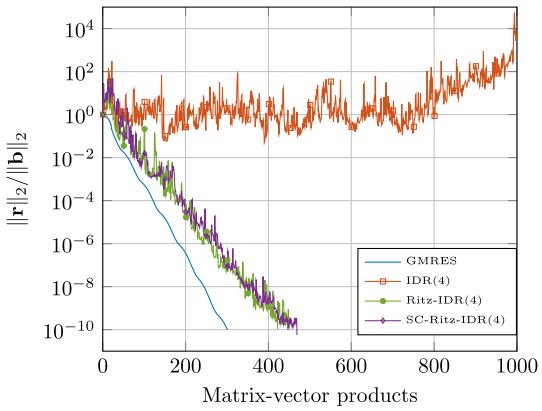

Fig. 3. (Example 3) Evolution of the residual norm of full GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4).

4. Part 2: Accelerating IDR(s) using Ritz vectors

In the previous sections we use the recurrences of IDR(s) to obtain an upper Hessenberg matrix $H_{\tilde{m}}$. From this matrix $H_{\tilde{m}}$, we obtain the Ritz values to accelerate the IDR(s) method. In this section, we incorporate the Ritz vectors to the Krylov basis generated by IDR(s). First, we present how to add additional vectors to the IDR(s) searching subspace basis, i.e., the

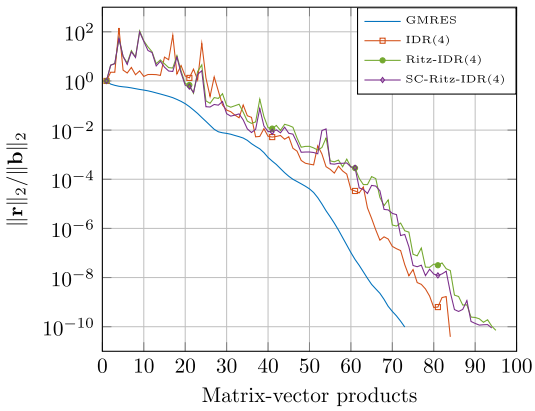

Fig. 4. (Example 4) Evolution of the residual norm of full GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4) for the matrix Sherman5 using ILU preconditioner.

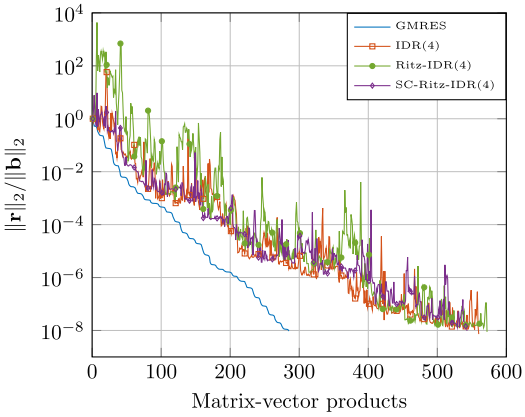

Fig. 5. (Example 5) Evolution of the residual norm of full GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4) for the matrix ADD20.

Table 1
CPU time consumed for the methods GMRES, IDR(4), Ritz-IDR(4), and SC-Ritz-IDR(4) for the solution of systems of linear systems.

Execution times [s]				
Experiment	Method			
	Full GMRES	IDR(4)	Ritz-IDR(4)	SC-Ritz-IDR(4)
Experiment 1(a)	0.435	0.043	0.087	0.085
Experiment 1(b)	4.784	0.259	0.367	0.392
Experiment 2(a)	0.382	0.051	0.124	0.086
Experiment 2(b)	3.251	0.561	0.511	0.361
Experiment 3	4.964	[a]	0.608	0.396
Experiment 4	0.207	0.0252	0.073	0.046
Experiment 5	1.086	0.156	0.198	0.174

[a] Indicates that the method diverges. The recorded time for Ritz-IDR(s) includes the call of $\bar{m}$ steps of the Arnoldi method.

augmented Krylov subspace

$$\mathcal{K}_{s+m}(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0,\ \mathbf{y}_1,\ \ldots, \mathbf{y}_s,\ A\mathbf{r}_0,\ \ldots, A^{m-1}\mathbf{r}_0\}. \tag{28}$$

Secondly, we use the matrix $H_{\bar{m}}$ to recover the Ritz vectors of the coefficient matrix, and add these Ritz vectors in IDR(s).

To add additional direction vectors to the Krylov basis created by IDR(s), we exploit the fact that $\mathcal{G}_0$ is $\mathbb{C}^n$. We can choose freely the first $s+1$ linearly independent direction vectors in IDR(s) and obtain their corresponding approximations and associated residuals. In the case of the biorthogonal variant, we have to ensure that each residual $\mathbf{r_i}$ is orthogonal to $\mathbf{p}_j$ for

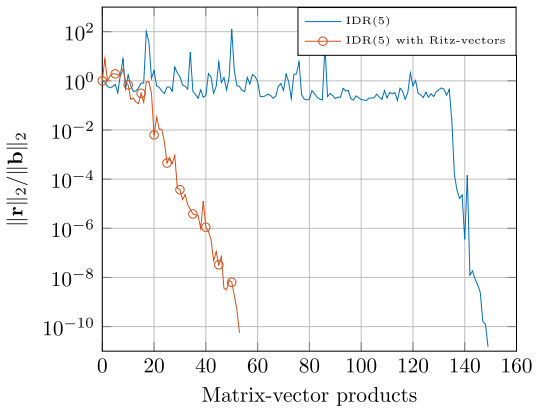

Fig. 6. (Example 6) Evolution of the residual norm of full IDR(5) and IDR(5) with recycling with the four eigenvectors associated with the smallest magnitude eigenvalues of the matrix (29).

$i = 1, 2, \ldots, s$ and $j = 1, 2, \ldots, i$, and each vector $\mathbf{g}_i$ is orthogonal to $\mathbf{p}_j$ for $i = 1, 2, \ldots, s$ and $j = 1, 2, \ldots, i-1$. In order to do so, we present the Algorithm 2, to create the first s biorthogonal residuals.

Algorithm 2 Injecting basis vectors in $\mathcal{G}_0$

1: **Input:** $\{\mathbf{y}_i\}_{i=1}^{s}$
2: **for** $k = 1$ to s **do**
3: $\quad \mathbf{u}_k = \mathbf{y}_k$
4: $\quad \mathbf{g}_k = A\mathbf{u}_k$
5: $\quad$ **for** $i = 1$ to $k-1$ **do** ▷ Make $\mathbf{g}_k$ orthogonal to P
6: $\quad\quad \alpha = \langle \mathbf{g}_k, \mathbf{p}_i \rangle / \mu_{i,i}$
7: $\quad\quad \mathbf{g}_k = \mathbf{g}_k - \alpha \mathbf{g}_i$
8: $\quad\quad \mathbf{u}_k = \mathbf{u}_k - \alpha \mathbf{u}_i$
9: $\quad$ **end for**
10: $\quad \mu_{i,k} = \langle \mathbf{g}_k, \mathbf{p}_i \rangle$, $M_{i,k} = \mu_{i,k}$, for $i = k, \ldots, s$ ▷ Update M
11: $\quad \beta = \phi_k / \mu_{k,k}$ ▷ Make the residual orthogonal to $\mathbf{p}_i$ for $i = 1, \ldots, k$
12: $\quad \mathbf{r} = \mathbf{r} - \beta \mathbf{g}_k$
13: $\quad \mathbf{x} = \mathbf{x} + \beta \mathbf{u}_k$
14: $\quad \phi_i = 0$ for $i = 1, \ldots, k$
15: $\quad \phi_i = \phi_i - \beta \mu_{i,k}$ for $i = k+1, \ldots, s$
16: **end for** ▷ Entering $\mathcal{G}_{j+1}$

To add the vectors $\{\mathbf{y}_i\}_{i=1}^{s}$ to the IDR(s), we should replace Algorithm 2 by the lines 5 and 6 in Algorithm 1. As is proposed in [9,10], and [26], we use as extra basis vectors the Ritz vector associated with the smallest-magnitude Ritz values.

Example 6 (*A Motivating Example*)**.** To exemplify the idea of using the spectral information in the initial subspace $\mathcal{G}_0$, we consider solving a system of linear equations with the following bidiagonal matrix

$$A = \begin{bmatrix} 1\times 10^{-8} & 1\times 10^{-5} & & & & & & & \\ & 2\times 10^{-8} & 1\times 10^{-5} & & & & & & \\ & & \ddots & & & & & & \\ & & & 5\times 10^{-8} & 1\times 10^{-5} & & & & \\ & & & & 6 & 1\times 10^{-5} & & & \\ & & & & & \ddots & & & \\ & & & & & & 100 & & \end{bmatrix}_{100\times 100}, \tag{29}$$

and the right-hand side vector is $\mathbf{b} = \mathbf{1}$. We compare IDR(5) and IDR(5) with recycling. As recycling vectors, we use the five eigenvectors associated with the smallest magnitude eigenvalues of the bidiagonal matrix A. The initial guess vector is $\mathbf{x}_0 = \mathbf{0}$. Fig. 6 shows the evolution of the norms of the residuals, one can see a considerable reduction in the number of matrix–vector products for IDR(s) with recycling.

It is worth mentioning the recently proposed M(s)STAB(ℓ) method by Neuenhofen [17]. M(s)STAB(ℓ) is a variant of the IDRstab [22], that is specialized to solve sequences of systems of linear equations where the coefficient matrix is constant. Based on a generalization of the IDR(s) Theorem, M(s)STAB(ℓ) uses as initial vectors basis in $\mathcal{G}_0$ the last $s+1$ vectors in the subspace $\mathcal{G}_j$, which were created during the solution of the previous system of linear equation. In this form, M(s)STAB(ℓ) reduces the computation and accelerates the solution of (2).

4.1. Adding the Ritz vectors to IDR(s): application to sequence of system of linear equations

Here we present the main application of IDR(s) with recycling, the solution of a sequence of systems of linear equations. We consider the case where the coefficient matrix A is constant, and the right-hand side vectors $\{\mathbf{b}^{(i)}\}_{i=1}^{p}$ are not available simultaneously.

The main idea is to compute a subset of Ritz vectors of the matrix A during the solution of the first system of linear equation, and then use these Ritz vectors to accelerate the solution of the subsequent systems of linear equations. The upper Hessenberg matrix $H_{\tilde{m}} \in \mathbb{C}^{\tilde{m}\times\tilde{m}}$ is computed using Algorithm 1. To compute the Ritz vectors after the first execution of IDR(s), we need to compute the Krylov basis $\hat{R}$ in (23). To compute this $\hat{R}$, we use (13) and obtain that,

$$\hat{\mathbf{r}}_0 = \mathbf{r}_0, \tag{30}$$

and taking into account the upper Hessenberg structure of the matrix $H_{\tilde{m}}$, we obtain the following recurrence formula for the vector $\hat{\mathbf{r}}_i$

$$\hat{\mathbf{r}}_i = \frac{1}{h_{i+1,i}}\left[A\hat{\mathbf{r}}_{i-1} - \sum_{j=\max(0,i-s)}^{i-1} h_{j,i}\hat{\mathbf{r}}_j\right]. \tag{31}$$

Because (31) uses only the last $s+1$ vectors, we can even obtain the Ritz vector saving temporally only the last $s+1$ basis vectors. Algorithm 3 presents how to obtain the Ritz vectors of A, after we have obtained the matrix H.

Algorithm 3 Obtaining the Ritz vectors

procedure RITZ VECTORSIDR($A,\ s,\ H,\ \mathbf{r}_0$)
  Input: $A \in \mathbb{C}^{n\times n}$, $s \in \mathbb{N}^{+}$, $\mathbf{x} \in \mathbb{C}^{n}$.
  Obtain $(\lambda_i,\ \hat{\mathbf{y}}_i)$ as the eigenpairs associated with the smallest magnitude eigenvalues of H.
  $\hat{\mathbf{r}}_0 = \mathbf{r}_0$
  $Y = \hat{\mathbf{r}}_0 \times [[\hat{\mathbf{y}}_1]_1, [\hat{\mathbf{y}}_2]_1, \ldots, [\hat{\mathbf{y}}_{\tilde{m}}]_1]$
  for $i = 1$ to $\tilde{m} - 1$ **do**
    $\hat{\mathbf{r}}_i = \frac{1}{h_{i+1,i}}\left[A\hat{\mathbf{r}}_{i-1} - \sum_{j=\max(0,i-s)}^{i-1} h_{j,i}\hat{\mathbf{r}}_j\right]$
    $Y = Y + \hat{\mathbf{r}}_i \times [[\hat{\mathbf{y}}_1]_{i+1}, [\hat{\mathbf{y}}_2]_{i+1}, \ldots, [\hat{\mathbf{y}}_{\tilde{m}}]_{i+1}]$
  end for
  return $\{\lambda\}_{i=1}^{\tilde{m}}$, Y.
end procedure

Once we compute the s Ritz vectors associated with the smallest magnitude, we proceed to use these vectors in IDR(s) with recycling to solve the remaining systems of linear equations. Algorithm 4 summarizes this procedure.

Algorithm 4 IDR(s) with recycling for sequences of system of linear equations

procedure IDR($A,\ \{\mathbf{b}_i\},\ s,\ tol,\ \mathbf{x}_0$)
  call IDR($A,\ \mathbf{b}_1,\ s,\ tol,\ \mathbf{x}_0$) to obtain $\mathbf{x}_1$ and the matrix $H_{\tilde{m}}$ (Algorithm 1).
  call Ritz vectorsIDR($A,\ s,\ H_{\tilde{m}},\ \mathbf{r}_1$) to obtain the Ritz vectors $\{\mathbf{y}_j\}_{j=1}^{s}$ (Algorithm 3)
  for each right-hand side vector $\mathbf{b}_i$ with $i = 2,\ 3,\ \ldots,\ p$ **do**
    call IDR(s) to solve $A\mathbf{x}_i = \mathbf{b}_i$ with the Ritz-vector $\{\mathbf{y}_j\}_{j=1}^{s}$.
  end for
  return $\mathbf{x}_i$ for $i = 1,\ \ldots,\ p$
end procedure

4.2. Numerical experiments

In this section, we conduct two numerical experiments of solving sequences of systems of linear equations (Algorithm 4). We use the same computer setting as is described in Section 3.1. The stopping criterion consider in this experiment is

$$\frac{\|\mathbf{b}_i - A\mathbf{x}_k\|}{\|\mathbf{b}_i\|} < 10^{-6}, \qquad \text{for } i = 1,\ 2,\ \ldots,\ p.$$

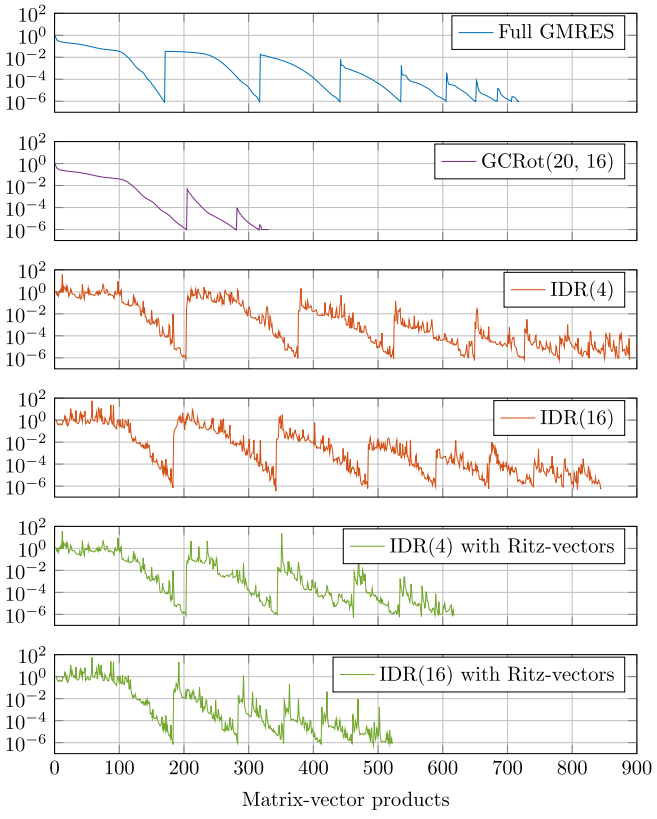

Fig. 7. (Example 7). Convergence residual history for the solution of (32) (diffusion-dominated example).

The initial vector for the first system of linear equations is the zero vector, and for the subsequent linear systems, we use the approximate solution of the previous linear system of equations.

Example 7. In this example, we consider the linear time-dependent convection–diffusion–reaction

$$\frac{\partial u}{\partial t} + \mathbf{v}^T \nabla u = \epsilon \Delta u + \rho u + f \tag{32}$$

with homogeneous Dirichlet conditions on the unit cube, and $u(t_0) = \mathbf{0}$, $\mathbf{v} = [1,\ 1,\ 1]$, $\epsilon = 0.1$ (diffusion-dominated) or $\epsilon = 0.005$ (convection-dominated), the reaction parameter ρ is 5, the function f is obtained from

$$u = \sqrt{x(1-x)y(1-y)z(1-z)}.$$

We solve (32) using Euler backward for time integration for $t \in [0,\ 10]$ with $\delta t = 1$. For space discretization, we use central finite differences with $h = 0.02$ obtaining a linear system of equations of size $125\,000 \times 125\,000$ per time-step. Figs. 7 and 8 show the residual norm behavior for full GMRES, GCROT, and IDR(s) with and without Ritz vector enrichment. First, we can see a good decrement in number of matrix–vector multiplication when IDR(s) is enriched with the Ritz vectors. Second, the long recurrences methods solve all the systems of linear equations using less number of matrix–vector multiplications. However, Tables 2 and 3 show that IDR(s) with Ritz vectors solves the convection and diffusion-dominated problems much faster that GMRES and GCROT, and other short recurrences methods.

5. Conclusions

In this work, we have derived a Hessenberg relation from the IDR(s) method for solving system of linear equations. This is a key component to obtain approximations to the eigenvalues and eigenvectors of the coefficient matrix involved. We have used this spectral information to accelerate the IDR(s) method.

In the first part of this paper, we have proposed a Ritz-IDR(s) variant, named SC-Ritz-IDR(s), to solve systems of linear equations based on the work by Simoncini and Szyld [2]. This algorithm uses the inverse of the Ritz values as parameters

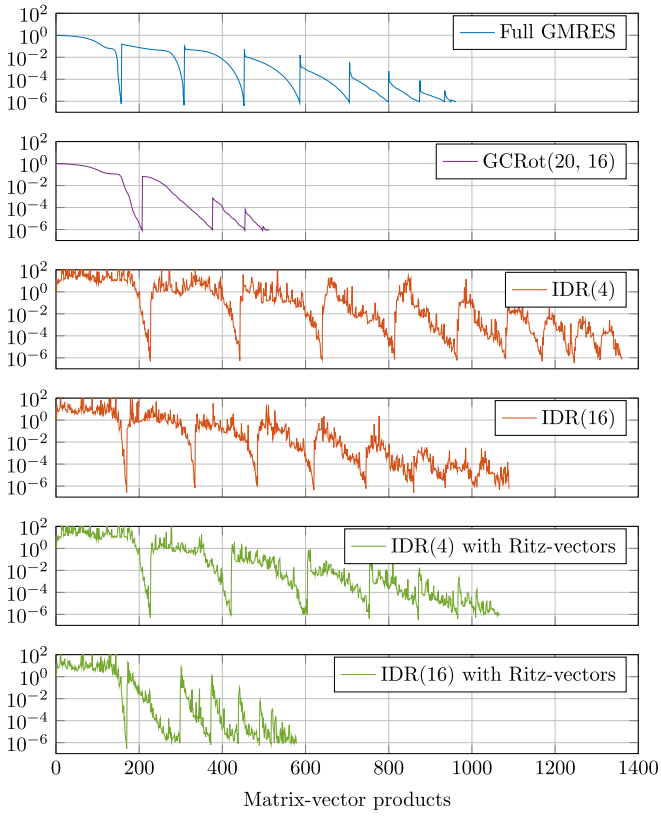

Fig. 8. (Example 7). Convergence residual history for the solution of (32) (convection-dominated example).

Table 2
(Example 7). Matrix–vector multiplications and time used for each method in the solution of (32). (diffusion dominated example)

Method	MATVECs	CPU time [s]
Full GMRES	718	185.93
GCRot(20, 4)	525	33.72
GCRot(20, 16)	332	51.2
BiCG [27]	1946	19.85
BiCGStab	1900	12.13
QMR [28]	1884	22.62
IDR(4) without recycling	889	20.34
IDR(4) with recycling	618	16.86
IDR(16) without recycling	845	36.61
IDR(16) with recycling	523	34.15

Table 3
(Example 7). Matrix–vector multiplications and time used for each method in the solution of (32) (convection dominated example).

Method	MATVECs	CPU time [s]
Full GMRES	962	281.43
GCRot(20, 4)	1380	96.01
GCRot(20, 16)	514	83.61
IDR(4) without recycling	1360	31.46
IDR(4) with recycling	1066	22.57
IDR(16) without recycling	1089	54.44
IDR(16) with recycling	578	37.58

ω_j for the creation of the residuals vectors into the subspaces $\mathcal{G}_j$. In contrast to Ritz-IDR(s), our proposed variant SC-Ritz-IDR(s) is a self-contained algorithm, i. e., it does not use an external sparse eigensolver to compute the Ritz values. In terms of CPU requirements and memory consumption, SC-Ritz-IDR(s) has a similar computational behavior as Ritz-IDR(s) [2].

Implementations of both methods Ritz-IDR(s) and SC-Ritz-IDR(s) may use complex arithmetic, even when the coefficient matrix and the right-hand side vectors are real, in the case of complex Ritz values as parameters ω_j.

In the second part of the paper, we have explained how to enrich the searching subspace of IDR(s) with the Ritz vectors. In particular, we have applied this enrichment to IDR(s) for solving sequences of systems of linear equations. After approximating the eigenvector during the solution of the first system of linear equations, IDR(s) uses this spectral information for the subsequent systems of equations. Numerical experiments show a significant reduction of the computational time.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments and suggestions to improve the quality of the paper.

References

[1] P. Sonneveld, M.B. van Gijzen, IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, SIAM J. Sci. Comput. 31 (2) (2008) 1035–1062. http://dx.doi.org/10.1137/070685804.
[2] V. Simoncini, D.B. Szyld, Interpreting IDR as a Petrov-Galerkin method, SIAM J. Sci. Comput. 32 (4) (2010) 1898–1912. http://dx.doi.org/10.1137/090774756.
[3] Y.-F. Jing, T.-Z. Huang, Y. Duan, B. Carpentieri, A comparative study of iterative solutions to linear systems arising in quantum mechanics, J. Comput. Phys. 229 (11) (2010) 8511–8520. http://dx.doi.org/10.1016/j.jcp.2010.07.034.
[4] M.B. van Gijzen, P. Sonneveld, Algorithm 913: An elegant IDR(s) variant that efficiently exploits biorthogonality properties, ACM Trans. Math. Softw. 38 (1) (2011) 5:1–5:19. http://dx.doi.org/10.1145/2049662.2049667.
[5] T.P. Collignon, M.B. van Gijzen, Minimizing synchronization in IDR(s), Numer. Linear Algebra Appl. 18 (5) (2011) 805–825. http://dx.doi.org/10.1002/nla.764.
[6] P. Sonneveld, On the convergence behavior of IDR(s) and related methods, SIAM J. Sci. Comput. 34 (5) (2012) A2576–A2598. http://dx.doi.org/10.1137/100789889.
[7] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29. http://dx.doi.org/10.1090/qam/42792.
[8] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Natl. Bur. Stand. 45 (4) (1950) 255–281. http://dx.doi.org/10.6028/jres.045.026.
[9] R.B. Morgan, A restarted GMRES method augmented with eigenvectors, SIAM J. Matrix Anal. Appl. 16 (4) (1995) 1154–1171. http://dx.doi.org/10.1137/S0895479893253975.
[10] E. de Sturler, Truncation strategies for optimal Krylov subspace methods, SIAM J. Numer. Anal. 36 (3) (1999) 864–889. http://dx.doi.org/10.1137/S0036142997315950.
[11] E. Agullo, L. Giraud, Y.-F. Jing, Block GMRES method with inexact breakdowns and deflated restarting, SIAM J. Matrix Anal. Appl. 35 (4) (2014) 1625–1651. http://dx.doi.org/10.1137/140961912.
[12] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869. http://dx.doi.org/10.1137/0907058.
[13] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (2) (1983) 345–357. http://dx.doi.org/10.1137/0720023.
[14] M.L. Park, E. de Sturler, G. Mackey, D.D. Johnson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM J. Sci. Comput. 28 (5) (2006) 1651–1674. http://dx.doi.org/10.1137/040607277.
[15] H.V. Nguyen, Krylov Methods for Solving a Sequence of Large Systems of Linear Equations (Ph.D. thesis), Baylor University, Waco, TX, USA, 2015. URL http://hdl.handle.net/2104/9511.
[16] K. Ahuja, Recycling Krylov Dubspaces and Preconditioners (Ph.D. thesis), Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2011 URL http://hdl.handle.net/10919/29539.
[17] M.P. Neuenhofen, M(s)stab(ℓ): A generalization of IDR(s)stab(ℓ) for sequences of linear systems, arXiv:1604.06043, 2016.
[18] M.-C. Yeung, T.F. Chan, ML(k)BiCGSTAB: A BiCGSTAB variant based on multiple Lanczos starting vectors, SIAM J. Sci. Comput. 21 (4) (1999) 1263–1290. http://dx.doi.org/10.1137/S1064827597321581.
[19] M.H. Gutknecht, J.-P.M. Zemke, Eigenvalue computations based on IDR, SIAM J. Matrix Anal. Appl. 34 (2) (2013) 283–311. http://dx.doi.org/10.1137/100804012.
[20] R. Astudillo, M.B. van Gijzen, A restarted induced dimension reduction method to approximate eigenpairs of large unsymmetric matrices, J. Comput. Appl. Math. 296 (2016) 24–35. http://dx.doi.org/10.1016/j.cam.2015.09.014.
[21] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644. http://dx.doi.org/10.1137/0913035.
[22] G.L.G. Sleijpen, M.B. van Gijzen, Exploiting BiCGstab(ℓ) strategies to induce dimension reduction, SIAM J. Sci. Comput. 32 (5) (2010) 2687–2709. http://dx.doi.org/10.1137/090752341.
[23] R. Astudillo, M.B. van Gijzen, The induced dimension reduction method applied to convection-diffusion-reaction problems, in: B. Karasözen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe, Ö. Uğur (Eds.), Numerical Mathematics and Advanced Applications ENUMATH 2015, Springer, Cham, ZG, Switzerland, 2016, pp. 295–303. http://dx.doi.org/10.1007/978-3-319-39929-4_29.
[24] R.F. Boisvert, R. Pozo, K. Remington, R.F. Barrett, J. Dongarra, Matrix market: A web resource for test matrix collections, in: R.F. Boisvert (Ed.), Proceedings of the IFIP TC2/WG2.5 Working Conference on Quality of Numerical Software: Assessment and Enhancement, Chapman & Hall Ltd., London, UK, 1997, pp. 125–137.
[25] J. Baglama, D. Calvetti, L. Reichel, Fast Leja points, Electron. Trans. Numer. Anal. 7 (1998) 124–140.
[26] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1112–1135. http://dx.doi.org/10.1137/S0895479897321362.
[27] R. Fletcher, Conjugate gradient methods for indefinite systems, in: G.A. Watson (Ed.), Proceedings of the Dundee Conference on Numerical Analysis, Springer-Verlag, Berlin, Heidelberg, New York, 1976, pp. 73–89. http://dx.doi.org/10.1007/BFb0080116.
[28] R.W. Freund, N.M. Nachtigal, QMR: A quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1) (1991) 315–339. http://dx.doi.org/10.1007/BF01385726.