Journal of Computational and Applied Mathematics 296 (2016) 24–35

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

A restarted Induced Dimension Reduction method to approximate eigenpairs of large unsymmetric matrices

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

[a] Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD, The Netherlands

[b] Universidad Central de Venezuela, Centro de Cálculo Científico y Tecnológico, Escuela de Computación, 1040, Caracas, Venezuela

ARTICLE INFO

Article history:
Received 16 January 2015

Keywords:
Eigenpairs approximation
Induced Dimension Reduction method
Implicitly restarting
Polynomial filter

ABSTRACT

This work presents a new algorithm to compute eigenpairs of large unsymmetric matrices. Using the Induced Dimension Reduction method (IDR(s)), which was originally proposed for solving systems of linear equations, we obtain a Hessenberg decomposition, from which we approximate the eigenvalues and eigenvectors of a matrix. This decomposition has two main advantages. First, IDR(s) is a short-recurrence method, which is attractive for large scale computations. Second, the IDR(s) polynomial used to create this Hessenberg decomposition is also used as a filter to discard the unwanted eigenvalues. Additionally, we incorporate the implicitly restarting technique proposed by D.C. Sorensen, in order to approximate specific portions of the spectrum and improve the convergence.

1. Introduction

A variety of applications involve the solution of the eigenvalue problem. This problem consists in finding a subset of pairs $(\lambda, \mathbf{x})$ of a matrix $A \in \mathbb{C}^{n\times n}$, such that:

$$A\mathbf{x} = \lambda\mathbf{x}, \quad (1)$$

where $\lambda \in \mathbb{C}$ is called eigenvalue, and the nonzero vector $\mathbf{x} \in \mathbb{C}^n$ is its corresponding eigenvector. When the matrix A is large and unsymmetric, solving the eigenvalue problem becomes computationally challenging.

Methods to approximate a subset of eigenpairs of large unsymmetric matrices are usually based on the construction of a standard Hessenberg decomposition associated with the matrix A, i.e.

$$AU_m = U_mB_m + \mathbf{u}_{m+1}\mathbf{e}_m^T, \quad (2)$$

where $U_m \in \mathbb{C}^{n\times m}$, B_m is a Hessenberg matrix of order m, $\mathbf{u}_{m+1} \in \mathbb{C}^n$, and $\mathbf{e}_m$ is the mth canonical vector, with m being typically much smaller than n. Under certain conditions, the eigenvalues of the matrix B_m approximate a subset of eigenvalues of A.

The Induced Dimension Reduction (IDR(s)) was introduced in 2008 for solving systems of linear equations [1]. IDR(s) is a short-recurrence method which has obtained attention for its rapid convergence and computational efficiency. IDR(s) as a method to compute eigenvalues was first studied by M. H. Gutknecht and J.-P. M. Zemke in [2]. The work that we present here is a continuation of [3]. We describe how to obtain an underlying Hessenberg decomposition of the form (2) from IDR(s), and

[*] Corresponding author at: Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD, The Netherlands.
E-mail addresses: R.A.Astudillo@tudelft.nl (R. Astudillo), M.B.vanGijzen@tudelft.nl (M.B. van Gijzen).

http://dx.doi.org/10.1016/j.cam.2015.09.014
0377-0427/© 2015 Elsevier B.V. All rights reserved.

we combine it with the implicitly restarting technique introduced by D.C. Sorensen [4] for Arnoldi in order to approximate the eigenpairs of interest. Additionally, we suggest a parameter selection for our proposed method which defines a filter polynomial for the spectrum.

This document is structured as follows. In Section 2, we present an overview of the Hessenberg decompositions, which are the basis for large scale eigenvalues/eigenvectors approximation. Section 3 explains how to compute a Hessenberg decomposition based on the IDR method. Two techniques to refine the information obtained from the IDR-Hessenberg factorization are discussed in Section 4. In Section 5, we present numerical experiments to illustrate the behavior of the method proposed.

We use the following notation: capital letters denote matrices, and the transpose of a matrix A is represented by A^T. Column vectors are represented by bold-face, lower case letters. Greek lower case letters represent complex scalars. I_n is the identity matrix of order n, and wherever the context is clear the subindex n is eliminated. Subspaces are denoted by uppercase calligraphic letters.

2. Background on Hessenberg decompositions

In Eq. (2), the columns of the matrix U_m represent a basis for the Krylov subspace,

$$\mathcal{K}_m(A, \mathbf{x}) = \{\mathbf{x},\ A\mathbf{x},\ A^2\mathbf{x}, \dots, A^{m-1}\mathbf{x}\}. \tag{3}$$

The upper Hessenberg matrix B_m is the projection of the matrix A over $\mathcal{K}_m(A, \mathbf{x})$. Projections onto Krylov subspaces are the basis for several methods to solve system of linear equations and eigenpairs approximation (see for example [5,6]). To compute eigenvalues of large, unsymmetric, and sparse matrices, the most common options between the Krylov methods are Bi-Lanczos [7] and the Arnoldi method [8]. Each of them creates a different Hessenberg decomposition associated with the matrix A. Bi-Lanczos method uses a short-recurrence formulas to create two Hessenberg tridiagonal decompositions of the form

$$AV_m = V_m T_m + \mathbf{f}\mathbf{e}_m^T$$

and

$$A^T W_m = W_m T_m^T + \mathbf{s}\mathbf{e}_m^T,$$

where $\mathbf{e}_m$ is the mth canonical vector, $\mathbf{f}$ and $\mathbf{s} \in \mathbb{C}^n,\ T_m \in \mathbb{C}^{m\times m}$ is a tridiagonal matrix, the matrix $V_m \in \mathbb{C}^{n\times m}$ is a basis for $\mathcal{K}_m(A, \mathbf{v}_1),\ W_m \in \mathbb{C}^{n\times m}$ is a basis for $\mathcal{K}_m(A^T, \mathbf{w}_1)$ and the matrices V_m and W_m are bi-orthogonal ($W_m^T V_m = I_m$). However, despite being an efficient short-recurrence method, Bi-Lanczos is numerically unstable (see [9]).

Arnoldi method, on the other hand, builds a Hessenberg decomposition

$$AV_m = V_m H_m + \mathbf{f}\mathbf{e}_m^T,$$

where $\mathbf{f} \in \mathbb{C}^n$ and V_m is a matrix with orthogonal columns and represents a basis for $\mathcal{K}_m(A, \mathbf{v}_1)$. This method is widely used to approximate a subset of the eigenpairs of A; nevertheless, its computational and memory cost increases per iteration. An option to overcome this issue is to restart the process (see [10]). Other Hessenberg decompositions to approximate eigenpairs based on Newton and Chebyshev polynomials can be found in [11–14].

The IDR(s) is a Krylov method proposed for solving systems of linear equations. It is based on the following theorem.

Theorem 1. *Let A be any matrix in $\mathbb{C}^{n\times n}$, let $P = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_s]$ be an $n \times s$ matrix, and let $\{\mu_j\}$ be a sequence in $\mathbb{C}$. With $\mathcal{G}_0 \equiv \mathbb{C}^n$, define*

$$\mathcal{G}_{j+1} \equiv (A - \mu_{j+1} I)(\mathcal{G}_j \cap P^\perp) \quad j = 0, 1, 2 \dots,$$

where $P^\perp$ represents the orthogonal complement of P. If $P^\perp$ does not contain an eigenvector of A, then, for all $j = 0,\ 1,\ 2 \dots,$ the following hold:

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

Proof. See [15,1]. □

In order to solve a system of linear equations, IDR(s) forces the residual vector $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ to be in the nested and shrinking spaces $\mathcal{G}_j$, and then extracts the approximate solution $\mathbf{x}_k$. In the original implementations of IDR(s), the authors do not create explicitly any Hessenberg decomposition [1,16]. M. H. Gutknecht and J.-P. M. Zemke in [17] deduce a generalized Hessenberg decomposition from the IDR(s) method

$$AW_m U_m = W_m \hat{H}_m + \mathbf{w}\mathbf{e}_m^T, \tag{4}$$

from which only the eigenvalues values of A are approximated by the solution of the eigenvalue pencil $(\hat{H}_m, U_m)$. Here the matrices U_m and $\hat{H}_m$ are in $\mathbb{C}^{m\times m}$, with U_m upper triangular and, H_m is a Hessenberg matrix. The matrix W_m is not explicitly built.

In the next section, we present a IDR(s)-based Hessenberg decomposition which generates, in exact arithmetic, the same eigenvalues of the generalized Hessenberg decomposition presented in [2]. It is worth to remark two advantages of the IDR(s) Hessenberg decomposition proposed here. First, we explicitly build the matrix W_m, and, we can approximate the eigenvectors. Second, the IDR(s) decomposition is of the form (2) and this is particularly suitable to apply the implicitly restarted technique of D.C. Sorensen [4].

3. A Hessenberg decomposition based on the IDR(s) method

This section proposes a method to build a standard Hessenberg decompositions using the IDR(s) method. First, we review the generalized Hessenberg decomposition presented in [2], then, we present an equivalent standard Hessenberg decomposition. A vector $\mathbf{w}_{i+1}$ in $\mathcal{G}_j$, according to [1], can be written as

$$\mathbf{w}_{i+1} = (A - \mu_j I)\left(\mathbf{w}_i - \sum_{\ell=1}^{s} c_\ell \mathbf{w}_{i-\ell}\right), \tag{5}$$

where the $s+1$ vectors $\mathbf{w}_{i-s}, \mathbf{w}_{i-s-1}, \dots, \mathbf{w}_i$ belong to $\mathcal{G}_{j-1}$, $\mu_j \in \mathbb{C}$ with $\lfloor \frac{i}{s+1} \rfloor = j$, and the constants c_ℓ are obtained from the solution of the $s \times s$ system of linear equations:

$$(P^T[\mathbf{w}_{i-s}, \mathbf{w}_{i-s+1}, \dots, \mathbf{w}_{i-1}])\mathbf{c} = P^T\mathbf{w}_i.$$

Using Eq. (5), we have:

$$A\mathbf{w}_i = \mathbf{w}_{i+1} + \mu_{j+1}\mathbf{w}_i - \mu_{j+1}\sum_{\ell=1}^{s} c_\ell \mathbf{w}_{i-\ell} + \sum_{\ell=1}^{s} c_\ell A\mathbf{w}_{i-\ell}, \tag{6}$$

or equivalently:

$$A\mathbf{w}_i - \sum_{\ell=1}^{s} c_\ell A\mathbf{w}_{i-\ell} = \mathbf{w}_{i+1} + \mu_{j+1}\mathbf{w}_i - \mu_{j+1}\sum_{\ell=1}^{s} c_\ell \mathbf{w}_{i-\ell}.$$

From the latter equation, the authors in [2] propose a generalized Hessenberg decomposition:

$$AW_mU_m = W_mH_m + \mathbf{w}\mathbf{e}_m^T, \tag{7}$$

where U_m is an upper triangular matrix and $\hat{H}_m$ is an upper Hessenberg matrix; their columns are defined as

$$\mathbf{u}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{and} \quad \hat{\mathbf{h}}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\mu_{j+1}\begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix} \\ \mu_{j+1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The matrix pencil $(\hat{H}_m, U_m)$ is called the Sonneveld pencil. The eigenvalues of this pencil are divided into two sets: $\{\mu_k\}_{k=1}^{t}$ with $t = \lfloor \frac{m-1}{s+1} \rfloor$, and the approximations to the eigenvalues of A or Ritz values $\{\theta_k\}_{k=t}^{m}$. We create an IDR(s)-based standard Hessenberg decomposition of the form (2). Setting $W_k = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k]$, and assuming that $A\mathbf{w}_{i-\ell}$ can be written as a linear combination of the vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_{i-\ell}, \mathbf{w}_{i-\ell+1}$, for $i = 1, 2, \dots, i-1$, we obtain

$$A\mathbf{w}_{i-\ell} = W_{i-\ell+1}\mathbf{h}_{i-\ell}. \tag{8}$$

Combining Eqs. (6) and (8), we obtain

$$A\mathbf{w}_i = W_{i+1}\mathbf{h}_i$$

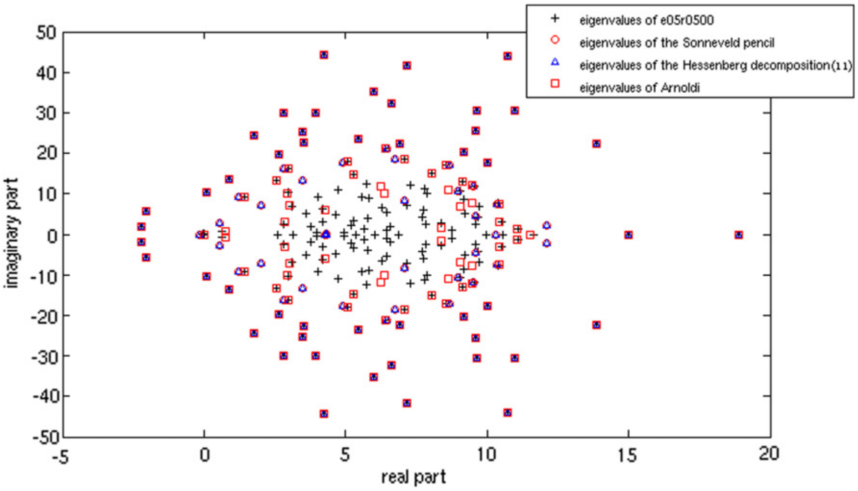

Fig. 1. This figure shows Ritz values generated by our proposed IDR($s = 4$) factorization, the Sonneveld pencil, and Arnoldi.

where

$$\mathbf{h}_i = \left(\begin{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ -\mu_{j+1} \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix} \\ \mu_{j+1} \\ 1 \end{bmatrix} + \sum_{\ell=1}^{s} c_\ell \mathbf{h}_{i-\ell} \right) \quad \text{for } i = s+1, \ldots, m. \tag{9}$$

Applying Eq. (9) for $i = 1, 2, \ldots, m$, we obtain a standard Hessenberg decomposition that we call the **IDR factorization**:

$$\begin{aligned} AW_m &= W_{m+1}\bar{H}_m \qquad (10) \\ &= W_m H_m + \mathbf{w}_{m+1}\mathbf{e}_m^*. \qquad (11) \end{aligned}$$

The matrix W_m is a non-orthogonal basis for the Krylov subspace and the Hessenberg matrix H_m has exactly the same eigenvalues as the matrix Sonneveld matrix pencil (see Fig. 1 for a comparison of Ritz values obtained from the IDR factorization, the Sonneveld pencil, and the Arnoldi method). This result is summarized in the following theorem.

Theorem 2. *Matrix H_m, whose columns are defined in (9), can be written as*

$$H_m = \hat{H}_m U_m^{-1}$$

where the matrices $(\hat{H}, U_m)$ define the Sonneveld pencil proposed in [2].

Proof. (Induction over the columns of H_m.) For $1 \le i \le s+1$, let us assume a starting standard Hessenberg decomposition for the Sonneveld pencil. As an inductive step let us assume that $H_{m\times i} = \hat{H}_{m\times i} U_i^{-1}$, or if we represent the columns of the inverse U_m as $[\mathbf{u}^{-1}]_i$, we can write the previous expression as $\mathbf{h}_k = \hat{H}_{m\times k}[\mathbf{u}^{-1}]_k$ for $1 \le k \le i$. For $k = i+1$, taking into account the structure of the matrix U_m, we obtain that the $i+1$th column is:

$$\hat{H}_{m\times i+1}[\mathbf{u}^{-1}]_{i+1} = \hat{H}_{m\times i+1}\left(\sum_{\ell=1}^{s} c_\ell [\mathbf{u}^{-1}]_{i-\ell} + \mathbf{e}_{i+1} \right).$$

Using the induction hypothesis, we obtain

$$\hat{H}_{m\times i+1}[\mathbf{u}^{-1}]_{i+1} = \sum_{\ell=1}^{s} c_\ell \mathbf{h}_{i-\ell} + \hat{\mathbf{h}}_{i+1},$$

and this is exactly equal to the proposed formula of the $(i+1)$th column of H_m in (9). □

If we assume that k vectors have been created in $\mathcal{G}_j$, then any linear combination of these is also a vector in $\mathcal{G}_j$. Therefore, we can rewrite this equation as

$$\mathbf{w}_{i+1} = (A - \mu_j I)\left(\mathbf{w}_i - \sum_{\ell=1}^{s} c_\ell \mathbf{w}_{i-\ell}\right) + \sum_{\ell=1}^{k} \beta_{i-\ell}\mathbf{w}_{i-\ell}.$$

The selection of the parameters βs yields different versions of IDR(s). For example, choosing the parameter β_ℓ to impose biorthogonality between the set of vectors $\{\mathbf{w}_\ell\}_{\ell=1}^{k}$ and $\{\mathbf{P}_\ell\}_{\ell=1}^{k}$ [16], or to make the vector $\mathbf{w}_{i+1}$ orthogonal to the previous vectors in the subspace [17]. Algorithm 1 outlines the process to create an IDR factorization of size m in which β_ℓs are selected to orthogonalize the vector in $\mathcal{G}_j$.

Algorithm 1 IDR(s) Process applied to a matrix A

```
Given $s \in \mathbb{N}$, $P \in \mathbb{R}^{n\times s}$, $W \in \mathbb{C}^{n\times s+1}$ and $H \in \mathbb{C}^{s+1\times s}$, such that $AW_s = W_{s+1}\bar{H}_s$
for $i = s+1, \ldots, m$ do
    if $i$ is multiple of $s+1$ then
        Choose the parameter $\mu_j$ for the subspace $\mathcal{G}_j$
    end if
    Solve the $s \times s$ system of linear equations
            $(P^T[\mathbf{w}_{i-s},\ \mathbf{w}_{i-s+1},\ \ldots,\ \mathbf{w}_{i-1}])\mathbf{c} = P^T\mathbf{w}_i$
    $\mathbf{v} = \mathbf{w}_i - \sum_{\ell=1}^{s} c_\ell w_{i-\ell}$        ▷ $\mathbf{v} \in \mathcal{G}_{j-1} \cap P^\perp$
    $\mathbf{w}_{i+1} = (A - \mu_j I)\mathbf{v}$        ▷ New vector in $\mathcal{G}_j$
    Create the $i$+1-th column of $H$ according to (9).
    $\beta_{i-\ell} = \mathbf{w}_i^T\mathbf{w}_{i-\ell}$ for $\ell = 1,\ 2, \ldots,\ k$, $\beta_i = \|\mathbf{w}_{i+1} - \sum_{\ell=1}^{k}\beta_{i-\ell}\mathbf{w}_{i-\ell}\|$.
    $\mathbf{w}_{i+1} = (\mathbf{w}_{i+1} - \sum_{\ell=1}^{k}\beta_{i-\ell}\mathbf{w}_{i-\ell})/\beta_i$
    $h_{i-\ell,i} = h_{i-\ell,i} + \beta_{i-\ell}$ for $\ell = 1,\ 2, \ldots,\ k$ and $h_{i+1,i} = \beta_i$
    $W_{i+1} = [\mathbf{w}_1,\ \mathbf{w}_2,\ \ldots,\ \mathbf{w}_i,\ \mathbf{w}_{i+1}]$        ▷ Update the IDR factorization
end for
```

The matrix H_m, created by Algorithm 1, is called the *Sonneveld matrix* in [2]. At this stage, it is worth to comment on the main differences between the work presented here and the work in [2]. First, we create a standard Hessenberg decomposition rather than the Generalized Hessenberg decomposition proposed in [2]. The standard Hessenberg decomposition is suitable to apply the implicit restarting technique. Another difference is that by means of our proposed IDR factorization, the eigenvectors of the matrix A can be approximated without extra calculations. The authors in [2] remove the μ_j values from the spectrum of the Sonneveld pencil $(\hat{H},\ U_m)$; this process is called purification and it creates a smaller pencil that only contains the approximations of the eigenvalues of A. The disadvantage of the purification process is that the Krylov basis has to be recalculated to approximate eigenvectors. The numerical stability difference between our proposed method and the Sonneveld pencil has been recently addressed in [18] by J.-P. M. Zemke.

3.1. Operation count and memory consumption

The Arnoldi method, in the mth iteration, requires one matrix–vector multiplication and $\frac{m(m+1)}{2} + 1$ inner products. For IDR(s), the number of inner products does not depend on the iteration number m. In Algorithm 1, every $s+1$ iterations, performs $\frac{3s^2+3s}{2} + 1$, inner products, $s+1$ matrix–vectors products, and it also requires the solution of $s+1$ systems of linear equations of order s. All of this indicates that the computational work of IDR(s) does not grow in every iteration, in contrast to the Arnoldi method. In terms of computational work, IDR(s) is an intermediate option between Bi-Lanczos and Arnoldi method.

IDR(s) and Arnoldi have similar memory requirements. In the mth iteration, IDR has to store the Hessenberg matrix $\bar{H}_m$ of size $(m+1) \times m$, the matrix W_{m+1} of size $n \times (m+1)$, and additionally the matrix P of size $n \times s$. In some application, however, where only the eigenvalues are required, the IDR(s) could be adapted to low memory requirements. IDR(s) would not need to save all the columns of the matrix W_m; it would only required the last $s+1$ vectors of W_{m+1}, the matrix $\bar{H}_m$, and the matrix P.

3.2. Approximation of the eigenpairs and stopping criteria

To obtain an approximation of the eigenpairs of the matrix A, we first compute an eigenpair of the small matrix H_m, i.e.,

$$H_m\mathbf{y}^{(i)} = \theta_i\mathbf{y}^{(i)} \quad \text{with } \|\mathbf{y}^{(i)}\| = 1.$$

Then, setting our eigenpair approximation as $(\theta_i,\ \mathbf{x}^{(i)} = W_m\mathbf{y}^{(i)})$, and using the relation (10), we have that

$$A\mathbf{x}^i - \theta_i\mathbf{x}^i = AW_m\mathbf{y}^i - \theta_i W_m\mathbf{y}^i = w_{m+1}\mathbf{e}_m^t\mathbf{y}^i.$$

From the previous equation and setting $[\mathbf{y}^{(i)}]_m$ as the mth component of the vector $y^{(i)}$, we can obtain the following bound:

$$\|A\mathbf{x}^{(i)} - \theta_i\mathbf{x}^{(i)}\| \le \|\mathbf{w}_{m+1}\||[\mathbf{y}^{(i)}]_m|, \tag{12}$$

or, if we normalize the vector $\mathbf{w}_m$

$$\|A\mathbf{x}^{(i)} - \theta_i\mathbf{x}^{(i)}\| \le h_{m+1,m}|[\mathbf{y}^{(i)}]_m|. \tag{13}$$

However, it is not suitable to use $(h_{m+1,m}[\mathbf{y}^{(i)}]_m) \le \epsilon$ as stopping criteria, because W_m is not a matrix with orthogonal columns, in consequence, the norm of this matrix produces scaling effect over (12). For this reason, we consider

$$\frac{\|A\mathbf{x}^{(i)} - \theta_i\mathbf{x}^{(i)}\|}{\|W_m\|} \le \epsilon.$$

Then, we need that

$$h_{m+1,m}|[\mathbf{y}^{(i)}]_m|\|W_m\| \le \epsilon.$$

In this way, we avoid the scaling effect of the matrix W_m over the residual bound. Furthermore, it is not necessary to compute the norm of W_m in every iteration. An observation made in [17] is that the matrix W_m has m blocks of size $s+1$ orthogonal vectors then

$$\|W_m\| \le \sqrt{m}.$$

We stop the process when

$$h_{m+1,m}|[\mathbf{y}^{(i)}]_m|\sqrt{m} \le \epsilon. \tag{14}$$

3.3. *Relation between the Arnoldi and other Hessenberg decompositions*

In this section, we review the relation between different Hessenberg decompositions. In particular, we are interested in the difference between a Hessenberg decomposition obtained by Arnoldi and an IDR(s)-Hessenberg decomposition. Let us assume that after m steps of the Arnoldi method applied to the matrix $A \in \mathbb{C}^{n\times n}$, with an initial vector $\mathbf{x} \in \mathbb{C}^n$ and without breakdown, we obtain the following Hessenberg decomposition:

$$AV_m = V_mH_m + \mathbf{f}\mathbf{e}_m^T = V_{m+1}\bar{H}_m. \tag{15}$$

On the other hand, let us consider another Hessenberg decomposition associated with the matrix A and the same initial vector $\mathbf{x}$

$$AX_m = X_mG_m + \mathbf{g}\mathbf{e}_m^T = X_{m+1}\bar{G}_m, \tag{16}$$

where the columns of the matrix X_{m+1} do not form an orthogonal set. One can relate Eqs. (15) and (16) using the reduced QR factorization of the matrix X_{m+1}

$$X_{m+1} = Q_{m+1}R_{m+1}, \tag{17}$$

and obtain

$$AQ_m = Q_{m+1}R_{m+1}\bar{H}_m^{(I)}R_m^{-1}. \tag{18}$$

Comparing Eqs. (15) and (18), we obtain by uniqueness of the Arnoldi Hessenberg decomposition (see Theorem 2.4 in [4]), that $Q_{m+1} = V_{m+1}$, and,

$$\bar{H}_m = R_{m+1}\bar{G}_mR_m^{-1}. \tag{19}$$

The latter equation can be rewritten as

$$H_m = R_mG_mR_m^{-1} + \frac{g_{m+1,m}}{r_{m,m}}\mathbf{r}\mathbf{e}_m^T, \tag{20}$$

where $\mathbf{r} = [r_{i,m+1}]_{i=1}^m$. Eq. (20) can also be found in [19,14]. A direct consequence of this equation is that the Ritz values obtained in an Arnoldi Hessenberg decomposition, and the Ritz values obtained from a factorization that generates a non-orthogonal Krylov basis, are not the same. The condition number of the Krylov basis generated X_m, which is at the same the condition number of R_m, might determine how far the eigenvalues of G_m are from the approximated eigenvalues resulting from the Arnoldi process. To exemplify this, we consider the following matrix, in Matlab notation:

$$\mathtt{A} = \mathtt{sprandn}(100, 100, 0.2). \tag{21}$$

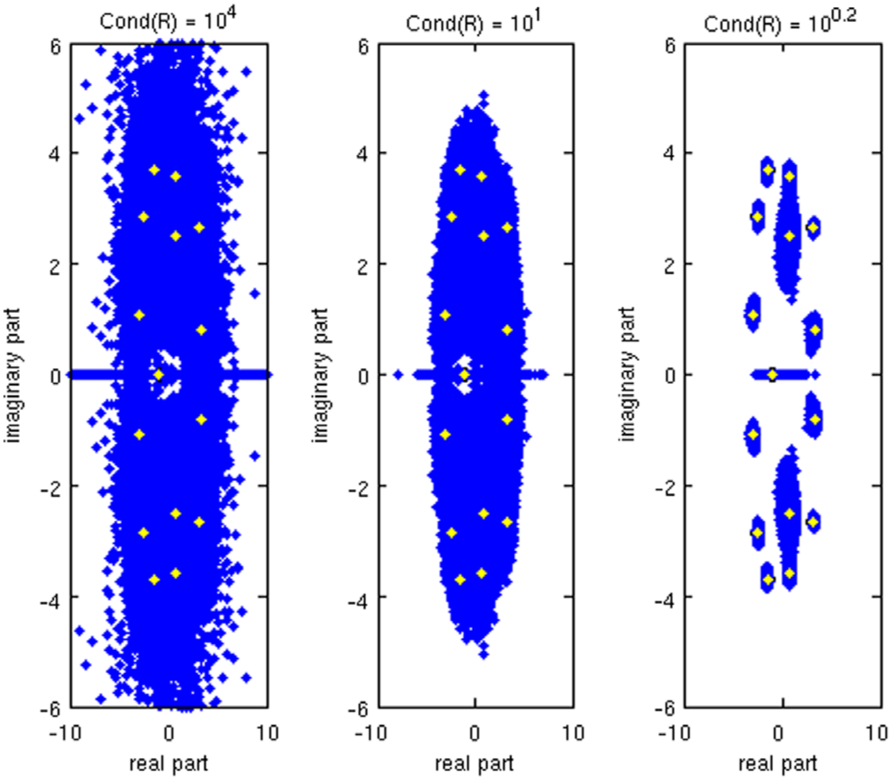

Fig. 2. For the matrix (21), we plot the Ritz values of $\bar{H}_{10}$ generated by Arnoldi (yellow dots) and the Ritz values of matrices G_{10} (blue dots), where $\bar{G}_{10} = R_{11}^{-1}\bar{H}_{10}R_{10}$. Matrices R are generated randomly in three groups. In each group, the matrices R have a fixed condition number. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In Fig. 2, we plot the Ritz values obtained by $\bar{H}_{15}$ generated by Arnoldi, and the Ritz values obtained by $\bar{G}_{15} = R_{16}^{-1}\bar{H}_{15}R_{15}$. We select randomly three groups of 10 000 matrices R each, with different condition numbers. One can observe that the Ritz values tend to be more clustered around Arnoldi's Ritz values, when the condition number of matrices R decrease.

Now, let us turn to the case of IDR(s), we consider again the matrix (21), and we obtain different IDR(s)-Hessenberg decompositions

$$AW_{70} = W_{70}H_{70} + \mathbf{w}\mathbf{e}_{70}^T,$$

for $s = 1, 2, 3 \ldots, 35$. Upper part of Fig. 3 shows the evolution of the condition number of the matrix W_{70} generated by IDR(s), when the value s increases. One can observe how the condition number of W_{70} decreases while parameter s increases. Despite the high condition numbers of the matrices W_{70}, IDR(s) generates, in this experiment, a good approximation of the largest magnitude Ritz value $\lambda_1^{(A)}$ generated by Arnoldi (see lower Fig. 3). This analysis suggests that the Ritz values of largest magnitude, generated by IDR(s), are closer to some of the Ritz values generated by Arnoldi when we select large values of s. In [20], P. Sonneveld drew a similar conclusion for IDR(s) in the context of solving systems of linear equations; using stochastic analysis, he related the behavior of the Arnoldi-based method GMRES [21] and IDR(s) when s tends to infinity.

4. Refining the spectral information

In some applications, it is important to find eigenvalues and their corresponding eigenvectors in a specific region of the complex plane. For example, the eigenvalues with largest real part for stability analysis, or the nearest eigenvalues to a given point for vibration analysis. For this reason, we implement two techniques to refine the spectral information obtained from the Hessenberg relation described in the previous section.

4.1. A polynomial filter based on the selection of the parameters μ_j

The explicit restart is one of the first ideas to restart a Hessenberg decomposition [10]. This is based on initiating the process with an improved initial guess. The new initial guess can be a linear combination of the approximated wanted eigenvectors, or an initial guess of the form

$$\mathbf{v}_1^+ = p_k(A)\mathbf{v}_1, \qquad (22)$$

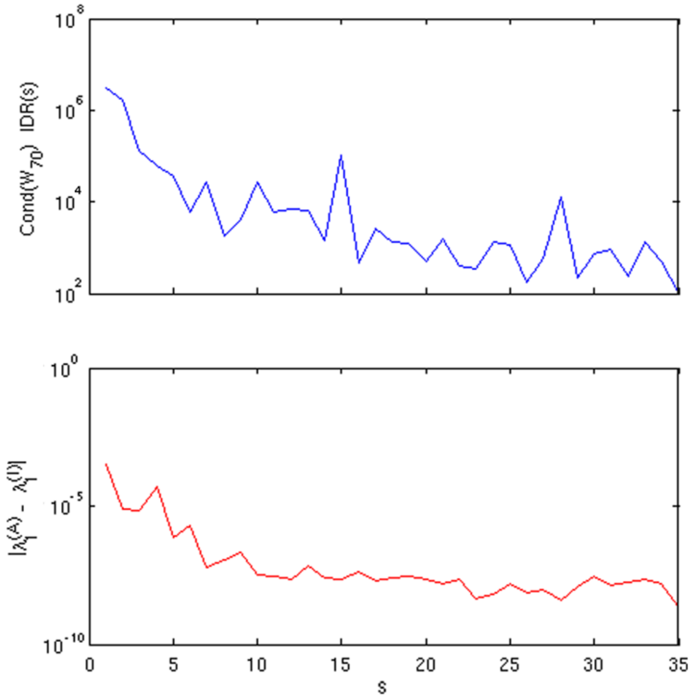

Fig. 3. Upper: condition number of the matrix W_{70} generated by IDR(s) as function of s. Lower: difference between the largest magnitude Ritz values generated by Arnoldi ($\lambda_1^{(A)}$) and IDR(s) ($\lambda_1^{(I)}$) as function of the value s.

where p_k is a polynomial which amplifies the components of $\mathbf{v}_1$ toward the desired eigenvectors and reducing those in the remaining eigenvectors (see [22]). The polynomial p_k is called a filter polynomial. An example of the form of p_k might be

$$p_k(t) = (t - \omega_1)(t - \omega_2) \dots (t - \omega_k). \tag{23}$$

Different options to select the parameter $\{\omega_i\}_{i=1}^k$ have been studied in [22,4,23].

For IDR(s), the authors prove in [1] that every vector in the subspaces $\mathcal{G}_j$ satisfies

$$\mathbf{w}_k = \Psi_{m-j}(A)\Omega_j(A)\mathbf{w}_1, \quad \text{where } \Omega_j(t) = (t - \mu_1)(t - \mu_2) \dots (t - \mu_j), \tag{24}$$

and $\Psi_{m-j}(A)$ is a polynomial of degree $m - j$ and its coefficients are fully determined by the IDR(s) procedure. There is an analogy between Eqs. (22) and (23), and Eq. (24). We exploit this fact by applying the polynomial Ω_j as a polynomial filter. This idea is similar to the one presented by Saad in [22]. We select the parameters μ_i in (24) to minimize the infinity norm of Ω_j in the area where the unwanted eigenvalues are localized. This is achieved by choosing μ_i as the Chebyshev nodes on the interval $[l, u]$, where l and u are the foci of the ellipse that encloses the unwanted portion spectrum of the matrix H_m (see Fig. 4). The polynomial filter $\Omega_j(A)$ is not fixed, the polynomial $\Omega_j(A)$ grows when the IDR process creates vectors in a new subspace $\mathcal{G}_j$. We stress that the IDR polynomial filter is implicitly applied over the vector $\mathbf{w}_k$: it does not require any additional computation; it is achieved by a special choice of the parameters μ_j.

While the authors in [2] remove the parameters $\{\mu_i\}$ by a process called purification, we exploit these parameters to turn $\Omega_j(A)$ into a polynomial filter inherent to the IDR(s) process.

4.2. Implicitly restarting

The most successful variant to approximate subsets of eigenvalues and their corresponding eigenvectors of large and sparse matrices is the Implicitly Restarted Arnoldi Method (IRAM) proposed by D.C. Sorensen in 1992. This method is also the basis for the software package ARPACK [24]. The idea is to truncate the Hessenberg decomposition by removing the uninteresting part of the spectrum using QR steps. After this truncation, the Hessenberg decomposition is expanded to improve the Ritz values and eigenvalues in the direction of the wanted portion of the spectrum.

The main idea of the implicit restart is to apply orthogonal transformations to the Hessenberg decomposition to reorder the Ritz values. To illustrate this, let us consider a Hessenberg decomposition of size $m + 1$

$$AZ_{m+1} = Z_{m+1}H_{m+1} + \mathbf{z}_{m+2}\mathbf{e}_{m+1}^*, \tag{25}$$

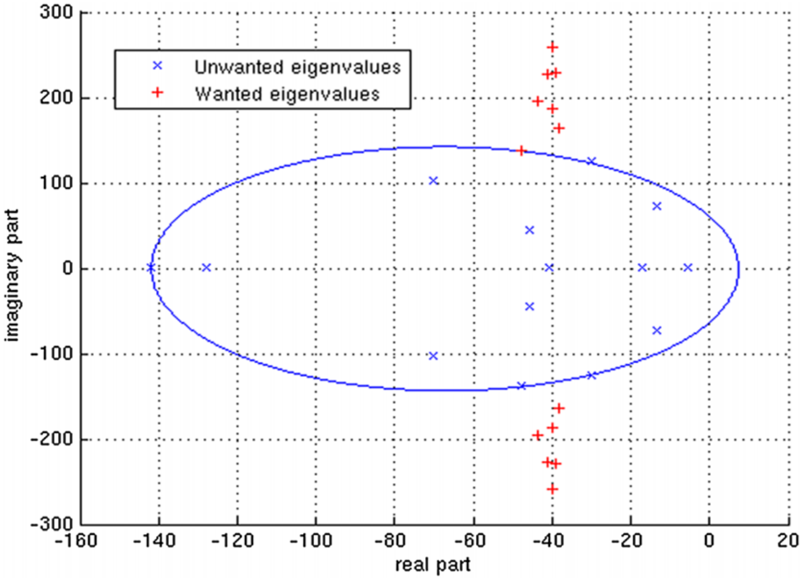

Fig. 4. Select μ_j to minimize the norm of the IDR polynomial (24) in the ellipse which encloses the unwanted eigenvalues.

and suppose that λ is an unwanted Ritz value (it is also called exact shift). Consider the orthogonal matrix Q and the upper triangular matrix R, such that:

$$H_{m+1} - \lambda I = QR.$$

If we multiply Eq. (25) by Q on the right, and define $\hat{H}_{m+1} = Q^* H_{m+1} Q$, and $\hat{Z}_{m+1} = Z_{m+1} Q$ we obtain

$$A\hat{Z}_{m+1} = \hat{Z}_{m+1}\hat{H}_{m+1} + \mathbf{z}_{m+2}\mathbf{e}^*_{m+1}Q.$$

Now, if we discard the vectors $\hat{\mathbf{z}}_{m+2}$, and truncate the matrix $\hat{Z}_{m+1}$ to m columns, we obtain a new Hessenberg decomposition of size m

$$A\hat{Z}_m = \hat{Z}_m\hat{H}_m + \hat{\mathbf{z}}_{m+1}\mathbf{e}^*_m Q$$

which does not contain the unwanted Ritz value λ.

The implicitly restarted Arnoldi has been analyzed in different works for example: [25,23,26]. Another variant of implicit restarting using the Schur factorization was proposed by Stewart in [27], and a new implementation was recently proposed by Bujanović and Drmač in [28].

In the context of an IDR factorization, we implement the implicit restarting technique taking advantage of the input parameter of Algorithm 1. After the creation of an IDR factorization of size m, we discard the unwanted Ritz values using the implicit restarting technique, and then we truncate it to obtain a new Hessenberg factorization of size s which is the input parameter of the iterative process. The value of s should be greater or equal to the number of wanted eigenpairs. Algorithm 2 outlines the IDR(s) with implicit restarting.

Algorithm 2 Implicitly restarting of an IDR factorization

```
1: Given an initial Hessenberg relation of size $s$. The value of $s$ should be greater or equal to the number of wanted eigenvalues, and $m > s$.
2: Expand the initial factorization using Algorithm 1, to obtain the IDR factorization of size $m$:
       $AW_s = W_sH_s + \mathbf{w}_{s+1}\mathbf{e}^*_s \rightarrow AW_m = W_mH_m + \mathbf{w}_{m+1}\mathbf{e}^*_m$
3: Reorder the IDR factorization, using as a shift $\lambda \in \{\mu_1, \ldots, \mu_j\} \cup \{\text{the unwanted eigenvalues}\}$.
4: Truncate the IDR factorization to obtain the new Hessenberg relation of size $s$ in which the unwanted Ritz values were eliminated.
5: Test for convergence. If no convergence **go to 2** with the new Hessenberg relation else **return**
```

5. Numerical experiments

In this section, we present six numerical experiments to illustrate the computational performance of the IDR(s) for eigenvalues computations. First, we compare the basic IDR (Algorithm 1) with the basic Arnoldi. In the other experiments, we

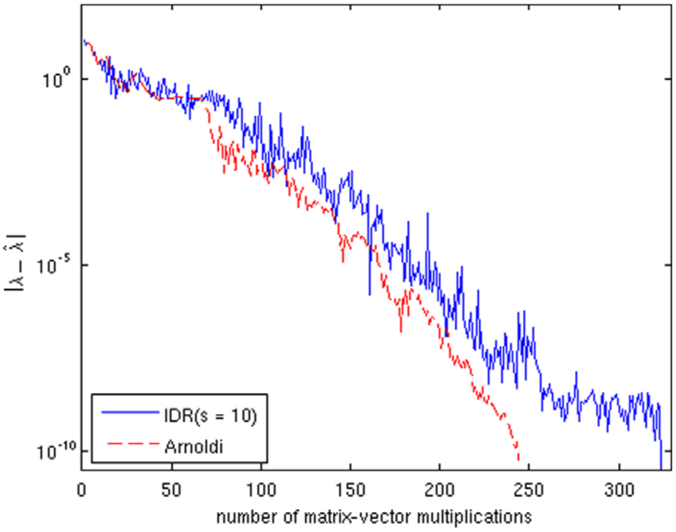

Fig. 5. Experiment 1. Evolution of the absolute error between $\lambda = -10.0581 + 0.27421i$ and its approximations $\hat{\lambda}$ obtained from Arnoldi and IDR($s = 10$).

compare the implicitly restarted version of IDR and Arnoldi. All the experiments were executed using Matlab 8.0 (R2012b) on a computer with I7 Intel processor 2.4 GHz, 4 GB of RAM memory.

In the case of the parameter selection for IDR(s) with implicit restart, we select s as the number of wanted eigenvalues, and, the parameter m is selected as $2 \times s$. $P \in \mathbb{C}^{n\times s}$ is a random matrix with orthogonal columns. The initial guess is selected randomly and we use this vector as initial guess in both IDR and IRAM. In the first iteration the initial Hessenberg factorization for IDR(s) of size s is computed using the Arnoldi method. The selection of the parameter μ_j is discussed in Section 4.1. We adapt the formula (14) for IDR(s) with implicit restart for multiple eigenpairs in the following way:

$$h_{m+1,m} \max_{1\le i\le s} |[\mathbf{y}^{(i)}]_m|\sqrt{m} \le \epsilon$$

where $\epsilon = 10^{-10}\|A\|_F$.

From the second to the sixth experiment, we compare the CPU time of IDR(s) with implicit restart implemented in Matlab and two implementations of IRAM: the first one is a Matlab interpreted code and the second one is the built-in command `eigs`. The command `eigs` is an interface for the ARPACK's FORTRAN-native-code. Therefore, in most of the experiments the command `eigs` produces the shortest CPU time, when this is compared with other native Matlab codes.

Experiment 1. We consider a random sparse matrix of dimension 1000. This matrix is generated in Matlab using the following command[1]

$$\mathtt{A} = \mathtt{sprandn}(1000, 1000, 0.1);$$

We compare the basic versions of IDR($s = 10$) and Arnoldi to find the eigenvalue of A with largest module $\lambda = -10.0581 + 0.27421i$. The parameters μ_j and P for IDR are selected randomly. Fig. 5 shows the evolution of the absolute error for each algorithm. We stop the algorithms when the absolute error is smaller than 10^{-10}. The Arnoldi method takes 244 matrix–vector multiplications to obtain the desired reduction in the absolute error, while IDR($s = 10$) executes 324 matrix–vector multiplications. The execution time for Arnoldi is 0.184 s and for IDR($s = 10$) is 0.09 s.

Experiment 2. As the second example, we consider the Toeplitz tridiagonal matrix

```
A = gallery('tridiag',n,-1,2,-1)
```

of order 1000. Table 1 shows the comparison between IRAM and the implicitly restarted IDR to compute the 15 largest algebraic eigenvalues of this matrix. Fig. 6 shows the absolute error of the methods computing the largest eigenvalue of this matrix $\lambda = 2 + 2\cos(\frac{\pi}{1001})$ in 85 implicit restarting cycles.

Experiment 3. We consider the real nonsymmetric matrix CK656 from the collection Non-Hermitian Eigenvalue Problem [29]. In this example we compute the 24 eigenvalues with largest module [29]. Table 2 shows the comparison of IRAM and the implicitly restarted IDR.

Experiment 4. In the fourth experiment, we compute 12 of the largest magnitude eigenvalues of the matrix AF23560 from the Non-Hermitian Eigenvalue Problem Collection (NEP) [29]. AF23560 is a real nonsymmetric matrix of order 23 560. Table 3 presents the comparison between IRAM and the implicitly restarted IDR with different parameter selection.

Experiment 5. We consider the real unsymmetric matrix HOR131 of dimension 434 × 434. We aim to compute the 8 eigenvalues with largest real part. Table 4 shows the results obtained from IRAM and different parameters of the implicitly restarted IDR.

[1] Using the default values for the random number generator with the command `rng('default')`.

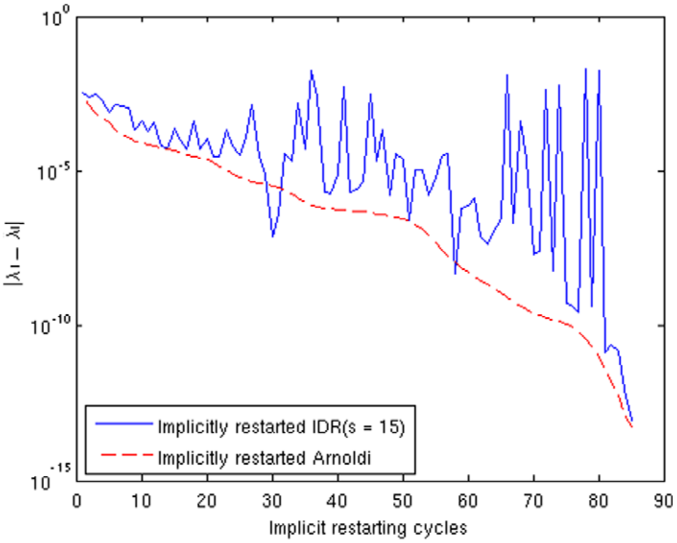

Fig. 6. Experiment 2. Evolution of the absolute error between $\lambda = 2 + 2\cos(\frac{\pi}{1001})$ and its approximations $\hat{\lambda}$ obtained from IRAM and the Implicitly restarted IDR ($s = 15$).

Table 1
Experiment 2. Comparison between IRAM and IDR target: the 15 right most eigenvalues.

Method	Restarts	Time (s)	Residual bound	Max. diff. with `eigs`
IRAM ($k = 15, p = 32$)	141	1.19	5.86e−09	6.70e−14
IDR ($s = 15, m = 32$)	91	0.80	3.70e−09	2.41e−08
IDR ($s = 15, m = 48$)	34	0.77	2.49e−09	1.83e−08
Time `eigs` command: 0.72 s				

Table 2
Experiment 3. Comparison between IRAM and IDR asked for the 24 leftmost eigenvalues.

Method	Restarts	Time (s)	Residual bound	Max. diff. with `eigs`
IRAM ($k = 24, p = 50$)	19	0.31	4.83e−10	5.21e−15
IDR ($s = 24, m = 50$)	20	0.35	1.69e−13	1.62e−12
IDR ($s = 24, m = 75$)	8	0.34	1.31e−09	2.29e−09
IDR ($s = 24, m = 100$)	6	0.58	1.13e−09	9.55e−13
Time `eigs` command: 0.17				

Table 3
Experiment 4. Comparison between IRAM and IDR asked for the 12 eigenvalues of largest magnitude.

Method	Restarts	Time (s)	Residual bound	Max. diff. with `eigs`
IRAM ($k = 12, p = 26$)	6	0.64	2.78e−08	4.74e−14
IDR ($s = 12, m = 26$)	7	0.47	7.23e−08	1.72e−10
IDR ($s = 12, m = 39$)	3	0.44	2.23e−07	9.74e−09
IDR ($s = 12, m = 53$)	2	0.54	7.66e−07	2.48e−09
Time `eigs` command: 0.26				

Table 4
Experiment 5. Comparison between IRAM and IDR asked for the 8 eigenvalues of largest real.

Method	Restarts	Time (s)	Residual bound	Max. diff. with `eigs`
IRAM ($k = 8, p = 18$)	6	0.03	2.19e−13	4.54e−15
IDR ($s = 8, m = 18$)	13	0.06	4.46e−11	8.05e−10
IDR ($s = 8, m = 27$)	7	0.19	1.38e−10	2.65e−10
IDR ($s = 8, m = 36$)	3	0.04	8.23e−13	1.70e−09
Time `eigs` command: 0.02				

Table 5
Experiment 6. Comparison between IRAM and IDR asked for the 16 leftmost eigenvalues.

Method	Restarts	Time (s)	Residual bound	Max. diff. with `eigs`
IRAM ($k = 16, p = 34$)	10	11.19	2.69e−11	2.06e−11
IDR ($s = 16, m = 34$)	12	11.88	9.73e−11	6.01e−08
IDR ($s = 16, m = 50$)	6	12.6	2.25e−11	1.72e−09
Time `eigs` command: 2.92				

Experiment 6. In the sixth experiment, we consider the matrix that arises from the finite difference discretization of the 2D Schrödinger equation. This equation models the energy levels of the confined hydrogen atom, and is given by

$$-u''(x,\ y) - \frac{2u(x,\ y)}{\|(x,\ y)\|} = \lambda u(x, y) \quad (x, y) \in (-16,\ 16) \times (-16,\ 16)\,, \tag{26}$$

with homogeneous Dirichlet boundary conditions. We use a nonuniform mesh refined near the origin and obtain a matrix of size 44100 × 44100. We are interested to approximate the 16 leftmost eigenvalues. We apply IRAM and the Implicitly Restarted IDR to the matrix $(A - \sigma I)^{-1}$, where $\sigma = -2.1$. Table 5 shows the comparison between these two methods.

6. Conclusions

This work has introduced an algorithm to compute eigenpairs of large matrices using a Hessenberg decomposition based on the IDR(s) method. The main advantage of the proposed Hessenberg decomposition is its low computational cost since it only uses recurrences of size $s + 1$. We have implemented two techniques in order to refine the spectral information obtained. The first technique is based on the parameter selection for our proposed algorithm and the second technique is Sorensen's implicitly restart.

The Krylov subspace basis created by our IDR-Hessenberg decomposition is only locally orthogonal, which might has a negative effect on the convergence speed or numerical stability. In the numerical examples presented in this work, however, IDR(s) for eigenvalues shows competitive performance respect to IRAM. This interesting fact, in conjunction with the computational efficiency to compute the IDR-Hessenberg factorization, can also be exploited in applications when only the eigenvalues are required.

References

[1] P. Sonneveld, M.B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Comput. 31 (2) (2008) 1035–1062.
[2] M.H. Gutknecht, J.-P.M. Zemke, Eigenvalue computations based on IDR, SIAM J. Matrix Anal. Appl. 34 (2) (2013) 283–311.
[3] R. Astudillo, M.B. van Gijzen, An induced dimension reduction algorithm to approximate eigenpairs of large nonsymmetric matrices, in: 11th International Conference of Numerical Analysis and Applied Mathematics 2013, vol. 1558, AIP Publishing, 2013, pp. 2277–2280.
[4] D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1) (1992) 357–385.
[5] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, USA, 2003.
[6] Y. Saad, Numerical Methods for Large Eigenvalue Problems: Revised Edition, in: Classics in Applied Mathematics, SIAM, Philadelphia, USA, 2011.
[7] 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.
[8] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.
[9] B.N. Parlett, D.R. Taylor, Z.A. Liu, A look-ahead lanczos algorithm for unsymmetric matrices, Math. Comp. 44 (169) (1985) 105–124.
[10] Y. Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, Linear Algebra Appl. 34 (1980) 269–295.
[11] W.D. Joubert, G.F. Carey, Parallelizable restarted iterative methods for nonsymmetric linear systems. Part I: Theory, Int. J. Comput. Math. 44 (1–4) (1992) 243–267.
[12] W.D. Joubert, G.F. Carey, Parallelizable restarted iterative methods for nonsymmetric linear systems. Part II: parallel implementation, Int. J. Comput. Math. 44 (1–4) (1992) 269–290.
[13] Z. Bai, D. Hu, L. Reichel, A Newton basis GMRES implementation, IMA J. Numer. Anal. 14 (4) (1994) 563–581.
[14] B. Philippe, L. Reichel, On the generation of Krylov subspace bases, Appl. Numer. Math. 62 (9) (2012) 1171–1186.
[15] G.L.G. Sleijpen, P. Sonneveld, M.B. van Gijzen, Bi-CGSTAB as an induced dimension reduction method, Appl. Numer. Math. 60 (2010) 1100–1114.
[16] M.B. van Gijzen, P. Sonneveld, Algorithm 913: An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties, ACM Trans. Math. Software 38 (1) (2011) 5:1–5:19.
[17] M.B. van Gijzen, G.L.G. Sleijpen, J.-P.M. Zemke, Flexible and multi-shift induced dimension reduction algorithms for solving large sparse linear systems, Numer. Linear Algebra Appl. (2014) 1–25.
[18] J.-P.M. Zemke, On Structured Pencils Arising in Sonneveld Methods. Technical Report 186, Technischen Universität Hamburg-Harburg, 2014.
[19] A. Essai, Weighted fom and gmres for solving nonsymmetric linear systems, Numer. Algorithms 18 (3–4) (1998) 277–292.
[20] P. Sonneveld, On the convergence behavior of IDR(s) and related methods, SIAM J. Sci. Comput. 34 (5) (2012) A2576–A2598.
[21] 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.
[22] Y. Saad, Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems, Math. Comp. 42 (166) (1984) 567–588.
[23] R.B. Morgan, On restarting the Arnoldi method for large nonsymmetric eigenvalue problems, Math. Comp. 65 (215) (1996) 1213–1230.
[24] R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, Philadelphia, USA, 1998.
[25] R.B. Lehoucq, D.C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, SIAM J. Matrix Anal. Appl. 17 (4) (1996) 789–821.
[26] R.B. Lehoucq, Implicitly restarted arnoldi methods and subspace iteration, SIAM J. Matrix Anal. Appl. 23 (2) (2001) 551–562.
[27] G.W. Stewart, A Krylov–Schur algorithm for large eigenproblems, SIAM J. Matrix Anal. Appl. 23 (3) (2001) 601–614.
[28] Z. Bujanović, Z. Drmač, A new framework for implicit restarting of the Krylov–Schur algorithm, Numer. Linear Algebra Appl. (2014) 1–13.
[29] Z. Bai, D. Day, J. Demmel, J. Dongarra, A Test Matrix Collection for Non-Hermitian Eigenvalue Problems. Technical Report, University of Tennessee, 1997.