Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Eigenvalue computations based on IDR

M.H. Gutknecht and J.-P.M. Zemke*

Research Report No. 2010-13
May 2010

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

*Institut für Numerische Simulation, Technische Universität Hamburg-Harburg, Germany

Research Report No. 2010-13 SAM, ETH Zurich &
Bericht 145 INS, TUHH

EIGENVALUE COMPUTATIONS BASED ON IDR*

MARTIN H. GUTKNECHT† AND JENS-PETER M. ZEMKE‡

Abstract. The Induced Dimension Reduction (IDR) method, which has been introduced as a transpose-free Krylov space method for solving nonsymmetric linear systems, can also be used to determine approximate eigenvalues of a matrix or operator. The IDR residual polynomials are the products of a residual polynomial constructed by successively appending linear smoothing factors and the residual polynomials of a two-sided (block) Lanczos process with one right-hand side and several left-hand sides. The Hessenberg matrix of the OrthoRes version of this Lanczos process is explicitly obtained in terms of the scalars defining IDR by deflating the smoothing factors. The eigenvalues of this Hessenberg matrix are approximations of eigenvalues of the given matrix or operator.

Key words. Krylov space method; iterative method; induced dimension reduction; large nonsymmetric eigenvalue problem

AMS subject classifications. 65F15 (primary); 65F10; 65F50

1. Introduction.

Induced Dimension Reduction (IDR) is a general concept for defining certain Krylov subspace methods for solving systems of linear equations. While the rationale behind the IDR approach differs considerably from other approaches to Krylov subspace solvers, the resulting methods are mathematically not much different from well-known Krylov methods. In particular, as we will show here, there is also the possibility to extract eigenvalue information from the recurrence coefficients constructed in IDR methods.

1.1. The IDR approach.

The first IDR method was presented 1979 and published 1980 by Sonneveld [47]. It is nearly mathematically equivalent to the BiCGStab method introduced ten years later by van der Vorst and Sonneveld [45, 44]. Consider $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is a general complex $N \times N$ matrix and $\mathbf{b} \in \mathbb{C}^N$. As for any standard Krylov subspace method, the approximate solutions generated by IDR satisfy

$$\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)\,, \tag{1.1}$$

where $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is the initial residual, and

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

is the nth Krylov subspace generated by $\mathbf{A}$ from $\mathbf{r}_0$. Clearly, $\mathcal{K}_n \subseteq \mathbb{C}^N$, and, if the data are real, $\mathcal{K}_n \subseteq \mathbb{R}^N$. Relation (1.1) implies that for the nth residual $\mathbf{r}_n := \mathbf{b} - \mathbf{A}\mathbf{x}_n$ holds

$$\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \subset \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)\,. \tag{1.2}$$

In the original IDR, it must additionally satisfy

$$\mathbf{r}_n \in \mathcal{G}_j\,, \qquad \text{where} \quad j := \left\lfloor \frac{n}{2} \right\rfloor \tag{1.3}$$

and where the Sonneveld spaces $\mathcal{G}_j$ [37] are defined by a recurrence of the form

$$\mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})\,. \tag{1.4}$$

---

*Version of May 4, 2010, 17:30.

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

‡Institut für Numerische Simulation, Technische Universität Hamburg-Harburg, D-21073 Hamburg, Germany (zemke@tu-harburg.de). Part of this work was carried out while the author was visiting Kyushu University in Fukuoka on invitation of Prof. Seiji Fujino and the University of Tokyo.

Here, $\mathcal{S}$ is a hyperplane of $\mathbb{C}^N$, and the parameters ω_j are normally chosen such that $\mathbf{r}_{2j}$ is as short as possible. At the beginning, $\mathcal{G}_0 = \mathcal{K}_N$, but no basis of this invariant Krylov subspace is needed. It turns out that the spaces $\mathcal{G}_j$ are nested: $\mathcal{G}_{j+1} \subset \mathcal{G}_j$, and that under mild restrictions on the hyperplane $\mathcal{S}$ in a finite number of steps they reduce to the null space, so that $\mathbf{r}_n = \mathbf{o}$. It can be shown that the even indexed IDR residuals are BiCGStab residuals: $\mathbf{r}_{2j} = \mathbf{r}_j^{\text{BiCGStab}}$. In particular, the polynomial $\Omega_j(z) := (1-\omega_1 z)\cdots(1-\omega_j z)$ is again a factor of the residual polynomial ρ_{2j} associated with $\mathbf{r}_{2j}$.

Recently, Sonneveld and van Gijzen [39, 40] considerably generalized and improved the original IDR method. In their IDR(s) the subset $\mathcal{S}$ is a subspace of codimension s, and the indices n and j in (1.3) are linked by

$$j := \left\lfloor \frac{n}{s+1} \right\rfloor .$$

This method can be seen to be related to the ML(k)BiCGStab method with $k = s$ of Yeung and Chan [50] and thus to the nonsymmetric band Lanczos process with one right-hand side starting vector and s left-hand side starting vectors [3]. In the case $s = 1$, the IDR(s) prototype algorithm of [39, 40] differs slightly from the original IDR of [47], but again $\mathbf{r}_{2j} = \mathbf{r}_j^{\text{BiCGStab}}$. Hence, it is still essentially equivalent to BiCGStab. If $s > 1$, there is additional freedom in the choice of the "intermediate" residuals $\mathbf{r}_n$ with $n \neq (s+1)j$. This has been capitalized upon in the IDRBiO variant of IDR(s) described in [46]. Reformulations of the IDR approach have been considered in [35, 36]. Modifications of the basic recurrence (1.4) have lead to further similar methods such as IDRStab by Sleijpen and van Gijzen [37], and GIDR(s, L) and GBi-CGSTAB(s, L) by Tanio and Sugihara [42, 43]. Currently, Abe and Sleijpen [1, 2] work out the details of IDR variants which adapt ideas of BiCGStab2 [18], BiCG×MR2 [27], and GPBiCG [56] to the IDR framework, which we term IDRStab2, IDR×MR2, and GPIDR, respectively.

The convergence behavior of IDR methods is largely not understood. A stochastic analysis of basic IDR variants which relates the convergence behavior to the one of GMRes has recently been published by Sonneveld [38]. A step towards understanding the behavior also in finite precision is our investigation of the relation of IDR to two-sided Lanczos processes.

It is straightforward how to extract partial eigenvalue information from a run of BiCGStab, since this method explicitly determines the recurrence coefficients of a nonsymmetric Lanczos process, and thus, of a tridiagonal "projection" of $\mathbf{A}$. In this paper we show how the same eigenvalue information can be extracted from a run of IDR(1), and we investigate the generalization of this eigensolver to the case $s > 1$. We only cover the prototype method of [39, 40], but our approach carries over to other members of the IDR family mentioned above.

In particular, we consider here the transition from the prototype IDR(s) method to a two-sided (block) Lanczos process with one right-hand side and s left-hand sides in OrthoRes form (defined below in Subsection 1.3). The OrthoRes form of this Lanczos process will be denoted for brevity by BiORes($s, 1$), in analogy to the terminology introduced in [5] and [17, 19]. As the IDR(s) variant in [39, 40] has similarities with OrthoRes [51] and is based on a short-term recurrence we denote it by IDR(s)ORes. The transition from IDR(s)ORes to BiORes($s, 1$) is analogous to the corresponding transition from classical IDR [47] or BiCGStab[1] [44] to the BiORes version of BiCG [17, 19]. The first similar transition from a linear equation solver to an eigenvalue solver, namely, from the Hestenes and Stiefel variant of CG, which is the OrthoMin-variant of CG, to the Lanczos method tailored to symmetric matrices was given in the book of Householder [22].

[1] BiCGStab amounts to a reformulation of the ideas behind classical IDR.

1.2. Motivation. There are four reasons to consider eigenvalue computations based on IDR. The obvious reason is to showcase that IDR can be used to compute eigenvalues, so without the need for the transpose of $\mathbf{A}$ and using recurrences of length s. The second reason is that the link worked out in [53] between quantities defined in Krylov subspace methods like IDR and interpolation at the computed Ritz values enables us to better understand the convergence of IDR in theory, as well as in finite precision. A third reason is the idea to enhance existing IDR algorithms in case of slow convergence by utilizing information about the location of the eigenvalues by, e.g., preconditioning based on deflation. This information on the spectrum of $\mathbf{A}$ may especially be useful in a parallel implementation, see the remarks in [54]. Last but not least, it deepens our understanding of the interrelations between IDR(s), ML(k)BiCGStab, and two-sided (block) Lanczos processes with one right-hand side and several left-hand sides.

For the second and third reason and to simplify the presentation, we only consider IDR(s)ORes, the most basic variant of IDR(s). Alternatively, we could have rewritten the recurrences to better suit eigenvalue computations, e.g., by normalizing the residuals, which are only used as *basis vectors* if we do not compute approximate solutions of linear systems. As IDR(s)ORes is of type OrthoRes, we assume that zero is well separated from the field of values, which implies that $\mathbf{A}$ is not too badly conditioned. For the sake of brevity and clarity we only show *how* to compute approximate eigenpairs; the convergence theory and the error analysis of IDR(s)ORes (and the variant of IDR(s)Eig based on it) will be published separately.

1.3. Notation and preliminaries. We use standard notation. The identity matrix of size s is denoted by $\mathbf{I} = \mathbf{I}_s$, its column vectors by $\mathbf{e}_j$ and its elements by the Kronecker delta δ_{ij}. The vector of the sums of all columns, i.e., the vector of all ones, is denoted by $\mathbf{e}$. The matrix $\mathbf{O} = \mathbf{O}_s$ denotes the zero matrix of size s, the zero column vector of length n is denoted by $\mathbf{o} = \mathbf{o}_n$. The matrix $\mathbf{N}_k$ denotes the nilpotent upshift matrix of size k with elements $\delta_{i,j-1}$. The sizes are omitted if easily deducible from the context. In step $n > s$ of IDR(s)ORes, $\mathbf{R}_{n-s:n}$ denotes the matrix of the last $s+1$ residual vectors,

$$\mathbf{R}_{n-s:n} := \big(\mathbf{r}_{n-s}, \ldots, \mathbf{r}_n\big). \tag{1.5}$$

The matrix of all residual vectors up to step n has $n+1$ columns and is denoted by

$$\mathbf{R}_{n+1} := \big(\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_n\big) = \mathbf{R}_{0:n}. \tag{1.6}$$

The forward and backward difference operators Δ and ∇ are defined by

$$\Delta\mathbf{r}_n := \mathbf{r}_{n+1} - \mathbf{r}_n \quad \text{and} \quad \nabla\mathbf{r}_n := \mathbf{r}_n - \mathbf{r}_{n-1}, \tag{1.7}$$

respectively. These finite difference operators are applied column-wise to matrices. Closely related to the forward and backward difference operators is MATLAB's `diff` operator defined column-wise by

$$\mathbf{c} = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_s \end{pmatrix} \in \mathbb{C}^s \quad \Rightarrow \quad \texttt{diff}(\mathbf{c}) = \begin{pmatrix} \gamma_2 - \gamma_1 \\ \vdots \\ \gamma_s - \gamma_{s-1} \end{pmatrix} \in \mathbb{C}^{s-1}. \tag{1.8}$$

We denote the space of polynomials of degree at most n by $\mathbb{P}_n$, and we let $\mathbb{P}_n^\circ$ be the subset of those polynomials that take at 0 the value 1.

We will repeatedly consider unreduced "extended" Hessenberg matrices that have an extra row at the bottom with a non-zero element only at the end. Let $\mathbf{H}_n \in \mathbb{C}^{n\times n}$ denote an unreduced Hessenberg matrix. Then $\underline{\mathbf{H}}_n \in \mathbb{C}^{(n+1)\times n}$ is used to denote the corresponding unreduced *extended* Hessenberg matrix. Here, most of the extended Hessenberg matrices will have the property that their column sums are zero. We say that these Hessenberg matrices are of OrthoRes-type, since each one defines a

Krylov subspace method of the general OrthoRes form [51]. Extended Hessenberg matrices of OrthoRes-type are denoted by appending a superscript $\circ$, like $\underline{\mathbf{Y}}_n^\circ$, $\underline{\mathbf{S}}_n^\circ$, or $\underline{\mathbf{L}}_n^\circ$. The property that the columns sum to zero is mathematically reflected by $\mathbf{e}^\mathsf{T}\underline{\mathbf{S}}_n^\circ = \mathbf{o}_n^\mathsf{T}$. OrthoRes-type matrices, i.e., unreduced extended Hessenberg matrices scaled such that $\mathbf{e}^\mathsf{T}\underline{\mathbf{S}}_n^\circ = \mathbf{o}_n^\mathsf{T}$, have an LDMT decomposition $\underline{\mathbf{S}}_n^\circ = \underline{\mathbf{E}}_n^\circ \mathbf{D}_n \mathbf{M}_n^\mathsf{H}$ with an extended bidiagonal unit lower triangular matrix $\underline{\mathbf{E}}_n^\circ \in \mathbb{C}^{(n+1)\times n}$ of OrthoRes-type, a diagonal matrix $\mathbf{D}_n \in \mathbb{C}^{n\times n}$, and a Hermitian transposed unit lower triangular matrix $\mathbf{M}_n \in \mathbb{C}^{n\times n}$, where

$$\underline{\mathbf{E}}_n^\circ := \begin{pmatrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{pmatrix} \in \mathbb{Z}^{(n+1)\times n} \subset \mathbb{C}^{(n+1)\times n}, \tag{1.9}$$
$$\mathbf{D}_n := -\mathsf{diag}\,(s_{2,1}^\circ, s_{3,2}^\circ \ldots, s_{n+1,n}^\circ) \in \mathbb{C}^{n\times n}.$$

This has two interesting implications: The determinants of the leading submatrices of $-\mathbf{S}_n^\circ$ are all nonsingular and given by

$$\det\,(-\mathbf{S}_k^\circ) = \prod_{j=1}^{k} s_{j+1,j}^\circ =: s_{1:k}^\circ, \quad 1 \leqslant k \leqslant n, \tag{1.10}$$

and the inverses of $\mathbf{S}_k^\circ$ are highly structured,

$$\mathbf{e}_j^\mathsf{T}(\mathbf{S}_k^\circ)^{-1}\mathbf{e}_j = \mathbf{e}_j^\mathsf{T}(\mathbf{S}_k^\circ)^{-1}\mathbf{e}_i = \mathbf{e}_j^\mathsf{T}(\mathbf{S}_k^\circ)^{-1}\mathbf{e}_1, \quad 1 \leqslant i \leqslant j \leqslant k \leqslant n. \tag{1.11}$$

The last equation is verified by noting that in $(\mathbf{S}_k^\circ)^{-1} = (\mathbf{M}_k^\mathsf{H})^{-1}(\mathbf{D}_k)^{-1}(\mathbf{E}_k^\circ)^{-1}$ the first factor, $(\mathbf{M}_k^\mathsf{H})^{-1}$, is upper triangular, the second, $(\mathbf{D}_k)^{-1}$, is diagonal, and $(\mathbf{E}_k^\circ)^{-1}$ is the lower triangular matrix of ones.

Associated with the residual $\mathbf{r}_n$ there is a residual polynomial $\mathcal{R}_n \in \mathbb{P}_n^\circ$ satisfying $\mathbf{r}_n = \mathcal{R}_n(\mathbf{A})\mathbf{r}_0$, see [41]. Whenever a Krylov space solver is of type OrthoRes, $\mathcal{R}_n$ has exactly degree n and is uniquely determined as long as $\mathcal{K}_{n+1}$ has dimension $n+1$. Moreover, in this case there is a corresponding Hessenberg decomposition[2] (of OrthoRes-type)

$$\mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\,\underline{\mathbf{S}}_n^\circ \tag{1.12}$$

with an extended unreduced Hessenberg matrix $\underline{\mathbf{S}}_n^\circ$ of size $(n+1)\times n$. This formula summarizes the recurrences for the residuals; it is mirrored by the formula

$$z\big(\mathcal{R}_0(z), \ldots, \mathcal{R}_n(z)\big) = \big(\mathcal{R}_0(z), \ldots, \mathcal{R}_{n+1}(z)\big)\,\underline{\mathbf{S}}_n^\circ \tag{1.13}$$

describing the recurrences for the residual polynomials. It is well known, see, e.g., [12, Section 7.11, p. 252, Eqn. (8)], or, the probably earliest reference [33, Erste Abtheilung, IV. Abschnitt, § 154, Seite 361, Gleichung 560)], that $\mathcal{R}_n$ is up to scaling the characteristic polynomial of the $n\times n$ Hessenberg matrix $\mathbf{S}_n^\circ$ and can be expressed as

$$\mathcal{R}_n(z) = \det\,(\mathbf{I}_n - z(\mathbf{S}_n^\circ)^{-1}) = \frac{\det\,(\mathbf{S}_n^\circ - z\mathbf{I}_n)}{\det\,(\mathbf{S}_n^\circ)}. \tag{1.14}$$

This can be verified by expanding $\det\,(\mathbf{S}_n^\circ - z\mathbf{I}_n)$ along its last row. In [53] it was shown that this representation cum grano salis carries over to the finite precision case, i.e., modulo additional polynomial error terms.

[2] We name these relations in honor of Karl Hessenberg. He was to our knowledge the first who considered relations of the type $\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n$ with a special unreduced extended Hessenberg matrix $\underline{\mathbf{H}}_n$, see [21]. Usually the names of Lanczos [23, 24] or Arnoldi [4] are associated with such relations.

Often, the residuals $\mathbf{r}_n$ of a Krylov subspace solver satisfy a Bubnov-Galërkin or a Petrov-Galërkin condition. For example, those of the biconjugate gradient (BiCG) method [24, 11], which is closely related to the original IDR and the IDR(1) methods, are characterized by

$$\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n\,, \qquad \mathbf{r}_n \perp \widetilde{\mathcal{K}}_n := \mathcal{K}_n(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{r}}_0)\,,$$

where $\widetilde{\mathbf{r}}_0 \in \mathbb{C}^N$ is an initial shadow residual that can be chosen nearly arbitrarily. The shadow space $\widetilde{\mathcal{K}}_n := \mathcal{K}_n(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{r}}_0)$ is the nth Krylov subspace generated by the adjoint $\mathbf{A}^{\mathsf{H}}$, i.e., the complex conjugate transpose of $\mathbf{A}$, from $\widetilde{\mathbf{r}}_0$.

In the ML(s)BiCG method[3] that Yeung and Chan [50] proposed as a theoretical tool for deriving their ML(s)BiCGStab method, this shadow space is replaced by a block Krylov space: after sj steps,

$$\mathbf{r}_{sj} \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{sj}\,, \qquad \mathbf{r}_{sj} \perp \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{R}}_0) := \sum_{i=1}^{s} \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{r}}_0^{(i)})\,, \tag{1.15}$$

where $\widetilde{\mathbf{R}}_0 := (\widetilde{\mathbf{r}}_0^{(1)}, \ldots, \widetilde{\mathbf{r}}_0^{(s)})$ is now an initial shadow block residual. For the steps with index $n = sj + \ell$, where $1 < \ell < s$, we have analogously

$$\begin{aligned} &\mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n\,, \\ &\mathbf{r}_n \perp \mathcal{K}_{j;\ell}(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{R}}_0) := \sum_{i=1}^{\ell} \mathcal{K}_{j+1}(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{r}}_0^{(i)}) + \sum_{i=\ell+1}^{s} \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{r}}_0^{(i)})\,. \end{aligned} \tag{1.16}$$

ML(s)BiCG is not exactly a generalization of BiCG, since it does not construct a pair of biorthogonal or block biorthogonal bases for $\mathcal{K}_{sj}$ and $\mathcal{K}_{j;\ell}(\mathbf{A}^{\mathsf{H}}, \widetilde{\mathbf{R}}_0)$, but the nonsymmetric band Lanczos process [3, 13, 14, 15] or the block Lanczos process [6, 25] can be adapted to yield such generalizations of BiCG; see Section 3.3 of Loher [25]. In particular, we could develop a generalization of the BiORes version of BiCG that is directly based on recurrences for residuals satisfying (1.15) and (1.16) — if such residuals exist for all j and ℓ up to a sufficiently large value of $n = sj$, that is, if the method does not break down early. The aforementioned method BiORes(s, 1) is a more general scheme, also based on s left starting vectors (shadow residuals) and only one right starting vector, the initial residual $\mathbf{r}_0$. In BiORes(s, 1) we allow far more flexibility as we still have (1.15), but no longer enforce (1.16).

1.4. Outline. In addition to the IDR residuals $\mathbf{r}_n$ we will encounter in this paper other sets of residuals and the corresponding residual polynomials. We will also use the Hessenberg pencils and Hessenberg matrices associated with the recurrences for both the residual vectors and the residual polynomials. The various quantities will be defined later, but Table 1.1 lists the residuals that play a rôle in this paper together with the corresponding extended Hessenberg pencils and extended Hessenberg matrices, and Table 1.2 lists the residual polynomials together with the corresponding residuals and some relations between them. By abuse of notation the matrix $\mathbf{Q}_{n+1}$ denotes either a generic matrix of column vectors which form a basis of a (rational) Krylov subspace or the special matrix of deflated residuals. In the last row of Table 1.1, D denotes the deflation operator that removes every $(s+1)$th row and column; see Figure 1.1 below.

In case of the purified residuals $\mathbf{w}_n$ we will not have an unreduced Hessenberg decomposition and we will use two different denominations of tantamount importance for the residual polynomials. One denomination, $\mathcal{W}_n$, better reflects the construction of the basis vectors and uniformly uses the column indices, the other, ρ_k versus $\widehat{\rho}_k$, is closer to the polynomial point of view: k is the degree of the polynomial and the 'hat' indicates that there are two different types of polynomials in use. We remark that the polynomials $\mathcal{W}_n := \mathcal{R}_n/\Omega_{\lfloor n/(s+1)\rfloor}$ are obviously residual polynomials, i.e., $\mathcal{W}_n(0) = 1$, since $\Omega_{\lfloor n/(s+1)\rfloor}(0) = 1$ and $\mathcal{R}_n(0) = 1$.

[3] Yeung and Chan [50] call the parameter k, but we are interested in the case where $k = s$.

IDR(s)	residuals	Hessenberg pencil	Hessenberg matrix
original	$\mathbf{R}_{n+1}$	$(\underline{\mathbf{Y}}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$	$\underline{\mathbf{S}}_n^\circ := \underline{\mathbf{Y}}_n^\circ(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}$
purified	$\mathbf{W}_{n+1}$	$(\underline{\mathbf{Y}}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$	does not exist
deflated	$\mathbf{Q}_{n+1}$	$(D(\underline{\mathbf{Y}}_n^\circ\mathbf{G}_n), D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$	$\underline{\mathbf{L}}_n^\circ := D(\underline{\mathbf{Y}}_n^\circ\mathbf{G}_n)(D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))^{-1}$

TABLE 1.1
Sets of residual matrices, extended Hessenberg pencils, and extended Hessenberg matrices.

residual vectors	residual polynomials
$\mathbf{R}_{n+1} = (\mathbf{r}_0, \ldots, \mathbf{r}_n)$	$\mathcal{R}_n(z) := \det(\mathbf{I}_n - z\,(\mathbf{S}_n^\circ)^{-1})$
never defined; not needed	$\Omega_j(z) := \prod_{k=1}^{j}(1-\omega_k z)$, $\Omega_0(z) \equiv 1$
$\mathbf{W}_{n+1} = (\mathbf{w}_0, \ldots, \mathbf{w}_n)$	$\mathcal{W}_n(z) := \det(\mathbf{I}_n - z\mathbf{U}_n\mathbf{D}_\omega^{(n)}\,(\mathbf{Y}_n^\circ)^{-1})$
$\mathbf{Q}_{n+1} = (\mathbf{q}_0, \ldots, \mathbf{q}_n)$	$\rho_n(z) := \det(\mathbf{I}_n - z\,(\mathbf{L}_n^\circ)^{-1})$
$= (\mathbf{w}_0, \ldots, \mathbf{w}_{s-1}, \mathbf{w}_{s+1}, \ldots)$	
residual polynomial relations (including the BIORES$(s,1)$-polynomials)	
$\mathcal{R}_{j(s+1)+k}/\Omega_j = \mathcal{W}_{j(s+1)+k} = \rho_{js+k}, \quad 0 \leqslant k < s$ $\mathcal{R}_{j(s+1)+s}/\Omega_j = \mathcal{W}_{j(s+1)+s} = \widehat{\rho}_{(j+1)s}, \quad 0 \leqslant j < \lfloor (n+1)/(s+1) \rfloor$	

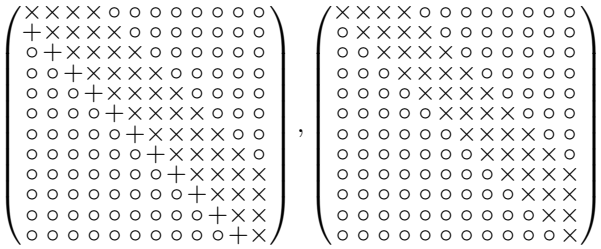
TABLE 1.2
Sets of residual matrices with residual vectors, and corresponding residual polynomials.

The transition from IDR(s)ORES to BIORES$(s,1)$ proceeds in two steps, a *purification step* and a *deflation step*. In the purification step some known roots of the IDR residual polynomial are removed, in the deflation step certain infinite eigenvalues are removed which prevent the existence of a corresponding Hessenberg matrix. In these transitions only certain entries of the Hessenberg pencils are modified. This is depicted in Figure 1.1, which corresponds to a small example, where $s = 3$ and $n = 12$. The third pencil $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ is obtained from the second one, the purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, by a multiplication from the right by the block-Gauß eliminator $\mathbf{G}_n$. For this reason we occasionally refer to it as the *eliminated pencil*. The deflated pencil consists of two block matrices with $s \times s$ blocks[4]. The matrix $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)})$ is block diagonal, the upper triangular diagonal blocks are defined directly in terms of (differences of) IDR(s)ORES quantities. We stress the fact that the matrix $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$ is simultaneously *unreduced Hessenberg* and *block tridiagonal*. The diagonal blocks of $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$ are unreduced Hessenberg matrices, and only the first rows and last columns are altered in the process of purification and deflation.

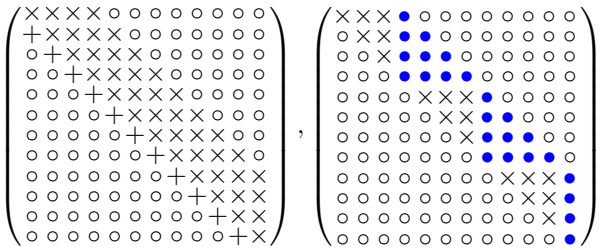
The paper is organized as follows. In Section 2 we define a refinement of the concept of Hessenberg decompositions necessary for the treatment of IDR(s), in Section 3 we consider the application of IDR(s) to eigenvalue computations for the case $s = 1$, in Section 4 we generalize and extend the results for $s = 1$ to the general case $s \geqslant 1$, and in Section 5 we show how to obtain eigenvector approximations and an estimator on the accuracy of the approximate eigenpair. We give some numerical examples in Section 6, and the source codes of our algorithms are listed in Section 7.

[4]The last block column has less columns than s, if n or $n+1$ is not divisible by $s+1$. This is not reflected in this educational example.

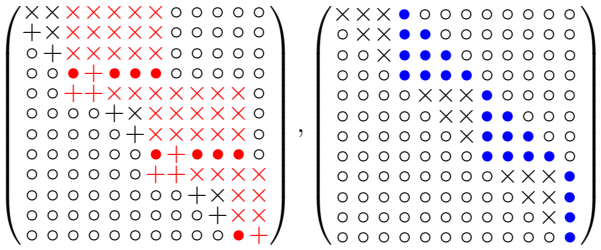
Original IDR(s)ORES pencil, the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$:



Purified IDR(s)ORES pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$:



Deflated purified IDR(s)ORES pencil, elimination step $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$:



Deflated purified IDR(s)ORES pencil, deflation step $(D(\mathbf{Y}_n^\circ\mathbf{G}_n), D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$:

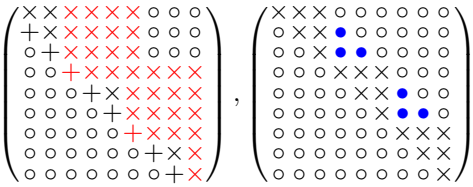


FIG. 1.1. *The transition from* IDR(s)ORES *to* BIORES(s, 1)*, here depicted pencil-wise for* $s = 3$*. Circles and bullets mark zeros, crosses mark entries defined in terms of* IDR(s)ORES *quantities, and plusses mark non-zero values. Black elements are original elements, blue bullet zero elements are those introduced in the purification step, red elements are those altered during deflation.*

2. Generalized Hessenberg decompositions. We define a generalized Hessenberg decomposition[5] to denote a matrix equation of the type

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n, \tag{2.1}$$

with $\mathbf{U}_n$ upper triangular and $\underline{\mathbf{H}}_n$ unreduced extended Hessenberg. When $\mathbf{U}_n = \mathbf{I}_n$, this definition collapses to the definition of a Hessenberg decomposition, e.g., like the one given as an example in Eqn. (1.12). We remark that here $\mathbf{Q}_{n+1} \in \mathbb{C}^{N\times(n+1)}$ denotes a generic matrix of basis vectors.

The matrix equation (2.1) corresponds in the case of full rank of $\mathbf{Q}_{n+1}$ and $\mathbf{U}_n$ to an oblique projection[6] of the pencil $(\mathbf{A},\mathbf{I})$, since

$$\begin{aligned}\widehat{\mathbf{Q}}_n^{\mathsf{H}}(\mathbf{A},\mathbf{I})\mathbf{Q}_n\mathbf{U}_n &= \widehat{\mathbf{Q}}_n^{\mathsf{H}}(\mathbf{A}\mathbf{Q}_n\mathbf{U}_n, \mathbf{Q}_n\mathbf{U}_n)\\ &= \widehat{\mathbf{Q}}_n^{\mathsf{H}}(\mathbf{Q}_{n+1}\underline{\mathbf{H}}_n, \mathbf{Q}_n\mathbf{U}_n) = (\underline{\mathbf{I}}_n^{\mathsf{T}}\underline{\mathbf{H}}_n, \mathbf{U}_n) = (\mathbf{H}_n, \mathbf{U}_n),\end{aligned} \tag{2.2}$$

where $\widehat{\mathbf{Q}}_n^{\mathsf{H}} := \underline{\mathbf{I}}_n^{\mathsf{T}}\mathbf{Q}_{n+1}^{\dagger}$. By definition, $\widehat{\mathbf{Q}}_n^{\mathsf{H}}\mathbf{Q}_{n+1} = \underline{\mathbf{I}}_n^{\mathsf{T}}$, i.e., $\widehat{\mathbf{Q}}_n^{\mathsf{H}}\mathbf{Q}_n = \mathbf{I}_n$ and $\widehat{\mathbf{Q}}_n^{\mathsf{H}}\mathbf{q}_{n+1} = \mathbf{o}_n$. If the upper triangular matrix $\mathbf{U}_n$ is singular, we will refer to this as a *singular projection*.

In case of IDR(s)ORes the matrix $\mathbf{Q}_n$ is the previously defined $\mathbf{R}_n$, the matrix of all residual vectors. As IDR(s)ORes is a Krylov method, these can be characterized using residual polynomials. We need another expression for the residual polynomials solely based on the matrices $\mathbf{U}_n$, $\underline{\mathbf{H}}_n$ defining the decomposition (2.1). To achieve this, we prove that the columns of matrices $\mathbf{Q}_n$ satisfying a generalized Hessenberg decomposition (2.1) can be described with the aid of determinants of leading principal submatrices of these pencils. The proof is based on a slight generalization of the proofs of [52, Lemma 3.1][7], see also [9, 10] and [12, Section 7.11], and [53, Theorem 2.1], where the case $\mathbf{U}_n = \mathbf{I}_n$ has been treated. We omit the index n for simplicity and consider an unreduced Hessenberg/upper triangular pencil ${}^z\mathbf{H} := z\mathbf{U} - \mathbf{H}$ with unreduced Hessenberg matrix $\mathbf{H} \in \mathbb{C}^{n\times n}$ and upper triangular matrix $\mathbf{U} \in \mathbb{C}^{n\times n}$. Similar to [52, Eqn. (3.1), p. 595] and [53, Eqn. (1.9), p. 410] we define the scalars $h_{i:j} := \prod_{\ell=i}^{j} h_{\ell+1,\ell}$ and polynomial vectors $\boldsymbol{\nu}(z)$ and $\check{\boldsymbol{\nu}}(z)$: Let ${}^z\mathbf{H}_{i:j}$ denote the principal submatrix of ${}^z\mathbf{H}$ consisting of the elements indexed by rows and columns i to j and define $\chi_{i:j}(z)$ by

$$\chi_{i:j}(z) := \begin{cases} \det\left({}^z\mathbf{H}_{i:j}\right), & 1 \leqslant i \leqslant j \leqslant n,\\ 1, & i-1 = j. \end{cases} \tag{2.3}$$

We define $\boldsymbol{\nu}(z)$ and $\check{\boldsymbol{\nu}}(z)$ by

$$\boldsymbol{\nu}(z) := \left(\frac{\chi_{i+1:n}(z)}{h_{i:n-1}}\right)_{i=1}^{n} \quad \text{and} \quad \check{\boldsymbol{\nu}}(z) := \left(\frac{\chi_{1:j-1}(z)}{h_{1:j-1}}\right)_{j=1}^{n}, \tag{2.4}$$

with the usual convention that the empty product is one.

First we generalize [52, Lemma 3.1] to unreduced Hessenberg/upper triangular pencils as already indicated in [52, Section 5, p. 605]. This proves amongst others that if θ is an eigenvalue of the pencil ${}^z\mathbf{H}$, then the vectors $\boldsymbol{\nu}(\theta)$ and $\check{\boldsymbol{\nu}}(\theta)$ are right and left eigenvectors, respectively.

Lemma 2.1. *Let $\mathbf{H} \in \mathbb{C}^{n\times n}$ be unreduced Hessenberg and $\mathbf{U} \in \mathbb{C}^{n\times n}$ upper triangular. Denote ${}^z\mathbf{H} = z\mathbf{U} - \mathbf{H}$, and let $\boldsymbol{\nu}(z)$ and $\check{\boldsymbol{\nu}}(z)$ be defined by Eqn. (2.4).*

[5] These generalized Hessenberg decompositions play a dominant rôle in rational Krylov subspace methods [28]; see the treatment in [29]. For this reason we occasionally refer to them as *rational* Hessenberg decompositions.

[6] This is a Petrov-Galërkin approach.

[7] In writing [52] the second author completely missed the technical report [8], which predates [52] by 16 years and already contains almost all the results needed in this paper. Unfortunately, the report by Ericsson has never been published in a journal. We use here the notation of [52].

Define $\chi_n(z) := \mathsf{det}\,({}^z\mathbf{H}) = \mathsf{det}\,(z\mathbf{U} - \mathbf{H})$. *Then*

$$({}^z\mathbf{H})\boldsymbol{\nu}(z) = \mathbf{e}_1 \frac{\chi_n(z)}{h_{1:n-1}}, \quad \check{\boldsymbol{\nu}}(z)^{\mathsf{T}}({}^z\mathbf{H}) = \frac{\chi_n(z)}{h_{1:n-1}}\,\mathbf{e}_n^{\mathsf{T}}. \tag{2.5}$$

Proof. By definition of the adjugate $\mathsf{Ad}(z) := \mathsf{adj}({}^z\mathbf{H})$, the matrix elements $\mathsf{ad}_{ij}(z)$ can be expressed for the lower triangular part ($i \geqslant j$) in terms of cofactors as follows:

$$\begin{aligned}
\mathsf{ad}_{ij}(z) &= (-1)^{i+j} \begin{vmatrix} {}^z\mathbf{H}_{1:j-1} & & \star \\ & \mathbf{N}_{i-j}\,{}^z\mathbf{H}_{j:i-1} & \\ \mathbf{O} & & {}^z\mathbf{H}_{i+1:n} \end{vmatrix} \\
&= \chi_{1:j-1}(z)\, h_{j:i-1}\, \chi_{i+1:n}(z) \\
&= \underbrace{\left(\frac{\chi_{1:j-1}(z)}{h_{1:j-1}}\right)}_{=\check{\nu}_j(z)} h_{1:n-1} \underbrace{\left(\frac{\chi_{i+1:n}(z)}{h_{i:n-1}}\right)}_{=\nu_i(z)} \\
&= h_{1:n-1}\mathbf{e}_i^{\mathsf{T}} \left(\boldsymbol{\nu}(z)\check{\boldsymbol{\nu}}(z)^{\mathsf{T}}\right)\mathbf{e}_j.
\end{aligned} \tag{2.6}$$

This establishes equality between the triangular lower parts of the adjugate $\mathsf{Ad}(z)$ and the outer product representation (2.6). We also know that the adjugate satisfies

$${}^z\mathbf{H}\,\mathsf{Ad}(z) = \mathsf{Ad}(z)\,{}^z\mathbf{H} = \chi_n(z)\,\mathbf{I}. \tag{2.7}$$

Since the first column and the last row are included in the triangular lower part, and since $\check{\nu}_1(z) \equiv \nu_n(z) \equiv 1$,

$$\begin{aligned}
\mathsf{Ad}(z)\mathbf{e}_1 &= \boldsymbol{\nu}(z)\check{\boldsymbol{\nu}}(z)^{\mathsf{T}}\mathbf{e}_1 h_{1:n-1} = \boldsymbol{\nu}(z) h_{1:n-1}, \\
\mathbf{e}_n^{\mathsf{T}}\mathsf{Ad}(z) &= h_{1:n-1}\mathbf{e}_n^{\mathsf{T}}\boldsymbol{\nu}(z)\check{\boldsymbol{\nu}}(z)^{\mathsf{T}} = h_{1:n-1}\check{\boldsymbol{\nu}}(z)^{\mathsf{T}}.
\end{aligned} \tag{2.8}$$

Now, (2.5) follows upon multiplication of (2.7) with $\mathbf{e}_1$ and $\mathbf{e}_n^{\mathsf{T}}$. ∎

Next we generalize [53, Theorem 2.1] to obtain an expression for the columns of $\mathbf{Q}_n$ as polynomials evaluated at $\mathbf{A}$ times the first column $\mathbf{q}_1$.

THEOREM 2.2. *Let the columns of* $\mathbf{Q}_{n+1}$ *be defined by a generalized Hessenberg decomposition* (2.1) *with unreduced extended Hessenberg matrix* $\underline{\mathbf{H}}_n$. *Let* $\chi_n(z)$ *be defined as in Lemma 2.1. Then*

$$\mathbf{q}_{n+1} = \frac{\chi_n(\mathbf{A})}{h_{1:n}}\mathbf{q}_1. \tag{2.9}$$

Proof. To prove (2.9), we start with the generalized Hessenberg decomposition (2.1). First we subtract both sides from the trivial equation $z\mathbf{Q}_n\mathbf{U}_n = z\mathbf{Q}_n\mathbf{U}_n$ to introduce a dependency on the variable z,

$$z\mathbf{Q}_n\mathbf{U}_n - \mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_n(z\mathbf{U}_n - \mathbf{H}_n) - \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^{\mathsf{T}}. \tag{2.10}$$

We multiply Eqn. (2.10) with the vector

$$\boldsymbol{\nu}(z) = \sum_{k=1}^{n} \mathbf{e}_k\,\mathbf{e}_k^{\mathsf{T}}\boldsymbol{\nu}(z) = \sum_{k=1}^{n} \mathbf{e}_k\,\nu_k(z) \tag{2.11}$$

and utilize Eqn. (2.5) and $\nu_n(z) \equiv 1$ to obtain

$$\begin{aligned}
&\sum_{k=1}^{n}\Big(z\mathbf{Q}_n\mathbf{U}_n\mathbf{e}_k\,\nu_k(z) - \mathbf{A}\mathbf{Q}_n\mathbf{U}_n\mathbf{e}_k\,\nu_k(z)\Big) = \\
&\qquad\sum_{k=1}^{n}\Big(z\nu_k(z)\mathbf{Q}_n\mathbf{U}_n\mathbf{e}_k - \mathbf{A}\nu_k(z)\mathbf{Q}_n\mathbf{U}_n\mathbf{e}_k\Big) = \\
&\sum_{k=1}^{n}\Big(z\nu_k(z) - \mathbf{A}\nu_k(z)\Big)\mathbf{Q}_n\mathbf{U}_n\mathbf{e}_k = \mathbf{q}_1\,\frac{\chi_n(z)}{h_{1:n-1}} - \mathbf{q}_{n+1}h_{n+1,n}.
\end{aligned} \tag{2.12}$$

Substituting $\mathbf{A}$ for z in the last line of Eqn. (2.12), which is possible as only *scalar* polynomials occur, gives $\mathbf{o}_n$ on the left-hand side, since the term in parentheses is zero for every k, $k = 1, \ldots, n$. After reformulating the right-hand side of the last line of Eqn. (2.12) we have proven the theorem. □

Thus, whenever we have computed a generalized Hessenberg decomposition with a matrix $\mathbf{Q}_{n+1}$ composed of residual vectors, we can immediately read off the residual polynomials. These are the characteristic polynomials of the unreduced Hessenberg/upper triangular pencil $(\mathbf{H}_n, \mathbf{U}_n)$ scaled by the product of the off-diagonal elements of $\underline{\mathbf{H}}_n$, see Eqn. (1.10) and Eqn. (1.14). The new feature is that when $\mathbf{U}_n$ is singular, the degree of the polynomial is no longer equal to the dimension of the space spanned thus far. We will come back to this point in Section 4. A side effect is that the roots of the residual polynomials can be computed as the eigenvalues of the pencil $(\mathbf{H}_n, \mathbf{U}_n)$.

3. The case $s = 1$. If $s = 1$, the recurrences of IDR(s) in [39, 40] are mathematically equivalent to[8]

$$\begin{aligned} \mathbf{v}_{n-1} &:= (1-\gamma_n)\mathbf{r}_{n-1} + \gamma_n \mathbf{r}_{n-2}\,, & \mathbf{x}'_{n-1} &:= (1-\gamma_n)\mathbf{x}_{n-1} + \gamma_n \mathbf{x}_{n-2}\,, \\ \mathbf{r}_n &:= (\mathbf{I} - \omega_j \mathbf{A})\,\mathbf{v}_{n-1}\,, & \mathbf{x}_n &:= \mathbf{x}'_{n-1} + \omega_j \mathbf{v}_{n-1}\,, \end{aligned} \tag{3.1}$$

where $n > 1$, $j = \lfloor n/2 \rfloor$. Here, γ_n has to be chosen such that $\mathbf{v}_{n-1} \perp \mathbf{p}$, where $\mathbf{p}$ is a basis vector of the one-dimensional orthogonal complement of $\mathcal{S}$, so that $\mathbf{v}_{n-1} \in \mathcal{S}$. If n is even, i.e., $j = n/2$, then ω_j is usually chosen such that $\|\mathbf{r}_n\|$ is minimal, but, basically, any non-zero value is acceptable.

For $n = 1$ we may write

$$\mathbf{r}_1 = (\mathbf{I} - \omega_0 \mathbf{A})\mathbf{r}_0\,.$$

Again ω_0 can be chosen arbitrarily non-zero, e.g. such that $\|\mathbf{r}_1\|$ is minimal.

By induction it is easily seen from (3.1) that $\mathbf{r}_n = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} = \Omega_j(\mathbf{A})\mathbf{w}_n$, where $\Omega_j(z) := (1-\omega_1 z)\cdots(1-\omega_j z)$, and that $\mathbf{w}_n \in \mathcal{K}_{n+1-j} = \mathcal{K}_{n+1-j}(\mathbf{A}, \mathbf{r}_0)$. So, we have

$$\begin{aligned} \mathbf{r}_n &= \Omega_j(\mathbf{A})\mathbf{w}_n = \begin{cases} \Omega_j(\mathbf{A})\rho_j(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j\,, \\ \Omega_j(\mathbf{A})\widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j+1\,, \end{cases} \\ \mathbf{v}_{n-1} &= \Omega_{j-1}(\mathbf{A})\mathbf{w}_n = \begin{cases} \Omega_{j-1}(\mathbf{A})\rho_j(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j\,. \\ \Omega_{j-1}(\mathbf{A})\widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j+1\,. \end{cases} \end{aligned} \tag{3.2}$$

Here ρ_j denotes the jth BiCG residual polynomial, which is the characteristic polynomial of the tridiagonal matrix of the Lanczos process, and is often referred to as a Lanczos polynomial, scaled by $\rho_j(0) = 1$, while $\widehat{\rho}_{j+1}$ denotes another residual polynomial, which has degree $j+1$.

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

$$\begin{aligned} \rho_j(z) &:= (1-\gamma_{2j})\,\widehat{\rho}_j(z) + \gamma_{2j}\,\rho_{j-1}(z) \\ \widehat{\rho}_{j+1}(z) &:= (1-\gamma_{2j+1})\,(1-\omega_j z)\,\rho_j(z) + \gamma_{2j+1}\,\widehat{\rho}_j(z) \end{aligned} \qquad (j = 1, 2, \ldots). \tag{3.3}$$

The initial settings are $\rho_0(z) := 1$ and $\widehat{\rho}_1(z) := (1 - \omega_0 z)$.

In matrix-vector notation, due to $\mathbf{w}_{2j} = \rho_j(\mathbf{A})\mathbf{r}_0$ and $\mathbf{w}_{2j+1} = \widehat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0$, the recurrences (3.3) can alternatively be expressed by

$$\begin{aligned} \mathbf{w}_{2j} &:= (1-\gamma_{2j})\,\mathbf{w}_{2j-1} + \gamma_{2j}\,\mathbf{w}_{2j-2} \\ \mathbf{w}_{2j+1} &:= (1-\gamma_{2j+1})\,(1-\omega_j \mathbf{A})\,\mathbf{w}_{2j} + \gamma_{2j+1}\,\mathbf{w}_{2j-1} \end{aligned} \qquad (j = 1, 2, \ldots). \tag{3.4}$$

[8] Compared to [40] and [20] the index of γ_n has been changed here by 1.

It describes a mixture of a classical Krylov subspace method given by a three-term recurrence and the construction of a new residual based on a weighting process. We can incorporate the latter into the three-term recurrence and thus remove the vectors $\mathbf{w}_{2j+1}$ with odd indices from the recurrence. In the following, this is is described using the language of *polynomials*. The general case $s \geqslant 1$ will be treated in Section 4 using the language of *matrix recurrences*.

Let us rewrite the recurrences (3.3). We let $\kappa_j := \gamma_{2j}$ and $\widehat{\kappa}_j := \gamma_{2j+1}$, and we move the scalars to the right-hand side of the polynomials:

$$\begin{aligned} \rho_j(z) - \rho_{j-1}(z)\,\kappa_j &= \widehat{\rho}_j(z)\,(1-\kappa_j) \\ \widehat{\rho}_{j+1}(z) - \widehat{\rho}_j(z)\,\widehat{\kappa}_j &= \rho_j(z)\,(1-\widehat{\kappa}_j) - z\rho_j(z)\,\omega_j(1-\widehat{\kappa}_j) \end{aligned} \qquad (j = 1, 2, \dots).$$

Next we gather some of the coefficients in a lower and an upper bidiagonal matrix:

$$\underline{\mathbf{B}}_m := \begin{pmatrix} -\kappa_1 & & & \\ 1 & -\kappa_2 & & \\ & \ddots & \ddots & \\ & & 1 & -\kappa_m \\ & & & 1 \end{pmatrix}, \quad \widehat{\mathbf{B}}_m := \begin{pmatrix} 1 & -\widehat{\kappa}_1 & & \\ & 1 & \ddots & \\ & & \ddots & -\widehat{\kappa}_{m-1} \\ & & & 1 \end{pmatrix}.$$

Here, $\underline{\mathbf{B}}_m$ is underlined because it is an $(m+1) \times m$ matrix, i.e., it contains an extra row at the bottom. We further need the three $m \times m$ diagonal matrices

$$\begin{aligned} \mathbf{D}_{\omega;m} &:= \mathsf{diag}\,(\omega_0, \omega_1, \dots, \omega_{m-1})\,, \\ \mathbf{D}_{\kappa;m} &:= \mathsf{diag}\,(1-\kappa_1, 1-\kappa_2, \dots, 1-\kappa_m)\,, \\ \mathbf{D}_{\widehat{\kappa};m} &:= \mathsf{diag}\,(1, 1-\widehat{\kappa}_1, \dots, 1-\widehat{\kappa}_{m-1}) \end{aligned}$$

and the row vectors

$$\boldsymbol{\rho}_m^{\mathsf{T}}(z) := \begin{pmatrix} \rho_0(z) & \cdots & \rho_{m-1}(z) \end{pmatrix}, \qquad \widehat{\boldsymbol{\rho}}_m^{\mathsf{T}}(z) := \begin{pmatrix} \widehat{\rho}_1(z) & \cdots & \widehat{\rho}_m(z) \end{pmatrix}.$$

Then the recurrences (3.3) for $0 \leqslant j < m$ (including the relation $\widehat{\rho}_1(z) = (1 - \omega_0 z)\rho_0(z)$) can be summarized as

$$\begin{aligned} \boldsymbol{\rho}_{m+1}^{\mathsf{T}}(z)\,\underline{\mathbf{B}}_m &= \widehat{\boldsymbol{\rho}}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\kappa;m}\,, \\ z\,\boldsymbol{\rho}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\omega;m}\,\mathbf{D}_{\widehat{\kappa};m} &= \boldsymbol{\rho}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\widehat{\kappa};m} - \widehat{\boldsymbol{\rho}}_m^{\mathsf{T}}(z)\,\widehat{\mathbf{B}}_m\,. \end{aligned} \qquad (3.5)$$

Eliminating $\widehat{\boldsymbol{\rho}}_m^{\mathsf{T}}(z)$ with the help of the first equation leads to

$$z\,\boldsymbol{\rho}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\omega;m}\,\mathbf{D}_{\widehat{\kappa};m} = \boldsymbol{\rho}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\widehat{\kappa};m} - \boldsymbol{\rho}_{m+1}^{\mathsf{T}}(z)\,\underline{\mathbf{B}}_m\,\mathbf{D}_{\kappa;m}^{-1}\,\widehat{\mathbf{B}}_m\,, \qquad (3.6)$$

or,

$$z\,\boldsymbol{\rho}_m^{\mathsf{T}}(z)\,\mathbf{D}_{\omega;m}\,\mathbf{D}_{\widehat{\kappa};m} = \boldsymbol{\rho}_{m+1}^{\mathsf{T}}(z)\left(\begin{pmatrix} \mathbf{D}_{\widehat{\kappa};m} \\ \mathbf{o}_m^{\mathsf{T}} \end{pmatrix} - \underline{\mathbf{B}}_m\,\mathbf{D}_{\kappa;m}^{-1}\,\widehat{\mathbf{B}}_m\right). \qquad (3.7)$$

This is the polynomial form of a Hessenberg relation. The standard form is obtained by multiplying from the right-hand side with the inverse of $\mathbf{D}_{\omega;m}\,\mathbf{D}_{\widehat{\kappa};m}$, followed by inserting $z := \mathbf{A}$ and applying both sides to $\mathbf{r}_0$. The $(m+1) \times m$ matrix

$$\underline{\mathbf{T}}_m := \left(\begin{pmatrix} \mathbf{I}_m \\ \mathbf{o}_m^{\mathsf{T}} \end{pmatrix} - \underline{\mathbf{B}}_m\,\mathbf{D}_{\kappa;m}^{-1}\,\widehat{\mathbf{B}}_m\mathbf{D}_{\widehat{\kappa};m}^{-1}\right)\mathbf{D}_{\omega;m}^{-1} \qquad (3.8)$$

is tridiagonal, and its leading $m \times m$ principal submatrix $\mathbf{T}_m$ can be understood as an oblique projection of $\mathbf{A}$ (an orthogonal one, diagonally scaled such that it is of OrthoRes-type, if $\mathbf{A}$ is Hermitian). Since for n even, $\mathbf{w}_n = \mathbf{w}_{2j} = \rho_j(\mathbf{A})\mathbf{r}_0$ is known to be a BiCG residual, $\mathbf{T}_m$ must be the same matrix as the one obtained by m steps of BiORes, the ORes-variant of BiCG [19]. The fact that the recurrence is of OrthoRes-type is clearly visible in Eqn. (3.7). Since all residual polynomials have a constant term equal to one (see the recurrences (3.3)), setting $z = 0$ proves that the columns of $\underline{\mathbf{T}}_m$ sum to zero.

4. The case $s \geqslant 1$. The relations of importance for eigenvalue computations in IDR(s)ORes are sketched in Algorithm 1. We have omitted the rules for computing the scalars ω_j and the approximate solutions $\mathbf{x}_n$ of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Every rule of computation of the vectors $\mathbf{c}_n$ and the scalars ω_j defines one particular instance of a corresponding IDR algorithm from the family of IDR methods.

```
   input  : A, b, x_0, s, P
   output: R_{n+1}, c_{s+1}, c_{s+2}, ..., ω_1, ω_2, ...
 1 r_0 = b − A x_0
 2 compute R_{s+1} = R_{0:s} = (r_0, ..., r_s) using, e.g., OrthoRes
 3 ∇R_{1:s} = (∇r_1, ..., ∇r_s)
 4 n ← s + 1, j ← 1
 5 while not converged do
 6 |   c_n = (P^H ∇R_{n−s:n−1})^{−1} P^H r_{n−1}
 7 |   v_{n−1} = r_{n−1} − ∇R_{n−s:n−1} c_n
 8 |   compute ω_j
 9 |   ∇r_n = −∇R_{n−s:n−1} c_n − ω_j A v_{n−1}
10 |   r_n = r_{n−1} + ∇r_n
11 |   n ← n + 1
12 |   ∇R_{n−s:n−1} = (∇r_{n−s}, ..., ∇r_{n−1})
13 |   for k = 1, ..., s do
14 |   |   c_n = (P^H ∇R_{n−s:n−1})^{−1} P^H r_{n−1}
15 |   |   v_{n−1} = r_{n−1} − ∇R_{n−s:n−1} c_n
16 |   |   ∇r_n = −∇R_{n−s:n−1} c_n − ω_j A v_{n−1}
17 |   |   r_n = r_{n−1} + ∇r_n
18 |   |   n ← n + 1
19 |   |   ∇R_{n−s:n−1} = (∇r_{n−s}, ..., ∇r_{n−1})
20 |   end
21 |   j ← j + 1
22 end
```

Algorithm 1: IDR(s)ORes

A few remarks are in order. The original IDR(s)ORes used in line 2 OrthoRes(1) to compute the residuals $\mathbf{r}_1$ to $\mathbf{r}_s$. Since we can use other Krylov subspace methods, as long as they correspond to a Hessenberg decomposition, we advocate the use of full OrthoRes [51] or GMRes [31]. They do not require extra memory space compared to the later IDR steps. Note that lines 14–19 differ from lines 6–12 only in that no new value of ω_j is defined. We have shifted the index of the residuals and residual differences by -1 compared to [39, 40]. The forward difference operator Δ in the original algorithm has been replaced by the backward difference operator ∇, as this appears to be more natural and makes the notation slightly easier to understand. Thus, in line 4 of Algorithm 1, we initialize n to be $s+1$ in place of s. It turns out that this shifted index n determines the nth column of certain Hessenberg and banded matrices to be introduced later on. There exist several alternative, but mathematically equivalent ways to update the approximate solutions $\mathbf{x}_n$ and the corresponding residuals $\mathbf{r}_n$ [40]. For the numerical experiments we have implemented the original algorithm of [39, 40], while the mathematically equivalent Algorithm 1 is used throughout in our analysis.

We need to access the vectors $\mathbf{v}_{n-1}$, $\mathbf{c}_n$ and the scalars ω_j in the eigenvalue computations. Thus, we have appended indices $n-1$ and n to $\mathbf{v}_{n-1}$ and $\mathbf{c}_n$, respectively, and introduced the index j of ω_j not present in [39, 40]. Obviously, j is given by

$$j = \left\lfloor \frac{n}{s+1} \right\rfloor$$

and remains constant for every $s+1$ steps. The vectors $\mathbf{c}_n$ have the elements

$$\mathbf{c}_n = \begin{pmatrix} \gamma_1^{(n)} & \dots & \gamma_s^{(n)} \end{pmatrix}^{\mathsf{T}}; \tag{4.1}$$

additionally we define $\gamma_0^{(n)} := 0$, $\gamma_{s+1}^{(n)} := 0$ for all $n \in \mathbb{N}$. We stress the fact that the ordering of the elements in $\mathbf{c}_n$ has changed compared to the IDR algorithm in [40, Figure 3.1] and does also not correspond to the order of elements in the prototype M-File implementation, ibid.

We remark that for reasons of matrix-vector multiplication economy, line 16 of Algorithm 1 cannot be found 'as such' in [39, 40]. In the original IDR(s)ORes the update was given by $\nabla\mathbf{x}_n = -\nabla\mathbf{X}_{n-s:n-1}\mathbf{c}_n + \omega_j\mathbf{v}_{n-1}$ followed by $\nabla\mathbf{r}_n = -\mathbf{A}\nabla\mathbf{x}_n$. As we removed in our variant the recurrences for the iterates from the original IDR(s)-ORes algorithm, we update $\mathbf{r}_n$ in Algorithm 1 according to

$$\nabla\mathbf{r}_n = -\mathbf{A}\nabla\mathbf{x}_n = \mathbf{A}\nabla\mathbf{X}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1} = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}\,. \tag{4.2}$$

The last equality sign in (4.2) is justified in exact arithmetic because in this case we can ensure that for all $n - s \leqslant k < n$ indeed

$$-\nabla\mathbf{r}_k = -\mathbf{r}_k + \mathbf{r}_{k-1} = -\mathbf{b} + \mathbf{A}\mathbf{x}_k + \mathbf{b} - \mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}\nabla\mathbf{x}_k\,. \tag{4.3}$$

In finite precision the gap between the negative *computed* residual differences $-\nabla\mathbf{r}_k$ and the *computed* differences $\nabla\mathbf{x}_k$ of the iterates multiplied by $\mathbf{A}$ has to be monitored. It indicates whether we can still trust the eigenpair approximations obtained using IDR.

4.1. The original IDR Hessenberg recurrence. We reformulate the recurrences slightly by expressing everything in terms of residual vectors instead of residual differences, which amounts to a discrete partial integration, i.e., application of Abel's summation formula,

$$\begin{aligned} \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n &= \sum_{j=0}^{s-1}(\mathbf{r}_{n-s+j} - \mathbf{r}_{n-s+j-1})\gamma_{j+1}^{(n)} \\ &= \sum_{j=0}^{s}\mathbf{r}_{n-s+j-1}(\gamma_j^{(n)} - \gamma_{j+1}^{(n)}) = -\mathbf{R}_{n-s-1:n-1}\,\mathtt{diff}\begin{pmatrix}0\\ \mathbf{c}_n\\ 0\end{pmatrix}. \end{aligned} \tag{4.4}$$

Eliminating the residual differences, we observe that in all inner steps we have the residual recurrence

$$\begin{aligned} \mathbf{r}_n &= (\mathbf{I} - \omega_j\mathbf{A})\mathbf{v}_{n-1} = (\mathbf{I} - \omega_j\mathbf{A})(\mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n) \\ &= (\mathbf{I} - \omega_j\mathbf{A})\left(\mathbf{r}_{n-1} + \mathbf{R}_{n-s-1:n-1}\,\mathtt{diff}\begin{pmatrix}0\\ \mathbf{c}_n\\ 0\end{pmatrix}\right) \\ &= (\mathbf{I} - \omega_j\mathbf{A})\,\mathbf{R}_{n-s-1:n-1}\,\mathtt{diff}\begin{pmatrix}0\\ \mathbf{c}_n\\ 1\end{pmatrix} =: (\mathbf{I} - \omega_j\mathbf{A})\,\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n, \end{aligned} \tag{4.5}$$

as in the left set of equations in the recurrence (3.1) for the case $s = 1$.

Classical Krylov subspace theory would embark upon the construction of a Hessenberg decomposition with the residuals as a "basis". The leading submatrices of the constructed Hessenberg matrix are oblique "projections" of the original matrix, thus we could use the eigenvalues of these submatrices as approximations of eigenvalues, the so-called Ritz values, and the prolongations of the Hessenberg eigenvectors, the so-called Ritz vectors, would serve as corresponding approximate eigenvectors. Obviously, the matrix of residuals will at some point be rank-deficient, as IDR terminates usually when the number of steps is larger than the dimension of $\mathbf{A}$. Nevertheless, the specially structured Hessenberg decomposition related to IDR will give good approximations to eigenvalues.

To demonstrate the general approach, we first derive the generalized Hessenberg decomposition for the IDR residuals.

THEOREM 4.1 (The original Hessenberg/upper triangular IDR(s)ORES pencil). *The generalized Hessenberg decomposition for the* IDR(s)ORES *residuals is given by*

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega^{(n)} = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ, \tag{4.6}$$

where $\mathbf{R}_{n+1} = \begin{pmatrix}\mathbf{r}_0,\ldots,\mathbf{r}_n\end{pmatrix}$ *is the matrix of all residual vectors up to step* n. *For* $s < k \leqslant n$, *the* k*th columns of the upper triangular matrix* $\mathbf{Y}_n \in \mathbb{C}^{n\times n}$ *and of the extended unreduced Hessenberg matrix* $\underline{\mathbf{Y}}_n^\circ \in \mathbb{C}^{(n+1)\times n}$ *of* ORTHORES*-type are defined by*

$$\mathbf{Y}_n\mathbf{e}_k := \begin{pmatrix}\mathbf{o}_{k-(s+1)}\\ \mathbf{y}_k\\ \mathbf{o}_{n-k}\end{pmatrix}, \qquad \underline{\mathbf{Y}}_n^\circ\mathbf{e}_k := \begin{pmatrix}\mathbf{o}_{k-(s+1)}\\ \mathbf{y}_k\\ -1\\ \mathbf{o}_{n-k}\end{pmatrix}, \qquad \text{where} \quad \mathbf{y}_k := \begin{pmatrix}\gamma_1^{(k)}\\ \gamma_2^{(k)}-\gamma_1^{(k)}\\ \vdots\\ \gamma_s^{(k)}-\gamma_{s-1}^{(k)}\\ 1-\gamma_s^{(k)}\end{pmatrix} \in \mathbb{C}^{s+1}, \tag{4.7}$$

while the diagonal elements $\mathbf{e}_k^\mathsf{T}\mathbf{D}_\omega^{(n)}\mathbf{e}_k$ *of the diagonal matrix* $\mathbf{D}_\omega^{(n)}$ *are defined by*

$$\mathbf{e}_k^\mathsf{T}\mathbf{D}_\omega^{(n)}\mathbf{e}_k := \omega_j, \quad j = \left\lfloor \frac{k}{s+1} \right\rfloor. \tag{4.8}$$

The leading portions of the matrices $\mathbf{Y}_n$, $\underline{\mathbf{Y}}_n^\circ$ *and* $\mathbf{D}_\omega^{(n)}$ *are given by the Hessenberg decomposition of the starting procedure chosen.*

Proof. We sort terms in Eqn. (4.5) according to the occurrence of the matrix $\mathbf{A}$ and obtain

$$\omega_j\mathbf{A}\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n - \mathbf{r}_n = \mathbf{R}_{n-s-1:n}\begin{pmatrix}\mathbf{y}_n\\ -1\end{pmatrix}, \tag{4.9}$$

which is the nth column

$$\omega_j\mathbf{A}\mathbf{R}_n\begin{pmatrix}\mathbf{o}_{n-(s+1)}\\ \mathbf{y}_n\end{pmatrix} = \mathbf{R}_{n+1}\begin{pmatrix}\mathbf{o}_{n-(s+1)}\\ \mathbf{y}_n\\ -1\end{pmatrix} \tag{4.10}$$

of the generalized Hessenberg decomposition (4.6). ∎

REMARK 4.2. In case of a Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_s = \mathbf{R}_{s+1}\underline{\mathbf{C}}_s^\circ \tag{4.11}$$

of a starting procedure of ORTHORES-type ($\mathbf{e}^\mathsf{T}\underline{\mathbf{C}}_s^\circ = \mathbf{o}_s^\mathsf{T}$) we set $\mathbf{Y}_s := \mathbf{I}_s$, $\mathbf{D}_\omega^{(s)} := \mathbf{I}_s$, and $\underline{\mathbf{Y}}_s^\circ := \underline{\mathbf{C}}_s^\circ$.

REMARK 4.3. In the original IDR(s)ORES variant a truncated ORTHORES (i.e., ORTHORES(1) $\approx$ ORES) was used for the first s steps. In this setting, we would define $\mathbf{Y}_s := \mathbf{I}_s$, $\mathbf{D}_\omega^{(s)} := \mathsf{diag}\,(\tilde{\omega}_1,\ldots,\tilde{\omega}_s)$, where $\tilde{\omega}_k$ is defined by the residual minimization in step k, and $\underline{\mathbf{Y}}_s^\circ := \underline{\mathbf{E}}_s^\circ$, where $\underline{\mathbf{E}}_s^\circ$ is defined in Eqn. (1.9).

DEFINITION 4.4 (Sonneveld pencil). *In honor of Peter Sonneveld, who developed with the classical IDR variant [47] in 1979 not only the first Lanczos-type product method (LTPM), but at the same time introduced an interesting complementary point of view on LTPMs, we name the banded Hessenberg/upper triangular pencil* $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$ *the* Sonneveld pencil.

REMARK 4.5. The Hessenberg/upper triangular Sonneveld pencil is banded with upper bandwidth s. This reflects that IDR(s)ORES is a $(s+1, s+1)$–step method

[16]. Those elements of the upper triangular parts of $\mathbf{Y}_n^\circ$ and $\mathbf{Y}_n$ that are given by IDR(s)ORes are identical and defined in terms of differences of the elements of the vectors $\mathbf{c}_k$. The matrix $\mathbf{D}_\omega^{(n)}$ is uniquely defined by n, s and ω_j, $1 \leqslant j \leqslant \lfloor n/(s+1) \rfloor$.

REMARK 4.6. Apart from the leading $(s+1) \times s$ block defined by the extended Hessenberg matrix $\underline{\mathbf{C}}_s^\circ$ from the starting Hessenberg decomposition, we only need to store all vectors $\mathbf{c}_k$, $s < k \leqslant n$. In our implementation we use

$$\mathbf{C}_{n-s-1} := \begin{pmatrix} \mathbf{c}_{s+1} & \cdots & \mathbf{c}_n \end{pmatrix} \in \mathbb{C}^{s \times (n-s-1)}, \tag{4.12}$$

and one vector containing all the ω_j, $1 \leqslant j \leqslant \lfloor n/(s+1) \rfloor$. The elements in the IDR part of the bands of $\underline{\mathbf{Y}}_n^\circ$ and of $\mathbf{Y}_n$ are then given by

$$\texttt{diff}\begin{pmatrix} \mathbf{o}^\mathsf{T} \\ \mathbf{C}_{n-s-1} \\ \mathbf{e}^\mathsf{T} \\ \mathbf{o}^\mathsf{T} \end{pmatrix} \quad \text{and} \quad \texttt{diff}\begin{pmatrix} \mathbf{o}^\mathsf{T} \\ \mathbf{C}_{n-s-1} \\ \mathbf{e}^\mathsf{T} \end{pmatrix}, \quad \text{respectively.} \tag{4.13}$$

As we are going to see next, the generalized Hessenberg decomposition (4.6) can be turned into a Hessenberg decomposition.

COROLLARY 4.7. *Suppose that the starting Hessenberg decomposition can be written as* $\mathbf{A}\mathbf{R}_s = \mathbf{R}_{s+1}\underline{\mathbf{C}}_s^\circ$. *Suppose further that* $\omega_j \neq 0$, $1 \leqslant j \leqslant \lfloor n/(s+1) \rfloor$, *and that* $\gamma_s^{(k)} \neq 1$, $s < k \leqslant n$. *Then the Hessenberg decomposition for the* IDR(s)ORes *residuals is given by*

$$\mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\underline{\mathbf{S}}_n^\circ, \tag{4.14}$$

where the Sonneveld matrix

$$\underline{\mathbf{S}}_n^\circ := \underline{\mathbf{Y}}_n^\circ (\mathbf{D}_\omega^{(n)})^{-1} \mathbf{Y}_n^{-1} \tag{4.15}$$

is an unreduced extended Hessenberg matrix of OrthoRes*-type.*

Proof. With this assumption on the starting Hessenberg decomposition, the diagonal matrix $\mathbf{D}_\omega^{(n)}$ is invertible if and only if $\omega_j \neq 0$, $1 \leqslant j \leqslant \lfloor n/(s+1) \rfloor$, and the upper triangular matrix $\mathbf{Y}_n$ is invertible if and only if $\gamma_s^{(k)} \neq 1$, $s+1 \leqslant k \leqslant n$. ∎

The Hessenberg matrix $\mathbf{S}_n^\circ$ can be used to compute eigenvalues and eigenvectors, which are prolonged by the residual matrix $\mathbf{R}_n$, to obtain Ritz values and Ritz vectors, respectively. The Ritz values, i.e., the eigenvalues of the Sonneveld matrix $\mathbf{S}_n^\circ$, are the roots of the residual polynomials $\mathcal{R}_n(z) := \mathsf{det}\,(\mathbf{I}_n - z(\mathbf{S}_n^\circ)^{-1})$. The Ritz vectors are the prolonged eigenvectors of the Sonneveld matrices; these will be discussed separately in Section 5. The Sonneveld matrix, a typically full Hessenberg matrix, is not formed explicitly; instead we apply the QZ algorithm to the banded Hessenberg/upper triangular Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$.

However, this direct way of computing eigenvalue approximations is not always the method of choice in Lanczos-type product methods (LTPM). Indeed, we *know* a portion of the roots of the residual polynomials, namely the roots $1/\omega_j$. These are among the eigenvalues of the Sonneveld matrix $\mathbf{S}_n^\circ$, as is captured by Figure 4.1, obtained by invoking IDR(5)ORes for a real matrix $\mathbf{A}$ of dimension 20 for 4 sweeps.

4.2. The purified IDR Hessenberg recurrence. The division of the residual polynomials $\mathcal{R}_k$ by the known linear factors, i.e., in step k by the residual polynomial factor

$$\Omega_j(z) = \prod_{i=1}^{j}(1 - \omega_i z), \quad j = \left\lfloor \frac{k}{s+1} \right\rfloor, \tag{4.16}$$

results in polynomial recurrences which correspond to a generalized Hessenberg decomposition with an unreduced Hessenberg/upper triangular pencil that has only the

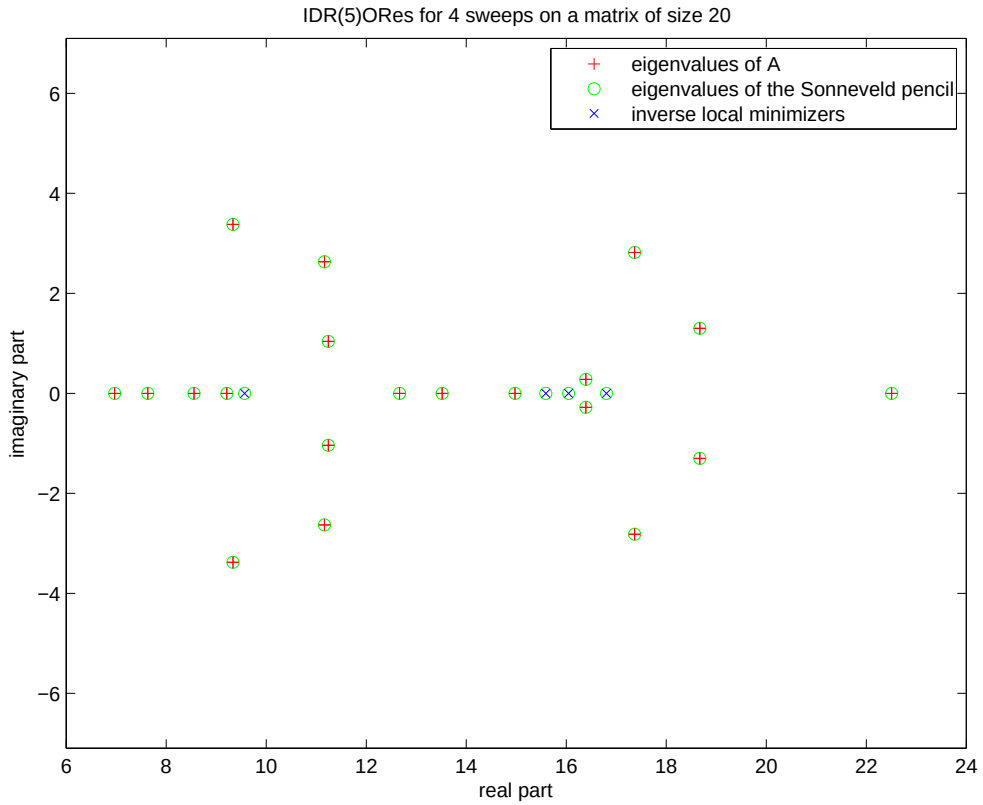


FIG. 4.1. *Approximation by Ritz values using the Sonneveld pencil. We used* IDR(s)ORES *with* $s = 5$ *for* 4 *sweeps. The* ω_j *were computed based on local minimization. The red plusses depict the eigenvalues of* $\mathbf{A}$*, the blue crosses depict the roots* $1/\omega_j$*,* $j = 1, \ldots, 4$*, and the green circles depict the eigenvalues of the Sonneveld pencil of size* $24 = 4 \cdot (5+1)$ *resulting from* IDR(s)ORES.

unknown residual polynomial roots as eigenvalues, and, additionally, some infinite eigenvalues.

To proceed, we rewrite the residual recurrence. We already have proven, see Eqn. (4.5), that the recurrences for $s \geqslant 1$ and $n > s$ are mathematically equivalent to

$$\begin{aligned} \mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ \mathbf{r}_n &:= (\mathbf{I} - \omega_j\mathbf{A})\,\mathbf{v}_{n-1}\,, \end{aligned} \tag{4.17}$$

where j is given by

$$j = \left\lfloor \frac{n}{s+1} \right\rfloor$$

and remains constant for every inner sweep of $s+1$ steps. We note that $k = n - j(s+1)$, which cycles between 0 and s, is the running index of the inner sweep.[9]

We consider a general starting procedure (e.g., GMRES, GCR $\approx$ ORTHORES) that produces the s residuals $\mathbf{r}_1, \ldots, \mathbf{r}_s$. This information uniquely defines the first s

[9] These indices j and k can be described by the so-called "index functions" g and r introduced by Yeung and Boley in [49]; see [48, page 3]: in their notation $j = g_{s+1}(n+1)$ and $k = r_{s+1}(n+1) - 1$.

residual polynomials we are interested in. We have that

$$\begin{aligned}
&\mathbf{r}_0 && \text{is initially given by } \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,\\
&\left.\begin{matrix}\mathbf{r}_1\\ \vdots\\ \mathbf{r}_s\end{matrix}\right\} && \text{is constructed by the starting procedure,}\\
&\mathbf{r}_{s+1} && \text{is based on the construction of } \omega_1,\\
&\left.\begin{matrix}\mathbf{r}_{s+2}\\ \vdots\\ \mathbf{r}_{2s+1}\end{matrix}\right\} && \text{is constructed using } \omega_1,\\
&\mathbf{r}_{2(s+1)} && \text{is based on the construction of } \omega_2,\\
&\left.\begin{matrix}\mathbf{r}_{2s+3}\\ \vdots\\ \mathbf{r}_{3s+2}\end{matrix}\right\} && \text{is constructed using } \omega_2,
\end{aligned} \tag{4.18a}$$

and so forth. In summary, for all $j \geqslant 1$,

$$\begin{aligned}
&\mathbf{r}_{j(s+1)+0} && \text{is based on the construction of } \omega_j,\\
&\left.\begin{matrix}\mathbf{r}_{j(s+1)+1}\\ \vdots\\ \mathbf{r}_{j(s+1)+s}\end{matrix}\right\} && \text{is constructed using } \omega_j.
\end{aligned} \tag{4.18b}$$

We define $\mathcal{S} := \mathcal{P}^{\perp}$ as the orthogonal complement of the range $\mathcal{P}$ of the columns of $\mathbf{P}$. We observe that every residual $\mathbf{r}_n$ with index n in the range $j(s+1)+k$, $0 \leqslant k \leqslant s$, is obtained from previous information by a multiplication of $\mathbf{v}_{n-1}$ with the linear mapping

$$(\mathbf{I} - \omega_j \mathbf{A}) : \mathcal{G}_{j-1} \cap \mathcal{S} \to \mathcal{G}_j, \tag{4.19}$$

that defines the jth Sonneveld space $\mathcal{G}_j$; see (1.4). Recall that $\mathcal{G}_0 = \mathcal{K}_N(\mathbf{A}, \mathbf{r}_0) \subset \mathbb{C}^N$ is the *full* Krylov subspace generated by $\mathbf{A}$ with starting vector $\mathbf{r}_0$. We use the fact pointed out in [40, page 1046] that all residuals constructed are of the form

$$\mathbf{r}_n = \Omega_j(\mathbf{A})\mathbf{w}_n, \quad \mathbf{v}_{n-1} = \Omega_{j-1}(\mathbf{A})\mathbf{w}_n, \quad j = \left\lfloor \frac{n}{s+1} \right\rfloor, \tag{4.20}$$

where the vectors $\mathbf{w}_n \in \mathcal{K}_{n-j}(\mathbf{A}, \mathbf{r}_0)$ can be interpreted as residuals too, since $\Omega_j(0) = 1$. We call these vectors $\mathbf{w}_n$ the *purified residuals*. The residual polynomial $\mathcal{W}_n$ corresponding to $\mathbf{w}_n$, which is of degree $n-j$, satisfies $\mathcal{W}_n(z) = \mathcal{R}_n(z)/\Omega_j(z)$. Since $\Omega_0(z) \equiv 1$, the first $s+1$ residuals $\mathbf{r}_j$ and $\mathbf{w}_j$, $0 \leqslant j \leqslant s$, coincide; especially $\mathbf{w}_0 = \mathbf{r}_0$. Utilizing these connections we now obtain recurrences for these residual polynomials and for the vectors $\mathbf{w}_n$. For $n = j(s+1)+k$, $0 \leqslant k \leqslant s$, we know that for two families of polynomials denoted by ρ_{sj+k}, $0 \leqslant k < s$, and $\widehat{\rho}_{(j+1)s}$, the latter corresponding to the missing case $k = s$, the following holds true:

$$\mathbf{w}_n = \mathcal{W}_n(\mathbf{A})\mathbf{r}_0 = \mathcal{W}_{j(s+1)+k}(\mathbf{A})\mathbf{r}_0 =: \begin{cases} \rho_{js+k}(\mathbf{A})\mathbf{w}_0, & 0 \leqslant k < s, \\ \widehat{\rho}_{(j+1)s}(\mathbf{A})\mathbf{w}_0, & k = s, \end{cases} \tag{4.21}$$

i.e.,

$$\begin{aligned}
\mathbf{r}_n \;=\; \Omega_j(\mathbf{A})\mathbf{w}_n \;&=\; \begin{cases} \Omega_j(\mathbf{A})\rho_{js+k}(\mathbf{A})\mathbf{w}_0, & 0 \leqslant k < s, \\ \Omega_j(\mathbf{A})\widehat{\rho}_{(j+1)s}(\mathbf{A})\mathbf{w}_0, & k = s, \end{cases}\\
\mathbf{v}_{n-1} = \Omega_{j-1}(\mathbf{A})\mathbf{w}_n &= \begin{cases} \Omega_{j-1}(\mathbf{A})\rho_{js+k}(\mathbf{A})\mathbf{w}_0, & 0 \leqslant k < s, \\ \Omega_{j-1}(\mathbf{A})\widehat{\rho}_{(j+1)s}(\mathbf{A})\mathbf{w}_0, & k = s, \end{cases}
\end{aligned} \tag{4.22}$$

In the case $s = 1$ of Section 3, the polynomials ρ_{sj+0} correspond to the BiCG polynomials ρ_j and the polynomials $\widehat{\rho}_{(j+1)s}$ correspond to the polynomials $\widehat{\rho}_{j+1}$. We

remark that the two consecutive polynomials $\mathcal{W}_{(j+1)(s+1)-1} = \mathcal{W}_{j(s+1)+s} = \widehat{\rho}_{(j+1)s}$ and $\mathcal{W}_{(j+1)(s+1)} = \rho_{(j+1)s+0}$ have the same degree $(j+1)s$.

We insert the polynomial expressions (4.22) into the second line of the relations (4.17), so that

$$\mathbf{v}_{n-1} = (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \tag{4.23}$$

and divide by $\Omega_{j-1}(\mathbf{A})$, which is a common factor of all polynomials involved, to obtain for n divisible by $s+1$, i.e., $n = j(s+1)$,

$$\mathbf{w}_{n+0} = \sum_{\ell=0}^{s} \mathbf{w}_{n-\ell-1}\cdot y_{s+1-\ell}^{(n+0)} \tag{4.24a}$$

$$\mathbf{w}_{n+1} = (\mathbf{I}-\omega_j\mathbf{A})\mathbf{w}_n\cdot y_{s+1}^{(n+1)} + \sum_{\ell=1}^{s} \mathbf{w}_{n-\ell}\cdot y_{s+1-\ell}^{(n+1)} \tag{4.24b}$$

$$\mathbf{w}_{n+2} = (\mathbf{I}-\omega_j\mathbf{A})\sum_{\ell=0}^{2-1} \mathbf{w}_{n+2-\ell-1}\cdot y_{s+1-\ell}^{(n+2)} + \sum_{\ell=2}^{s} \mathbf{w}_{n+2-\ell-1}\cdot y_{s+1-\ell}^{(n+2)} \tag{4.24c}$$

$$\mathbf{w}_{n+k} = (\mathbf{I}-\omega_j\mathbf{A})\sum_{\ell=0}^{k-1} \mathbf{w}_{n+k-\ell-1}\cdot y_{s+1-\ell}^{(n+k)} + \sum_{\ell=k}^{s} \mathbf{w}_{n+k-\ell-1}\cdot y_{s+1-\ell}^{(n+k)} \tag{4.24d}$$

$$\mathbf{w}_{n+s} = (\mathbf{I}-\omega_j\mathbf{A})\sum_{\ell=0}^{s-1} \mathbf{w}_{n+s-\ell-1}\cdot y_{s+1-\ell}^{(n+s)} + \mathbf{w}_{n-1}\cdot y_1^{(n+s)}. \tag{4.24e}$$

Here, $y_1^{(n)}$ to $y_{s+1}^{(n)}$ denote the elements of $\mathbf{y}_n \in \mathbb{C}^{s+1}$.

The index $n = j(s+1)$ has been chosen such that all "purified" residuals $\mathbf{w}_{n-\ell-1}$, $0 \leqslant \ell \leqslant s$, have no additional linear factor $(\mathbf{I}-\omega_j\mathbf{A})$. The *purified residual polynomial* corresponding to $\mathbf{w}_n = \mathbf{w}_{j(s+1)}$ is ρ_{js}, which is in theory a ML(s)BiCG polynomial.

Let us consider again the degrees of the polynomials. In the first step (4.24a) of a new cycle, the polynomial of interest is obtained as a linear combination of the previously computed polynomials, which results in a polynomial of the same degree as the last one (provided that $\gamma_s^{(n)} \neq 1$). Every other step increases the degree of the polynomial by one (provided that all $\gamma_s^{(n+k)} \neq 1$, $1 \leqslant k \leqslant s$). This corresponds to a slightly generalized Hessenberg decomposition, where the Hessenberg pencil is "reduced" at every $(s+1)$th column. The precise meaning of "reduced" becomes obvious in the following theorem.

Theorem 4.8 (The purified Hessenberg/upper triangular IDR(s)Ores pencil). *The generalized Hessenberg decomposition for the* purified IDR(s)Ores *residuals is given by*

$$\mathbf{A}\mathbf{W}_n\mathbf{U}_n\mathbf{D}_\omega^{(n)} = \mathbf{W}_{n+1}\underline{\mathbf{Y}}_n^\circ, \tag{4.25}$$

where

$$\mathbf{W}_{n+1} := \big(\mathbf{w}_0,\ldots,\mathbf{w}_n\big) \tag{4.26}$$

is the matrix of all purified residual vectors up to step n, $\underline{\mathbf{Y}}_n^\circ \in \mathbb{C}^{(n+1)\times n}$ and $\mathbf{D}_\omega^{(n)} \in \mathbb{C}^{n\times n}$ are defined in Theorem 4.1, and the upper triangular $\mathbf{U}_n \in \mathbb{C}^{n\times n}$ is obtained from $\mathbf{Y}_n$ by setting to zero all elements in the lower triangular parts of the square submatrices $(y_{(\ell-1)(s+1)+1+k,\ell(s+1)-1+j}; k,j = 0,\ldots,s), \ell = 1,2,\ldots,$ such that the resulting matrix is block-diagonal with alternating $s\times s$ upper triangular blocks of elements from $\mathbf{Y}_n$ and 1×1 zero blocks; see Figure 1.1 on page 7.

Proof. The general equation (4.24d) can be sorted according to terms involving $\omega_j\mathbf{A}$ and others. For all $0 \leqslant k \leqslant s$ we have

$$\omega_j\mathbf{A}\sum_{\ell=0}^{k-1} \mathbf{w}_{n+k-\ell-1}\cdot y_{s+1-\ell}^{(n+k)} = -\mathbf{w}_{n+k} + \sum_{\ell=0}^{s} \mathbf{w}_{n+k-\ell-1}\cdot y_{s+1-\ell}^{(n+k)}. \tag{4.27}$$

For $n = j(s+1)$, the $(n+k)$th column of the resulting generalized Hessenberg decomposition is

$$\omega_j \mathbf{A}\mathbf{W}_{n+k} \begin{pmatrix} \mathbf{o}_n \\ \mathbf{y}_{n+k}^{s+2-k:s+1} \end{pmatrix} = \mathbf{W}_{n+k+1} \begin{pmatrix} \mathbf{o}_{n+k-s-1} \\ \mathbf{y}_{n+k} \\ -1 \end{pmatrix}, \quad 0 \leqslant k \leqslant s. \tag{4.28}$$

Together with the Hessenberg decomposition defining the starting block of residuals scaled to be of OrthoRes-type,

$$\mathbf{A}\mathbf{R}_s = \mathbf{R}_{s+1}\underline{\mathbf{C}}_s^\circ = \mathbf{R}_s\mathbf{C}_s^\circ - \mathbf{r}_s c_{s+1,s}^\circ \mathbf{e}_s^\mathsf{T}, \tag{4.29}$$

and the observation that $\mathbf{W}_{s+1} = \mathbf{R}_{s+1}$ we obtain Eqn. (4.25). □

Remark 4.9. The generalized Hessenberg decomposition (4.25) corresponds to a *singular* projection, since $\mathbf{U}_n$ is singular. In fact, the rows and columns of index $(s+1)j$ contain only zeros.

Remark 4.10. As the columns of $\underline{\mathbf{Y}}_n^\circ$ sum to zero, (4.25) defines again a method of OrthoRes-type. In the part of $\underline{\mathbf{Y}}_n^\circ$ that does not belong to the starting procedure, the lower co-diagonal consists of minus ones.

Remark 4.11. The transition from the generalized Hessenberg decomposition (4.6) for $\mathbf{R}_n$ to the generalized Hessenberg decomposition (4.25) for $\mathbf{W}_n$ is computationally rather simple: To obtain the only new matrix $\mathbf{U}_n$ we just blanked certain lower triangles in $\mathbf{Y}_n$ and thus cut this upper triangular band matrix into $s \times s$ upper triangular matrices alternating with zeros; compare the first two matrix pairs in Figure 1.1. In fact, the linear combination of $s+1$ terms implicit on the left-hand side of (4.10) has been replaced by a linear combination of k terms, where $0 \leqslant k \leqslant s$, implicit on the left-hand side of (4.27) and (4.28). At this stage no additional round-off errors occur.

We have shown in Section 2 that the residual polynomials are up to the known non-zero factors $\mathsf{det}\,(-\mathbf{Y}_k^\circ) = y_{1:k}^\circ$ the leading subdeterminants of the regular[10] Hessenberg pencil

$${}^z\mathbf{H}_n := z\mathbf{U}_n\mathbf{D}_\omega^{(n)} - \mathbf{Y}_n^\circ = \left(\mathbf{I}_n - z\mathbf{U}_n\mathbf{D}_\omega^{(n)}(\mathbf{Y}_n^\circ)^{-1}\right)(-\mathbf{Y}_n^\circ). \tag{4.30}$$

The elements

$$h_{j+1,j}(z) = -y_{j+1,j}^\circ = \begin{cases} \text{defined by starting procedure} & \text{if} \quad j \leqslant s \\ 1 & \text{if} \quad j > s \end{cases} \tag{4.31}$$

of the lower co-diagonal of ${}^z\mathbf{H}$ are all non-zero and constant and thus the scaling by products of elements from the lower co-diagonal can be neglected as we are only interested in the *roots* of the residual polynomials.

The pencil ${}^z\mathbf{H}_n$ has exactly

$$j = \left\lfloor \frac{n}{s+1} \right\rfloor \tag{4.32}$$

infinite eigenvalues, as every $(s+1)$th diagonal element of the upper triangular matrix $\mathbf{U}_n$ is zero. The resulting determinant, the characteristic polynomial $\chi_n(z) := \mathsf{det}\,({}^z\mathbf{H}_n)$, is of degree $n-j$. The degree of the polynomial remains constant in every step where in $\mathbf{U}_n$ a new zero is introduced in the diagonal, otherwise the degree increases by one, which is in accordance with our observation for the purified residual polynomials.

Remark 4.12. Every $s+1$ steps we have two consecutive purified residual polynomials with the same degree $n-j = js$, $j \geqslant 1$, namely $\mathcal{W}_{j(s+1)-1} = \mathcal{W}_{(j-1)(s+1)+s} =$

[10] The pencil ${}^z\mathbf{H}_n$ is regular, as the determinant of $-\mathbf{Y}_n^\circ$ is given by the product of the non-zero subdiagonal elements of $\mathbf{Y}_n^\circ$, see Eqn. (1.10) and Eqn. (4.31).

$\widehat{\rho}_{js}$ and $\mathcal{W}_{j(s+1)} = \rho_{js}$. This is reflected by the positions of the zeros in the diagonal of $\mathbf{U}_n$.

This shows that we can compute the roots of the purified residual polynomials by computing the eigenvalues of the pencil ${}^z\mathbf{H}_n$, i.e., by application of the QZ algorithm to the banded unreduced upper Hessenberg/upper triangular pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, e.g., by invoking $\texttt{eig}(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ in MATLAB; see Figure 4.2, which shows the results of the computation of the eigenvalues of the regular purified pencil obtained by invoking four sweeps of IDR(5)ORES for the real matrix $\mathbf{A}$ of dimension 20 used already for Figure 4.1. By comparing the eigenvalues of the Sonneveld pencil in Figure 4.1 and the finite eigenvalues of the purified pencil in Figure 4.2 the purification of the residual polynomial from the known roots $1/\omega_j$ becomes clearly visible.

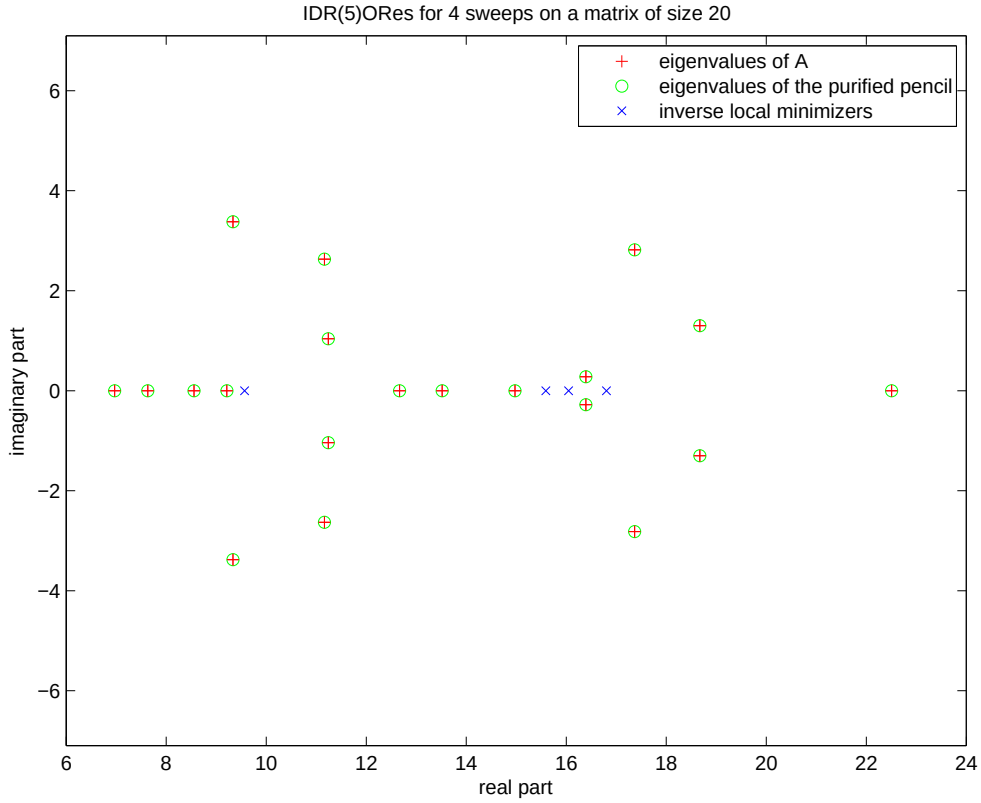


FIG. 4.2. *Approximation by Ritz values using the purified pencil. We used* IDR(s)ORES *with* $s = 5$ *for* 4 *sweeps. The* ω_j *were computed based on local minimization. The red plusses depict the eigenvalues of* $\mathbf{A}$*, the blue crosses depict the roots* $1/\omega_j$*,* $j = 1, \ldots, 4$*, and the green circles depict the* $20 = 4 \cdot 5$ *finite eigenvalues of the purified pencil of size* $24 = 4 \cdot (5+1)$ *resulting from* IDR(s)ORES.

In summary, the computation of the eigenvalues of the purified pencil including the infinite eigenvalues is viable.

4.3. The deflated IDR Hessenberg recurrence. It turns out to be easy to deflate the purified pencil in order to obtain a smaller pencil having only the finite eigenvalues. This also enhances our understanding of IDR(s)ORES, as this results in the reconstruction of the underlying BIORES$(s, 1)$ process. The deflation of the purified pencil is based on the Schur complement [32, pp. 216–217], i.e., block Gaussian elimination. To be more precise, with the submatrix

$$ {}^z\mathbf{H}_{(j-1)(s+1)+1:(j+1)(s+1)-1} =: \begin{pmatrix} {}^z\mathbf{H}^\star & \mathbf{h}_c & \mathbf{L}^\star \\ \mathbf{e}_s^\mathsf{T} & \gamma^\star - 1 & \mathbf{h}_r^\mathsf{T} \\ \mathbf{O} & \mathbf{e}_1 & {}^z\mathbf{H}_\star \end{pmatrix} \in \mathbb{C}^{(2s+1)\times(2s+1)} \qquad (4.33) $$

of the purified pencil ${}^z\mathbf{H}_n = z\mathbf{U}_n\mathbf{D}_\omega^{(n)} - \mathbf{Y}_n^\circ$, for some matching sweep index j,

$$1 \leqslant j < \left\lfloor \frac{n}{s+1} \right\rfloor, \tag{4.34}$$

we let ${}^z\mathbf{H}^\star := {}^z\mathbf{H}_{(j-1)(s+1)+1:j(s+1)-1}$ and ${}^z\mathbf{H}_\star := {}^z\mathbf{H}_{j(s+1)+1:(j+1)(s+1)-1}$ denote the leading and trailing Hessenberg matrix of size $s \times s$ of this submatrix, respectively. These are the only matrices that depend on the variable z in the purified pencil; compare with the zero structure of $\mathbf{U}_n\mathbf{D}_\omega^{(n)}$ in the purified pencil depicted in Figure 1.1. The element in the diagonal in position $j(s+1)$ is denoted by $\gamma^\star - 1$, where $\gamma^\star := \gamma_s^{(j(s+1))}$. The possibly non-zero elements in the $j(s+1)$th column and row of the upper triangular part of ${}^z\mathbf{H}_n$, i.e., of $-\mathbf{Y}_n^\circ$ are collected in the vectors $\mathbf{h}_c \in \mathbb{C}^s$ and $\mathbf{h}_r \in \mathbb{C}^s$, respectively. The matrix $\mathbf{L}^\star \in \mathbb{C}^{s\times s}$ is a strictly lower triangular matrix whose elements are independent of z.

We now use the following variant of the so-called *Schur determinant formula* [55, Eqn. (0.3.2), p. 5],

$$\begin{aligned}
&\begin{pmatrix} {}^z\mathbf{H}^\star & \mathbf{h}_c & \mathbf{L}^\star \\ \mathbf{e}_s^\mathsf{T} & \gamma^\star - 1 & \mathbf{h}_r^\mathsf{T} \\ \mathbf{O} & \mathbf{e}_1 & {}^z\mathbf{H}_\star \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{o} & \mathbf{O} \\ -\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & 1/(\gamma^\star-1) & -\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ \mathbf{O} & \mathbf{o} & \mathbf{I} \end{pmatrix} = \\
&\qquad \begin{pmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{h}_c^\mathsf{T}/(\gamma^\star-1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ \mathbf{o}^\mathsf{T} & 1 & \mathbf{o}^\mathsf{T} \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{e}_1^\mathsf{T}/(\gamma^\star-1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \end{pmatrix} = \\
&\begin{pmatrix} {}^z\mathbf{H}^\star & \mathbf{h}_c^\mathsf{T}/(\gamma^\star-1) & \mathbf{L}^\star \\ \mathbf{o}^\mathsf{T} & 1 & \mathbf{o}^\mathsf{T} \\ \mathbf{O} & \mathbf{e}_1^\mathsf{T}/(\gamma^\star-1) & {}^z\mathbf{H}_\star \end{pmatrix} - \begin{pmatrix} \mathbf{h}_c\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{o} & \mathbf{h}_c\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ \mathbf{o}^\mathsf{T} & 0 & \mathbf{o}^\mathsf{T} \\ \mathbf{e}_1\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{o} & \mathbf{e}_1\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \end{pmatrix}.
\end{aligned} \tag{4.35}$$

As the determinant of the block-Gauß eliminator is given by the non-zero value

$$\begin{vmatrix} \mathbf{I} & \mathbf{o} & \mathbf{O} \\ -\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & 1/(\gamma^\star-1) & -\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ \mathbf{O} & \mathbf{o} & \mathbf{I} \end{vmatrix} = \frac{1}{\gamma^\star - 1}, \tag{4.36}$$

we have proven that

$$\begin{aligned}
\frac{\chi_{(j-1)(s+1)+1:(j+1)(s+1)-1}(z)}{\gamma^\star - 1} &= \\
&\begin{vmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{h}_c^\mathsf{T}/(\gamma^\star-1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ \mathbf{o}^\mathsf{T} & 1 & \mathbf{o}^\mathsf{T} \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{e}_1^\mathsf{T}/(\gamma^\star-1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \end{vmatrix} \\
&= \begin{vmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T}/(\gamma^\star-1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^\mathsf{T}/(\gamma^\star-1) \end{vmatrix}.
\end{aligned} \tag{4.37}$$

The multiplication of the pencil ${}^z\mathbf{H}_n$ by an enlarged block-Gauß eliminator $\mathbf{G}_n^{(j)}$ does not affect the roots of the residual polynomial, i.e., the eigenvalues of the pencil, as this is a scaling by the constant $(\gamma^\star - 1)$. The resulting rank-one update (4.35) of the pencil only affects the elements of the submatrix given in Eqn. (4.33) without changing the Hessenberg structure of the leading and trailing block. Looking closely we realize that the resulting Hessenberg/upper triangular pencil is of course still regular and has lost one infinite eigenvalue by construction, but, moreover, the Hessenberg part is of OrthoRes-type. This follows from

$$\mathbf{e}^\mathsf{T} \begin{pmatrix} \mathbf{h}_c \\ \gamma^\star - 1 \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad -\mathbf{e}^\mathsf{T} \begin{pmatrix} \mathbf{h}_c/(\gamma^\star-1) \\ 1/(\gamma^\star-1) \end{pmatrix} = 1, \tag{4.38}$$

and the observation that only the last column of the pencil ${}^z\mathbf{H}^\star$ is modified, where the element 1, which is needed for column sum zero, is missing. The column sums of

the second half of the Hessenberg part are zero, since the omitted row elements have been dispersed across the rows by the rank one update.

The pattern of the original banded Hessenberg/upper triangular pencil is only modified in the upper triangle of the Hessenberg matrix. The constant lower triangular matrices $\mathbf{L}^\star \in \mathbb{C}^{s\times s}$ become full, as the rank-one update $\mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^\mathsf{T}/(\gamma^\star - 1)$ usually gives a full $s \times s$ matrix. Apart from altering the lower triangular matrices $\mathbf{L}^\star$ to full matrices by a rank-one update, the non-zero structure is only changed in the last column before and the first row after a diagonal zero element in $\mathbf{U}_n$. We remark that the change in the last column of the Hessenberg blocks is just a variation of the correction in [30, Theorem 3.1, Corollary 3.3, pp. 286–287].

Computing the block-Gauß eliminators $\mathbf{G}_n^{(j)}$, $j = 1, \ldots$, where a little care has to be taken for the last eliminator which, depending on the relation of n to $s+1$, may work on a smaller trailing Hessenberg matrix, and applying them to the pencil before computing the next eliminator is equivalent to computing *all* block-Gauß eliminators at once and thus their product $\mathbf{G}_n := \mathbf{G}_n^{(1)}\mathbf{G}_n^{(2)}\ldots$, as the eliminators solely depend on quantities which are not altered in previous steps. The basic eliminators $\mathbf{G}_n^{(j)}$ commute, thus the order of multiplication is not important. Due to the zero structure of the pencil we compute in our implementation the Schur complement of the matrix indexed by all diagonal elements equal to zero in $\mathbf{U}_n$ to enable the use of the BLAS Level 3. The elimination, i.e., multiplication of the pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ by $\mathbf{G}_n$ from the right results in the modified pencil $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, as the elimination does not change the upper triangular matrix $\mathbf{U}_n\mathbf{D}_\omega^{(n)}$ due to the zero structure, see the second and third pencil in Figure 1.1. After elimination, every $(s+1)$th row of the pencil $z\mathbf{U}_n\mathbf{D}_\omega^{(n)} - \mathbf{Y}_n^\circ\mathbf{G}_n$ is independent of the variable z and has its only non-zero element in the diagonal position, see the third pencil in Figure 1.1.

In the next step we remove every $(s+1)$th row and column from the pencil, which results in the smaller, fourth pencil depicted in Figure 1.1. The infinite eigenvalues have been deflated, and the transformed but not yet deflated pencil clearly reveals that the corresponding eigenvectors are the standard unit vectors $\mathbf{e}_{j(s+1)}$, $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$. Turning our attention to the underlying generalized Hessenberg decompositions, we observe that the deflation removes every $(s+1)$th vector from $\mathbf{W}_n$. Because we enumerate the purified residuals starting with index 0, the deflated vectors are $\mathbf{w}_{(j+1)(s+1)-1} = \mathbf{w}_{j(s+1)+s} = \widehat{\rho}_{(j+1)s}(\mathbf{A})\mathbf{w}_0$, $0 \leqslant j < \lfloor n/(s+1)\rfloor$. Denoting this deflation operator by D, we have proven the following:

THEOREM 4.13 (The deflated Hessenberg/upper triangular IDR(s)ORES pencil). *The generalized Hessenberg decomposition for the deflated* IDR(s)ORES *residuals is given by*

$$\mathbf{A}\widetilde{\mathbf{Q}}_m D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}) = \widetilde{\mathbf{Q}}_{m+1} D(\underline{\mathbf{Y}}_n^\circ\mathbf{G}_n), \quad m := n - \left\lfloor \frac{n}{s+1} \right\rfloor, \tag{4.39}$$

where D is a deflation operator that removes every $(s+1)$th row and column up to the nth, $\widetilde{\mathbf{Q}}_m := D(\mathbf{W}_n)$, $\widetilde{\mathbf{Q}}_{m+1} := \left(\widetilde{\mathbf{Q}}_m, \mathbf{w}_n\right)$ and $\mathbf{G}_n$ is the block-Gauß eliminator defined above.

As the deflated triangular matrix $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)})$ is now an invertible block-diagonal matrix with $s \times s$ upper triangular blocks, we can construct an algebraic eigenvalue problem by inversion. To retain the zero column sums we multiply the inverse from the right to the Hessenberg matrix. We obtain a banded Hessenberg matrix, which has triangles poking out of the band:

COROLLARY 4.14. *Suppose that all $\omega_j \neq 0$ and that all $\gamma_s^j \neq 1$. Then the Hessenberg decomposition for the deflated* IDR(s)ORES *residuals is given by*

$$\mathbf{A}\widetilde{\mathbf{Q}}_m = \widetilde{\mathbf{Q}}_{m+1}\underline{\widetilde{\mathbf{L}}}_m^\circ, \quad m = n - \left\lfloor \frac{n}{s+1} \right\rfloor, \tag{4.40}$$

where

$$\widetilde{\underline{\mathbf{L}}}_m^\circ := D(\underline{\mathbf{Y}}_n^\circ \mathbf{G}_n)\,(D(\mathbf{U}_n \mathbf{D}_\omega^{(n)}))^{-1} \in \mathbb{C}^{(m+1)\times m}. \tag{4.41}$$

The last column of the matrix $\widetilde{\underline{\mathbf{L}}}_m^\circ$ *is altered in the next step of* IDR(s)ORes *if and only if* $n = j(s+1)-1$ *for some* $j \geqslant 1$. *This matrix is an unreduced extended Hessenberg matrix of* OrthoRes*-type, and, at the same time, block-tridiagonal with blocks of dimension* s.

In the transition from step $n = j(s+1)-1$ to step $n = j(s+1)$ the index m and thus the sizes of $\widetilde{\underline{\mathbf{L}}}_m^\circ$ and $\widetilde{\mathbf{Q}}_m$ remain unchanged, but the last columns of these two matrices are changed in the deflation process. So, for these particular values of m there exists two different pairs of matrices $\widetilde{\underline{\mathbf{L}}}_m^\circ$, $\widetilde{\mathbf{Q}}_m$. In the next s steps new columns are added to $\widetilde{\underline{\mathbf{L}}}_m^\circ$ and $\widetilde{\mathbf{Q}}_m$, but all other elements of the matrices from the previous steps remain unchanged. We propose here to introduce a subset of these two matrix sequences obtained by deleting those matrices that are generated in steps where $n = j(s+1)-1$ for some j. For distinction, we rename the remaining matrices $\underline{\mathbf{L}}_m^\circ$ and $\mathbf{Q}_m$. They correspond to the leading unchanged parts of the decomposition (4.40). Interestingly, we can interpret these leading unchanged parts as those of the BiORes(s, 1) process underlying IDR(s)ORes.

Definition 4.15. *We define for* $m = js+k \geqslant 1$, $j = \lfloor m/s \rfloor$, $0 \leqslant k < s$, *the* m*th extended* BiORes(s, 1) *Lanczos matrix* $\underline{\mathbf{L}}_m^\circ$ *and the* m*th matrix of the* BiORes(s, 1) *residuals* $\mathbf{Q}_m$ *by*

$$\underline{\mathbf{L}}_m^\circ := D(\underline{\mathbf{Y}}_n^\circ \mathbf{G}_n)\,(D(\mathbf{U}_n \mathbf{D}_\omega^{(n)}))^{-1} \in \mathbb{C}^{(m+1)\times m} \tag{4.42a}$$

and

$$\mathbf{Q}_m := D(\mathbf{W}_n), \tag{4.42b}$$

respectively, where the index n *is defined by*

$$n := j(s+1)+k, \quad 0 \leqslant k < s. \tag{4.43}$$

We remark that by construction the index n *omits all values of the form* $j(s+1)+s$.

For the reader's convenience, the block-tridiagonal unreduced upper Hessenberg matrix $\mathbf{L}_9^\circ$ corresponding to Figure 1.1 with $s = 3$ for 3 sweeps is depicted in Figure 4.3.

The BiORes(3, 1) matrix $\mathbf{L}_9^\circ$ obtained with 3 full sweeps:

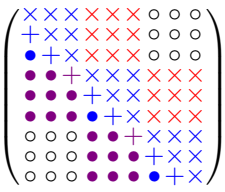


Fig. 4.3. *The* BiORes(3, 1) *matrix* $\mathbf{L}_9^\circ$ *corresponding to the pictorial example given in Figure 1.1.*

For $k = 0, \ldots, s-1$ and $j = 0, 1, \ldots$, the reduced residuals $\mathbf{q}_{js+k}$ are defined by

$$\Omega_j(\mathbf{A})\mathbf{q}_{js+k} := \mathbf{r}_{j(s+1)+k} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{j(s+1)+k-1}.$$

Every $\mathbf{v}_{j(s+1)+k-1}$ is orthogonal to $\mathcal{P}$, the range of the columns of $\mathbf{P}$. Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathcal{P}$. Using induction [35] one can prove that $\mathbf{q}_{js+k} \perp \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P})$; thus, the Hessenberg/block tridiagonal matrices $\mathbf{L}_m^\circ$ can be identified as those of a two-sided Lanczos process with s left and one right starting vectors:

Theorem 4.16 (The Hessenberg/block tridiagonal BiORes(s, 1) matrix). *The Hessenberg decomposition of the* BiORes(s, 1) *residuals is given by*

$$\mathbf{A}\mathbf{Q}_m = \mathbf{Q}_{m+1}\underline{\mathbf{L}}_m^\circ, \tag{4.44}$$

where, for $k = 0, \ldots, s-1$ *and* $j = 0, 1, \ldots$, *the* BiORes$(s,1)$ *residuals satisfy*

$$\mathbf{q}_{js+k} \in \mathbf{q}_0 + \mathbf{A}\mathcal{K}_{js+k}(\mathbf{A}, \mathbf{q}_0), \quad \mathbf{q}_{js+k} \perp \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}). \tag{4.45}$$

This follows easily using the representations collected in the next Lemma 4.17, which has to be compared with Theorem 4.2 in [35] and with Theorem 4.1 in [34]. In the latter report a similar result is obtained, but the authors use a slightly different method of proof.

Lemma 4.17 (Representations of the Sonneveld spaces). *Let* $\mathcal{P}$ *denote the range of the columns of* $\mathbf{P}$*, and let* $\mathcal{S} = \mathcal{P}^{\perp}$*. The Sonneveld spaces* $\mathcal{G}_j$ *are given by*

$$\mathcal{G}_0 = \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \textit{where } \mathcal{K}(\mathbf{A}, \mathbf{r}_0) \textit{ denotes the full Krylov subspace}, \tag{4.46a}$$

$$\mathcal{G}_j = \bigcap_{k=1}^{j} (\mathbf{I} - \omega_j \mathbf{A}) \cdots (\mathbf{I} - \omega_k \mathbf{A})(\mathcal{S}) = \bigcap_{k=0}^{j-1} \Omega_k(\mathbf{A})^{-1} \Omega_j(\mathbf{A})(\mathcal{S}) \tag{4.46b}$$

$$= \Big(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}} \Omega_k(\mathbf{A})^{\mathsf{H}} \mathcal{P} \Big)^{\perp} \tag{4.46c}$$

$$= \Big(\Omega_j(\mathbf{A})^{-\mathsf{H}} \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \Big)^{\perp} = \Omega_j(\mathbf{A}) \Big(\mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \Big)^{\perp}. \tag{4.46d}$$

Proof. The first $s+1$ residuals obviously are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$. The first set of equalities (4.46b)[11] follows from the observations that the next $s+1$ residuals (or any other vectors in $\mathcal{G}_1$) are in the $\mathbf{I} - \omega_1 \mathbf{A}$ image of $\mathcal{S}$, the last $s+1$ residuals are in the $\mathbf{I} - \omega_j \mathbf{A}$ image of $\mathcal{S}$, and, since they are computed as images of linear combinations of previous information, also images of linear combinations of previously obtained vectors in $(\mathbf{I} - \omega_{j-1}\mathbf{A}) \cdots (\mathbf{I} - \omega_k \mathbf{A})(\mathcal{S})$. The second equality (4.46c) is based on the two relations $\mathbf{B}\mathcal{P}^{\perp} = (\mathbf{B}^{-\mathsf{H}}\mathcal{P})^{\perp}$ and $\mathcal{U}^{\perp} \cap \mathcal{V}^{\perp} = (\mathcal{U} \cup \mathcal{V})^{\perp} = (\mathcal{U} + \mathcal{V})^{\perp}$[12]. The first relation follows from the observation that

$$\mathbf{B}\mathcal{P}^{\perp} = \left\{ \mathbf{B}\mathbf{v} \in \mathbb{C}^n \mid \mathbf{P}^{\mathsf{H}}\mathbf{v} = \mathbf{o}_s \right\} = \left\{ \mathbf{u} \in \mathbb{C}^n \mid \mathbf{P}^{\mathsf{H}}\mathbf{B}^{-1}\mathbf{u} = (\mathbf{B}^{-\mathsf{H}}\mathbf{P})^{\mathsf{H}}\mathbf{u} = \mathbf{o}_s \right\}, \tag{4.47}$$

see also [34]. The third set of equalities (4.46d) is satisfied since the polynomials $\Omega_k(\mathbf{A})$, $0 \leqslant k < j$, form a basis of the space of polynomials of degree less j, and by the property proved above, respectively. □

A plot of the eigenvalues of the deflated pencil $(D(\mathbf{Y}_n^{\circ}\mathbf{G}_n), D(\mathbf{U}_n \mathbf{D}_{\omega}^{(n)}))$ and thus of the eigenvalues of the matrix $\mathbf{L}_m^{\circ}$ gives in case of the previous example a picture that is indistinguishable from the picture given in Figure 4.2.

5. Ritz vectors and accuracy. Ritz vectors are obtained from a given *generalized* or *rational* Hessenberg decomposition (2.1),

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n, \tag{5.1}$$

by computing the eigenvalues θ_j and eigenvectors $\mathbf{s}_j$ of the pencil $(\mathbf{H}_n, \mathbf{U}_n)$, followed by a prolongation $\mathbf{z}_j := \mathbf{Q}_n\mathbf{U}_n\mathbf{s}_j$ of the eigenvectors to obtain the Ritz vectors $\mathbf{z}_j$, see also [29, page 595]. We only consider eigenvectors, the transition to handle principal vectors is straightforward and thus omitted. The residual of the unscaled Ritz pair $(\theta_j, \mathbf{z}_j)$ is then given by

$$\mathbf{A}\mathbf{z}_j - \mathbf{z}_j\theta_j = \mathbf{Q}_n(\mathbf{H}_n\mathbf{s}_j - \mathbf{U}_n\mathbf{s}_j\theta_j) + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^{\mathsf{T}}\mathbf{s}_j = \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^{\mathsf{T}}\mathbf{s}_j. \tag{5.2}$$

As is the case in standard Hessenberg decompositions, the size of the last element $\mathbf{e}_n^{\mathsf{T}}\mathbf{s}_j$ of the eigenvector $\mathbf{s}_j$ of the pencil dominates the accuracy,

$$\|\mathbf{A}\mathbf{z}_j - \mathbf{z}_j\theta_j\|_2 = \|\mathbf{q}_{n+1}\|_2 |h_{n+1,n}\mathbf{e}_n^{\mathsf{T}}\mathbf{s}_j|. \tag{5.3}$$

[11] This has been proven in the case $s = 1$ by Peter Sonneveld in 1976 and was the key for the proof of the first IDR Theorem (personal communication).

[12] The second author thanks Linde Yver for pointing out this basic linear algebra relation exactly at the right time.

In contrast to the standard case, the eigenvector $\mathbf{s}_j$ is not only prolonged with the matrix $\mathbf{Q}_n$ spanning the basis, but beforehand multiplied with the upper triangular, possibly singular $\mathbf{U}_n$ to obtain the Ritz vector $\mathbf{z}_j$. This affects the 2-norm backward error

$$\begin{aligned}\eta_{(\theta_j,\mathbf{z}_j)} &:= \min\{\eta \mid \|\mathbf{A}-\widetilde{\mathbf{A}}\|_2 \leqslant \eta, \widetilde{\mathbf{A}}\mathbf{z}_j = \mathbf{z}_j\theta_j\} \\ &= \frac{\|\mathbf{A}\mathbf{z}_j - \mathbf{z}_j\theta_j\|_2}{\|\mathbf{z}_j\|_2} = \frac{\|\mathbf{q}_{n+1}\|_2|h_{n+1,n}\mathbf{e}_n^{\mathsf{T}}\mathbf{s}_j|}{\|\mathbf{z}_j\|_2}\end{aligned} \tag{5.4}$$

of the Ritz pair $(\theta_j, \mathbf{z}_j)$ whenever the length of $\mathbf{z}_j$ differs significantly from unity.

Using the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$ (or, equivalently, the Sonneveld matrix $\mathbf{S}_n^\circ = \mathbf{Y}_n^\circ(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}$) we can easily prolong the eigenvectors $\mathbf{s}_j^{\text{Sonneveld}}$, $1 \leqslant j \leqslant n$, to obtain Ritz vectors $\mathbf{z}_j^{\text{Sonneveld}} := \mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega^{(n)}\mathbf{s}_j^{\text{Sonneveld}}$, $1 \leqslant j \leqslant n$, as we know the basis vectors, i.e., the residuals $\mathbf{r}_0, \ldots, \mathbf{r}_{n-1}$. Of course we do not store them, but similar to the way in Lanczos' method we can recompute them one by one if we store all the scalars (which saves some work). Thus, one chooses some Ritz values and computes afterwards the Ritz vectors. This choice usually is based on an interest in eigenvalues closest to some target or in some given region and some estimation of the accuracy obtained, mostly a measure or guess on the backward error. This can be done for the Sonneveld pencil or matrix representation, since we can compute the size of the residuals of the unscaled eigenpairs. Unfortunately, in this case, this is rather a *guess* of the accuracy of the eigenvalue, as the expression $\|\mathbf{r}_n\||s_{n+1,n}^\circ s_{nj}^{\text{Sonneveld}}|$ is an error bound for the residual of an unnormalized approximation to an eigenpair, where $\mathbf{r}_n$ and $s_{n+1,n}^\circ$ are the last column of the matrix $\mathbf{R}_{n+1}$ and the only non-trivial element in the last row of the Sonneveld matrix $\underline{\mathbf{S}}_n^\circ$ in (4.14), respectively.

To make use of the purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, the deflated pencil $(D(\mathbf{Y}_n^\circ\mathbf{G}_n)$, $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$, or even the BiORes$(s,1)$ matrix $\mathbf{L}_m^\circ$, we have to prolong using the unknown vectors $\mathbf{w}_\ell$ or the unknown $\mathbf{q}_\ell$, i.e., a subset of the $\mathbf{w}_\ell$. These could be computed using the scalar recurrences (4.44). The question arises how much the process of computing the vectors $\mathbf{w}_\ell$ in finite precision deteriorates, see also [38]. Naïvely implemented, this turns out to be a bad idea in finite precision, as the errors are amplified far beyond norm one and thus the computed vectors $\mathbf{w}_\ell$ are useless. Numerical experiments support the assumption that the computed $\mathbf{w}_\ell$ are accurate as long as the true and updated residuals in IDR(s)ORes are close, see Eqn. (4.3) and the remark thereafter. Unfortunately, there exist examples where this no longer holds true. A picture that compares the norms of the $\mathbf{w}_\ell$, computed using the matrix $\underline{\mathbf{L}}_n^\circ$ and $\mathbf{w}_0 = \mathbf{r}_0$ as starting vector, the differences between the recomputed $\mathbf{r}_\ell = \Omega_{\lfloor \ell/(s+1)\rfloor}(\mathbf{A})\mathbf{w}_\ell$, and the computed $\mathbf{r}_\ell$ is given in Figure 5.1. Of course, only those $\mathbf{w}_\ell$ are computed, which are needed for the prolongation, that is, omitting the columns $\ell = j(s+1)$ with the indices $\ell - 1$.

As we usually compute the vectors $\mathbf{w}_\ell$ only after we have decided which Ritz vectors we are interested in, the question arises whether we could measure or guess the accuracy of a Ritz pair without actually computing the $\mathbf{w}_\ell$. A related question is whether we can transform an eigenvector of the purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ (and the pencils and matrix beyond) to an eigenvector of the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$. Fortunately, the answer is yes for both questions:

THEOREM 5.1 (The eigenvector transformations). *Let $\mathbf{s}_j^{\text{purified}}$ denote an eigenvector to the eigenvalue θ_j of the purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$. Then the vector $\tilde{\mathbf{s}}_j^{\text{Sonneveld}}$ defined componentwise by*

$$\tilde{s}_{\ell j}^{\text{Sonneveld}} := \frac{s_{\ell j}^{\text{purified}}}{\Omega_{\lfloor \ell/(s+1)\rfloor}(\theta_j)}, \quad 1 \leqslant \ell \leqslant n, \tag{5.5}$$

is an eigenvector to the eigenvalue θ_j of the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$.

Let $\mathbf{s}_j^{\text{eliminated}}$ denote an eigenvector to the eigenvalue θ_j of the eliminated pencil $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ and let $\mathbf{s}_j^{\text{deflated}}$ denote an eigenvector to the eigenvalue θ_j of the

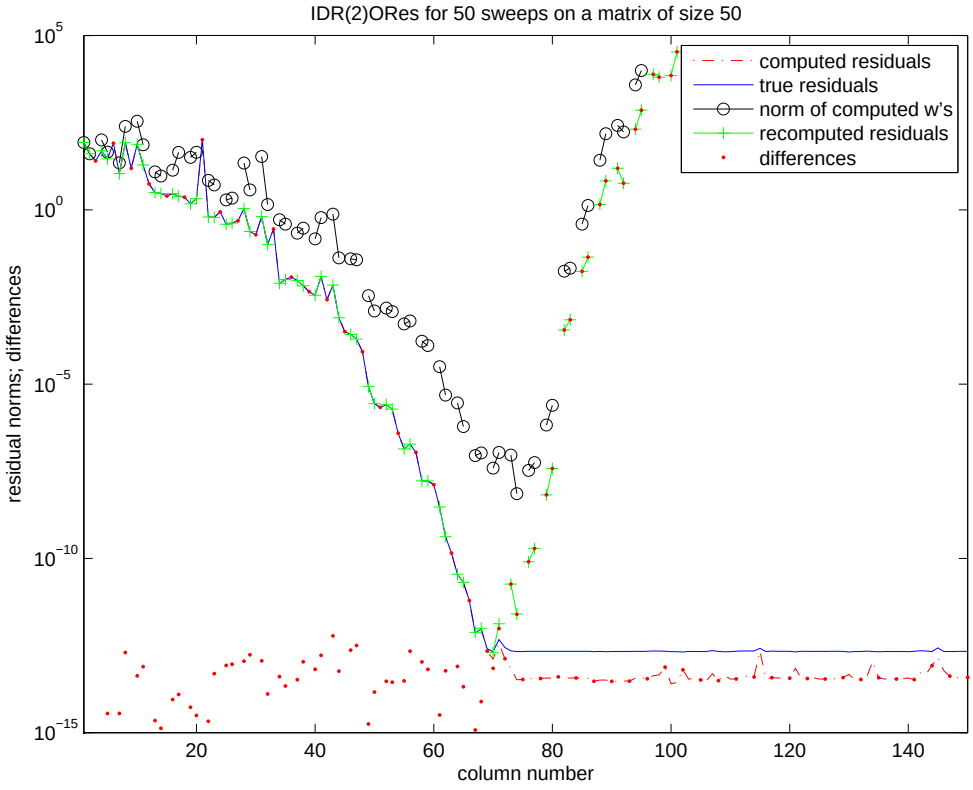


FIG. 5.1. *Quality of computed* $\mathbf{W}_n$ *measured by column norms. We compute the vectors* $\mathbf{w}$ *with the aid of the computed Lanczos-*BIORES$(s,1)$ *matrix* $\underline{\mathbf{L}}^\circ$*. To quickly check the quality of the obtained vectors we multiply them with the polynomials* $\Omega_j(\mathbf{A})$ *and compare the norms of the resulting recomputed residuals with the norms of the actually computed residuals* $\mathbf{r}$*.*

deflated pencil $(D(\mathbf{Y}_n^\circ\mathbf{G}_n), D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$*. Let* $D^\dagger$ *denote the prolongation operator that inserts the value zero after every* s *elements of a vector. Then the vectors* $\tilde{\mathbf{s}}_j^{\text{purified}}$ *and* $\tilde{\mathbf{s}}_j^{\text{eliminated}}$ *defined by*

$$\tilde{\mathbf{s}}_j^{\text{purified}} := \mathbf{G}_n\mathbf{s}_j^{\text{eliminated}} \quad \textit{and} \quad \tilde{\mathbf{s}}_j^{\text{eliminated}} := D^\dagger(\mathbf{s}_j^{\text{deflated}}) \tag{5.6}$$

are eigenvectors to the eigenvalue θ_j *of the purified pencil* $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ *and the eliminated pencil* $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$*, respectively. Additionally,* $\|\tilde{\mathbf{s}}_j^{\text{purified}}\|_2 \geqslant \|\mathbf{s}_j^{\text{eliminated}}\|_2$ *and* $\|\tilde{\mathbf{s}}_j^{\text{eliminated}}\|_2 = \|\mathbf{s}_j^{\text{deflated}}\|_2$*.*

Suppose that all ω_j*,* $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$*, are distinct and that no* $1/\omega_j$ *is an eigenvalue of the* BIORES$(s,1)$ *matrix. Then unnormalized eigenvectors of the Sonneveld pencil to the* $\lfloor n/(s+1)\rfloor$ *eigenvalues* $1/\omega_j$*,* $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$*, are given by*

$$\mathbf{s}_j^{\text{Sonneveld};\omega_j} := \begin{pmatrix} \mathbf{o}_{j(s+1)-1} \\ 1 \\ \boldsymbol{\nu}_j(\omega_j) \end{pmatrix} \in \mathbb{C}^n, \quad 1 \leqslant j \leqslant \left\lfloor \frac{n}{s+1} \right\rfloor, \tag{5.7}$$

where $\boldsymbol{\nu}_j(\omega_j) \in \mathbb{C}^{n-j(s+1)}$ *is the solution of the trailing linear Hessenberg system*

$$(\mathbf{Y}^\circ_{j(s+1)+1:n} - \omega_j^{-1}\mathbf{Y}_{j(s+1)+1:n}\mathbf{D}_\omega^{(j(s+1)+1:n)})\boldsymbol{\nu}_j(\omega_j) = \mathbf{e}_1 \in \mathbb{C}^{n-j(s+1)}. \tag{5.8}$$

Normalized eigenvectors corresponding to the $\lfloor n/(s+1)\rfloor$ *infinite eigenvalues of the purified pencil* $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ *and the eliminated pencil* $(\mathbf{Y}_n^\circ\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$ *are given by the standard unit vectors* $\mathbf{e}_{j(s+1)}$*,* $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$*.*

Proof. The polynomial vector $\boldsymbol{\nu}(z)$ defined in (2.4), evaluated at an eigenvalue of an unreduced Hessenberg/upper triangular pencil $(\mathbf{H}, \mathbf{U})$, gives a corresponding eigenvector, see (2.5). In particular, the components of the polynomial vector $\boldsymbol{\nu}^{\text{Sonneveld}}$ of

the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$ and of the polynomial vector $\boldsymbol{\nu}^{\text{purified}}$ of the purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, respectively, are defined with the aid of the determinants of trailing sections of the characteristic matrices, i.e., by

$$\nu_\ell^{\text{Sonneveld}}(z) := (h_{\ell:n-1})^{-1} \cdot \det\left(z\mathbf{Y}_{\ell+1:n}\mathbf{D}_\omega^{(\ell+1:n)} - \mathbf{Y}_{\ell+1:n}^\circ\right) \tag{5.9a}$$

and

$$\nu_\ell^{\text{purified}}(z) := (h_{\ell:n-1})^{-1} \cdot \det\left(z\mathbf{U}_{\ell+1:n}\mathbf{D}_\omega^{(\ell+1:n)} - \mathbf{Y}_{\ell+1:n}^\circ\right), \tag{5.9b}$$

respectively. The values $h_{\ell:n-1} := \prod_{j=\ell}^{n-1}(-y_{j+1,j})$ are the non-zero products of the subdiagonal elements (4.31) of the unreduced Hessenberg matrices $-\mathbf{Y}_{\ell+1:n}^\circ$ appearing in the definition of the ℓth component of both polynomial vectors. When $\ell \geqslant s$, the constant $h_{\ell:n-1}$ is simply one. To proceed, we first need to consider the eigenvectors of the Sonneveld pencil to the eigenvalues $1/\omega_j$.

We prove that $1/\omega_j$ is an eigenvalue of the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$ and reveal part of the zero-structure of the corresponding eigenvector. Let $\ell = j(s+1)$ denote the first index where the diagonal of $\mathbf{D}_\omega^{(n)}$ is equal to ω_j and let $k = (j+1)(s+1) = \ell + s + 1$. We investigate the structure of the matrix

$$\mathbf{H}_n := \mathbf{Y}_n^\circ - \omega_j^{-1}\mathbf{Y}_n\mathbf{D}_\omega^{(n)} = \begin{pmatrix} \boxed{\begin{matrix} \mathbf{H}_{1:\ell-1} & \mathbf{o}_{\ell-1} \\ -\mathbf{e}_{\ell-1}^\mathsf{T} & 0 \end{matrix}} & \mathbf{O}_{\ell,n-\ell} \\ -\mathbf{e}_1\mathbf{e}_\ell^\mathsf{T} & \boxed{\begin{matrix} -\mathbf{N}_s^\mathsf{T} & \mathbf{L}_{s,n-s-\ell} \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T} & \mathbf{H}_{k:n} \end{matrix}} \end{pmatrix}. \tag{5.10}$$

Here, by abuse of notation, $\mathbf{e}_1$ denotes first standard unit vectors of different length, $\mathbf{H}_{1:\ell-1} \in \mathbb{C}^{(\ell-1)\times(\ell-1)}$ and $\mathbf{H}_{k:n} \in \mathbb{C}^{(n-k+1)\times(n-k+1)}$ denote the leading and trailing portion of the Hessenberg matrix $\mathbf{H}_n$, respectively, and $\mathbf{L}_{s,n-s-\ell} \in \mathbb{C}^{s\times(n-s-\ell)}$ denotes a lower triangular rectangular matrix comprising elements of $\mathbf{H}_n$. The unreduced Hessenberg leading $\ell \times \ell$ block

$$\begin{pmatrix} \mathbf{H}_{1:\ell-1} & \mathbf{o}_{\ell-1} \\ -\mathbf{e}_{\ell-1}^\mathsf{T} & 0 \end{pmatrix} \in \mathbb{C}^{\ell\times\ell} \tag{5.11}$$

of $\mathbf{H}_n$ has rank $\ell-1$, i.e., is singular with null-vector $\mathbf{e}_\ell$. This and the zero block in the position $(1,2)$ of the 2×2 block matrix (5.10) implies singularity of $\mathbf{H}_n$. Thus, we have proven that $1/\omega_j$ is an eigenvalue of the Sonneveld pencil, $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$. As the eigenvalues of the Sonneveld pencil are the union of the $1/\omega_j$, which are assumed to be pairwise distinct, and the eigenvalues of the BiOres$(s,1)$ matrix, every eigenvalue $1/\omega_j$ is simple. Thus, zero can be no eigenvalue of the trailing block

$$\begin{pmatrix} -\mathbf{N}_s^\mathsf{T} & \mathbf{L}_{s,n-s-\ell} \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T} & \mathbf{H}_{k:n} \end{pmatrix} \in \mathbb{C}^{(n-\ell)\times(n-\ell)} \tag{5.12}$$

and we can uniquely solve the Hessenberg linear system

$$\begin{aligned} \mathbb{C}^{n-\ell} \ni \mathbf{e}_1 &= \left(\mathbf{Y}_{\ell+1:n}^\circ - \omega_j^{-1}\mathbf{Y}_{\ell+1:n}\mathbf{D}_\omega^{(\ell+1:n)}\right)\boldsymbol{\nu}_j(\omega_j) \\ &= \begin{pmatrix} -\mathbf{N}_s^\mathsf{T} & \mathbf{L}_{s,n-s-\ell} \\ -\mathbf{e}_1\mathbf{e}_s^\mathsf{T} & \mathbf{H}_{k:n} \end{pmatrix}\boldsymbol{\nu}_j(\omega_j) \end{aligned} \tag{5.13}$$

arising in re-ordering the homogeneous system

$$\mathbf{H}_n\begin{pmatrix} \mathbf{e}_\ell \\ \boldsymbol{\nu}_j(\omega_j) \end{pmatrix} = \mathbf{o}_n, \quad \mathbf{e}_\ell \in \mathbb{C}^\ell, \quad \boldsymbol{\nu}_j(\omega_j) \in \mathbb{C}^{n-\ell} \tag{5.14}$$

for the determination of the kernel of (5.10). This gives us an eigenvector of $1/\omega_j$.

In the preceeding part of the proof we have shown that also the Sonneveld trailing subdeterminants appearing in (5.9a) have some of the $1/\omega_j$ as roots, regardless of the ORTHORES-property. The Hessenberg matrix

$$\mathbf{Y}^{\circ}_{\ell+1:n} - \omega_j^{-1}\mathbf{Y}_{\ell+1:n}\mathbf{D}_{\omega}^{(\ell+1:n)} \tag{5.15}$$

is singular whenever $\ell+1 \leqslant j(s+1)$, i.e., whenever $\lceil(\ell+1)/(s+1)\rceil \leqslant j$. Let a trailing counterpart of the Sonneveld matrix be defined by the Hessenberg matrix $\mathbf{S}^{\times}_{\ell+1:n} := \mathbf{Y}^{\circ}_{\ell+1:n}(\mathbf{D}_{\omega}^{(\ell+1:n)})^{-1}\mathbf{Y}^{-1}_{\ell+1:n}$, where the notation has been altered from $\mathbf{S}^{\circ}$ to $\mathbf{S}^{\times}$ to reflect the loss of the ORTHORES-property. Consider the trivial generalized Hessenberg decomposition

$$\mathbf{S}^{\times}_{\ell+1:n}\mathbf{I}_{n-\ell}\mathbf{Y}_{\ell+1:n}\mathbf{D}_{\omega}^{(\ell+1:n)} = \mathbf{I}_{n-\ell}\mathbf{Y}^{\circ}_{\ell+1:n}, \tag{5.16}$$

where the identity matrices on both sides of the equation play the rôle of the matrix of basis vectors. This generalized Hessenberg decomposition can be interpreted as a recurrence with the system matrix $\mathbf{S}^{\times}_{\ell+1:n}$ and starting vector $\mathbf{e}_1 \in \mathbb{C}^{n-\ell}$. By Theorem 2.2 the basis vectors, i.e., the standard unit vectors $\mathbf{e}_j \in \mathbb{C}^{n-\ell}$, $1 \leqslant j \leqslant n-\ell$, can be interpreted as scaled polynomials in $\mathbf{S}^{\times}_{\ell+1:n}$ times the first standard unit vector $\mathbf{e}_1$. A purification of the polynomials appearing in (5.9a) from the known roots $1/\omega_j$, with index j satisfying $\lceil(\ell+1)/(s+1)\rceil = \lfloor\ell/(s+1)\rfloor + 1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$, analogous to the purification in the set of equations (4.24), results in the purified trailing subdeterminants appearing in (5.9b). Thus,

$$\nu_{\ell}^{\text{Sonneveld}}(z) = \Big(\prod_{j=\lfloor\ell/(s+1)\rfloor+1}^{\lfloor n/(s+1)\rfloor}(1-\omega_j z)\Big)\nu_{\ell}^{\text{purified}}(z), \tag{5.17}$$

or, after a cosmetic division of the vector $\nu_{\ell}^{\text{Sonneveld}}(\theta_j)$ defining the eigenspace by the constant factor $\Omega_{\lfloor n/(s+1)\rfloor}(\theta_j)$,

$$\frac{\nu_{\ell}^{\text{Sonneveld}}(\theta_j)}{\Omega_{\lfloor n/(s+1)\rfloor}(\theta_j)} = \frac{\nu_{\ell}^{\text{purified}}(\theta_j)}{\Omega_{\lfloor \ell/(s+1)\rfloor}(\theta_j)}. \tag{5.18}$$

As any eigenvector of the purified pencil $(\mathbf{Y}^{\circ}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)})$ is collinear to the non-zero vector $\nu_{\ell}^{\text{purified}}(\theta_j)$, we have constructed a vector collinear to the non-zero vector $\nu_{\ell}^{\text{Sonneveld}}(\theta_j)$, which is an eigenvector of the Sonneveld pencil $(\mathbf{Y}^{\circ}_n, \mathbf{Y}_n\mathbf{D}_{\omega}^{(n)})$ to the eigenvalue θ_j.

The remaining portion of the theorem is based on a few simple observations on the structure of the deflation process. The deflation can be reversed: An eigenvector $\mathbf{s}_j^{\text{deflated}}$ of the deflated pencil $(D(\mathbf{Y}^{\circ}_n\mathbf{G}_n), D(\mathbf{U}_n\mathbf{D}_{\omega}^{(n)}))$ is prolonged by inserting zeros after every multiple of s to obtain an eigenvector $\tilde{\mathbf{s}}_j^{\text{eliminated}} = D^{\dagger}(\mathbf{s}_j^{\text{deflated}})$ to the same eigenvalue of the eliminated pencil $(\mathbf{Y}^{\circ}_n\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)}) = (\mathbf{Y}^{\circ}_n\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)}\mathbf{G}_n)$. Especially, the norm of a vector is not altered by inserting additional zero elements. The vectors $\mathbf{e}_{j(s+1)}$, $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$, are in the null space of $\mathbf{U}_n\mathbf{D}_{\omega}^{(n)} = \mathbf{U}_n\mathbf{D}_{\omega}^{(n)}\mathbf{G}_n$ and thus are eigenvectors to the $\lfloor n/(s+1)\rfloor$-fold semisimple eigenvalue at infinity of both the eliminated and the purified pencil. As all finite eigenvalues are simple, any zero elements in the eigenvectors are uniquely determined. The block-Gauß eliminator $\mathbf{G}_n$ maps eigenvectors $\mathbf{s}_j^{\text{eliminated}}$ of the eliminated pencil $(\mathbf{Y}^{\circ}_n\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)}) = (\mathbf{Y}^{\circ}_n\mathbf{G}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)}\mathbf{G}_n)$ to eigenvectors $\tilde{\mathbf{s}}_j^{\text{purified}}$ of the purified pencil $(\mathbf{Y}^{\circ}_n, \mathbf{U}_n\mathbf{D}_{\omega}^{(n)})$, as

$$(\mathbf{Y}^{\circ}_n - \theta_j\mathbf{U}_n\mathbf{D}_{\omega}^{(n)})\tilde{\mathbf{s}}_j^{\text{purified}} = (\mathbf{Y}^{\circ}_n\mathbf{G}_n - \theta_j\mathbf{U}_n\mathbf{D}_{\omega}^{(n)})\mathbf{s}_j^{\text{eliminated}} = \mathbf{o}_n. \tag{5.19}$$

By inspection, the multiplication by the block-Gauß eliminator only alters the $\lfloor n/(s+1)\rfloor$ zero elements in positions $j(s+1)$, $1 \leqslant j \leqslant \lfloor n/(s+1)\rfloor$; thus, the norm of the image is bounded from below by the norm of the preimage. □

Utilizing the theorem, we can compute the Sonneveld eigenvectors without using directly the Sonneveld pencil. The last component of the scaled eigenvector together with the norm of the last residual yields an estimator of the backward error. So, error estimation without the Sonneveld pencil is indeed possible.

6. Numerical examples. We present several examples chosen to highlight the behavior of IDR(s)Eig based on IDR(s)ORes in finite precision. First, we identify numerically the "best" approach using the five different approaches, namely, the QZ or QR algorithm, as implemented in MATLAB, applied to the computed Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$ and Sonneveld matrix $\mathbf{S}_n^\circ = \mathbf{Y}_n^\circ(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}$, to the computed purified pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, and to the computed deflated pencil $(D(\mathbf{Y}_n^\circ\mathbf{G}_n), D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$ and BiORes$(s,1)$ matrix $\mathbf{L}_m^\circ$; see Figure 1.1 on page 7 and Figure 4.3 on page 23. Next, we depict with selected examples the phenomena of "misconvergence" and of "barriers" around the minimizers, influencing the reachable set of eigenvalues.

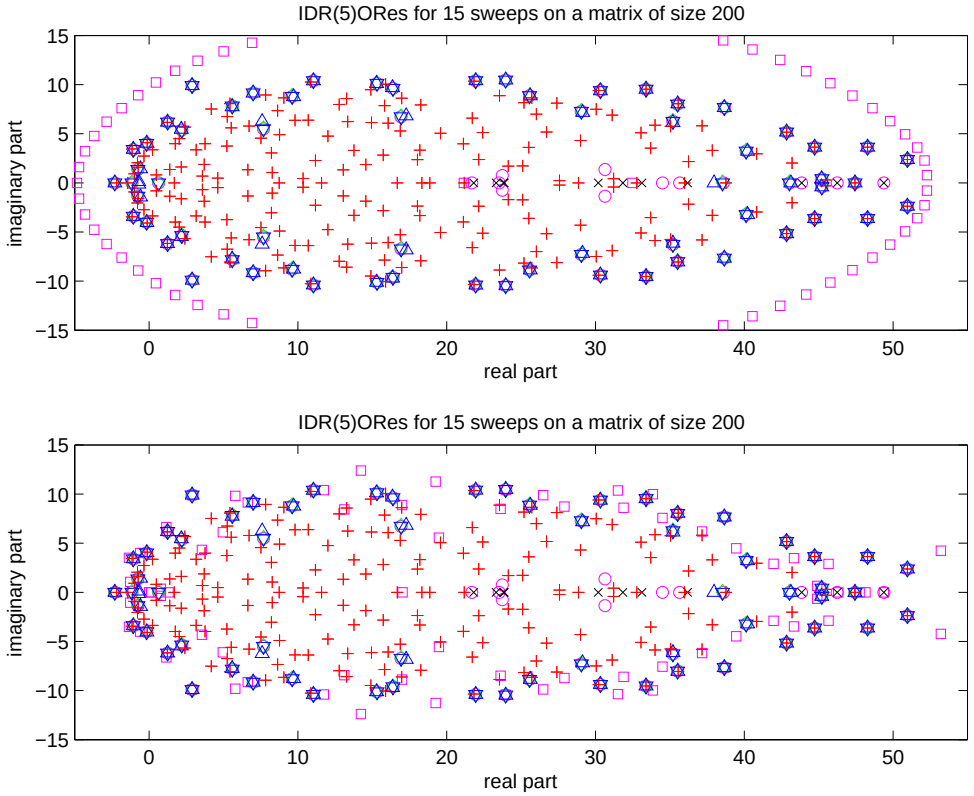


FIG. 6.1. *Approximation by finite precision Ritz values using* 15 *sweeps of* IDR(5)ORes *on a matrix* $\mathbf{A}$ *of dimension* 200. *The red plusses depict the eigenvalues of* $\mathbf{A}$, *the black crosses depict the minimizers* $1/\omega_j$, *the magenta circles depict the eigenvalues of the Sonneveld pencil, the magenta squares depict the eigenvalues of the Sonneveld matrix, the green diamonds depict the eigenvalues of the purified pencil, the blue downward triangles depict the eigenvalues of the deflated pencil, and the blue upward triangles depict the eigenvalues of the* BiORes$(5,1)$ *matrix. The upper and lower subplot differ only in the computation of the eigenvalues of the Sonneveld matrix, see the remarks in the text.*

For 15 sweeps of IDR(5)ORes, a comparison of all the five different eigenvalue approximations subject to finite precision is shown in the first subplot of Figure 6.1. The dimension of the matrix $\mathbf{A}$ is 200, the starting residual chosen is $\mathbf{b} = \mathbf{A}\mathbf{e}$. All eigenvalue approximations give the same results except the eigenvalue approximations based on the Sonneveld matrix, which gives worse approximations. The difference between the approximation quality of the Sonneveld matrix approach and the Sonneveld pencil are too pronounced to be random. Numerical experiments indicate that the eigenvalue approximations obtained using the Sonneveld matrix $\mathbf{S}_n^\circ = \mathbf{Y}_n^\circ(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}$ differ sometimes considerably from the eigenvalues of the similar matrix $(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}\mathbf{Y}_n^\circ$, which usually gives better eigenvalue approximations. This seems to be related to missed data dependencies and partially to small ω_j, which result in residual polynomial roots $1/\omega_j$ outside the field of values of $\mathbf{A}$, compare with the statements in the IDR-Ritz approach by Simoncini and Szyld [34]. To minimize data dependencies we computed in a second run the eigenvalues of the latter matrix

using the relation

$$(\mathbf{D}_\omega^{(n)})^{-1}\mathbf{Y}_n^{-1}\mathbf{Y}_n^\circ = (\mathbf{D}_\omega^{(n)})^{-1}(\mathbf{I}_n - \mathbf{Y}_n^{-1}\mathbf{N}_n^\mathsf{T}), \tag{6.1}$$

based on $\mathbf{Y}_n^\circ = \mathbf{Y}_n - \mathbf{N}_n^\mathsf{T}$. Still, the eigenvalue approximations obtained in this manner are not of the quality obtained with the Sonneveld pencil, see the lower subplot in Figure 6.1. For this stability reason and because the Sonneveld matrix is a *full* Hessenberg matrix, which makes the computation of approximate eigenvalues for many sweeps computationally intractable, we focus on the other approaches. The approaches connected to Hessenberg/triangular pencils, i.e., those based on the QZ algorithm, behave more stable. In the following examples we solely used the deflated pencil for the computation of the eigenvalue approximations.

The Ritz values of the purified process stop to converge close to the minimizers. The series of Figures 6.4–6.10 reminds us of the pseudospectra of polynomials, see, e.g., Figure 6.8. The matrix $\mathbf{A}$ has dimension 200 and we used k sweeps of IDR(2)-ORes for k equal to 20, 50, 100, 150, 300, 500 and 1000. We experimentally tested our conjecture on the relation to pseudospectra and used a *complex* random matrix $\mathbf{P}$. This ensures that the minimizers are allowed to become complex. As can be seen in the series of Figures 6.11–6.17, the BiORes$(s,1)$-part also captures the roots of the residual polynomials Ω_j. We observe that numerically the roots due to the one-dimensional minimizations are also found by the Lanczos process. This is due to the fact that we work with a perturbed mapped left Krylov space and the mapping is not "precise" enough.

It is a remarkable feature (at least in the numerical experiments carried out thus far) that in some steps there appears an eigenvalue of the purified Hessenberg pencil that is close to zero. The elements of the corresponding eigenvector seem to measure the deterioration of the process with a rate approximately inverse to the convergence rate measured by the norm of the constructed residuals. This Ritz value zero occurs after IDR(s)ORes has attained its ultimately attainable accuracy. This can be nicely seen in Figure 6.2 and Figure 6.3, which depict the convergence curves of the runs of IDR(s)ORes, first with real, then with a complex matrix $\mathbf{P}$. This occurrence of a zero Ritz value seems to be related to the comment following Eqn. (4.3) on the differences of true and updated residuals on page 13.

In [20, page 146] it was conjectured that higher values of s give a better conditioned basis. We might then get faster convergence to outliers. To test this conjecture, we used the matrix `e05r0500` from the Matrix Market [26]. This matrix models a 2D fluid flow in a driven cavity with 5×5-elements and Reynolds number 500. This matrix is non-symmetric real and zero is in the field of values. We used the given right-hand side `e05r0500_rhs1` and the zero vector as starting solution. This linear system is known to be a hard test case for any transpose-free short-term recurrence Krylov subspace method, i.e., LTPM. These methods converge to a more or less accurate solution after an extremely large number of steps, i.e., between 50.000 and 300.000 steps for a matrix of size 236; in [7] the authors even conjectured that these methods fail. We used 60 steps for different $s = 1, 2, 3, 4, 5, 9$, chosen such that 60 is divisible by $s + 1$. The resulting series of Figures 6.18–6.23 supports the conjecture that higher values of s give faster approximations to outliers. The best approximation is found for $s = 5$, see Figure 6.22. The bad approximation in case of $s = 9$ seems to be related to small minimizers ω_j magnifying the influence of rounding errors.

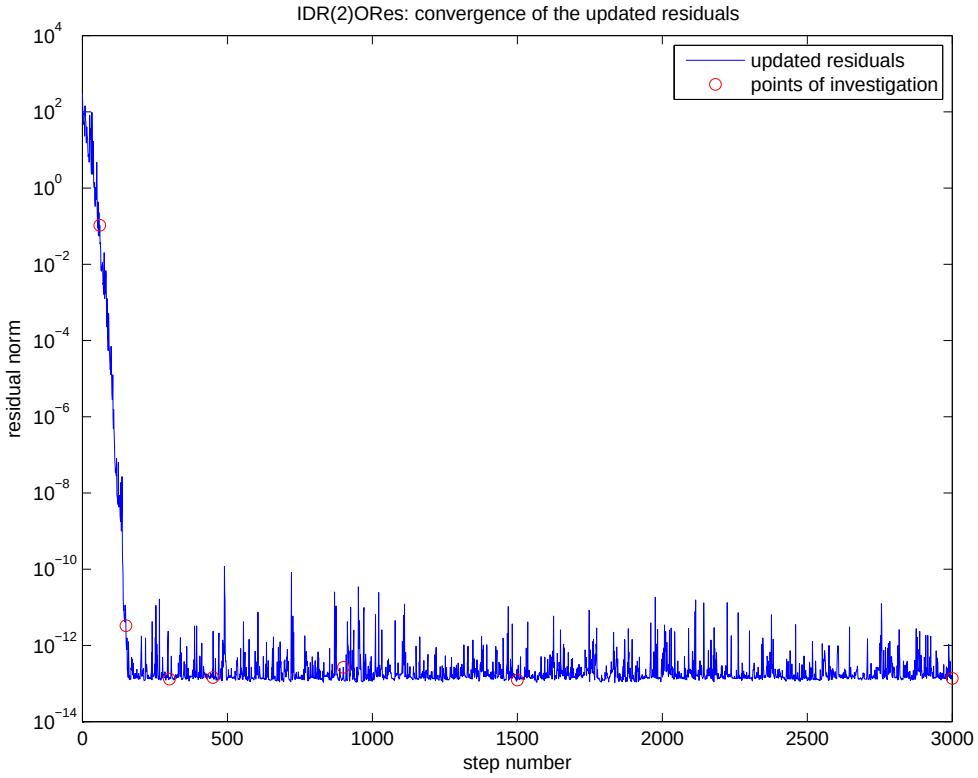


FIG. 6.2. *The size of the computed residuals for* 1000 *sweeps of* IDR(2)ORES *with a random real matrix* $\mathbf{P} \in \mathbb{R}^{n \times s}$ *with orthonormal columns. The red circles depict the points where we computed the eigenvalues of the underlying* BIORES$(2,1)$ *process.*

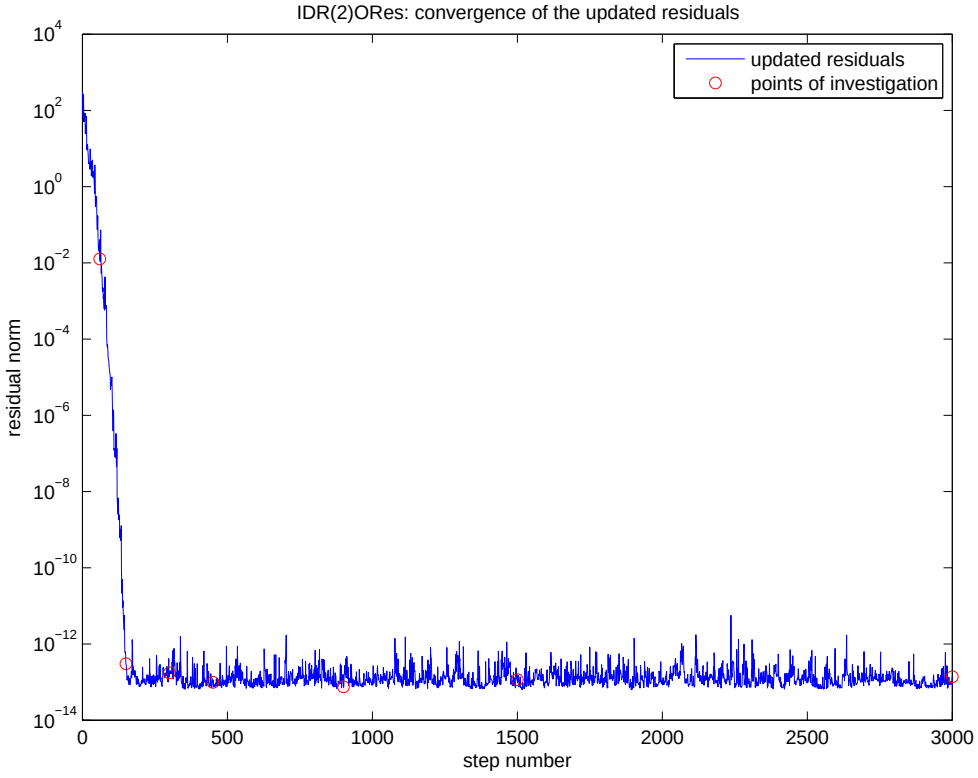


FIG. 6.3. *The size of the computed residuals for* 1000 *sweeps of* IDR(2)ORES *with a random complex matrix* $\mathbf{P} \in \mathbb{C}^{n \times s}$ *with orthonormal columns. The red circles depict the points where we computed the eigenvalues of the underlying* BIORES$(2,1)$ *process.*

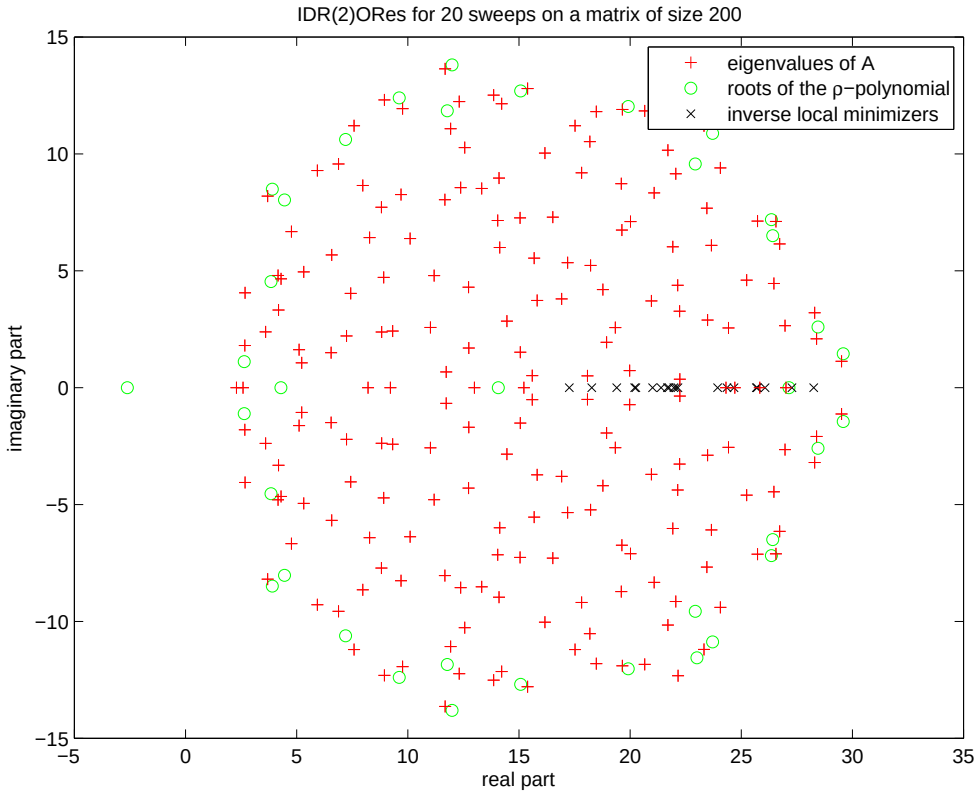


FIG. 6.4. *The formation of "pseudospectra": 20 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

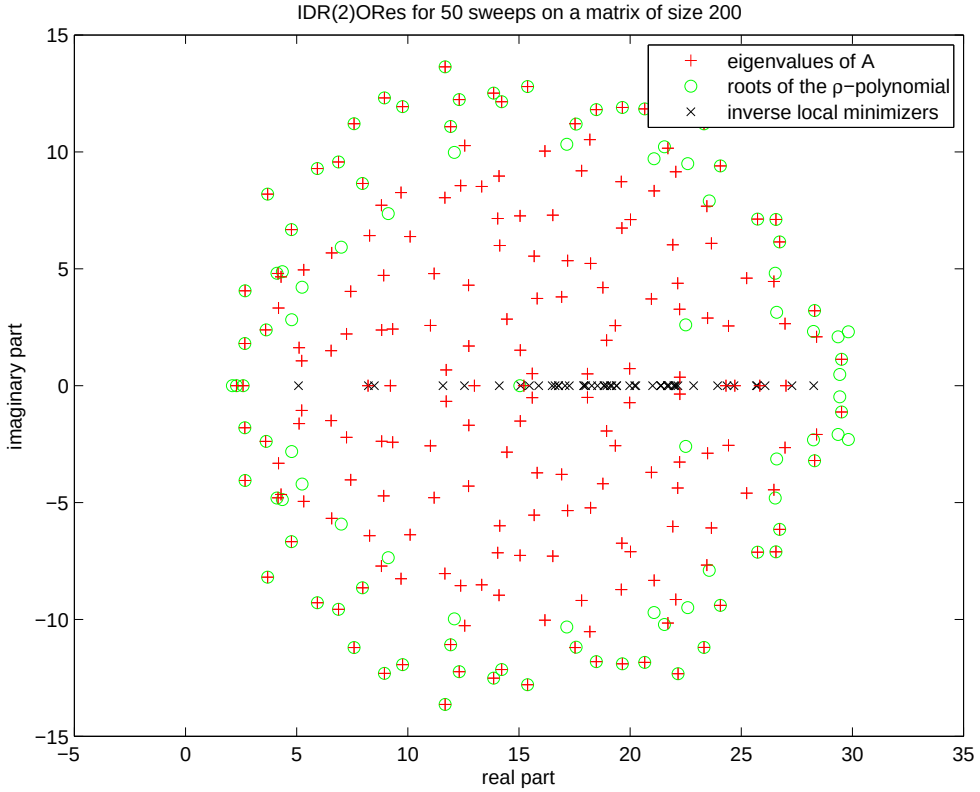


FIG. 6.5. *The formation of "pseudospectra": 50 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

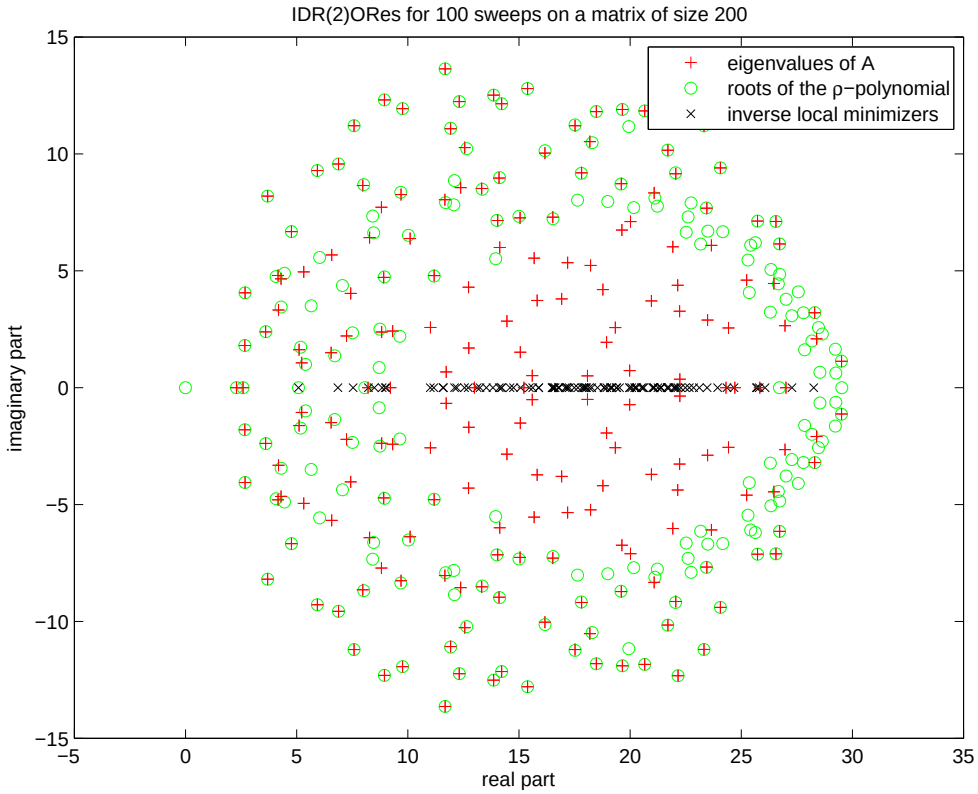


FIG. 6.6. *The formation of "pseudospectra": 100 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

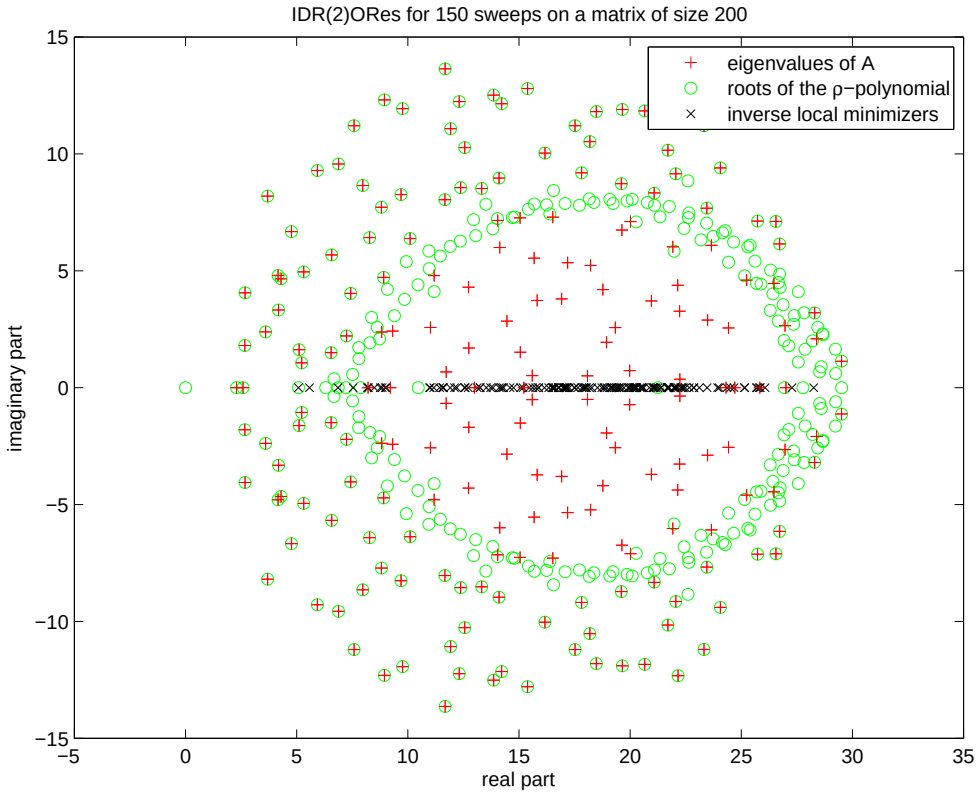


FIG. 6.7. *The formation of "pseudospectra": 150 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

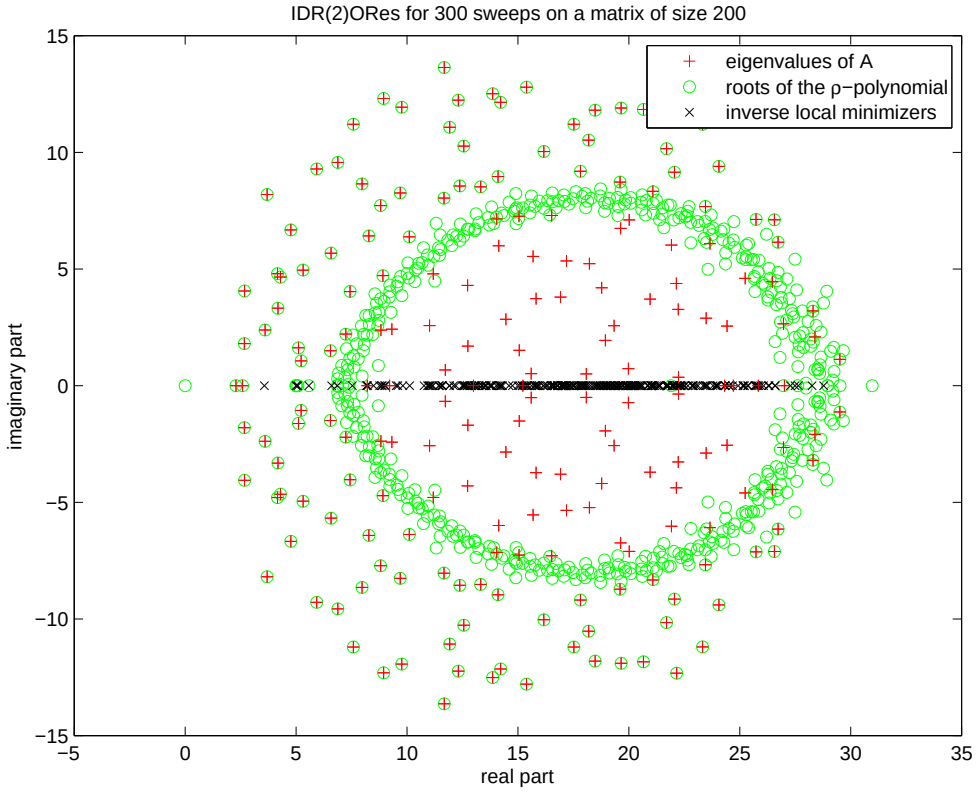


FIG. 6.8. *The formation of "pseudospectra": 300 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

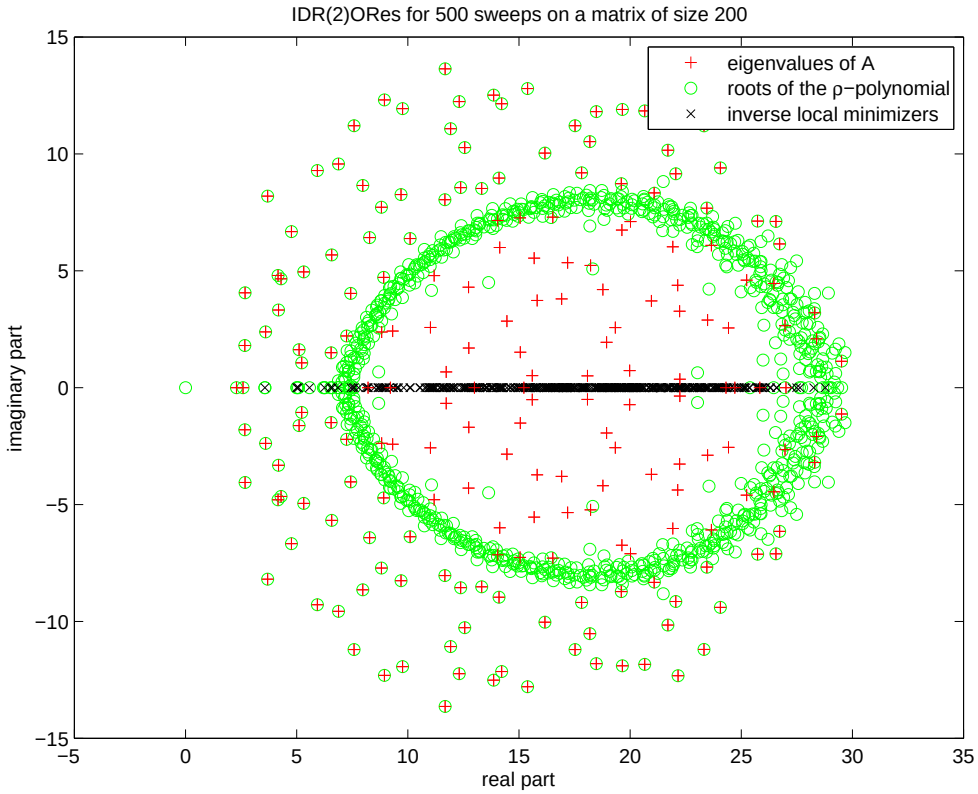


FIG. 6.9. *The formation of "pseudospectra": 500 sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

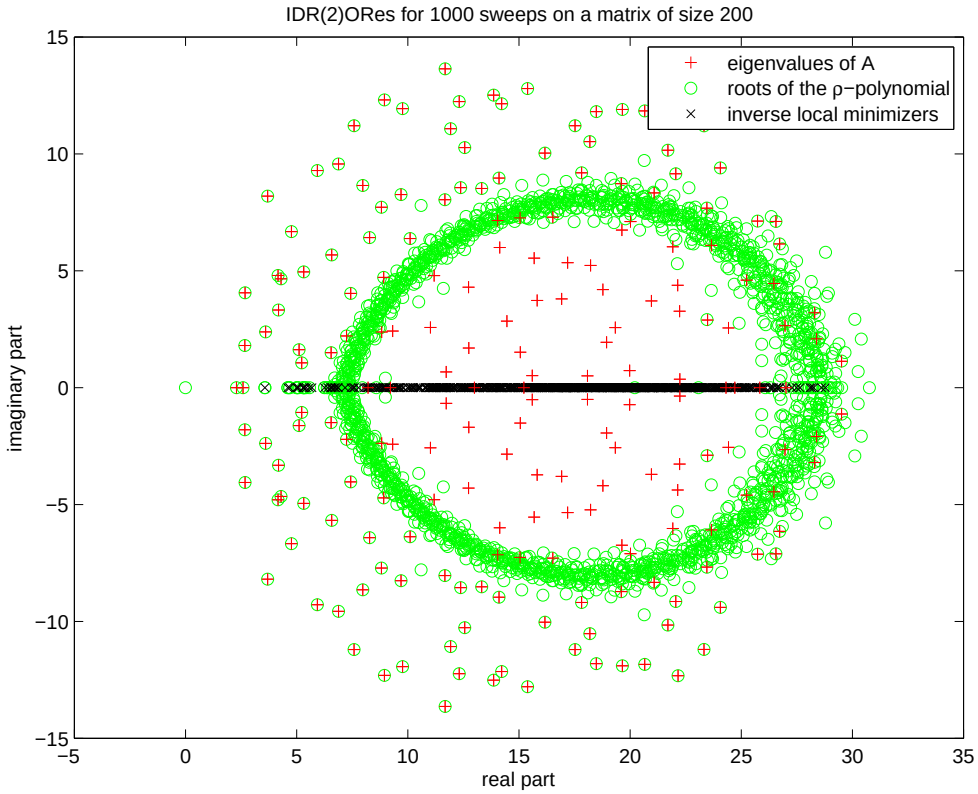


FIG. 6.10. *The formation of "pseudospectra":* 1000 *sweeps of* IDR(2)ORES *with a real orthonormal* $\mathbf{P} \in \mathbb{R}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located around the real minimizers on the real line. From this point on, we observe a finite precision Ritz value close to zero. Some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

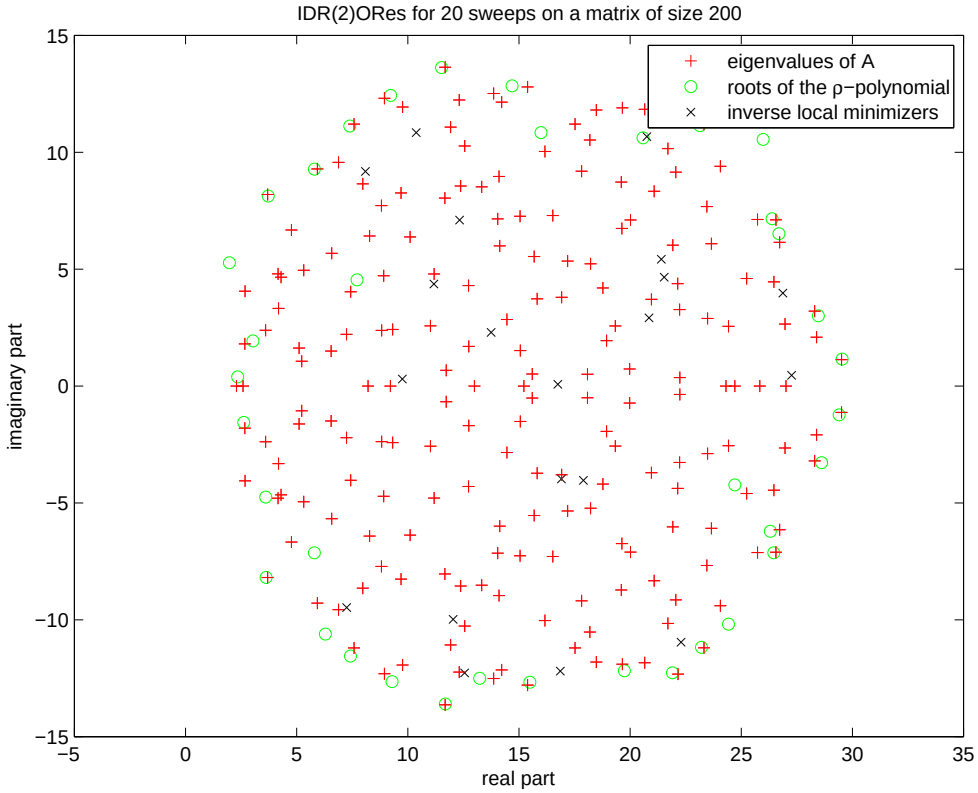


FIG. 6.11. *The formation of a "barrier": 20 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

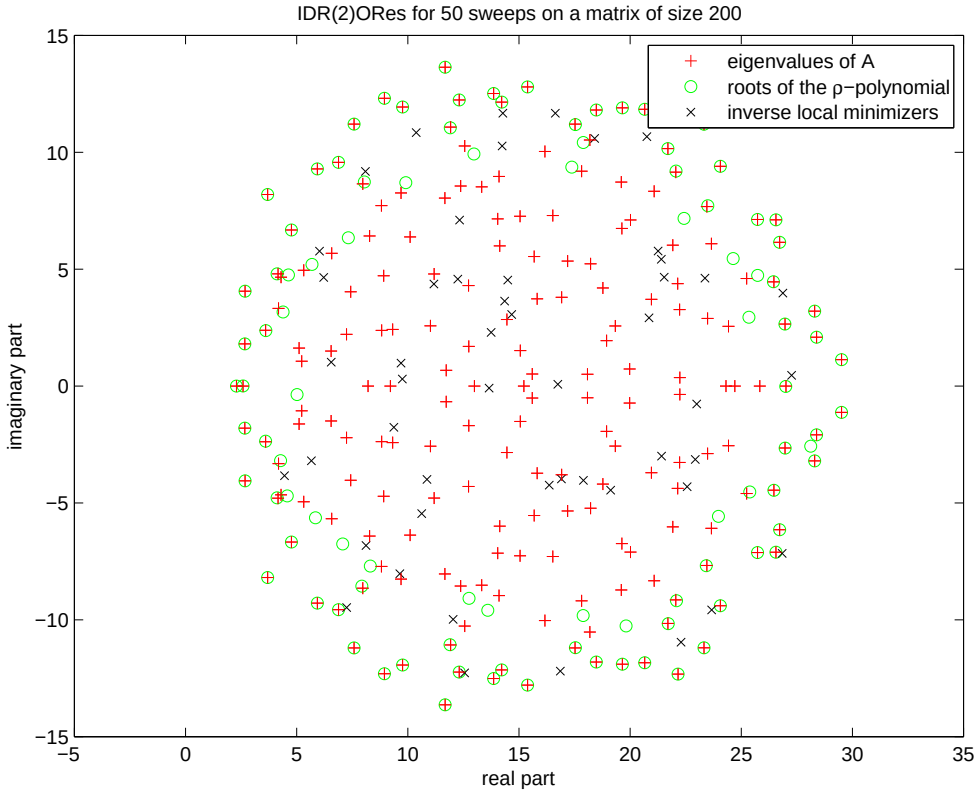


FIG. 6.12. *The formation of a "barrier": 50 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

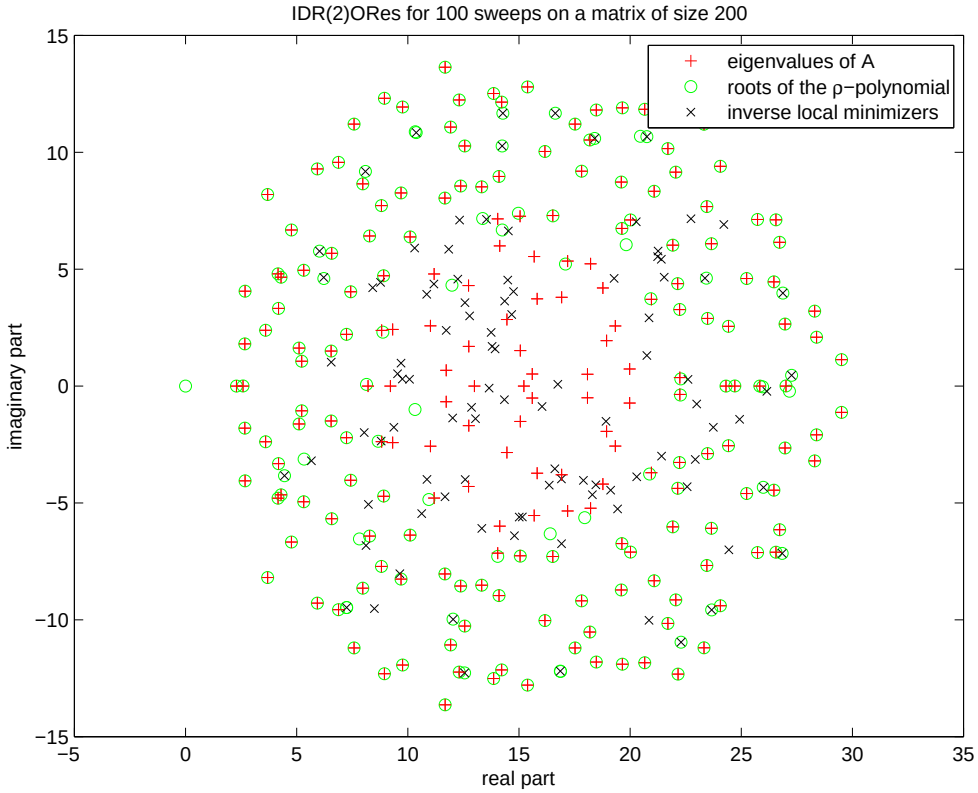


FIG. 6.13. *The formation of a "barrier": 100 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

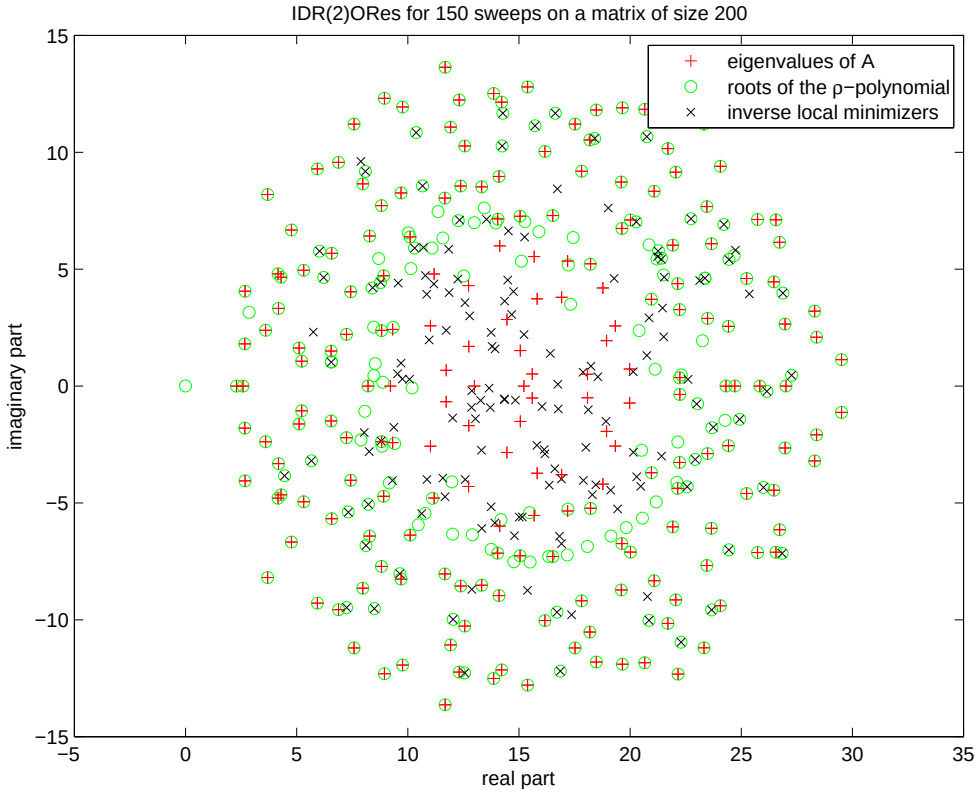


FIG. 6.14. *The formation of a "barrier": 150 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

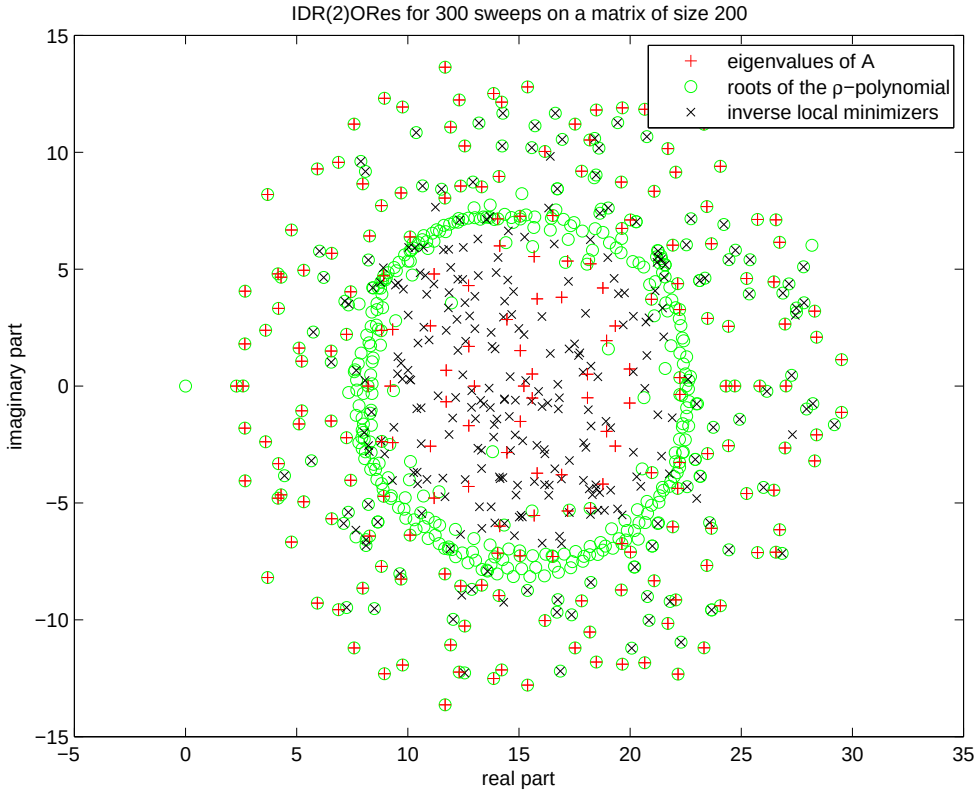


FIG. 6.15. *The formation of a "barrier": 300 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

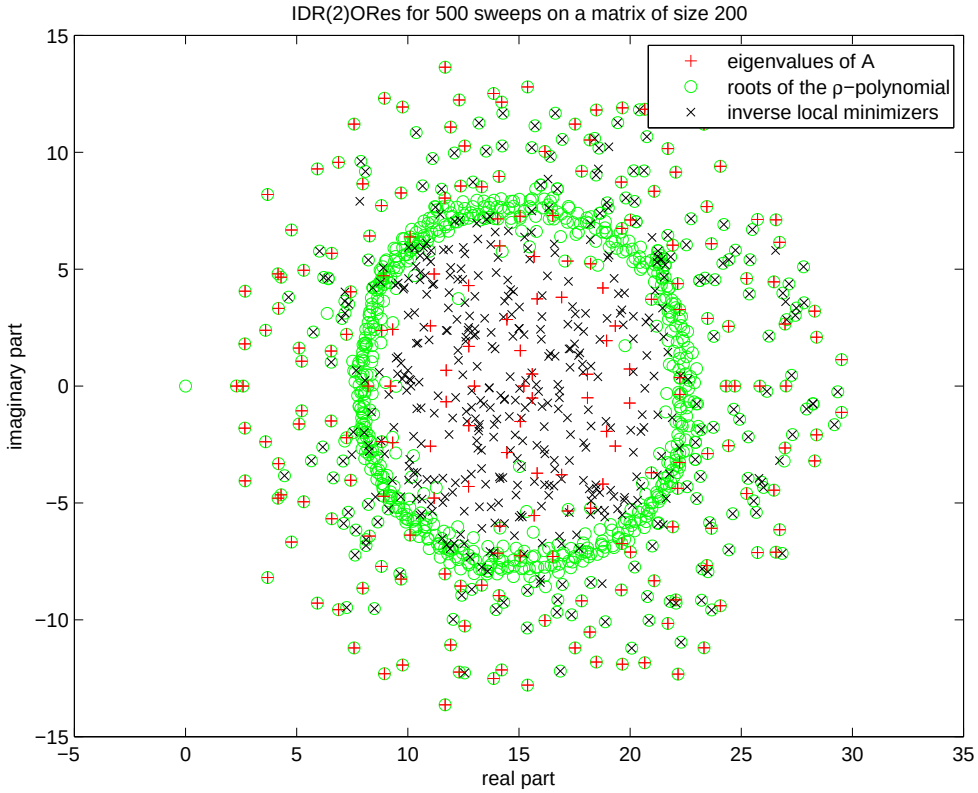


FIG. 6.16. *The formation of a "barrier": 500 sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

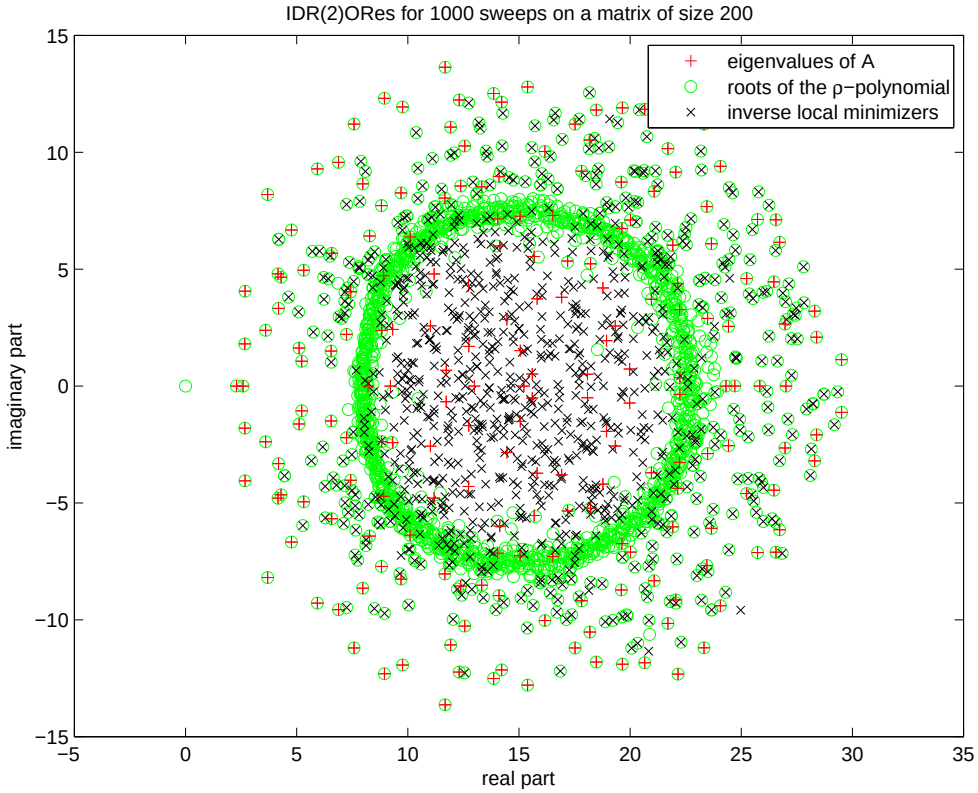


FIG. 6.17. *The formation of a "barrier":* 1000 *sweeps of* IDR(2)ORES *with a complex orthonormal* $\mathbf{P} \in \mathbb{C}^{200\times 2}$. *The eigenvalues of the matrix* $\mathbf{A} \in \mathbb{R}^{200\times 200}$ *are depicted with red plusses, the minimizers* $1/\omega_j$ *are depicted by black crosses, and the roots of the* ρ*-polynomial are depicted by green circles. This is a series of seven pictures. The first two pictures show behavior to be expected also in exact arithmetic. From the third picture onwards we observe the formation of a circle of finite precision Ritz values located somewhere in the region where complex minimizers occur most frequently. From this point on, we observe a finite precision Ritz value close to zero. In this complex case one can better observe that some of the roots of the* ρ*-polynomial approximate some previously computed minimizers* $1/\omega_j$.

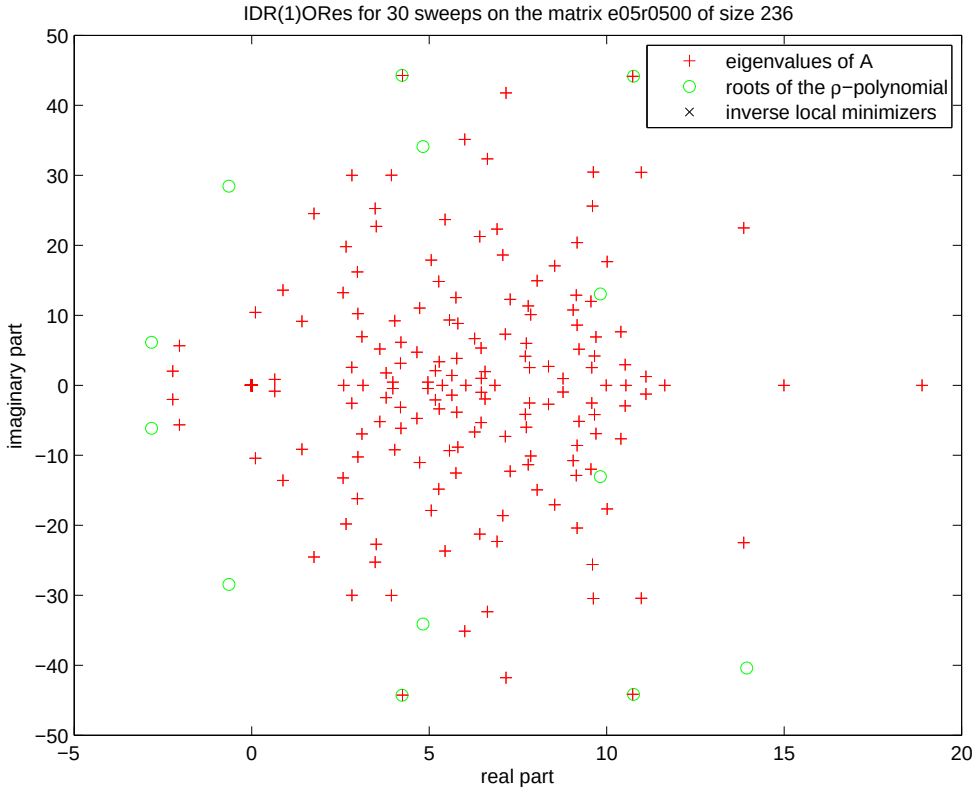


FIG. 6.18. *Approximation by Ritz values for* 60 *steps and* $s = 1$.

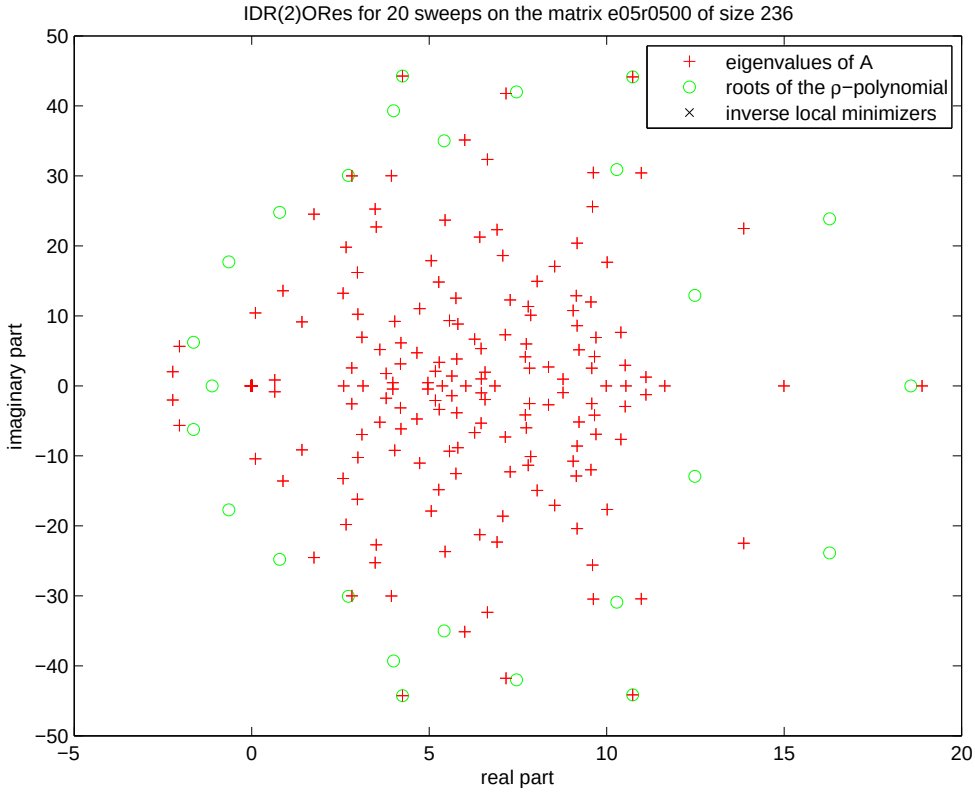


FIG. 6.19. *Approximation by Ritz values for* 60 *steps and* $s = 2$.

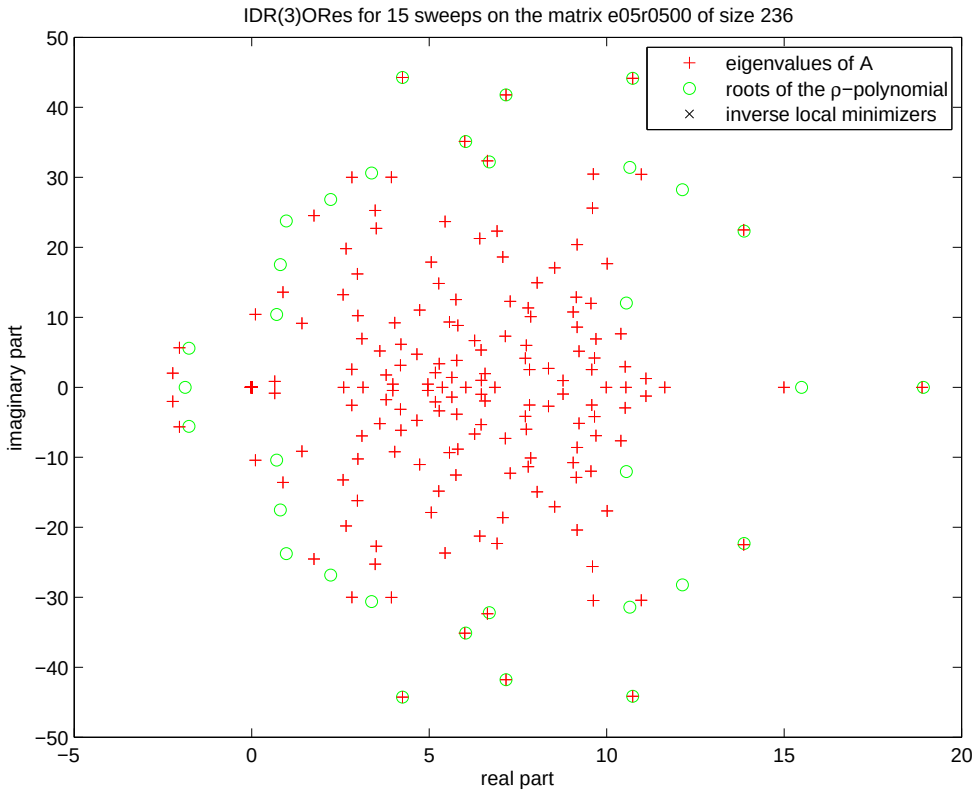


FIG. 6.20. *Approximation by Ritz values for* 60 *steps and* $s = 3$.

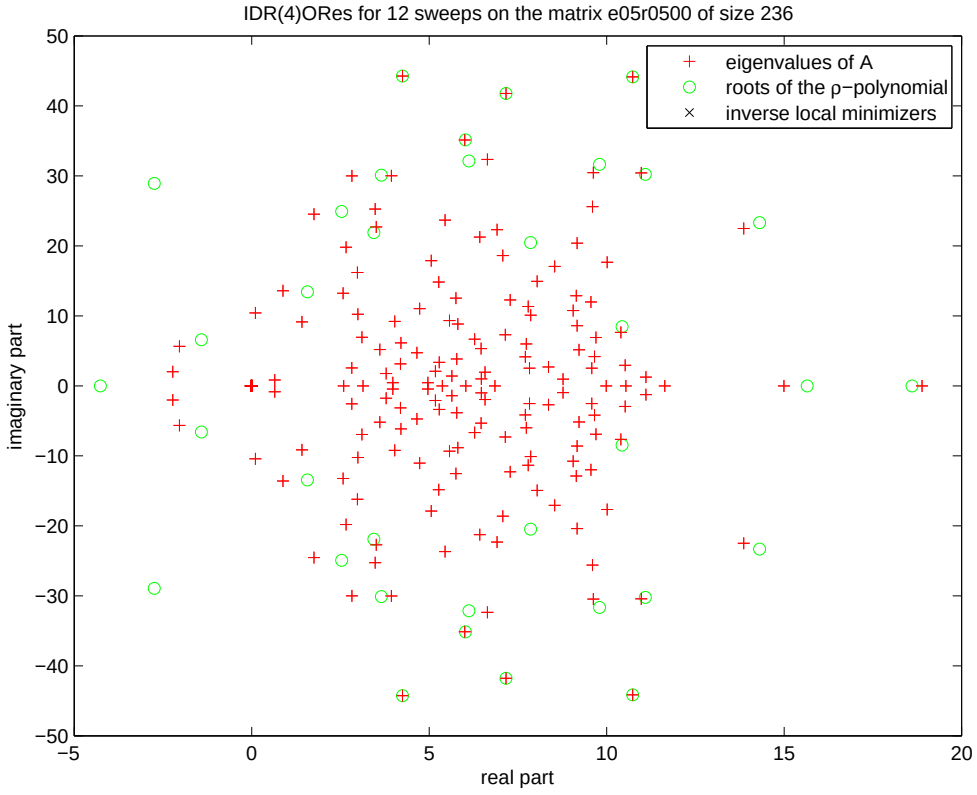


FIG. 6.21. *Approximation by Ritz values for* 60 *steps and* $s = 4$.

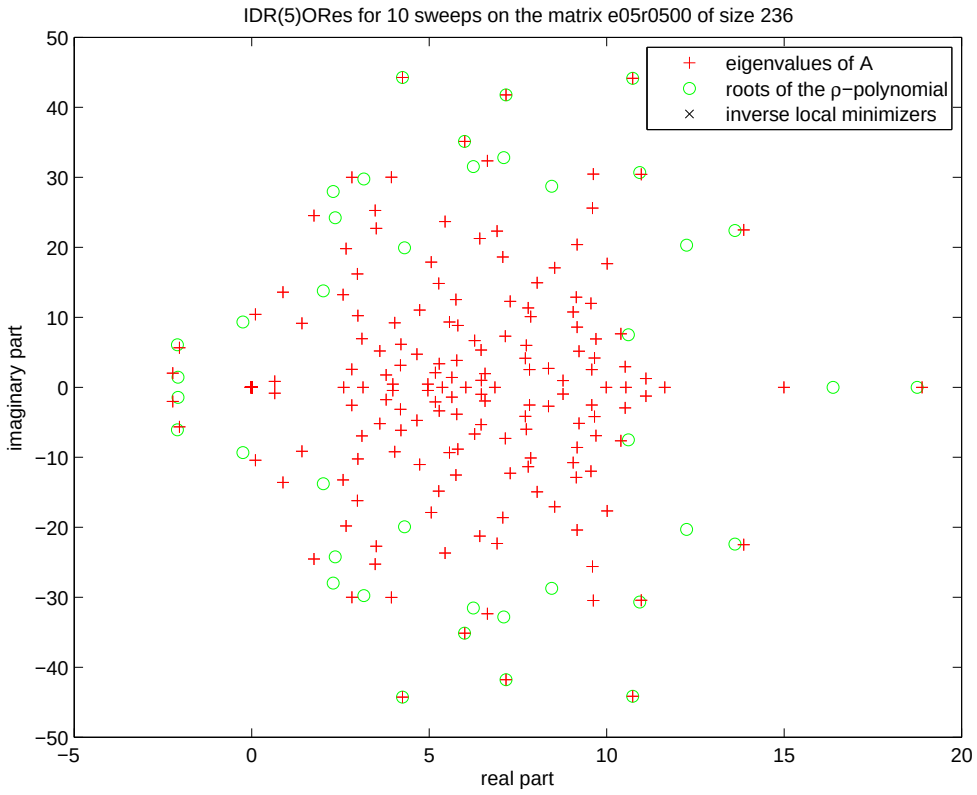


FIG. 6.22. *Approximation by Ritz values for* 60 *steps and* $s = 5$.

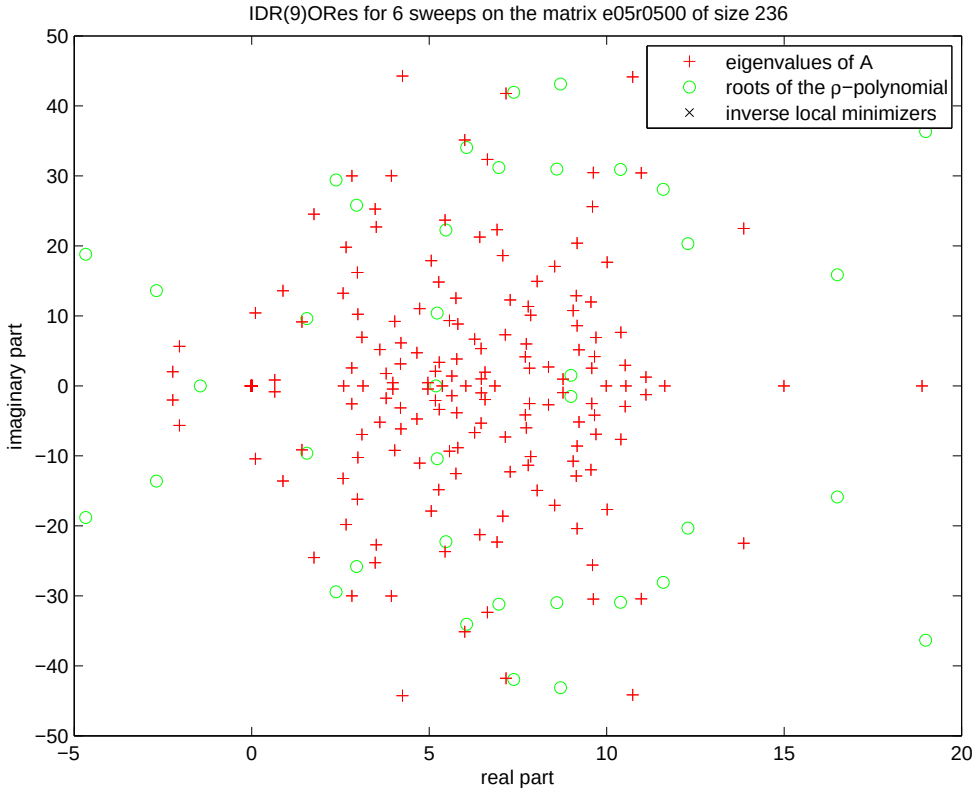


FIG. 6.23. *Approximation by Ritz values for* 60 *steps and* $s = 9$.

7. Source code listings. Here are the listings of the programs used for our numerical experiments.

Listing 1
matlab/IDREig_driver.m

```
%IDREIG_DRIVER is an example driver for the IDREig package.
%
%   The script IDREig_driver generates some matrices and
%   vectors and calls five different functions:
%
%     1) IDRORes_academic, an academic implementation of
%   the prototype IDR(s) [Sonneveld/van Gijzen, SISC 31, 2008].
%     2) IDRORes, the less academic implementation of the same
%   prototype IDR(s), which generates only the basic output
%   of IDRORes_academic, but saves workspace.
%     3) IDREig, a routine that presents how to extract
%   eigenvalue information from a run of IDRORes(_academic)
%   based on five different approaches outlined in the report
%   [Gutknecht/Zemke, 2010].
%     4) IDRRitzPurified, a routine that shows how to cheaply
%   estimate the accuracy of the eigenpair approximation.
%     5) compute_W, a routine that exemplifies how stable the
%   computation of a certain set of basis vectors useful for
%   the computation of certain Ritz vectors is for the chosen
%   matrix and starting vector.
%
%   See also: IDRORes, IDRORes_academic, IDREig, IDRRitzPurified.

%   Coypright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.00, Date: 2010/04/13 14:20:34 CEST.

n = 50;
A = randn(n)+(1+sqrt(n))*eye(n);
b = A*ones(n,1);
x0 = randn(n,1);
s = 3;
PH = orth(randn(n,s)+i*randn(n,s)).';
maxj = 20;
start = 1;

[x,resvec,uH,UT,C,omega,trueresvec,R] = ...
                    IDRORes_academic(A,b,x0,s,PH,maxj,start);
[x2,resvec2,uH2,UT2,C2,omega2] = ...
                    IDRORes(A,b,x0,s,PH,maxj,start);

if sum(abs([norm(x-x2) norm(resvec-resvec2) ...
            norm(full(uH-uH2)) norm(full(UT-UT2)) ...
            norm(C-C2) norm(omega-omega2)])) ~= 0
  error('Something went terribly wrong.')
end

colordef black;

figure(1)
steps = (s+1)*maxj;
semilogy(0:steps,trueresvec,'b-',...
         0:steps,resvec,'r-.');
axen = axis;
axis([0 steps axen(3:4)])
legend('true residual norm',...
       'computed residual norm')

[uSH,SR,uSM,PR,uDH,DR,uLM] = IDREig(uH,UT,C,omega,s,maxj);

SH = full(uSH(1:size(SR,1),:));
SM = full(uSM(1:size(SR,1),:));
DH = full(uDH(1:size(DR,1),:));
LM = full(uLM(1:size(DR,1),:));

if issparse(A)
  eigA = eigs(A,min(200,n/2),'SM');
else
```

```
  eigA = eig(A);
end

eigSP = eig(SH,full(SR));
eigSM = eig(SM);
eigPP = eig(SH,full(PR));
eigDP = eig(DH,full(DR));
eigLM = eig(LM);

figure(2)
plot(real(eigA),imag(eigA),'r+',...
     real(eigSP),imag(eigSP),'mo',...
     real(eigSM),imag(eigSM),'ms',...
     real(eigPP),imag(eigPP),'gd',...
     real(eigDP),imag(eigDP),'bv',...
     real(eigLM),imag(eigLM),'b^',...
     real(1./omega),imag(1./omega),'wx',...
     real(eigA),imag(eigA),'r+');
legend('eigenvalues of A',...
       'eigenvalues of the Sonneveld pencil',...
       'eigenvalues of the Sonneveld matrix',...
       'eigenvalues of the purified pencil',...
       'eigenvalues of the deflated pencil',...
       'eigenvalues of the deflated matrix',...
       'inverse local minimizers')

if maxj*(s+1) < 301

  lambda = max(eigA);
  [v,theta,acc,estconv,trueconv,allconv,backerr] = ...
     IDRRitzPurified(A,R,uSH,SR,PR,omega,resvec,lambda,s,maxj);

  normA = sqrt(norm(A,inf)*norm(A,1));
  figure(3)
  semilogy(1:steps,trueconv/normA,'b-',...
           1:steps,estconv/normA,'r-',...
           1:steps,backerr/normA,'y-',...
           1:steps,allconv/normA,'g-',...
           1:steps,trueconv/normA,'b-',...
           1:steps,estconv/normA,'r-',...
           1:steps,backerr/normA,'y-');
  axis([1 steps 1e-16 1e5])
  legend('true distance to eigenvalue',...
         'estimated rate of convergence',...
         'backward error of the Ritz pair',...
         'distances to other Ritz values')
  title(sprintf('convergence to the eigenvalue lambda = %g',lambda));

end

[RW,pnRW,W,pnW] = compute_W(A,LM,R(:,1),omega,resvec,s,maxj);

figure(4)
semilogy(1:steps,resvec(1:steps),'r-.',...
         1:steps,trueresvec(1:steps),'b-',...
         1:steps,pnW,'w-',...
         1:steps,pnRW,'g-',...
         1:steps,abs(resvec(1:steps)-pnRW'),'r.')
legend('updated residual norms',...
       'true residual norms',...
       'purified residual norms',...
       'recomputed residual norms',...
       'differences')
top = 10^(round(max(log10(resvec))));
axis([1,steps,1e-21*top,1e2*top]);
```

Listing 2
matlab/IDROres_academic.m

```
function [x,resvec,uH,UT,C,omega,trueresvec,R] ...
                        = IDROres_academic(A,b,x0,s,PH,maxj,start)

%IDROres_academic approximate solutions to Ax=b using original IDR.
%
%   x = IDROres_academic(A,b) computes an approximate solution x
%   to the linear system Ax = b using the IDR(s) prototype algorithm
%   by [Sonneveld/van Gijzen, SISC 31, 2008].
%
%   In the call
%   [x,resvec,uH,UT,C,omega,trueresvec,R] =
%                       IDROres_academic(A,b,x0,s,PH,maxj,start)
%   of IDR(s)ORes,
%
%     - x0    is the starting guess,
%     - s     is the number of left Lanczos vectors,
%     - PH    is the left Petrov-Galerkin matrix,
%     - maxj  is the number of inner sweeps
%                (comprising of s left and 1 right steps),
%     - start is one of:
%          1 == OrthoRes(1) = ORes,
%          2 == full OrthoRes,
%
%   and
%
%     - x       is the approximate solution,
%     - resvec  is the vector of the norms of the
%                  computed residuals,
%     - uH      is a sparse unreduced extended upper
%                  Hessenberg matrix,
%     - UT      is a sparse upper triangular matrix,
%     - C       is a (s x n)-matrix that contains in the
%                  columns the coefficients obtained in
%                  the orthogonalization against P,
%     - omega   contains the local minimizers,  and
%     - trueresvec is the vector of the true residuals.
%
%   The extended pencil (uH,UT) is defined by the starting
%   procedure 1 or 2. This pencil, together with the IDR
%   quantities contained in the matrix C and the vector
%   omega, defines the underlying Hessenberg decomposition
%   for the residuals that can be used to compute eigenvalue
%   approximations.
%
%   This is an academic implementation with two drawbacks:
%     o The computation of the true residual norm is based on
%       a second matrix-vector multiplication in every step,
%     o All the residuals are stored in the matrix R.
%   For these reasons, in the non-academic variant IDROres
%   we do not compute true residual norms and do not store
%   the residual vectors.
%
%   See also: IDROres, IDREig, IDREig_driver, IDRRitzPurified.

%   Coyright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.03, Date: 2010/03/10 16:12:24 CEST.

%% TODO: implement full GMRes

if nargin < 1
  error('No system matrix was given.');
end

[n,m] = size(A);

if nargin < 2
  error('No right-hand side was given.');
end

```

```
70 if nargin < 3
71   x0 = randn(n,1);
72 end
73
74 if nargin < 4
75   s = min(n,4);
76 end
77
78 if nargin < 5
79   PH = orth(randn(n,s)).';
80 end
81
82 if nargin < 6
83   maxj = min(round((2*(n+1))/(s+1)),round(300/(s+1)));
84 end
85
86 if nargin < 7
87   start = 1;
88 end
89
90 maxsteps = maxj*(s+1)+1;
91 R = zeros(n,maxsteps);
92 resvec = zeros(1,maxsteps);
93 trueresvec = zeros(1,maxsteps);
94
95 x = x0;
96 r = b-A*x0;
97 R(:,1) = r;
98
99 normr = norm(r);
100 resvec(1) = normr;
101 trueresvec(1) = normr;
102
103 UT = sparse(s);
104 uH = sparse(s+1,s);
105
106   nablaR = zeros(n,s);
107   nablaX = zeros(n,s);
108 PHnablaR = zeros(s,s);
109
110 % ORes, i.e., OrthoRes(1)
111 if start == 1
112
113   for k = 1:s
114
115     Ar = A*r;
116     tomega = dot(Ar,r)/dot(Ar,Ar);
117     UT(k,k) = tomega;
118
119     nablaX(:,k) = tomega*r;
120     x = x + nablaX(:,k);
121
122     nablaR(:,k) = -tomega*Ar;
123     r = r - tomega*Ar;
124     R(:,k+1) = r;
125
126     PHnablaR(:,k) = PH*nablaR(:,k);
127
128     resvec(k+1) = norm(r);
129     trueresvec(k+1) = norm(b-A*x);
130
131   end
132   uH = spdiags(ones(s+1,1)*[-1 1],[-1 0],s+1,s);
133
134 % full OrthoRes, i.e., OrthoRes(s+1)
135 elseif start == 2
136
137   X = zeros(n,s+1);
138   rHr = zeros(s+1,1);
139
140   X(:,1) = x;
141   rHr(1) = dot(r,r);
```

```
142
143   for k = 1:s
144
145     Ar = A*R(:,k);
146
147     uH(1:k,k) = (R(:,1:k)'*Ar)./rHr(1:k);
148     uH(k+1,k) = -sum(uH(1:k,k));
149
150     R(:,k+1) =  (Ar-R(:,1:k)*uH(1:k,k))/uH(k+1,k);
151     X(:,k+1) = -(R(:,k)+X(:,1:k)*uH(1:k,k))/uH(k+1,k);
152
153     resvec(k+1) = norm(R(:,k+1));
154     trueresvec(k+1) = norm(b-A*X(:,k+1));
155     rHr(k+1) = resvec(k+1)^2;
156
157   end
158
159   UT = speye(s);
160   r = R(:,s+1);
161   x = X(:,s+1);
162
163   nablaR = R(:,2:s+1)-R(:,1:s);
164   nablaX = X(:,2:s+1)-X(:,1:s);
165   PHnablaR = PH*nablaR;
166
167 % other methods, e.g., full explicit GMRes, i.e., GMRes(s+1)
168 else
169
170     disp('not implemented yet')
171
172 end
173
174 PHr = PH*r;
175 normr = norm(r);
176
177 sp1 = s+1;
178 step = sp1+1;
179 colnum = 1;
180
181 indexR = (1:s)';
182 indexC = (1:s)';
183
184 C = zeros(s,(maxj-1)*(s+1));
185 omega = zeros(1,maxj);
186
187 for j = 1:maxj-1
188
189   c = (PHnablaR)\(PHr);
190   C(:,colnum) = c(indexC);
191
192   oldest = indexR == 1;
193   indexR = circshift(indexR, 1);
194   indexC = circshift(indexC,-1);
195
196   update = -nablaR*c;
197   v = r + update;
198   Av = A*v;
199   omega(j) = dot(Av,v)/dot(Av,Av);
200
201   nablaX(:,oldest) = -nablaX*c + omega(j)*v;
202   x = x + nablaX(:,oldest);
203
204   nablaR(:,oldest) = update - omega(j)*Av;
205   r = r + nablaR(:,oldest);
206   R(:,step) = r;
207
208   PHnablaR(:,oldest) = PH*nablaR(:,oldest);
209   PHr = PHr + PHnablaR(:,oldest);
210
211   normr = norm(r);
212   resvec(step) = normr;
213   trueresvec(step) = norm(b-A*x);
```

```
214
215   for k = 1:s
216
217     c = (PHnablaR)\(PHr);
218     C(:,colnum+k) = c(indexC);
219
220     oldest = indexR == 1;
221     indexR = circshift(indexR, 1);
222     indexC = circshift(indexC,-1);
223
224     update = -nablaR*c;
225     v = r + update;
226
227     nablaX(:,oldest) = -nablaX*c + omega(j)*v;
228     x = x + nablaX(:,oldest);
229
230     nablaR(:,oldest) = -A*nablaX(:,oldest);
231     r = r + nablaR(:,oldest);
232     R(:,step+k) = r;
233
234     PHnablaR(:,oldest) = PH*nablaR(:,oldest);
235     PHr = PHr + PHnablaR(:,oldest);
236
237     normr = norm(r);
238     resvec(step+k) = normr;
239     trueresvec(step+k) = norm(b-A*x);
240
241   end
242
243   colnum = colnum + sp1;
244   step = step + sp1;
245
246 end
247
248 c = (PHnablaR)\(PHr);
249 C(:,colnum) = c(indexC);
250
251 oldest = indexR == 1;
252 indexC = circshift(indexC,-1);
253
254 update = -nablaR*c;
255 v = r + update;
256 Av = A*v;
257 omega(j+1) = dot(Av,v)/dot(Av,Av);
258
259 nablaX(:,oldest) = -nablaX*c + omega(j+1)*v;
260 x = x + nablaX(:,oldest);
261
262 nablaR(:,oldest) = update - omega(j+1)*Av;
263 r = r + nablaR(:,oldest);
264 R(:,step) = r;
265
266 normr = norm(r);
267 resvec(step) = normr;
268 trueresvec(step) = norm(b-A*x);
```

Listing 3
matlab/IDROres.m

```
function [x,resvec,uH,UT,C,omega] = IDROres(A,b,x0,s,PH,maxj,start)

%IDROres approximate solutions to Ax=b using original IDR.
%
%   x = IDROres(A,b) computes an approximate solution x to the
%   linear system Ax = b using the IDR(s) prototype algorithm
%   by [Sonneveld/van Gijzen, SISC 31, 2008].
%
%   In the call
%   [x,resvec,uH,UT,C,omega] =
%                          IDROres(A,b,x0,s,PH,maxj,start)
%   of IDR(s)ORes,
%
%     - x0    is the starting guess,
%     - s     is the number of left Lanczos vectors,
%     - PH    is the left Petrov-Galerkin matrix,
%     - maxj  is the number of inner sweeps
%               (comprising of s left and 1 right steps),
%     - start is one of:
%          1 == OrthoRes(1) = ORes,
%          2 == full OrthoRes,
%
%   and
%
%     - x       is the approximate solution,
%     - resvec  is the vector of the norms of the
%                 computed residuals,
%     - uH      is a sparse unreduced extended upper
%                 Hessenberg matrix,
%     - UT      is a sparse upper triangular matrix,
%     - C       is a (s x n)-matrix that contains in the
%                 columns the coefficients obtained in
%                 the orthogonalization against P,
%     - omega   contains the local minimizers.
%
%   The extended pencil (uH,UT) is defined by the starting
%   procedure 1 or 2. This pencil, together with the IDR
%   quantities contained in the matrix C and the vector
%   omega, defines the underlying Hessenberg decomposition
%   for the residuals that can be used to compute eigenvalue
%   approximations.
%
%   There is also an academic implementation with two drawbacks:
%     o The computation of the true residual norm is based on
%       a second matrix-vector multiplication in every step,
%     o All the residuals are stored in the matrix R.
%   The academic variant is named IDROres_academic.
%
%   See also: IDROres_academic, IDREig, IDREig_driver, IDRRitzPurified.

%   Coypright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.03, Date: 2010/03/10 16:31:43 CEST.

%% TODO: implement full GMRes

if nargin < 1
  error('No system matrix was given.');
end

[n,m] = size(A);

if nargin < 2
  error('No right-hand side was given.');
end

if nargin < 3
  x0 = randn(n,1);
end

```

```
70  if nargin < 4
71    s = min(n,4);
72  end
73
74  if nargin < 5
75    PH = orth(randn(n,s)).';
76  end
77
78  if nargin < 6
79    maxj = min(round((2*(n+1))/(s+1)),round(300/(s+1)));
80  end
81
82  if nargin < 7
83    start = 1;
84  end
85
86  resvec = zeros(1,maxj*(s+1)+1);
87
88  x = x0;
89  r = b-A*x0;
90
91  resvec(1) = norm(r);
92
93  UT = sparse(s);
94  uH = sparse(s+1,s);
95
96    nablaR = zeros(n,s);
97    nablaX = zeros(n,s);
98  PHnablaR = zeros(s,s);
99
100 % ORes, i.e., OrthoRes(1)
101 if start == 1
102
103   for k = 1:s
104
105     Ar = A*r;
106     tomega = dot(Ar,r)/dot(Ar,Ar);
107     UT(k,k) = tomega;
108
109     nablaX(:,k) = tomega*r;
110     x = x + nablaX(:,k);
111
112     nablaR(:,k) = -tomega*Ar;
113     r = r - tomega*Ar;
114
115     PHnablaR(:,k) = PH*nablaR(:,k);
116
117     resvec(k+1) = norm(r);
118
119   end
120   uH = spdiags(ones(s+1,1)*[-1 1],[-1 0],s+1,s);
121
122 % full OrthoRes, i.e., OrthoRes(s+1)
123 elseif start == 2
124
125   R = zeros(n,s+1);
126   X = zeros(n,s+1);
127   rHr = zeros(s+1,1);
128
129   R(:,1) = r;
130   X(:,1) = x;
131   rHr(1) = dot(r,r);
132
133   for k = 1:s
134
135     Ar = A*R(:,k);
136
137     uH(1:k,k) = (R(:,1:k)'*Ar)./rHr(1:k);
138     uH(k+1,k) = -sum(uH(1:k,k));
139
140     R(:,k+1) =  (Ar-R(:,1:k)*uH(1:k,k))/uH(k+1,k);
141     X(:,k+1) = -(R(:,k)+X(:,1:k)*uH(1:k,k))/uH(k+1,k);
```

```
142
143     resvec(k+1) = norm(R(:,k+1));
144     rHr(k+1) = resvec(k+1)^2;
145
146   end
147
148   UT = speye(s);
149   r = R(:,s+1);
150   x = X(:,s+1);
151
152   nablaR = R(:,2:s+1)-R(:,1:s);
153   nablaX = X(:,2:s+1)-X(:,1:s);
154   PHnablaR = PH*nablaR;
155
156 % other methods, e.g., full explicit GMRes, i.e., GMRes(s+1)
157 else
158
159     disp('not implemented yet')
160
161 end
162
163 PHr = PH*r;
164 normr = norm(r);
165
166 sp1 = s+1;
167 step = sp1+1;
168 j = 1;
169 colnum = 1;
170
171 indexR = (1:s)';
172 indexC = (1:s)';
173
174 C = zeros(s,(maxj-1)*(s+1));
175 omega = zeros(1,maxj);
176
177 for j = 1:maxj-1
178
179   c = (PHnablaR)\(PHr);
180   C(:,colnum) = c(indexC);
181
182   oldest = indexR == 1;
183   indexR = circshift(indexR, 1);
184   indexC = circshift(indexC,-1);
185
186   update = -nablaR*c;
187   v = r + update;
188   Av = A*v;
189   omega(j) = dot(Av,v)/dot(Av,Av);
190
191   nablaX(:,oldest) = -nablaX*c + omega(j)*v;
192   x = x + nablaX(:,oldest);
193
194   nablaR(:,oldest) = update - omega(j)*Av;
195   r = r + nablaR(:,oldest);
196
197   PHnablaR(:,oldest) = PH*nablaR(:,oldest);
198   PHr = PHr + PHnablaR(:,oldest);
199
200   normr = norm(r);
201   resvec(step) = normr;
202
203   for k = 1:s
204
205     c = (PHnablaR)\(PHr);
206     C(:,colnum+k) = c(indexC);
207
208     oldest = indexR == 1;
209     indexR = circshift(indexR, 1);
210     indexC = circshift(indexC,-1);
211
212     update = -nablaR*c;
213     v = r + update;
```

```
214
215     nablaX(:,oldest) = -nablaX*c + omega(j)*v;
216     x = x + nablaX(:,oldest);
217
218     nablaR(:,oldest) = -A*nablaX(:,oldest);
219     r = r + nablaR(:,oldest);
220
221     PHnablaR(:,oldest) = PH*nablaR(:,oldest);
222     PHr = PHr + PHnablaR(:,oldest);
223
224     normr = norm(r);
225     resvec(step+k) = normr;
226
227   end
228
229   colnum = colnum + sp1;
230   step = step + sp1;
231
232 end
233
234 c = (PHnablaR)\(PHr);
235 C(:,colnum) = c(indexC);
236
237 oldest = indexR == 1;
238 indexC = circshift(indexC,-1);
239
240 update = -nablaR*c;
241 v = r + update;
242 Av = A*v;
243 omega(j+1) = dot(Av,v)/dot(Av,Av);
244
245 nablaX(:,oldest) = -nablaX*c + omega(j+1)*v;
246 x = x + nablaX(:,oldest);
247
248 nablaR(:,oldest) = update - omega(j+1)*Av;
249 r = r + nablaR(:,oldest);
250
251 normr = norm(r);
252 resvec(step) = normr;
```

Listing 4
matlab/IDREig.m

```
function [uSH,SR,uSM,PR,uDH,DR,uLM,D] = IDREig(uH,UT,C,omega,s,maxj);

%IDREIG constructs matrices for eigenvalue computations for IDRORes.
%
%   [uSH,SR] = IDREig(uH,UT,C,omega,s,maxj) computes the sparse
%   extended Sonneveld pencil from the quantities returned by a
%   run of the IDRORes implementation of the prototype IDR(s) of
%   [Sonneveld/van Gijzen, SISC 31, 2008].
%
%   In the call
%
%   [uSH,SR,uSM,PR,uDH,DR,uLM] = IDREig(uH,UT,C,omega,s,maxj)
%
%     - uH is the leading extended Hessenberg matrix defined by
%       the starting procedure,
%     - UT is the upper triangular matrix defined by the starting
%       procedure,
%     - C contains the coefficients of the oblique projections
%       onto the differences of the last residuals to make the
%       resulting vector orthogonal to the shadow vectors in P,
%     - omega contains the vector of all chosen omega,
%     - s is a non-negative integer which defines the size of
%       the shadow space,  and
%     - maxj gives the number of sweeps.
%
%  From the output,
%
%     (uSH,SR) is the sparse extended Sonneveld pencil,
%      uSM     is the sparse extended Sonneveld matrix,
%     (uSH,PR) is the sparse extended purified pencil,
%     (uDH,DR) is the sparse extended deflated pencil,  and
%      uLM     is the sparse extended Lanczos-BiORes(s,1) matrix.
%
%   See also: IDRORes, IDRORes_academic, IDREig_driver, IDRRitzPurified.

%   Coypright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.00, Date: 2010/04/13 14:33:02 CEST.

j = maxj-1;
n = (s+1)*j+1;

o = zeros(1,n);
e = ones(1,n);

band_Yo = diff([o;C;e;o]);
band_Y  = diff([o;C;e]);

diag_omega = diag(kron(diag(sparse([1,omega(1:j)])),speye(s+1)));
D = spdiags([diag_omega(2:end);omega(end)],0,n+s,n+s);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %%
%%        compute the extended Sonneveld pencil (uSH,SR)                      %%
%%                                                                            %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

uSH = [[sparse(uH);sparse(n,s)],...
        spdiags(fliplr(band_Yo.'),[-(s+1):0],n+s+1,n)];
SR  = [[sparse(UT);sparse(n,s)],...
        spdiags(fliplr(band_Y.'), [-s:0],n+s,n)]*D;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %%
%%        compute the extended Sonneveld matrix uSM                           %%
%%                                                                            %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

uSM = uSH/SR;

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %%
%%       compute the matrix PR of the purified pencil                         %%
%%                                                                            %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

band_U = band_Y;
band_U(repmat(fliplr(triu(logical(ones(s+1)))),1,j)) = 0;
band_U(:,n) = 0;
PR = [[sparse(UT);sparse(n,s)],...
       spdiags(fliplr(band_U.'),[-s:0],n+s,n)]*D;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %%
%%       compute the deflated purified pencil (uDH,DR)                        %%
%%                                                                            %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

G = speye(n+s);
indicesG = (1:maxj).*(s+1);
for refine = 1:2
  G(indicesG,:) = -uSH(indicesG,indicesG)\uSH(indicesG,:);
  G(indicesG,indicesG) = speye(maxj);
  uSHG = uSH*G;
end
indicesD = (1:n+s) ~= (s+1)*floor((1:n+s)./(s+1));

uDH = uSHG([indicesD logical(1)],indicesD);
DR = PR(indicesD,indicesD);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %%
%%       compute the BiORes(s,1) matrix uLM                                   %%
%%                                                                            %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

uLM = uDH/DR;
```

Listing 5
matlab/IDRRitzPurified.m

```
function [v,theta,acc,estconv,trueconv,allconv,backerr] = ...
             IDRRitzPurified(A,R,uSH,SR,PR,omega,resvec,lambda,s,maxj)

%IDRRITZPURIFIED is a howto: estimate accuracy without long vectors.
%
%   [v,theta,acc,estconv,trueconv,allconv,backerr] = ...
%         IDRRitzPurified(A,R,uSH,SR,PR,omega,resvec,lambda,s,maxj)
%
%   computes with given
%
%    A         - the square system matrix,
%    R         - the matrix of all IDROres residuals,
%    (uSH,SR)  - extended Sonneveld pencil,
%    (uSH,PR)  - extended purified pencil,
%    omega     - vector of all chosen omega,
%    resvec    - vector of norms of all IDROres residuals,
%    lambda    - eigenvalue of A of interest,
%    s         - parameter 's' of IDR(s)ORes,  and
%    maxj      - number of sweeps of IDROres
%
%  the values
%
%    v - normalized eigenvector of the purified pencil,
%    theta - corresponding Ritz value,
%    acc - final convergence estimator,
%    estconv - vector of convergence estimators of all steps,
%    trueconv - vector of smallest distances to lambda of all steps,
%    allconv - matrix of all distances to lambda of all steps,  and
%    backerr - vector of backward errors of best Ritz pairs of all
%      steps.
%
%   See also: IDROres, IDROres_academic, IDREig, IDREig_driver.

%   Coypright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.01, Date: 2010/04/13 14:57:36 CEST.

n = (s+1)*maxj;
omega_theta = ones(n,1);
allconv = zeros(n);
estconv = zeros(n,1);
trueconv = zeros(n,1);
backerr = zeros(n,1);

for i = 1:n

  %%% compute the eigenpairs of the purified pencil
  [S,Theta] = eig(full(uSH(1:i,1:i)),full(PR(1:i,1:i)));
  dtheta = diag(Theta).';

  %%% sort according to the distance to the given lambda
  [allconv(i,1:i),index] = sort(abs(lambda-dtheta));
  theta = dtheta(index(1));
  trueconv(i) = min(allconv(i,1:i));

  %%% compute the scaling by polynomial terms
  for k = s+1:i
    omega_theta(k) = prod(1-(omega(1:floor(k/(s+1))).*theta));
  end

  %%% compute the approximate eigenvector of the Sonneveld pencil
  v = S(:,index(1))./omega_theta(1:i);

  %%% still cheap: refine using Sonneveld upper triangular matrix
  w = SR(1:i,1:i)*v;

  %%% costly: prolongation using the matrix of residual vectors
  w = R(:,1:i)*w;

  %%% costly: computation of the backward error
```

```
70    w = w/norm(w);
71    backerr(i) = norm(A*w-theta*w);
72
73    %%% normalize the short vector at hand
74    v = v/norm(v);
75
76    %%% estimate cheaply the rate of convergence
77    estconv(i) = resvec(i+1)*abs(uSH(i+1,i)*v(i));
78
79  end
80
81  acc = estconv(n);
```

Listing 6
matlab/compute_W.m

```
function [RW,pnRW,W,pnW] = compute_W(A,L,r,omega,resvec,s,maxj)

%COMPUTE_W computes the basis vectors in W for IDRORes.
%
%   [RW,pnRW,W,pnW] = compute_W(A,L,r,omega,resvec,s,maxj)
%   computes the matrix W consisting column-wise of the basis
%   vectors of the BiORes(s,1)-process underlying IDR(s)ORes.
%   This is done using the Lanczos-BiORes(s,1) matrix computed
%   from IDREig with the information obtained and stored by
%   IDRORes_academic.
%
%   On input,
%
%      A      - denotes the quadratic system matrix,
%      L      - denotes the Lanczos-BiORes(s,1) matrix
%               computed by IDREig,
%      r      - is the starting residual used for IDRORes,
%      omega  - is the vector of all omega chosen in IDRORes,
%      resvec - contains the norms of the updated residuals,
%      s      - denotes the parameter 's' of IDR(s)ORes, and
%      maxj   - denotes the number of sweeps carried out.
%
%   On output,
%
%      RW   - contains the recomputed residuals r,
%      pnRW - is a prolonged vector of norms of these recomputed
%             residuals,
%      W    - contains the computed Lanczos-BiORes(s,1) residual
%             vectors, and
%      pnW  - is a prolonged vector of norms of these computed
%             Lanczos-BiORes(s,1) residuals.
%
%   See also: IDRORes_academic, IDREig, IDREig_driver.

%   Coypright 2009-2010 Jens-Peter M. Zemke.
%   Version 1.00, Date: 2010/04/13 14:01:23 CEST.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                     %%
%%   use the OrthoRes-Lanczos-matrix for the computation               %%
%%              of the residual polynomials                            %%
%%                                                                     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = maxj*s;
n = size(A,1);
maxit = (s+1)*maxj;

W = zeros(n,m+1);

W(:,1) = r;
nW = eye(m,1)*norm(W(:,1));

for j = 1:m-1
  W(:,j+1) = (A*W(:,j)-W(:,1:j)*L(1:j,j))/L(j+1,j);
  nW(j+1) = norm(W(:,j+1));
end

indvec = ((1:maxit) ~= (s+1)*floor((1:maxit)./(s+1))) ~= 0;

%% prolong nW

pnW = zeros(maxit,1);
pnW(indvec) = nW;

%%% compute R by Omega_j(A)*W by a Horner scheme

RW = W;
nRW = nW;
```

```
70
71 for j = 1:maxj-1
72   RW(:,j*s+1:m) = RW(:,j*s+1:m)-omega(j)*A*RW(:,j*s+1:m);
73   for k = 1:s
74     nRW(j*s+k) = norm(RW(:,j*s+k));
75   end
76 end
77
78 pnRW = zeros(maxit,1);
79 pnRW(indvec) = nRW;
```

REFERENCES

[1] Kuniyoshi Abe and Gerard L. G. Sleijpen, *BiCGStab2 and BiCG×MR2 variants of the IDR(s) method for solving linear systems*, technical report, Faculty of Economics and Information, Gifu Shotoku University, Nakauzura, Gifu 500-8288, Japan, 2010. Publication in preparation.

[2] ———, *Global polynomials and induced dimension reduction*, technical report, Faculty of Economics and Information, Gifu Shotoku University, Nakauzura, Gifu 500-8288, Japan, 2010. Publication in preparation.

[3] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, *A Lanczos-type method for multiple starting vectors*, Math. Comp., (2000), pp. 1577–1601. Received Oct. 10, 1996, electr. publ. May 20, 1999.

[4] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29. Received May 15, 1950.

[5] Steven F. Ashby, Thomas A. Manteuffel, and Paul E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568. Received Apr. 14, 1988.

[6] Zhaojun Bai, David Day, and Qiang Ye, *ABLE: an adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082 (electronic). Received Mar. 4, 1997.

[7] Charles George Broyden and Maria Teresa Vespucci, *Krylov solvers for linear algebraic systems: Krylov solvers*, vol. 11 of Studies in Computational Mathematics, Elsevier B. V., Amsterdam, 2004.

[8] Thomas Ericsson, *On the eigenvalues and eigenvectors of Hessenberg matrices*, Numerical Analysis Group, Göteborg, Report 10, Chalmers University of Technology and the University of Göteborg, Department of Computer Sciences, S-412 96 Göteborg, Sweden, June 1990. Online available at http://www.cs.chalmers.se/pub/num_analysis/reports/.

[9] D. K. Faddeev, *Some properties of a matrix that is the inverse of a Hessenberg matrix*, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI), 111 (1981), pp. 177–179, 238. Numerical methods and questions in the organization of calculations. 5 (Russian), English translation available [10].

[10] ———, *Properties of a matrix, inverse to a Hessenberg matrix*, Journal of Mathematical Sciences, 24 (1984), pp. 118–120. English translation of [9].

[11] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), G. A. Watson, ed., vol. 506 of Lecture Notes in Math., Berlin, 1976, Springer, pp. 73–89.

[12] Joel N. Franklin, *Matrix theory*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1968.

[13] R. Freund, *Hermitian Eigenvalue Problems: Band Lanczos Method (Section 4.6)*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 80–88. Online available at http://www.cs.utk.edu/~dongarra/etemplates/node131.html.

[14] ———, *Non-Hermitian Eigenvalue Problems: Band Lanczos Method (Section 7.10)*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 205–216. Online available at http://www.cs.utk.edu/~dongarra/etemplates/node256.html.

[15] Roland W. Freund and Manish Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., 254 (1997), pp. 119–157. Received Dec. 11, 1995.

[16] Martin H. Gutknecht, *Stationary and almost stationary iterative (k,l)-step methods for linear and nonlinear systems of equations*, Numer. Math., 56 (1989), pp. 179–213. Received May 31, 1988.

[17] ———, *The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, and the qd algorithm.* Preliminary Proceedings of the Copper Mountain Conference on Iterative Methods, April 1990. Online available at http://www.sam.math.ethz.ch/~mhg/pub/CopperMtn90.ps.gz and http://www.sam.math.ethz.ch/~mhg/pub/CopperMtn90-7.ps.gz.

[18] ———, *Variants of BiCGStab for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033. Received Sep. 9, 1991.

[19] ———, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numer., 6 (1997), pp. 271–397.

[20] ———, *IDR explained*, ETNA, 36 (2010), pp. 126–148. Received Dec. 28, 2008, electr. publ. Feb. 22, 2010.

[21] Karl Hessenberg, *Behandlung linearer Eigenwertaufgaben mit Hilfe der Hamilton-Cayleyschen Gleichung*, Numerische Verfahren, Bericht 1, Institut für Praktische Mathematik (IPM), Technische Hochschule Darmstadt, July 1940. Scanned report and biographical sketch of Karl Hessenberg's life online available at http://www.hessenberg.de/karl1.html.

[22] Alston S. Householder, *The theory of matrices in numerical analysis*, Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, 1964.

[23] Cornelius Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.

[24] ———, *Solution of systems of linear equations by minimized iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[25] Damian Loher, *New block Lanczos solvers for linear systems with multiple right-hand side*, PhD thesis, Diss. no. 16337, ETH Zurich, Zurich, Switzerland, 2006.

[26] *Matrix Market*. http://math.nist.gov/MatrixMarket.

[27] Stefan Röllin and Martin H. Gutknecht, *Variations of Zhang's Lanczos-type product method*, Appl. Numer. Math., 41 (2002), pp. 119–133. Developments and trends in iterative methods for large systems of equations—in memoriam Rüdiger Weiss (Lausanne, 2000).

[28] Axel Ruhe, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551. Received May 4, 1995.

[29] Axel Ruhe and Daniel Skoogh, *Rational Krylov algorithms for eigenvalue computation and model reduction*, in Applied parallel computing (Umeå, 1998), vol. 1541 of Lecture Notes in Comput. Sci., Springer, Berlin, 1998, pp. 491–502.

[30] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295. Received Nov. 7, 1979.

[31] Youcef Saad and Martin H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869. Received Nov. 29, 1983.

[32] J. Schur, *Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind*, Journal für die reine und angewandte Mathematik, 147 (1917), pp. 205–232.

[33] Ferd. Schweins, *Theorie der Differenzen und Differentiale*, Verlag der Universitäts-Buchhandlung von C. F. Winter, Heidelberg, 1825. Digitized by Google, online available at http://books.google.de/books?id=dntNAAAAMAAJ.

[34] Valeria Simoncini and Daniel B. Szyld, *Interpreting IDR as a Petrov-Galerkin method*, Report 09-10-22, Dipartimento di Matematica, Università di Bologna and Department of Mathematics, Temple University, Philadelphia, 2009.

[35] Gerard L. G. Sleijpen, Peter Sonneveld, and Martin B. van Gijzen, *Bi-CGSTAB as an induced dimension reduction method*, Report 08-07, Department of Applied Mathematical Analysis, Delft University of Technology, 2008. To appear as [36].

[36] Gerard L. G Sleijpen, Peter Sonneveld, and Martin B. van Gijzen, *Bi-CGSTAB as an induced dimension reduction method*, Applied Numerical Mathematics, (2009). Article in press, http://dx.doi.org/10.1016/j.apnum.2009.07.001.

[37] Gerard L. G. Sleijpen and Martin B. van Gijzen, *Exploiting BiCGstab(ℓ) strategies to induce dimension reduction*, Report 09-02, Department of Applied Mathematical Analysis, Delft University of Technology, 2009.

[38] Peter Sonneveld, *On the convergence behaviour of IDR(s)*, Report 10-08, Department of Applied Mathematical Analysis, Delft University of Technology, 2010.

[39] Peter Sonneveld and Martin B. van Gijzen, *IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, Report 07-07, Department of Applied Mathematical Analysis, Delft University of Technology, 2007. Published as [40].

[40] ———, *IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Comput., 31 (2008), pp. 1035–1062. Received Mar. 20, 2007.

[41] E. Stiefel, *Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme*, Comment. Math. Helv., 29 (1955), pp. 157–179. Received Dec. 9, 1954.

[42] Masaaki Tanio and Masaaki Sugihara, *GIDR(s,L): generalized IDR(s)*, in The 2008 annual conference of the Japan Society for Industrial and Applied Mathematics, Chiba, Japan, September 2008, pp. 411–412. (Japanese).

[43] Masaaki Tanio and Masaaki Sugihara, *GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials*, Tech. Report METR 2009-16, Department of Mathematical Informatics, Graduate School of information Science and Technology, The University of Tokio, April 2009.

[44] H. A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644. Received May 21, 1990.

[45] H. A. van der Vorst and P. Sonneveld, *CGSTAB, a more smoothly converging variant of CG-S*, Report 90-50, Department of Mathematics and Informatics, Delft University of Technology, 1990.

[46] Martin B. van Gijzen and Peter Sonneveld, *An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties*, Report 08-21, Department of Applied Mathematical Analysis, Delft University of Technology, 2008.

[47] P. Wesseling and P. Sonneveld, *Numerical experiments with a multiple grid and a preconditioned Lanczos type method*, in Approximation methods for Navier-Stokes problems (Proc. Sympos., Univ. Paderborn, Paderborn, 1979), R. Rautmann, ed., vol. 771 of Lecture Notes in Math., Berlin, Heidelberg, New York, 1980, Springer-Verlag, pp. 543–562.

[48] Man-Chung Yeung, *ML(N)BiCGSTAB: reformulation, analysis and implementation*. Report 09-01, Math. Dept., UW, version dating to 2009-05-21 2009. (a draft subject to change).

[49] Man-Chung Yeung and Daniel Boley, *Transpose-free multiple Lanczos and its application to Padé approximation*, J. Comput. Appl. Math., 177 (2005), pp. 101–127.

[50] Man-Chung Yeung and Tony F. Chan, *ML(K)BiCGSTAB: a BiCGSTAB variant based on multiple Lanczos starting vectors*, SIAM J. Sci. Comput., 21 (1999), pp. 1263–1290. Received May 16, 1997, electr. publ. Dec. 15, 1999.

[51] David M. Young and Kang C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.

[52] Jens-Peter M. Zemke, *Hessenberg eigenvalue–eigenmatrix relations*, Linear Algebra Appl., 414 (2006), pp. 589–606.

[53] ———, *Abstract perturbed Krylov methods*, Linear Algebra Appl., 424 (2007), pp. 405–434.

[54] ———, *IDR(s) and IDR(s)Eig in parallel computing*, Supercomputing News, 12 (2010), pp. 31–48. http://www.cc.u-tokyo.ac.jp/publication/news/VOL12/No2/201003ase2.pdf.

[55] Fuzhen Zhang, ed., *The Schur complement and its applications*, vol. 4 of Numerical Methods and Algorithms, Springer, New York, 2005.

[56] Shao-Liang Zhang, *GPBi-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 18 (1997), pp. 537–551.

Research Reports

No.	Authors/Title
10-13	*M.H. Gutknecht and J.-P.M. Zemke* Eigenvalue computations based on IDR
10-12	*H. Brandsmeier, K. Schmidt and Ch. Schwab* A multiscale hp-FEM for 2D photonic crystal band
10-11	*V.H. Hoang and C. Schwab* Sparse tensor Galerkin discretizations for parametric and random parabolic PDEs. I: Analytic regularity and gpc-approximation
10-10	*V. Gradinaru, G.A. Hagedorn, A. Joye* Exponentially accurate semiclassical tunneling wave functions in one dimension
10-09	*B. Pentenrieder and C. Schwab* *hp*-FEM for second moments of elliptic PDEs with stochastic data. Part 2: Exponential convergence
10-08	*B. Pentenrieder and C. Schwab* *hp*-FEM for second moments of elliptic PDEs with stochastic data. Part 1: Analytic regularity
10-07	*C. Jerez-Hanckes and J.-C. Nédélec* Asymptotics for Helmholtz and Maxwell solutions in 3-D open waveguides
10-06	*C. Schwab and O. Reichmann* Numerical analysis of additive, Lévy and Feller processes with applications to option pricing
10-05	*C. Schwab and R. Stevenson* Fast evaluation of nonlinear functionals of tensor product wavelet expansions
10-04	*B.N. Khoromskij and C. Schwab* Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs
10-03	*A. Cohen, R. DeVore and C. Schwab* Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs
10-02	*V. Gradinaru, G.A. Hagedorn, A. Joye* Tunneling dynamics and spawning with adaptive semi-classical wavepackets