SIAM J. Sci. Comput.
Vol. 34, No. 5, pp. A2576–A2598

ON THE CONVERGENCE BEHAVIOR OF IDR(s) AND RELATED METHODS[*]

PETER SONNEVELD[†]

Abstract. An explanation is given of the convergence behavior of IDR(s) methods. The convergence mechanism of these algorithms has two components. The first consists of damping properties of certain factors in the residual polynomials, which becomes less important for large values of s. The second component depends on the behavior of Lanczos polynomials that occur in the theoretical description. This part of the residual polynomials is related to Lanczos methods with s left starting vectors, as described in a paper by Yeung and Chan on their ML(k)BiCGSTAB method, in [*SIAM J. Sci. Comput.*, 21 (1999), pp. 1263–1290]. In this paper, the behavior of the second component is compared with the full GMRES method [*SIAM J. Sci. Statist. Comput.*, 7 (1986), pp. 856–869] and an expected rate of convergence is given, based on a random choice of the s shadow vectors.

Key words. iterative methods, IDR, Krylov subspace methods, Bi-CGSTAB, nonsymmetric linear systems

AMS subject classifications. 65F10, 65F50

DOI. 10.1137/100789889

1. Introduction.

IDR(s) [32, 25, 30] is a recently developed family of short-term recurrence Krylov subspace solvers for sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$, in which $\boldsymbol{A}$ is not necessarily Hermitian. The algorithms do not require multiplications with the transpose matrix and belong therefore to the "Lanczos-type product methods" [12]. The algorithms are related to the so-called block Lanczos algorithms [4, 9] with multiple left starting vectors. These vectors occur also in the IDR(s) algorithms and are called shadow vectors. The parameter s is a positive integer, indicating the number of shadow vectors.

Relations with Bi-CGSTAB are described in [19] and the combination of IDR(s) with BiCGSTAB(ℓ) in [20]. Recent extensions to IDR(s) can be found in [3, 29], and IDR(s)-eigenvalue algorithms have been developed [13].

1.1. Motivation.

It has been observed that the rate of convergence of IDR(s) usually increases at increasing s. However, the memory requirements as well as the overhead CPU time grow quadratically with s, so the choice of s is a compromise between these requirements and convergence of the method. Often $s = 4$ is a good choice, but sometimes a (much) larger value is required. We provide some insight to this problem.

The convergence behavior depends not only on the number of shadow vectors but also on their choice, but a "best set" of these vectors has not yet been found. It appears that best results are obtained if the shadow vectors have as little to do with the problem as possible. Choices of the shadow vectors that were related to properties of the matrix proved to be rather disappointing. For that reason, it is advised in [25] to choose the shadow vectors randomly. This random choice is the basis for the convergence analysis presented in this paper.

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 23, 2010; accepted for publication (in revised form) July 17, 2012; published electronically September 25, 2012.

http://www.siam.org/journals/sisc/34-5/78988.html

[†] Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD, Netherlands (p.sonneveld@ewi.tudelft.nl).

Extensive experiments with IDR(s) often show a convergence behavior which is more or less similar to the full GMRES method [16], but without the necessity of storing a growing set of Krylov vectors. In this paper we try to explain this behavior by observing two distinct convergence mechanisms in IDR(s): A "damping part" and a "Lanczos part." The residuals of the Lanczos part, which are similar to the residuals of the ML(k)BiCG method [33], are compared to the full GMRES method.

1.2. Notation and prerequisites.

We use boldface letters for vectors, boldface capitals for matrices, and calligraphic symbols like $\mathcal{S}$ and $\mathcal{G}$ for spaces, sets, and classes. The identity and zero matrices are denoted by $\boldsymbol{I}$ and $\boldsymbol{O}$, or $\boldsymbol{I}_k$ and $\boldsymbol{O}_k$ if the size is not clear from the context. The symbol $\boldsymbol{P}$ is used for the matrix of shadow vectors $(\boldsymbol{p}_1\ \boldsymbol{p}_2\ \dots\ \boldsymbol{p}_s)$ and to denote a projection. The range and the nullspace of a matrix $\boldsymbol{A}$ are denoted by $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{N}(\boldsymbol{A})$, respectively. The transpose and the Hermitian transpose of a matrix $\boldsymbol{A}$ are denoted by $\boldsymbol{A}^T$ and $\boldsymbol{A}^H$, respectively. $\boldsymbol{A}^{-H}$ means $(\boldsymbol{A}^H)^{-1}$. By $\subseteq$ and $\subset$ we mean subset and proper subset, respectively. The indexed Krylov subspaces $\mathcal{K}_j$ are defined by $\mathcal{K}_j(\boldsymbol{A},\boldsymbol{b}) = \operatorname{span}(\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \dots, \boldsymbol{A}^j\boldsymbol{b}\})$, so the dimension is $j+1$ in general. Polynomials are denoted by capital Greek symbols Φ_n, Ω_j, etc. The index refers to the degree.

This paper is partly based on probability theory. The probability of an event E is denoted by $\Pr(E)$. The conditional probability for an event E_1 *conditional with respect to an event* E_2 ("conditional on E_2") is denoted by $\Pr(E_1|E_2)$ and satisfies the following identity: $\Pr(E_1 \cap E_2) = \Pr(E_1|E_2)\cdot\Pr(E_2)$. Two events are stochastically independent if $\Pr(E_1|E_2) = \Pr(E_1)$. So for independent events we have $\Pr(E_1 \cap E_2) = \Pr(E_1)\cdot\Pr(E_2)$. A stochastic variable is a variable which attains its value by chance. So a proper dice can be represented by a stochastic variable which can take values $1\dots6$, all with equal probability $1/6$. The probability density function f (*pdf*) for a continuous stochastic variable $\boldsymbol{x}$ in $\mathbb{R}^n$ has the following meaning: $\Pr(\boldsymbol{x}\in\mathcal{D}) = \int_{\boldsymbol{t}\in\mathcal{D}} f(\boldsymbol{t})dt_1dt_2\cdots dt_n$. The *pdf* $f(\boldsymbol{x})$ is also called the joint *pdf* or simultaneous *pdf* for the stochastic variables $x_1, x_2, \dots, x_n$. Since the stochastic variable $\boldsymbol{x}$ cannot have *no value*, the probability for having *a value* will be 1. Hence the integral $\Pr(\boldsymbol{x}\in\mathbb{R}^n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(\boldsymbol{t})dt_1dt_2\cdots dt_n = 1$. The integral $g(x_1, x_2, \dots, x_k) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(x_1, x_2, \dots, x_n)dx_{k+1}dx_{k+2}\cdots dx_n$ is called the marginal *pdf* for the variables $x_1, \dots, x_k$. If $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ are two stochastic vectors in $\mathbb{R}^n$, related by the differentiable transformation $\boldsymbol{x} = \phi(\widetilde{\boldsymbol{x}})$, and if $f(\boldsymbol{x})$ is the *pdf* for $\boldsymbol{x}$, then the *pdf* for $\widetilde{\boldsymbol{x}}$ reads $g(\widetilde{\boldsymbol{x}}) = f(\phi(\widetilde{\boldsymbol{x}}))|\det(\frac{\partial\boldsymbol{x}}{\partial\widetilde{\boldsymbol{x}}})|$, where $\frac{\partial\boldsymbol{x}}{\partial\widetilde{\boldsymbol{x}}}$ is the Jacobian of the transformation.

1.3. Outline.

The paper is organized as follows. In section 2 some basic information about IDR(s) is recalled, and some elementary observations of the convergence behavior are stated. In section 3 the splitting of the IDR(s) residuals in a damping part and a Lanczos part is described. In section 4, the residuals of the Lanczos part are interpreted as Galerkin residuals and compared with the residuals of the full GMRES algorithm. In section 5 the random choice of the shadow vectors is analyzed, which results in a "stochastic convergence analysis." In section 6 the convergence analysis is verified with experiments. Finally, in section 7 the results are discussed, and some conclusions are drawn.

2. Background and basic experience.

2.1. Theoretical basis.

The IDR(s) algorithms are based on the following proposition [25]:

Proposition 2.1. *Let $\boldsymbol{A}$ be an $N \times N$ complex matrix, let $\boldsymbol{b}$ be a vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the full Krylov subspace $\mathcal{K}(\boldsymbol{A}, \boldsymbol{b})$. Let $\mathcal{S}$ be a proper subspace of $\mathcal{G}_0$ of codimension s, not containing a nontrivial invariant subspace of $\boldsymbol{A}$, and let the sequence of spaces $\mathcal{G}_j,\ j = 1, 2, \dots,$ be defined recursively by*

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

where ω_j are nonzero complex numbers. Then the spaces $\mathcal{G}_j$ are nested in the following way:

$$\mathcal{G}_j \subset \mathcal{G}_{j-1}.$$

Moreover, apart from exceptional circumstances, the dimensions of the spaces satisfy

$$\dim(\mathcal{G}_j) = \dim(\mathcal{G}_{j-1}) - s.$$

This is the so-called dimension reduction phenomenon, and it is proved in two theorems in [25].

The IDR(s) algorithms make use of this property by constructing a sequence of residual vectors $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$ that are forced to be in $\mathcal{G}_j$ for increasing values of j. The space $\mathcal{S}$ is chosen to be the nullspace of $\boldsymbol{P}^H$, where $\boldsymbol{P}$ is an $N \times s$ matrix of full column rank. Usually $\boldsymbol{P}$ is chosen randomly, and this choice is essential in the following analysis.

Basically the construction principle is as follows. Suppose we have $s+1$ independent residuals $\boldsymbol{r}_{j,1}, \boldsymbol{r}_{j,2}, \dots, \boldsymbol{r}_{j,s+1}$ in $\mathcal{G}_j$. We can find a nontrivial combination $\sum_{k=1}^{s+1} c_k \boldsymbol{r}_{j,k}$ in $\mathcal{S} \cap \mathcal{G}_j$ by solving $c_1, c_2, \dots, c_{s+1}$ from the homogeneous linear system

$$\boldsymbol{P}^H \left(\sum_{k=1}^{s+1} c_k \boldsymbol{r}_{j,k} \right) = \boldsymbol{0}. \tag{1}$$

By applying the mapping $\boldsymbol{I} - \omega_{j+1}\boldsymbol{A}$ to this combination, we obtain a vector in $\mathcal{G}_{j+1}$. By suitable scaling, e.g., requiring $\sum_{k=1}^{s+1} c_k = 1$, this vector can be made a residual $\boldsymbol{r}_{n+1}$, and an update $\boldsymbol{x}_{n+1}$ for the solution can be found. This process can be repeated using this new residual, because the $\mathcal{G}$-spaces are nested, and every vector in $\mathcal{G}_{j+1}$ is in $\mathcal{G}_j$. After $s+1$ steps, we have found $s+1$ vectors in $\mathcal{G}_{j+1}$, and therefore we can enter the space $\mathcal{G}_{j+2}$, etc. In the generic case we have, after finding l new vectors in $\mathcal{G}_{j+1}$, $s+1+l$ independent vectors in $\mathcal{G}_j$. So we may find a combination in $\mathcal{N}(\boldsymbol{P}^H)$ by solving the $s \times (s+1+l)$ system

$$\boldsymbol{P}^H \left(\sum_{k=0}^{s+l} c_k \boldsymbol{r}_{j(s+1)+k} \right) = \boldsymbol{0}. \tag{2}$$

Hence we have a lot of freedom in constructing combinations of them which are in $\mathcal{N}(\boldsymbol{P}^H)$. Also it is not necessary to choose these vectors from the set of residuals [30]. But in all variants, the first residuals $\boldsymbol{r}_{j(s+1)}$ in $\mathcal{G}_j$ are unique.

In the generic case, every $s+1$ iterations we enter a new space of which the dimension is s less than the previous one. Therefore in about $\frac{s+1}{s}N$ steps, the space $\mathcal{G}_j$ with $js \geq N$ has dimension zero, and hence the residuals in it are zero. In this sense, the IDR(s) method is finite. The process may break down by inconsistency of the small linear system (2). In exact arithmetic, and using random shadow vectors, such a breakdown has zero probability. In finite precision, however, a breakdown can occur, but at increasing s breakdowns become rare.

2.2. Finite precision issues. Most short-term recurrence Krylov solvers may suffer from heavy loss of digits. Sometimes this is caused by bad properties of the linear system itself, such as some Matrix Market examples. IDR(s) for reasonably small values of s will probably not converge in these cases. This phenomenon is almost incurable since the original problem is ill-posed.

For these problems full GMRES sometimes works well, in the sense that it produces the best possible approximation of the solution in a finite number of steps. But this finite number is often close to the theoretical bound N, the size of the system, which is much larger than what we expect from an iterative solver.

But also in the case of well-posed problems, short-term recurrence Krylov solvers may lose digits, caused by the fact that the residuals can behave very irregularly. Using finite precision arithmetic, these irregularities can degrade the accuracy dramatically.

One way of reducing degradation is to use *reliable updates* [23, 10, 22]. This is an implementation style in which updates of residuals are calculated as $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \boldsymbol{A}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)$ whenever possible. This is common practice in implementing Krylov subspace solvers.

Despite using reliable updates, degradation may occur by heavy fluctuations in order of magnitude of the residuals. Also, this kind of heavy round-off can be attacked rather well. In the basic form of the IDR(s) method, it has been observed that the occurrence of residuals that are some orders of magnitude larger than the initial residual is followed by a final (stagnation) level that is comparably higher than the theoretically possible level, i.e., the level dictated by the condition number of the matrix. Roughly speaking, occurrence of a residual $\|\boldsymbol{r}_k\| > 10^d\|\boldsymbol{r}_0\|$ leads to a loss of about d digits in the end. This is explainable on very elementary grounds.

Backgrounds of these phenomena can be found in [28, 31]. Most remedies are based on a careful application of *residual replacement*. This means that at a certain stage the recursively calculated residual is replaced by the actual one: $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$, at the cost of one extra matrix vector product. This may not be done near the end of the process, since then the convergence will deteriorate by another cause.

In the experiments, we used the following, somewhat simplified, variant of this principle. As soon as $\|\boldsymbol{r}_n\| \geq 10 \cdot \|\boldsymbol{r}_0\|$ for some n (which almost always happens in the beginning of the process), a repair flag is set. As soon as some residual $\boldsymbol{r}_{n'}$ satisfies $\|\boldsymbol{r}_{n'}\| < \frac{\|\boldsymbol{r}_0\|}{1000}$, while the repair flag is set, the current residual is replaced by the true residual $\boldsymbol{r}_{n'} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{n'}$, and the repair flag is reset.

In many experiments using this strategy, the stagnation level turned out to be comparable to the stagnation level of full GMRES, and sometimes even lower. The number of extra matrix vector products never exceeded 3, so it is done at marginal cost.

The original version of IDR(s) [25] was extra sensitive for round-off at higher values of s. Although in most cases $s = 4$ is a suitable choice, we want to analyze the behavior of the method for a wide range of values of s. This was an important argument for choosing the "elegant" IDR(s) variant [30] for our experiments, which does not suffer from this problem. Actually, this choice is also very good for other reasons, as will become clear in section 3.

In all experiments, the residual replacement strategy is used as described above, in order to simulate exact arithmetic as much as possible. Finally, in practical use of the method, the residuals as produced by the recurrence relations are also used as indicator for convergence. Now if a recursive residual is below the tolerance level, the true residual is calculated in order to verify the actual convergence, and if necessary some more steps can be done, or a message about the actual accuracy can be produced.

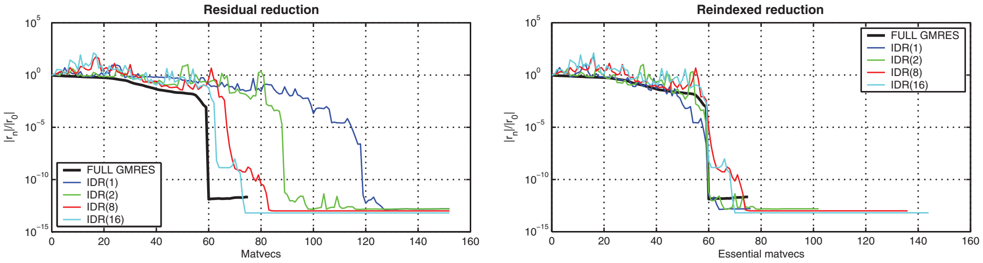

Fig. 1. *History for Problem* 1.

Since we are analyzing the analytic convergence behavior of the method, we let IDR(s) calculate the residuals recursively, except when residual replacement must occur, but *we always show and analyze the exact residuals* $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$.

2.3. First observations on the convergence behavior of IDR(s). The behavior of IDR(s) is illustrated by applying the method on the following simple test problems:

Problem 1. A discretized one-dimensional diffusion equation, leading to 60 linear equations.

Problem 2. A two-dimensional convection diffusion equation on a square 60×65 grid, with mesh-Peclet numbers $[0.5\,,\,0]$, leading to 3900 linear equations.

Problem 3. The same problem as Problem 2, but with mesh-Peclet numbers $[20\,,\,0]$.

The finiteness property of IDR(s) is illustrated in the left plot of Figure 1, showing the convergence history of IDR(s) applied to Problem 1, for several values of s.

In the right plot of Figure 1, the same graph is shown, but now with the horizontal axis stretched by a factor $\frac{s}{s+1}$. This plot confirms the $\frac{s+1}{s}N$ behavior as predicted by the theory. Plots in which this scaling has been applied could be called *rescaled* plots. In the actual implementation, however, we did not scale the horizontal axis, but we obtained the effect by simply skipping the residuals with index $j \cdot (s+1)$ for $j \geq 1$. Consequently we call these convergence plots *reindexed plots*. In these plots, the horizontal axis shows no longer the number of matrix-vector multiplications but only $\frac{s}{s+1}$ times that number. We call these *essential matrix-vector-multiplications*.

Similarly as in the CG-algorithm, the method frequently converges as an iterative method in a number of steps that is far below the theoretical bound $\frac{s+1}{s}N$. This is illustrated in the left plot of Figure 2, showing the convergence history for IDR(s) applied to Problem 2. Here only about 200 iterations are required to gain 13 decimal digits, which is much less than the theoretical bound $\frac{s+1}{s} \cdot 3900$. The maximal attainable accuracy is related to the minimal attainable residual via the the condition number of the matrix. Generally speaking, since full GMRES is a minimization method, we may expect that the final stagnation level of this method indicates the minimal attainable level and indirectly indicates the maximal attainable accuracy.

If the IDR(s) methods are compared with full GMRES, the convergence characteristics for increasing s seem to "converge" to a limiting curve close to the full GMRES curve for the same problem. Now the GMRES curve is a "lower bound" for all Krylov subspace methods, because it minimizes the residual norm $\|\boldsymbol{r}_n\|$ over the Krylov space span$\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \ldots, \boldsymbol{A}^n\boldsymbol{r}_0\}$, so the observed increasing similarity of GMRES and IDR(s) for increasing values of s is quite promising.

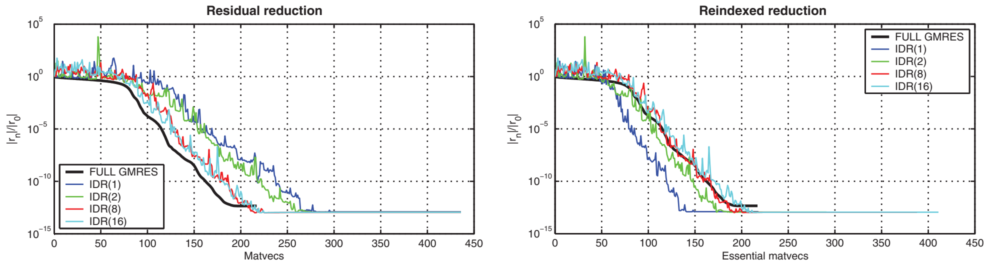

FIG. 2. *History for Problem* 2.

If we take a closer look to the left plots of Figures 1 and 2, it can be observed that the convergence curves for different s-values in both plots tend to the GMRES curve for increasing s, in a similar way. The convergence speed of IDR(s) seems to be about $\frac{s}{s+1}$ times that of GMRES.

Whether this is true can be verified by applying the same stretching (with the factor $\frac{s}{s+1}$) of the horizontal axis as in the right plot of Figure 1. The result is shown in the right plot of Figure 2.

We can make two observations about this picture:

1. The curves for $s = 1$ and $s = 2$ seem to show faster convergence than GMRES, which is not possible. This has a simple explanation: For $s = 1$, only half the actual matrix-vector multiplications are shown, and for $s = 2$, only 66%.

2. The curves for $s = 8$ and $s = 16$ are nearly covering the fast convergence part of the full GMRES curve. It may be expected that this will also be the case for higher values of s. This behavior will be analyzed and explained in the following sections.

3. The polynomial background of the IDR(s) residuals.

3.1. Damping and Lanczos part of the convergence. In each step of an IDR(s) algorithm a new residual is constructed, involving one matrix-vector multiplication. Therefore, as in other Krylov subspace methods, we can write

$$\boldsymbol{r}_n = \Phi_n(\boldsymbol{A})\boldsymbol{r}_0, \tag{3}$$

where the so-called residual polynomial Φ_n is an nth degree polynomial satisfying $\Phi_n(0) = 1$. We call this the *IDR-polynomial*.

Furthermore, as described in [25], for residuals that are in $\mathcal{G}_j$, the IDR-polynomial Φ_n can be explicitly written as a product of two polynomials:

$$\Phi_n(\boldsymbol{A}) = \Omega_j(\boldsymbol{A})\Psi_{n-j}(\boldsymbol{A}), \tag{4}$$

where

$$\Omega_0(t) \equiv 1,\ \Omega_j(t) = \prod_{k=1}^{j}(1 - \omega_k t),\ j \geq 1, \quad \Psi_{n-j}(t) = 1 - \sum_{l=1}^{n-j} c_l t^l. \tag{5}$$

The choice 1 for the zero-order coefficient follows from the required property $\Phi_n(0) = 1$. The polynomials Ψ_{n-j} are not all uniquely determined. For $n = j(s+1) - 1$ and $n = j(s+1)$, corresponding to the last residual in $\mathcal{G}_{j-1}$ and the first residual in $\mathcal{G}_j$, respectively, we get different definitions of Ψ_{n-j}, since in general

$$\frac{\Phi_{j(s+1)-1}(\boldsymbol{A})}{\Omega_{j-1}(\boldsymbol{A})} \neq \frac{\Phi_{j(s+1)}(\boldsymbol{A})}{\Omega_j(\boldsymbol{A})}.$$

We can deal with this ambiguity by dropping one of the "definitions." The obvious choice is to drop the variant for $n = j(s+1) - 1$ and to keep the variant based on the first residual in the new space. See also [13]. The ambiguity does not occur in the variant of IDR(s) described in [30], since in this variant the critical residuals satisfy

$$\boldsymbol{r}_{j(s+1)} = (\boldsymbol{I} - \omega_j \boldsymbol{A})\boldsymbol{r}_{j(s+1)-1}, \;\; j = 1, 2, 3, \ldots. \tag{6}$$

We call the factors $\Omega_j(\boldsymbol{A})$ *damping factors* or *stabilization factors* and the polynomials $\Psi_{n-j}(\boldsymbol{A})$ *Lanczos factors*. The damping factors have their name because the coefficients ω_j are usually calculated with the purpose of minimizing the norm of $\boldsymbol{r}_{j(s+1)} = (\boldsymbol{I} - \omega_j \boldsymbol{A})\boldsymbol{v}$ for some vector $\boldsymbol{v}$ that arises in the algorithm at that stage.

It is plausible to expect that the matrix polynomial $\Omega_j(\boldsymbol{A})$ will act as a contraction, but this is not always the case. In many cases, however, the damping factors in the residuals are at least partly responsible for the convergence. So we still call them "damping factors" or "stabilization factors," even if they do not damp or stabilize at all.

The name *Lanczos factors* for the polynomials $\Psi_{n-j}(\boldsymbol{A})$ is chosen because they occur explicitly in a block Lanczos process with s starting vectors at the left-hand side. In Yeung and Chan [33], a theoretical Krylov solver called ML(k)BiCG is defined with these factors as residual polynomials. In a similar way as is done in the derivation of BiCGSTAB in [27], the handling of the left-hand starting vectors is changed such that matrix-vector products with the transpose are avoided. This leads to the practical solver ML(k)BiCGSTAB, which is, at least mathematically, closely related to the IDR(s) algorithms.

The Lanczos polynomials Ψ_{n-j} in the IDR(s) algorithms are not precisely those of the ML(k)BiCG algorithm. This is because the small linear systems (2) leave a lot of freedom in the chosen solution.

We define the *Lanczos residuals* $\widetilde{\boldsymbol{r}}_{n-j}$ by

$$\widetilde{\boldsymbol{r}}_{n-j} = \Psi_{n-j}(\boldsymbol{A})\boldsymbol{r}_0. \tag{7}$$

These vectors are residuals indeed, since according to (5) $\Psi_{n-j}(0) = 1$. The IDR(s) algorithms do not produce the Lanczos residuals directly. In the IDR(s) variant [30], it is easy to calculate the Lanczos residuals $\widetilde{\boldsymbol{r}}_{n-j}$ and corresponding $\widetilde{\boldsymbol{x}}_{n-j}$ by implementing a copy of this IDR(s) algorithm, running together with the IDR(s) algorithm itself and using the same coefficients, but in which the explicit multiplication with $\boldsymbol{I} - \omega_j \boldsymbol{A}$ at step $j(s+1)$ is omitted. This "shadow process" produces the Lanczos iterates and residuals. Strictly speaking, we deviate here from the condition "keep the newest residual" in the definition of the Lanczos polynomial Ψ_{n-j}. However, in the elegant variant of IDR(s) the two variants of Ψ_{n-j} are identical according to (6).

In [13] a theoretical framework is developed, in which a similar procedure is described, based on a reduction of a Hessenberg pencil related to IDR(s). Although this procedure is applicable to general variants of IDR(s), the author's implementation on the prototype IDR(s) turned out to be rather unstable. In the case of the IDR(s) variant [30] the procedure can be reduced to a simple ad hoc modification of a working algorithm, as described.

In the shadow process for producing the Lanczos residuals, we apply residual replacement with the same strategy as in the IDR(s) algorithm itself, and we always show the true residual norms $\|\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}_j\|$.

Note that the skipping procedure in the shadow process leads to the same reduction by a factor $\frac{s}{s+1}$ as is applied in the rescaling of convergence plots, which actually was carried out by dropping one matrix-vector multiplication every $s + 1$ steps.

The procedure described above has also been suggested in [2] as a tool for retrieving f.i. *BiCG* residuals from the *BiCGSTAB* algorithm. It requires s extra (hidden) matrix vector products every $s+1$ steps, so probably this is not a practical solver. However, the (theoretical) solver ML(k)BiCG [33] requires k direct matrix vector products and k products with the transpose for every k steps, so the variant used in our experiments is easier, and not much more expensive then ML(k)BiCG would be.

3.2. Experiments on damping and Lanczos parts. We can view the contributions of the Lanczos part, and the damping part by comparing the plots of the Lanczos residuals with the reindexed plots of IDR(s) residuals.

In Figure 3 the reindexed IDR(s) residuals are compared with the Lanczos residuals for Problem 2 with $s = 1, 2, 8, 16$. It is quite clear that IDR(1) has rather large profit from the damping part of the convergence. For larger values of s this effect becomes smaller. So this is a case in which the damping factors actually produce damping.

In Figure 4, similar results are shown for Problem 3, i.e., the same convection diffusion equation as in Problem 2, but with a rather large mesh-Peclet number 20 in the x-direction. For classical iterative procedures this is disastrous, and this effect is visible in the left plot for $s = 1$ and $s = 2$. For the larger values of s, the method converges.

The Lanczos residuals, depicted in the left plot of Figure 4, converge also for $s = 1$ and $s = 2$, albeit rather slowly. So apparently, the damping factor of the IDR(s) residual polynomial does not damp at all in Problem 3, and the convergence is completely due to the Lanczos part.

The cause of the failing damping property is that the calculated real values for ω_j would be very small, causing *stagnation* of the procedure (as in Bi-CGSTAB). Therefore a modified calculation of ω_j is done, described in [21], which in these circumstances cause *growth* instead of a damping.

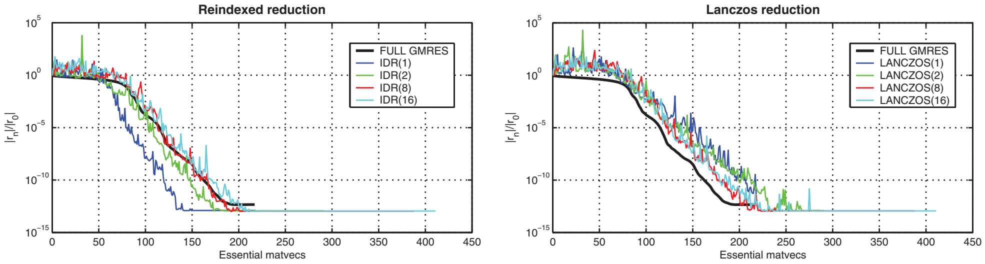

FIG. 3. *Reindexed versus Lanczos residuals in Problem* 2.

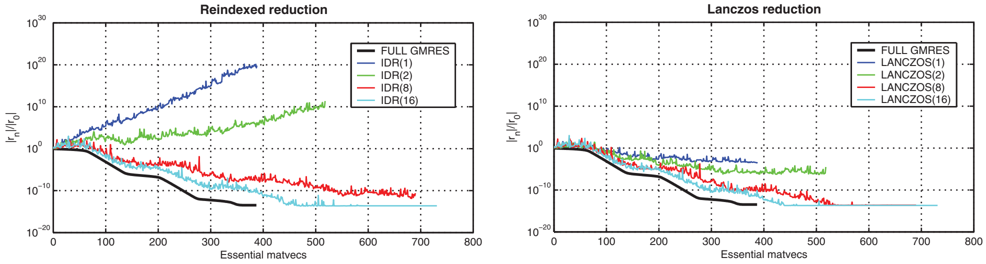

FIG. 4. *No damping in Problem* 3.

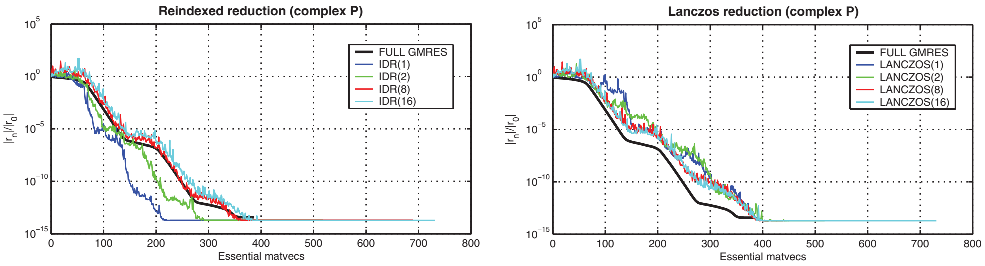

FIG. 5. *Remedy by complex* $\boldsymbol{P}$.

So it is quite reasonable to look seriously for alternative criteria for choosing the parameters ω_j. For the case $s = 1$, which is equivalent to Bi-CGSTAB, Gutknecht [11] and Sleijpen and Fokkema [18] have developed variants of the algorithm, in which the factors $\boldsymbol{I} - \omega_j \boldsymbol{A}$ are combined into higher degree polynomials, allowing for better minimization possibilities in this kind of cases. Recently Tanio and Sugihara [26] and Sleijpen and van Gijzen [20] have developed similar techniques for IDR(s) with $s > 1$.

Simoncini and Szyld [17] developed alternative criteria for the ω-values, based on the field of values of $\boldsymbol{A}$.

One extremely easy way to handle the nondamping problem in IDR(s) is shown in [25]. By choosing the matrix $\boldsymbol{P}$ complex, the algorithm is forced to use complex arithmetic and therefore may find better factors. The result is shown in the left and right plots of Figure 5. Apart from the increase in computational work caused by complex arithmetic, this seems to be a perfect remedy.

The right plots in Figures 4 and 5 show an interesting property of the Lanczos residuals. The convergence plots follow quite closely the typical plateau behavior of GMRES in this kind of problem. Apparently the behavior of the Lanczos residuals is somehow related to the behavior of the GMRES residuals. This will be analyzed in the next sections.

4. Analysis of the Lanczos factors. It is shown in [25] that the polynomials Ψ_{n-j} satisfy relations of the following type:

$$\boldsymbol{p}_r^H \Omega_l(\boldsymbol{A}) \Psi_{n-j}(\boldsymbol{A}) \boldsymbol{r}_0 = 0,\ \ l = 0, 1, \ldots, j-1,\ \ r = 1, 2, \ldots, s, \tag{8}$$

where s is the "order" of the IDR(s) algorithm. Since the polynomials Ω_0, $\Omega_1, \ldots, \Omega_{j-1}$ are a basis for the space of polynomials of degree up to $j-1$, these relations are equivalent with

$$\boldsymbol{p}_r^H \boldsymbol{A}^l \Psi_{n-j}(\boldsymbol{A}) \boldsymbol{r}_0 = 0,\ \ l = 0, 1, \ldots, j-1,\ \ r = 1, 2, \ldots, s. \tag{9}$$

Since Ψ_{n-j} has $n - j$ coefficients to be determined, the above relations can be consistent as long as $n - j \geq s \cdot j$. If the equality sign holds, the coefficients are determined uniquely, which is the case if $n = (s+1) \cdot j$, corresponding exactly with the calculation of the very first residual in the space $\mathcal{G}_j$. In the calculations for the residuals $\boldsymbol{r}_n$ for intermediate values of n, there is freedom, which can be used for stability purposes.

One important conclusion can be drawn from the relations in (9): the Lanczos factor Ψ_{n-j} *is independent from the damping polynomial* Ω_j. So the Lanczos part of the convergence is not influenced by the damping strategy, at least as long as finite precision does not play a role.

Define the Krylov vectors $\boldsymbol{k}_l$ and the reduced Krylov matrices $\boldsymbol{K}_l$ by

$$\boldsymbol{k}_l = \boldsymbol{A}^l \boldsymbol{r}_0,\ l = 0, 1, 2, \ldots, \qquad \boldsymbol{K}_l = (\boldsymbol{k}_1\ \boldsymbol{k}_2\ \ldots\ \boldsymbol{k}_l),\ l = 1, 2, \ldots, \tag{10}$$

and let $\Psi_{n-j}(\boldsymbol{A})$ be written as

$$\Psi_{n-j}(\boldsymbol{A}) = \sum_{l=0}^{n-j} c_l \boldsymbol{A}^l \ = \ \boldsymbol{I} - \sum_{l=1}^{n-j} c_l \boldsymbol{A}^l;$$

then the equations for the coefficients c_l can be written as

$$\boldsymbol{p}_r^H \boldsymbol{A}^l [\boldsymbol{K}_{n-j}\boldsymbol{c} - \boldsymbol{r}_0] = 0,\ r = 1, 2, \ldots, s,\ l = 0, 1, \ldots, j-1.$$

These relations can be written in the following form:

$$\boldsymbol{T}^H \boldsymbol{K}_{n-j}\boldsymbol{c} = \boldsymbol{T}^H \boldsymbol{r}_0, \tag{11}$$

where

$$\boldsymbol{T} = (\boldsymbol{t}_1\, \boldsymbol{t}_2\ \ldots\ \boldsymbol{t}_{sj}), \quad \boldsymbol{t}_{ls+r} = (\boldsymbol{A}^H)^l \boldsymbol{p}_r,\ l = 0, 1, \ldots, j-1,\ r = 1, 2, \ldots, s. \tag{12}$$

The vectors $\boldsymbol{t}_i$ can be considered as *test vectors* and the matrix $\boldsymbol{T}$ as a *test matrix* in a *Galerkin context*. In fact, the Lanczos residuals can be regarded as *Galerkin residuals*, produced by a Galerkin approximation

$$\widetilde{\boldsymbol{r}}_{n-j} = \boldsymbol{r}_0 - \boldsymbol{K}_{n-j}\boldsymbol{c} \tag{13}$$

of the overdetermined system

$$\boldsymbol{K}_{n-j}\boldsymbol{c} = \boldsymbol{r}_0. \tag{14}$$

The result of a Galerkin procedure depends on the *model space* $\mathcal{M} = \mathcal{R}(\boldsymbol{K}_{n-j})$ and the *test space* $\mathcal{T} = \mathcal{R}(\boldsymbol{T})$ rather than on the matrices $\boldsymbol{K}_{n-j}$ and $\boldsymbol{T}$.

The Galerkin connection described here is valid for all values of n in the IDR(s)-variant [30]. For other variants of IDR(s) the validity is restricted to the iteration steps with $n = (s+1)\cdot j$, but still the analysis gives an explanation of the convergence behavior. We call the procedure defined by (10)–(14) the *Krylov–Galerkin approximation.*

The experiments as depicted in Figures 3–5 support the suggestion of a relation between the Krylov–Galerkin approximations and the full GMRES algorithm. This last method is a special Krylov–Galerkin approach, in which the test space coincides with the model space, so in fact GMRES produces least-squares approximations.

Comparisons of GMRES with other Krylov subspace methods are done, e.g., in [1, 5, 7, 15]. Brown [1] compares Arnoldi's method with GMRES. Cullum and Greenbaum [5] compare BiCG and QMR, both being Galerkin methods, with GMRES.

Hochbruck and Lubich [15] give a short and clear basis for an error analysis of BiCG, QMR, FOM, and GMRES, as well as for comparing these methods. One of the formulas in the proof of Theorem 1 in their paper implies formula (15) in our analysis.

Finally, Eierman and Ernst [7] make a geometrical analysis of the relation between *minimal residual methods*, like GMRES, and *orthogonal residual methods*, like BiCG. It turns out that the quality of the Galerkin method depends on the angle between the model spaces $\mathcal{K}_j(\boldsymbol{A}, \boldsymbol{r}_0)$ and the test spaces $\mathcal{K}_j(\boldsymbol{A}^H, \widehat{\boldsymbol{r}}_0)$. Here, small angles mean "good quality." The hard part is to find bounds for these angles, especially if $\boldsymbol{A}$ is far from Hermitian positive definite.

We give a simple derivation of the essential formulas, applicable to any Galerkin procedure for finite linear problems. Let $\boldsymbol{Q}$ be a matrix with orthonormal columns that span the model space and let the test space be spanned by the columns of a matrix $\boldsymbol{T}$,

$$\mathcal{M} = \mathcal{R}(\boldsymbol{Q}), \;\; \mathcal{T} = \mathcal{R}(\boldsymbol{T}).$$

In the following analysis we assume $\boldsymbol{T}^H\boldsymbol{Q}$ to be nonsingular. The corresponding *Galerkin solution* for the system $\boldsymbol{Q}\boldsymbol{c} = \boldsymbol{b}$ satisfies

$$\boldsymbol{T}^H(\boldsymbol{b} - \boldsymbol{Q}\boldsymbol{c}) = \boldsymbol{0} \Longrightarrow \boldsymbol{c} = (\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{b}.$$

The residual vector $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{Q}\boldsymbol{c}$ satisfies

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{Q}(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{b} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b},$$

where $\boldsymbol{P} = \boldsymbol{Q}(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H$ represents an oblique projection on the model space.

For the least-squares approximation we have the same relation, but with the orthogonal projector $\widehat{\boldsymbol{P}} = \boldsymbol{Q}(\boldsymbol{Q}^H\boldsymbol{Q})^{-1}\boldsymbol{Q}^H = \boldsymbol{Q}\boldsymbol{Q}^H$ instead:

$$\widehat{\boldsymbol{r}} = \boldsymbol{b} - \boldsymbol{Q}\boldsymbol{Q}^H\boldsymbol{b} = (\boldsymbol{I} - \widehat{\boldsymbol{P}})\boldsymbol{b}.$$

For comparing the Galerkin residual $\boldsymbol{r}$ with the least-squares residual $\widehat{\boldsymbol{r}}$, we make use of a remarkable yet simple property of projectors on the same space.

LEMMA 4.1. *Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be projections, and let $\mathcal{R}(\boldsymbol{P}_1) = \mathcal{R}(\boldsymbol{P}_2)$. Then*

(i) $\boldsymbol{P}_1\boldsymbol{P}_2 = \boldsymbol{P}_2$,

(ii) $(\boldsymbol{I} - \boldsymbol{P}_1)(\boldsymbol{I} - \boldsymbol{P}_2) = \boldsymbol{I} - \boldsymbol{P}_1$.

Proof. Let $\boldsymbol{x}$ be arbitrary, and let $\boldsymbol{y} = \boldsymbol{P}_2\boldsymbol{x}$; then $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{P}_2)$, and since $\mathcal{R}(\boldsymbol{P}_1) = \mathcal{R}(\boldsymbol{P}_2)$, $\boldsymbol{y} = \boldsymbol{P}_1\widetilde{\boldsymbol{x}}$, for some $\widetilde{\boldsymbol{x}}$. Hence $(\boldsymbol{P}_1\boldsymbol{P}_2)\boldsymbol{x} = \boldsymbol{P}_1\boldsymbol{P}_1\widetilde{\boldsymbol{x}} = \boldsymbol{P}_1\widetilde{\boldsymbol{x}} = \boldsymbol{y} = \boldsymbol{P}_2\boldsymbol{x}$. This being true for every $\boldsymbol{x}$, property (i) follows. Property (ii) follows directly:

$$(\boldsymbol{I} - \boldsymbol{P}_1)(\boldsymbol{I} - \boldsymbol{P}_2) = \boldsymbol{I} - \boldsymbol{P}_1 - \boldsymbol{P}_2 + \boldsymbol{P}_1\boldsymbol{P}_2 = \boldsymbol{I} - \boldsymbol{P}_1. \qquad \Box \tag{15}$$

Application of this lemma with $\boldsymbol{P}$ and $\widehat{\boldsymbol{P}}$ playing the role of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively, yields

$$\boldsymbol{r} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b} = (\boldsymbol{I} - \boldsymbol{P})(\boldsymbol{I} - \widehat{\boldsymbol{P}})\boldsymbol{b} = (\boldsymbol{I} - \boldsymbol{P})\widehat{\boldsymbol{r}}, \tag{16}$$

so we have a direct relation between $\boldsymbol{r}$ and $\widehat{\boldsymbol{r}}$, not containing $\boldsymbol{b}$ anymore. Formula (16) remains valid if $\widehat{\boldsymbol{r}}$ is from any other Galerkin method, but in our case this is not relevant.

The least-squares residual is in the orthogonal complement of $\mathcal{R}(\boldsymbol{Q})$. Let $\boldsymbol{Q}'$ be an $N \times (N-k)$ matrix with orthonormal columns that span this orthogonal complement. Then $\widehat{\boldsymbol{r}} = \boldsymbol{Q}'\boldsymbol{s}$ for some $\boldsymbol{s} \in \mathbb{C}^{N-k}$. In this way the Galerkin residual can be split into two mutually orthogonal components:

$$\widehat{\boldsymbol{r}} = \boldsymbol{Q}'\boldsymbol{s} \;\in\; \mathcal{R}(\boldsymbol{Q}') \qquad \text{(least-squares residual)}, \tag{17a}$$

$$d\boldsymbol{r} = \boldsymbol{r} - \widehat{\boldsymbol{r}} = \boldsymbol{P}\widehat{\boldsymbol{r}} = \boldsymbol{Q}(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{Q}'\boldsymbol{s} \;\in \mathcal{R}(\boldsymbol{Q}) \qquad \text{(residual surplus)}. \tag{17b}$$

We are interested in the component $d\boldsymbol{r}$. We have $\|\widehat{\boldsymbol{r}}\| = \|\boldsymbol{Q}'\boldsymbol{s}\| = \|\boldsymbol{s}\|$, and

$$\|d\boldsymbol{r}\| = \|(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{Q}'\boldsymbol{s}\|. \tag{18}$$

Let $\boldsymbol{B} = \boldsymbol{T}^H\boldsymbol{Q}$ and $\boldsymbol{B}' = \boldsymbol{T}^H\boldsymbol{Q}'$; then

$$\frac{\|d\boldsymbol{r}\|}{\|\widehat{\boldsymbol{r}}\|} \le \|\boldsymbol{B}^{-1}\boldsymbol{B}'\| \le \|\boldsymbol{B}^{-1}\| \cdot \|\boldsymbol{B}'\|. \tag{19}$$

The equality (18) is valid, independent of the choice of basis for the test space. This is not the case, however, for the estimate (19). If we change the basis for $\mathcal{T}$ by $\boldsymbol{T} = \widetilde{\boldsymbol{T}}\boldsymbol{C}$, for any invertible matrix $\boldsymbol{C}$, then $\boldsymbol{B} = \boldsymbol{C}^H\widetilde{\boldsymbol{B}}$, and $\boldsymbol{B}' = \boldsymbol{C}^H\widetilde{\boldsymbol{B}}'$. Then $\boldsymbol{B}'^{-1}\boldsymbol{B}$ is not affected, but $\|\boldsymbol{B}^{-1}\| \cdot \|\boldsymbol{B}'\|$ is.

We now assume the new basis for $\mathcal{T}$ to be orthonormal: $\mathcal{T} = \mathcal{R}(\widetilde{\boldsymbol{Q}})$, where $\widetilde{\boldsymbol{Q}}$ is an $N \times k$ matrix with orthonormal columns. With this choice we have

$$\widetilde{\boldsymbol{B}} = \widetilde{\boldsymbol{Q}}^H\boldsymbol{Q} \in \mathbb{C}^{k\times k}, \quad \widetilde{\boldsymbol{B}}' = \widetilde{\boldsymbol{Q}}^H\boldsymbol{Q}' \in \mathbb{C}^{k\times(N-k)}.$$

Let $\sigma_1 \le \sigma_2 \le \cdots \sigma_k$ denote the singular values of $\widetilde{\boldsymbol{B}}$, and let similarly $\sigma_1' \le \sigma_2' \le \cdots \sigma_k'$ denote the dominant singular values of $\widetilde{\boldsymbol{B}}'$. Then we have

$$\|\widetilde{\boldsymbol{B}}^{-1}\| = \frac{1}{\sigma_1}, \;\; \|\widetilde{\boldsymbol{B}}'\| = \sigma_k'.$$

Now the singular values of $\widetilde{\boldsymbol{B}}$ and $\widetilde{\boldsymbol{B}}'$ satisfy the following remarkable relations:

$$\sigma_{k+1-j}' = \sqrt{1-\sigma_j^2}, \;\; j = 1, 2, \ldots, k. \tag{20}$$

To see this, consider the $k \times N$ composite matrix $\widehat{\boldsymbol{B}} = [\widetilde{\boldsymbol{B}} \,|\, \widetilde{\boldsymbol{B}}']$. We have

$$\widehat{\boldsymbol{B}}\widehat{\boldsymbol{B}}^H = \widetilde{\boldsymbol{Q}}^H \cdot [\boldsymbol{Q} \,|\, \boldsymbol{Q}'] \cdot [\boldsymbol{Q} \,|\, \boldsymbol{Q}']^H\widetilde{\boldsymbol{Q}} = \boldsymbol{I}_k$$

since $[\boldsymbol{Q} \,|\, \boldsymbol{Q}']$ is unitary. Hence

$$\widehat{\boldsymbol{B}}\widehat{\boldsymbol{B}}^H = \widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}^H + \widetilde{\boldsymbol{B}}'\widetilde{\boldsymbol{B}}'^H = \boldsymbol{I}_k.$$

The singular values of $\widetilde{\boldsymbol{B}}$ and $\widetilde{\boldsymbol{B}}'$ are the positive square roots of the eigenvalues of $\widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}^H$ and $\widetilde{\boldsymbol{B}}'\widetilde{\boldsymbol{B}}'^H$, respectively, from which (20) follows. So we can write

$$\|\widetilde{\boldsymbol{B}}^{-1}\| = \frac{1}{\sigma_1}, \;\; \|\widetilde{\boldsymbol{B}}'\| = \sqrt{1-\sigma_1^2},$$

and hence

$$\frac{\|d\boldsymbol{r}\|}{\|\widehat{\boldsymbol{r}}\|} \le \frac{\sqrt{1-\sigma_1^2}}{\sigma_1}.$$

This last formula suggests the use of trigonometric functions. We can define a positive angle ϑ such that $\sigma_1 = \cos(\vartheta)$. Then

$$\frac{\|d\boldsymbol{r}\|}{\|\widehat{\boldsymbol{r}}\|} \le \tan(\vartheta). \tag{21}$$

The angle ϑ in (21) is precisely the angle between the model space and the test space, according to the definition

$$\angle(\mathcal{U},\mathcal{V}) = \max_{\boldsymbol{x}\in\mathcal{U}} \left\{ \min_{\boldsymbol{y}\in\mathcal{V}} \angle(\boldsymbol{x},\boldsymbol{y}) \right\}. \tag{22}$$

This result, although derived in a different way, can also be found in [7].

If ϑ is small, the estimate (21) guarantees that $\|\boldsymbol{r}-\widehat{\boldsymbol{r}}\|$ is small compared to $\|\widehat{\boldsymbol{r}}\|$. On the other hand, if a Galerkin-based method performs poorly compared with least squares, then ϑ is close to $\pi/2$. So if a Krylov–Galerkin method converges poorly compared to GMRES, this is caused by bad relative orientations of the Krylov subspaces $\mathcal{K}_j(\boldsymbol{A},\boldsymbol{b})$ and $\mathcal{K}_j(\boldsymbol{A}^H,\widetilde{\boldsymbol{b}})$.

We shall apply the framework differently, because the test matrix in the Krylov–Galerkin procedure has a different background. Instead of being built up from images $(\boldsymbol{A}^H)^n\widetilde{\boldsymbol{r}}_0$ of one shadow residual, the sequence of test vectors is defined by $\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_s$, $\boldsymbol{A}^H\boldsymbol{p}_1,\boldsymbol{A}^H\boldsymbol{p}_2,\ldots,\boldsymbol{A}^H\boldsymbol{p}_s,(\boldsymbol{A}^H)^2\boldsymbol{p}_1,\ldots$. So the amount of $\boldsymbol{A}^H$-influence in the test matrices is far less than in BiCG and in other classical two-sided Lanczos methods.

In the next section, we exploit the fact that the shadow vectors $\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_s$ are chosen at random.

5. Random test vectors. In the prototype IDR(s) algorithm as presented in [25], the s shadow vectors are chosen randomly. During the development of the method, choices related to spectral properties of the equation matrix have been tried out, but these choices often produced poor results. In fact, the random choice for the shadow vectors arose because nothing better could be found.

In the prototype code, every component of every vector was chosen independently uniformly distributed in the interval $[0,1]$. This choice, however, does not provide true random vectors in $\boldsymbol{C}^N$, because the density of vectors in directions like $[1,1,\ldots,1]^T$ is considerably higher that in directions like $[1,0,0\ldots,0]^T$.

Since in the IDR(s) algorithms the length of the vectors $\boldsymbol{p}_k$ is not of importance, we want the vectors uniformly distributed over the *directions* in space. So if we normalize them, we want the results to be uniformly distributed over the unit sphere in $\mathbb{C}^N$ or $\mathbb{R}^N$. This is accomplished by choosing vectors of which all components are stochastically independent variables, with a standard normal distribution, i.e., with zero mean and unit variance. The simultaneous probability density then reads

$$f(\boldsymbol{P}) = C\cdot\exp\left(-\frac{1}{2}\left(\sum_{k=1}^{N}\sum_{l=1}^{s}|p_{k,l}|^2\right)\right) \tag{23}$$

with $C=(\sqrt{2\pi})^{-Ns}$, providing $\iint\cdots\int f(\boldsymbol{P})dp_{11}dp_{12}\cdots dp_{Ns}=1$. In the version [30] of the IDR(s) algorithm, this choice for $\boldsymbol{P}$ is made.

For simplicity and readability, we define the class $\mathcal{N}$ as the set of stochastic variables normally distributed with zero mean and unit variance:

$$x\in\mathcal{N}\mapsto x \overset{\text{iid}}{\sim} N(0,1).$$

Then by $\mathcal{N}^k$ and $\mathcal{N}^{k\times l}$ we mean classes of vectors, respectively, matrices, of which all entries are in $\mathcal{N}$ and are mutually stochastically independent. Formula (23) is the probability density of a variable $\boldsymbol{P}$ in $\mathcal{N}^{N\times s}$.

In the practical IDR(s) algorithms, the Galerkin test matrices are built from the columns

$$\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s, \boldsymbol{A}^H\boldsymbol{p}_1, \boldsymbol{A}^H\boldsymbol{p}_2, \ldots, (\boldsymbol{A}^H)^j\boldsymbol{p}_s,$$

spanning the left Krylov block subspace $\mathcal{K}_j(\boldsymbol{A}^H, \boldsymbol{P})$. These columns are not stochastically independent for $k > s$ and therefore are not easily analyzed. In the numerical tests of IDR(s), the convergence curves were shifted to the left at increasing s, with a virtual limiting position close to the full-GMRES curve. Similarly, the Lanczos convergence curves were located more or less on a fixed distance of the full-GMRES curve, for s large enough, say, $s > 4$ in some cases.

As a first attempt for analyzing the role of random shadow vectors we choose a case where s is greater than the number of iterations. In this case we might expect to get a convergence curve that is to the left of all practical IDR(s) curves. We refer to this—highly impractical—method as the "full random Galerkin–Krylov" algorithm. Let $\boldsymbol{T}$ be the test matrix corresponding to the kth iteration step; then $\boldsymbol{T} \in \mathcal{N}^{N\times k}$.

Then the residual surplus $d\boldsymbol{r} = \boldsymbol{r} - \widehat{\boldsymbol{r}}$ is a stochastic vector, and its norm is a stochastic variable $v = \|d\boldsymbol{r}\|$. We figure out properties for v. According to (18), v satisfies

$$v = \|(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{Q}'\boldsymbol{s}\|,$$

where $\boldsymbol{s}$ satisfies $\|\boldsymbol{s}\| = \|\widehat{\boldsymbol{r}}\|$. The matrix $\widehat{\boldsymbol{Q}} = [\boldsymbol{Q}\,|\,\boldsymbol{Q}']$ is a unitary $N \times N$ matrix. Let $\boldsymbol{x} \in \mathcal{N}^N$, and let $\widetilde{\boldsymbol{x}} = \widehat{\boldsymbol{Q}}^H\boldsymbol{x}$; then

$$\begin{aligned} &C \cdot \exp\left(-\frac{1}{2}\|\boldsymbol{x}\|^2\right) dx_1 dx_2 \cdots dx_N \\ &\quad = C \cdot \exp\left(-\frac{1}{2}\|\widehat{\boldsymbol{Q}}\widetilde{\boldsymbol{x}}\|^2\right) \det(\widehat{\boldsymbol{Q}}) d\widetilde{x}_1 d\widetilde{x}_2 \cdots d\widetilde{x}_N \\ &\quad = C \cdot \exp\left(-\frac{1}{2}\|\widetilde{\boldsymbol{x}}\|^2\right) d\widetilde{x}_1 d\widetilde{x}_2 \cdots d\widetilde{x}_N. \end{aligned} \tag{24}$$

Hence $\widetilde{\boldsymbol{x}} \in \mathcal{N}^N$ too. Therefore all columns of $\widehat{\boldsymbol{Q}}^H\boldsymbol{T}$ are in $\mathcal{N}^N$, and since the columns of $\boldsymbol{T}$ are stochastically independent, the images also are, hence

$$\widehat{\boldsymbol{T}} = \widehat{\boldsymbol{Q}}^H\boldsymbol{T} \in \mathcal{N}^{N\times k}.$$

The matrix $\widehat{\boldsymbol{T}}^H = \boldsymbol{T}^H\widehat{\boldsymbol{Q}}$ can be split,

$$\widehat{\boldsymbol{T}}^H = \boldsymbol{T}^H\widehat{\boldsymbol{Q}} = [\boldsymbol{T}^H\boldsymbol{Q}\,|\,\boldsymbol{T}^H\boldsymbol{Q}'] = [\widetilde{\boldsymbol{T}}^H\,|\,\widetilde{\boldsymbol{T}}'^H],$$

from which follows that $\widetilde{\boldsymbol{T}}^H \in \mathcal{N}^{k\times k}$, $\widetilde{\boldsymbol{T}}'^H \in \mathcal{N}^{k\times(N-k)}$.

The stochastic variable v satisfies

$$v = \|(\boldsymbol{T}^H\boldsymbol{Q})^{-1}\boldsymbol{T}^H\boldsymbol{Q}'\boldsymbol{s}\| = \|(\widetilde{\boldsymbol{T}}^H)^{-1}\widetilde{\boldsymbol{T}}'^H\boldsymbol{s}\|.$$

The vector $\boldsymbol{s}$ is completely determined by the least-squares procedure and is therefore not stochastic. We determine the probability distribution of $\boldsymbol{y} = \widetilde{\boldsymbol{T}}'^H\boldsymbol{s}$. Let $\boldsymbol{U}$ be an $(N-k)\times(N-k)$ unitary matrix such that $\boldsymbol{s} = \|\boldsymbol{s}\|\boldsymbol{U}\boldsymbol{e}_1$; then $\widetilde{\boldsymbol{T}}'^H\boldsymbol{s} = \|\boldsymbol{s}\|(\widetilde{\boldsymbol{T}}'^H\boldsymbol{U})\boldsymbol{e}_1$. Since $\boldsymbol{U}$ is unitary, $\boldsymbol{T}'^H\boldsymbol{U} \in \mathcal{N}^{k\times(N-k)}$. The vector $\boldsymbol{z} = (\widetilde{\boldsymbol{T}}'^H\boldsymbol{U})\boldsymbol{e}_1$ is its first column, and therefore $\boldsymbol{z} \in \mathcal{N}^k$. So finally we find

$$\boldsymbol{y} = \|\boldsymbol{s}\|\boldsymbol{z} = \|\widehat{\boldsymbol{r}}\|\boldsymbol{z} \text{ with } \boldsymbol{z} \in \mathcal{N}^k.$$

It follows that $d\boldsymbol{r} = \boldsymbol{r} - \widehat{\boldsymbol{r}}$ can be written as

$$d\boldsymbol{r} = \widetilde{\boldsymbol{T}}^{-H}\boldsymbol{y} = \|\widehat{\boldsymbol{r}}\| \cdot \widetilde{\boldsymbol{T}}^{-H}\boldsymbol{z} \Longrightarrow v = \|\widehat{\boldsymbol{r}}\| \cdot \|\widetilde{\boldsymbol{T}}^{-H}\boldsymbol{z}\| \tag{25}$$

with $\widetilde{\boldsymbol{T}} \in \mathcal{N}^{k\times k}$ and $\boldsymbol{z} \in \mathcal{N}^k$.

Usually in numerical mathematics, one is interested in the worst case that can happen. In many practical situations, however, the worst case produces a far too pessimistic prediction. In the case of (25), the worst case bound is infinity, since $\|\boldsymbol{T}_1^{-1}\boldsymbol{z}\|$ is a stochastic variable with unbounded range. Therefore we must apply statistical concepts like expectations and standard deviations.

The norm of the inverse of a matrix is the reciprocal of the smallest singular value of the matrix. Edelman [6] has derived an asymptotic probability density for the smallest singular value of a matrix in $\mathcal{N}^{k\times k}$. Denote this singular value by σ_1; then the probability density for $\sigma_1\sqrt{k}$ reads

$$f(\sigma) = (1+\sigma)e^{-\sigma^2/2-\sigma}. \tag{26}$$

This means that with probability close to 1, $\sigma_1\sqrt{k} \geq C$ for moderate C, and consequently $\|\widetilde{\boldsymbol{T}}^{-H}\| \leq \widetilde{C}\sqrt{k}$ for moderate $\widetilde{C}$.

For the random vector $\boldsymbol{z}$ in (25), we may expect, by elementary statistics, that $\|\boldsymbol{z}\| \leq D\sqrt{k}$ with probability close to 1, again for moderate D. Therefore, since $\boldsymbol{T}_1$ and $\boldsymbol{z}$ are stochastically independent, we may expect

$$\|d\boldsymbol{r}\| \leq \widehat{C} \cdot k \cdot \|\widehat{\boldsymbol{r}}\| \tag{27}$$

with probability close to 1 and $\widehat{C} = \widetilde{C}D$ a moderate constant.

In order to verify whether this kind of bound is realistic, we do an investigation on linear systems $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathcal{N}^{k\times k}$ and $\boldsymbol{b} \in \mathcal{N}^k$. Such systems could be named *random systems*, but this term is sometimes used for systems in which only the matrix is random. Therefore we use the term *completely random systems*.

5.1. Completely random linear systems. In order to verify the quality of (27), we study the probabilistic properties of linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ with $\boldsymbol{A} \in \mathcal{N}^{k\times k}$ and $\boldsymbol{b} \in \mathcal{N}^k$.

Definition 5.1. 1. *A $k \times k$ linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is called a completely random linear system if all entries of $\boldsymbol{A}$ and $\boldsymbol{b}$ are stochastically independent and normally distributed stochastic variables.*

2. *The class of solutions $\boldsymbol{x}$ of a completely random $k \times k$ system is denoted by $\mathcal{Q}^k$ if the matrix and the right-hand side are real and by $\mathcal{Q}^k_{\mathbb{C}}$ if they are complex.*

We start with constructing an *experimental probability density function* (*e-pdf*) of $\|\boldsymbol{x}\|$, for $\boldsymbol{x} \in \mathcal{Q}^k$, by computing 500 samples of this stochastic variable for sizes $k = 25, 50, 100, 200$.

We actually plotted the histograms of $u = \log_{10}(\|\boldsymbol{x}\|)$ instead of $\|\boldsymbol{x}\|$, since they represent *e-pdf*s for the number of decimal digits that random Galerkin will "be behind" the least-squares method. The results are shown in Figure 6, and related statistical information is given in Table 1. For each k the histogram is plotted in a different color. The patterns have similar bell shapes, shifting slightly to the right at increasing k. According to (27), the shift should be about $\log_{10}(2)$ between two subsequent k-values. From the calculated means in Table 1, the shift is closer to $\frac{1}{2}\log_{10}(2)$. Therefore we also plot the histograms shifted to the left at an amount of $\frac{1}{2}\log_{10}(k)$ for the case corresponding to size k. The result is shown in Figure 7, and the $\frac{1}{2}\log_{10}(k)$ shift seems to be confirmed.

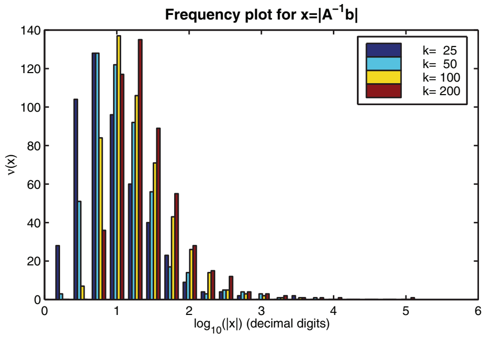

FIG. 6. e-pdf *of* $\log_{10}(\|\boldsymbol{x}\|),\ \boldsymbol{x} \in \mathcal{Q}^k$.

TABLE 1
Parameters for $\log_{10}(\|\boldsymbol{x}\|),\ \boldsymbol{x} \in \mathcal{Q}^k$.

k	Mean	Shifted	Var	stdd
25	0.953	0.254	0.239	0.488
50	1.099	0.249	0.226	0.476
100	1.270	0.270	0.211	0.459
200	1.413	0.263	0.264	0.514

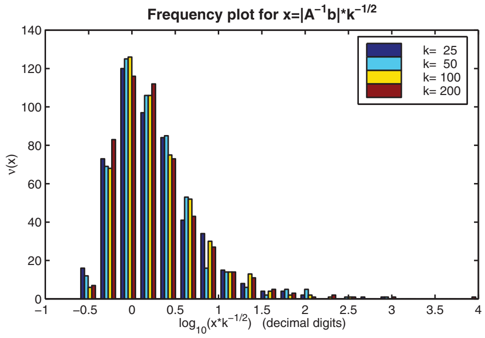

FIG. 7. e-pdf *of* $\log_{10}(\|\boldsymbol{x}\|/\sqrt{k}),\ \boldsymbol{x} \in \mathcal{Q}^k$.

TABLE 2
Parameters for $\log_{10}(\|\boldsymbol{x}\|),\ \boldsymbol{x} \in \mathcal{Q}^k_{\mathbb{C}}$.

k	Mean	Shifted	Var	stdd
25	0.816	0.117	0.080	0.283
50	0.996	0.146	0.083	0.288
100	1.097	0.097	0.064	0.253
200	1.270	0.120	0.068	0.261

So experimentally we observe that $\|\boldsymbol{x}\| \le C\sqrt{k}$ with probability close to one for moderate C.

For the classes $\mathcal{Q}^k_{\mathbb{C}}$, the solutions of complex completely random systems, similar histograms can be made; these can be found in [24]. The corresponding statistical quantities are presented in Table 2.

The calculated means as well as the histograms indicate that the variable $\log_{10}(\|\boldsymbol{x}\|/\sqrt{k})$ may have a distribution function that is nearly independent of k.

In searching for an explanation for the observed behavior, the analytical probability densities of vectors in $\mathcal{Q}^k$ and $\mathcal{Q}^k_{\mathbb{C}}$ and of their norms were discovered by chance.

Theorem 5.2. *Let $\boldsymbol{x}$ belong to $\mathcal{Q}^k$, and let the stochastic variable $x = \|\boldsymbol{x}\|$ have the probability density function f_k. Then*

$$f_k(x) = C\frac{x^{k-1}}{(1+x^2)^{(k+1)/2}} \quad \text{with} \quad C = \frac{2}{\sqrt{\pi}}\frac{\Gamma(\frac{k+1}{2})}{\Gamma(\frac{k}{2})}. \tag{28}$$

If $\boldsymbol{x}$ is in $\mathcal{Q}^N_{\mathbb{C}}$, the density function for $\|\boldsymbol{x}\|$ reads

$$\widehat{f}_k(x) = \frac{2kx^{2k-1}}{(1+x^2)^{k+1}}. \tag{29}$$

For (28), we give a sketch of the proof by induction.

Step 1. The idea for the following reduction was found in [6].

The main inductions step is based on a Householder transformation on the system

$$\boldsymbol{Ax} = \boldsymbol{b} \Longrightarrow \boldsymbol{HAx} = \boldsymbol{Hb},$$

where $\boldsymbol{H}$ represents is the unitary reflection that transforms the first column $\boldsymbol{a}_1$ of $\boldsymbol{A}$ to $\alpha\boldsymbol{e}_1$ with $\alpha = \|\boldsymbol{a}_1\|$:

$$\boldsymbol{HA} = \begin{bmatrix} \alpha & \boldsymbol{p}^T \\ \boldsymbol{0} & \widetilde{\boldsymbol{A}} \end{bmatrix}, \quad \boldsymbol{Hb} = \begin{bmatrix} c \\ \widetilde{\boldsymbol{b}} \end{bmatrix}.$$

Here the scalar $\alpha = \|\boldsymbol{a}_1\|$ satisfies the χ_k distribution; the scalar c, the vectors $\boldsymbol{p}$ and $\widetilde{\boldsymbol{b}}$ and the matrix $\widetilde{\boldsymbol{A}}$ are mutually stochastically independent, as well as independent from α, and distributed according $\mathcal{N}$, $\mathcal{N}^{k-1}$, and $\mathcal{N}^{(k-1)\times(k-1)}$, respectively.

Let $\boldsymbol{x}^T = [x_1\,,\,\widetilde{\boldsymbol{x}}^T]$; then

$$\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}, \quad x_1 = \frac{c - \boldsymbol{p}^T\widetilde{\boldsymbol{x}}}{\alpha}.$$

It follows that $\widetilde{\boldsymbol{x}}$ belongs to $\mathcal{Q}^{k-1}$. We denote the *pdf* 's of $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ by $g_1(\boldsymbol{x})$ and $\widetilde{g}_1(\widetilde{\boldsymbol{x}})$, respectively.

Step 2. The following idea of using conditional probability densities was found in [8], a two-page paper on an elementary derivation of the Wishart distribution.

Consider $\widetilde{\boldsymbol{x}}$ as a given (= nonstochastic) vector. Then the scalar $c - \boldsymbol{p}^T\widetilde{\boldsymbol{x}}$ is normally distributed with variance $1 + \|\widetilde{\boldsymbol{x}}\|^2$, so $c - \boldsymbol{p}^T\widetilde{\boldsymbol{x}} = \sqrt{1+\|\widetilde{\boldsymbol{x}}\|^2}\cdot z$, with $z \in \mathcal{N}$. Therefore

$$x_1 = \sqrt{1+\|\widetilde{\boldsymbol{x}}\|^2}\frac{z}{\alpha}.$$

The *pdf* of this variable can be considered as the *conditional pdf* of x_1 with respect to $\widetilde{\boldsymbol{x}}$. Denote the *pdf* of x_1 by $g_2(x_1)$, and it follows that

$$g_1(\boldsymbol{x}) = g_1(x_1, \widetilde{\boldsymbol{x}}) = g_2(x_1)\widetilde{g}_1(\widetilde{\boldsymbol{x}}). \tag{30}$$

Step 3. For the derivation of $g_2(x_1)$, consider the simultaneous *pdf* for z and α:

$$F(z,\alpha) = C\exp\left(-\frac{1}{2}z^2\right)\exp\left(-\frac{1}{2}\alpha^2\right)\cdot\alpha^{k-1} = C\exp\left(-\frac{1}{2}(z^2+\alpha^2)\right)\cdot\alpha^{k-1},$$

where C is a normalization constant, providing $\int F(z,\alpha)dz d\alpha = 1$.

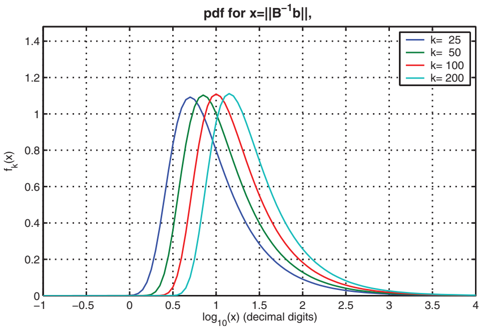

FIG. 8. *pdf's for* $\log_{10}(\|\boldsymbol{x}\|)$, $\boldsymbol{x} \in \mathcal{Q}^k$.

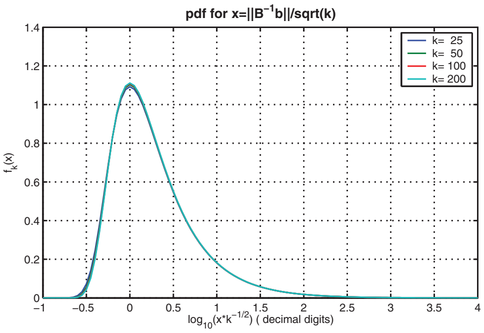

FIG. 9. *pdf's for* $\log_{10}(\|\boldsymbol{x}\|/\sqrt{k})$, $\boldsymbol{x} \in \mathcal{Q}^k$.

Choose new variables ξ and t according to $z = \xi t$, $\alpha = t$, so $\xi = z/\alpha$. The *pdf* $\widetilde{F}$ for ξ and t satisfies

$$\widetilde{F}(\xi, t) = F(\xi t, t) \cdot \left| \det \left(\frac{\partial(z, \alpha)}{\partial(\xi, t)} \right) \right| = C \exp \left(\frac{1}{2} t^2 (1 + \xi^2) \right) \cdot t^k.$$

Integrating over t, we get the marginal distribution of ξ, which turns out to be the so-called Student distribution:

$$g_3(\xi) = \frac{C'}{(1 + |\xi|^2)^{\frac{k+1}{2}}}.$$

We have $x_1 = \lambda \frac{z}{\alpha}$, with $\lambda = \sqrt{1 + \|\widetilde{\boldsymbol{x}}\|^2}$, so the *pdf* g_2 of x_1 reads

$$g_2(x_1) = \frac{1}{\lambda} g_3 \left(\frac{\xi}{\lambda} \right) = C' \frac{(1 + \|\widetilde{\boldsymbol{x}}\|^2)^{\frac{k}{2}}}{(1 + \|\boldsymbol{x}\|^2)^{\frac{k+1}{2}}}.$$

According to (30), we have the following recursion between $g_1(\boldsymbol{x})$ and $\widetilde{g}_1(\widetilde{\boldsymbol{x}})$:

$$g_1(\boldsymbol{x}) = C' \frac{(1 + \|\widetilde{\boldsymbol{x}}\|^2)^{\frac{k}{2}}}{(1 + \|\boldsymbol{x}\|^2)^{\frac{k+1}{2}}} \cdot \widetilde{g}_1(\widetilde{\boldsymbol{x}}).$$

For $k = 1$, the *pdf* g_1 coincides with g_3. Using this as initial condition for the recursion, we arrive at

$$g_1(\boldsymbol{x}) = \frac{C''}{(1 + \|\boldsymbol{x}\|^2)^{\frac{k+1}{2}}}.$$

This distribution is known as the *multivariate Cauchy distribution.*

Step 4. Equation (28) is derived by using hyperspherical coordinates $dx_1 dx_2 \cdots dx_k = r^{k-1} dr d\Omega$ and integrating over the "angle" Ω.

A complete proof, written for nonstatisticians, can be found in in a Delft University report [24].

In Figures 8 and 9 the densities $f_k(\|\boldsymbol{x}\|)$ and $f_k(\frac{\|\boldsymbol{x}\|}{\sqrt{k}})$ are plotted on a logarithmic horizontal scale, in the real case for $k = 25, 50, 100, 200$.

For this family of *pdf*'s, the expectation $\mu_k = E(\log_{10}(x))$ and the variance $\sigma_k^2 = \sigma^2(\log_{10}(x))$ can be derived analytically. We give the asymptotic behavior for large k:

$$\text{(31a)}\quad \mu_k \approx \frac{1}{2}\log_{10}(k) + 0.27586, \quad \sigma_k^2 \approx 0.23269 \qquad \text{(real case)},$$

$$\text{(31b)}\quad \mu_k \approx \frac{1}{2}\log_{10}(k) + 0.12534, \quad \sigma_k^2 \approx 0.077563 \qquad \text{(complex case)}.$$

The behavior of $\mu_k \approx C + \frac{1}{2}\log_{10}(k)$ indicates that the experimentally observed shift has a theoretical basis.

For practical use, it is interesting for which values of $\|\boldsymbol{x}\|$, the probability is less than, say, 10^{-j}. It is easy to give estimates that are quite sharp for $j \geq 2$,

$$\text{(32a)}\quad \Pr\left(\log_{10}(x) > \frac{1}{2}\log_{10}\left(\frac{2k}{\pi}\right) + j\right) < 10^{-j} \qquad \text{(real case)},$$

$$\text{(32b)}\quad \Pr\left(\log_{10}(x) > \frac{1}{2}\log_{10}(k) + j/2\right) < 10^{-j} \qquad \text{(complex case)}.$$

The graphs for different k in Figure 9 are nearly identical, again confirming the $\log_{10}(\sqrt{k})$ shifting behavior. This can be explained by an asymptotic approximation of the *pdf*. Let the stochastic variable y be defined by $x = y\sqrt{k}$; then the probability density of y satisfies

$$g_k(y)dy = f_k(x)dx = f_k(\sqrt{k}y)\sqrt{k}dy$$
$$\Longrightarrow g_k(y) = C\frac{x^{k-1}\sqrt{k}}{(1+x^2)^{\frac{1}{2}(k+1)}} = C\frac{y^{k-1}k^{\frac{k}{2}}}{(1+ky^2)^{\frac{1}{2}(k+1)}}$$

with C as defined in (28).

With the help of Stirling's formula we find $C \approx \sqrt{\frac{2k}{\pi}}$. Then by applying the elementary approximation $(1+a)^b \approx e^{ab}(1+O(a^2b))$ for $|a| < 1$, to $[1+\frac{1}{ky^2}]^{-\frac{1}{2}(k+1)}$, we arrive at

$$g_k(y) \approx \sqrt{\frac{2}{\pi}}\frac{\exp(-\frac{1}{2y^2})}{y^2}.$$

For $y \to 0$ and k fixed, $g_k(y) = O(y^{k-1})$, so it tends to zero rapidly. The asymptotic approximation tends to zero extremely fast as y^2 becomes small. Although there is a difference in behavior, this difference is hardly visible in practice. Therefore the asymptotic formula might replace the original distribution perfectly well if only k is not too small, say, $k > 20$.

This analysis explains the coinciding graphs in Figure 9.

6. Experimental verification. According to the theory described in section 4, the Lanczos residuals $\widetilde{\boldsymbol{r}}_k$ can be regarded as Krylov–Galerkin residuals with increasing Galerkin dimension k. The GMRES residuals are considered as least-squares residuals and denoted by $\widehat{\boldsymbol{r}}_k$. The residual surplus, as defined in (17b), is then $d\boldsymbol{r}_k = \widetilde{\boldsymbol{r}}_k - \widehat{\boldsymbol{r}}_k$, and its norm is calculated as

$$\|d\boldsymbol{r}_k\| = \sqrt{\|\widetilde{\boldsymbol{r}}_k\|^2 - \|\widehat{\boldsymbol{r}}_k\|^2}.$$

We want to test whether the quantities $v_k = \|d\boldsymbol{r}_k\| \,/\, \|\widehat{\boldsymbol{r}}_k\|$ can be regarded as realizations of $\|\boldsymbol{z}\|$ with $\boldsymbol{z} \in \mathcal{Q}^k$ (or $\mathcal{Q}_{\mathbb{C}}^k$ in the case of complex $\boldsymbol{P}$).

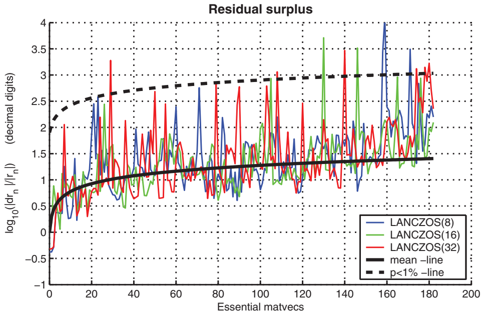

FIG. 10. $\log_{10}(\|d\widetilde{\boldsymbol{r}}\|/\|\widehat{\boldsymbol{r}}\|)$, *Prob.* 2, $\boldsymbol{P}\in\mathbb{R}^{N\times s}$.

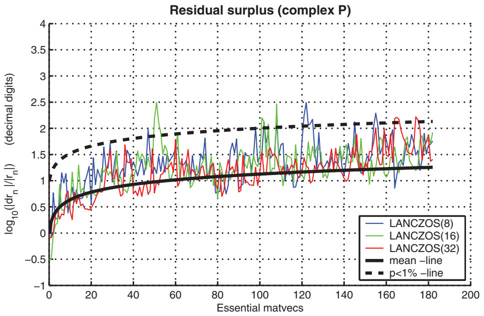

FIG. 11. $\log_{10}(\|d\widetilde{\boldsymbol{r}}\|/\|\widehat{\boldsymbol{r}}\|)$, *Prob.* 2, $\boldsymbol{P}\in\mathbb{C}^{N\times s}$.

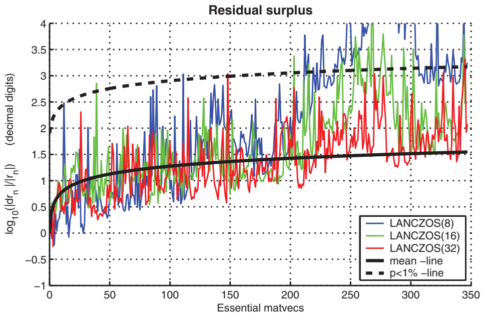

FIG. 12. $\log_{10}(\|d\widetilde{\boldsymbol{r}}\|/\|\widehat{\boldsymbol{r}}\|)$, *Prob.* 3, $\boldsymbol{P}\in\mathbb{R}^{N\times s}$.

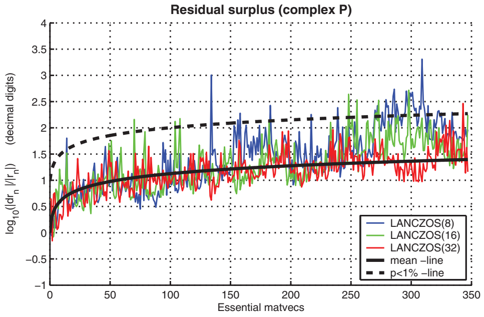

FIG. 13. $\log_{10}(\|d\widetilde{\boldsymbol{r}}\|/\|\widehat{\boldsymbol{r}}\|)$, *Prob.* 3, $\boldsymbol{P}\in\mathbb{C}^{N\times s}$.

The values v_k are calculated from the observed residual norms:

$$v_k = \sqrt{\frac{\|\widetilde{\boldsymbol{r}}_k\|^2}{\|\widehat{\boldsymbol{r}}_k\|^2} - 1}.$$

We plot $\log_{10}(v_k)$ against k. The test problems are Problem 2 (convection diffusion with numbers $[0.5, 0]$) and Problem 3 (similar, but with Peclet numbers $[20, 0]$), as described in section 2 and section 3.1. We obtained the Lanczos residuals from IDR(s) for values $s = 8\,, 16\,, 32$ and with real as well as complex $\boldsymbol{P}$.

The results are shown in Figures 10–13. In these plots, a solid black curve denotes the expectation of this stochastic variable; a dashed black curve depicts the number of digits that the Lanczos residual is behind the GMRES curve with a probability of at most 1%.

The theoretical formulas are designed for $s = \infty$. For finite s, they are valid as long as $k < s$. From the pictures we can see whether the theory remains valid for lower values of s.

The upper plots show the results for Problem 2, the lower for Problem 3. In the left plots $\boldsymbol{P}$ is real, in the right plots $\boldsymbol{P}$ is complex. It can be seen that for complex $\boldsymbol{P}$, for all three s-values the quantity $\log_{10}(v_k)$ obey the theory as if s were infinity. For real $\boldsymbol{P}$, the same can be said for the "easy problem." For the difficult problem, only for $s \geq 32$ the theoretical behavior for $s = \infty$ can be observed.

7. Discussion and conclusions.

Convergence mechanisms. We have shown that the convergence of IDR(s) depends on two mutually independent regimes. The Lanczos regime is completely determined by the choice of $\boldsymbol{P}$, whereas the damping regime depends on the choice of the ω-parameters. For convection diffusion equations with moderate mesh-Peclet numbers, the damping polynomials $\Omega_j(\boldsymbol{A})$ actually play a role in the convergence, but for $s \geq 4$ this role is negligible. Figure 4 shows an example in which the polynomials $\Omega_j(\boldsymbol{A})$ are not damping. For small values of s, the amplification is stronger than the convergence by the Lanczos part in this example. Only for higher values of s is the Lanczos component of the convergence stronger, because of the relatively low degree of the $\Omega_j(\boldsymbol{A})$ polynomials.

Since higher values of s come with a substantial increase of inner products and linear combinations, it is important that alternative choice mechanisms are used for the ω_j parameters, aiming *reduction of growth* instead of damping.

The convergence of the Lanczos part. The central part of this paper is devoted to the Lanczos part of the convergence. This part of the convergence can be regarded as produced by a Galerkin method, in which the shadow vectors and their iterated $\boldsymbol{A}^H$-images act as test vectors. Convergence of such a Galerkin method can be poor if the angle between the test space and the model space is close to $\pi/2$.

For keeping this angle away from $\pi/2$, it seems to be essential that the shadow vectors have only a weak interrelationship with the direct Krylov vectors in the process. For very large values of s, this can be accomplished by choosing the shadow vectors randomly. In this respect, the IDR(s) method belongs to the class of methods described in Halko, Martinsson, and Tropp [14], in which randomness is exploited to create vector sets that are "as independent as possible."

For small s, however, $(\boldsymbol{A}^H)^k$-images of the shadow vectors for large k are involved, introducing possibly "wrong" relationships between test vectors and model vectors. It appears that these wrong relationships (Figure 12, lower s-values) occur in the cases that the damping factors are actually amplifying factors (Figure 4).

Stochastic theory for the Lanczos residuals. In section 5, a stochastic theory was developed, based on the use of independent standard normally distributed shadow vectors, resulting in a probability distribution for the norm of the residual surplus $\|\boldsymbol{r}_{\text{Lanczos}} - \boldsymbol{r}_{\text{GMRES}}\|$. The theory is valid for very large values of s. However, Figures 10–13 show that that the results of this theory are also valid for lower values of s.

This stochastic theory in fact applies as well to the convergence behavior of the (theoretical) ML(k)BiCG method of Yeung and Chan, described in [33] as an intermediate step in the derivation of their ML(k)BiCGSTAB algorithm. In the experiments, we actually obtained the ML(k)BiCG residuals from a shadow process in the used version [30] of IDR(s), at the cost of twice the number of matrix vector operations.

The stochastic theory provides a plausible explanation of the limiting behavior of the IDR(s) residual curves for increasing s, as observed in many experiments.

Complex choice of shadow vectors. The results shown in Figure 5 demonstrate that in the case of bad damping properties caused by high mesh-Peclet numbers, this effect can be reduced considerably by choosing the shadow vectors complex instead of real. In this way, the algorithm can search optimal ω_j values in the complex plane, yielding damping instead of amplification.

For noncomplex problems, the complex choice of $\boldsymbol{P}$ is a bit expensive. The work load for inner product calculations and linear combinations is about four times as high compared to the real choice of $\boldsymbol{P}$. Only the matrix vector multiplications can be carried out in twice the work for the real case, by calculating $\Re(\boldsymbol{y}) = \boldsymbol{A}\Re(\boldsymbol{x})$, and $\Im(\boldsymbol{y}) = \boldsymbol{A}\Im(\boldsymbol{x})$ instead of simply $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

For complex problems, a real $\boldsymbol{P}$ could do the job as well, because we automatically get complex arithmetic in that case. However, Figure 13 justifies the use of complex

$\boldsymbol{P}$ also in these cases, because there is an extra advantage in the significantly lower level of the expected residual surplus.

But besides that, contributions as in [11, 18, 17] and recently in [20] are important for better mastering the stabilization/damping issue.

Bad damping by the $\Omega_j(\boldsymbol{A})$ factors certainly occurs in cases where $\boldsymbol{A}$ has eigenvalues on both sides of the imaginary axis, as in the Helmholtz equation. In [25] is shown that IDR(s) may converge very well in these cases. Unfortunately we could not test the stochastic theory on this kind of problem, because we did not succeed in obtaining reliable Lanczos residuals.

Practical expectations. Concluding from Figures 10–13, it appears that for a wide range of problems and Krylov dimensions, $\|\widetilde{\boldsymbol{r}}_{j(s+1)}^{\mathrm{IDR}(s)}\| \lesssim 10\|\boldsymbol{r}_{js}^{\mathrm{GMRES}}\|$, as long as the "damping factors" are not too amplifying. Whether a larger value of s is required can be decided if one has some prior information about the spectrum of the matrix.

Finally if the damping factors are very amplifying, it would be attractive to use the Lanczos residuals instead of the IDR(s) residuals. This would require nearly twice the number of matrix vector multiplications. But probably a large value of s is also required to reduce the angles between the left and right Krylov subspaces, which is necessary for the Lanczos residuals to converge. However, a large s-value reduces the amplification of the bad damping polynomials, which is in favor of IDR(s) itself. Experiments on this question may produce interesting results.

If IDR(s) is compared to full GMRES when the latter is in a phase of fast convergence, the factor 10 is visible as only a slight shift to the right, meaning IDR(s) is only a few iteration steps behind. For problems that are not intrinsically suffering from a matrix with an ill-structured spectrum, the IDR(s) method therefore provides a relatively cheap and reliable alternative for full GMRES.

Acknowledgments. The author wishes to thank Jos van Kan, Martin van Gijzen, Tijmen Collignon, and Kees Vuik for many fruitful discussions, Jens Zemke for his critical remarks, and Kees Vuik, Martin van Gijzen, and both referees for carefully reading the manuscript and for their many useful hints.

REFERENCES

[1] P. N. Brown, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

[2] T. F. Chan, L. de Pillis, and H. van der Vorst, *Transpose-free formulations of Lanczos-type methods for nonsymmetric linear systems*, Numer. Algorithms, 17 (1998), pp. 51–66.

[3] T. P. Collignon and M. B. van Gijzen, *Minimizing synchronization in IDR(s)*, Numer. Linear Algebra Appl., 18 (2011), pp. 805–825.

[4] J. K. Cullum and W. E. Donath, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace for large, sparse symmetric matrices*, in Proceedings of the 1994 IEEE Conference on Decision and Control, IEEE Press, Piscataway, NJ, 1974, pp. 505–509.

[5] J. Cullum and A. Greenbaum, *Relations between Galerkin and norm-minimizing iterative methods for solving linear systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 223–247.

[6] A. Edelman, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.

[7] M. Eierman and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[8] M. Ghosh and B. K. Sinha, *A simple derivation of the Wishart distribution*, American Statistician, 56 (2002), pp. 100–101.

[9] G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 361–377.

[10] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[11] M. H. Gutknecht, *Variants of BICGSTAB for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033.

[12] M. H. Gutknecht and K. J. Ressel, *Look-ahead procedures for Lanczos-type product methods based on three-term Lanczos recurrences*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1051–1078.

[13] M. H. Gutknecht and J. P. M. Zemke, *Eigenvalue Computation Based on IDR*, Research report 2010-13 SAM, ETH Zürich & Bericht 145 INS, TUHH, 2010.

[14] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness. Stochastic algorithms for constructing approximate matrix decompositions*, ACM Report 2009-05, Applied & Computational Mathematics, California Institute of Technology, 2009.

[15] M. Hochbruck and C. Lubich, *Error analysis of Krylov methods in a nutshell*, SIAM J. Sci. Comp., 19 (1998), pp. 695–701.

[16] Y. Saad and M. H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[17] V. Simoncini and D. B. Szyld, *Interpreting IDR as a Petrov-Galerkin Method*, Research report 9-10-22, Department of Mathematics, Temple University, Philadelphia, 2009.

[18] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(ℓ) for linear equations involving matrices with complex spectrum*, Electron. Trans. Numer. Anal., 1 (1994), pp. 11–32.

[19] G. L. G. Sleijpen, P. Sonneveld, and M. B. van Gijzen, *Bi-CGSTAB as an induced dimension reduction method*, Appl. Numer. Math., 60 (2010), pp. 1100–1114.

[20] G. L. G. Sleijpen and M. B. van Gijzen, *Exploiting BiCGstab(ℓ) strategies to induce dimension reduction*, SIAM J. Sci. Comput., 32 (2010), pp. 2687–2709.

[21] G. L. G. Sleijpen and H. A. van der Vorst, *Maintaining convergence properties of BiCGstab methods in finite precision arithmetic*, Numer. Algorithms, 10 (1995), pp. 203–223.

[22] G. L. G. Sleijpen and H. A. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods*, Computing, 56 (1996), pp. 141–163.

[23] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema, *BiCGSTAB(ℓ) and other hybrid Bi-CG methods*, Numer. Algorithms, 7 (1994), pp. 75–109.

[24] P. Sonneveld, *On the Statistical Properties of Solutions of Completely Random Linear Systems*, Reports of the Department of Applied Mathematical Analysis, 10–09, Delft University of Technology, 2010.

[25] P. Sonneveld and M. B. van Gijzen, *IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Statist. Comput., 31 (2008), pp. 1035–1062.

[26] M. Tanio and M. Sugihara, *GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials*, J. Comput. Appl. Math., 235 (2010), pp. 765–784.

[27] 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. Comput., 13 (1992), pp. 631–644.

[28] H. A. van der Vorst and Q. Ye, *Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals*, SIAM J. Sci. Comput., 22 (2001), pp. 835–852.

[29] M. B. van Gijzen, G. L. G. Sleijpen, and J. P. M. Zemke, *Flexible and Multi-Shift Induced Dimension Reduction Algorithms for Solving Large Sparse Linear Systems*, Reports of the Department of Applied Mathematical Analysis, 11–06, Delft University of Technology, 2011.

[30] M. B. van Gijzen and P. Sonneveld, *Algorithm* 913: *An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties*, ACM Trans. Math. Softw., 38 (2011), pp. 5:1–5:19.

[31] H. F. Walker, *Residual smoothing and peak/plateau behaviour in Krylov subspace methods*, Appl. Numer. Math., 19 (1995), pp. 279–286.

[32] P. Wesseling and P. Sonneveld, *Numerical Experiments with a Multiple Grid- and a Preconditioned Lanczos Type Method*, in Lecture Notes in Math. 771, Springer-Verlag, Berlin, 1980, pp. 543–562.

[33] M. Yeung and T. F. Chan, *ML(k)BiCGSTAB: A BiCGSTAB variant based on multiple Lanczos starting vectors*, SIAM J. Sci. Comput., 21 (1999), pp. 1263–1290.