INDUCED DIMENSION REDUCTION (IDR) METHOD

[1]Amanda ZEQIRI, [2]Armela HABILI
[1]Lecturer, Department of Applied Mathematics, Faculty of Natural Science, University of Tirana, Albania
[2]Morix Solutions (Software Development Company), Tirana, Albania

e-mail: amanda.zeqiri@fshn.edu.al

Abstract
As a set of modern successful iterative methods in numerical linear algebra, Krylov subspace methods are currently used to solve large systems of linear equations. The Induced Dimension Reduction (IDR) method was developed in 1979 by Peter Sonneveld. In 2007 Sonneveld and van Gijzen reconsidered IDR and generalized it to IDR(s). The paper explains why IDR approach differs from traditional approaches to Krylov subspace methods. IDR is a transpose free method based on the Lanczos approach for non-symmetric linear systems, successor of the Biconjugate Gradient (BiCG) method of Lanczos and predecessor of BiCGStab method. Based on the latter we also present the differences between IDR and BiCGStab method (or similar methods). Finally, some numerical result of the method are shown through different experiments that point out theoretical properties.

Key words: *Numerical Analysis, Krylov Subspaces, IDR(s), CG Method, BiCGStab method.*

1. Introduction

Let $Ax = b$ be a linear system where the general system matrix $A \in \mathbb{C}^{N\times N}$, $b \in \mathbb{C}^N$and x_0 the initial approximation of the solution with its initial residual approximation $r_0 = b - Ax_0$. The Krylov subspace approach generates solutions x_n by the recursion:

$$x_n \in x_0 + K_n(A, r_0) \tag{1.1}$$

where $K_n(A, r_0) = span\{r_0, Ar_0, A^2r_0, \dots, A^{n-1}r_0\}$ is the n-th Krylov subspace generated by A from r_0.

Since $K_n \subseteq \mathbb{C}^N$ (or $K_n \subseteq \mathbb{R}^N$ if the data are real) and based on relation (1.1) for the n-th residual $r_n = b - Ax_n$ we have the implication:

$$r_n \in r_0 + AK_n(A, r_0) \subset K_{n+1}(A, r_0) \tag{1.2}$$

The purpose is to find the approximation with short recursions very close the real solution x_* in few iterations (based on the limited computer memory). Also to approximate r_n, as small as possible, by elements of AK_n. This implies that we have to choose r_n as the perpendicular r_0 into its orthogonal projection into

AK_n, based on the minimum norm residual approach [3]. Which is the basis of the Conjugate Residual (CR) Method for Hermitian systems (very similar to the conjugate gradient method, with similar construction and convergence properties) or its variations GCR (Generalized Conjugate Residual) & GMRes (Generalized Minimal Residual) methods, more appropriate for Non-Hermitian systems [9] [10]. For this methods we have:

$$r_n \in r_0 + AK_n(A, r_0), \quad r_n \perp AK_n \tag{1.3}$$

Some other Krylov subspace methods are based on other orthogonal or oblique projections. For example Conjugate Gradient (CG) method,

$$r_n \in r_0 + AK_n(A, r_0), \quad r_n \perp K_n \tag{1.4}$$

or the Bi-Conjugate Gradient (BiCG) method based on the projection below:

$$r_n \in r_0 + AK_n(A, r_0), \quad r_n \perp \widetilde{K}_n = K_n(A^H, \tilde{r}_0) \tag{1.5}$$

where $\tilde{r}_0 \in \mathbb{C}^N$ is an arbitrary initial vector called the shadow vector and $\widetilde{K}_n = K_n(A^H, \tilde{r}_0)$ in the n-th Krylov subspace generated by A^H the complex conjugate transpose of A, from initial shadow residual $\tilde{r}_0$. In general is used matrix A^* and $\widetilde{K}_n$ are called the left Krylov subspaces [2] [3].

The vectors from Krylov subspaces can be described in terms of polynomials:

Consider $\mathcal{P}_n$ the space of polynomials of degree at most n and by relating again to (1.1):

$$x_n = \sum_{j=0}^{n-1} c_{j+1} A^j r_0 = p_{n-1}(A) r_0 \tag{1.6}$$

Implying that $\exists\, p_n \in \mathcal{P}_n$ residual polynomials that are normalized by the condition $p_n(0) = 1$ such that satisfy:

$$r_n = p_n(A) r_0 \tag{1.7}$$

Since $r_n = r_0 - Ax_n = (I - Ap_{n-1}(A)r_0) = p_n(A)r_0$. This means that the residual norm $\|r\|$ is small if $|p_n(t)|$ is small at the eigenvalues of A. The residuals in the BiCG method are defined as in (1.7) while the residuals in Conjugate Gradient Squared (CGS) method are defined by

$$r_n = p_n(A)^2 r_0 \tag{1.8}$$

In this case the residual polynomials of CGS are the squared of BiCG residual polynomials. It is obvious that at a certain eigenvalue when $|p_n(t)| \ll 1$ then $|p_n(t)^2|$ is even much smaller [5] [9].

2. The IDR approach and IDR(s)

Consider $Ax = b$ a linear system where the general system matrix $A \in \mathbb{C}^{N\times N}$, $b \in \mathbb{C}^N$and x_0 the initial approximation of the solution with its initial residual approximation $r_0 = b - Ax_0$. Let $\mathcal{G}_0$ be the complete Krylov subspace $K_n(A, r_0)$ and $\mathcal{S} \subset \mathbb{C}^N$ a linear subspace of codimention $s \ll N$ ($\mathcal{S}$ in case $s = 1$ is a hyperplane of $\mathbb{C}^N$) such that $\mathcal{G}_0 \cap \mathcal{S}$ contains no eigenvectors of A. The recursion

$$\mathcal{G}_j = (I - \omega_j A)(\mathcal{G}_{j-1} \cap \mathcal{S}), \qquad j = 1,2,3, \dots, m \quad (2.1)$$

defines the IDR subspaces $\mathcal{G}_j$ for an arbitrary m and $\omega_j \in \mathbb{C}$ non zero scalars that are chosen in order to boost convergence. This means that IDR subspaces are nested:

$$\mathcal{G}_{j+1} \subset \mathcal{G}_j \text{ for } \forall j > 0 \quad (2.2)$$

that in a finite number of steps they reduce to the null space $\mathcal{G}_j = \{0\}$ for some $j \leq N$, due to the mild restrictions of the hyperplane $\mathcal{S}$, so $r_n = 0$. The approach of Sonneveld and van Gijzen is based on a *geometric understanding*, namely, the IDR theorem [8].

Let define d_j the dimension of $\mathcal{G}_j$ as well $d_j = d_{j-1} - s$, then $(\mathcal{G}_{j-1} \cap \mathcal{S})$ is represented by $N - d_j + s$ linear equations likely to be linear independent but we can't generalize the property since $\mathcal{G}_{j-1}$ actually depends on $\mathcal{S}$.

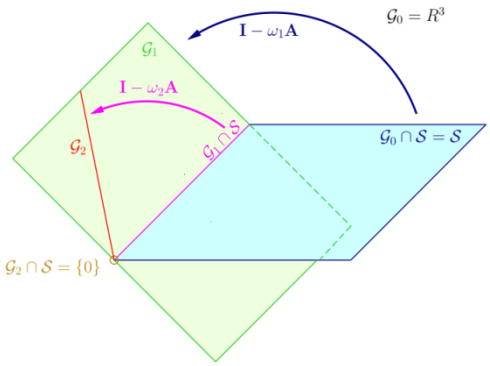

Figure 2.1: Case $s = 1$, the spaces $\mathbb{R}^3 = K_3 = \mathcal{G}_0 \supseteq \mathcal{G}_1 \supseteq \mathcal{G}_2$.

The IDR theorem can be applied by generating $s + 1$ residuals r_n that are forced to be in the subspaces $\mathcal{G}_j$, where j is nondecreasing with increasing n.

$$r_{n+1} \in \mathcal{G}_j \cap (r_0 + AK_{n+1}), \quad n = (s + 1)j \quad (2.3)$$

Then under the assumptions of IDR theorem, the system will be solved after at most N dimension-reduction steps. The recursion for the residuals is:

$$r_{n+1} = (I - \omega_j A)v_n, \quad v_n \in \mathcal{G}_{j-1} \cap \mathcal{S} \cap (r_0 + AK_n) \qquad (2.4)$$

To generate $r_{n+1} \in \mathcal{G}_j$ we need search directions vectors $v_n \in \mathcal{G}_{j-1} \cap \mathcal{S}$. To generate $v_n \in \mathcal{G}_{j-1} \cap \mathcal{S}$we need to intersect an s-dimensional affine subspace of $\mathcal{G}_{j-1}$with the subspace $\mathcal{S}$ represented by s homogeneous linear equations that we may write as $Pv_n = 0$ with an $N \times s$ matrix $P = (p_1, p_2, \dots, p_s)$ whose columns are form a basis of $\mathcal{S}^{\perp} = \mathcal{R}(P)$ (recall $\mathcal{R}$ is the Shadow space of dimension s). Also we can represent s -dimensional affine subspace of $\mathcal{G}_{j-1}$with a linear combination of $s+1$ for example last computed residuals $r_{n-s}, \dots, r_n$ in $\mathcal{G}_{j-1}$, hence v_n can be computed as:

$$v_n = r_n - \sum_{j=1}^{\hat{j}} \gamma_j \Delta r_{n-j} = r_n - \Delta R_n c_n \qquad (2.5)$$

for a fixed $s \le \hat{j} \ll N$(usually $\hat{j} = s$), $\Delta r_n = r_{n+1} - r_n$, $\Delta R_n = (\Delta r_{n-1}, \dots, \Delta r_{n-\hat{j}})$ and vector $c_n = [\gamma_1, \dots, \gamma_{\hat{j}}] \in \mathbb{R}^s$ such as $(P\Delta R_n)c_n = Pr_n$, since we want to enforce $v_n \in \mathcal{S}$ we enforce $v_n \perp \mathcal{S}^{\perp}$ meaning $Pv_n = 0$. This means that calculating the first vector in $\mathcal{G}_{j+1}$ are required $s+1$ vectors in $\mathcal{G}_j$ and we expect r_n to be in $\mathcal{G}_{j+1}$ only for $n = (j+1)(s+1)$.

Referring to (2.5) and (2.5) we can calculate residuals:

$$r_{n+1} = r_n - \Delta R_n c_n - \omega_j A(r_n - \Delta R_n c_n) \qquad (2.6)$$

This formula manifests that IDR(s) differs considerably from most commonly used Krylov subspace methods such as CG, BiCG, or GCR. The difference is that in (2.6) not only r_n is multiplied by A but also $r_{n-1}, \dots, r_n - s$.

We may choose ω_j freely in the calculation of the first residual r_{n+1} in $\mathcal{G}_j$ but the same value must be used in the calculations of the subsequent residuals in $\mathcal{G}_j$. A suitable choice for ω_j according to (2.4) is the value that minimizes $\|r_{n+1}\|$ the norm of r_{n+1} similarly as is done in the Bi-CGStab method [5]. We choose $\omega_j$such that $r_{n+1} \perp Av_n$:

$$\omega_j = \frac{\langle Av_n, v_n \rangle}{\|Av_n\|^2}, \; j = n/2 \text{ for } s = 1 \text{ and } j = n/(s+1) \text{ for } s > 1 \quad (2.7)$$

Now to approximate the solutions x_n it is not sufficient to just approximate residuals r_n and search direction v_n. We also need to approximate 'intermediate' iterates $x'_n \in x_0 + K_n(A, r_0)$ such that $v_n = b - Ax'_n$ 'intermediate' residuals. All the required updates are [5]:

$$\begin{aligned} v_n &= r_n - \Delta R_n c_n \;\; implying \; x'_n = x_n - \Delta X_n c_n \\ r_{n+1} &= \big(I - \omega_j A\big) v_n \; implying \;\; x_{n+1} = x'_n + \omega_j v_n \end{aligned} \qquad (2.8)$$

where $\Delta X_n = (\Delta x_{n-1}, \dots, \Delta x_{n-\hat{j}})$ lead us to the analog recursion of (2.6) for x_{n+1}:

$$x_{n+1} = x_n - \Delta X_n c_n + \omega_j (r_n - \Delta R_n c_n) \qquad (2.9)$$

where $\Delta r_n = -A \Delta x_n$.

3. Polynomial representations

In section 2 we saw that IDR can be generalized: instead of using one hyperplane $\mathcal{S}$ one uses the intersection of s hyperplanes. This makes the dimension reduction step less frequent but the reduction a larger one. The generalized IDR, described as IDR(s), was developed in 2006 by Peter Sonneveld and Martin van Gijzen [8]. In the context of Krylov subspace methods, IDR(s) can be thought of as a two-sided Lanczos method. The dimension reduction is viewed as the basic force behind IDR and gave the method its name. However, dimension reduction in Krylov subspace methods is not at all a unique feature of IDR. In fact, projection based methods can be understood in a similar way. For example the relation (1.5) of the BiCG residuals in the first section can be brought as:

$$r_n \in \widetilde{K}_n^{\perp} \cap r_0 + AK_n(A, r_0), \quad \widetilde{K}_n = K_n(A^H, \tilde{r}_0) \qquad (3.1)$$

The same approach can be used in other similar methods as CR, GCR or GMRes:

$$r_n \in (AK_n)^{\perp} \cap r_0 + AK_n(A, r_0), \qquad (3.2)$$

What is different in IDR is that $\mathcal{G}_j$ is not an orthogonal complement of a Krylov subspace. However, due to the form of the recursion for $\{\mathcal{G}_j\}$ turns out to be the image of an orthogonal complement of a Krylov subspace.

Let us refer again to the polynomial description, in order to analyze the dimension reduction and comperation to other similar Krylov subspace methods. IDR(s) is a Lanczos-type product method, that is most residuals can be written as (according to (2.8) and (2.4) in section 2):

$$r_{n+1} = \big(I - \omega_j A\big) v_n = \Omega_j(A) w_{n+1}, \;\; w_{n+1} \in \mathcal{G}_0 \cap \mathcal{S} \cap (r_0 + AK_{n+1-j}) \qquad (3.3)$$

where $\Omega_j(t) = (1 - \omega_1 t) \cdot (1 - \omega_2 t) \dots (1 - \omega_j t)$, $\Omega_0(t) = 1$ and $w_{n+1} \perp K_j(A^H, P)$. Generally for $n = (s+1)j$, $w_{n+1} \in (r_0 + AK_{js})$ and $w_{n+1} \perp K_j(A^H, P)$ with dimension js.

We recall that in BiCG ${r_n}^{BiCG} = p_n(A)r_0$ and shadow residuals $\tilde{r}_n{}^{BiCG} = \overline{p_n}(A^H)\tilde{r}_0,\ \ \tilde{r}_n \in \tilde{r}_0 + A^H\widetilde{K_n}$, where $\overline{p_n}$ denote the polynomial obtained fromLanczos residual polynomials p_n by complex conjugation of coefficients. Also search directions ${v_n}^{BiCG} = \sigma_n(A)r_0$ and shadow search directions $\tilde{v}_n{}^{BiCG} = \overline{\sigma_n}(A^H)\tilde{r}_0,\ \ \tilde{v}_n \in \tilde{v}_0 + A^H\widetilde{K_n}$ where σ_n are the search directions polynomials and $\overline{\sigma_n}$the polynomials obtained from σ_n by complex conjugation of coefficients. From the recursions of these four set of vectors it is derived the polynomials recursion [1]:

$$\begin{aligned} p_{n+1}(t) &= p_n(t) - t\alpha_n\sigma_n(t) \\ \sigma_{n+1}(t) &= p_{n+1}(t) - \beta_n\sigma_n(t) \end{aligned} \qquad (3.4)$$

In case $s = 1$, the subspace $\mathcal{S}$ is hyperplane determined by a single vector $p = \tilde{r}_0,\ \ p \perp \mathcal{S}$ (hence P has only one column), we can express the IDR residuals and IDR search directions in terms of the Lanczos residual polynomials as [4]:

$$r_n = \Omega_j(A)w_n = \begin{cases} \Omega_j(A)p_j(A)r_0,\ \ if\ n = 2j \\ \Omega_j(A)\hat{p}_{j+1}(A)r_0\ if\ n = 2j+1 \end{cases} \qquad (3.5)$$

$$v_n = \Omega_{j-1}(A)w_n = \begin{cases} \Omega_{j-1}(A)p_j(A)r_0,\ \ if\ n = 2j \\ \Omega_{j-1}(A)\hat{p}_{j+1}(A)r_0\ if\ n = 2j+1 \end{cases} \qquad (3.6)$$

where $w_{2j} = {r_j}^{BiCG} = p_j(A)r_0 \perp \widetilde{K}_j$ and $\hat{p}_{j+1} = \hat{p}_{j+1}(A)r_0 \perp \widetilde{K}_j$ another residual polynomial of degree $j+1$ that represents w_{2j+1} such as $\hat{p}_1 = (1 - \omega_0 t)$. The polynomials recursion for IDR(1) are:

$$\begin{aligned} p_{j+1}(t) &= \big(1 - \gamma_{2j+1}\big)\hat{p}_{j+1}(t) + \gamma_{2j+1}p_j(t),\ \ j = 0,1,2,\ldots \\ \hat{p}_{j+1}(t) &= \big(1 - \gamma_{2j}\big)\big(1 - \omega_j t\big)p_j(t) + \gamma_{2j+1}\hat{p}_j(t), j = 0,1,2,\ldots \end{aligned} \qquad (3.7)$$

where γ_n has to be chosen such that $v_{n-1} \perp p,\ v_{n-1} \in \mathcal{S}$:

$$\gamma_n = \langle p, r_n\rangle / \langle p, \Delta r_{n-1}\rangle, \text{ we use matrix } P \text{ in case } s > 1 \qquad (3.8)$$

In BiCGStab method residuals have the recurrence$r_n = \Omega_n(A)p_n(A)r_0$, [1][2].Here we get the relationship between even IDR(1) residuals and BiCGStab:

$${r_{2j}}^{IDR(1)} = {r_j}^{BiCGStab},\ \ {x_{2j}}^{IDR(1)} = x_{BiCGStab} \qquad (3.9)$$

According to (3.4) and (3.7) resurrections we express the odd IDR(1) residual in terms of quantities from BiCGStab by [4] [5]:

$${r_{2j+1}}^{IDR(1)} = {r_{2j}}^{IDR(1)} - \frac{\alpha_j}{1-\gamma_{2j+1}} A\Omega_j(A){v_j}^{BiCG} \qquad (3.10)$$

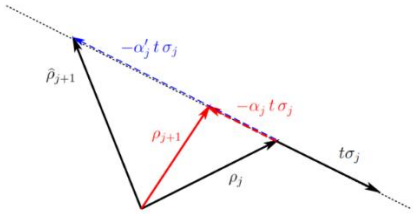

Figure 3.1: The connection between IDR(1) and BiCGStab.

While BiCGStab implicitly constructs p_{j+1} by enforcing a bi-orthogonality condition on a polynomial that lies on the line determined by p_j and $t\sigma_j$, IDR(1) first generates by the second recursion in (3.7) the polynomial $\hat{p}_{j+1}$ that lies on that line and then also enforces this condition:

$$\hat{p}_{j+1}(t) = p_j(t) - \frac{\alpha_j}{1-\gamma_{2j+1}} t\sigma_j(t) \qquad (3.11)$$

4. Numerical experiments

The algorithm used is an almost direct translation of the IDR theorem. IDR(s) algorithms are quite sensitive to round-off errors.

To illustrate the equivalence of BiCGStab and IDR we consider the following experiments: First as matrix A we choose from the MATRIX MARKET collection of Matlab the Sherman4 matrix (a real unsymmetric matrix of size $1{,}104 \times 1{,}104$) and a right-hand side vector corresponding to a solution vector that consists of ones. We have only plotted the residual norms at every $s + 1$-st step and the BiCGStab residual norms basically coincide with the IDR residual norms at these very steps. Both methods (for $s = 4$) use the recursively updated residual tocheck for convergence [7]. Figure 4.1 displays for both methods the norm of the true residual (scaled by the norm of the right-hand side vector) $\|r\|/\|b\|$ as function of the number of MATVECs.

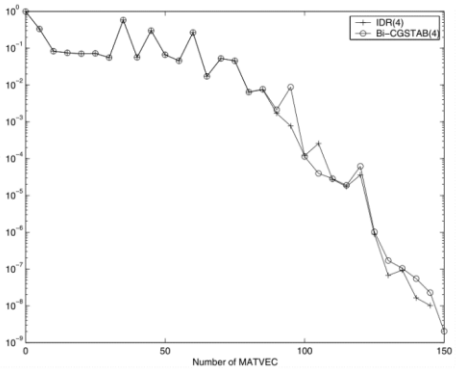

Figure 4.1: IDR(4) and BiCGStab(4) convergence for Sherman4.

IDR(s) and Bi-CGSTAB require basically the same number of MATVECs for the same dimension of the shadow space, which Table 4.1 confirms the mathematical equivalence between the methods:

Method	Number of MVs	$\|\mathbf{b}-\mathbf{Ax}\|/\|\mathbf{b}\|$
GMRES	120	9.8×10^{-9}
IDR(1)	179	6.6×10^{-9}
IDR(2)	161	8.3×10^{-9}
IDR(3)	153	8.7×10^{-9}
IDR(4)	146	3.0×10^{-9}
IDR(5)	150	1.2×10^{-9}
Bi-CGSTAB(1)	180	9.0×10^{-9}
Bi-CGSTAB(2)	162	8.6×10^{-9}
Bi-CGSTAB(3)	156	4.6×10^{-9}
Bi-CGSTAB(4)	150	1.8×10^{-7}
Bi-CGSTAB(5)	144	7.1×10^{-9}

Table 4.1: Number of MATVECs to solve the SHERMAN4 system such that $\|r\|/\|b\|$ of the updated residual is less than 10^{-8}.

We have also included GMRES MATVECs and IDR(s) and BiCGStab are very close to this number.

Now let us take another example, the problem of 3D convection-diffusion-reaction problem, described by partial differential equations on the unit cube $[0\,,1]\times[0\,,1]\times[0\,,1]$ with Dirichlet boundary conditions:

$$\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}+\frac{\partial^2 U}{\partial z^2}+1000\frac{\partial U}{\partial x}=F \qquad (4.1)$$

The vector F is defined by the solution:

$$U(x,y,z) = \exp(\mathrm{xyz})\sin(\mathrm{px})\sin(\mathrm{py})\sin(\mathrm{pz}) \qquad (4.2)$$

Using the Finite Differences Method, (4.1) is discretized using second/first order central differences for the diffusion and the convection term. Solving the implicit scheme, we have 125,000 equations, including boundary points.

This is a problem of interest since BiCGStab does not work well, due to the strong nonsymmetry of the system matrix. This matrix has eigenvalues with large imaginary parts.

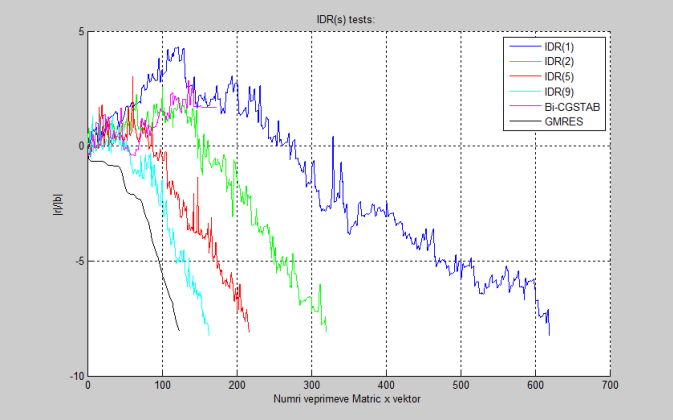

Figure 4.2: Convergence for IDR(s) (with real P and $s = 1,2,5,9$), Bi-CGSTAB and GMRES.

The convergence curves of IDR(s) are in between the convergence curves of GMRES and BiCGStab, and are increasingly closer to the GMRES curve for higher s. The iterative process is over when $\|r\|/\|b\|$ of the updated residual is less than 10^{-8}. The figure shows the poor convergence behavior of BiCGStab for this problem, and as can be expected also for IDR(1). Increasing s significantly improves the convergence behavior of IDR(s). However, compared with the optimal convergence of GMRES, the rate of convergence is still rather poor.

5. Conclusions

Transpose free CGS, BiCG, BiCGStab methods, unlike GMRES, they do not find the approximate solutions with smallest residual norm. In contrast to GMRES, these methods use short recurrences (Lanczos approach), and as a result are often much more efficient both with respect to memory and to computations, for problems where GMRES needs many iterations to find a sufficiently accurate solution.

Efforts of finding appropriate approximate solutions have mainly focused on constructing residuals with small or smallest norm. For instance, for symmetric systems (i.e. $A^T = A$), CR constructs residuals with smallest Euclidean norm, while the residuals for CG have smallest $\|A^{-1}\|$. Bi-CG has been viewed as a CG process with respect to a quadratic form rather than an inner-product, and residuals were considered to be 'minimized' with respect to this quadratic form.

An alternative approach has been taken, the construction of the IDR, 'squeezing' the residuals to zero. The IDR method finds residuals in a sequence of shrinking subspaces. BiCG can also be viewed in such a perspective. BiCG residuals belong to a sequence of growing Krylov subspaces, but they also belong to a

sequence of shrinking subspaces: BiCG uses so-called 'shadow' Krylov subspaces for testing, that is, the residuals are constructed to be orthogonal to this sequence of growing shadow spaces.

The starting point for the IDR construction is more abstract, more elegant than for BiCG. Residuals are constructed in IDR spaces $\mathcal{G}_j$. Here, $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$ that are defined recursively by $\mathcal{G}_j$. If A has no eigenvector that is orthogonal to the subspace $\mathcal{S}$, then $\mathcal{G}_{j+1} \subset \mathcal{G}_j$ and the dimension of $\mathcal{G}_j$ reduces with increasing j. The hyperplane $\mathcal{S}$ consists of all vectors orthogonal to some random sharow residual vector $\tilde{r_0}$, IDR(1). For $s > 1$ the IDR concept allows incorporation of a space of vectors orthogonal to all columns of a $N \times s$ shadow residual matrix $\widetilde{R_0}$. Unlike CR, GCR and BiCG, IDR differs since $\mathcal{G}_j$ is not an orthogonal complement of a Krylov subspace.

For small values of s for example $s = 4$, even for very complicated linear equations, IDR(s) often achieves almost the same convergence as GMRES, which means, comparable accuracy for the same number of MVs [10]. However, the additional costs per MV in IDR are modest and limited (as with Bi-CGSTAB), whereas these costs in GMRES grow proportional to the iteration step [8]. Therefore IDR(s) method provides a more cheap and reliable alternative for full GMRES. Since for a wide range of problems and Krylov dimensions, $\left\| r_n{}^{IDR(s)} \right\| \lesssim 10 \left\| r_n{}^{GMRES} \right\|$. The Lanczos approach is determined completely by the choice of P, whereas the IDR approach depends also on the choice of the ω_j parameters (2.4).

IDR(s) method has favorable features such as: uses short recurrences since is a Lanczos method, hence uses a small number of vectors for the process and a limited amount of memory. IDR(s) computes the exact solution in at most $2N$ steps (matrix-vector multiplications) in exact arithmetic (BiCGStab needs exactly $2N$ steps) and at most $N + N/s$ matrix-vector multiplications.

In IDR(s) there is also some freedom in choosing the s 'intermediate" iterates and residuals because, the vectors w_n in (3.3) need to satisfy only the weaker block orthogonality condition $w_n \perp K_j(A^*, P)$ unlike other methods ML(s)BiCG. With the IDR approach, such vectors can be obtained with much simpler recursions, and in addition, allow considerable flexibility to the algorithm. We recall that the "intermediate" residuals v_n as mentioned in (2.8) do not exist in BiCGStab.

BiCG, CGS, BiCGStab, and BiCGstab(ℓ) are essentially based on the computation of two mutually bi-orthogonal bases for the Krylov subspaces

$K_n(A, r_0)$ dhe $K_n(A^H, \tilde{r_0})$. On the other hand, the IDR method generates residuals that are forced to be in subspaces $\mathcal{G}_j$'s of decreasing dimension.

As seen in section 4 the generalized BiCGStab method and IDR(s) produce the same residuals every $s + 1$-st step and the BiCGSTAB residual norms basically coincide with the IDR residual norms at the same steps, until numerical effects start to play a role. IDR(s) and BiCGStab basically require the same number of MVs for the same dimension of the shadow space, which confirms the mathematical equivalence between the methods. All even IDR(1) approximations, requires that all BiCG residuals exists, and therefore any serious Lanczos breakdown cause BiCGStab and IDR(1) to break down at the same time. A breakdown for both methods might be due to $\omega_j = 0$ since the choice of the parameters ω_j is the same in both methods. Increasing s also yields a significant decrease in the number of iterations. IDR(s), for higher dimensions, is superior to BiCGStab (very sensitive to perturbations) and more stable.

6. Further work

There is also the possibility to extract eigenvalue information from the recurrence coefficients constructed in IDR method for large sparse matrices:

$$Av = v\lambda. \qquad (6.1)$$

The obvious reason is to showcase that IDR can be used to compute eigenvalues, so without the need for the transpose of A and using recurrences of length s [6].

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, while the pencil of the generalized Hessenberg decomposition directly produced by IDR(s) also contains eigenvalues that just represent the chosen moothing parameters [4] [6].

Experiments with eigenvalue computations based on other IDR(s) variants indicated that these methods deteriorate in finite precision; in contrast to other Krylov subspace methods based on short recurrences, we observe no multiple Ritz values, but ghost Ritz values at the roots of the stabilizing polynomials.

References

[1] Gerard L.G Sleijpen, P. Sonneveld, and Martin B. van Gijzen: Bi-CGSTAB as an induced dimension reduction method, Preprint 1369, Department of Mathematics, Utrecht University, Utrecht, the Netherlands, 2008.

[2] 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 (2) (1992) 631–644.

[3] H A.vander Vorst: Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, ISBN-13 978-0-521-81828-5.

[4] M. B van Gijzen and P. Sonneveld: An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties. Delft University of Technology, Reports of the Department of Applied Mathematical Analysis, Report 08-21, 2008.

[5] Martin H. Gutknecht; IDR in Variations ETH Zurich, Seminar for Applied Mathematics Kolloquium INS, TU Hamburg-Harburg, Jan. 28, 2009

[6] O. Rendel, A. Rizvanolli, Jens-Peter M. Zemke; IDR: A new generation of Krylov subspace methods?, Linear Algebra and its Applications 439 (2013) 1040–1061, ISSN 0024-3795.

[7] Peter Sonneveld: On the convergence behaviour of IDR(s), Delft University of Technology, Report 10-08, 2010, ISSN 1389-6520.

[8] 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. and Statist. Comput. 31:2: pp. (2008) 1035-1062.

[9] P. Sonneveld: CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 10 (1989) 36–52.

[10] Y. Saad and M.H. Schultz; GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems; SIAM J. Sci. Statist. Comput. 7: pp. 856-869.