

IDR Explained

Martin H. Gutknecht

ETH Zurich, Seminar for Applied Mathematics

8th GAMM Workshop Applied and Numerical Linear Algebra
Hamburg-Harburg, Sep. 11-12, 2008



History

IDR basics

Case $s=1$

Case $s>1$

Conclusions

History: References



P. WESSELING AND P. SONNEVELD, *Numerical experiments with a multiple grid and a preconditioned Lanczos type method*, in Approximation Methods for Navier-Stokes Problems, R. Rautmann, ed., Lecture Notes in Mathematics, vol. 771, Springer, 1980, pp. 543–562.

Introduce **Induced Dimension Reduction (IDR) method**, attributed to Sonneveld (“Public. in preparation”), on $3\frac{1}{2}$ pages: a “Lanczos-type method” for nonsymmetric linear systems which does not require \mathbf{A}^T .



P. SONNEVELD, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

Received Apr. 24, 1984. Introduces **(Bi)CG Squared**.

History: References (cont'd)



H. A. VAN DER VORST, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.

Received May 21, 1990. Presented at Householder Tylosand. 1st preprint: “**CGSTAB**: A more smoothly converging variant of CG-S”, coauthored by P. Sonneveld.



M.-C. 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.

Received May 16, 1997. Introduce **ML(k)BiCGSTAB** — a version of BiCGSTAB with “multiple left (shadow) residuals”. Astonishing numerical results.

History: References (cont'd)



P. SONNEVELD AND M. B. VAN GIJZEN, *IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, Report 07-07, Department of Applied Mathematical Analysis, Delft University of Technology.

Generalizing $IDR \approx IDR(1)$ to **IDR(s)**. Detailed description, connection to BiCGSTAB; very good numerical results.



G. SLEIJPEN, P. SONNEVELD, AND M. B. VAN GIJZEN, *Bi-CGSTAB as an induced dimension reduction method*, Report 08-07, Department of Applied Mathematical Analysis, Delft University of Technology.

Partly different view; partly different notation. Introduce variation of $ML(k)BiCGSTAB$; but no comparison.

IDR(s) basics: the setting

Given: linear system $\mathbf{Ax} = \mathbf{b} \in \mathbb{C}^N$, initial approx. \mathbf{x}_0 . Let:

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0,$$

$$\mathcal{K}_m := \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) := \text{span} \{ \mathbf{r}_0, \mathbf{Ar}_0, \dots, \mathbf{A}^{m-1} \mathbf{r}_0 \},$$

ν such that $\mathcal{G}_0 := \mathcal{K}_\nu$ invariant,

$\mathcal{S} \subset \mathbb{C}^N$ linear subspace of dimension $N - s$,

for $j = 1, 2, \dots$: choose $\omega_j \neq 0$ and let

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

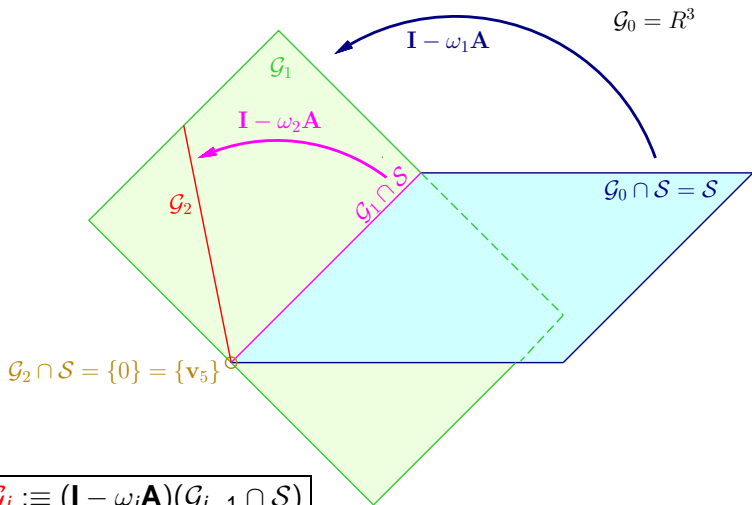
for $n = n_j, \dots, n_{j+1} - 1$:

choose \mathbf{x}_n such that

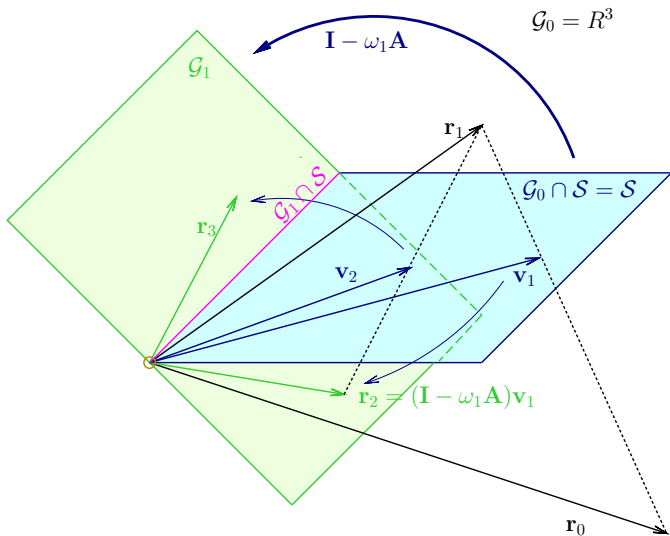
$$\mathbf{r}_n \in \mathcal{G}_j \cap \underbrace{(\mathbf{r}_0 + \mathbf{AK}_n)}_{\subset \mathcal{K}_{n+1}}.$$

Note: Typically $n_{j+1} := n_j + s + 1$.

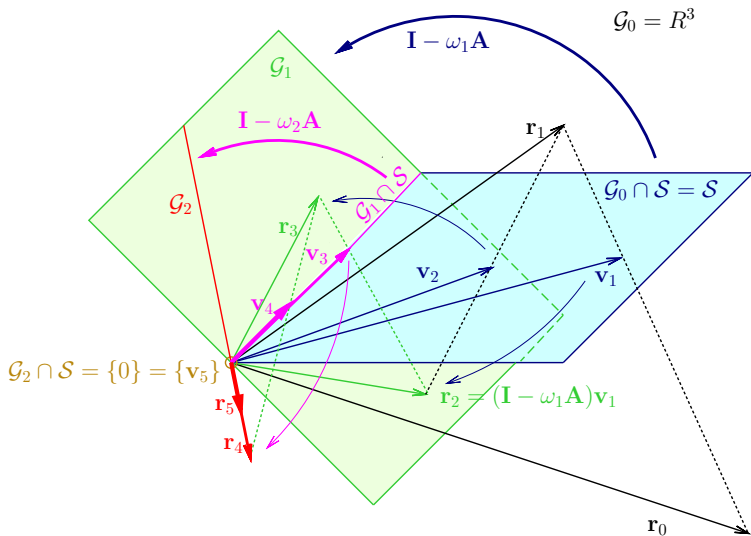
IDR(s) basics: the spaces \mathcal{G}_j (case $s = 1$)



IDR(s) basics: the first three steps (case $s = 1$)



IDR(s) basics: all the residuals (case $s = 1$)



IDR(s) basics: IDR theorem

Recall: $\mathcal{G}_j \equiv (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad \mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n).$

Genericness assumption: $\mathcal{S} \cap \mathcal{G}_0$ contains no eigenvector of \mathbf{A} .

THEOREM (IDR THEOREM (WES/SON80, SON/VGI07))

$$\mathcal{G}_j \subsetneq \mathcal{G}_{j-1} \text{ unless } \mathcal{G}_{j-1} = \{\mathbf{o}\}.$$

IDR(s) basics: IDR theorem

Recall: $\mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad \mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{AK}_n).$

Genericness assumption: $\mathcal{S} \cap \mathcal{G}_0$ contains no eigenvector of \mathbf{A} .

THEOREM (IDR THEOREM (WES/SON80, SON/VGI07))

$$\mathcal{G}_j \subsetneq \mathcal{G}_{j-1} \text{ unless } \mathcal{G}_{j-1} = \{\mathbf{o}\}.$$

Consequently: $\mathcal{G}_j = \{\mathbf{o}\}$ for some $j \leq N$.

Can expect: $\mathbf{r}_n = \mathbf{o}$ once $j = N$, that is, $n = (s + 1)N$.

But typically: $\mathbf{r}_n = \mathbf{o}$ once $n = N$, because residuals turn out to be linearly independent.

Hence: IDR Thm. strongly underestimates convergence rate.

IDR(s) basics: what's different?

Most currently used **KSS** (= Krylov subspace solvers) are based on a different kind of “induced dimension reduction”:

$$\mathbf{r}_n \in \mathcal{L}_n^\perp \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)) ,$$

where, e.g.,

$$\mathcal{L}_n = \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (\text{CG}) ,$$

$$\mathcal{L}_n = \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (\text{CR, GCR, GMRES}) ,$$

$$\mathcal{L}_n = \mathcal{K}_n(\mathbf{A}^*, \tilde{\mathbf{r}}_0) \quad (\text{BiCG}) .$$

What differs now is the special recursion for $\{\mathcal{G}_j\}$.

IDR(s) basics: what's different?

Most currently used **KSS** (= Krylov subspace solvers) are based on a different kind of “induced dimension reduction”:

$$\mathbf{r}_n \in \mathcal{L}_n^\perp \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)) ,$$

where, e.g.,

$$\mathcal{L}_n = \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (\text{CG}) ,$$

$$\mathcal{L}_n = \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (\text{CR, GCR, GMRES}) ,$$

$$\mathcal{L}_n = \mathcal{K}_n(\mathbf{A}^*, \tilde{\mathbf{r}}_0) \quad (\text{BiCG}) .$$

What differs now is the special recursion for $\{\mathcal{G}_j\}$.

\mathcal{G}_j is not the orthogonal complement of a Krylov subspace.

However, we will see that \mathcal{G}_j is the image of an orthogonal complement of a Krylov subspace.

IDR(s) basics: recursions for $\{\mathbf{r}_n\}$

Recall: $\mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad \mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n).$

Assume: $\mathbf{r}_n \notin \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-1} \quad (\forall n).$

$\Rightarrow \mathbf{r}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n, \quad \mathbf{v}_n \in \mathcal{G}_{j-1} \cap \mathcal{S} \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n),$

$\Rightarrow \mathbf{v}_n := \mathbf{r}_n - \sum_{k=1}^{K(n)} \gamma_k^{(n)} \Delta \mathbf{r}_{n-k} = \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n, \quad (1)$

where

$$K(n) \leq n,$$

$$\Delta \mathbf{r}_n := \mathbf{r}_{n+1} - \mathbf{r}_n,$$

$$\Delta \mathbf{R}_n := \begin{bmatrix} \Delta \mathbf{r}_{n-1} & \dots & \Delta \mathbf{r}_{n-K(n)} \end{bmatrix},$$

$$\mathbf{c}_n := \begin{bmatrix} \gamma_1^{(n)} & \dots & \gamma_{K(n)}^{(n)} \end{bmatrix}.$$

“Consistency condition” for residuals is implicit in (1).

Recall:
$$\mathbf{v}_n = \mathbf{r}_n - \sum_{k=1}^{K(n)} \gamma_k^{(n)} \Delta \mathbf{r}_{n-k} = \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n \in \mathcal{G}_j \cap \mathcal{S}. \quad (1)$$

Since $\dim \mathcal{S} = N - s$, there is $\mathbf{P} \in \mathbb{C}^{N \times s}$ s.t. $\mathcal{S}^\perp = \mathcal{R}(\mathbf{P})$:

$$\mathbf{v}_n \in \mathcal{S} \iff \mathbf{v}_n \perp \mathcal{S}^\perp = \mathcal{R}(\mathbf{P}) \iff \mathbf{P}^* \mathbf{v}_n = \mathbf{o}.$$

To achieve this, the term $\Delta \mathbf{R}_n \mathbf{c}_n$ in (1) must be the oblique projection of \mathbf{r}_n into $\mathcal{R}(\Delta \mathbf{R}_n)$ along \mathcal{S} .

Recall:
$$\mathbf{v}_n = \mathbf{r}_n - \sum_{k=1}^{K(n)} \gamma_k^{(n)} \Delta \mathbf{r}_{n-k} = \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n \in \mathcal{G}_j \cap \mathcal{S}. \quad (1)$$

Since $\dim \mathcal{S} = N - s$, there is $\mathbf{P} \in \mathbb{C}^{N \times s}$ s.t. $\mathcal{S}^\perp = \mathcal{R}(\mathbf{P})$:

$$\mathbf{v}_n \in \mathcal{S} \iff \mathbf{v}_n \perp \mathcal{S}^\perp = \mathcal{R}(\mathbf{P}) \iff \mathbf{P}^* \mathbf{v}_n = \mathbf{o}.$$

To achieve this, the term $\Delta \mathbf{R}_n \mathbf{c}_n$ in (1) must be the oblique projection of \mathbf{r}_n into $\mathcal{R}(\Delta \mathbf{R}_n)$ along \mathcal{S} .

In order that this projection is uniquely defined, we need $\mathbf{P}^* \Delta \mathbf{R}_n$ to be nonsingular. Then

$$\mathbf{v}_n = \mathbf{r}_n - \Delta \mathbf{R}_n \underbrace{(\mathbf{P}^* \Delta \mathbf{R}_n)^{-1} \mathbf{P}^* \mathbf{r}_n}_{\equiv: \mathbf{c}_n} = \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n.$$

We need $K(n) = s (\forall n)$ to make $\mathbf{P}^* \Delta \mathbf{R}_n$ a square matrix.

Recall: $\mathbf{r}_{n+1} = (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n$ (2)

Here, ω_j is fixed for $n + 1 = n_j, \dots, n_{j+1}$.

So, only for $n + 1 = n_j$, we may choose ω_j s.t. $\|\mathbf{r}_{n+1}\|$ is minimal among all \mathbf{r} of the form $\mathbf{r} = (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n$, i.e., $\mathbf{r} \perp \mathbf{A} \mathbf{v}_n$:

$$\omega_j := \frac{\langle \mathbf{A} \mathbf{v}_n, \mathbf{v}_n \rangle}{\|\mathbf{A} \mathbf{v}_n\|^2}.$$

Recall: $\mathbf{r}_{n+1} = (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n$ (2)

Here, ω_j is fixed for $n + 1 = n_j, \dots, n_{j+1}$.

So, only for $n + 1 = n_j$, we may choose ω_j s.t. $\|\mathbf{r}_{n+1}\|$ is minimal among all \mathbf{r} of the form $\mathbf{r} = (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n$, i.e., $\mathbf{r} \perp \mathbf{A} \mathbf{v}_n$:

$$\omega_j := \frac{\langle \mathbf{A} \mathbf{v}_n, \mathbf{v}_n \rangle}{\|\mathbf{A} \mathbf{v}_n\|^2}.$$

Note: Inserting (1) in (2) yields

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n - \omega_j \mathbf{A} (\mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n) \quad (3)$$

which means that IDR(s) with $K(n) = s$ is a (k, ℓ) -step method (G. '89_{NM}) with $k = \ell = s + 1$.

Recall: CG, BICG are (2, 1)-step methods, ORTHOMIN(k) is a $(k, 1)$ -step method.

IDR(s) basics: recursions for $\{\mathbf{x}_n\}$

Note 1: $\mathbf{v}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n \implies \exists \mathbf{x}'_n$ s.t. $\mathbf{v}_n = \mathbf{b} - \mathbf{A}\mathbf{x}'_n$,
i.e., \mathbf{v}_n is the residual of an “intermediate” iterate $\mathbf{x}'_n \in \mathbf{x}_0 + \mathcal{K}_n$.

Note 2: $\Delta \mathbf{r}_n = -\mathbf{A} \Delta \mathbf{x}_n$, $\Delta \mathbf{R}_n = -\mathbf{A} \Delta \mathbf{R}_n$,

Hence:

$$\mathbf{v}_n := \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n \implies \mathbf{x}'_n := \mathbf{x}_n - \Delta \mathbf{X}_n \mathbf{c}_n, \quad (4)$$

$$\mathbf{r}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n \implies \mathbf{x}_{n+1} := \omega_j \mathbf{v}_n + \mathbf{x}'_n. \quad (5)$$

IDR(s) basics: recursions for $\{\mathbf{x}_n\}$

Note 1: $\mathbf{v}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n \implies \exists \mathbf{x}'_n$ s.t. $\mathbf{v}_n = \mathbf{b} - \mathbf{A}\mathbf{x}'_n$,
i.e., \mathbf{v}_n is the residual of an “intermediate” iterate $\mathbf{x}'_n \in \mathbf{x}_0 + \mathcal{K}_n$.

Note 2: $\Delta \mathbf{r}_n = -\mathbf{A} \Delta \mathbf{x}_n$, $\Delta \mathbf{R}_n = -\mathbf{A} \Delta \mathbf{R}_n$,

Hence:

$$\mathbf{v}_n := \mathbf{r}_n - \Delta \mathbf{R}_n \mathbf{c}_n \implies \mathbf{x}'_n := \mathbf{x}_n - \Delta \mathbf{X}_n \mathbf{c}_n, \quad (4)$$

$$\mathbf{r}_{n+1} := (\mathbf{I} - \omega_j \mathbf{A}) \mathbf{v}_n \implies \mathbf{x}_{n+1} := \omega_j \mathbf{v}_n + \mathbf{x}'_n. \quad (5)$$

There are several ways to rearrange these four recursions and to combine them with the iterate-residual relationships; see [Sle/Son/vGi08]. Also [Son/vGi07] use different, but equivalent recursions that can be viewed as implementation of (3) and its analog for \mathbf{x}_{n+1} . (Even in their “prototype algorithm”.)

IDR(s) basics: characterization by orthogonality

THEOREM (SON/VGI07, SLE/SON/VGI08)

Let $\Omega_0(t) \equiv 1$, $\Omega_j(t) \equiv (1 - \omega_1 t) \cdots (1 - \omega_j t) \in \mathcal{P}_j^\circ$, where $\mathcal{P}_j^\circ \equiv \{\text{polyns. of degree } \leq j \text{ that are } 1 \text{ at } 0\}$. Then

$$\mathcal{G}_j = \left\{ \Omega_j(\mathbf{A})\mathbf{w} \mid \mathbf{w} \perp \underbrace{\mathcal{K}_j(\mathbf{A}^*, \mathbf{P})}_{\equiv: \mathcal{L}_j} \right\} = \Omega_j(\mathbf{A}) \underbrace{[\mathcal{K}_j(\mathbf{A}^*, \mathbf{P})]^\perp}_{=\mathcal{L}_j^\perp}.$$

Note: \mathcal{L}_j is the left-hand side (LHS) block Krylov space that appears in the block Lanczos process with LHS block size s .

Note: We may have Lanczos breakdowns and a collapsing block Krylov space (which requires deflation).

Since $\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$, and since the residual polynomials must have full degree, we have for $n = n_j, \dots, n_{j+1} - 1$:

$$\mathbf{r}_n = \Omega_j(\mathbf{A}) \mathbf{w}_n, \quad \mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j}) \cap \mathcal{L}_j^\perp, \quad \mathbf{w}_n \notin \mathcal{K}_{n-j}. \quad (6)$$

Since $\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$, and since the residual polynomials must have full degree, we have for $n = n_j, \dots, n_{j+1} - 1$:

$$\mathbf{r}_n = \Omega_j(\mathbf{A}) \mathbf{w}_n, \quad \mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j}) \cap \mathcal{L}_j^\perp, \quad \mathbf{w}_n \notin \mathcal{K}_{n-j}. \quad (6)$$

Generically, $n_{j+1} - n_j = s + 1$, so, for fixed j , we hope for $s + 1$ linearly independent vectors \mathbf{w}_n with $n_j \leq n < n_{j+1}$.

Generically, for $n = n_j = j(s + 1)$, where $\mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{js})$ and $\mathbf{w}_n \perp \mathcal{L}_j$ with $\dim \mathcal{L}_j = js$, there is a unique \mathbf{w}_n satisf. (6).

Since $\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$, and since the residual polynomials must have full degree, we have for $n = n_j, \dots, n_{j+1} - 1$:

$$\mathbf{r}_n = \Omega_j(\mathbf{A}) \mathbf{w}_n, \quad \mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j}) \cap \mathcal{L}_j^\perp, \quad \mathbf{w}_n \notin \mathcal{K}_{n-j}. \quad (6)$$

Generically, $n_{j+1} - n_j = s + 1$, so, for fixed j , we hope for $s + 1$ linearly independent vectors \mathbf{w}_n with $n_j \leq n < n_{j+1}$.

Generically, for $n = n_j = j(s + 1)$, where $\mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{js})$ and $\mathbf{w}_n \perp \mathcal{L}_j$ with $\dim \mathcal{L}_j = js$, there is a unique \mathbf{w}_n satisf. (6).

Cost of computing $\mathbf{r}_{n_j} \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n_j}$: $n_j + 1$ MVs with \mathbf{A} .

But the s other vectors \mathbf{w}_n are not uniquely determined by (6).

Since $\mathbf{r}_n \in \mathcal{G}_j \cap (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_n)$, and since the residual polynomials must have full degree, we have for $n = n_j, \dots, n_{j+1} - 1$:

$$\mathbf{r}_n = \Omega_j(\mathbf{A}) \mathbf{w}_n, \quad \mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n-j}) \cap \mathcal{L}_j^\perp, \quad \mathbf{w}_n \notin \mathcal{K}_{n-j}. \quad (6)$$

Generically, $n_{j+1} - n_j = s + 1$, so, for fixed j , we hope for $s + 1$ linearly independent vectors \mathbf{w}_n with $n_j \leq n < n_{j+1}$.

Generically, for $n = n_j = j(s + 1)$, where $\mathbf{w}_n \in (\mathbf{r}_0 + \mathbf{A}\mathcal{K}_{js})$ and $\mathbf{w}_n \perp \mathcal{L}_j$ with $\dim \mathcal{L}_j = js$, there is a unique \mathbf{w}_n satisf. (6).

Cost of computing $\mathbf{r}_{n_j} \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{n_j}$: $n_j + 1$ MVs with \mathbf{A} .

But the s other vectors \mathbf{w}_n are not uniquely determined by (6).

Note: The larger s , the larger is \mathcal{L}_j and the smaller is \mathcal{G}_j .

Note: For $n = n_j - 1$ and $n = n_j$, the polynomials associated with \mathbf{w}_n have the same degree: $\mathbf{w}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_{js}$.

$s = 1$: IDR(1) \sim BICGSTAB

In this case, the recursions simplify since $K(n) = s = 1$ and every other set of vectors ($\mathbf{w}_n, \mathbf{r}_n, \mathbf{v}_{n-1}, \mathbf{x}_n, \dots$) is uniquely determined — up to the choice of the parameters ω_j .

If the latter are chosen as in BICGSTAB,

$$\mathbf{r}_{2j} = \mathbf{r}_j^{\text{STAB}}, \quad \mathbf{x}_{2j} = \mathbf{x}_j^{\text{STAB}}, \quad \mathbf{w}_{2j} = \mathbf{r}_j^{\text{BICG}},$$

where $\mathbf{r}_j^{\text{BICG}}$ is the j th residual of BICG: $\mathbf{r}_j^{\text{BICG}} = \rho_j(\mathbf{A})\mathbf{r}_0$.

Recursions (4) and (5), with $\gamma_n := \gamma_1^{(n)} = \langle \mathbf{p}, \mathbf{r}_n \rangle / \langle \mathbf{p}, \Delta \mathbf{r}_{n-1} \rangle$:

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

(7)

s = 1: polynomial recursions

$$\mathbf{r}_n = \Omega_j(\mathbf{A})\mathbf{w}_n = \begin{cases} \Omega_j(\mathbf{A})\rho_j(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j, \\ \Omega_j(\mathbf{A})\hat{\rho}_{j+1}(\mathbf{A})\mathbf{r}_0 & \text{if } n = 2j + 1, \end{cases}$$

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

s = 1: polynomial recursions

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

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

$$\begin{aligned} \hat{\rho}_{j+1}(t) &:= (1 - \gamma_{2j})(1 - \omega_j t)\rho_j(t) + \gamma_{2j}\hat{\rho}_j(t), \\ \rho_{j+1}(t) &:= (1 - \gamma_{2j+1})\hat{\rho}_{j+1}(t) + \gamma_{2j+1}\rho_j(t). \end{aligned} \quad (8)$$

Recall: $\mathbf{w}_{2j} = \rho_j(\mathbf{A})\mathbf{r}_0 \perp \mathcal{L}_j$ and $\mathbf{w}_{2j+1} = \hat{\rho}_j(\mathbf{A})\mathbf{r}_0 \perp \mathcal{L}_j$.

s = 1: FOPs, BICG, BICGSTAB

The Lanczos (residual) polynomials ρ_j and the BICG search direction polynomials σ_j are **formal orthogonal polynomials (FOPs)** in the sense that, for $i \neq j$,

$$\rho_i \perp \rho_j \iff \underbrace{\langle \overline{\rho}_i(\mathbf{A}^*) \tilde{\mathbf{r}}_0, \rho_j(\mathbf{A}) \mathbf{r}_0 \rangle}_{= \langle \tilde{\mathbf{r}}_i^{\text{BICG}}, \mathbf{r}_j^{\text{BICG}} \rangle} = 0,$$

$$\sigma_i \perp_t \sigma_j \iff \underbrace{\langle \overline{\sigma}_i(\mathbf{A}^*) \tilde{\mathbf{r}}_0, \mathbf{A} \sigma_j(\mathbf{A}) \mathbf{r}_0 \rangle}_{= \langle \tilde{\mathbf{v}}_i^{\text{BICG}}, \mathbf{A} \mathbf{v}_j^{\text{BICG}} \rangle = \langle \tilde{\mathbf{v}}_i^{\text{BICG}}, \mathbf{v}_j^{\text{BICG}} \rangle_{\mathbf{A}}} = 0,$$

where $\mathbf{v}_j^{\text{BICG}}$ and $\tilde{\mathbf{v}}_i^{\text{BICG}}$ are the search directions and the “shadow” search directions, respectively. In particular:

$$\boxed{\rho_j \perp \mathcal{P}_{j-1}, \quad \sigma_j \perp_t \mathcal{P}_{j-1},} \iff \boxed{\mathbf{r}_j^{\text{BICG}} \perp \mathcal{L}_j, \quad \mathbf{v}_j^{\text{BICG}} \perp_{\mathbf{A}} \mathcal{L}_j.}$$

BICG uses in its standard version coupled two-term recursions:

$$\mathbf{r}_{j+1}^{\text{BICG}} := \mathbf{r}_j^{\text{BICG}} - \alpha_j \mathbf{A} \mathbf{v}_j^{\text{BICG}}, \quad \mathbf{v}_{j+1}^{\text{BICG}} := \mathbf{r}_{j+1}^{\text{BICG}} + \beta_j \mathbf{v}_j^{\text{BICG}}.$$

The corresp. recursions for ρ_j and σ_j are

$$\underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} := \underbrace{\rho_j(t)}_{\perp \mathcal{P}_{j-1}} - \alpha_j \underbrace{t \sigma_j(t)}_{\perp \mathcal{P}_{j-1}}, \quad \underbrace{\sigma_{j+1}(t)}_{\perp t \mathcal{P}_j} := \underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} + \beta_j \underbrace{\sigma_j(t)}_{\perp t \mathcal{P}_{j-1}}.$$

In contrast, in IDR(1), by (8),

$$\underbrace{\hat{\rho}_{j+1}(t)}_{\perp \mathcal{P}_{j-1}} := (1 - \gamma_{2j}) \underbrace{(1 - \omega_j t) \rho_j(t)}_{\perp \mathcal{P}_{j-2}} + \gamma_{2j} \underbrace{\hat{\rho}_j(t)}_{\perp \mathcal{P}_{j-2}},$$
$$\underbrace{\rho_{j+1}(t)}_{\perp \mathcal{P}_j} := (1 - \gamma_{2j+1}) \underbrace{\hat{\rho}_{j+1}(t)}_{\perp \mathcal{P}_{j-1}} + \gamma_{2j+1} \underbrace{\rho_j(t)}_{\perp \mathcal{P}_{j-1}}.$$

Comparing the recursions for (ρ_j, σ_j) with those for $(\rho_j, \hat{\rho}_j)$ we easily see:

$$(1 - \gamma_{2j+1}) (\hat{\rho}_{j+1}(t) - \rho_j(t)) = -\alpha_j t \sigma_j(t),$$

or,

$$\hat{\rho}_{j+1}(t) = \rho_j(t) - \frac{\alpha_j}{1 - \gamma_{2j+1}} t \sigma_j(t),$$

or,

$$\mathbf{r}_{2j+1} = \mathbf{r}_{2j} - \frac{\alpha_j}{(1 - \gamma_{2j+1})} \underbrace{\mathbf{A} \Omega_j(\mathbf{A}) \mathbf{v}_j^{\text{BICG}}}_{\equiv: \mathbf{s}_j^{\text{STAB}}}.$$

This formula expresses the odd indexed IDR(1) residuals in terms of quantities from BICGSTAB and the coefficient γ_{2j+1} .

$s = 1$: Comments and conclusions

- Even indexed IDR(1) residuals are BICGSTAB residuals.

$s = 1$: Comments and conclusions

- Even indexed IDR(1) residuals are BICGSTAB residuals.
- ρ_j and σ_j are fully determined by their degree, their orthogonality condition, the consistency condition $\rho_j(0) = 1$, and equal leading coefficients of ρ_j and σ_j .

s = 1: Comments and conclusions

- Even indexed IDR(1) residuals are BICGSTAB residuals.
- ρ_j and σ_j are fully determined by their degree, their orthogonality condition, the consistency condition $\rho_j(0) = 1$, and equal leading coefficients of ρ_j and σ_j .
- In IDR, ρ_j is the same as in BICG. But $\hat{\rho}_j$ is not determined uniquely by degree, orthogonality, and $\hat{\rho}_j(0) = 1$. The last degree of freedom is fixed by the form of the recursions.

- IDR(1) can be viewed as a minor variation of BICGSTAB.
- It is not clear, why one or the other should be more stable.

- IDR(1) can be viewed as a minor variation of BICGSTAB.
- It is not clear, why one or the other should be more stable.
- In fact, the existence of all even indexed IDR(1) residuals requires (like the BIORES version of BICG) that no Lanczos breakdowns and no pivot breakdowns occur.
- The smoothing step breakdown ($\omega_j = 0$) is also the same and can be treated easily by choosing a non-optimal ω_j .

- Variations of the “prototype algorithm” for IDR(1) method allow us to avoid certain breakdowns; but not all details are clear.
- Likewise there are variations of BICGSTAB that avoid breakdowns by using look-ahead [G./Ressel '00_{SIMAX}]; they may be more complicated, however.
- So far there are no IDR analogs of BICGSTAB2 [G. '93_{SISC}] or BICGSTAB(ℓ) [Sleijpen/Fokkema '93_{ETNA}]. In particular, IDR(s) with $s > 1$ has nothing to do with BICGSTAB(ℓ) with $\ell > 1$.

s = 1: How did the original IDR differ from IDR(1)?

In contrast to IDR(1) of [Son/vGi07], where we had

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

the original IDR of [Wes/Son80] used for n odd the recursions

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

with $\gamma'_n \equiv \gamma_1^{(n)} = \langle \mathbf{p}, \mathbf{r}_n \rangle / \langle \mathbf{p}, \Delta \mathbf{r}_{n-2} \rangle$.

s = 1: How did the original IDR differ from IDR(1)?

In contrast to IDR(1) of [Son/vGi07], where we had

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

the original IDR of [Wes/Son80] used for n odd the recursions

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

with $\gamma'_n \equiv \gamma_1^{(n)} = \langle \mathbf{p}, \mathbf{r}_n \rangle / \langle \mathbf{p}, \Delta \mathbf{r}_{n-2} \rangle$.

Moreover, in discrepancy of what we have stated here, the new IDR(s) computes the residual differences as $\Delta \mathbf{R} = -\mathbf{A} \Delta \mathbf{X}$. This couples the recursions for \mathbf{x}_n and \mathbf{r}_n more tightly.

$s > 1$: $IDR(s) \sim ML(s)BICGSTAB$

- Relation $IDR(1) \sim BICGSTAB \Leftarrow BICG$ is matched by relation $IDR(s) \sim ML(s)BICGSTAB \Leftarrow ML(s)BICG$.
- $ML(s)BICG$ and $ML(s)BICGSTAB$ are due to Man-Chung Yeung and Tony Chan '97/'99_{SISC}. Connection to nonsym. block Lanczos [Aliaga/Boley/Freund/Hernández '96/'99_{MC}].
- Oddly, essentially the same methods are also introduced in [Sle/Son/vGi08]. Yeung/Chan are cited in introduction only.

- The fundamental discovery that, in the framework of Lanczos-type product methods, multiple left projections can both speed up the convergence and reduce the MV count (per search space dimension) is due to Yeung and Chan. But Sonneveld and van Gijzen rediscovered it 10 years later independently.
- The Yeung/Chan paper is well written, but the formulas look horrible. Amazingly, their Matlab program <http://www.uwo.edu/mathmyeung/p12/mlbicgstab.txt> is only 187 lines (incl. 30 lines of comments).

Conclusions

- IDR(1) is as good as BICGSTAB or better.
- IDR(s) is as good as ML(s)BICGSTAB or better.
- Typically, IDR(s) outperforms other methods for a nonsymmetric problem.
- What is missing is the IDR-like generalization of BiCGStab2 to cover the case where \mathbf{A} is real, but has (strongly) non-real eigenvalues.