Outline

Prerequisites

History

IDR basics

Case $s=1$

Case $s>1$

Conclusions

Appendix: original IDR

Appendix: another variant: IDR($s$)BiO
Prerequisites: Krylov (sub)space solvers

Given: linear system $Ax = b$, initial approx. $x_0 \in \mathbb{C}^N$.

Construct: approximate solutions ("iterates")

$$x_n \in x_0 + \mathcal{K}_n(A, r_0),$$

where $r_0 \equiv b - Ax_0$ is the initial residual, and

$$\mathcal{K}_n \equiv \mathcal{K}_n(A, r_0) \equiv \text{span} \{ r_0, Ar_0, \ldots, A^{n-1}r_0 \}.$$

is the $n$th **Krylov subspace** generated by $A$ from $r_0$. 
\[ \mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \] implies that
\[ \mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A} \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \subseteq \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \]

so:
\[ \mathbf{x}_n = \mathbf{x}_* \iff \mathbf{r}_n = 0 \implies \mathbf{r}_0 \in \mathbf{A} \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0). \]

So, we need to approximate \( \mathbf{r}_0 \) by elements from \( \mathbf{A} \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \).

E.g., we may choose \( \mathbf{r}_n \) as the perpendicular ("Lot") from \( \mathbf{r}_0 \) to its orthogonal projection into \( \mathbf{A} \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \).

- **Conjugate Residual (CR) method** (Stiefel, 1955):
  \[ \mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A} \mathcal{K}_n, \quad \mathbf{r}_n \perp \mathbf{A} \mathcal{K}_n. \]

- Same holds for **GCR** and **GMRES**.
Some Krylov space solvers are based on other orthogonal or oblique ("schiefe") projections:

- **Conjugate Gradient (CG) method** (Hestenes/Stiefel, 1952):
  \[ r_n \in r_0 + A\mathcal{K}_n, \quad r_n \perp \mathcal{K}_n. \]

- **Biconjugate Gradient (BiCG) method** (Lanczos, 1952; Fletcher, 1976):
  \[ r_n \in r_0 + A\mathcal{K}_n, \quad r_n \perp \tilde{\mathcal{K}}_n :\equiv \mathcal{K}_n(A^*, \tilde{r}_0). \]

- **ML(s)BiCG method** (M.-C. Yeung and T. F. Chan, 1999):
  \[ r_{sj} \in r_0 + A\mathcal{K}_{sj}, \quad r_{sj} \perp \mathcal{K}_j(A^*, \tilde{R}_0) :\equiv \bigoplus_{i=1}^{s} \mathcal{K}_j(A^*, \tilde{r}_0^{(i)}). \]
Prerequisites: residual polynomials

\[ r_n \in r_0 + A\mathcal{K}_n(A, r_0) \] implies that

\[ \exists \rho_n \in \mathcal{P}_n, \, \rho_n(0) = 1 : \quad r_n = \rho_n(A)r_0. \]

- Means roughly: \( \|r_n\| \) is small if \( |\rho_n(t)| \) is small at the eigenvalues of \( A \).
Special cases:

- In CG the residual polynomials are orthogonal polynomials (OPs) w.r.t. a weight function determined by EVals of $A$ (symmetric) and by $r_0$.
- In BICG the residual polynomials are formal orthogonal polynomials (FOPs). $\leadsto$ Lanczos polynomials.
- In (Bi)Conjugate Gradient Squared (CGS),
  \[
  \rho_{CGS}^n = \left( \rho_{BICG}^n \right)^2.
  \]
- In BICGSTAB,\[
  \rho_{BICGSTAB}^n = \rho_{BICG}^n \Omega_n,
  \]
  where $\Omega_n(t) :\equiv (1 - \omega_1 t) \cdots (1 - \omega_n t)$. Here, at step $n$, $\omega_n$ is chosen to minimize the residual on a straight line.
History of IDR: references


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

Received May 16, 1997. Introduce first **ML(k)BiCG** and then **ML(k)BiCGSTAB** — versions of **BiCG** and **BiCGSTAB**, resp., with “multiple left (shadow) residuals”. Well written, but some details complicated. Astonishing numerical results.
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 ≈ 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 new view; partly different notation; partly confusing. Introduce variants of ML(\(k\))BiCG and ML(\(k\))BiCGSTAB; insufficient reference to Yeung/Chan.
M. B. van Gijzen and P. Sonneveld, *An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties*, Report 08-21, Department of Applied Mathematical Analysis, Delft University of Technology.

Explores the freedom in the choice of the “intermediate” residuals to come up with a new version of IDR(s) that is slightly more efficient and particularly ingenious — but even harder to understand in detail.
IDR(s) basics: the setting

Given: linear system $Ax = b \in \mathbb{C}^N$, initial approx. $x_0$. Let:

$r_0 :\equiv b - Ax_0$,  
$\mathcal{K}_m :\equiv \mathcal{K}_m(A, r_0) :\equiv \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$,  
$\nu$ such that $G_0 :\equiv \mathcal{K}_\nu$ invariant,  
$S \subset \mathbb{C}^N$ linear subspace of dimension $N - s$,  
for $j = 1, 2, \ldots$: choose $\omega_j \neq 0$ and let

$G_j :\equiv (I - \omega_jA)(G_{j-1} \cap S)$,

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

choose $x_n$ such that

$r_n \in G_j \cap (r_0 + A\mathcal{K}_n) \subset \mathcal{K}_{n+1}$.

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

M.H. Gutknecht  
TU Berlin  
p. 12
IDR(s) basics: the spaces $\mathcal{G}_j$ (case $s = 1$)

$$\mathcal{G}_0 = \mathbb{R}^3$$

$$\mathcal{G}_2 \cap S = \{0\}$$

$$\mathcal{G}_2 \cap S = \{0\}$$

$$\mathcal{G}_2 \cap S = \{0\}$$

$$\mathcal{G}_j := (I - \omega_j A)(\mathcal{G}_{j-1} \cap S)$$
IDR(s) basics: the first two steps (case $s = 1$)

$G_0 = \mathbb{R}^3$

$G_0 \cap S = S$

$r_2 = (I - \omega_1 A)v_1$

$r_0$

$r_1$

$v_1$

$v_2$

$r_3$

$I - \omega_1 A$
IDR(s) basics: all the residuals (case $s = 1$)

$$\mathcal{G}_0 = \mathbb{R}^3$$

$$\mathcal{G}_1$$

$$\mathcal{G}_2 \cap \mathcal{S} = \{0\} = \{v_5\}$$

$$r_2 = (I - \omega_1 A)v_1$$

$$r_0$$

$$r_1$$
IDR(s) basics: IDR theorem

Recall: \( G_j \equiv (I - \omega_j A)(G_{j-1} \cap S) \), \( r_n \in G_j \cap (r_0 + A \mathcal{K}_n) \).

Genericness assumption: \( S \cap G_0 \) contains no eigenvector of \( A \).

**Theorem (IDR Theorem (Wes/Son80, Son/vGi07))**

\[ G_j \subsetneq G_{j-1} \text{ unless } G_{j-1} = \{0\}. \]

Consequently: \( G_j = \{0\} \) for some \( j \leq N \).

Can expect: \( r_n = 0 \) once \( j = N \), that is, \( n = (s + 1)N \).

But typically: \( r_n = 0 \) 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”:

\[ r_n \in \mathcal{L}_n^+ \cap (r_0 + A\mathcal{K}_n(A, r_0)) , \]

where, e.g.,

\[
\begin{align*}
\mathcal{L}_n &= \mathcal{K}_n(A, r_0) \quad &\text{(CG)}, \\
\mathcal{L}_n &= A\mathcal{K}_n(A, r_0) \quad &\text{(CR, GCR, GMRES)}, \\
\mathcal{L}_n &= \tilde{\mathcal{K}}_n := \mathcal{K}_n(A^*, \tilde{r}_0) \quad &\text{(BiCG)}.
\end{align*}
\]

IDR: \( \mathcal{G}_j \) is not an orthogonal complement of a Krylov subspace.

However, due to form of the recursion for \( \{\mathcal{G}_j\} \), \( \mathcal{G}_j \) turns out to be the image of an orthogonal complement of a Krylov subspace.
IDR(s) basics: recursions for \( \{ r_n \} \)

Recall: \[ G_j \equiv (I - \omega_j A)(G_{j-1} \cap S). \]

Wanted: \[ r_{n+1} \in G_j \cap (r_0 + AK_{n+1}). \]

\[ \implies r_{n+1} := (I - \omega_j A) v_n, \quad v_n \in G_{j-1} \cap S \cap (r_0 + AK_n), \]

\[ \implies v_n := r_n - \sum_{i=1}^{\nu(n)} \gamma_i^{(n)} \Delta r_{n-i} = r_n - \Delta R_n c_n, \quad (1) \]

where \[ s \leq \nu(n) \leq n - n_{j-1}, \quad (\implies \Delta r_{n-i} \in G_{j-1}) \]

\[ \Delta r_n \equiv r_{n+1} - r_n, \]

\[ \Delta R_n \equiv \begin{bmatrix} \Delta r_{n-1} & \cdots & \Delta r_{n-\nu(n)} \end{bmatrix}, \]

\[ c_n \equiv \begin{bmatrix} \gamma_1^{(n)} & \cdots & \gamma_{\nu(n)}^{(n)} \end{bmatrix}. \]
Recall (1):

\[ v_n := r_n - \sum_{i=1}^{\nu(n)} \gamma_i^{(n)} \Delta r_{n-i} = r_n - \Delta R_n c_n \in G_{j-1} \cap S. \]

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

\[ v_n \in S \iff v_n \perp S^\perp = R(P) \iff P^*v_n = 0. \]

To achieve this, the term \( \Delta R_n c_n \) in (1) must be the oblique projection of \( r_n \) into \( R(\Delta R_n) \) along \( S \).

In order that this projection is uniquely defined, we need \( P^* \Delta R_n \) to be nonsingular. Then

\[ v_n := r_n - \Delta R_n \left( P^* \Delta R_n \right)^{-1} P^*r_n = r_n - \Delta R_n c_n \]  

\[ \equiv: c_n \]

We need \( \nu(n) = s \) to make \( P^* \Delta R_n \) a square matrix.
IDR(s) basics: choice of $\omega_j$

Recall: \[ r_{n+1} = (I - \omega_j A) v_n \] (3)

Here, $\omega_j$ is fixed for $n + 1 = n_j, \ldots, n_{j+1} - 1$.

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

\[ \omega_j \equiv \frac{\langle Av_n, v_n \rangle}{\|Av_n\|^2}. \]
IDR(s) basics: recursions for \( \{x_n\} \)

**Note 1:** \( v_n \in r_0 + AK_n \implies \exists x'_n \text{ s.t. } v_n = b - Ax'_n \),
i.e., \( v_n \) is the residual of an “intermediate” iterate \( x'_n \in x_0 + K_n \).

**Note 2:** \( \Delta r_n = -A \Delta x_n, \quad \Delta R_n = -A \Delta X_n \),

Hence:

\[
\begin{align*}
v_n &:= r_n - \Delta R_n c_n \implies x'_n := x_n - \Delta X_n c_n, \\
r_{n+1} &:= (I - \omega_j A) v_n \implies x_{n+1} := \omega_j v_n + x'_n.
\end{align*}
\]

There are several ways to rearrange these four recursions and
to combine them with the iterate-residual relationships; see
[Sle/Son/vGi08].
The two formulas

\[ r_{n+1} := (I - \omega_j A) v_n, \quad v_n := r_n - \Delta R_n c_n \]

can be combined into

\[ r_{n+1} := r_n - \Delta R_n c_n - \omega_j A v_n = \Delta r_n \]  \hspace{1cm} (6)

Along with it:

\[ x_{n+1} := x_n - \Delta X_n c_n + \omega_j v_n = \Delta x_n \]  \hspace{1cm} (7)

We may also combine the second formula and \( \Delta r_n = -A \Delta x_n \).

This is the choice in the “prototype algorithm” of [Son/vGi07].

So, there are many ways to implement IDR(s) — and more to come!
IDR(s) basics: characterization by orthogonality

**Theorem (Son/vGi07, Sle/Son/vGi08)**

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

$$G_j = \left\{ \Omega_j(A)w \mid w \perp \underbrace{\mathcal{K}_j(A^*, P)}_{= \tilde{\mathcal{K}}_j} \right\} = \Omega_j(A) \underbrace{[\mathcal{K}_j(A^*, P)]^\perp}_{= \tilde{\mathcal{K}}_j^\perp}.$$

**Note:** $\tilde{\mathcal{K}}_j$ 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 \( r_n \in G_j \cap (r_0 + AK_n) \), and since the residual polynomials must have full degree, we have for \( n = n_j, \ldots, n_{j+1} - 1 \):

\[
 r_n = \Omega_j(A)w_n, \quad w_n \in (r_0 + AK_{n-j}) \cap \tilde{K}_j^\perp, \quad w_n \notin K_{n-j}.
\]

(8)

Generically, \( n_{j+1} - n_j = s + 1 \), so, for fixed \( j \), we hope for \( s + 1 \) linearly independent vectors \( w_n \) with \( n_j \leq n < n_{j+1} \).

Generically, for \( n = n_j = j(s + 1) \), where \( w_n \in (r_0 + AK_{js}) \) and \( w_n \perp \tilde{K}_j \) with \( \dim \tilde{K}_j = js \), there is a unique \( w_n \) satisf. (8).

But the \( s \) other vectors \( w_n \) are not uniquely determined by (8).

Cost of computing \( r_{n_j} \in r_0 + AK_{n_j} \): \( n_j + 1 \) MVs with \( A \).
\( s = 1: \text{IDR}(1) \sim \text{BiCGSTAB} \)

Every other set of vectors \((w_n, r_n, v_{n-1}, x_n, \ldots)\) is uniquely determined — up to the choice of the parameters \(\omega_j\).

Normally, the latter are chosen as in BiCGSTAB, and thus

\[
\begin{align*}
    r_{2j} &= r^\text{BiCGSTAB}_j, \\
    x_{2j} &= x^\text{BiCGSTAB}_j, \\
    w_{2j} &= r^\text{BICG}_j,
\end{align*}
\]

where \(r^\text{BICG}_j\) is the \(j\)th residual of BiCG: \(r^\text{BICG}_j = \rho_j(A)r_0\).

Recursions (4) and (5), with \(\gamma_n \equiv \gamma_1^{(n)} = \langle P, r_n \rangle / \langle P, \Delta r_{n-1} \rangle \):

\[
\begin{align*}
    v_n &:= (1 - \gamma_n)r_n + \gamma_n r_{n-1}, \\
    r_{n+1} &:= (I - \omega_j A)v_n, \\
    x'_n &:= (1 - \gamma_n)x_n + \gamma_n x_{n-1}, \\
    x_{n+1} &:= x'_n + \omega_j v_n.
\end{align*}
\]

(9)
s = 1: polynomial recursions

\[ r_n = \Omega_j(A)w_n = \begin{cases} 
\Omega_j(A)\rho_j(A)r_0 & \text{if } n = 2j, \\
\Omega_j(A)\hat{\rho}_{j+1}(A)r_0 & \text{if } n = 2j + 1,
\end{cases} \]

\[ v_n = \Omega_{j-1}(A)w_{n+1} = \begin{cases} 
\Omega_{j-1}(A)\rho_j(A)r_0 & \text{if } n = 2j - 1, \\
\Omega_{j-1}(A)\hat{\rho}_{j+1}(A)r_0 & \text{if } n = 2j,
\end{cases} \]

Inserting these formulas into \( v_n = (1 - \gamma_n)r_n + \gamma_n r_{n-1} \) we get, after a short calculation, for \( n = 2j \) and \( n = 2j + 1 \), respectively,

\[
\hat{\rho}_{j+1}(t) := (1 - \gamma_{2j})(1 - \omega_j t) \rho_j(t) + \gamma_{2j} \hat{\rho}_j(t), \\
\rho_{j+1}(t) := (1 - \gamma_{2j+1}) \hat{\rho}_{j+1}(t) + \gamma_{2j+1} \rho_j(t). \tag{10}
\]

Recall: \( w_{2j} = \rho_j(A)r_0 \perp \tilde{\mathcal{K}}_j \) and \( w_{2j+1} = \hat{\rho}_j(A)r_0 \perp \tilde{\mathcal{K}}_j. \)
BICG uses in its standard version coupled two-term recursions:

\[
\begin{align*}
\mathbf{r}_{j+1} & := \mathbf{r}_j - \alpha_j A \mathbf{v}_j, \\
\mathbf{v}_{j+1} & := \mathbf{r}_{j+1} + \beta_j \mathbf{v}_j.
\end{align*}
\]

The corresp. recursions for \( \rho_j \) and \( \sigma_j \) are

\[
\begin{align*}
\rho_{j+1}(t) & := \rho_j(t) - \alpha_j t \sigma_j(t), \\
\sigma_{j+1}(t) & := \rho_{j+1}(t) + \beta_j \sigma_j(t).
\end{align*}
\]

In contrast, in IDR(1), by (10),

\[
\begin{align*}
\hat{\rho}_{j+1}(t) & := (1 - \gamma_{2j}) (1 - \omega_j t) \rho_j(t) + \gamma_{2j} \hat{\rho}_j(t), \\
\rho_{j+1}(t) & := (1 - \gamma_{2j+1}) \hat{\rho}_{j+1}(t) + \gamma_{2j+1} \rho_j(t).
\end{align*}
\]
Comparing the recursions for \((\rho_j, \sigma_j)\) 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,

\[
\textcolor{red}{r_{2j+1} = r_{2j} - \frac{\alpha_j}{(1 - \gamma_{2j+1})} A \underbrace{\Omega_j(A) v_j}_{\text{BICGSTAB}}}. \quad \equiv: \text{S}_j
\]

This formula expresses the odd indexed IDR(1) residuals in terms of quantities from BICGSTAB and the coefficient \(\gamma_{2j+1}\).
\[
\hat{\rho}_{j+1}(t) = \rho_j(t) - \frac{\alpha_j}{1 - \gamma_{2j+1}} t \sigma_j(t),
\]

\[
\text{BICG/BICGSTAB: } \quad \rho_{j+1}(t) := \rho_j(t) - \alpha_j t \sigma_j(t),
\]

\[
\text{IDR(1): } \quad \rho_{j+1}(t) := (1 - \gamma_{2j+1}) \hat{\rho}_{j+1}(t) + \gamma_{2j+1} \rho_j(t).
\]
\( s = 1 \): Comments and conclusions

- 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 BiOR\$ES 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 \( \omega_j \).
s > 1: IDR(s) \sim ML(s)BICGSTAB

- Relation IDR(1) \sim BICGSTAB \iff BICG is matched by relation IDR(s) \sim ML(s)BICGSTAB \iff ML(s)BICG.

- If the parameters \( \omega_j \) were chosen the same in IDR(s) and ML(s)BICGSTAB every \((s + 1)\)th iterate were the same in both methods — but normally the parameters are not chosen the same.

- 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 “degree” of $w_n$, which is what is determined by orthogonality) is due to Yeung and Chan. Sonneveld and van Gijzen rediscovered it 10 years later independently.

• Yeung/Chan paper is well written, but derivation of the recursions is complicated due to use of the “wrong” basis for $\tilde{K}_j$ and complex manipulations for reducing cost. Amazingly, their Matlab program

http://www.uwyo.edu/mathmyeung/p12/mlbicgstab.txt

is only 187 lines (incl. 30 lines of comments).

• ML(s)BICGSTAB requires a few more inner products and vector updates than IDR(s) and may be less stable.
Conclusions

- IDR(1) is as good as BiCGSTAB.
- IDR(s) is as good as ML(s)BiCGSTAB.
- The IDR(s) recurrence coefficients are simpler to compute than those of ML(s)BiCGSTAB.
- Typically, IDR(s) and ML(s)BiCGSTAB outperform other methods for a nonsymmetric problem.
- What is missing is the IDR-like generalization of BiCGStab2 to cover the case where A is real, but has (strongly) non-real eigenvalues.
\[ s = 1: \] How did the original IDR differ from IDR(1)?

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

\[
\begin{align*}
\mathbf{v}_n & := \mathbf{r}_n - \gamma_n (\mathbf{r}_n - \mathbf{r}_{n-1}), \\
\mathbf{r}_{n+1} & := (I - \omega_j \mathbf{A}) \mathbf{v}_n,
\end{align*}
\]

\[
\begin{align*}
\mathbf{x}'_n & := \mathbf{x}_n - \gamma_n (\mathbf{x}_n - \mathbf{x}_{n-1}), \\
\mathbf{x}_{n+1} & := \mathbf{x}'_n + \omega_j \mathbf{v}_n,
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{v}_n & := \mathbf{r}_n - \gamma'_n (\mathbf{r}_{n-1} - \mathbf{r}_{n-2}), \\
\mathbf{r}_{n+1} & := (I - \omega_j \mathbf{A}) \mathbf{v}_n,
\end{align*}
\]

\[
\begin{align*}
\mathbf{x}'_n & := \mathbf{x}_n - \gamma'_n (\mathbf{x}_{n-1} - \mathbf{x}_{n-2}), \\
\mathbf{x}_{n+1} & := \mathbf{x}'_n + \omega_j \mathbf{v}_n.
\end{align*}
\]

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

The IDR theorem still applies, and still \( \mathbf{x}_{2j} = \mathbf{x}_j^{\text{BiCGSTAB}} \).
Another variant: IDR(s)BiO

Recall (1): in IDR(s) of [Son/vGi07]

\[ v_n := r_n - \sum_{i=1}^{\nu(n)} \gamma_i^{(n)} \Delta r_{n-i} = r_n - \Delta R_n c_n \in G_{j-1} \cap S. \]

where \( s \leq \nu(n) \leq n - n_{j-1} \), and normally \( \nu(n) = s \).

In [vGiSon08] the authors propose instead a variant where the index \( n - i \) in (1) ranges from \( n_{j-1} \) to \( n_j - 2 \).
At the same time they replace in (1) the subspace basis

\[ \{ \Delta r_{n-i} \}_{i=1}^{s} \subset G_{j-1} \cap A K_n \]

by a basis of another \( s \)-dim. subspace depending on \( j - 1 \) only:

\[ \{ g_{n_{j-1}+i-1} \}_{i=1}^{s} \subset G_{j-1} \cap A K_{n_{j-1}} \]
This means that now (after inverting the summation index)

\[ v_n := r_n - \sum_{i=1}^{s} \gamma_i^{(n)} g_{n_{j-1}+i-1} = r_n - G_{j-1} c_n \in G_{j-1} \cap S, \]

where \( n = n_j + k \) (\( k = 0, \ldots, s \)),

\[ G_{j-1} := \begin{bmatrix} g_{n_{j-1}} & \cdots & g_{n_j} \end{bmatrix}, \quad c_n := \begin{bmatrix} \gamma_1^{(n)} & \cdots & \gamma_s^{(n)} \end{bmatrix}. \]

\( c_n \) is determined by the condition \( v_n \perp \mathcal{R}(P) \). So, as in (2),

\[ c_n := (P^* G_{j-1})^{-1} P^* r_n, \quad v_n := r_n - G_{j-1} c_n. \]

Also the matrix \( M_{j-1} \) of the linear system for \( c_n \) only depends on \( j - 1 \), not on \( n \) (i.e., \( k \)).
Using a “one-sided biorthogonalization” process [vGiSon08] enforce that

\[ M_{j-1} \equiv P^* G_{j-1} \quad \text{and} \quad F_j \equiv P^* \left[ r_{nj} \ldots r_{nj+s-1} \right] \]

are lower triangular, so

\[ C_j \equiv \begin{bmatrix} c_{nj} & \cdots & c_{nj+s-1} \end{bmatrix} \equiv M_{j-1}^{-1} F_j \]

is lower triangular too. Consequently:

- The sum in (11) starts at \( k + 1 \) only.
- In the \( k \)-loop of the code, no longer used parts of \( G_{j-1} \) and \( M_{j-1} \) can be overwritten with new data for \( G_j \) and \( M_j \).
- This applies also to \( U_{j-1} \equiv A^{-1} G_{j-1} \) needed to update \( x_n \).
- \( \rightsquigarrow \) Reduction in computing time and memory usage.
Conclusions

- The new variant, called IDR(s)BIO here, is particularly ingenious.
- Computing time and memory usage are reduced a bit.
- Ansatz for $\mathbf{v}_n$ generalizes the original IDR of 1980, but IDR(1)BIO $\neq$ IDR$_{1980}$.
- It remains unclear why the latter was said to be less stable than BICGSTAB.
Merry Christmas