On the genealogy of the IDR family

Jens-Peter M. Zemke
zemke@tu-harburg.de
(partially joint work with Martin Gutknecht)

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

TU Bergakademie Freiberg
December 17th, 2009

Outline

The origin of transpose-free methods . . .

“ Instead of iterating with $\mathbf{A}$ and $\mathbf{A}^T$ n times, we can also iterate with $\mathbf{A}$ alone $2n$ times. [..] The transposed matrix is not used here at all. E. C. Bouwer of the Douglas Aircraft Co. points out to the author that from the machine viewpoint a uniform iteration scheme of $2n$ iterations is preferable to a divided scheme of $n + n$ iterations. [..] In case of a symmetric matrix it is evident that after n iterations the basic scalars should be formed, instead of continuing with n more iterations. ”

The origin of transpose-free methods . . .

> *“ Instead of iterating with $\mathbf{A}$ and $\mathbf{A}^T$ n times, we can also iterate with $\mathbf{A}$ alone $2n$ times. [..] The transposed matrix is not used here at all. E. C. Bouwer of the Douglas Aircraft Co. points out to the author that from the machine viewpoint a uniform iteration scheme of $2n$ iterations is preferable to a divided scheme of $n + n$ iterations. [..] In case of a symmetric matrix it is evident that after n iterations the basic scalars should be formed, instead of continuing with n more iterations. ”*

— Cornelius Lanczos, footnote on page 263 in (Lanczos, 1950), referring to his progressive algorithm based on Hankel determinants.

The origin of IDR: poor man's secant method

In 1976 Peter Sonneveld (Sonneveld, 2006; Sonneveld, 2008) prepared notes for a course on Numerical Analysis at TU Delft. The secant method was part of the course. He generalized it to a multidimensional secant method . . .

The origin of IDR: poor man's secant method

In 1976 Peter Sonneveld (Sonneveld, 2006; Sonneveld, 2008) prepared notes for a course on Numerical Analysis at TU Delft. The secant method was part of the course. He generalized it to a multidimensional secant method ...

Let $f(\mathbf{x}) := \mathbf{b} - \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$ are given. Then

$$\mathbf{F}_k := f(\mathbf{X}_k) := \begin{pmatrix} f(\mathbf{x}_0) & \cdots & f(\mathbf{x}_n) \end{pmatrix} \in \mathbb{C}^{n\times(n+1)}$$

is rank deficient.

The origin of IDR: poor man's secant method

In 1976 Peter Sonneveld (Sonneveld, 2006; Sonneveld, 2008) prepared notes for a course on Numerical Analysis at TU Delft. The secant method was part of the course. He generalized it to a multidimensional secant method ...

Let $f(\mathbf{x}) := \mathbf{b} - \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$ are given. Then

$$\mathbf{F}_k := f(\mathbf{X}_k) := \begin{pmatrix} f(\mathbf{x}_0) & \cdots & f(\mathbf{x}_n) \end{pmatrix} \in \mathbb{C}^{n\times(n+1)}$$

is rank deficient. For every solution $\hat{\mathbf{x}}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$,

$$\mathbf{F}_k = \mathbf{A}(\hat{\mathbf{x}}\mathbf{e}^T - \mathbf{X}_k), \quad \text{where} \quad \mathbf{e} := \texttt{ones}\,(n+1, 1)\,.$$

The origin of IDR: poor man's secant method

In 1976 Peter Sonneveld (Sonneveld, 2006; Sonneveld, 2008) prepared notes for a course on Numerical Analysis at TU Delft. The secant method was part of the course. He generalized it to a multidimensional secant method ...

Let $f(\mathbf{x}) := \mathbf{b} - \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$ are given. Then

$$\mathbf{F}_k := f(\mathbf{X}_k) := \begin{pmatrix} f(\mathbf{x}_0) & \cdots & f(\mathbf{x}_n) \end{pmatrix} \in \mathbb{C}^{n\times(n+1)}$$

is rank deficient. For every solution $\hat{\mathbf{x}}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$,

$$\mathbf{F}_k = \mathbf{A}(\hat{\mathbf{x}}\mathbf{e}^T - \mathbf{X}_k), \quad \text{where} \quad \mathbf{e} := \texttt{ones}\,(n+1, 1)\,.$$

Thus, for $\mathbf{F}_k\mathbf{c}_k = \mathbf{o}_n$ and $\mathbf{e}^T\mathbf{c}_k \neq 0$,

$$\mathbf{b}\mathbf{e}^T\mathbf{c}_k = \mathbf{A}\hat{\mathbf{x}}\mathbf{e}^T\mathbf{c}_k = \mathbf{A}\mathbf{X}_k\mathbf{c}_k \quad \Rightarrow \quad \hat{\mathbf{x}} = \frac{\mathbf{X}_k\mathbf{c}_k}{\mathbf{e}^T\mathbf{c}_k}.$$

The origin of IDR: poor man's secant method

For genuine non-linear (smooth) functions f, we replace $\mathbf{A}$ by the Jacobi matrix and $\mathbf{b}$ by the function evaluation at an initial guess.

The origin of IDR: poor man's secant method

For genuine non-linear (smooth) functions f, we replace $\mathbf{A}$ by the Jacobi matrix and $\mathbf{b}$ by the function evaluation at an initial guess.

Then the process described gives a linearization and updates iterates to give better approximations.

The origin of IDR: poor man's secant method

For genuine non-linear (smooth) functions f, we replace $\mathbf{A}$ by the Jacobi matrix and $\mathbf{b}$ by the function evaluation at an initial guess.

Then the process described gives a linearization and updates iterates to give better approximations.

Updating all columns of $\mathbf{F}_k$ is ill-conditioned, as all columns converge to the same vector $\mathbf{f} := f(\hat{\mathbf{x}})$. Sonneveld updated only the last two columns:

$$\mathbf{F}_k := \left(\mathbf{F}_{n-1}^{\text{const}} \quad \mathbf{f}_{k-1} \quad \mathbf{f}_k\right).$$

The origin of IDR: poor man's secant method

For genuine non-linear (smooth) functions f, we replace $\mathbf{A}$ by the Jacobi matrix and $\mathbf{b}$ by the function evaluation at an initial guess.

Then the process described gives a linearization and updates iterates to give better approximations.

Updating all columns of $\mathbf{F}_k$ is ill-conditioned, as all columns converge to the same vector $\mathbf{f} := f(\hat{\mathbf{x}})$. Sonneveld updated only the last two columns:

$$\mathbf{F}_k := \begin{pmatrix} \mathbf{F}_{n-1}^{\text{const}} & \mathbf{f}_{k-1} & \mathbf{f}_k \end{pmatrix}.$$

Therefore, with $\mathbf{A} := \nabla f(\hat{\mathbf{x}})$,

$$\mathbf{F}_k = \begin{pmatrix} \mathbf{A}(\hat{\mathbf{x}}\mathbf{e}^{\mathsf{T}} - \mathbf{X}_{n-1}) + \mathbf{E}_{n-1} & \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}_{k-1}) + \mathbf{d}_{k-1} & \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}_k) + \mathbf{d}_k \end{pmatrix},$$

where $\mathbf{E}_{n-1}$ is a constant matrix and the vectors $\mathbf{d}_k$ converge to zero.

The origin of IDR: poor man's secant method

Sonneveld used the example $\mathbf{Ax} = \mathbf{o}_n$ and mimicked the non-linearity by the presence of a constant matrix $\mathbf{E}_{n-1}$ in the process.

The origin of IDR: poor man's secant method

Sonneveld used the example $\mathbf{Ax} = \mathbf{o}_n$ and mimicked the non-linearity by the presence of a constant matrix $\mathbf{E}_{n-1}$ in the process.

If used for a matrix of dimension $n \in \mathbb{N}$, the process gave (an approximation to) the value zero in step $2n$.

The origin of IDR: poor man's secant method

Sonneveld used the example $\mathbf{Ax} = \mathbf{o}_n$ and mimicked the non-linearity by the presence of a constant matrix $\mathbf{E}_{n-1}$ in the process.

If used for a matrix of dimension $n \in \mathbb{N}$, the process gave (an approximation to) the value zero in step $2n$. In the following example I used Maple to exclude finite precision and a badly conditioned matrix $\mathbf{A}$ of size 5.

$$\begin{aligned}
\|\mathbf{r}_0\|_2 &= 7.416198487, & \|\mathbf{r}_1\|_2 &= 31.28897569,\\
\|\mathbf{r}_2\|_2 &= 3.838120391, & \|\mathbf{r}_3\|_2 &= 3.944190988,\\
\|\mathbf{r}_4\|_2 &= 1.035754508, & \|\mathbf{r}_5\|_2 &= 1.035728492,\\
\|\mathbf{r}_6\|_2 &= 0.983756197, & \|\mathbf{r}_7\|_2 &= 0.983648677,\\
\|\mathbf{r}_8\|_2 &= 0.520741201, & \|\mathbf{r}_9\|_2 &= 0.520740892,\\
\|\mathbf{r}_{10}\|_2 &= \|\mathbf{r}_{2n}\|_2 = 0.
\end{aligned}$$

The origin of IDR: poor man's secant method

Sonneveld used the example $\mathbf{Ax} = \mathbf{o}_n$ and mimicked the non-linearity by the presence of a constant matrix $\mathbf{E}_{n-1}$ in the process.

If used for a matrix of dimension $n \in \mathbb{N}$, the process gave (an approximation to) the value zero in step $2n$. In the following example I used Maple to exclude finite precision and a badly conditioned matrix $\mathbf{A}$ of size 5.

$$\begin{aligned}
\|\mathbf{r}_0\|_2 &= 7.416198487, & \|\mathbf{r}_1\|_2 &= 31.28897569,\\
\|\mathbf{r}_2\|_2 &= 3.838120391, & \|\mathbf{r}_3\|_2 &= 3.944190988,\\
\|\mathbf{r}_4\|_2 &= 1.035754508, & \|\mathbf{r}_5\|_2 &= 1.035728492,\\
\|\mathbf{r}_6\|_2 &= 0.983756197, & \|\mathbf{r}_7\|_2 &= 0.983648677,\\
\|\mathbf{r}_8\|_2 &= 0.520741201, & \|\mathbf{r}_9\|_2 &= 0.520740892,\\
\|\mathbf{r}_{10}\|_2 &= \|\mathbf{r}_{2n}\|_2 = 0.
\end{aligned}$$

He analyzed this startling behavior: the first IDR method was born.

The origin of IDR: poor man's secant method

To analyze, he realized that $\mathbf{c}_k$ is of interest up to a scalar non-zero factor. He considered the case that $c_{n-1} + c_n = 1$, i.e., that the sum of the last two elements is scaled to be one. He sets $c_{n-1} := \gamma_k$ and thus $c_n = 1 - \gamma_k$.

The origin of IDR: poor man's secant method

To analyze, he realized that $\mathbf{c}_k$ is of interest up to a scalar non-zero factor. He considered the case that $c_{n-1} + c_n = 1$, i.e., that the sum of the last two elements is scaled to be one. He sets $c_{n-1} := \gamma_k$ and thus $c_n = 1 - \gamma_k$.

Now, for $\mathbf{c}_{n-1}^{(k)}$, we have to solve the overdetermined consistent linear system

$$\mathbf{F}_{n-1}\mathbf{c}_{n-1}^{(k)} = -\mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}).$$

The origin of IDR: poor man's secant method

To analyze, he realized that $\mathbf{c}_k$ is of interest up to a scalar non-zero factor. He considered the case that $c_{n-1} + c_n = 1$, i.e., that the sum of the last two elements is scaled to be one. He sets $c_{n-1} := \gamma_k$ and thus $c_n = 1 - \gamma_k$.

Now, for $\mathbf{c}_{n-1}^{(k)}$, we have to solve the overdetermined consistent linear system

$$\mathbf{F}_{n-1}\mathbf{c}_{n-1}^{(k)} = -\mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}).$$

As $\mathbf{F}_{n-1} \in \mathbb{C}^{n\times(n-1)}$, there exists a non-zero vector $\mathbf{p} \in \mathbb{C}^n$ in the left null space of $\mathbf{F}_{n-1}$. With this vector,

$$0 = \mathbf{p}^{\mathsf{H}}\mathbf{F}_{n-1}\mathbf{c}_{n-1}^{(k)} = \mathbf{p}^{\mathsf{H}}(-\mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})),$$

The origin of IDR: poor man's secant method

To analyze, he realized that $\mathbf{c}_k$ is of interest up to a scalar non-zero factor. He considered the case that $c_{n-1} + c_n = 1$, i.e., that the sum of the last two elements is scaled to be one. He sets $c_{n-1} := \gamma_k$ and thus $c_n = 1 - \gamma_k$.

Now, for $\mathbf{c}_{n-1}^{(k)}$, we have to solve the overdetermined consistent linear system

$$\mathbf{F}_{n-1}\mathbf{c}_{n-1}^{(k)} = -\mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}).$$

As $\mathbf{F}_{n-1} \in \mathbb{C}^{n\times(n-1)}$, there exists a non-zero vector $\mathbf{p} \in \mathbb{C}^n$ in the left null space of $\mathbf{F}_{n-1}$. With this vector,

$$0 = \mathbf{p}^{\mathsf{H}}\mathbf{F}_{n-1}\mathbf{c}_{n-1}^{(k)} = \mathbf{p}^{\mathsf{H}}(-\mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})),$$

i.e., γ_k is uniquely (in case of no breakdown) determined by

$$\gamma_k := -\frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_k - \mathbf{f}_{k-1})}.$$

The origin of IDR: poor man's secant method

The vector $\mathbf{c}_{n-1}^{(k)}$ is then (because of the consistency of the given overdetermined system) given by

$$\mathbf{c}_{n-1}^{(k)} := -\mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})).$$

The origin of IDR: poor man's secant method

The vector $\mathbf{c}_{n-1}^{(k)}$ is then (because of the consistency of the given overdetermined system) given by

$$\mathbf{c}_{n-1}^{(k)} := -\mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})).$$

The new residual $\mathbf{f}_{k+1} = \mathbf{o}_n - \mathbf{A}\mathbf{x}_{k+1}$ satisfies

$$\begin{aligned}
\mathbf{f}_{k+1} &= -\frac{\mathbf{A}(\mathbf{X}_{n-1}\mathbf{c}_{n-1}^{(k)} + \mathbf{x}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}))}{\mathbf{e}^{\mathsf{T}}\mathbf{c}_{n-1}^{(k)} + 1} \\
&= \frac{(\mathbf{E}_{n-1} - \mathbf{F}_{n-1})\mathbf{c}_{n-1}^{(k)} - \mathbf{f}_k - \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})}{\mathbf{e}^{\mathsf{T}}\mathbf{c}_{n-1}^{(k)} + 1} = \frac{\mathbf{E}_{n-1}\mathbf{c}_{n-1}^{(k)}}{\mathbf{e}^{\mathsf{T}}\mathbf{c}_{n-1}^{(k)} + 1} \\
&= \frac{\mathbf{E}_{n-1}\mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))}{\mathbf{e}^{\mathsf{T}}\mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})) - 1} = \rho_k\mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))
\end{aligned}$$

The origin of IDR: poor man's secant method

As the method usually converges, the vector $\mathbf{c}_k$ in the null space of $\mathbf{F}_n^{(k)}$ will not change much, thus the scaling will not change much, thus for $k \gg 1$

$$\rho_k := \frac{1}{\mathbf{e}^T \mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})) - 1} = \frac{1}{\mathbf{e}^T \mathbf{c}_k} \approx \text{const} \neq 0.$$

The origin of IDR: poor man's secant method

As the method usually converges, the vector $\mathbf{c}_k$ in the null space of $\mathbf{F}_n^{(k)}$ will not change much, thus the scaling will not change much, thus for $k \gg 1$

$$\rho_k := \frac{1}{\mathbf{e}^T \mathbf{F}_{n-1}^{\dagger}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})) - 1} = \frac{1}{\mathbf{e}^T \mathbf{c}_k} \approx \text{const} \neq 0.$$

The finite termination property of the resulting three-term recurrence

$$\mathbf{f}_{k+1} = \rho_k \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))$$

can thus not depend on the scaling, but only on the way γ_k and thus $\mathbf{f}_k$ is computed. For this reason, Sonneveld considered the case $\rho_k = 1$ for all k.

The origin of IDR: poor man's secant method

As the method usually converges, the vector $\mathbf{c}_k$ in the null space of $\mathbf{F}_n^{(k)}$ will not change much, thus the scaling will not change much, thus for $k \gg 1$

$$\rho_k := \frac{1}{\mathbf{e}^\mathsf{T}\mathbf{F}_{n-1}^\dagger(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})) - 1} = \frac{1}{\mathbf{e}^\mathsf{T}\mathbf{c}_k} \approx \text{const} \neq 0.$$

The finite termination property of the resulting three-term recurrence

$$\mathbf{f}_{k+1} = \rho_k \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))$$

can thus not depend on the scaling, but only on the way γ_k and thus $\mathbf{f}_k$ is computed. For this reason, Sonneveld considered the case $\rho_k = 1$ for all k.

Do we need the information that the matrix $\mathbf{B} \in \mathbb{C}^{n\times n}$ is defined by

$$\mathbf{B} := \mathbf{E}_{n-1}\mathbf{F}_{n-1}^\dagger ?$$

The origin of IDR: poor man's secant method

The constant matrix $\mathbf{E}_{n-1}$ was arbitrarily chosen. Thus, we could represent every at most rank $n-1$ matrix with the same kernel as $\mathbf{F}_{n-1}^{\dagger}$.

The origin of IDR: poor man's secant method

The constant matrix $\mathbf{E}_{n-1}$ was arbitrarily chosen. Thus, we could represent every at most rank $n-1$ matrix with the same kernel as $\mathbf{F}_{n-1}^{\dagger}$.

The right kernel of $\mathbf{F}_{n-1}^{\dagger}$ is the left kernel of $\mathbf{F}_{n-1}$, i.e., it is spanned by the vector $\mathbf{p}$ used in the computation of γ_k,

$$\gamma_k := -\frac{\mathbf{p}^{\mathsf{H}}\,\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_k - \mathbf{f}_{k-1})}.$$

The origin of IDR: poor man's secant method

The constant matrix $\mathbf{E}_{n-1}$ was arbitrarily chosen. Thus, we could represent every at most rank $n-1$ matrix with the same kernel as $\mathbf{F}_{n-1}^{\dagger}$.

The right kernel of $\mathbf{F}_{n-1}^{\dagger}$ is the left kernel of $\mathbf{F}_{n-1}$, i.e., it is spanned by the vector $\mathbf{p}$ used in the computation of γ_k,

$$\gamma_k := -\frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_k - \mathbf{f}_{k-1})}.$$

The simplified (i.e., scaled) three-term recurrence

$$\mathbf{f}_{k+1} = \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))$$

is "immune" to changes in $\mathbf{B}$ in direction of $\mathbf{p}$, as the γ_k are chosen to construct vectors orthogonal to $\mathbf{p}$.

The origin of IDR: poor man's secant method

The constant matrix $\mathbf{E}_{n-1}$ was arbitrarily chosen. Thus, we could represent every at most rank $n-1$ matrix with the same kernel as $\mathbf{F}_{n-1}^{\dagger}$.

The right kernel of $\mathbf{F}_{n-1}^{\dagger}$ is the left kernel of $\mathbf{F}_{n-1}$, i.e., it is spanned by the vector $\mathbf{p}$ used in the computation of γ_k,

$$\gamma_k := -\frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_k - \mathbf{f}_{k-1})}.$$

The simplified (i.e., scaled) three-term recurrence

$$\mathbf{f}_{k+1} = \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1}))$$

is "immune" to changes in $\mathbf{B}$ in direction of $\mathbf{p}$, as the γ_k are chosen to construct vectors orthogonal to $\mathbf{p}$.

We could use any $\mathbf{B} \in \mathbb{C}^{n\times n}$ without spoiling the finite termination property!

The origin of IDR: primitive IDR

Sonneveld first made experiments and then gave a rigorous proof. It is easy to see that apart from the first two (arbitrarily chosen) residuals the constructed residuals are in the $\mathbf{B}$ image of the space $\mathcal{S} := \mathbf{p}^{\perp}$.

The origin of IDR: primitive IDR

Sonneveld first made experiments and then gave a rigorous proof. It is easy to see that apart from the first two (arbitrarily chosen) residuals the constructed residuals are in the $\mathbf{B}$ image of the space $\mathcal{S} := \mathbf{p}^{\perp}$.

The same argument proves that in general (observe that the first two residuals $\mathbf{f}_0, \mathbf{f}_1$ are usually not in $\mathcal{S}$) for $k \geqslant 1$

$$\mathbf{f}_{2k}, \mathbf{f}_{2k+1} \in \mathcal{G}_k := \bigcap_{j=1}^{k} \mathbf{B}^j(\mathcal{S}) = \left(\mathop{+}_{j=1}^{k} \mathbf{B}^{-j\mathsf{H}}\{\mathbf{p}\} \right)^{\perp} = \left(\mathcal{K}_k(\mathbf{B}^{-\mathsf{H}}, \mathbf{B}^{-\mathsf{H}}\mathbf{p}) \right)^{\perp}.$$

The origin of IDR: primitive IDR

Sonneveld first made experiments and then gave a rigorous proof. It is easy to see that apart from the first two (arbitrarily chosen) residuals the constructed residuals are in the $\mathbf{B}$ image of the space $\mathcal{S} := \mathbf{p}^{\perp}$.

The same argument proves that in general (observe that the first two residuals $\mathbf{f}_0, \mathbf{f}_1$ are usually not in $\mathcal{S}$) for $k \geqslant 1$

$$\mathbf{f}_{2k}, \mathbf{f}_{2k+1} \in \mathcal{G}_k := \bigcap_{j=1}^{k} \mathbf{B}^j(\mathcal{S}) = \left(\mathop{+}_{j=1}^{k} \mathbf{B}^{-j\mathsf{H}}\{\mathbf{p}\} \right)^{\perp} = \left(\mathcal{K}_k(\mathbf{B}^{-\mathsf{H}}, \mathbf{B}^{-\mathsf{H}}\mathbf{p}) \right)^{\perp}.$$

Sonneveld proved that the dimensions of the spaces constructed are shrinking. This is the essence of the first IDR Theorem. He did not use the description as an orthogonal complement of a Krylov subspace as it is done here. We remark that generically $\dim(\mathcal{K}_n(\mathbf{B}^{-\mathsf{H}}, \mathbf{B}^{-\mathsf{H}}\mathbf{p})) = n$.

The origin of IDR: primitive IDR

Sonneveld first made experiments and then gave a rigorous proof. It is easy to see that apart from the first two (arbitrarily chosen) residuals the constructed residuals are in the $\mathbf{B}$ image of the space $\mathcal{S} := \mathbf{p}^{\perp}$.

The same argument proves that in general (observe that the first two residuals $\mathbf{f}_0, \mathbf{f}_1$ are usually not in $\mathcal{S}$) for $k \geqslant 1$

$$\mathbf{f}_{2k}, \mathbf{f}_{2k+1} \in \mathcal{G}_k := \bigcap_{j=1}^{k} \mathbf{B}^j(\mathcal{S}) = \left(\mathop{+}_{j=1}^{k} \mathbf{B}^{-j\mathsf{H}}\{\mathbf{p}\} \right)^{\perp} = \left(\mathcal{K}_k(\mathbf{B}^{-\mathsf{H}}, \mathbf{B}^{-\mathsf{H}}\mathbf{p}) \right)^{\perp}.$$

Sonneveld proved that the dimensions of the spaces constructed are shrinking. This is the essence of the first IDR Theorem. He did not use the description as an orthogonal complement of a Krylov subspace as it is done here. We remark that generically $\dim(\mathcal{K}_n(\mathbf{B}^{-\mathsf{H}}, \mathbf{B}^{-\mathsf{H}}\mathbf{p})) = n$.

Using the Krylov subspace point of view and the explicit orthogonalization against $\mathbf{p}$ before multiplication with $\mathbf{B}$, we see that indeed $\mathbf{f}_{2n} = \mathbf{B}\mathbf{o}_n = \mathbf{o}_n$.

The origin of IDR: primitive IDR

The three-term recurrence

$$\mathbf{f}_{k+1} = \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})), \quad \text{where} \quad \gamma_k = \frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_{k-1} - \mathbf{f}_k)},$$

is an "implementation" of the Induced Dimension Reduction (IDR) Theorem. The vectors constructed live in spaces of shrinking dimensions. Methods like this are called "IDR Algorithms".

The origin of IDR: primitive IDR

The three-term recurrence

$$\mathbf{f}_{k+1} = \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})), \quad \text{where} \quad \gamma_k = \frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_{k-1} - \mathbf{f}_k)},$$

is an "implementation" of the Induced Dimension Reduction (IDR) Theorem. The vectors constructed live in spaces of shrinking dimensions. Methods like this are called "IDR Algorithms".

Another implementation by Sonneveld can be used to solve "genuine" linear systems. The idea is to rewrite the linear system to Richardson iteration form,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = (\mathbf{I} - \mathbf{A})\mathbf{x} + \mathbf{b} =: \mathbf{B}\,\mathbf{x} + \mathbf{b}.$$

The origin of IDR: primitive IDR

The three-term recurrence

$$\mathbf{f}_{k+1} = \mathbf{B}(\mathbf{f}_k + \gamma_k(\mathbf{f}_k - \mathbf{f}_{k-1})), \quad \text{where} \quad \gamma_k = \frac{\mathbf{p}^{\mathsf{H}}\mathbf{f}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{f}_{k-1} - \mathbf{f}_k)},$$

is an "implementation" of the Induced Dimension Reduction (IDR) Theorem. The vectors constructed live in spaces of shrinking dimensions. Methods like this are called "IDR Algorithms".

Another implementation by Sonneveld can be used to solve "genuine" linear systems. The idea is to rewrite the linear system to Richardson iteration form,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = (\mathbf{I} - \mathbf{A})\mathbf{x} + \mathbf{b} =: \mathbf{B}\,\mathbf{x} + \mathbf{b}.$$

The classical Richardson iteration with a starting guess $\mathbf{x}_0$ is then given by

$$\mathbf{x}_{k+1} = (\mathbf{I} - \mathbf{A})\mathbf{x}_k + \mathbf{b}.$$

The origin of IDR: primitive IDR

With $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, the Richardson iteration is carried out as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k, \quad \mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A})\mathbf{r}_k.$$

The origin of IDR: primitive IDR

With $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, the Richardson iteration is carried out as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k, \quad \mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A})\mathbf{r}_k.$$

In a Richardson-type IDR Algorithm, the second equation is replaced by the update

$$\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A})(\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})), \quad \gamma_k = \frac{\mathbf{p}^{\mathsf{H}}\mathbf{r}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{r}_{k-1} - \mathbf{r}_k)}.$$

The origin of IDR: primitive IDR

With $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, the Richardson iteration is carried out as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k, \quad \mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A})\mathbf{r}_k.$$

In a Richardson-type IDR Algorithm, the second equation is replaced by the update

$$\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A})(\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})), \quad \gamma_k = \frac{\mathbf{p}^{\mathsf{H}}\mathbf{r}_k}{\mathbf{p}^{\mathsf{H}}(\mathbf{r}_{k-1} - \mathbf{r}_k)}.$$

The update of the iterates has to be modified accordingly,

$$\begin{aligned} -\mathbf{A}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{r}_{k+1} - \mathbf{r}_k &= (\mathbf{I} - \mathbf{A})(\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1}) - \mathbf{r}_k \\ &= (\mathbf{I} - \mathbf{A})(\mathbf{r}_k - \gamma_k\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}) - \mathbf{r}_k \\ &= -\mathbf{A}(\mathbf{r}_k + \gamma_k(\mathbf{I} - \mathbf{A})(\mathbf{x}_k - \mathbf{x}_{k-1})) \\ \Leftrightarrow \quad \mathbf{x}_{k+1} - \mathbf{x}_k &= \mathbf{r}_k + \gamma_k(\mathbf{I} - \mathbf{A})(\mathbf{x}_k - \mathbf{x}_{k-1}) \\ &= \mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1} + \mathbf{r}_k - \mathbf{r}_{k-1}). \end{aligned}$$

The origin of IDR: primitive IDR

Sonneveld terms the outcome the Primitive IDR Algorithm (Sonneveld, 2006):

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$$
$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{r}_0$$
$$\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{A}\mathbf{r}_0$$

For $k = 1, 2, \ldots$ do

$$\gamma_k = \mathbf{p}^\mathsf{T}\mathbf{r}_k / \mathbf{p}^\mathsf{T}(\mathbf{r}_{k-1} - \mathbf{r}_k)$$
$$\mathbf{s}_k = \mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) + \mathbf{s}_k$$
$$\mathbf{r}_{k+1} = \mathbf{s}_k - \mathbf{A}\mathbf{s}_k$$

done

The origin of IDR: primitive IDR

Sonneveld terms the outcome the Primitive IDR Algorithm (Sonneveld, 2006):

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$$
$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{r}_0$$
$$\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{A}\mathbf{r}_0$$

For $k = 1, 2, \ldots$ do

$$\gamma_k = \mathbf{p}^{\mathsf{T}}\mathbf{r}_k / \mathbf{p}^{\mathsf{T}}(\mathbf{r}_{k-1} - \mathbf{r}_k)$$
$$\mathbf{s}_k = \mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) + \mathbf{s}_k$$
$$\mathbf{r}_{k+1} = \mathbf{s}_k - \mathbf{A}\mathbf{s}_k$$

done

$$\mathbf{x}_{\text{old}} = \mathbf{x}_0$$
$$\mathbf{r}_{\text{old}} = \mathbf{b} - \mathbf{A}\mathbf{x}_{\text{old}}$$
$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \mathbf{r}_{\text{old}}$$
$$\mathbf{r}_{\text{new}} = \mathbf{r}_{\text{old}} - \mathbf{A}\mathbf{r}_{\text{old}}$$

While “not converged” do

$$\gamma = \mathbf{p}^{\mathsf{T}}\mathbf{r}_{\text{new}} / \mathbf{p}^{\mathsf{T}}(\mathbf{r}_{\text{old}} - \mathbf{r}_{\text{new}})$$
$$\mathbf{s} = \mathbf{r}_{\text{new}} + \gamma(\mathbf{r}_{\text{new}} - \mathbf{r}_{\text{old}})$$
$$\mathbf{x}_{\text{tmp}} = \mathbf{x}_{\text{new}} + \gamma(\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}) + \mathbf{s}$$
$$\mathbf{r}_{\text{tmp}} = \mathbf{s} - \mathbf{A}\mathbf{s}$$
$$\mathbf{x}_{\text{old}} = \mathbf{x}_{\text{new}},\ \mathbf{x}_{\text{new}} = \mathbf{x}_{\text{tmp}}$$
$$\mathbf{r}_{\text{old}} = \mathbf{r}_{\text{new}},\ \mathbf{r}_{\text{new}} = \mathbf{r}_{\text{tmp}}$$

done

The origin of IDR: primitive IDR

Sonneveld terms the outcome the Primitive IDR Algorithm (Sonneveld, 2006):

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$$
$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{r}_0$$
$$\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{A}\mathbf{r}_0$$

For $k = 1, 2, \ldots$ do

$$\gamma_k = \mathbf{p}^{\mathsf{T}}\mathbf{r}_k / \mathbf{p}^{\mathsf{T}}(\mathbf{r}_{k-1} - \mathbf{r}_k)$$
$$\mathbf{s}_k = \mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) + \mathbf{s}_k$$
$$\mathbf{r}_{k+1} = \mathbf{s}_k - \mathbf{A}\mathbf{s}_k$$

done

$$\mathbf{x}_{\text{old}} = \mathbf{x}_0$$
$$\mathbf{r}_{\text{old}} = \mathbf{b} - \mathbf{A}\mathbf{x}_{\text{old}}$$
$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \mathbf{r}_{\text{old}}$$
$$\mathbf{r}_{\text{new}} = \mathbf{r}_{\text{old}} - \mathbf{A}\mathbf{r}_{\text{old}}$$

While “not converged” do

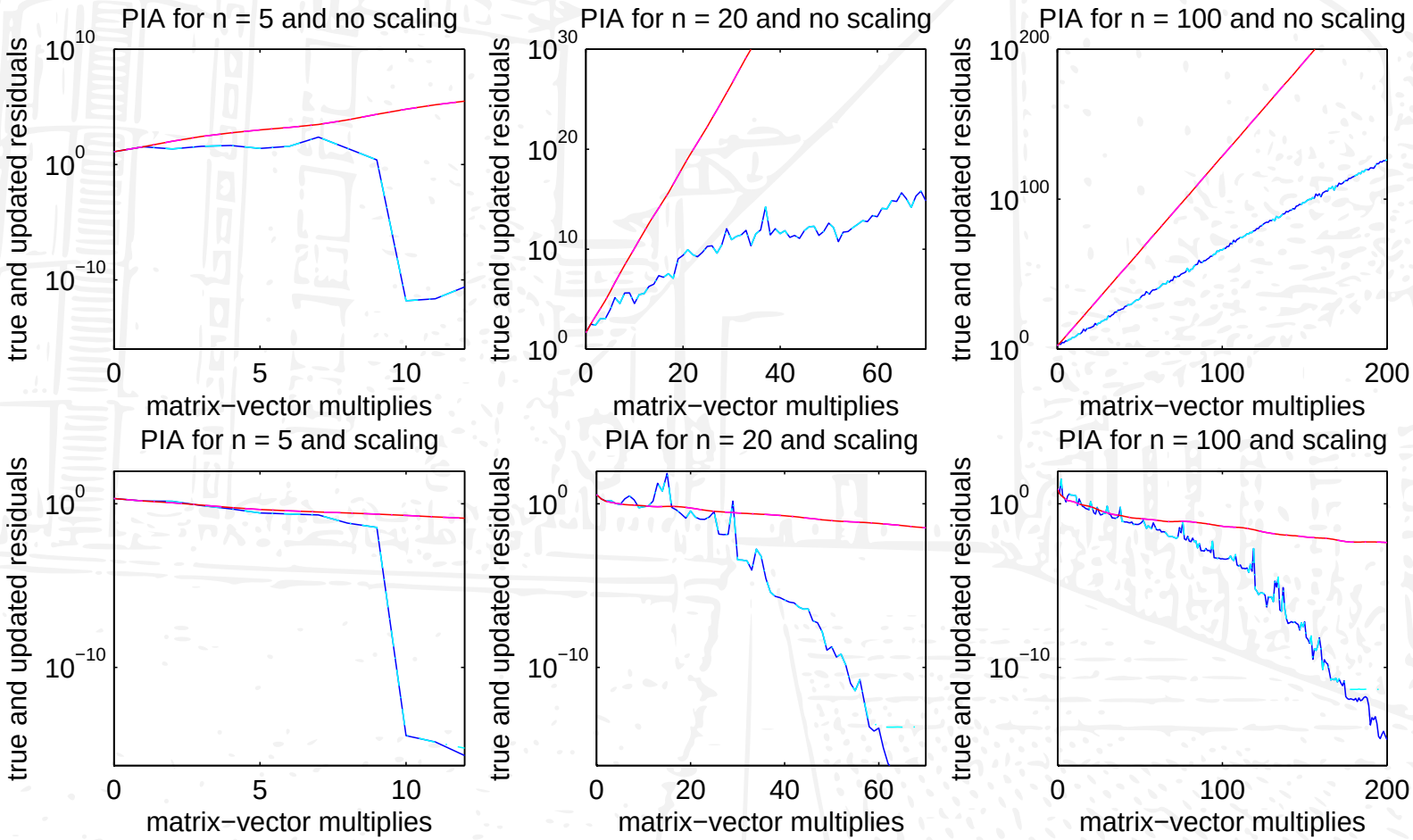
$$\gamma = \mathbf{p}^{\mathsf{T}}\mathbf{r}_{\text{new}} / \mathbf{p}^{\mathsf{T}}(\mathbf{r}_{\text{old}} - \mathbf{r}_{\text{new}})$$
$$\mathbf{s} = \mathbf{r}_{\text{new}} + \gamma(\mathbf{r}_{\text{new}} - \mathbf{r}_{\text{old}})$$
$$\mathbf{x}_{\text{tmp}} = \mathbf{x}_{\text{new}} + \gamma(\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}) + \mathbf{s}$$
$$\mathbf{r}_{\text{tmp}} = \mathbf{s} - \mathbf{A}\mathbf{s}$$
$$\mathbf{x}_{\text{old}} = \mathbf{x}_{\text{new}},\ \mathbf{x}_{\text{new}} = \mathbf{x}_{\text{tmp}}$$
$$\mathbf{r}_{\text{old}} = \mathbf{r}_{\text{new}},\ \mathbf{r}_{\text{new}} = \mathbf{r}_{\text{tmp}}$$

done

On the next slide we compare Richardson iteration (red) and PIA (blue).

The origin of IDR: primitive IDR

Impressions of “finite termination” and acceleration in finite precision:



The origin of IDR: primitive IDR

Sonneveld never did use PIA, as he considered it to be too unstable, instead he went on with a corresponding acceleration of the Gauß-Seidel method. In (Sonneveld, 2008) he terms this method Accelerated Gauß-Seidel (AGS) and refers to it as “[t]he very first IDR-algorithm [..]”, see page 6, Ibid.

The origin of IDR: primitive IDR

Sonneveld never did use PIA, as he considered it to be too unstable, instead he went on with a corresponding acceleration of the Gauß-Seidel method. In (Sonneveld, 2008) he terms this method Accelerated Gauß-Seidel (AGS) and refers to it as “[t]he very first IDR-algorithm [..]”, see page 6, Ibid.

This part of the story took place “in the background” in the year 1976.

The origin of IDR: primitive IDR

Sonneveld never did use PIA, as he considered it to be too unstable, instead he went on with a corresponding acceleration of the Gauß-Seidel method. In (Sonneveld, 2008) he terms this method Accelerated Gauß-Seidel (AGS) and refers to it as “[t]he very first IDR-algorithm [..]”, see page 6, Ibid.

This part of the story took place “in the background” in the year 1976.

In September 1979 Sonneveld did attend the IUTAM Symposium on Approximation Methods for Navier-Stokes Problems in Paderborn, Germany. At this symposium he presented a new variant of IDR based on a variable splitting $\mathbf{I} - \omega_j \mathbf{A}$, where ω_j is fixed for two steps and otherwise could be chosen freely, but non-zero.

The origin of IDR: primitive IDR

Sonneveld never did use PIA, as he considered it to be too unstable, instead he went on with a corresponding acceleration of the Gauß-Seidel method. In (Sonneveld, 2008) he terms this method Accelerated Gauß-Seidel (AGS) and refers to it as “[t]he very first IDR-algorithm [..]”, see page 6, Ibid.

This part of the story took place “in the background” in the year 1976.

In September 1979 Sonneveld did attend the IUTAM Symposium on Approximation Methods for Navier-Stokes Problems in Paderborn, Germany. At this symposium he presented a new variant of IDR based on a variable splitting $\mathbf{I} - \omega_j \mathbf{A}$, where ω_j is fixed for two steps and otherwise could be chosen freely, but non-zero.

This algorithm with minimization of every second residual is included in the proceedings from 1980 (Wesseling and Sonneveld, 1980). The connection to Krylov methods, e.g., BiCG/Lanczos, is also given there.

The origin of IDR: classical IDR

$\gamma_0 = 0$, $\mathbf{f}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$, $\Delta\mathbf{g}_0 = \mathbf{o}_n$, $\Delta\mathbf{y}_0 = \mathbf{o}_n$
For $k = 1, \ldots$ do
  $\mathbf{s}_k = \mathbf{f}_{k-1} + \gamma_{k-1}\Delta\mathbf{g}_{k-1}$
  $\mathbf{t}_k = \mathbf{A}\mathbf{s}_k$
  if $k = 1$ or k is even
    $\omega_k = (\mathbf{t}_k^{\mathsf{H}}\mathbf{s}_k)/(\mathbf{t}_k^{\mathsf{H}}\mathbf{t}_k)$
  else
    $\omega_k = \omega_{k-1}$
  end
  $\Delta\mathbf{x}_k = \gamma_{k-1}\Delta\mathbf{y}_{k-1} - \omega_k\mathbf{s}_k$
  $\Delta\mathbf{f}_k = \gamma_{k-1}\Delta\mathbf{g}_{k-1} - \omega_k\mathbf{t}_k$
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$
  $\mathbf{f}_k = \mathbf{f}_{k-1} + \Delta\mathbf{f}_k$
  if k is even
    $\Delta\mathbf{y}_k = \Delta\mathbf{y}_{k-1}$
    $\Delta\mathbf{g}_k = \Delta\mathbf{g}_{k-1}$
  else
    $\Delta\mathbf{y}_k = \Delta\mathbf{x}_k$
    $\Delta\mathbf{g}_k = \Delta\mathbf{f}_k$
  end
  $\gamma_k = -(\mathbf{p}^{\mathsf{H}}\mathbf{f}_k)/(\mathbf{p}^{\mathsf{H}}\Delta\mathbf{g}_k)$
done

The origin of IDR: classical IDR

$\gamma_0 = 0$, $\mathbf{f}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$, $\Delta\mathbf{g}_0 = \mathbf{o}_n$, $\Delta\mathbf{y}_0 = \mathbf{o}_n$
For $k = 1, \ldots$ do
  $\mathbf{s}_k = \mathbf{f}_{k-1} + \gamma_{k-1}\Delta\mathbf{g}_{k-1}$
  $\mathbf{t}_k = \mathbf{A}\mathbf{s}_k$
  if $k = 1$ or k is even
    $\omega_k = (\mathbf{t}_k^{\mathsf{H}}\mathbf{s}_k)/(\mathbf{t}_k^{\mathsf{H}}\mathbf{t}_k)$
  else
    $\omega_k = \omega_{k-1}$
  end
  $\Delta\mathbf{x}_k = \gamma_{k-1}\Delta\mathbf{y}_{k-1} - \omega_k\mathbf{s}_k$
  $\Delta\mathbf{f}_k = \gamma_{k-1}\Delta\mathbf{g}_{k-1} - \omega_k\mathbf{t}_k$
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$
  $\mathbf{f}_k = \mathbf{f}_{k-1} + \Delta\mathbf{f}_k$
  if k is even
    $\Delta\mathbf{y}_k = \Delta\mathbf{y}_{k-1}$
    $\Delta\mathbf{g}_k = \Delta\mathbf{g}_{k-1}$
  else
    $\Delta\mathbf{y}_k = \Delta\mathbf{x}_k$
    $\Delta\mathbf{g}_k = \Delta\mathbf{f}_k$
  end
  $\gamma_k = -(\mathbf{p}^{\mathsf{H}}\mathbf{f}_k)/(\mathbf{p}^{\mathsf{H}}\Delta\mathbf{g}_k)$
done

This is the original IDR Algorithm from page 551 of (Wesseling and Sonneveld, 1980).

The origin of IDR: classical IDR

$\gamma_0 = 0$, $\mathbf{f}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$, $\Delta\mathbf{g}_0 = \mathbf{o}_n$, $\Delta\mathbf{y}_0 = \mathbf{o}_n$
For $k = 1, \ldots$ do
  $\mathbf{s}_k = \mathbf{f}_{k-1} + \gamma_{k-1}\Delta\mathbf{g}_{k-1}$
  $\mathbf{t}_k = \mathbf{A}\mathbf{s}_k$
  if $k = 1$ or k is even
    $\omega_k = (\mathbf{t}_k^{\mathsf{H}}\mathbf{s}_k)/(\mathbf{t}_k^{\mathsf{H}}\mathbf{t}_k)$
  else
    $\omega_k = \omega_{k-1}$
  end
  $\Delta\mathbf{x}_k = \gamma_{k-1}\Delta\mathbf{y}_{k-1} - \omega_k\mathbf{s}_k$
  $\Delta\mathbf{f}_k = \gamma_{k-1}\Delta\mathbf{g}_{k-1} - \omega_k\mathbf{t}_k$
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$
  $\mathbf{f}_k = \mathbf{f}_{k-1} + \Delta\mathbf{f}_k$
  if k is even
    $\Delta\mathbf{y}_k = \Delta\mathbf{y}_{k-1}$
    $\Delta\mathbf{g}_k = \Delta\mathbf{g}_{k-1}$
  else
    $\Delta\mathbf{y}_k = \Delta\mathbf{x}_k$
    $\Delta\mathbf{g}_k = \Delta\mathbf{f}_k$
  end
  $\gamma_k = -(\mathbf{p}^{\mathsf{H}}\mathbf{f}_k)/(\mathbf{p}^{\mathsf{H}}\Delta\mathbf{g}_k)$
done

This is the original IDR Algorithm from page 551 of (Wesseling and Sonneveld, 1980).

It uses OrthoRes(1) in the first step and a residual (these are the $-\mathbf{f}_{2j}$) minimization every second step.

The origin of IDR: classical IDR

$\gamma_0 = 0$, $\mathbf{f}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$, $\Delta\mathbf{g}_0 = \mathbf{o}_n$, $\Delta\mathbf{y}_0 = \mathbf{o}_n$
For $k = 1, \ldots$ do
  $\mathbf{s}_k = \mathbf{f}_{k-1} + \gamma_{k-1}\Delta\mathbf{g}_{k-1}$
  $\mathbf{t}_k = \mathbf{A}\mathbf{s}_k$
  if $k = 1$ or k is even
    $\omega_k = (\mathbf{t}_k^{\mathsf{H}}\mathbf{s}_k)/(\mathbf{t}_k^{\mathsf{H}}\mathbf{t}_k)$
  else
    $\omega_k = \omega_{k-1}$
  end
  $\Delta\mathbf{x}_k = \gamma_{k-1}\Delta\mathbf{y}_{k-1} - \omega_k\mathbf{s}_k$
  $\Delta\mathbf{f}_k = \gamma_{k-1}\Delta\mathbf{g}_{k-1} - \omega_k\mathbf{t}_k$
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$
  $\mathbf{f}_k = \mathbf{f}_{k-1} + \Delta\mathbf{f}_k$
  if k is even
    $\Delta\mathbf{y}_k = \Delta\mathbf{y}_{k-1}$
    $\Delta\mathbf{g}_k = \Delta\mathbf{g}_{k-1}$
  else
    $\Delta\mathbf{y}_k = \Delta\mathbf{x}_k$
    $\Delta\mathbf{g}_k = \Delta\mathbf{f}_k$
  end
  $\gamma_k = -(\mathbf{p}^{\mathsf{H}}\mathbf{f}_k)/(\mathbf{p}^{\mathsf{H}}\Delta\mathbf{g}_k)$
done

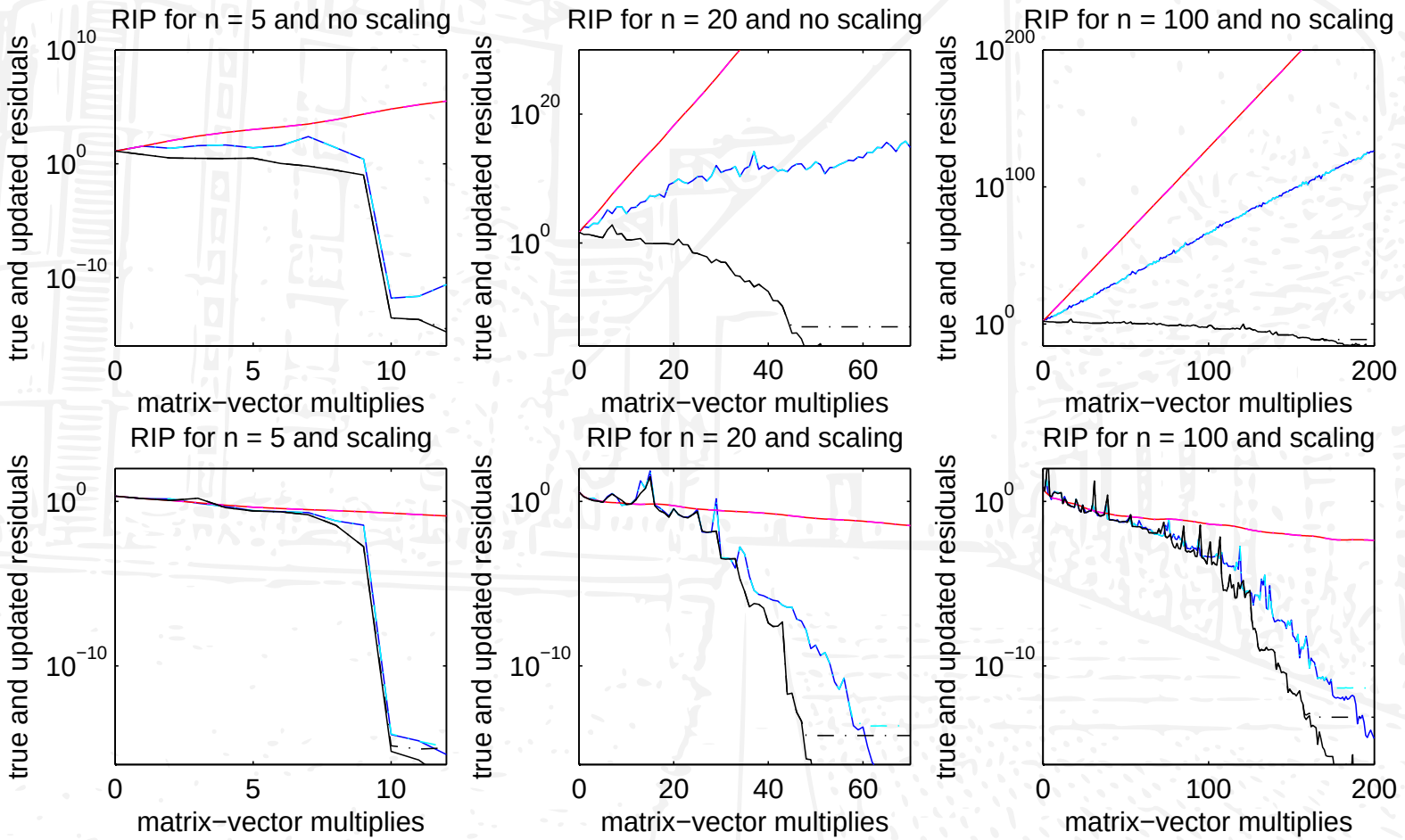
This is the original IDR Algorithm from page 551 of (Wesseling and Sonneveld, 1980).

It uses OrthoRes(1) in the first step and a residual (these are the $-\mathbf{f}_{2j}$) minimization every second step.

The finite termination property follows from a generalization of the IDR Theorem based on commutativity of the linear polynomials $\mathbf{I} - \omega_j\mathbf{A}$.

The origin of IDR: classical IDR

A numerical comparison of Richardson iteration, original IDR, and PIA.



Evolution: CGS and BiCGStab

IDR was presented at a Symposium on CFD. The Numerical Linear Algebra community missed it completely. This changed, when Sonneveld gained more understanding of Krylov subspace methods and developed “better variants” of IDR.

Evolution: CGS and BiCGStab

IDR was presented at a Symposium on CFD. The Numerical Linear Algebra community missed it completely. This changed, when Sonneveld gained more understanding of Krylov subspace methods and developed “better variants” of IDR.

There are two well-known methods based on IDR: CGS and BiCGStab.

Evolution: CGS and BiCGStab

IDR was presented at a Symposium on CFD. The Numerical Linear Algebra community missed it completely. This changed, when Sonneveld gained more understanding of Krylov subspace methods and developed “better variants” of IDR.

There are two well-known methods based on IDR: CGS and BiCGStab.

CGS, dating to 1984 (Sonneveld, 1984; Sonneveld, 1989), was the outcome of the understanding that one can do Lanczos without the need for $\mathbf{A}^T$, which follows from the analysis of IDR.

Evolution: CGS and BiCGStab

IDR was presented at a Symposium on CFD. The Numerical Linear Algebra community missed it completely. This changed, when Sonneveld gained more understanding of Krylov subspace methods and developed “better variants” of IDR.

There are two well-known methods based on IDR: CGS and BiCGStab.

CGS, dating to 1984 (Sonneveld, 1984; Sonneveld, 1989), was the outcome of the understanding that one can do Lanczos without the need for $\mathbf{A}^T$, which follows from the analysis of IDR.

The analysis of IDR from the Krylov subspace point of view was based on the orthogonality properties of the residual polynomials. This immediately leads to the observation that all IDR methods construct residual polynomials that are products of auxiliary polynomials with the Lanczos polynomials.

Evolution: CGS and BiCGStab

CGS was based on choosing the auxiliary polynomial equal to the Lanczos polynomial. This has two advantages: It is at hand and the contraction is enhanced in case of contraction.

Evolution: CGS and BiCGStab

CGS was based on choosing the auxiliary polynomial equal to the Lanczos polynomial. This has two advantages: It is at hand and the contraction is enhanced in case of contraction.

CGS has a severe disadvantage: Also the erratic behavior is amplified, thus CGS is more prone to rounding errors than BiCG and the ultimately attainable accuracy is larger.

Evolution: CGS and BiCGStab

CGS was based on choosing the auxiliary polynomial equal to the Lanczos polynomial. This has two advantages: It is at hand and the contraction is enhanced in case of contraction.

CGS has a severe disadvantage: Also the erratic behavior is amplified, thus CGS is more prone to rounding errors than BiCG and the ultimately attainable accuracy is larger.

If only a moderate backward error reduction is of interest and BiCG converges quite well, CGS is a better choice. But many problems are not of this type, and for these one might want to use a transpose-free method.

Evolution: CGS and BiCGStab

CGS was based on choosing the auxiliary polynomial equal to the Lanczos polynomial. This has two advantages: It is at hand and the contraction is enhanced in case of contraction.

CGS has a severe disadvantage: Also the erratic behavior is amplified, thus CGS is more prone to rounding errors than BiCG and the ultimately attainable accuracy is larger.

If only a moderate backward error reduction is of interest and BiCG converges quite well, CGS is a better choice. But many problems are not of this type, and for these one might want to use a transpose-free method.

Sonneveld thought about rewriting the IDR Algorithm from (Wesseling and Sonneveld, 1980) and discussed this during a weekend with Henk van der Vorst. The resulting BiCGStab (van der Vorst and Sonneveld, 1990; van der Vorst, 1992) is mathematically equivalent to IDR. In the title of the report CGS was explicitely mentioned and Sonneveld was one of the authors . . .

Evolution: CGS and BiCGStab

> *Early ideas by Sonneveld (1984) for improvements in the bi-Conjugate Gradient (Bi-CG) method, for the solution of unsymmetric linear systems, intrigued me for a long time. Sonneveld had a brilliant idea for doubling the speed of convergence of Bi-CG for virtually the same computational costs: CGS. He also published a rather obscure method under the name of IDR. I doubt whether that paper got more than two or three citations altogether. The eventual understanding of that method and the reformulation of it, so that rounding errors had much less bad influence on its speed of convergence, led to the so frequently cited Bi-CGSTAB paper (1992).*

— Henk van der Vorst on IDR and CGS by Peter Sonneveld, see in-cites, September 2001, `http://www.in-cites.com/papers/dr-henk-van-der-vorst.html`.

Evolution: LTPM

Soon it was realized by other researchers that the new methods are based on residual polynomials which are products of auxiliary polynomials and the Lanczos polynomials.

Evolution: LTPM

Soon it was realized by other researchers that the new methods are based on residual polynomials which are products of auxiliary polynomials and the Lanczos polynomials.

Gutknecht (Gutknecht, 1997) coined the term "Lanczos-type product method" (LTPM) for these methods. A plethora of new Krylov subspace methods popped into existence:

Evolution: LTPM

Soon it was realized by other researchers that the new methods are based on residual polynomials which are products of auxiliary polynomials and the Lanczos polynomials.

Gutknecht (Gutknecht, 1997) coined the term "Lanczos-type product method" (LTPM) for these methods. A plethora of new Krylov subspace methods popped into existence:

- BiCGStab2 (Gutknecht, 1993),
- BiCGStab(ℓ) (Sleijpen and Fokkema, 1993),
- GCGS (Fokkema et al., 1996), includes CGS2 and shifted CGS,
- GPBiCG (Zhang, 1997) = BiCG$\times$MR2 (Gutknecht, 1997),
- ML(k)BiCGStab (Yeung and Chan, 2000),
- BiCG$\times$MR2_2$\times$2 (Röllin and Gutknecht, 2002),
- GPBiCG(m,l) (Fujino, 2002),
- BiCGSafe (Fujino et al., 2005), ...

Evolution: LTPM

Soon people observed that smoothed variants can be squared and product-type methods can be smoothed. This added to the plethora:

Evolution: LTPM

Soon people observed that smoothed variants can be squared and product-type methods can be smoothed. This added to the plethora:

- QMRS (Freund and Szeto, 1991; Freund and Szeto, 1992a; Freund and Szeto, 1992b),
- TFQMR (Freund, 1993),
- QMRCGStab (Chan et al., 1994),
- general smoothing techniques: (Zhou and Walker, 1994).

Evolution: LTPM

Soon people observed that smoothed variants can be squared and product-type methods can be smoothed. This added to the plethora:

- QMRS (Freund and Szeto, 1991; Freund and Szeto, 1992a; Freund and Szeto, 1992b),
- TFQMR (Freund, 1993),
- QMRCGStab (Chan et al., 1994),
- general smoothing techniques: (Zhou and Walker, 1994).

It was even considered to implement algorithms based on the (two-sided) Lanczos process via "transpose-free implementations" (Chan et al., 1991; Chan et al., 1998). These are called

- squared Lanczos,
- TFiBiCG, and
- TFiQMR.

Evolution: LTPM

The main problem, namely the breakdown of the underlying Lanczos process and its instability in finite precision has only partially been addressed.

Evolution: LTPM

The main problem, namely the breakdown of the underlying Lanczos process and its instability in finite precision has only partially been addressed.

"Look-ahead for (Bi)CG"S was considered in (Brezinski and Redivo Zaglia, 1994), the resulting algorithm is called BSMRZS; look-ahead for BiCGStab (and related LTPM) was considered in (Brezinski and Redivo-Zaglia, 1995). In (Gutknecht and Ressel, 2000) look-ahead for general LTPM based on three-term recurrences was considered.

Evolution: LTPM

The main problem, namely the breakdown of the underlying Lanczos process and its instability in finite precision has only partially been addressed.

"Look-ahead for (Bi)CG"S was considered in (Brezinski and Redivo Zaglia, 1994), the resulting algorithm is called BSMRZS; look-ahead for BiCGStab (and related LTPM) was considered in (Brezinski and Redivo-Zaglia, 1995). In (Gutknecht and Ressel, 2000) look-ahead for general LTPM based on three-term recurrences was considered.

Stability in finite precision was investigated by very few people.

Evolution: LTPM

The main problem, namely the breakdown of the underlying Lanczos process and its instability in finite precision has only partially been addressed.

"Look-ahead for (Bi)CG"S was considered in (Brezinski and Redivo Zaglia, 1994), the resulting algorithm is called BSMRZS; look-ahead for BiCGStab (and related LTPM) was considered in (Brezinski and Redivo-Zaglia, 1995). In (Gutknecht and Ressel, 2000) look-ahead for general LTPM based on three-term recurrences was considered.

Stability in finite precision was investigated by very few people.

Of all "new" methods, only ML(k)BiCGStab differs substantially from the others: This method is based on s left starting vectors (shadow vectors) and one right starting vector (zeroth residual).

The origin of IDR(s): ancestors

ML(k)BiCGStab was largely neglected by the Numerical Linear Algebra community. The main reason is the very technical paper, where the appendix contained the derivation of the computation of the scalars. Currently, Man-Chung Yeung is reconsidering ML(k)BiCGStab and developing variants that exploit the freedom inherent in the method (Yeung, 2009).

The origin of IDR(s): ancestors

ML(k)BiCGStab was largely neglected by the Numerical Linear Algebra community. The main reason is the very technical paper, where the appendix contained the derivation of the computation of the scalars. Currently, Man-Chung Yeung is reconsidering ML(k)BiCGStab and developing variants that exploit the freedom inherent in the method (Yeung, 2009).

Without knowing anything about ML(k)BiCGStab in 2006 the IDR idea was reconsidered. Peter Sonneveld together with Martin van Gijzen developed a new variant of IDR based on multiple shadow vectors: IDR(s) ($\approx$ IDR(s)ORes).

The origin of IDR(s): ancestors

ML(k)BiCGStab was largely neglected by the Numerical Linear Algebra community. The main reason is the very technical paper, where the appendix contained the derivation of the computation of the scalars. Currently, Man-Chung Yeung is reconsidering ML(k)BiCGStab and developing variants that exploit the freedom inherent in the method (Yeung, 2009).

Without knowing anything about ML(k)BiCGStab in 2006 the IDR idea was reconsidered. Peter Sonneveld together with Martin van Gijzen developed a new variant of IDR based on multiple shadow vectors: IDR(s) ($\approx$ IDR(s)ORes).

Nobody was thinking any more about IDR and Peter Sonneveld calls this "an example of serendipity" ...

... so what did happen?

The origin of IDR(s): ancestors

ML(k)BiCGStab was largely neglected by the Numerical Linear Algebra community. The main reason is the very technical paper, where the appendix contained the derivation of the computation of the scalars. Currently, Man-Chung Yeung is reconsidering ML(k)BiCGStab and developing variants that exploit the freedom inherent in the method (Yeung, 2009).

Without knowing anything about ML(k)BiCGStab in 2006 the IDR idea was reconsidered. Peter Sonneveld together with Martin van Gijzen developed a new variant of IDR based on multiple shadow vectors: IDR(s) ($\approx$ IDR(s)ORes).

Nobody was thinking any more about IDR and Peter Sonneveld calls this "an example of serendipity" . . .

. . . so what did happen?

The following is an excerpt of an e-mail and a copy of slide 36 of the after dinner talk by Peter Sonneveld at the Thirty-fourth Woudschoten Conference.

The origin of IDR(s): rebirth of IDR

```
Date: Wed, 17 May 2006 14:02:27 +0200 (CEST)
From: Jens-Peter M. Zemke <zemke@xxxxxxxxxxxxx>
To:  <p.sonneveld@xxxxxxxxxxxxxx>
Cc: Jens-Peter M. Zemke <zemke@xxxxxxxxxxxxx>
Subject: A question about IDR

[..] entitled
   "The method of induced dimension reduction, an
    iterative solver for non-symmetric linear systems"
with the annotation "Publication in preparation".

My question is: What happended to this paper?

More precisely formulated:
   - Did it evolve into the CGS paper?
or [..]
```

The origin of IDR(s): rebirth of IDR

Zemke, and a short monologue

- 2006: Jens-Peter Zemke, from Hamburg, mails: What happened to IDR?
- Have to read carefully the 1980 version of the theorem, and the ancient history.
- Theorem used a space $\mathcal{S}$, not just $\boldsymbol{p}^{\perp}$.
- Serendipity moment: Why didn't I use more vectors $\boldsymbol{p}$, say s instead of 1???
- Because it costs $s+1$ matvecs per $\mathcal{G}_j$-space.
- But maybe there is more dimension reduction per $\mathcal{G}_j$
- .... Never thought about, must try... and call it IDR(s)

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^{\mathsf{H}}\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^{\mathsf{H}}\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n$, $n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n$, $n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The steps in the s-loop only differ from the first block in that no new ω_j is computed.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The steps in the s-loop only differ from the first block in that no new ω_j is computed.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

$\Rightarrow \mathbf{v}_{n-1} = (\mathbf{I} - \nabla\mathbf{R}_{n-s:n-1}(\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H)\mathbf{r}_{n-1}$

A few remarks:

We can start with any (simple) Krylov subspace method.

The steps in the s-loop only differ from the first block in that no new ω_j is computed.

IDR(s)ORes is based on oblique projections.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H \nabla\mathbf{R}_{n-s:n-1})^{-1} \mathbf{P}^H \mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j \mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while

A few remarks:

We can start with any (simple) Krylov subspace method.

The steps in the s-loop only differ from the first block in that no new ω_j is computed.

IDR(s)ORes is based on oblique projections.

The prototype IDR(s) (without the recurrences for $\mathbf{x}_n$ and thus already slightly rewritten)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
compute $\mathbf{R}_{s+1} = \mathbf{R}_{0:s} = (\mathbf{r}_0, \ldots, \mathbf{r}_s)$ using, e.g., ORTHORES
$\nabla\mathbf{R}_{1:s} = (\nabla\mathbf{r}_1, \ldots, \nabla\mathbf{r}_s) = (\mathbf{r}_1 - \mathbf{r}_0, \ldots, \mathbf{r}_s - \mathbf{r}_{s-1})$
$n \leftarrow s+1, j \leftarrow 1$
while not converged
  $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
  $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
  compute ω_j
  $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
  $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
  $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  for $k = 1, \ldots, s$
    $\mathbf{c}_n = (\mathbf{P}^H\nabla\mathbf{R}_{n-s:n-1})^{-1}\mathbf{P}^H\mathbf{r}_{n-1}$
    $\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n$
    $\nabla\mathbf{r}_n = -\nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n - \omega_j\mathbf{A}\mathbf{v}_{n-1}$
    $\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla\mathbf{r}_n, n \leftarrow n+1$
    $\nabla\mathbf{R}_{n-s:n-1} = (\nabla\mathbf{r}_{n-s}, \ldots, \nabla\mathbf{r}_{n-1})$
  end for
  $j \leftarrow j+1$
end while
$\Rightarrow \mathbf{r}_n = (\mathbf{I} - \omega_j\mathbf{A})\mathbf{v}_{n-1}$

A few remarks:

We can start with any (simple) Krylov subspace method.

The steps in the s-loop only differ from the first block in that no new ω_j is computed.

IDR(s)ORes is based on oblique projections and $s+1$ consecutive multiplications with the same linear factor $\mathbf{I} - \omega_j\mathbf{A}$.

Understanding IDR: Hessenberg decompositions

Essential features of Krylov subspace methods can be described by a Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T. \tag{1}$$

Here, $\mathbf{H}_n$ denotes an unreduced Hessenberg matrix.

Understanding IDR: Hessenberg decompositions

Essential features of Krylov subspace methods can be described by a Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T. \tag{1}$$

Here, $\mathbf{H}_n$ denotes an unreduced Hessenberg matrix.

In the perturbed case, e.g., in finite precision and/or based on inexact matrix-vector multiplies, we obtain a perturbed Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T. \tag{2}$$

Understanding IDR: Hessenberg decompositions

Essential features of Krylov subspace methods can be described by a Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T. \tag{1}$$

Here, $\mathbf{H}_n$ denotes an unreduced Hessenberg matrix.

In the perturbed case, e.g., in finite precision and/or based on inexact matrix-vector multiplies, we obtain a perturbed Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T. \tag{2}$$

The matrix $\mathbf{H}_n$ of the perturbed variant will, in general, still be unreduced.

IDR: Generalized Hessenberg decompositions

In case of IDR, we have to consider generalized Hessenberg decompositions

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T \tag{3}$$

with upper triangular (possibly even singular) $\mathbf{U}_n$.

IDR: Generalized Hessenberg decompositions

In case of IDR, we have to consider generalized Hessenberg decompositions

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T \tag{3}$$

and perturbed generalized Hessenberg decompositions

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T \tag{4}$$

with upper triangular (possibly even singular) $\mathbf{U}_n$.

IDR: Generalized Hessenberg decompositions

In case of IDR, we have to consider generalized Hessenberg decompositions

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T \tag{3}$$

and perturbed generalized Hessenberg decompositions

$$\mathbf{A}\mathbf{Q}_n\mathbf{U}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{H}}_n = \mathbf{Q}_n\mathbf{H}_n + \mathbf{q}_{n+1}h_{n+1,n}\mathbf{e}_n^T \tag{4}$$

with upper triangular (possibly even singular) $\mathbf{U}_n$.

Generalized Hessenberg decompositions correspond to a skew projection of the pencil $(\mathbf{A}, \mathbf{I})$ to the pencil $(\mathbf{H}_n, \mathbf{U}_n)$ as long as $\mathbf{Q}_{n+1}$ has full rank.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$,

Ritz-Galërkin: approximate $\mathbf{J} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ by $\mathbf{J}_n := \mathbf{S}_n^{-1}\mathbf{H}_n\mathbf{S}_n$.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$,

Ritz-Galërkin: approximate $\mathbf{J} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ by $\mathbf{J}_n := \mathbf{S}_n^{-1}\mathbf{H}_n\mathbf{S}_n$,
and $\mathbf{V}$ by $\mathbf{V}_n := \mathbf{Q}_n\mathbf{S}_n$.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$,

Ritz-Galërkin: approximate $\mathbf{J} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ by $\mathbf{J}_n := \mathbf{S}_n^{-1}\mathbf{H}_n\mathbf{S}_n$,

and $\mathbf{V}$ by $\mathbf{V}_n := \mathbf{Q}_n\mathbf{S}_n$,

"functions": approximate $f(\mathbf{A})\mathbf{q} = p(\mathbf{A})\mathbf{q}$ by $\mathbf{Q}_n f(\mathbf{H}_n)\mathbf{e}_1$ or $\mathbf{Q}_{n+1} f([\underline{\mathbf{H}}_n, \mathbf{f}])\underline{\mathbf{e}}_1$.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$,

Ritz-Galërkin: approximate $\mathbf{J} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ by $\mathbf{J}_n := \mathbf{S}_n^{-1}\mathbf{H}_n\mathbf{S}_n$,
and $\mathbf{V}$ by $\mathbf{V}_n := \mathbf{Q}_n\mathbf{S}_n$,

"functions": approximate $f(\mathbf{A})\mathbf{q} = p(\mathbf{A})\mathbf{q}$ by $\mathbf{Q}_n f(\mathbf{H}_n)\mathbf{e}_1$ or $\mathbf{Q}_{n+1} f([\underline{\mathbf{H}}_n, \mathbf{f}])\underline{\mathbf{e}}_1$.

To every method from one class corresponds a method of the other.

Understanding IDR: QOR/QMR/Ritz-Galërkin

There are various well-known approaches based on such Hessenberg decompositions, e.g.,

QOR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\mathbf{x}_n := \mathbf{Q}_n\mathbf{H}_n^{-1}\mathbf{e}_1\|\mathbf{r}_0\|$,

QMR: approximate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}_0$ by $\underline{\mathbf{x}}_n := \mathbf{Q}_n\underline{\mathbf{H}}_n^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$,

Ritz-Galërkin: approximate $\mathbf{J} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ by $\mathbf{J}_n := \mathbf{S}_n^{-1}\mathbf{H}_n\mathbf{S}_n$,
and $\mathbf{V}$ by $\mathbf{V}_n := \mathbf{Q}_n\mathbf{S}_n$,

"functions": approximate $f(\mathbf{A})\mathbf{q} = p(\mathbf{A})\mathbf{q}$ by $\mathbf{Q}_n f(\mathbf{H}_n)\mathbf{e}_1$ or $\mathbf{Q}_{n+1} f([\underline{\mathbf{H}}_n, \mathbf{f}])\underline{\mathbf{e}}_1$.

To every method from one class corresponds a method of the other.

These approaches extend easily to generalized Hessenberg decompositions.

Understanding IDR: OrthoRes-type methods

The entries of the Hessenberg matrices of these Hessenberg decompositions are defined in different variations.

Understanding IDR: OrthoRes-type methods

The entries of the Hessenberg matrices of these Hessenberg decompositions are defined in different variations.

Three well-known ways for implementing the QOR/QMR approach are commonly denoted as OrthoRes/OrthoMin/OrthoDir.

Understanding IDR: OrthoRes-type methods

The entries of the Hessenberg matrices of these Hessenberg decompositions are defined in different variations.

Three well-known ways for implementing the QOR/QMR approach are commonly denoted as OrthoRes/OrthoMin/OrthoDir.

OrthoRes-type methods have a Hessenberg decomposition

$$\mathbf{AR}_n = \mathbf{R}_{n+1}\underline{\mathbf{H}}_n^{\circ} = \mathbf{R}_n\mathbf{H}_n^{\circ} + \mathbf{r}_{n+1}h_{n+1,n}^{\circ}\mathbf{e}_n^{\mathsf{T}}, \tag{5}$$

where $\mathbf{e}^{\mathsf{T}}\underline{\mathbf{H}}_n^{\circ} = \mathbf{o}_n^{\mathsf{T}}$, $\mathbf{e}^{\mathsf{T}} = (1, \ldots, 1)$.

Understanding IDR: OrthoRes-type methods

The entries of the Hessenberg matrices of these Hessenberg decompositions are defined in different variations.

Three well-known ways for implementing the QOR/QMR approach are commonly denoted as OrthoRes/OrthoMin/OrthoDir.

OrthoRes-type methods have a Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\underline{\mathbf{H}}_n^{\circ} = \mathbf{R}_n\mathbf{H}_n^{\circ} + \mathbf{r}_{n+1}h_{n+1,n}^{\circ}\mathbf{e}_n^{\mathsf{T}}, \tag{5}$$

where $\mathbf{e}^{\mathsf{T}}\underline{\mathbf{H}}_n^{\circ} = \mathbf{o}_n^{\mathsf{T}}$, $\mathbf{e}^{\mathsf{T}} = (1,\ldots,1)$, and the matrix

$$\mathbf{R}_{n+1} = (\mathbf{r}_0,\ldots,\mathbf{r}_n) = \mathbf{Q}_{n+1}\,\mathrm{diag}\left(\frac{\|\mathbf{r}_0\|}{\|\mathbf{q}_1\|},\ldots,\frac{\|\mathbf{r}_n\|}{\|\mathbf{q}_{n+1}\|}\right) \tag{6}$$

is diagonally scaled to be the matrix of residual vectors.

Understanding IDR: OrthoRes-type methods

The entries of the Hessenberg matrices of these Hessenberg decompositions are defined in different variations.

Three well-known ways for implementing the QOR/QMR approach are commonly denoted as OrthoRes/OrthoMin/OrthoDir.

OrthoRes-type methods have a generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{U}_n = \mathbf{R}_{n+1}\underline{\mathbf{H}}_n^{\circ} = \mathbf{R}_n\mathbf{H}_n^{\circ} + \mathbf{r}_{n+1}h_{n+1,n}^{\circ}\mathbf{e}_n^{\mathsf{T}}, \tag{5}$$

where $\mathbf{e}^{\mathsf{T}}\underline{\mathbf{H}}_n^{\circ} = \mathbf{o}_n^{\mathsf{T}}$, $\mathbf{e}^{\mathsf{T}} = (1,\ldots,1)$, and the matrix

$$\mathbf{R}_{n+1} = (\mathbf{r}_0,\ldots,\mathbf{r}_n) = \mathbf{Q}_{n+1}\,\mathrm{diag}\left(\frac{\|\mathbf{r}_0\|}{\|\mathbf{q}_1\|},\ldots,\frac{\|\mathbf{r}_n\|}{\|\mathbf{q}_{n+1}\|}\right) \tag{6}$$

is diagonally scaled to be the matrix of residual vectors.

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}
\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\
&= (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \qquad (7)\\
\mathbf{r}_n &:= (\mathbf{I} - \omega_j\mathbf{A})\,\mathbf{v}_{n-1}\,.
\end{aligned}$$

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= \left(1-\gamma_s^{(n)}\right)\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}\left(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)}\right)\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ \mathbf{r}_n &:= \left(\mathbf{I}-\omega_j\mathbf{A}\right)\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

compare with the so-called “index functions” (Yeung/Boley, 2005).

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ \mathbf{r}_n &:= (\mathbf{I}-\omega_j\mathbf{A})\,\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

compare with the so-called “index functions” (Yeung/Boley, 2005).

Removing $\mathbf{v}_{n-1}$ from the recurrence we obtain the generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ. \tag{8}$$

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= \left(1-\gamma_s^{(n)}\right)\mathbf{r}_{n-1} + \sum_{\ell=1}^{s-1}\left(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)}\right)\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ \mathbf{r}_n &:= \left(\mathbf{I}-\omega_j\mathbf{A}\right)\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

compare with the so-called “index functions” (Yeung/Boley, 2005).

Removing $\mathbf{v}_{n-1}$ from the recurrence we obtain the generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ. \tag{8}$$

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ \mathbf{r}_n &:= (\mathbf{I} - \omega_j\mathbf{A})\,\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

compare with the so-called "index functions" (Yeung/Boley, 2005).

Removing $\mathbf{v}_{n-1}$ from the recurrence we obtain the generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ. \tag{8}$$

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= (1-\gamma_s^{(n)})\,\mathbf{r}_{n-1} + \textstyle\sum_{\ell=1}^{s-1}(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)})\,\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ 1\cdot\mathbf{r}_n &:= (\mathbf{I}-\omega_j\mathbf{A})\,\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

compare with the so-called "index functions" (Yeung/Boley, 2005).

Removing $\mathbf{v}_{n-1}$ from the recurrence we obtain the generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ. \tag{8}$$

IDR: The underlying Hessenberg decomposition

The IDR recurrences of IDR(s)ORes can be summarized by

$$\begin{aligned}\mathbf{v}_{n-1} &:= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_n = \mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \\ &= \left(1 - \gamma_s^{(n)}\right)\mathbf{r}_{n-1} + \sum_{\ell=1}^{s-1}\left(\gamma_{s-\ell+1}^{(n)} - \gamma_{s-\ell}^{(n)}\right)\mathbf{r}_{n-\ell-1} + \gamma_1^{(n)}\,\mathbf{r}_{n-s-1}\,, \\ 1\cdot\mathbf{r}_n &:= \left(\mathbf{I} - \omega_j\mathbf{A}\right)\mathbf{v}_{n-1}\,.\end{aligned} \tag{7}$$

Here, $n > s$, and the index of the scalar ω_j is defined by

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

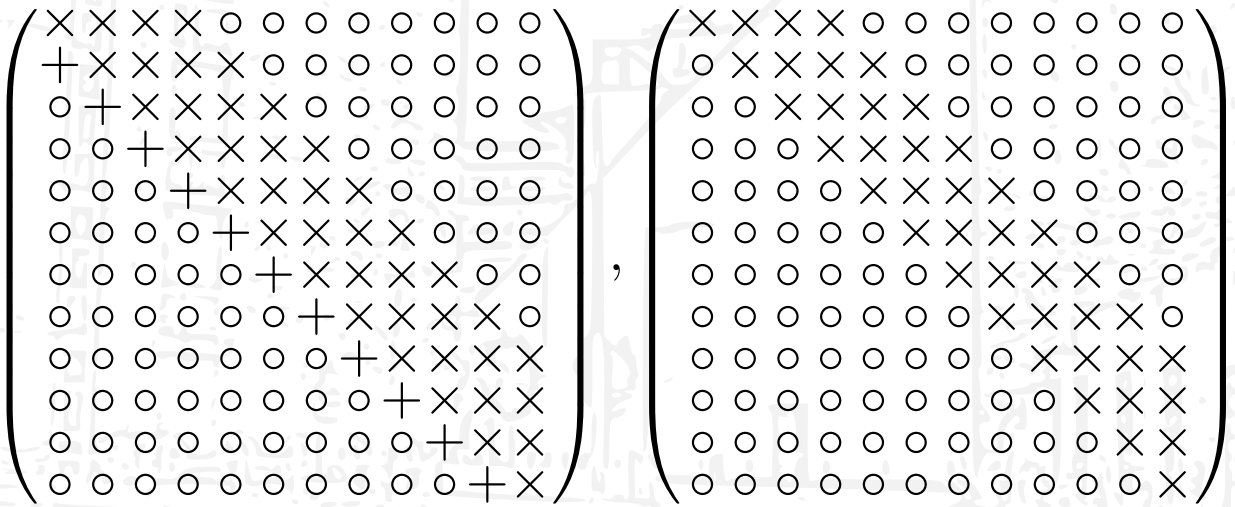
compare with the so-called “index functions” (Yeung/Boley, 2005).

Removing $\mathbf{v}_{n-1}$ from the recurrence we obtain the generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^{\circ}. \tag{8}$$

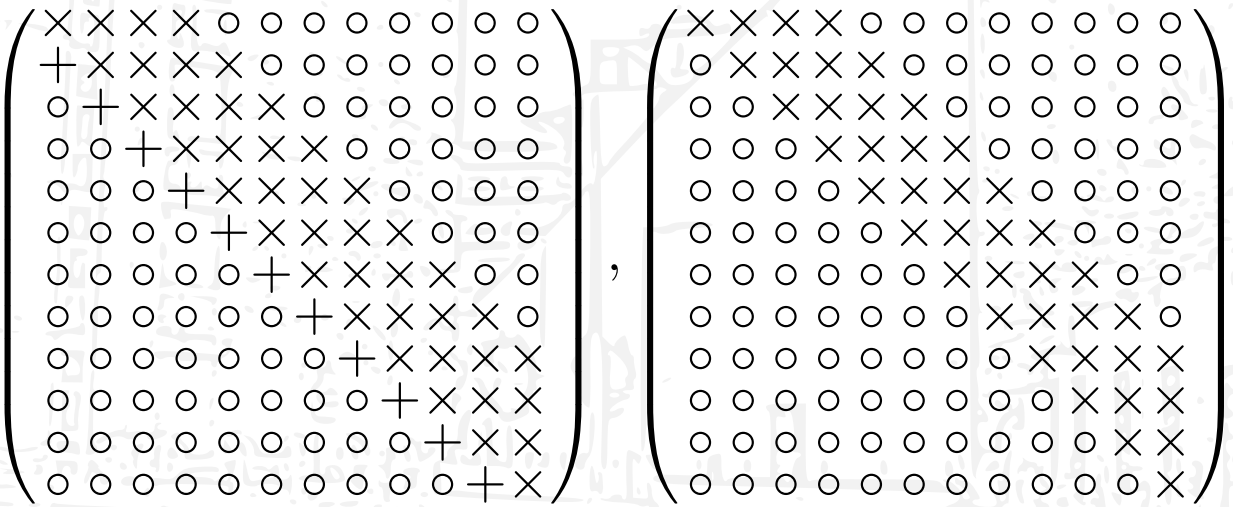
IDR: Sonneveld pencil and Sonneveld matrix

The IDR(s)ORes pencil, the so-called Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n \mathbf{D}_\omega^{(n)})$, can be depicted by



IDR: Sonneveld pencil and Sonneveld matrix

The IDR(s)ORes pencil, the so-called Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega^{(n)})$, can be depicted by



The upper triangular matrix $\mathbf{Y}_n\mathbf{D}_\omega^{(n)}$ could be inverted, which results in the Sonneveld matrix, a full unreduced Hessenberg matrix.

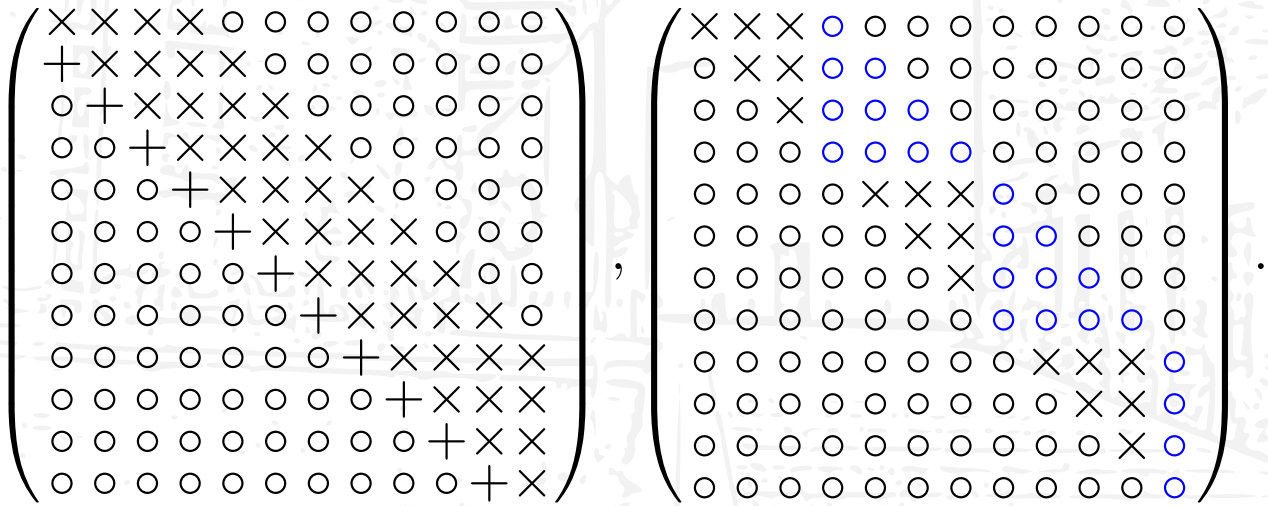
Understanding IDR: Purification

We know the eigenvalues $\approx$ roots of kernel polynomials $1/\omega_j$. We are only interested in the other eigenvalues.

Understanding IDR: Purification

We know the eigenvalues $\approx$ roots of kernel polynomials $1/\omega_j$. We are only interested in the other eigenvalues.

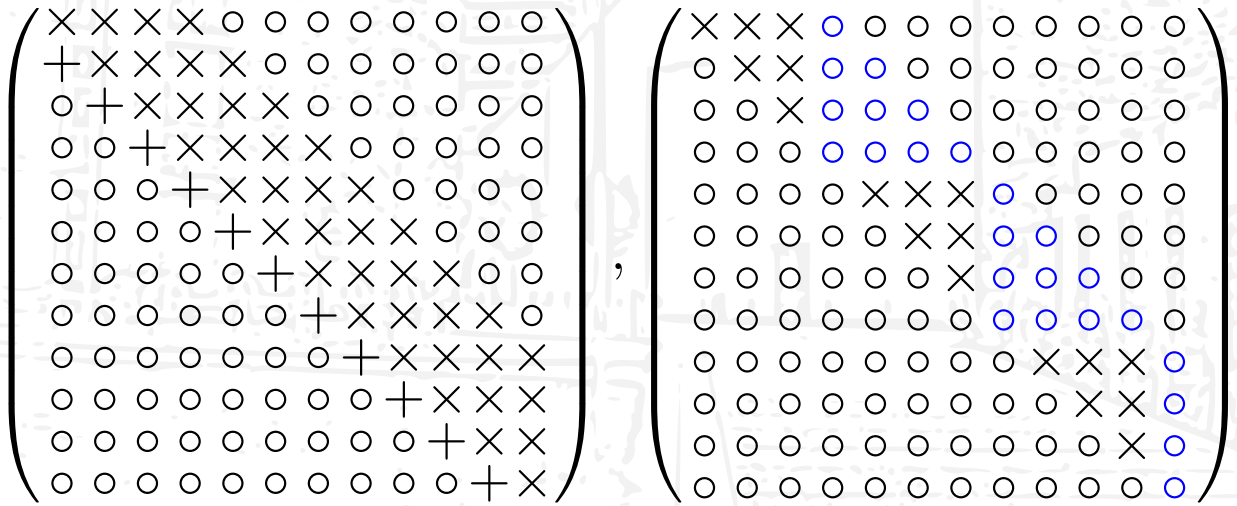
The purified IDR(s)ORes pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, that has only the remaining eigenvalues and some infinite ones as eigenvalues, can be depicted by



Understanding IDR: Purification

We know the eigenvalues $\approx$ roots of kernel polynomials $1/\omega_j$. We are only interested in the other eigenvalues.

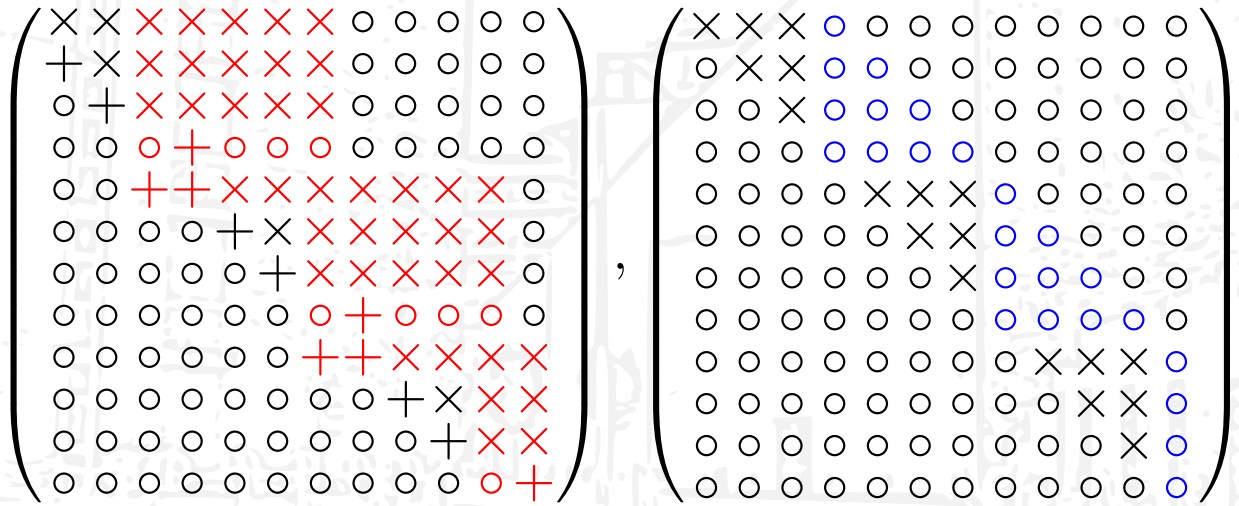
The purified IDR(s)ORes pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, that has only the remaining eigenvalues and some infinite ones as eigenvalues, can be depicted by



We get rid of the infinite eigenvalues using a change of basis (Gauß/Schur).

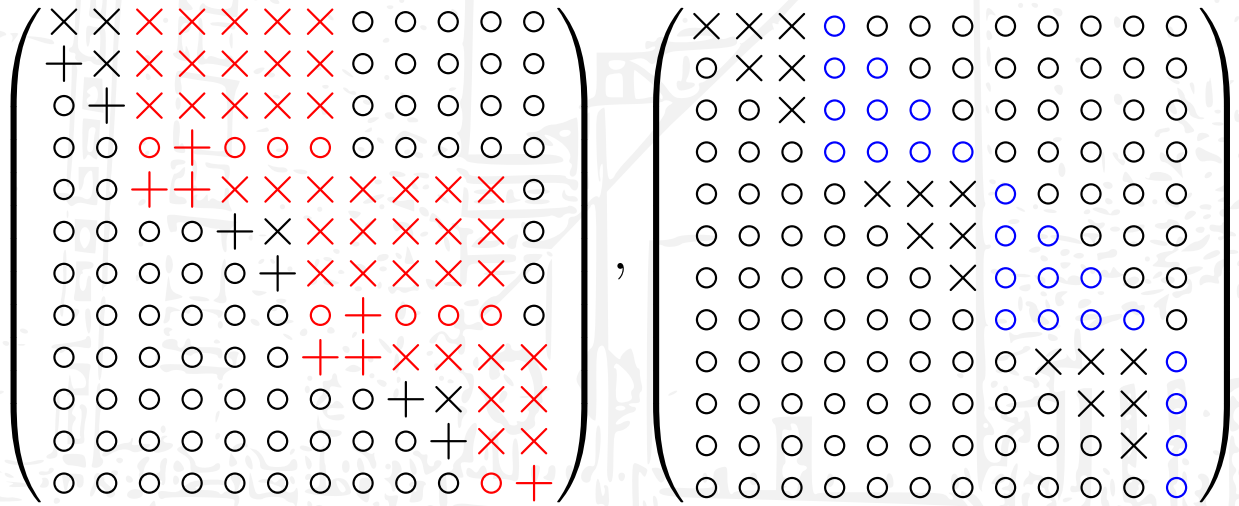
Understanding IDR: Gaussian elimination

The deflated purified IDR(s)ORes pencil, after the elimination step $(\mathbf{Y}_n^\circ \mathbf{G}_n, \mathbf{U}_n \mathbf{D}_\omega^{(n)})$, can be depicted by



Understanding IDR: Gaussian elimination

The deflated purified IDR(s)ORes pencil, after the elimination step $(\mathbf{Y}_n^\circ \mathbf{G}_n, \mathbf{U}_n \mathbf{D}_\omega^{(n)})$, can be depicted by



Using Laplace expansion of the determinant of $z\mathbf{U}_n\mathbf{D}_\omega^{(n)} - \mathbf{Y}_n^\circ\mathbf{G}_n$ we can get rid of the trivial constant factors corresponding to infinite eigenvalues. This amounts to a deflation.

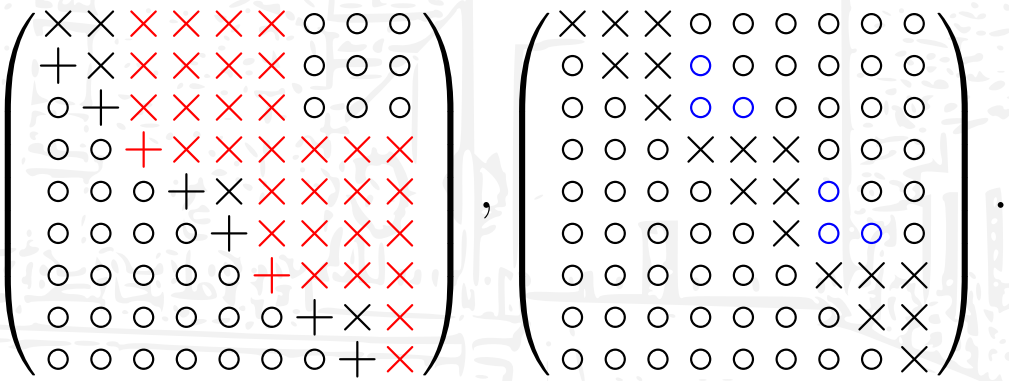
Understanding IDR: Deflation

Let D denote an deflation operator that removes every $s+1$th column and row from the matrix the operator is applied to.

Understanding IDR: Deflation

Let D denote an deflation operator that removes every $s+1$th column and row from the matrix the operator is applied to.

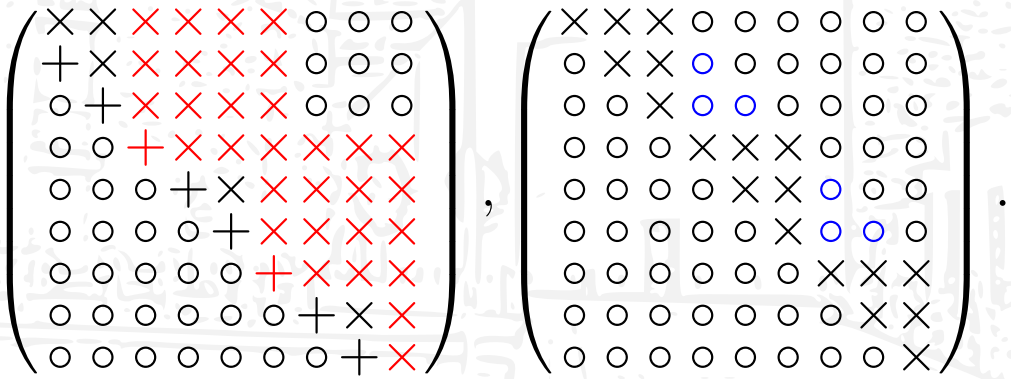
The deflated purified IDR(s)ORes pencil, after the deflation step $(D(\mathbf{Y}_n^\circ \mathbf{G}_n), D(\mathbf{U}_n \mathbf{D}_\omega^{(n)}))$, can be depicted by



Understanding IDR: Deflation

Let D denote an deflation operator that removes every $s+1$th column and row from the matrix the operator is applied to.

The deflated purified IDR(s)ORes pencil, after the deflation step $(D(\mathbf{Y}_n^\circ \mathbf{G}_n), D(\mathbf{U}_n \mathbf{D}_\omega^{(n)}))$, can be depicted by



The block-diagonal matrix $D(\mathbf{U}_n \mathbf{D}_\omega^{(n)})$ has invertible upper triangular blocks and can be inverted to expose the underlying Lanczos process.

IDR: a Lanczos process with multiple left-hand sides

Inverting the block-diagonal matrix $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix

$$\mathbf{L}_n := D(\mathbf{Y}_n^\circ\mathbf{G}_n)\cdot D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))^{-1} = \begin{pmatrix} \times & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ + & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & + & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & \circ & + & \times & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & + & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & + & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & + & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & + & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & \circ & + & \times \end{pmatrix}.$$

IDR: a Lanczos process with multiple left-hand sides

Inverting the block-diagonal matrix $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix

$$\mathbf{L}_n := D(\mathbf{Y}_n^\circ\mathbf{G}_n)\cdot D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))^{-1} = \begin{pmatrix} \times & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ + & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & + & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & \circ & + & \times & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & + & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & + & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & + & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & + & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & \circ & + & \times \end{pmatrix}.$$

This is the matrix of the underlying BiORes$(s, 1)$ process.

IDR: a Lanczos process with multiple left-hand sides

Inverting the block-diagonal matrix $D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix

$$\mathbf{L}_n := D(\mathbf{Y}_n^\circ\mathbf{G}_n)\cdot D(\mathbf{U}_n\mathbf{D}_\omega^{(n)}))^{-1} = \begin{pmatrix} \times & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ + & \times & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & + & \times & \times & \times & \times & \circ & \circ & \circ \\ \circ & \circ & + & \times & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & + & \times & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & + & \times & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & + & \times & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & + & \times & \times \\ \circ & \circ & \circ & \circ & \circ & \circ & \circ & + & \times \end{pmatrix}.$$

This is the matrix of the underlying BiORes$(s,1)$ process.

This matrix (in the extended version) satisfies

$$\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{L}}_n,$$

where the reduced residuals $\mathbf{q}_{js+k}$, $k = 0,\ldots,s-1$, $j = 0,1,\ldots$, with $\Omega_0(z) \equiv 1$ and $\Omega_j(z) = \prod_{k=1}^{j}(1-\omega_k z)$ are given by

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

IDR: a Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

and every $\mathbf{v}_{j(s+1)+k-1}$ is orthogonal to $\mathbf{P}$.

IDR: a Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

and every $\mathbf{v}_{j(s+1)+k-1}$ is orthogonal to $\mathbf{P}$. Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathbf{P}$.

IDR: a Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

and every $\mathbf{v}_{j(s+1)+k-1}$ is orthogonal to $\mathbf{P}$. Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathbf{P}$.

Using induction (Sleijpen et al., 2008) one can prove that $\mathbf{q}_{js+k} \perp \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P})$; thus, this is a two-sided Lanczos process with s left and one right starting vectors.

IDR: a Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

and every $\mathbf{v}_{j(s+1)+k-1}$ is orthogonal to $\mathbf{P}$. Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathbf{P}$.

Using induction (Sleijpen et al., 2008) one can prove that $\mathbf{q}_{js+k} \perp \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P})$; thus, this is a two-sided Lanczos process with s left and one right starting vectors.

This can more easily be proven using the representations ($\mathcal{S} := \mathbf{P}^{\perp}$)

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

$$\mathcal{G}_j = \bigcap_{k=0}^{j-1} \Omega_k(\mathbf{A})^{-1}\Omega_j(\mathbf{A})(\mathcal{S}) = \left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}}\Omega_k(\mathbf{A})^{\mathsf{H}}\{\mathbf{P}\}\right)^{\perp}$$

$$= \left(\Omega_j(\mathbf{A})^{-\mathsf{H}}\mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P})\right)^{\perp} = \Omega_j(\mathbf{A})\left(\mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P})\right)^{\perp}$$

of the Sonneveld spaces.

IDR: a Lanczos process with multiple left-hand sides

This has to be compared with Theorem 4.2 in (Sleijpen et al., 2008) and with Theorem 4.1 in (Simoncini and Szyld, 2009) (similar result; slightly different method of proof).

IDR: a Lanczos process with multiple left-hand sides

This has to be compared with Theorem 4.2 in (Sleijpen et al., 2008) and with Theorem 4.1 in (Simoncini and Szyld, 2009) (similar result; slightly different method of proof).

The first equality

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

IDR: a Lanczos process with multiple left-hand sides

This has to be compared with Theorem 4.2 in (Sleijpen et al., 2008) and with Theorem 4.1 in (Simoncini and Szyld, 2009) (similar result; slightly different method of proof).

The first equality

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

follows from the observations that

- the first $s+1$ residuals obviously are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,
- the next $s+1$ residuals (or any other vectors in $\mathcal{G}_1$) are in the $\mathbf{I} - \omega_1\mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,
- the last $s+1$ residuals are in the $\mathbf{I} - \omega_j\mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,
- the last residuals are $\mathbf{I} - \omega_j\mathbf{A}$ images of linear combinations of previously obtained images $(\mathbf{I} - \omega_{j-1}\mathbf{A}) \cdots (\mathbf{I} - \omega_k\mathbf{A})$ of $\mathcal{S} = \mathbf{P}^{\perp}$.

IDR: a Lanczos process with multiple left-hand sides

The second equality

$$\bigcap_{k=0}^{j-1} \Omega_k(\mathbf{A})^{-1}\Omega_j(\mathbf{A})(\mathcal{S}) = \left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}}\Omega_k(\mathbf{A})^{\mathsf{H}}\{\mathbf{P}\}\right)^{\perp}$$

IDR: a Lanczos process with multiple left-hand sides

The second equality

$$\bigcap_{k=0}^{j-1} \Omega_k(\mathbf{A})^{-1}\Omega_j(\mathbf{A})(\mathcal{S}) = \left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}}\Omega_k(\mathbf{A})^{\mathsf{H}}\{\mathbf{P}\}\right)^{\perp}$$

is based on

$$\mathbf{B}\mathbf{P}^{\perp} = (\mathbf{B}^{-\mathsf{H}}\mathbf{P})^{\perp}$$

and

$$\mathcal{U}^{\perp} \cap \mathcal{V}^{\perp} = (\mathcal{U} \cup \mathcal{V})^{\perp} = (\mathcal{U} + \mathcal{V})^{\perp}.$$

IDR: a Lanczos process with multiple left-hand sides

The second equality

$$\bigcap_{k=0}^{j-1} \Omega_k(\mathbf{A})^{-1}\Omega_j(\mathbf{A})(\mathcal{S}) = \left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}}\Omega_k(\mathbf{A})^{\mathsf{H}}\{\mathbf{P}\}\right)^{\perp}$$

is based on

$$\mathbf{B}\mathbf{P}^{\perp} = (\mathbf{B}^{-\mathsf{H}}\mathbf{P})^{\perp}$$

and

$$\mathcal{U}^{\perp} \cap \mathcal{V}^{\perp} = (\mathcal{U} \cup \mathcal{V})^{\perp} = (\mathcal{U} + \mathcal{V})^{\perp}.$$

The second relations are basic linear algebra. The first relation follows from

$$\mathbf{P}^{\perp} = \left\{\mathbf{v} \in \mathbb{C}^n \mid \mathbf{P}^{\mathsf{H}}\mathbf{v} = \mathbf{o}_n\right\} \quad \Rightarrow \quad \mathbf{B}\mathbf{P}^{\perp} = \left\{\mathbf{B}\mathbf{v} \in \mathbb{C}^n \mid \mathbf{P}^{\mathsf{H}}\mathbf{v} = \mathbf{o}_n\right\},$$

since, for invertible $\mathbf{B}$,

$$\mathbf{y} \in \mathbf{B}\mathbf{P}^{\perp} \;\Leftrightarrow\; \left\{\mathbf{y} = \mathbf{B}\mathbf{v} \wedge \mathbf{P}^{\mathsf{H}}\mathbf{v} = \mathbf{o}_n\right\} \;\Leftrightarrow\; \mathbf{P}^{\mathsf{H}}\mathbf{v} = \mathbf{P}^{\mathsf{H}}\mathbf{B}^{-1}\mathbf{y} = (\mathbf{B}^{-\mathsf{H}}\mathbf{P})^{\mathsf{H}}\mathbf{y} = \mathbf{o}_n.$$

IDR: a Lanczos process with multiple left-hand sides

The third and fourth equality

$$\left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}} \Omega_k(\mathbf{A})^{\mathsf{H}} \{\mathbf{P}\} \right)^{\perp} = \left(\Omega_j(\mathbf{A})^{-\mathsf{H}} \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \right)^{\perp}$$

$$= \Omega_j(\mathbf{A}) \left(\mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \right)^{\perp}$$

IDR: a Lanczos process with multiple left-hand sides

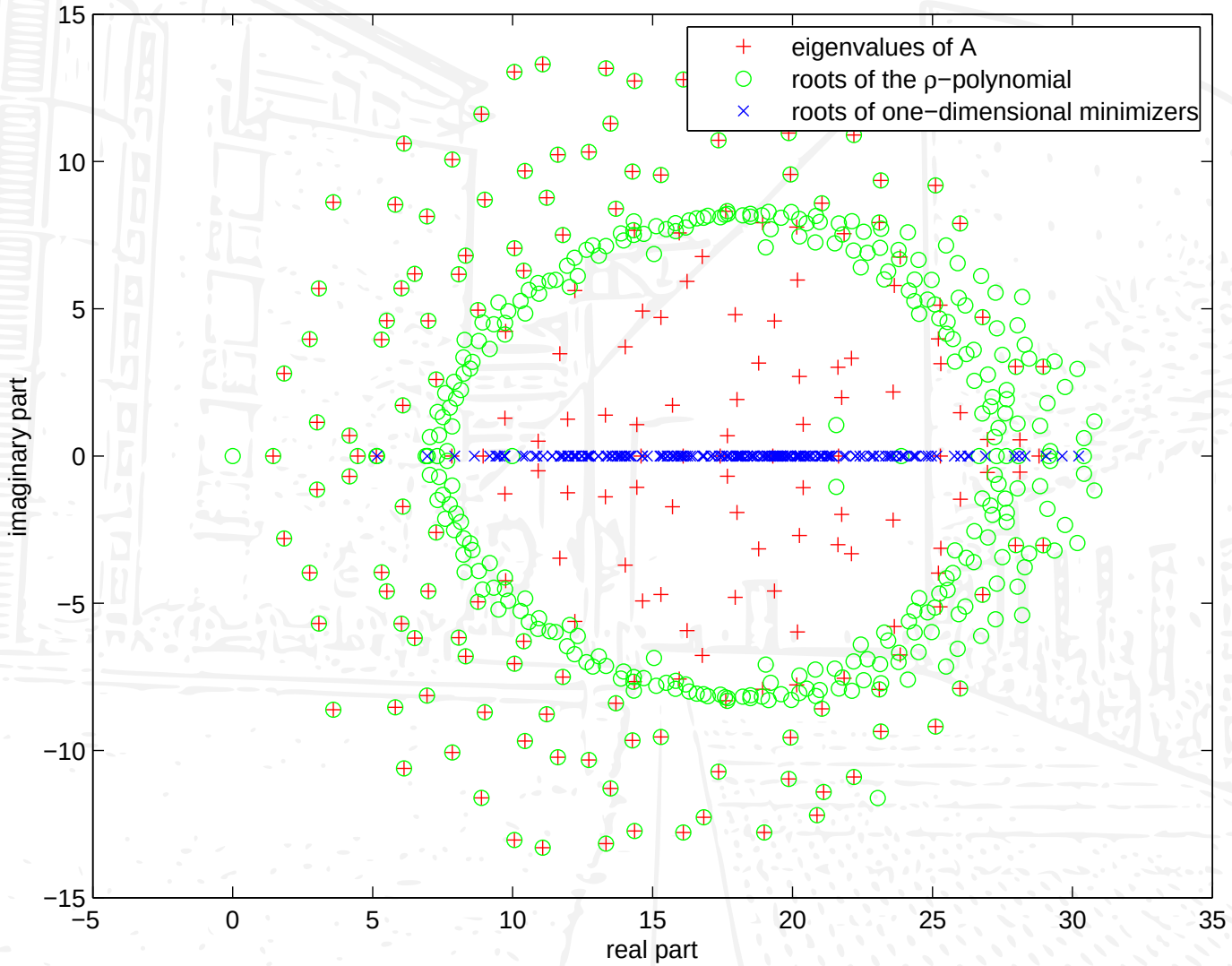
The third and fourth equality

$$\left(\mathop{+}_{k=0}^{j-1} \Omega_j(\mathbf{A})^{-\mathsf{H}} \Omega_k(\mathbf{A})^{\mathsf{H}} \{\mathbf{P}\} \right)^{\perp} = \left(\Omega_j(\mathbf{A})^{-\mathsf{H}} \mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \right)^{\perp}$$
$$= \Omega_j(\mathbf{A}) \left(\mathcal{K}_j(\mathbf{A}^{\mathsf{H}}, \mathbf{P}) \right)^{\perp}$$

are satisfied

- since the polynomials $\Omega_k(\mathbf{A})$, $0 \leqslant k < j$ form a basis of the space of polynomials of degree less j, and
- by the property proved on the last slide, respectively.

Understanding IDR: 600 steps for $s = 2$



IDR: recent developments

The analysis of IDR carried out by Gutknecht and Z. (to be finished early 2010) as a byproduct enables the computation of approximate eigenvalues and eigenvectors. We will make the implementation in Matlab of IDREig based on IDR(s)ORes publicly available after the report is finished.

IDR: recent developments

The analysis of IDR carried out by Gutknecht and Z. (to be finished early 2010) as a byproduct enables the computation of approximate eigenvalues and eigenvectors. We will make the implementation in Matlab of IDREig based on IDR(s)ORes publicly available after the report is finished.

The analysis should carry over to the new developments in IDR methods. The recent developments in the IDR family and some current trends are summarized on the next slides.

IDR: recent developments

The analysis of IDR carried out by Gutknecht and Z. (to be finished early 2010) as a byproduct enables the computation of approximate eigenvalues and eigenvectors. We will make the implementation in Matlab of IDREig based on IDR(s)ORes publicly available after the report is finished.

The analysis should carry over to the new developments in IDR methods. The recent developments in the IDR family and some current trends are summarized on the next slides.

These can be grouped into

- incorporation of higher degree polynomials: like BiCGStab2, BiCGStab(ℓ), GCGS (CGS2, shifted CGS) and GPBiCG,
- a better understanding of the IDR Theorem: construction of better IDR algorithms,
- choices of ω_j: eigenvalue estimates, classical splitting methods,
- changing the shadow space: sparsification, (non-)overlapping Schwarz.

IDR: recent developments

The residuals computed first in a complete cycle are uniquely defined. Based on the analysis of a possible breakdown of IDR(s), Sonneveld and van Gijzen came up with their new implementation IDR(s)BiO (van Gijzen and Sonneveld, 2008) of the IDR Theorem.

IDR: recent developments

The residuals computed first in a complete cycle are uniquely defined. Based on the analysis of a possible breakdown of IDR(s), Sonneveld and van Gijzen came up with their new implementation IDR(s)BiO (van Gijzen and Sonneveld, 2008) of the IDR Theorem.

Here, they use basis vectors $\mathbf{g}_{-1}, \ldots, \mathbf{g}_{-s} \in \mathcal{G}_j$, which are not simply residual differences but linear combinations.

IDR: recent developments

The residuals computed first in a complete cycle are uniquely defined. Based on the analysis of a possible breakdown of IDR(s), Sonneveld and van Gijzen came up with their new implementation IDR(s)BiO (van Gijzen and Sonneveld, 2008) of the IDR Theorem.

Here, they use basis vectors $\mathbf{g}_{-1}, \ldots, \mathbf{g}_{-s} \in \mathcal{G}_j$, which are not simply residual differences but linear combinations.

The new vectors $\mathbf{v}_n$ and $\mathbf{r}_{n+1}$ are in this general setting given by the updates

$$\mathbf{v}_n = \mathbf{r}_n - \sum_{i=1}^{s} \mathbf{g}_{n-i}\gamma_i =: \mathbf{r}_n - \mathbf{G}_n\mathbf{c}_n, \quad \text{and thus,}$$

$$\mathbf{r}_{n+1} = (\mathbf{I} - \omega\mathbf{A})\mathbf{v}_n = \mathbf{r}_n - \omega\mathbf{A}\mathbf{v}_n - \sum_{i=1}^{s} \mathbf{g}_{n-i}\gamma_i,$$

where $\mathbf{c}_n$ is determined such that $\mathbf{P}^{\mathsf{H}}\mathbf{v}_n = \mathbf{o}$.

IDR: recent developments

As basis vectors $\mathbf{g}_{n+k}$ they compute as before updates of residuals which are in $\mathcal{G}_j \cap \mathcal{S}$. The update vectors $\mathbf{u}_n$ of the iterates $\mathbf{x}_n$ are stored in $\mathbf{U}_n$, "prototypes" of these are given by

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \widetilde{\mathbf{u}}_n = \mathbf{x}_n + \omega \mathbf{v}_n - \sum_{i=1}^{s} \mathbf{u}_{n-i}\gamma_i =: \mathbf{x}_n + \omega \mathbf{v}_n - \mathbf{U}_n \mathbf{c}_n.$$

IDR: recent developments

As basis vectors $\mathbf{g}_{n+k}$ they compute as before updates of residuals which are in $\mathcal{G}_j \cap \mathcal{S}$. The update vectors $\mathbf{u}_n$ of the iterates $\mathbf{x}_n$ are stored in $\mathbf{U}_n$, "prototypes" of these are given by

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \widetilde{\mathbf{u}}_n = \mathbf{x}_n + \omega \mathbf{v}_n - \sum_{i=1}^{s} \mathbf{u}_{n-i}\gamma_i =: \mathbf{x}_n + \omega \mathbf{v}_n - \mathbf{U}_n \mathbf{c}_n.$$

If we had computed all these, we could use a basis transformation to ensure a simple structure of the update formulas and the systems to be solved. In order to not destroy the nested structure and to obtain an explicit formula, these transformations should be triangular, e.g., QR- or LR-like.

IDR: recent developments

As basis vectors $\mathbf{g}_{n+k}$ they compute as before updates of residuals which are in $\mathcal{G}_j \cap \mathcal{S}$. The update vectors $\mathbf{u}_n$ of the iterates $\mathbf{x}_n$ are stored in $\mathbf{U}_n$, "prototypes" of these are given by

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \widetilde{\mathbf{u}}_n = \mathbf{x}_n + \omega \mathbf{v}_n - \sum_{i=1}^{s} \mathbf{u}_{n-i}\gamma_i =: \mathbf{x}_n + \omega \mathbf{v}_n - \mathbf{U}_n \mathbf{c}_n.$$

If we had computed all these, we could use a basis transformation to ensure a simple structure of the update formulas and the systems to be solved. In order to not destroy the nested structure and to obtain an explicit formula, these transformations should be triangular, e.g., QR- or LR-like.

Sonneveld and van Gijzen impose the bi-orthogonality conditions

$$\mathbf{g}_{n+k} \perp \mathbf{p}_1, \ldots, \mathbf{p}_{k-1},$$
$$\mathbf{r}_{n+k+1} \perp \mathbf{p}_1, \ldots, \mathbf{p}_k.$$

IDR: recent developments

The triangular basis transformations result in the general update formulas

$$\mathbf{v}_{n+k} = \mathbf{r}_{n+k} - \mathbf{G}_{n+k}\mathbf{c}_{n+k}, \quad (\mathbf{c}_{n+k} \text{ is determined by orthogonality to } \mathbf{P})$$

$$\widetilde{\mathbf{u}}_{n+k} = \mathbf{U}_{n+k}\mathbf{c}_{n+k} + \omega\mathbf{v}_{n+k}, \qquad \mathbf{u}_{n+k} = \widetilde{\mathbf{u}}_{n+k} - \sum_{i=1}^{k-1} \mathbf{u}_{n+i}\alpha_i,$$

$$\widetilde{\mathbf{g}}_{n+k} = \mathbf{A}\widetilde{\mathbf{u}}_{n+k}, \qquad \mathbf{g}_{n+k} = \widetilde{\mathbf{g}}_{n+k} - \sum_{i=1}^{k-1} \mathbf{g}_{n+i}\alpha_i = \mathbf{A}\mathbf{u}_{n+k},$$

$$\mathbf{r}_{n+k+1} = \mathbf{r}_{n+k} - \sum_{i=1}^{k} \mathbf{g}_{n+i}\beta_i, \qquad \mathbf{x}_{n+k+1} = \mathbf{x}_{n+k} + \sum_{i=1}^{k} \mathbf{u}_{n+i}\beta_i.$$

IDR: recent developments

The triangular basis transformations result in the general update formulas

$$\mathbf{v}_{n+k} = \mathbf{r}_{n+k} - \mathbf{G}_{n+k}\mathbf{c}_{n+k}, \quad (\mathbf{c}_{n+k} \text{ is determined by orthogonality to } \mathbf{P})$$

$$\widetilde{\mathbf{u}}_{n+k} = \mathbf{U}_{n+k}\mathbf{c}_{n+k} + \omega\mathbf{v}_{n+k}, \qquad \mathbf{u}_{n+k} = \widetilde{\mathbf{u}}_{n+k} - \sum_{i=1}^{k-1} \mathbf{u}_{n+i}\alpha_i,$$

$$\widetilde{\mathbf{g}}_{n+k} = \mathbf{A}\widetilde{\mathbf{u}}_{n+k}, \qquad \mathbf{g}_{n+k} = \widetilde{\mathbf{g}}_{n+k} - \sum_{i=1}^{k-1} \mathbf{g}_{n+i}\alpha_i = \mathbf{A}\mathbf{u}_{n+k},$$

$$\mathbf{r}_{n+k+1} = \mathbf{r}_{n+k} - \sum_{i=1}^{k} \mathbf{g}_{n+i}\beta_i, \qquad \mathbf{x}_{n+k+1} = \mathbf{x}_{n+k} + \sum_{i=1}^{k} \mathbf{u}_{n+i}\beta_i.$$

In the approach by Sonneveld and van Gijzen, the vector $\widetilde{\mathbf{g}}_{n+k}$ is orthogonalized against the vectors $\mathbf{p}_i$, $1 \leqslant i < k$ using the α_i, $1 \leqslant i < k$, the vector $\mathbf{r}_{n+k+1}$ is orthogonal to $\mathbf{p}_1, \ldots, \mathbf{p}_k$ by choice of the β_i, $1 \leqslant i \leqslant k$.

IDR: recent developments

This (modified) Gram-Schmidt-like approach results in the kth inner step in nested systems of the form (indices omitted)

$$\begin{pmatrix} \mu_{11} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \mu_{k-1,1} & \cdots & \mu_{k-1,k-1} & 0 & \cdots & 0 \\ \mu_{k,1} & \cdots & \mu_{k,k-1} & \mu_{kk} & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & 0 \\ \mu_{s1} & \cdots & \mu_{s,k-1} & \mu_{sk} & \cdots & \mu_{ss} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \gamma_k \\ \vdots \\ \gamma_s \end{pmatrix} = \underbrace{(\mathbf{P}^{\mathsf{H}}\mathbf{G})}_{=:\mathbf{M}} \mathbf{c} = \mathbf{P}^{\mathsf{H}}\mathbf{r} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \phi_k \\ \vdots \\ \phi_s \end{pmatrix},$$

IDR: recent developments

This (modified) Gram-Schmidt-like approach results in the kth inner step in nested systems of the form (indices omitted)

$$\begin{pmatrix} \mu_{11} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \mu_{k-1,1} & \cdots & \mu_{k-1,k-1} & 0 & \cdots & 0 \\ \mu_{k,1} & \cdots & \mu_{k,k-1} & \mu_{kk} & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & 0 \\ \mu_{s1} & \cdots & \mu_{s,k-1} & \mu_{sk} & \cdots & \mu_{ss} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \gamma_k \\ \vdots \\ \gamma_s \end{pmatrix} = \underbrace{(\mathbf{P}^{\mathsf{H}}\mathbf{G})}_{=:\mathbf{M}}\mathbf{c} = \mathbf{P}^{\mathsf{H}}\mathbf{r} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \phi_k \\ \vdots \\ \phi_s \end{pmatrix},$$

i.e., $\mathbf{r} \perp \mathbf{p}_1, \ldots, \mathbf{p}_{k-1}$ and $\mathbf{g}_{-k} \perp \mathbf{p}_1, \ldots, \mathbf{p}_{k-1}$, and thus we only have to solve

$$\begin{pmatrix} \mu_{kk} & & \\ \vdots & \ddots & \\ \mu_{sk} & \cdots & \mu_{ss} \end{pmatrix} \begin{pmatrix} \gamma_k \\ \vdots \\ \gamma_s \end{pmatrix} = \begin{pmatrix} \phi_k \\ \vdots \\ \phi_s \end{pmatrix}.$$

IDR: recent developments

This (modified) Gram-Schmidt-like approach results in the kth inner step in nested systems of the form (indices omitted)

$$\begin{pmatrix} \mu_{11} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \mu_{k-1,1} & \cdots & \mu_{k-1,k-1} & 0 & \cdots & 0 \\ \mu_{k,1} & \cdots & \mu_{k,k-1} & \mu_{kk} & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & 0 \\ \mu_{s1} & \cdots & \mu_{s,k-1} & \mu_{sk} & \cdots & \mu_{ss} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \gamma_k \\ \vdots \\ \gamma_s \end{pmatrix} = \underbrace{(\mathbf{P}^{\mathsf{H}}\mathbf{G})}_{=:\mathbf{M}} \mathbf{c} = \mathbf{P}^{\mathsf{H}}\mathbf{r} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \phi_k \\ \vdots \\ \phi_s \end{pmatrix},$$

i.e., $\mathbf{r} \perp \mathbf{p}_1, \ldots, \mathbf{p}_{k-1}$ and $\mathbf{g}_{-k} \perp \mathbf{p}_1, \ldots, \mathbf{p}_{k-1}$, and thus we only have to solve

$$\begin{pmatrix} \mu_{kk} & & \\ \vdots & \ddots & \\ \mu_{sk} & \cdots & \mu_{ss} \end{pmatrix} \begin{pmatrix} \gamma_k \\ \vdots \\ \gamma_s \end{pmatrix} = \begin{pmatrix} \phi_k \\ \vdots \\ \phi_s \end{pmatrix}.$$

Initially, $\mathbf{M} := \mathbf{I}_s$. The resulting IDR($s$)BiO is cheaper and seems more stable.

IDR: recent developments

Recently, the relations between IDR(s) and BiCGStab(ℓ) and combinations of both methods have been investigated.

- In (Sleijpen et al., 2008) the authors derive different implementations of ML(k)BiCGStab-like algorithms.

IDR: recent developments

Recently, the relations between IDR(s) and BiCGStab(ℓ) and combinations of both methods have been investigated.

- In (Sleijpen et al., 2008) the authors derive different implementations of ML(k)BiCGStab-like algorithms.
- In (Sleijpen and van Gijzen, 2009) the authors combine the IDR philosophy with higher degree stabilization polynomials. The resulting method is named IDR(s)Stab(ℓ). The approach is comparable to the one resulting in BiCGStab(ℓ).

IDR: recent developments

Recently, the relations between IDR(s) and BiCGStab(ℓ) and combinations of both methods have been investigated.

- In (Sleijpen et al., 2008) the authors derive different implementations of ML(k)BiCGStab-like algorithms.
- In (Sleijpen and van Gijzen, 2009) the authors combine the IDR philosophy with higher degree stabilization polynomials. The resulting method is named IDR(s)Stab(ℓ). The approach is comparable to the one resulting in BiCGStab(ℓ).
- In (Tanio and Sugihara, 2009) the authors derive the algorithm GBiCGStab(s,L), which is similar to IDR(s)Stab(ℓ). In their own words: “Our algorithm is to theirs what the Gauss-Seidel iteration is to the Jacobi iteration.” A predecessor of GBiCGStab(s,L) seems to be the method called GIDR(s,L) in (Tanio and Sugihara, 2008).

IDR: recent developments

Recently, the relations between IDR(s) and BiCGStab(ℓ) and combinations of both methods have been investigated.

- In (Sleijpen et al., 2008) the authors derive different implementations of ML(k)BiCGStab-like algorithms.
- In (Sleijpen and van Gijzen, 2009) the authors combine the IDR philosophy with higher degree stabilization polynomials. The resulting method is named IDR(s)Stab(ℓ). The approach is comparable to the one resulting in BiCGStab(ℓ).
- In (Tanio and Sugihara, 2009) the authors derive the algorithm GBiCGStab(s,L), which is similar to IDR(s)Stab(ℓ). In their own words: "Our algorithm is to theirs what the Gauss-Seidel iteration is to the Jacobi iteration." A predecessor of GBiCGStab(s,L) seems to be the method called GIDR(s,L) in (Tanio and Sugihara, 2008).
- In (Sleijpen and Abe, 2010) the ideas behind BiCGStab2 by Gutknecht and GPBiCG by Zhang are considered. The resulting algorithms IDR(s)Stab2 and GIDR(s) seem to be less efficient when compared with IDR(s)Stab(ℓ).

IDR: recent developments

The relation of IDR to Petrov-Galërkin with a rational Krylov space motived the method IDR-Ritz (Simoncini and Szyld, 2009).

IDR: recent developments

The relation of IDR to Petrov-Galërkin with a rational Krylov space motived the method IDR-Ritz (Simoncini and Szyld, 2009).

Another, simpler motivation is that the residual polynomials should be designed to dampen the spectrum. Using the residual polynomial representation of IDR we could choose the $1/\omega_j$ close but not equal to eigenvalues, at least we should choose them in the field of values of $\mathbf{A}$.

IDR: recent developments

The relation of IDR to Petrov-Galërkin with a rational Krylov space motived the method IDR-Ritz (Simoncini and Szyld, 2009).

Another, simpler motivation is that the residual polynomials should be designed to dampen the spectrum. Using the residual polynomial representation of IDR we could choose the $1/\omega_j$ close but not equal to eigenvalues, at least we should choose them in the field of values of $\mathbf{A}$.

The minimization used in IDR(s)ORes and IDR(s)BiO results in values ω_j which are in the field of values of $\mathbf{A}^{-\mathsf{H}}$, thus Simoncini and Szyld suggest to use a few steps of Arnoldi to compute some Ritz values, which are then used in some ordering as $1/\omega_j$ values.

IDR: recent developments

The relation of IDR to Petrov-Galërkin with a rational Krylov space motived the method IDR-Ritz (Simoncini and Szyld, 2009).

Another, simpler motivation is that the residual polynomials should be designed to dampen the spectrum. Using the residual polynomial representation of IDR we could choose the $1/\omega_j$ close but not equal to eigenvalues, at least we should choose them in the field of values of $\mathbf{A}$.

The minimization used in IDR(s)ORes and IDR(s)BiO results in values ω_j which are in the field of values of $\mathbf{A}^{-\mathsf{H}}$, thus Simoncini and Szyld suggest to use a few steps of Arnoldi to compute some Ritz values, which are then used in some ordering as $1/\omega_j$ values.

For real nonsymmetric matrices this typically results in an algorithm based on complex arithmetic in place of real arithmetic.

IDR: recent developments

Last but not least: Certain old ideas have been reactivated. Sonneveld presented the hitherto unpublished Accelerated Gauß-Seidel (AGS) method at the Kyoto Forum on Krylov Subspace Methods in 2008.

IDR: recent developments

Last but not least: Certain old ideas have been reactivated. Sonneveld presented the hitherto unpublished Accelerated Gauß-Seidel (AGS) method at the Kyoto Forum on Krylov Subspace Methods in 2008.

Based on the algorithm in the proceedings, Seiji Fujino et al. considered the acceleration of the classical splitting methods (Jacobi, Gauß-Seidel and SOR). The resulting methods are called

- IDR(s)-Jacobi (w/o adaptive tuning),
- IDR(s)-GS,
- IDR(s)-SOR.

IDR: recent developments

Last but not least: Certain old ideas have been reactivated. Sonneveld presented the hitherto unpublished Accelerated Gauß-Seidel (AGS) method at the Kyoto Forum on Krylov Subspace Methods in 2008.

Based on the algorithm in the proceedings, Seiji Fujino et al. considered the acceleration of the classical splitting methods (Jacobi, Gauß-Seidel and SOR). The resulting methods are called

- IDR(s)-Jacobi (w/o adaptive tuning),
- IDR(s)-GS,
- IDR(s)-SOR.

These approaches result in a “tight packing” of preconditioning and Krylov subspace methods, compare with PIA. In most of these methods the ω_j are fixed by the splitting chosen.

IDR: recent developments

We mentioned the influence of a few choices of the ω_j. The influence of the choice of $\mathbf{P}$ has been analyzed only partially. We mention two aspects.

IDR: recent developments

We mentioned the influence of a few choices of the ω_j. The influence of the choice of $\mathbf{P}$ has been analyzed only partially. We mention two aspects.

In Monterey, Peter Sonneveld presented some new ideas for a convergence analysis of IDR based on stochastic. He compares a re-scaled IDR with GMRes, where the scaling is along the axis of matrix-vector-multiplies.

IDR: recent developments

We mentioned the influence of a few choices of the ω_j. The influence of the choice of $\mathbf{P}$ has been analyzed only partially. We mention two aspects.

In Monterey, Peter Sonneveld presented some new ideas for a convergence analysis of IDR based on stochastic. He compares a re-scaled IDR with GMRes, where the scaling is along the axis of matrix-vector-multiplies.

The choice of the shadow vectors has been analyzed experimentally by Seiji Fujino (non-overlapping Schwarz-like to save computational costs), Man-Chung Yeung (sparse $\pm 1, 0$-random vectors).

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?
- look-ahead in IDR(s)?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?
- look-ahead in IDR(s)?
- non-linear IDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?
- look-ahead in IDR(s)?
- non-linear IDR?
- BiOMin(s,1) and BiODir(s,1) variants of IDR?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?
- look-ahead in IDR(s)?
- non-linear IDR?
- BiOMin(s,1) and BiODir(s,1) variants of IDR?
- IDR/IDRStab/GIDR based on Lanczos(s,ℓ)?

IDR: predicting the future

There is (as always) room for improvement. New ideas include:

- an Arnoldi-like IDR?
- QMRIDR?
- recycling in IDR?
- function approximation using IDR?
- look-ahead in IDR(s)?
- non-linear IDR?
- BiOMin(s,1) and BiODir(s,1) variants of IDR?
- IDR/IDRStab/GIDR based on Lanczos(s,ℓ)?
- variable preconditioning?

Conclusion

- We presented a sketch of the vast IDR family.

Conclusion

- We presented a sketch of the vast IDR family.
- Many “new” developments are reconsidered “old” approaches.

Conclusion

- We presented a sketch of the vast IDR family.
- Many “new” developments are reconsidered “old” approaches.
- IDR(s) is not really “new” . . .

Conclusion

- We presented a sketch of the vast IDR family.
- Many “new” developments are reconsidered “old” approaches.
- IDR(s) is not really “new” . . .
- . . . but a lot of new IDR(s) are behind the horizon.

Conclusion

- We presented a sketch of the vast IDR family.
- Many “new” developments are reconsidered “old” approaches.
- IDR(s) is not really “new” . . .
- . . . but a lot of new IDR(s) are behind the horizon.

My personal overall conclusion:

The IDR(s)-approach to LTPM by Sonneveld is more easy to follow (compared to other “multiple Lanczos” approaches).

. . . und zu guter Letzt:

Thank you for your attention

—

Vielen Dank für die Aufmerksamkeit,
Frohe Weihnachten!

(und für einige hier: Eine schöne Weihnachtsfeier!)

Brezinski, C. and Redivo Zaglia, M. (1994).
Treatment of near-breakdown in the CGS algorithm.
Numerical Algorithms, 7(1):33–73.

Brezinski, C. and Redivo-Zaglia, M. (1995).
Look-ahead in Bi-CGSTAB and other product methods for linear systems.
BIT. Numerical Mathematics, 35(2):169–201.

Chan, T. F., de Pillis, L., and van der Vorst, H. (1998).
Transpose-free formulations of Lanczos-type methods for nonsymmetric linear systems.
Numerical Algorithms, 17(1-2):51–66.

Chan, T. F., de Pillis, L. G., and van der Vorst, H. A. (1991).
A transpose-free squared Lanczos algorithm and application to solving nonsymmetric linear systems.
Computational and Applied Mathematics Report 91-17, Department of Mathematics, UCLA, Los Angeles.

Chan, T. F., Gallopoulos, E., Simoncini, V., Szeto, T., and Tong, C. H. (1994).
A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems.
SIAM J. Sci. Comput., 15(2):338–347.

Fokkema, D. R., Sleijpen, G. L. G., and van der Vorst, H. A. (1996).
Generalized conjugate gradient squared.
Journal of Computational and Applied Mathematics, 71(1):125–146.

Freund, R. W. (1993).
A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems.
SIAM Journal on Scientific Computing, 14(2):470–482.

Freund, R. W. and Szeto, T. (1991).
A quasi-minimal residual squared algorithm for non-Hermitian linear systems.
Technical Report 91.26, RIACS, NASA Ames Research Center, Moffett Field, CA.

Freund, R. W. and Szeto, T. (1992a).
A quasi-minimal residual squared algorithm for non-Hermitian linear systems.
CAM Report 92-19, UCLA Dept. of Math.

Freund, R. W. and Szeto, T. (1992b).
A transpose-free quasi-minimal residual squared algorithm for non-Hermitian linear systems.
In Vichnevetsky, R., Knight, D., and Richter, G., editors, *Advances in Computer Methods for Partial Differential Equations – VII*, pages 258–264. IMACS.

Fujino, S. (2002).
GPBiCG(m, l): a hybrid of BiCGSTAB and GPBiCG methods with efficiency and robustness.
Applied Numerical Mathematics. An IMACS Journal, 41(1):107–117. Developments and trends in iterative methods for large systems of equations—in memoriam Rüdiger Weiss (Lausanne, 2000).

Fujino, S., Fujiwara, M., and Yoshida, M. (2005).
BiCGSafe method based on minimization of associate residual (In Japanese).
Transactions of the Japan Society for Computational Engineering and Science, 2005:8 pages.

Gutknecht, M. H. (1993).
Variants of BICGSTAB for matrices with complex spectrum.
SIAM J. Sci. Comput., 14(5):1020–1033.

Gutknecht, M. H. (1997).
Lanczos-type solvers for nonsymmetric linear systems of equations.
Acta Numerica, 6:271–397.

Gutknecht, M. H. and Ressel, K. J. (2000).
Look-ahead procedures for Lanczos-type product methods based on three-term Lanczos recurrences.
SIAM Journal on Matrix Analysis and Applications, 21(4):1051–1078 (electronic).

Lanczos, C. (1950).
An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.
J. Res. Natl. Bur. Stand., 45(4):255–282.

Röllin, S. and Gutknecht, M. H. (2002).
Variations of Zhang's Lanczos-type product method.
Applied Numerical Mathematics. An IMACS Journal, 41(1):119–133. Developments and trends in iterative methods for large systems of equations—in memoriam Rüdiger Weiss (Lausanne, 2000).

Simoncini, V. and Szyld, D. (2009).
Interpreting IDR as a Petrov-Galerkin method.
Report 09-10-22, Dipartimento di Matematica, Università di Bologna and Department of Mathematics, Temple University, Philadelphia.

Sleijpen, G. L. G. and Abe, K. (2010).
publication in preparation (December 2009).

Sleijpen, G. L. G. and Fokkema, D. R. (1993).
BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum.
Electronic Transactions on Numerical Analysis, 1(Sept.):11–32 (electronic only).

Sleijpen, G. L. G., Sonneveld, P., and van Gijzen, M. B. (2008).
Bi-CGSTAB as an induced dimension reduction method.
Reports of the Department of Applied Mathematical Analysis Report 08-07, Delft University of Technology.
ISSN 1389-6520.

Sleijpen, G. L. G. and van Gijzen, M. B. (2009).
Exploiting BiCGstab(ℓ) strategies to induce dimension reduction.
Reports of the Department of Applied Mathematical Analysis Report 09-02, Delft University of Technology.
ISSN 1389-6520.

Sonneveld, P. (1984).
CGS, a fast Lanczos-type solver for nonsymmetric linear systems.
Report 84-16, Department of Mathematics and Informatics, Delft University of Technology.

Sonneveld, P. (1989).
CGS: A fast Lanczos-type solver for nonsymmetric linear systems.
SIAM J. Sci. Stat. Comput., 10:36–52.

Sonneveld, P. (2006).
History of IDR: an example of serendipity.
PDF file sent by Peter Sonneveld on Monday, 24th of July 2006.
8 pages; evolved into (Sonneveld, 2008).

Sonneveld, P. (2008).
AGS-IDR-CGS-BiCGSTAB-IDR(s): The circle closed. A case of serendipity.
In *Proceedings of the International Kyoto Forum 2008 on Krylov subspace methods*, pages 1–14.

Tanio, M. and Sugihara, M. (2008).
GIDR(s,l): generalized IDR(s).
In *The 2008 annual conference of the Japan Society for Industrial and Applied Mathematic*, pages 411–412, Chiba, Japan.
(In Japanese).

Tanio, M. and Sugihara, M. (2009).
GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials.
Technical Report METR 2009-16, Department of Mathematical Informatics, Graduate School of information Science and Technology, University of Tokio.

van der Vorst, H. A. (1992).
Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems.
SIAM J. Sci. Stat. Comput., 13(2):631–644.

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

van Gijzen, M. B. and Sonneveld, P. (2008).
An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties.
Reports of the Department of Applied Mathematical Analysis Report 08-21, Delft University of Technology.
ISSN 1389-6520.

Wesseling, P. and Sonneveld, P. (1980).
Numerical experiments with a multiple grid and a preconditioned Lanczos type method.
In *Approximation Methods for Navier-Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 543–562. Springer.

Yeung, M.-C. (2009).
ML(n)BiCGSTAB: reformulation, analysis and implementation.
Report 09-01, Math. Dept., UW.
(A draft — subject to change).

Yeung, M.-C. and Chan, T. F. (1999/2000).
ML(k)BiCGSTAB: a BiCGSTAB variant based on multiple Lanczos starting vectors.
SIAM Journal on Scientific Computing, 21(4):1263–1290 (electronic).

Zhang, S.-L. (1997).
GPBi-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems.
SIAM Journal on Scientific Computing, 18(2):537–551.

Zhou, L. and Walker, H. F. (1994).
Residual smoothing techniques for iterative methods.
SIAM Journal on Scientific Computing, 15(2):297–312.
Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).