IDR – A New Class of Krylov Subspace Solvers: Benefits and Drawbacks

Jens-Peter M. Zemke
zemke@tu-harburg.de

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

May 26th, 2011

Outline

Outline

Is IDR a new method?

IDR is yet another acronym, here, for Induced Dimension Reduction.

- The original IDR method was developed in 1976 by Peter Sonneveld while preparing a lecture in numerical analysis for students of TU Delft.

Is IDR a new method?

IDR is yet another acronym, here, for Induced Dimension Reduction.

- The original IDR method was developed in 1976 by Peter Sonneveld while preparing a lecture in numerical analysis for students of TU Delft.
- It was presented in 1979 at a CFD conference and published in 1980.

Is IDR a new method?

IDR is yet another acronym, here, for Induced Dimension Reduction.

- The original IDR method was developed in 1976 by Peter Sonneveld while preparing a lecture in numerical analysis for students of TU Delft.
- It was presented in 1979 at a CFD conference and published in 1980.
- In 2006, Peter Sonneveld and Martin van Gijzen reconsidered the original variant and constructed something “new”, namely, IDR(s).

Is IDR a new method?

IDR is yet another acronym, here, for Induced Dimension Reduction.

- The original IDR method was developed in 1976 by Peter Sonneveld while preparing a lecture in numerical analysis for students of TU Delft.
- It was presented in 1979 at a CFD conference and published in 1980.
- In 2006, Peter Sonneveld and Martin van Gijzen reconsidered the original variant and constructed something “new”, namely, IDR(s).
- Is IDR/IDR(s) really new? Are parts of it new?

Is IDR a new method?

IDR is yet another acronym, here, for Induced Dimension Reduction.

- The original IDR method was developed in 1976 by Peter Sonneveld while preparing a lecture in numerical analysis for students of TU Delft.
- It was presented in 1979 at a CFD conference and published in 1980.
- In 2006, Peter Sonneveld and Martin van Gijzen reconsidered the original variant and constructed something “new”, namely, IDR(s).
- Is IDR/IDR(s) really new? Are parts of it new?

Spoiler:

- IDR marks the beginning of a new era in Krylov subspace methods,
- IDR(s) is closely related to ML(k)BiCGStab by Yeung and Chan.

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 $k \gg 1$

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

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.

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.

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

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

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

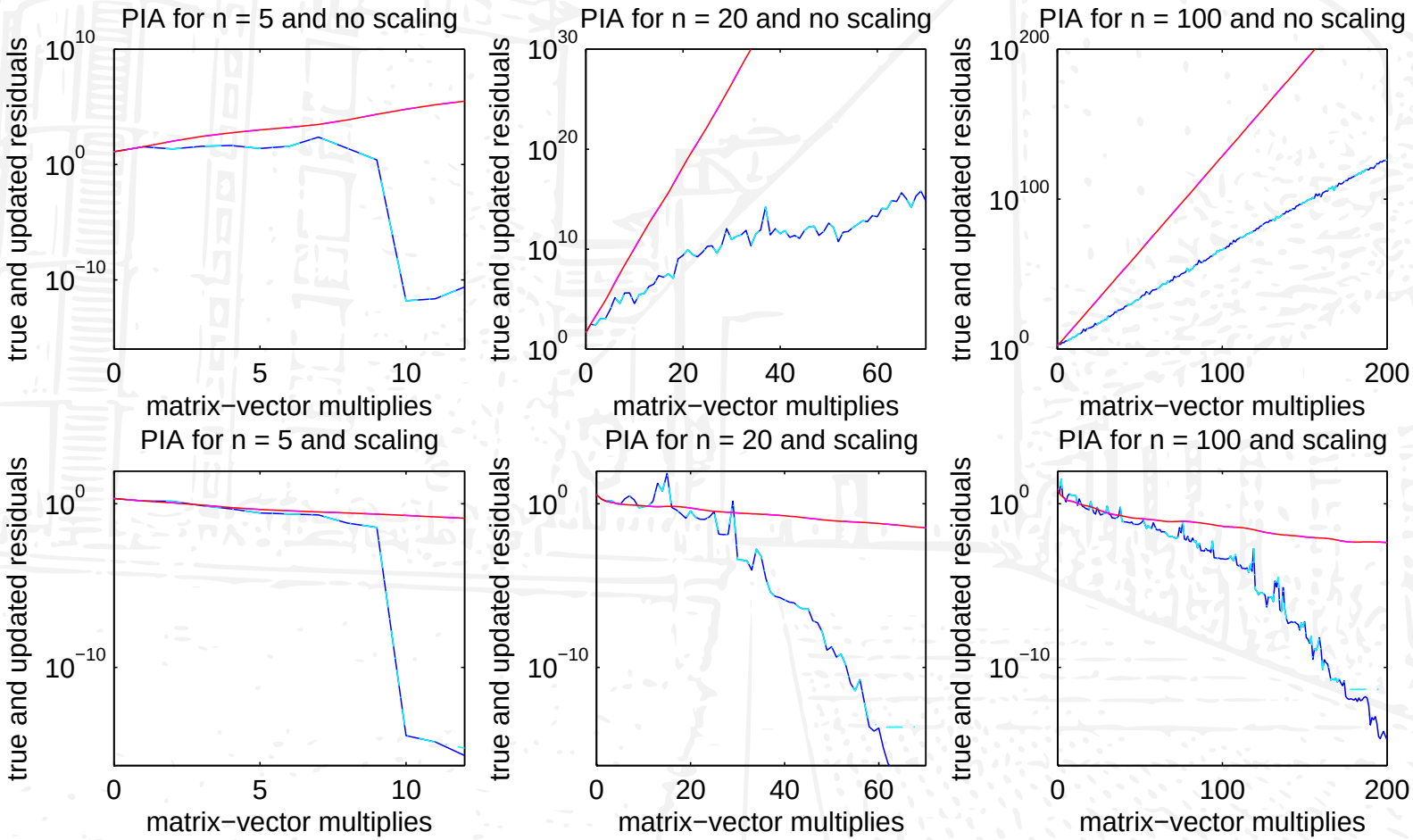
$$\gamma = \mathbf{p}^{\mathsf{T}}\mathbf{r}_{\text{new}} / \mathbf{p}^{\mathsf{T}}(\mathbf{r}_{\text{old}} - \mathbf{r}_{\text{new}})$$
$$\mathbf{s} = \mathbf{r}_{\text{new}} + \gamma(\mathbf{r}_{\text{new}} - \mathbf{r}_{\text{old}})$$
$$\mathbf{x}_{\text{tmp}} = \mathbf{x}_{\text{new}} + \gamma(\mathbf{x}_{\text{new}}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

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.

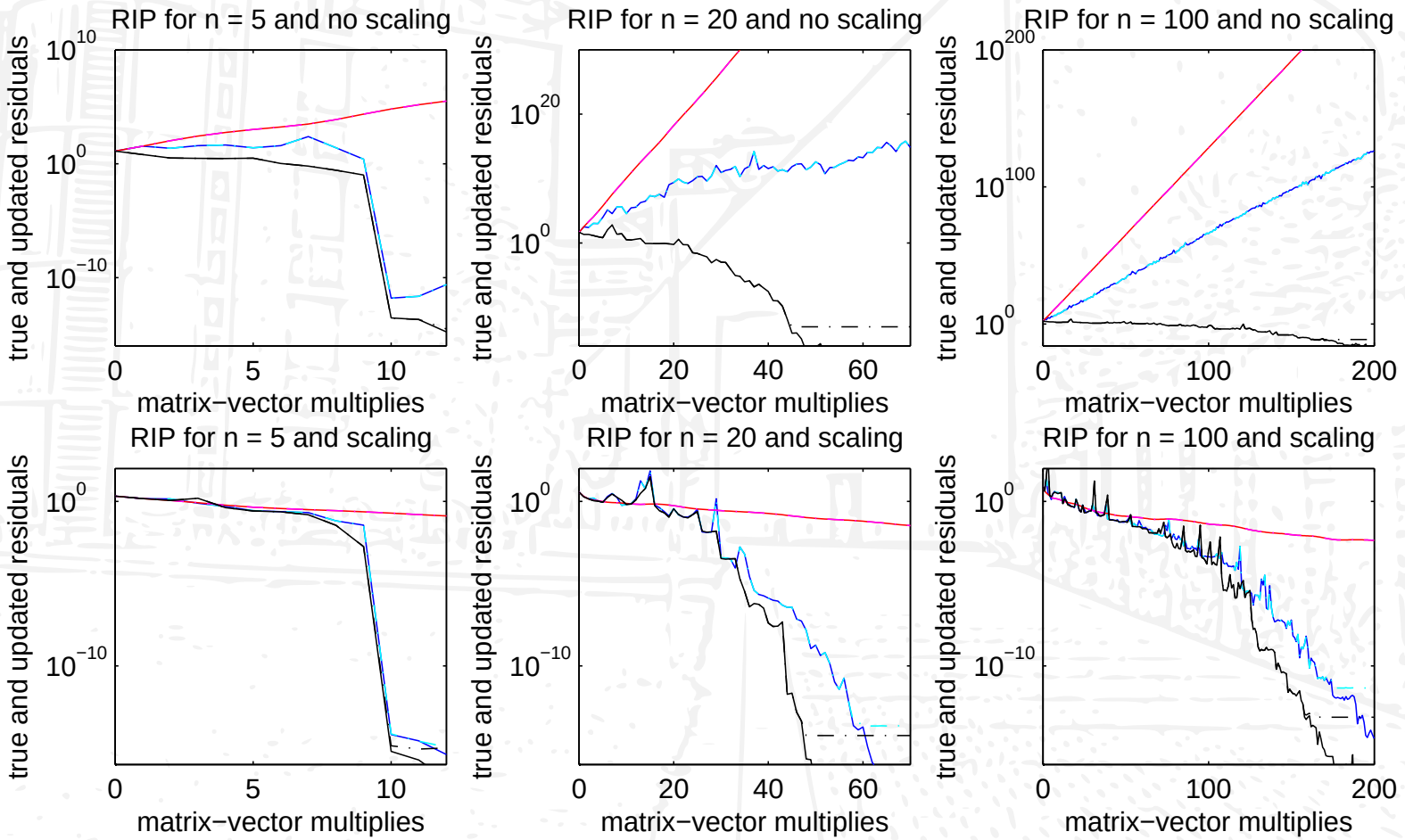
This part of the story took place "in the background" in the year 1976.

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

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



Building blocks of IDR(s)

IDR(s) is a Krylov subspace method based on two building blocks:

- Multiplication by polynomials in $\mathbf{A}$.
  (IDR(s): linear, IDR(s)Stab(ℓ): higher degree)
- Oblique projection perpendicular to $\mathbf{P} \in \mathbb{C}^{n \times s}$.

Building blocks of IDR(s)

IDR(s) is a Krylov subspace method based on two building blocks:

- Multiplication by polynomials in $\mathbf{A}$.
  (IDR(s): linear, IDR(s)Stab(ℓ): higher degree)
- Oblique projection perpendicular to $\mathbf{P} \in \mathbb{C}^{n \times s}$.

IDR(s) constructs nested subspaces of shrinking dimensions.

Building blocks of IDR(s)

IDR(s) is a Krylov subspace method based on two building blocks:

- Multiplication by polynomials in $\mathbf{A}$.
  (IDR(s): linear, IDR(s)Stab(ℓ): higher degree)
- Oblique projection perpendicular to $\mathbf{P} \in \mathbb{C}^{n \times s}$.

IDR(s) constructs nested subspaces of shrinking dimensions.

The prototype IDR(s) method constructs spaces $\mathcal{G}_j$ as follows:

- Define $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots\}$.
- Iterate $\mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots, \quad \mathbb{C} \ni \omega_j \neq 0$

Building blocks of IDR(s)

IDR(s) is a Krylov subspace method based on two building blocks:

- Multiplication by polynomials in $\mathbf{A}$.
  (IDR(s): linear, IDR(s)Stab(ℓ): higher degree)
- Oblique projection perpendicular to $\mathbf{P} \in \mathbb{C}^{n \times s}$.

IDR(s) constructs nested subspaces of shrinking dimensions.

The prototype IDR(s) method constructs spaces $\mathcal{G}_j$ as follows:

- Define $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots\}$.
- Iterate $\mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots, \quad \mathbb{C} \ni \omega_j \neq 0$

Only sufficiently many vectors in each space are constructed.

IDR is Lanczos times something

It turns out that:

- IDR(s) is a transpose-free variant of a Lanczos process with one right-hand side and s left-hand sides.

IDR is Lanczos times something

It turns out that:

- IDR(s) is a transpose-free variant of a Lanczos process with one right-hand side and s left-hand sides.
- IDR(s) is a Lanczos-type product method, i.e., most residuals can be written as

$$\mathbf{r}_{j(s+1)+k}^{\text{IDR}} = \Omega_j(\mathbf{A})\rho_{js+k}(\mathbf{A})\mathbf{r}_0, \quad 1 \leqslant k \leqslant s$$

  where ρ_{js+k} are residual polynomials of the Lanczos process.

IDR is Lanczos times something

It turns out that:

- IDR(s) is a transpose-free variant of a Lanczos process with one right-hand side and s left-hand sides.
- IDR(s) is a Lanczos-type product method, i.e., most residuals can be written as

$$\mathbf{r}_{j(s+1)+k}^{\text{IDR}} = \Omega_j(\mathbf{A})\rho_{js+k}(\mathbf{A})\mathbf{r}_0, \quad 1 \leqslant k \leqslant s$$

  where ρ_{js+k} are residual polynomials of the Lanczos process.

Reminder: Residual polynomials are polynomials that

- satisfy $\mathbf{r}_k = \rho_k(\mathbf{A})\mathbf{r}_0$ and
- are normalized by the condition $\rho_k(0) = 1$.

Outline

Krylov subspace: try thinking in polynomials

IDR/IDR(s)/IDR(s)Stab(ℓ) are classes of Krylov subspace methods, they construct approximations from Krylov subspaces

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}.$$

Krylov subspace: try thinking in polynomials

IDR/IDR(s)/IDR(s)Stab(ℓ) are classes of Krylov subspace methods, they construct approximations from Krylov subspaces

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}.$$

Krylov subspaces are isomorphic (up to a certain degree) to polynomial spaces,

$$\mathbf{x} \in \mathcal{K}_k \quad \Leftrightarrow \quad \mathbf{x} = \sum_{j=0}^{k-1} \mathbf{A}^j \mathbf{r}_0 c_j = p_{k-1}(\mathbf{A})\mathbf{r}_0, \quad p_{k-1}(z) = \sum_{j=0}^{k-1} c_j z^j.$$

Krylov subspace: try thinking in polynomials

IDR/IDR(s)/IDR(s)Stab(ℓ) are classes of Krylov subspace methods, they construct approximations from Krylov subspaces

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}.$$

Krylov subspaces are isomorphic (up to a certain degree) to polynomial spaces,

$$\mathbf{x} \in \mathcal{K}_k \quad \Leftrightarrow \quad \mathbf{x} = \sum_{j=0}^{k-1} \mathbf{A}^j \mathbf{r}_0 c_j = p_{k-1}(\mathbf{A})\mathbf{r}_0, \quad p_{k-1}(z) = \sum_{j=0}^{k-1} c_j z^j.$$

Residual polynomials arise because

$$\mathbf{r}_k := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_k = (\mathbf{I} - \mathbf{A}p_{k-1}(\mathbf{A}))\mathbf{r}_0 =: \rho_k(\mathbf{A})\mathbf{r}_0.$$

IDR: a Lanczos process with multiple left-hand sides

The derivation and the theoretical properties of IDR are easy to describe using the language of polynomials.

IDR: a Lanczos process with multiple left-hand sides

The derivation and the theoretical properties of IDR are easy to describe using the language of polynomials.

Denote $\Omega_k(\mathbf{A}) := \prod_{\ell=1}^{k}(\mathbf{I} - \omega_\ell \mathbf{A})$. It can easily be proven that ($\mathcal{S} := \mathbf{P}^\perp$)

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

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

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

IDR: a Lanczos process with multiple left-hand sides

The derivation and the theoretical properties of IDR are easy to describe using the language of polynomials.

Denote $\Omega_k(\mathbf{A}) := \prod_{\ell=1}^{k}(\mathbf{I} - \omega_\ell \mathbf{A})$. It can easily be proven that ($\mathcal{S} := \mathbf{P}^\perp$)

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

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

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

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

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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:

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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 $s + 1$ vectors are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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 $s+1$ vectors are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,
- the next $s+1$ vectors in $\mathcal{G}_1$ are in the $\mathbf{I} - \omega_1 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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 $s+1$ vectors are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,
- the next $s+1$ vectors in $\mathcal{G}_1$ are in the $\mathbf{I} - \omega_1 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,
- the next $s+1$ vectors in $\mathcal{G}_2$ are in the $\mathbf{I} - \omega_2 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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 $s+1$ vectors are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,
- the next $s+1$ vectors in $\mathcal{G}_1$ are in the $\mathbf{I} - \omega_1 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,
- the next $s+1$ vectors in $\mathcal{G}_2$ are in the $\mathbf{I} - \omega_2 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,
- the last $s+1$ vectors are in the $\mathbf{I} - \omega_j \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^{\perp}$,

IDR: a Lanczos process with multiple left-hand sides

Recall that $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0), \quad \mathcal{G}_j := (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots$

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 $s+1$ vectors are in $\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$,
- the next $s+1$ vectors in $\mathcal{G}_1$ are in the $\mathbf{I} - \omega_1 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^\perp$,
- the next $s+1$ vectors in $\mathcal{G}_2$ are in the $\mathbf{I} - \omega_2 \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^\perp$,
- the last $s+1$ vectors are in the $\mathbf{I} - \omega_j \mathbf{A}$ image of $\mathcal{S} = \mathbf{P}^\perp$,
- the last vectors 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

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

IDR: a Lanczos process with multiple left-hand sides

The second equality

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

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

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

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

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

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

IDR: a Lanczos process with multiple left-hand sides

The third and fourth equality

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

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.

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.

This is of interest in round-off error analysis (Lanczos): “Local orthogonality” is preserved, the inner products with the oldest basis vectors, i.e., those that are the columns of $\mathbf{P}$, are “small”.

Hessenberg decompositions: basic linear algebra

The implementation and (round-off error) analysis of IDR is more closely related to so-called generalized Hessenberg decompositions.

Hessenberg decompositions: basic linear algebra

The implementation and (round-off error) analysis of IDR is more closely related to so-called generalized Hessenberg decompositions.

"Classical" Krylov subspace methods generate

- "basis" matrices $\mathbf{Q}_{k+1} = \left(\mathbf{Q}_k, \mathbf{q}_{k+1}\right) \in \mathbb{C}^{n\times(k+1)}$ and
- unreduced extended Hessenberg matrices $\underline{\mathbf{H}}_k \in \mathbb{C}^{(k+1)\times k}$

which form the Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k.$$

Hessenberg decompositions: basic linear algebra

The implementation and (round-off error) analysis of IDR is more closely related to so-called generalized Hessenberg decompositions.

"Classical" Krylov subspace methods generate

- "basis" matrices $\mathbf{Q}_{k+1} = \left(\mathbf{Q}_k, \mathbf{q}_{k+1}\right) \in \mathbb{C}^{n\times(k+1)}$ and
- unreduced extended Hessenberg matrices $\underline{\mathbf{H}}_k \in \mathbb{C}^{(k+1)\times k}$

which form the Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k.$$

IDR based Krylov subspace methods additionally generate upper triangular matrices $\mathbf{U}_k \in \mathbb{C}^{k\times k}$ such that we obtain a generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_k\mathbf{U}_k = \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k.$$

Hessenberg decompositions: basic linear algebra

The implementation and (round-off error) analysis of IDR is more closely related to so-called generalized Hessenberg decompositions.

"Classical" Krylov subspace methods generate

- "basis" matrices $\mathbf{Q}_{k+1} = \left(\mathbf{Q}_k, \mathbf{q}_{k+1}\right) \in \mathbb{C}^{n\times(k+1)}$ and
- unreduced extended Hessenberg matrices $\underline{\mathbf{H}}_k \in \mathbb{C}^{(k+1)\times k}$

which form the Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k.$$

IDR based Krylov subspace methods additionally generate upper triangular matrices $\mathbf{U}_k \in \mathbb{C}^{k\times k}$ such that we obtain a generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_k\mathbf{U}_k = \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k.$$

IDR based methods include BiCGStab (rewritten version of IDR), and CGS.

Karl Hessenberg & “his” matrix + decomposition

Behandlung linearer Eigenwertaufgaben mit Hilfe der Hamilton-Cayleyschen Gleichung, Karl Hessenberg, 1. Bericht der Reihe „Numerische Verfahren“, July, 23rd 1940, page 23:

Man kann nun die Vektoren $z_\nu^{(\nu-1)}$ $(\nu = 1,2,\ldots,n)$ ebenfalls in einer Matrix zusammenfassen, und zwar ist nach Gleichung (55) und (56)

(57) $$(z_1 z_2' z_3'' \cdots z_n^{(n-1)}) = \mathfrak{A} \cdot z' = z' \cdot \mathfrak{P},$$

worin die Matrix $\mathfrak{P}$ zur Abkürzung gesetzt ist für

(58) $$\mathfrak{P} = \begin{pmatrix} \alpha_{10} & \alpha_{20} & \cdots & \alpha_{n-1,0} & \alpha_{n0} \\ 1 & \alpha_{21} & \cdots & \alpha_{n-1,1} & \alpha_{n1} \\ 0 & 1 & \cdots & \alpha_{n-1,2} & \alpha_{n2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \alpha_{n,n-1} \end{pmatrix}.$$

- Hessenberg decomposition, Eqn. (57),
- Hessenberg matrix, Eqn. (58).

Karl Hessenberg (* September 8th, 1904, † February 22nd, 1959)

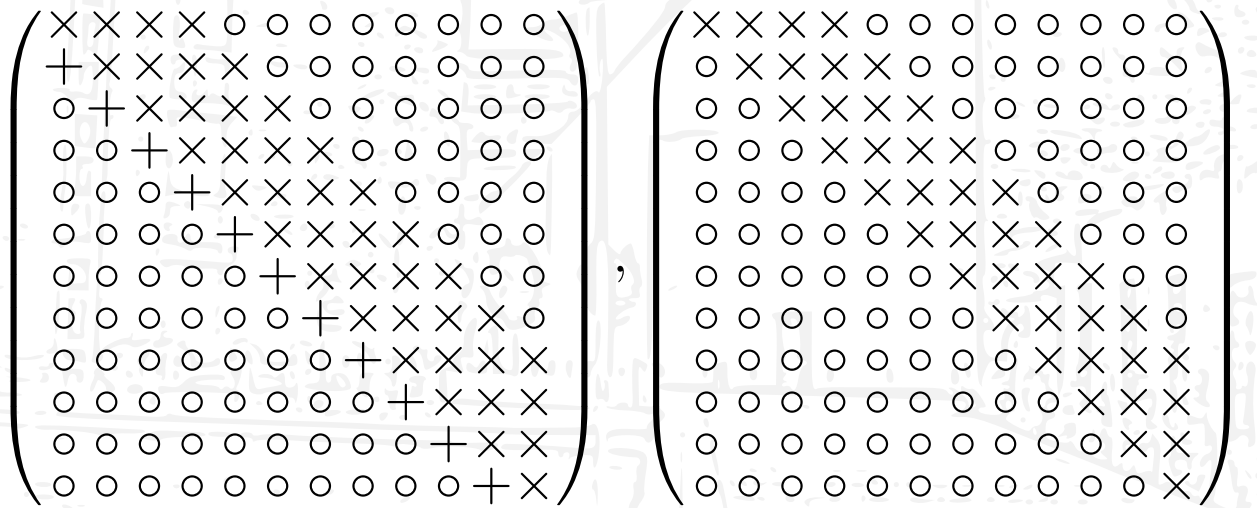
IDR: Sonneveld pencil and Sonneveld matrix

We consider the prototype IDR(s) by Sonneveld/van Gijzen (IDR(s)ORes).

IDR: Sonneveld pencil and Sonneveld matrix

We consider the prototype IDR(s) by Sonneveld/van Gijzen (IDR(s)ORes).

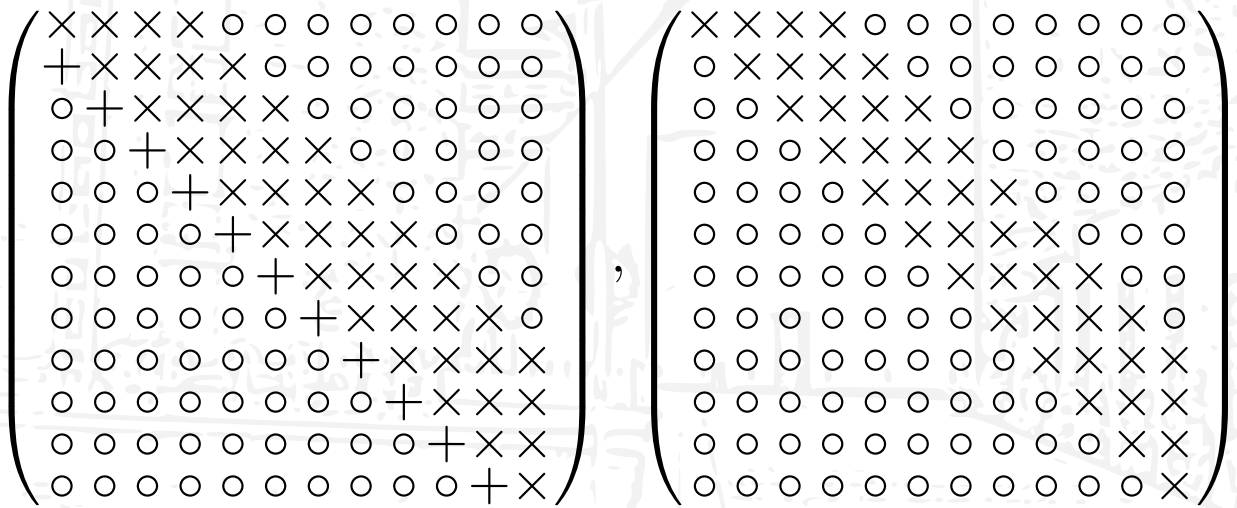
The IDR(s)ORes pencil, the so-called Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n \mathbf{D}_\omega^{(n)})$, can be depicted by



IDR: Sonneveld pencil and Sonneveld matrix

We consider the prototype IDR(s) by Sonneveld/van Gijzen (IDR(s)ORes).

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.

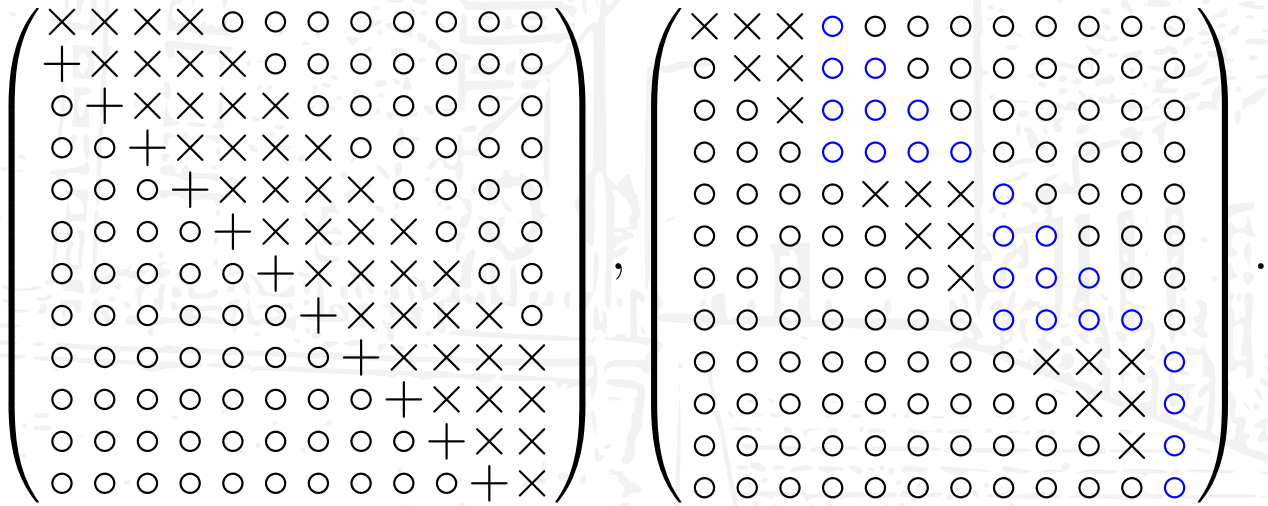
Understanding IDR: Purification

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.

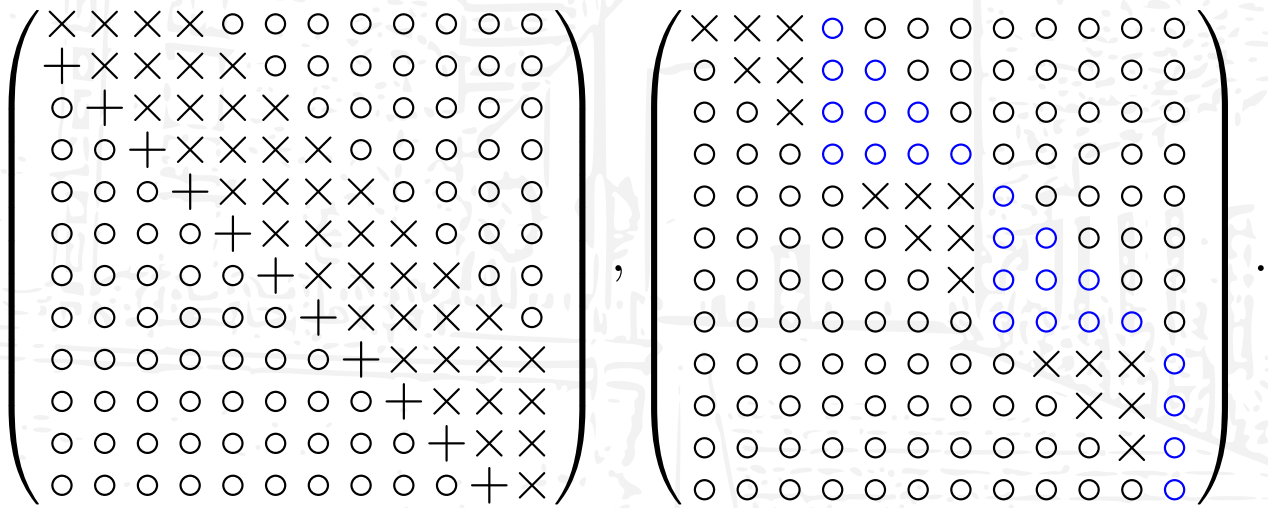
The purified IDR(s)ORes pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega^{(n)})$, that has only the remaining eigenvalues and some infinite ones as eigenvalues, can be depicted by



Understanding IDR: Purification

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

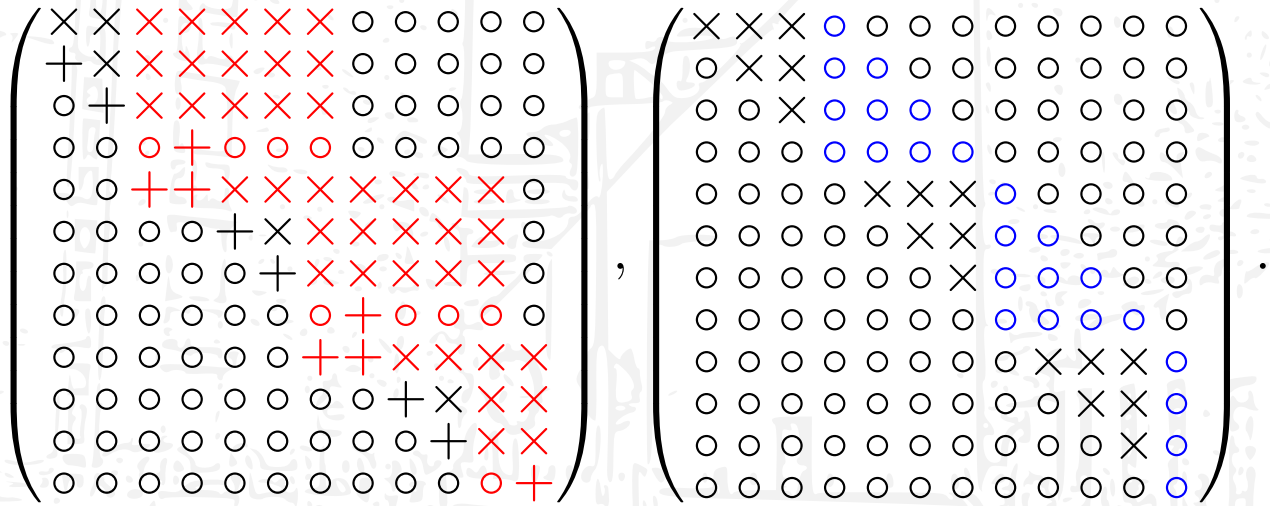
The purified IDR(s)ORes pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n \mathbf{D}_\omega^{(n)})$, that has only the remaining eigenvalues and some infinite ones as eigenvalues, can be depicted by



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

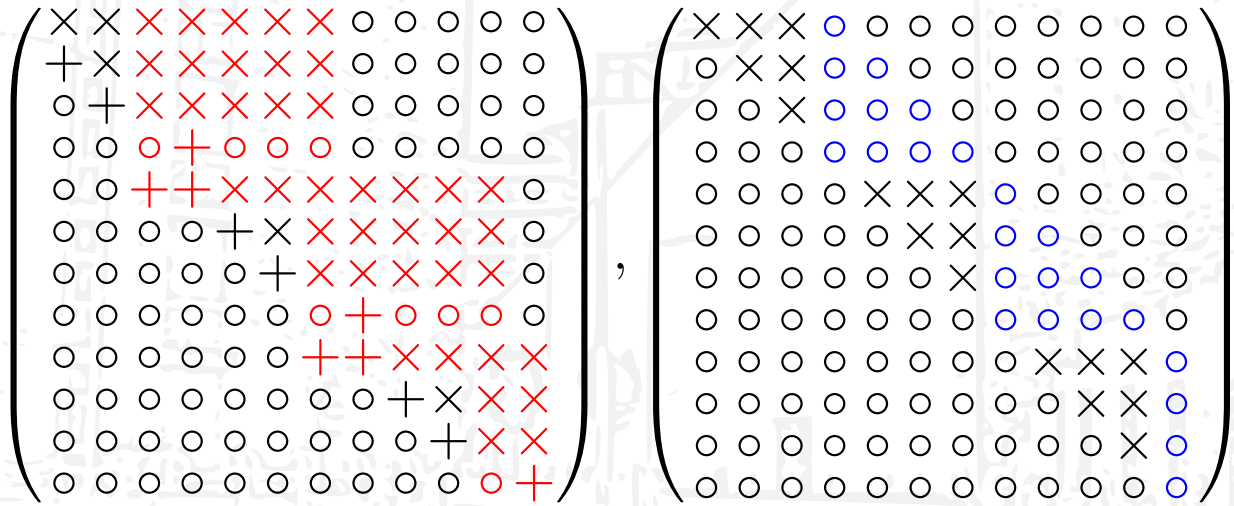
Understanding IDR: Gaussian elimination

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.

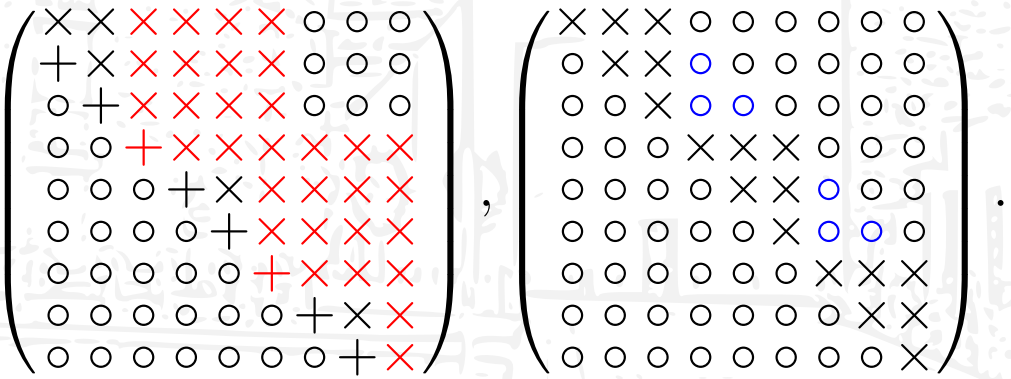
Understanding IDR: Deflation

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.

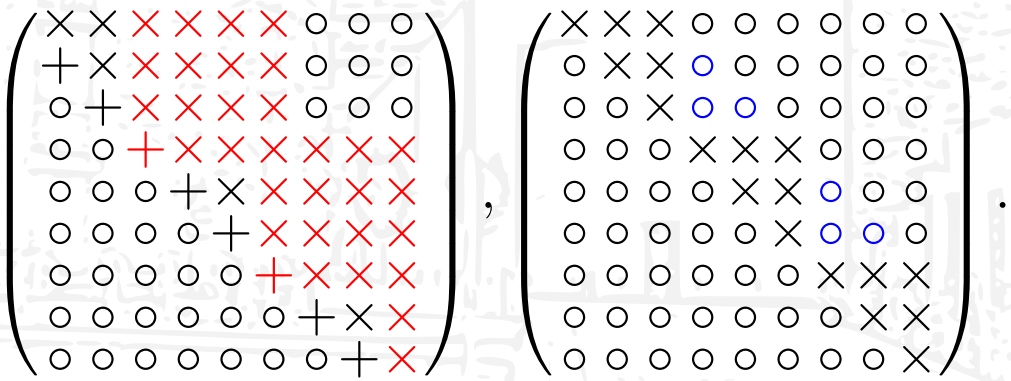
The deflated purified IDR(s)ORes pencil, after the deflation step $(D(\mathbf{Y}_n^\circ \mathbf{G}_n), D(\mathbf{U}_n \mathbf{D}_\omega^{(n)}))$, can be depicted by



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 $\mathbf{q}_{js+k}$, $k = 0,\ldots,s-1$, $j = 0,1,\ldots$, are given by

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

Outline

IDR is based on a short-term recurrence, e.g., a Lanczos method. It is well known that Lanczos methods tend to deviate. Thus, we might expect the same behaviour in IDR based methods.

IDR is based on a short-term recurrence, e.g., a Lanczos method. It is well known that Lanczos methods tend to deviate. Thus, we might expect the same behaviour in IDR based methods.

At least we might expect some deviation, as IDR is based on short term recurrences.

IDR is based on a short-term recurrence, e.g., a Lanczos method. It is well known that Lanczos methods tend to deviate. Thus, we might expect the same behaviour in IDR based methods.

At least we might expect some deviation, as IDR is based on short term recurrences.

In the following plots we depict (known) behavior of Lanczos algorithms and compare if to the (yet to be analyzed) behavior of IDR algorithms.

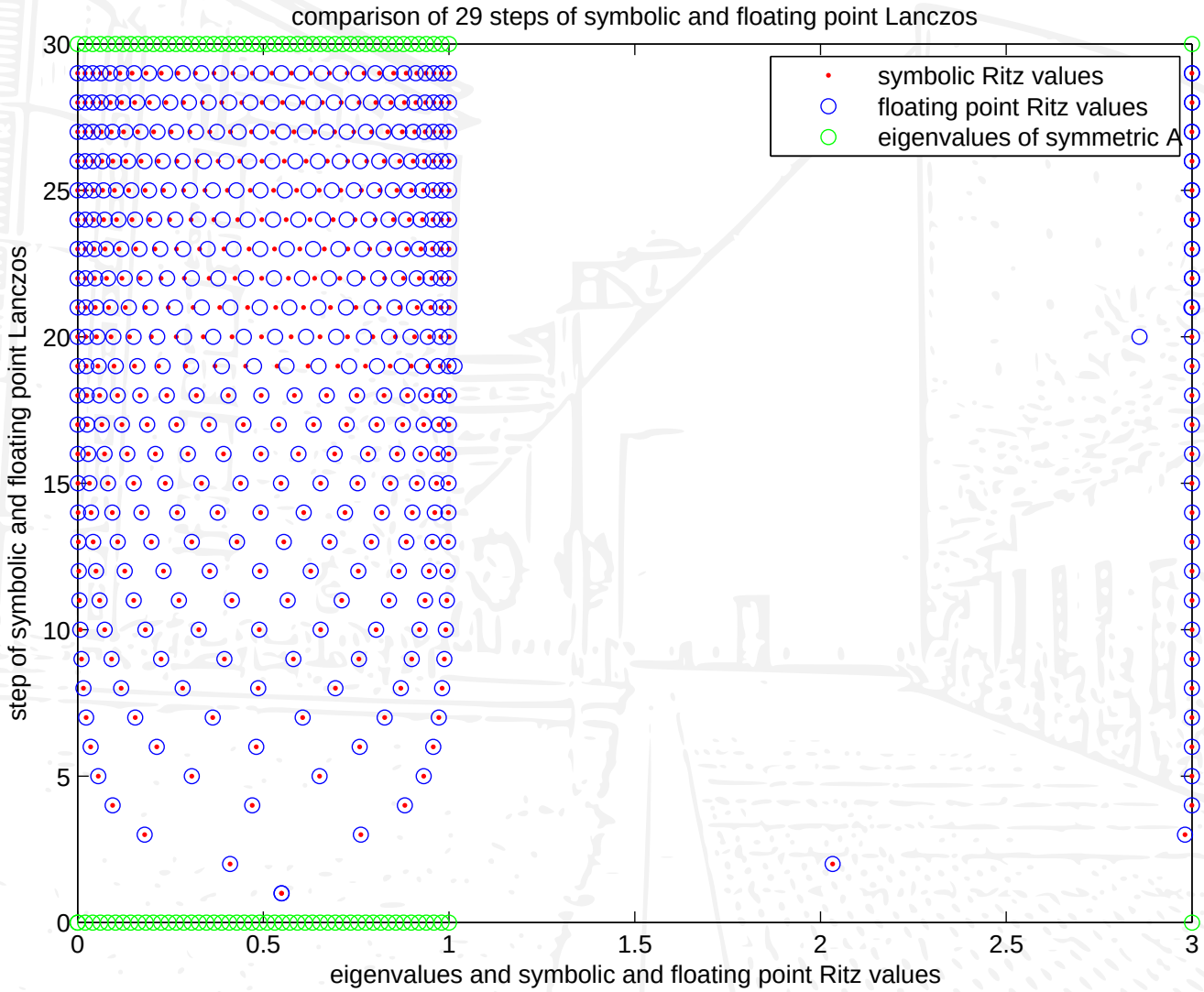
IDR is based on a short-term recurrence, e.g., a Lanczos method. It is well known that Lanczos methods tend to deviate. Thus, we might expect the same behaviour in IDR based methods.

At least we might expect some deviation, as IDR is based on short term recurrences.

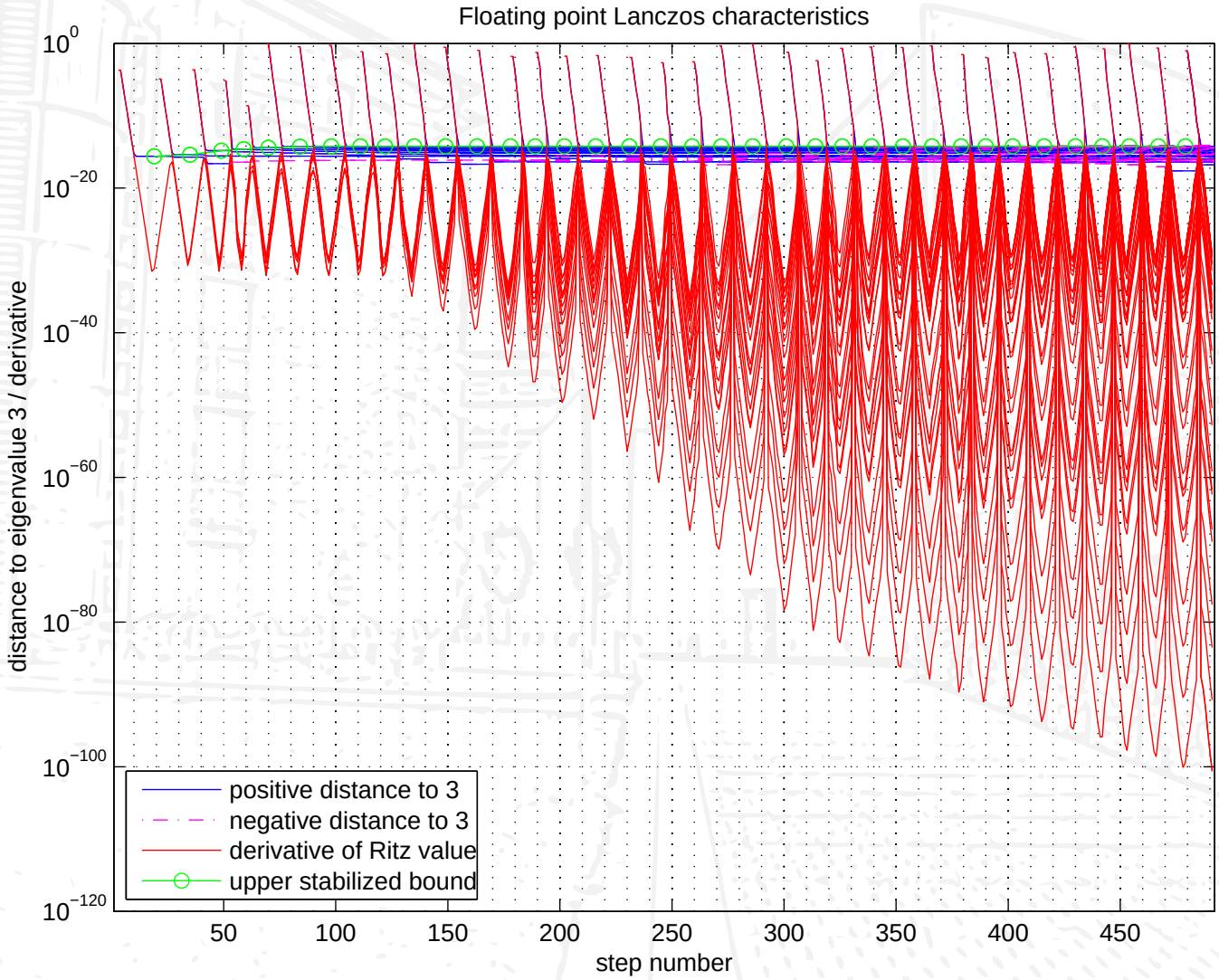
In the following plots we depict (known) behavior of Lanczos algorithms and compare if to the (yet to be analyzed) behavior of IDR algorithms.

In a recent report, Collignon, Sleijpen and van Gijzen show that IDR can be interpreted as a sort of preconditioning based on deflation; the preconditioned matrix has the polynomial roots as eigenvalues.

Lanczos' method in finite precision



Lanczos' method in finite precision



Lanczos' method in finite precision

The theory of the Lanczos method in case of non-selfadjoint matrices is still less satisfactory. Some of the conclusions carry over, and the behavior in finite precision shows some similarities.

Lanczos' method in finite precision

The theory of the Lanczos method in case of non-selfadjoint matrices is still less satisfactory. Some of the conclusions carry over, and the behavior in finite precision shows some similarities.

The next example uses the matrix `pores_2` of size 1224×1224 from Matrix Market. The left and right starting vectors have been chosen such that all components are equal.

Lanczos' method in finite precision

The theory of the Lanczos method in case of non-selfadjoint matrices is still less satisfactory. Some of the conclusions carry over, and the behavior in finite precision shows some similarities.

The next example uses the matrix `pores_2` of size 1224×1224 from Matrix Market. The left and right starting vectors have been chosen such that all components are equal.

As there does not exist the best Lanczos method, we have chosen one of the more stable ones, namely the variant described in (Bai, 1994).

Lanczos' method in finite precision

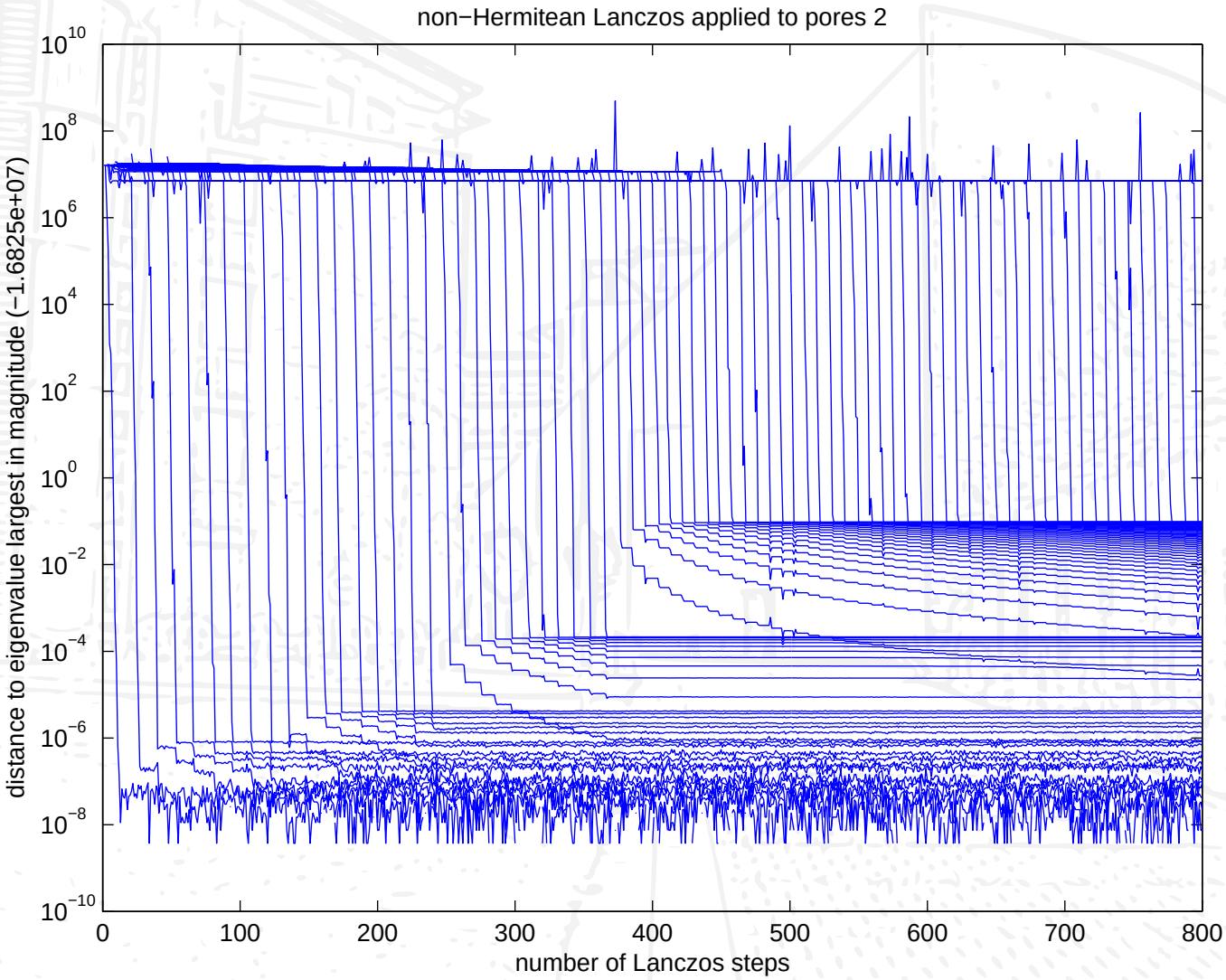
The theory of the Lanczos method in case of non-selfadjoint matrices is still less satisfactory. Some of the conclusions carry over, and the behavior in finite precision shows some similarities.

The next example uses the matrix `pores_2` of size 1224×1224 from Matrix Market. The left and right starting vectors have been chosen such that all components are equal.

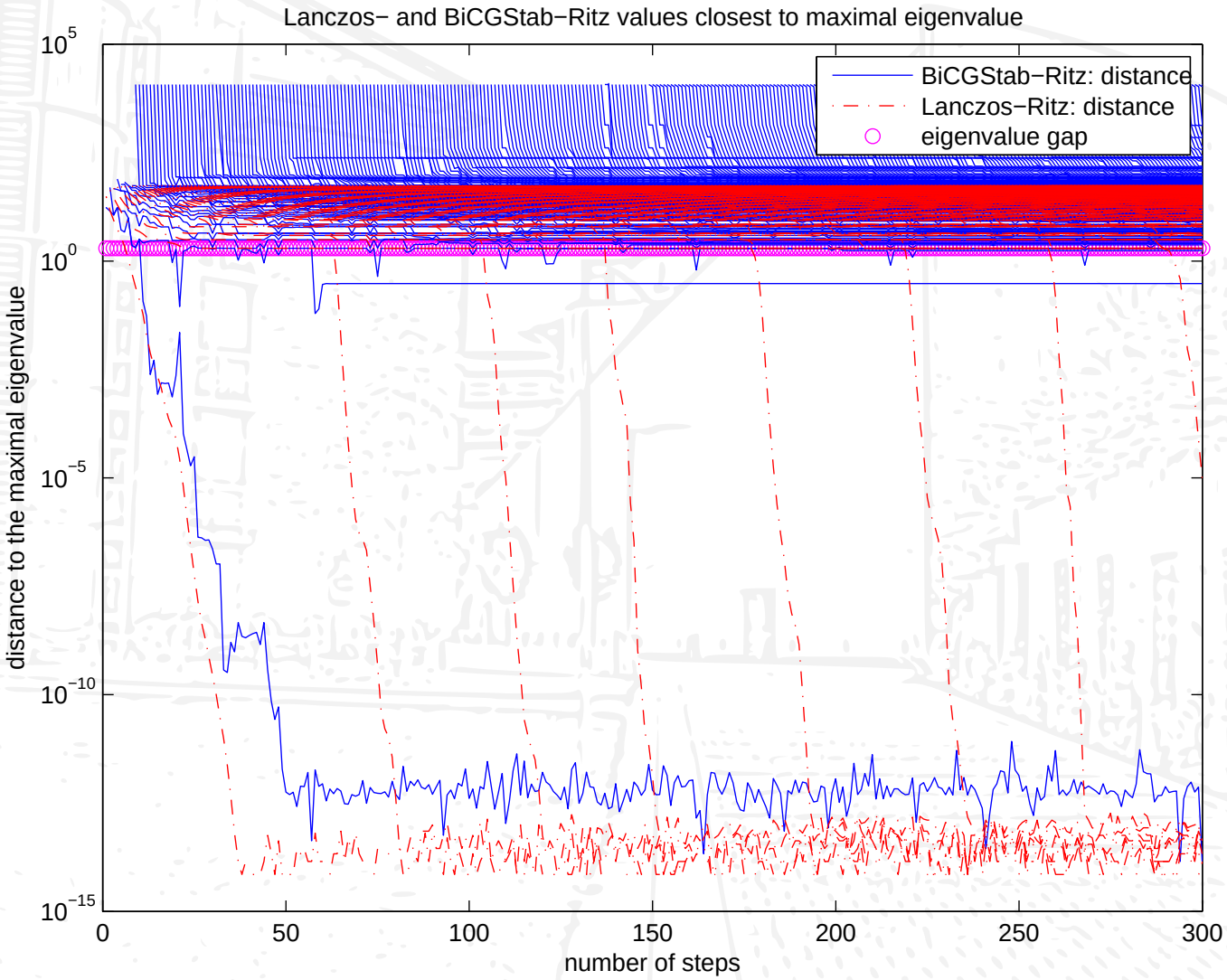
As there does not exist the best Lanczos method, we have chosen one of the more stable ones, namely the variant described in (Bai, 1994).

We note that we can observe multiple copies, but this time the approximation quality is reduced after a couple of steps, all Ritz values computed after certain steps show worse behavior than before.

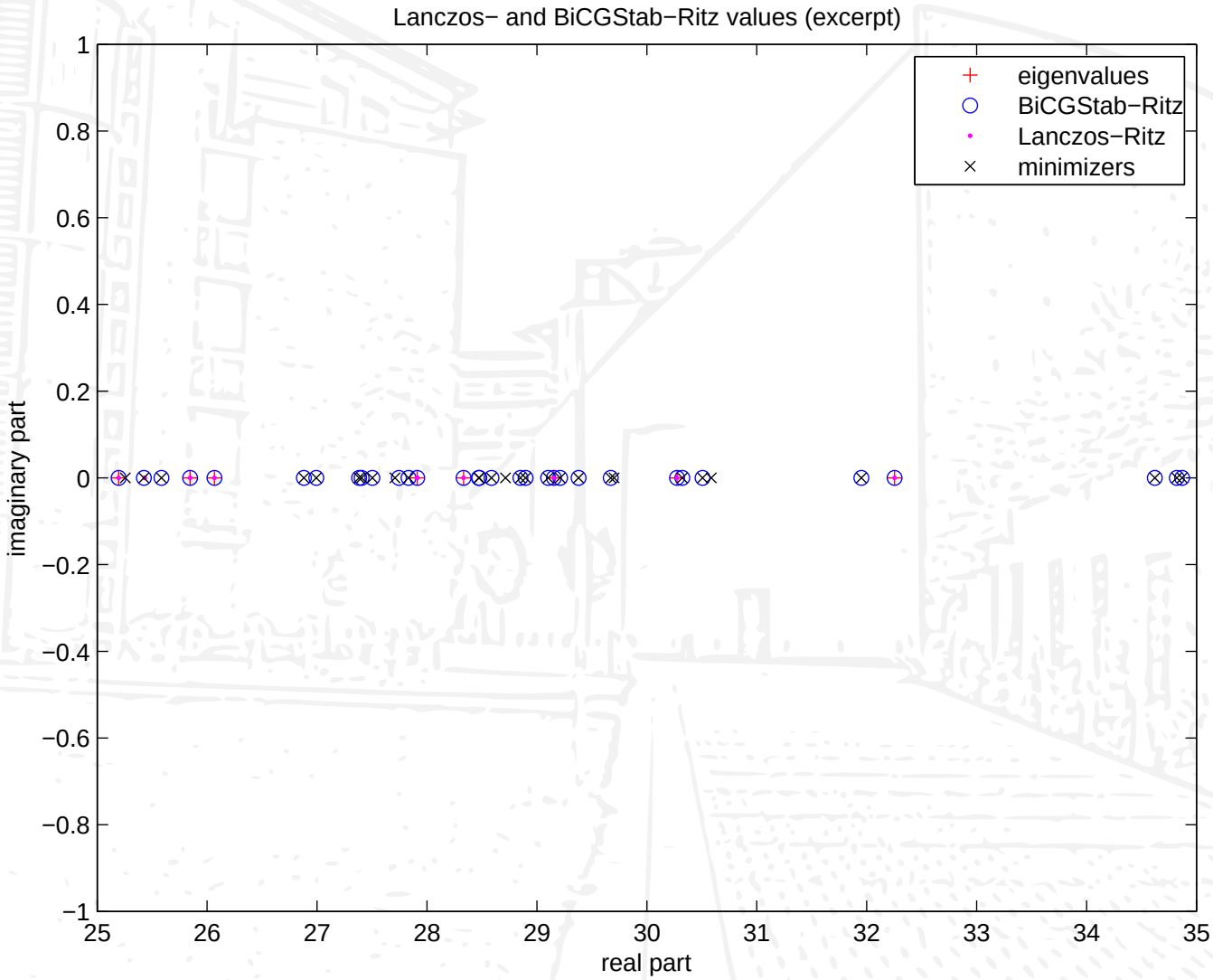
Lanczos' method in finite precision



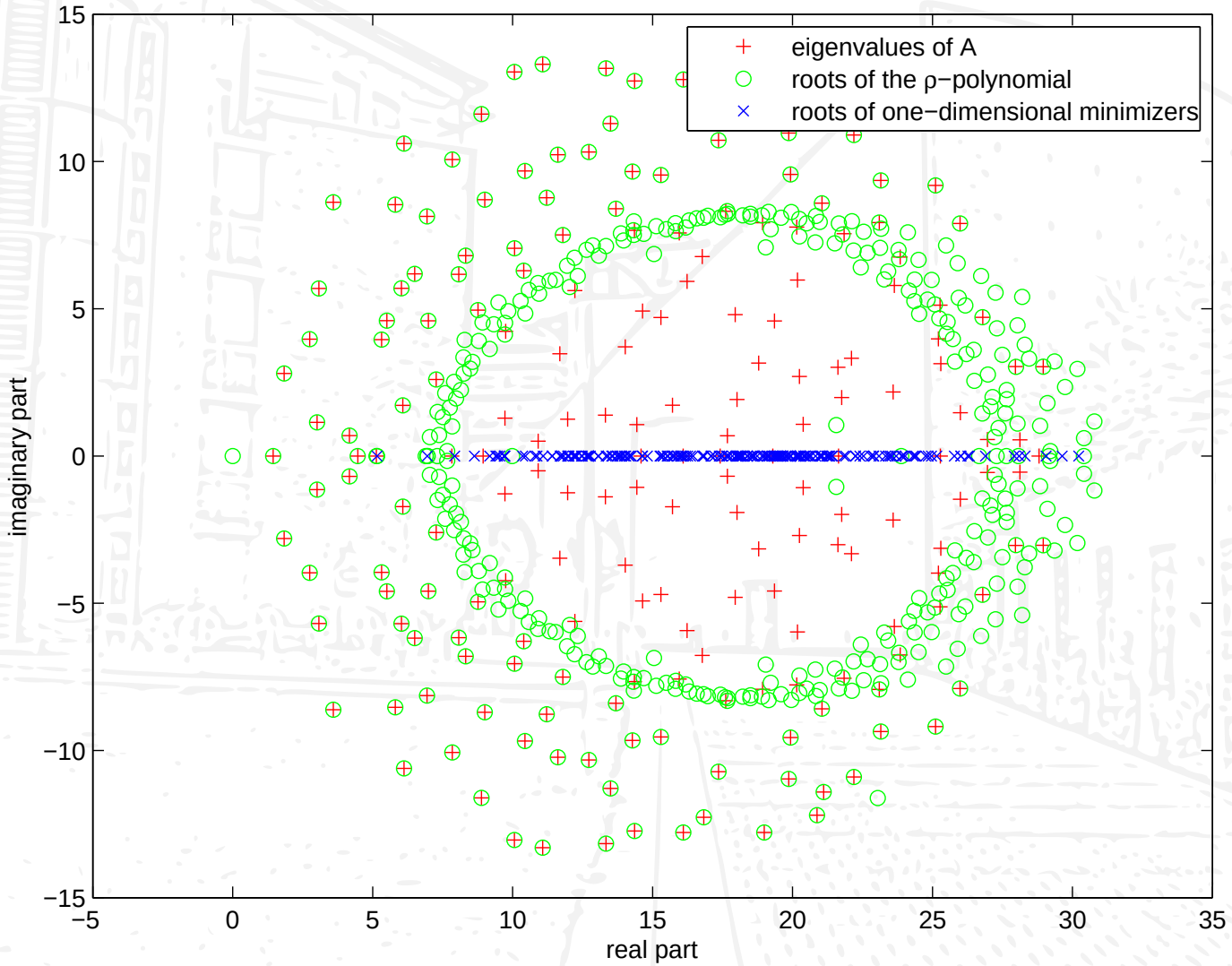
IDR, IDR(1), and BiCGStab



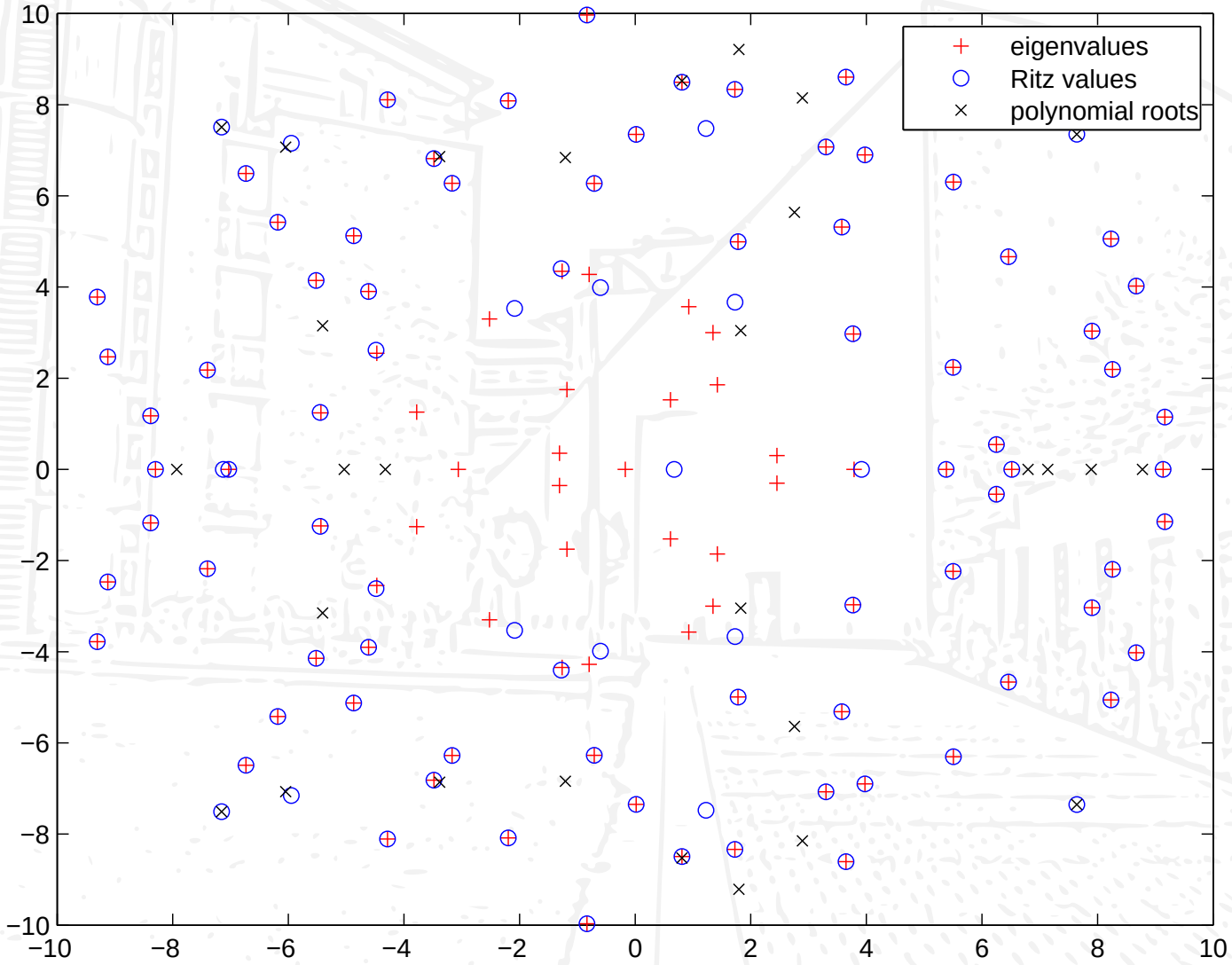
IDR, IDR(1), and BiCGStab



Understanding IDR: 600 steps for $s = 2$



“Ghost” Polynomial Roots



Some issues in IDR(s)

- IDR(s) uses larger subspaces and thus is advantageous with respect to performance (BLAS 3 instead of BLAS 1).

Some issues in IDR(s)

- IDR(s) uses larger subspaces and thus is advantageous with respect to performance (BLAS 3 instead of BLAS 1).
- Influences of round-off errors are diminished, as more information is used in the cycles.

Some issues in IDR(s)

- IDR(s) uses larger subspaces and thus is advantageous with respect to performance (BLAS 3 instead of BLAS 1).
- Influences of round-off errors are diminished, as more information is used in the cycles.
- Experiments by Seiji Fujino indicate that IDR(s) applied to SPD matrices is comparable to CG, both with the best available preconditioners, yet IDR(s) is a general purpose solver.

Some issues in IDR(s)

- IDR(s) uses larger subspaces and thus is advantageous with respect to performance (BLAS 3 instead of BLAS 1).
- Influences of round-off errors are diminished, as more information is used in the cycles.
- Experiments by Seiji Fujino indicate that IDR(s) applied to SPD matrices is comparable to CG, both with the best available preconditioners, yet IDR(s) is a general purpose solver.
- Using real values for the polynomial roots gives bad results. To use real arithmetic, IDR(s)Stab(ℓ) can be used, e.g., $\ell = 2, 4, 8$. Unfortunately, in this case $(\ell + 1) \cdot s$ vectors have to be stored.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.
- IDR/IDR(s) are short term Krylov subspace methods, but came into existence in disguise.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.
- IDR/IDR(s) are short term Krylov subspace methods, but came into existence in disguise.
- The error analysis and convergence theory of IDR(s) is much more complicated than for the classical (symmetric) Lanczos process.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.
- IDR/IDR(s) are short term Krylov subspace methods, but came into existence in disguise.
- The error analysis and convergence theory of IDR(s) is much more complicated than for the classical (symmetric) Lanczos process.
- There are no multiple Ritz values, but “ghost polynomial roots”.

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.
- IDR/IDR(s) are short term Krylov subspace methods, but came into existence in disguise.
- The error analysis and convergence theory of IDR(s) is much more complicated than for the classical (symmetric) Lanczos process.
- There are no multiple Ritz values, but “ghost polynomial roots”.
- We currently work on variants: IDREig (with Olaf Rendel and Anisa Rizvanolli); analysis of IDRStab (with Anisa Rizvanolli); QMRIDR (with Olaf Rendel, Gerard Sleijpen, and Martin van Gijzen).

Conclusion and Outview

- IDR, dating to 1976, marks the beginning of transpose-free Lanczos methods/Lanczos-type product methods (LTPM).
- IDR is the forgotten predecessor of CGS and BiCGStab.
- IDR(s) is based on Lanczos with multiple left-hand sides.
- IDR/IDR(s) are short term Krylov subspace methods, but came into existence in disguise.
- The error analysis and convergence theory of IDR(s) is much more complicated than for the classical (symmetric) Lanczos process.
- There are no multiple Ritz values, but “ghost polynomial roots”.
- We currently work on variants: IDREig (with Olaf Rendel and Anisa Rizvanolli); analysis of IDRStab (with Anisa Rizvanolli); QMRIDR (with Olaf Rendel, Gerard Sleijpen, and Martin van Gijzen).
- What about “continous” IDR?


Thank you for your attention!

Bai, Z. (1994).
Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem.
Mathematics of Computation, 62(205):209–226.

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., 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.

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.

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.