A quick and dirty introduction to IDR

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

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

May 27th, 2011
Outline

Basics

Internal guidelines
Krylov subspace methods
Hessenberg decompositions
Polynomial representations
Perturbations
Outline

Basics

Internal guidelines
Krylov subspace methods
Hessenberg decompositions
Polynomial representations
Perturbations

$\text{IDR}(s)$

IDR
$\text{IDR}(s)$
IDREig
$\text{IDR}(s)\text{Stab}(\ell)$
QMRIDR
Outline

Basics
Internal guidelines
Krylov subspace methods
Hessenberg decompositions
Polynomial representations
Perturbations

$\text{IDR}(s)$
$\text{IDR}$
$\text{IDR}(s)$
$\text{IDREig}$
$\text{IDR}(s)\text{Stab}(\ell)$
$\text{QMRIDR}$
What is the problem you’re considering?

I am trying to motivate why the method of Induced Dimension Reduction (IDR) and its generalization IDR(s) are worth considering when looking for iterative solvers for your type of problem, e.g.,

- (large sparse) linear systems: \( Ax = r_0, \ A \in \mathbb{C}^{n \times n}, \ r_0 \in \mathbb{C}^n, \) or
- (large sparse) eigenvalue problems: \( Av = \upsilon \lambda. \)
What is the problem you’re considering?

I am trying to motivate why the method of Induced Dimension Reduction (IDR) and its generalization IDR($s$) are worth considering when looking for iterative solvers for your type of problem, e.g.,

- (large sparse) linear systems: $A\mathbf{x} = \mathbf{r}_0$, $A \in \mathbb{C}^{n \times n}$, $\mathbf{r}_0 \in \mathbb{C}^n$, or
- (large sparse) eigenvalue problems: $A\mathbf{v} = \mathbf{v}\lambda$.

I have a general interest in Krylov subspace methods, for me IDR($s$) is just a new Krylov subspace method that offers interesting new possibilities.
What is the problem you’re considering?

I am trying to motivate why the method of Induced Dimension Reduction (IDR) and its generalization IDR\((s)\) are worth considering when looking for iterative solvers for your type of problem, e.g.,

- (large sparse) linear systems: \(Ax = r_0, A \in \mathbb{C}^{n \times n}, r_0 \in \mathbb{C}^n\), or
- (large sparse) eigenvalue problems: \(Av = v\lambda\).

I have a general interest in Krylov subspace methods, for me IDR\((s)\) is just a new Krylov subspace method that offers interesting new possibilities.

My personal interest lies in the error analysis of perturbed Krylov subspace methods and their convergence properties. These perturbations are

- always caused by finite precision,
- sometimes caused deliberately, e.g., in inexact methods.
Why do you find this interesting?

The error analysis of Krylov subspace methods is by no means simple:

▪ Krylov subspace methods are highly sophisticated tools,
Why do you find this interesting?

The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
- the short-term methods tend to deviate very soon but still converge, but now at another “rate” of convergence.
The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
- the short-term methods tend to deviate very soon but still converge, but now at another “rate” of convergence.

The known analysis of short term recurrence Krylov subspace methods is...
Why do you find this interesting?

The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
- the short-term methods tend to deviate very soon but still converge, but now at another “rate” of convergence.

The known analysis of short term recurrence Krylov subspace methods is

- mostly restricted to the simplest method, the symmetric Lanczos method,
The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
- the short-term methods tend to deviate very soon but still converge, but now at another “rate” of convergence.

The known analysis of short term recurrence Krylov subspace methods is

- mostly restricted to the simplest method, the symmetric Lanczos method,
- based on tools from a variety of areas that do not seem to be related to Krylov subspace methods at all,
The error analysis of Krylov subspace methods is by no means simple:

- Krylov subspace methods are highly sophisticated tools,
- most analysis is based on the fact that, in theory, Krylov subspace methods are direct methods, which no longer remains true,
- the error propagation is highly non-linear,
- the short-term methods tend to deviate very soon but still converge, but now at another “rate” of convergence.

The known analysis of short term recurrence Krylov subspace methods is

- mostly restricted to the simplest method, the symmetric Lanczos method,
- based on tools from a variety of areas that do not seem to be related to Krylov subspace methods at all,
- either for very specific implementations or does offer very little insight.
Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:
Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
- (Padé) Approximation,
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
- (Padé) Approximation,
- (Lagrange/Hermite) Interpolation,
Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
- (Padé) Approximation,
- (Lagrange/Hermite) Interpolation,
- (Formal) Orthogonal Polynomials,
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
- (Padé) Approximation,
- (Lagrange/Hermite) Interpolation,
- (Formal) Orthogonal Polynomials,
- Riemann-Stieltjes Integrals,
What is the background?

Krylov subspace methods are based on very basic ideas from Linear Algebra, namely, linear combinations, subspaces, and projections. Yet, the analysis of these methods relates them to various other interesting areas.

The tools of trade include:

- Matrix Analysis (Matrix Functions),
- Potential Theory (Green’s Functions, Capacity),
- Holomorphic Functions (Residue Theorem),
- Laurent Expansions,
- (Padé) Approximation,
- (Lagrange/Hermite) Interpolation,
- (Formal) Orthogonal Polynomials,
- Riemann-Stieltjes Integrals,
- and many, many more . . .
I will

- give a brief introduction to Krylov subspace methods,
What are you going to talk about?

I will

- give a brief introduction to Krylov subspace methods,
- present a sketch of IDR/IDR\((s)\),
What are you going to talk about?

I will

- give a brief introduction to Krylov subspace methods,
- present a sketch of IDR/IDR(s),
- explain, why it is different,
I will
▶ give a brief introduction to Krylov subspace methods,
▶ present a sketch of IDR/IDR$(s)$,
▶ explain, why it is different,
▶ report on the observed behavior,
What are you going to talk about?

I will

- give a brief introduction to Krylov subspace methods,
- present a sketch of IDR/IDR(s),
- explain, why it is different,
- report on the observed behavior,
- sketch possible generalizations.

If I succeed, you will have a feeling for some of the important aspects of IDR/IDR(s) and can read the papers on the subject for more details of particular methods.

In passing, I will note some aspects not to be found in the literature and outline some paths of possible generalizations.
What are you going to talk about?

I will

- give a brief introduction to Krylov subspace methods,
- present a sketch of IDR/IDR\((s)\),
- explain, why it is different,
- report on the observed behavior,
- sketch possible generalizations.

If I succeed, you will have a feeling for some of the important aspects of IDR/IDR\((s)\) and can read the papers on the subject for more details of particular methods.
What are you going to talk about?

I will

- give a brief introduction to Krylov subspace methods,
- present a sketch of IDR/IDR($s$),
- explain, why it is different,
- report on the observed behavior,
- sketch possible generalizations.

If I succeed, you will have a feeling for some of the important aspects of IDR/IDR($s$) and can read the papers on the subject for more details of particular methods.

In passing, I will note some aspects not to be found in the literature and outline some paths of possible generalizations.
Large linear systems are solved by projection onto smaller subspaces,

\[ Ax = r_0, \quad x_k := Q_k z_k, \quad \hat{Q}_k^H A x = (\hat{Q}_k^H A Q_k) z_k = \hat{Q}_k^H r_0. \]
Large linear systems are solved by projection onto smaller subspaces,

\[ \mathbf{Ax} = \mathbf{r}_0, \quad \mathbf{x}_k := \mathbf{Q}_k \mathbf{z}_k, \quad \hat{\mathbf{Q}}_k^H \mathbf{Ax} = (\hat{\mathbf{Q}}_k^H \mathbf{A} \mathbf{Q}_k) \mathbf{z}_k = \hat{\mathbf{Q}}_k^H \mathbf{r}_0. \]

Galärkin method:

- Bubnov-Galärkin: \( \hat{\mathbf{Q}}_k = \mathbf{Q}_k \), \( \mathbf{Q}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (orthonormal basis),
- Petrov-Galärkin: \( \hat{\mathbf{Q}}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (bi-orthonormal bases),
Large linear systems are solved by projection onto smaller subspaces,

\[ \mathbf{Ax} = \mathbf{r}_0, \quad \mathbf{x}_k := \mathbf{Q}_k \mathbf{z}_k, \quad \mathbf{Q}_k^H \mathbf{Ax} = (\mathbf{Q}_k^H \mathbf{A} \mathbf{Q}_k) \mathbf{z}_k = \mathbf{Q}_k^H \mathbf{r}_0. \]

Galärkin method:

- Bubnov-Galärkin: \( \hat{\mathbf{Q}}_k = \mathbf{Q}_k \), \( \mathbf{Q}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (orthonormal basis),
- Petrov-Galärkin: \( \hat{\mathbf{Q}}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (bi-orthonormal bases),

Subspaces of increasing dimension. As starting vector use \( \mathbf{r}_0 \), e.g.,

\[ \mathbf{Q}_1 := \mathbf{q}_1 := \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}, \quad \mathbf{H}_1 := \mathbf{Q}_1^H \mathbf{A} \mathbf{Q}_1, \quad \mathbf{z}_1 := \mathbf{H}_1^{-1} \mathbf{e}_1 \|\mathbf{r}_0\|, \quad \mathbf{x}_1 := \mathbf{Q}_1 \mathbf{z}_1. \]
Large linear systems are solved by projection onto smaller subspaces,

\[ \mathbf{Ax} = \mathbf{r}_0, \quad \mathbf{x}_k := \mathbf{Q}_k \mathbf{z}_k, \quad \hat{\mathbf{Q}}_k^H \mathbf{Ax} = (\hat{\mathbf{Q}}_k^H \mathbf{A} \mathbf{Q}_k) \mathbf{z}_k = \hat{\mathbf{Q}}_k^H \mathbf{r}_0. \]

**Galärkin method:**

- **Bubnov-Galärkin:** \( \hat{\mathbf{Q}}_k = \mathbf{Q}_k, \mathbf{Q}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (orthonormal basis),
- **Petrov-Galärkin:** \( \hat{\mathbf{Q}}_k^H \mathbf{Q}_k = \mathbf{I}_k \) (bi-orthonormal bases),

Subspaces of increasing dimension. As starting vector use \( \mathbf{r}_0 \), e.g.,

\[ \mathbf{Q}_1 := \mathbf{q}_1 := \mathbf{r}_0/\|\mathbf{r}_0\|, \quad \mathbf{H}_1 := \mathbf{Q}_1^H \mathbf{A} \mathbf{Q}_1, \quad \mathbf{z}_1 := \mathbf{H}_1^{-1} \mathbf{e}_1 \|\mathbf{r}_0\|, \quad \mathbf{x}_1 := \mathbf{Q}_1 \mathbf{z}_1. \]

Compute residual: \( \mathbf{r}_1 := \mathbf{r}_0 - \mathbf{A} \mathbf{x}_1 = \mathbf{Q}_1 \mathbf{e}_1 \|\mathbf{r}_0\| - \mathbf{A} \mathbf{Q}_1 \mathbf{z}_1. \)
Background

Large linear systems are solved by projection onto smaller subspaces,

\[ Ax = r_0, \quad x_k := Q_k z_k, \quad \hat{Q}_k^H A x = (\hat{Q}_k^H A Q_k) z_k = \hat{Q}_k^H r_0. \]

Galärkin method:

- Bubnov-Galärkin: \( \hat{Q}_k = Q_k, \) \( Q_k^H Q_k = I_k \) (orthonormal basis),
- Petrov-Galärkin: \( \hat{Q}_k^H Q_k = I_k \) (bi-orthonormal bases),

Subspaces of increasing dimension. As starting vector use \( r_0, \) e.g.,

\[ Q_1 := q_1 := r_0 / \| r_0 \|, \quad H_1 := Q_1^H A Q_1, \quad z_1 := H_1^{-1} e_1 \| r_0 \|, \quad x_1 := Q_1 z_1. \]

Compute residual: \( r_1 := r_0 - A x_1 = Q_1 e_1 \| r_0 \| - A Q_1 z_1. \) Both steps involve \( A q_1. \)

Expand space:

\[ K_2 := \text{span} \{ r_0, A r_0 \} = \text{span} \{ q_1, q_2 \}. \]
Krylov subspaces

Natural generalization of this simple idea: Krylov subspaces. Obtained by multiplication of last basis vector by $A$,

$$\mathcal{K}_k := \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} = \text{span} \{ q_1, q_2, \ldots, q_k \}. $$
Krylov subspaces

Natural generalization of this simple idea: Krylov subspaces. Obtained by multiplication of last basis vector by \( A \),

\[
\mathcal{K}_k := \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} = \text{span} \{ q_1, q_2, \ldots, q_k \}.
\]

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

\[
x \in \mathcal{K}_k \iff x = \sum_{j=0}^{k-1} A^j r_0 c_{j+1} = p_{k-1}(A) r_0, \quad p_{k-1}(z) = \sum_{j=0}^{k-1} c_{j+1} z^j.
\]
Krylov subspaces

Natural generalization of this simple idea: Krylov subspaces. Obtained by multiplication of last basis vector by $A$,

$$\mathcal{K}_k := \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} = \text{span} \{ q_1, q_2, \ldots, q_k \}.$$ 

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

$$x \in \mathcal{K}_k \iff x = \sum_{j=0}^{k-1} A^j r_0 c_{j+1} = p_{k-1}(A)r_0, \quad p_{k-1}(z) = \sum_{j=0}^{k-1} c_{j+1} z^j.$$ 

Residual polynomials are polynomials that

- satisfy $r_k = \rho_k(A)r_0$ and
- are normalized by the condition $\rho_k(0) = 1$. 

Krylov subspaces

Natural generalization of this simple idea: Krylov subspaces. Obtained by multiplication of last basis vector by \( A \),

\[
\mathcal{K}_k := \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} = \text{span} \{ q_1, q_2, \ldots, q_k \}.
\]

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

\[
x \in \mathcal{K}_k \iff x = \sum_{j=0}^{k-1} A^j r_0 c_{j+1} = p_{k-1}(A)r_0, \quad p_{k-1}(z) = \sum_{j=0}^{k-1} c_{j+1} z^j.
\]

Residual polynomials are polynomials that

- satisfy \( r_k = \rho_k(A)r_0 \) and
- are normalized by the condition \( \rho_k(0) = 1 \).

Residual polynomials arise because

\[
r_k := r_0 - Ax_k = (I - A p_{k-1}(A))r_0 =: \rho_k(A)r_0.
\]
There are mainly two classes of Krylov subspace methods:

- long-term (Hessenberg, Arnoldi),
- short-term (Lanczos).
Krylov subspace methods

There are mainly two classes of Krylov subspace methods:

- long-term (Hessenberg, Arnoldi),
- short-term (Lanczos).

Arnoldi: Example of a long-term method building an orthonormal basis.
There are mainly two classes of Krylov subspace methods:

- long-term (Hessenberg, Arnoldi),
- short-term (Lanczos).

Arnoldi: Example of a long-term method building an orthonormal basis.

\[
\begin{align*}
    r &= r_0, \quad q = r / \|r\| \\
    Q &= q, \quad H = () \\
    &\text{for } k = 1, \ldots \\
    r &= Aq \\
    c &= Q^H r \\
    r &= r - Qc \\
    H &= (H, c; o^T, \|r\|) \\
    q &= r / \|r\| \\
    Q &= (Q, q)
\end{align*}
\]
The construction of basis vectors is resembled in the structure of the arising Hessenberg decomposition

\[ AQ_k = Q_{k+1} H_k, \]

where

- \( Q_{k+1} = (Q_k, q_{k+1}) \in \mathbb{C}^{n \times (k+1)} \) collects the basis vectors,
- \( H_k \in \mathbb{C}^{(k+1) \times k} \) is an unreduced extended Hessenberg matrix.
**Hessenberg decompositions**

The construction of basis vectors is resembled in the structure of the arising Hessenberg decomposition

\[ AQ_k = Q_{k+1}H_k, \]

where

- \( Q_{k+1} = (Q_k, q_{k+1}) \in \mathbb{C}^{n \times (k+1)} \) collects the basis vectors,
- \( H_k \in \mathbb{C}^{(k+1) \times k} \) is an unreduced extended Hessenberg matrix.

Aspects of perturbed Krylov subspace methods can be captured with perturbed Hessenberg decompositions

\[ AQ_k + F_k = Q_{k+1}H_k, \]

where \( F_k \in \mathbb{C}^{n \times k} \) accounts for the perturbations.
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 $\mathbf{z}^{(v)} (v = 1,2,\ldots,n)$ ebenfalls in einer Matrix zusammenfassen, und zwar ist nach Gleichung (55) und (56)

$$\mathbf{z}^{(v)} = \mathbf{z}^{(1)} \cdot \mathbf{p},$$

worin die Matrix $\mathbf{p}$ zur Abkürzung gesetzt ist für

$$\mathbf{p} = \begin{pmatrix}
\alpha_{0} & \alpha_{1} & \cdots & \alpha_{n-1} & \alpha_{n} \\
1 & \alpha_{0} & \cdots & \alpha_{n-2} & \alpha_{n-1} \\
0 & 1 & \cdots & \alpha_{n-2} & \alpha_{n-1} \\
0 & 0 & \cdots & \alpha_{n-2} & \alpha_{n-1} \\
\vdots & \vdots & & \vdots & \vdots
\end{pmatrix}.$$ 

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

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

The vectors from Krylov subspaces can be described in terms of polynomials. This representation carries over to the perturbed case with minor changes.
Important Polynomials

The vectors from Krylov subspaces can be described in terms of polynomials. This representation carries over to the perturbed case with minor changes.

The residuals of the OR approximation $x_k := Q_k z_k$ and the MR approximation $x_k := Q_k z_k$ with coefficient vectors $z_k := H_k^{-1} e_1 \|r_0\|$ and $z_k := H_k^\dagger e_1 \|r_0\|$ satisfy

$$r_k := r_0 - A x_k = R_k(A) r_0$$
$$r_k := r_0 - A x_k = R_k(A) r_0$$

with residual polynomials $R_k$ and $R_k$ given by

$$R_k(z) := \det (I_k - z H_k^{-1})$$
$$R_k(z) := \det (I_k - z H_k^\dagger I_k).$$
Important Polynomials

The vectors from Krylov subspaces can be described in terms of polynomials. This representation carries over to the perturbed case with minor changes.

The residuals of the OR approximation \( x_k := Q_k z_k \) and the MR approximation \( x_k := Q_k z_k \) with coefficient vectors

\[
z_k := H_k^{-1} e_1 \| r_0 \| \quad \text{and} \quad z_k := H_k^\dagger e_1 \| r_0 \|
\]

satisfy

\[
r_k := r_0 - Ax_k = R_k(A) r_0 \quad \text{and} \quad r_k := r_0 - Ax^k = \overline{R_k}(A) r_0
\]

with residual polynomials \( R_k \) and \( \overline{R_k} \) given by

\[
R_k(z) := \det (I_k - zH_k^{-1}) \quad \text{and} \quad \overline{R_k}(z) := \det (I_k - zH_k^\dagger I_k).
\]

The convergence of OR and MR depends on the Ritz and harmonic Ritz values, respectively.
We sketch briefly how the setting changes when perturbations enter the stage in the special case of an OR method.

In the perturbed case, we have:

\[ \mathbf{Q}_k + \mathbf{F}_k = \mathbf{Q}_k + 1 \mathbf{H}_k \]

under the assumption that all trailing square Hessenberg matrices are regular, the polynomial representation for the OR residuals changes to:

\[ r_k = \mathbf{R}_k(A)r_0 - k \sum_{\ell=1}^{k} z_{\ell}^{k} \mathbf{R}_{\ell+1}^{k} : k(A)f_{\ell} + \mathbf{F}_k z_k, \]

where \( \mathbf{R}_{\ell+1}^{k}(A) : (z_{\ell}) \) := \det \left( \mathbf{I}_k - \ell - z H_{\ell+1}^{k} \right). \]

We can expect convergence when \( \mathbf{F}_k z_k \) remains bounded (inexact methods) and all \( \mathbf{R}_{\ell+1}^{k}(A) \) are "small".
We sketch briefly how the setting changes when perturbations enter the stage in the special case of an OR method.

In the perturbed case

\[ \mathbf{A} \mathbf{Q}_k + \mathbf{F}_k = \mathbf{Q}_{k+1} \mathbf{H}_k \]

under the assumption that all trailing square Hessenberg matrices are regular, the polynomial representation for the OR residuals changes to

\[ \mathbf{r}_k = \mathcal{R}_k(\mathbf{A})\mathbf{r}_0 - \sum_{\ell=1}^{k} z_{\ell k} \mathcal{R}_{\ell+1:k}(\mathbf{A})\mathbf{f}_\ell + \mathbf{F}_k \mathbf{z}_k, \]
Perturbed OR methods

We sketch briefly how the setting changes when perturbations enter the stage in the special case of an OR method.

In the perturbed case

\[ AQ_k + F_k = Q_{k+1} H_k \]

under the assumption that all trailing square Hessenberg matrices are regular, the polynomial representation for the OR residuals changes to

\[ r_k = R_k(A)r_0 - \sum_{\ell=1}^{k} z_\ell k R_{\ell+1:k}(A)f_\ell + F_k z_k, \]

where

\[ R_{\ell+1:k}(z) := \det (I_{k-\ell} - zH_{\ell+1:k}^{-1}). \]
Perturbed OR methods

We sketch briefly how the setting changes when perturbations enter the stage in the special case of an OR method.

In the perturbed case

\[ AQ_k + F_k = Q_{k+1}H_k \]

under the assumption that all trailing square Hessenberg matrices are regular, the polynomial representation for the OR residuals changes to

\[ r_k = R_k(A)r_0 - \sum_{\ell=1}^{k} z_{\ell k} R_{\ell+1:k}(A)f_{\ell} + F_kz_k, \]

where

\[ R_{\ell+1:k}(z) := \det (I_{k-\ell} - zH_{\ell+1:k}^{-1}). \]

We can expect convergence when \( F_kz_k \) remains bounded (inexact methods) and all \( R_{\ell+1:k}(A) \) are “small”.
Outline

Basics
- Internal guidelines
- Krylov subspace methods
- Hessenberg decompositions
- Polynomial representations
- Perturbations

$\text{IDR}^{(s)}$
- IDR
- $\text{IDR}^{(s)}$
- IDREig
- $\text{IDR}^{(s)}\text{Stab}(\ell)$
- QMRIDR
Birth of a method

In 1976, Peter Sonneveld of TU Delft “stumbled upon” the three-term recurrence

\[ r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})), \quad \text{where} \quad \gamma_k := \frac{p^H r_k}{p^H(r_{k-1} - r_k)}. \]
Birth of a method

In 1976, Peter Sonneveld of TU Delft “stumbled upon” the three-term recurrence

\[ r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})) \]

where \( \gamma_k := \frac{p^H r_k}{p^H(r_{k-1} - r_k)} \).

This recurrence (almost) always results in the zero vector after \( 2n \) steps, where \( A \in \mathbb{C}^{n \times n} \) and \( r_0 \in \mathbb{C}^n \), \( r_1 = Ar_0 \), and \( p \in \mathbb{C}^n \) are arbitrarily chosen.
Birth of a method

In 1976, Peter Sonneveld of TU Delft “stumbled upon” the three-term recurrence

\[ r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})) , \quad \text{where} \quad \gamma_k := \frac{p^H r_k}{p^H(r_{k-1} - r_k)}. \]

This recurrence (almost) always results in the zero vector after \(2n\) steps, where \(A \in \mathbb{C}^{n \times n}\) and \(r_0 \in \mathbb{C}^n\), \(r_1 = Ar_0\), and \(p \in \mathbb{C}^n\) are arbitrarily chosen.

He realized that the recurrence constructs vectors in spaces \(G_j\) of shrinking dimensions:

\[ G_0 := \mathcal{K}(A, r_0) = \text{span} \{r_0, Ar_0, A^2r_0, \ldots\} \]

\[ G_j := (I - A)(G_{j-1} \cap S), \quad S = \text{span} \{p\}^\perp, \quad j = 1, \ldots \]
Birth of a method

In 1976, Peter Sonneveld of TU Delft “stumbled upon” the three-term recurrence

\[ r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})) \]

where \( \gamma_k := \frac{p^H r_k}{p^H(r_{k-1} - r_k)} \).

This recurrence (almost) always results in the zero vector after \( 2n \) steps, where \( A \in \mathbb{C}^{n \times n} \) and \( r_0 \in \mathbb{C}^n \), \( r_1 = Ar_0 \), and \( p \in \mathbb{C}^n \) are arbitrarily chosen.

He realized that the recurrence constructs vectors in spaces \( G_j \) of shrinking dimensions:

\[
G_0 := K(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots\}
\]

\[
G_j := (I - A)(G_{j-1} \cap S), \quad S = \text{span}\{p\}^\perp, \quad j = 1, \ldots
\]

More precisely,

\[ r_{2j}, r_{2j+1} \in G_j, \quad j = 0, 1, \ldots \]
With $r_0 := b - Ax_0$, the Richardson iteration is carried out as follows:

$$x_{k+1} = x_k + r_k, \quad r_{k+1} = (I - A)r_k.$$
The origin of IDR: primitive IDR

With $r_0 := b - Ax_0$, the Richardson iteration is carried out as follows:

$$x_{k+1} = x_k + r_k, \quad r_{k+1} = (I - A)r_k.$$ 

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

$$r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})), \quad \gamma_k = \frac{p^H r_k}{p^H(r_{k-1} - r_k)}.$$
The origin of IDR: primitive IDR

With $r_0 := b - Ax_0$, the Richardson iteration is carried out as follows:

$$x_{k+1} = x_k + r_k, \quad r_{k+1} = (I - A)r_k.$$

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

$$r_{k+1} = (I - A)(r_k + \gamma_k(r_k - r_{k-1})), \quad \gamma_k = \frac{p^H r_k}{p^H(r_{k-1} - r_k)}.$$

The update of the iterates has to be modified accordingly,

$$-A(x_{k+1} - x_k) = r_{k+1} - r_k = (I - A)(r_k + \gamma_k(r_k - r_{k-1})) - r_k$$

$$= (I - A)(r_k - \gamma_k A(x_k - x_{k-1})) - r_k$$

$$= -A(r_k + \gamma_k(I - A)(x_k - x_{k-1}))$$

$$\Leftrightarrow \quad x_{k+1} - x_k = r_k + \gamma_k(I - A)(x_k - x_{k-1})$$

$$= r_k + \gamma_k(x_k - x_{k-1} + r_k - r_{k-1}).$$
The origin of IDR: primitive IDR

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

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

\[
\begin{align*}
\gamma_k &= \frac{p^T r_k}{p^T(r_{k-1} - r_k)} \\
 s_k &= r_k + \gamma_k (r_k - r_{k-1}) \\
x_{k+1} &= x_k + \gamma_k (x_k - x_{k-1}) + s_k \\
r_{k+1} &= s_k - As_k
\end{align*}
\]

done
The origin of IDR: primitive IDR

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

\[ r_0 = b - Ax_0 \]
\[ x_1 = x_0 + r_0 \]
\[ r_1 = r_0 - Ar_0 \]

For \( k = 1, 2, \ldots \) do

\[ \gamma_k = \frac{p^T r_k}{p^T (r_{k-1} - r_k)} \]
\[ s_k = r_k + \gamma_k (r_k - r_{k-1}) \]
\[ x_{k+1} = x_k + \gamma_k (x_k - x_{k-1}) + s_k \]
\[ r_{k+1} = s_k - As_k \]

done

\[ x_{\text{old}} = x_0 \]
\[ r_{\text{old}} = b - Ax_{\text{old}} \]
\[ x_{\text{new}} = x_{\text{old}} + r_{\text{old}} \]
\[ r_{\text{new}} = r_{\text{old}} - Ar_{\text{old}} \]

While “not converged” do

\[ \gamma = \frac{p^T r_{\text{new}}}{p^T (r_{\text{old}} - r_{\text{new}})} \]
\[ s = r_{\text{new}} + \gamma (r_{\text{new}} - r_{\text{old}}) \]
\[ x_{\text{tmp}} = x_{\text{new}} + \gamma (x_{\text{new}} - x_{\text{old}}) + s \]
\[ r_{\text{tmp}} = s - As \]
\[ x_{\text{old}} = x_{\text{new}}, x_{\text{new}} = x_{\text{tmp}} \]
\[ r_{\text{old}} = r_{\text{new}}, r_{\text{new}} = r_{\text{tmp}} \]

done
The origin of IDR: primitive IDR

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

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

\[ \gamma_k = p^T r_k / p^T (r_{k-1} - r_k) \]
\[ s_k = r_k + \gamma_k (r_k - r_{k-1}) \]
\[ x_{k+1} = x_k + \gamma_k (x_k - x_{k-1}) + s_k \]
\[ r_{k+1} = s_k - As_k \]

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.

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 $I - \omega_j A$, where $\omega_j$ 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.
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 $I - \omega_j A$, where $\omega_j$ 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 $\omega_j$ 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.
A numerical comparison of Richardson iteration, original IDR, and PIA.
Later, Peter Sonneveld developed CGS based on the ideas behind IDR and, together with Henk van der Vorst, rewrote the IDR variant to one that explicitly constructs the coefficients of the underlying Lanczos recurrence.
Later, Peter Sonneveld developed **CGS** based on the ideas behind IDR and, together with Henk van der Vorst, rewrote the IDR variant to one that explicitly constructs the coefficients of the underlying Lanczos recurrence.

This rewritten variant was published by Henk van der Vorst under the name **BiCGStab**.
Later, Peter Sonneveld developed CGS based on the ideas behind IDR and, together with Henk van der Vorst, rewrote the IDR variant to one that explicitly constructs the coefficients of the underlying Lanczos recurrence.

This rewritten variant was published by Henk van der Vorst under the name BiCGStab.

In short: BiCGStab is (almost mathematically equivalent to) IDR.
IDR can be generalized: instead of using one hyperplane \((\text{span}\{p\})^\perp\), one uses the intersection of \(s\) hyperplanes. This makes the dimension reduction step less frequent but the reduction a larger one.

This generalized IDR, termed IDR\((s)\), was developed in 2006 by Peter Sonneveld and Martin van Gijzen. In the context of Krylov subspace methods, IDR\((s)\) can be thought of as a two-sided Lanczos method. There is a predecessor to such a method, namely, ML\((k)\)BiCGStab by Man-Chung Yeung and Tony Chan.
IDR can be generalized: instead of using one hyperplane \((\text{span } \{p\})^\perp\), one uses the intersection of \(s\) hyperplanes. This makes the dimension reduction step less frequent but the reduction a larger one.

This generalized IDR, termed IDR\((s)\), was developed in 2006 by Peter Sonneveld and Martin van Gijzen.
IDR can be generalized: instead of using one hyperplane \((\text{span}\{p\})^\perp\), one uses the intersection of \(s\) hyperplanes. This makes the dimension reduction step less frequent but the reduction a larger one.

This generalized IDR, termed IDR\((s)\), was developed in 2006 by Peter Sonneveld and Martin van Gijzen.

In the context of Krylov subspace methods, IDR\((s)\) can be thought of as a two-sided Lanczos method. There is a predecessor to such a method, namely, ML\((k)\)BiCGStab by Man-Chung Yeung and Tony Chan.
IDR($s$) is a Krylov subspace method based on two building blocks:

- Multiplication by polynomials in $A$.
  (IDR($s$): linear, IDR($s$)\text{Stab($\ell$)}: higher degree)
- Oblique projection perpendicular to $P \in \mathbb{C}^{n \times s}$.
Building blocks of IDR\(_{(s)}\)

IDR\(_{(s)}\) is a Krylov subspace method based on two building blocks:

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

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 $A$.
  (IDR($s$): linear, IDR($s$)Stab($\ell$): higher degree)
▶ Oblique projection perpendicular to $P \in \mathbb{C}^{n \times s}$.

IDR($s$) constructs nested subspaces of shrinking dimensions.

The prototype IDR($s$) method constructs spaces $G_j$ as follows:

▶ Define $G_0 := K(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots\}$.
▶ Iterate $G_j := (I - \omega_jA)(G_{j-1} \cap S)$, $j = 1, 2, \ldots$, $\mathbb{C} \ni \omega_j \neq 0$
Building blocks of IDR($s$)

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

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

IDR($s$) constructs nested subspaces of shrinking dimensions.

The prototype IDR($s$) method constructs spaces $G_j$ as follows:

- Define $G_0 := K(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots\}$.
- Iterate $G_j := (I - \omega_j A)(G_{j-1} \cap S)$, $j = 1, 2, \ldots$, $\mathbb{C} \ni \omega_j \neq 0$

Only sufficiently many vectors in each space are constructed.
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.
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
  \[
  r_{j(s+1)+k}^{\text{IDR}} = \Omega_j(A)\rho_{js+k}(A)r_0, \quad 1 \leq k \leq s
  \]
  where \(\rho_{js+k}\) 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

$$r_{j(s+1)+k}^{IDR} = \Omega_j(A)\rho_{js+k}(A)r_0, \quad 1 \leq k \leq s$$

where $\rho_{js+k}$ are residual polynomials of the Lanczos process.

Reminder: Residual polynomials are polynomials that

- satisfy $r_k = \rho_k(A)r_0$ and
- are normalized by the condition $\rho_k(0) = 1$. 
Generalized Hessenberg decomposition

IDR(\(s\)) can be captured using a generalized Hessenberg decomposition

\[
AQ_k U_k = Q_{k+1} H_k.
\]
Generalized Hessenberg decomposition

IDR(\(s\)) can be captured using a generalized Hessenberg decomposition

\[
AQ_k U_k = Q_{k+1} H_k.
\]

IDR based methods include BiCGStab (rewritten version of IDR), and CGS.
IDR(s) can be captured using a generalized Hessenberg decomposition

\[ AQ_k U_k = Q_{k+1} H_k. \]

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

\[ x_k := Q_k U_k z_k, \quad z_k := H_k^{-1} e_1 \| r_0 \|, \]

Tacitly assuming \( \| q_{k+1} \| = 1 \), we have \( \| r_k \| = | h_k + 1 |. \)
Generalized Hessenberg decomposition

IDR\(_(s)\) can be captured using a generalized Hessenberg decomposition

\[ AQ_k U_k = Q_{k+1} H_k. \]

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

OR based IDR methods use

\[ x_k := Q_k U_k z_k, \quad z_k := H_k^{-1} e_1 \| r_0 \|, \]

the residual is described by

\[ r_k := r_0 - A x_k = r_0 - AQ_k U_k z_k = r_0 - Q_{k+1} H_k z_k \]
\[ = Q_k (e_1 \| r_0 \| - H_k z_k) - q_{k+1} h_{k+1} e_k^T z_k \]
\[ = \mathcal{R}_k(A) r_0, \quad \mathcal{R}_k(z) := \det (I_k - z U_k H_k^{-1}). \]
Generalized Hessenberg decomposition

IDR(\(s\)) can be captured using a generalized Hessenberg decomposition

\[ AQ_k U_k = Q_{k+1} H_k. \]

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

OR based IDR methods use

\[ x_k := Q_k U_k z_k, \quad z_k := H_k^{-1} e_1 \| r_0 \|, \]

the residual is described by

\[ r_k := r_0 - Ax_k = r_0 - AQ_k U_k z_k = r_0 - Q_{k+1} H_k z_k \]
\[ = Q_k (e_1 \| r_0 \| - H_k z_k) - q_{k+1} h_{k+1,k} e_k^T z_k \]
\[ = R_k(A) r_0, \quad R_k(z) := \det (I_k - z U_k H_k^{-1}). \]

Tacitly assuming \( \| q_{k+1} \| = 1 \), we have \( \| r_k \| = | h_{k+1,k} z_k |. \)
We consider the prototype IDR(s) by Sonneveld/van Gijzen (IDR(s)OREs).
We consider the prototype IDR\((s)\) by Sonneveld/van Gijzen (IDR\((s)\)OREs).

The IDR\((s)\)OREs pencil, the so-called Sonneveld pencil \((Y_n^\circ, Y_n D_\omega^{(n)})\), can be depicted by

\[
\begin{pmatrix}
\times \times \times \times \times \times \times \times \times \times \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{pmatrix}
\quad,
\begin{pmatrix}
\times \times \times \times \times \times \times \times \times \times \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{pmatrix}
\]
We consider the prototype IDR\((s)\) by Sonneveld/van Gijzen (IDR\((s)\)ORes).

The IDR\((s)\)ORes pencil, the so-called Sonneveld pencil \((Y_n^\circ, Y_n D_\omega^{(n)})\), can be depicted by

\[
\begin{pmatrix}
\times \times \times \circ \circ \circ \circ \circ \\
+ \times \times \times \circ \circ \circ \circ \circ \\
\circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ + \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ + \times \times \times \\
\end{pmatrix}, \quad \begin{pmatrix}
\times \times \times \circ \circ \circ \circ \circ \\
\circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \times \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \times \times \times \\
\end{pmatrix}.
\]

The upper triangular matrix \(Y_n D_\omega^{(n)}\) 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.
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 $(Y_n^\circ, U_n D^{(n)}_{\omega})$, that has only the remaining eigenvalues and some infinite ones as eigenvalues, can be depicted by

$$
\begin{pmatrix}
\times & \times & \times & \times & \circ & \circ & \circ & \circ \\
+ & \times & \times & \times & \circ & \circ & \circ & \circ \\
\circ & + & \times & \times & \times & \circ & \circ & \circ \\
\circ & \circ & + & \times & \times & \times & \circ & \circ \\
\circ & \circ & \circ & + & \times & \times & \times & \circ \\
\circ & \circ & \circ & \circ & + & \times & \times & \times \\
\circ & \circ & \circ & \circ & \circ & + & \times & \times \\
\circ & \circ & \circ & \circ & \circ & \circ & + & \times \\
\end{pmatrix}
\begin{pmatrix}
\times & \times & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \times & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \times & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \times & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \times & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \times & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \times & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\end{pmatrix}.
$$

We get rid of the infinite eigenvalues using a change of basis (Gauß/Schur).
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 $(Y_n^\circ, U_n D^{(n)}_\omega)$, 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).
The deflated purified IDR(s)OREs pencil, after the elimination step \((Y_n^o G_n, U_n D^{(n)}_\omega)\), can be depicted by

\[
\begin{pmatrix}
\times \times \times \times \times \times \circ \circ \circ \circ \\
\times \times \times \times \times \times \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{pmatrix}
\]

and

\[
\begin{pmatrix}
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\times \times \times \times \times \times \times \times \times \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{pmatrix}
\]
The deflated purified IDR\((s)\)ORes pencil, after the elimination step \((Y_n^\circ G_n, U_n D^{(n)}_\omega)\), can be depicted by

\[
\begin{pmatrix}
\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\times\time
Let $D$ denote an **deflation operator** that removes every $(s + 1)$th column and row from the matrix the operator is applied to.
Let $D$ denote an \textbf{deflation operator} that removes every $(s+1)$th column and row from the matrix the operator is applied to.

The \textbf{deflated purified IDR($s$)ORes pencil}, after the deflation step $(D(Y_n^\circ G_n), D(U_nD_\omega^n))$, can be depicted by

\[\begin{bmatrix}
\times \times \times \times \times \circ \circ \circ \\
+ \times \times \times \times \circ \circ \circ \\
\circ + \times \times \times \circ \circ \circ \\
\circ \circ + \times \times \times \times \times \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{bmatrix}, \quad \begin{bmatrix}
\times \times \times \circ \circ \circ \circ \circ \\
\circ \times \times \circ \circ \circ \circ \circ \\
\circ \circ \times \times \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\circ \circ \circ \circ \circ \circ \circ \circ \circ \\
\end{bmatrix}.\]
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 $(D(Y_n^oG_n), D(U_nD_\omega^{(n)}))$, can be depicted by

$$
\begin{pmatrix}
\times\times\times\times\times\times\circ\circ\circ
+\times\times\times\times\times\times\circ\circ\circ
\circ+\times\times\times\times\times\times\circ\circ\circ
\circ\circ+\times\times\times\times\times\times\times
\circ\circ\circ+\times\times\times\times\times\times
\circ\circ\circ\circ+\times\times\times\times\times
\circ\circ\circ\circ\circ+\times\times\times
\circ\circ\circ\circ\circ\circ+\times\times
\end{pmatrix},
\begin{pmatrix}
\times\times\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\circ\circ\circ\circ\circ\circ\circ
\end{pmatrix}.
$$

The block-diagonal matrix $D(U_nD_\omega^{(n)})$ has invertible upper triangular blocks and can be inverted to expose the underlying **Lanczos process**.
Inverting the block-diagonal matrix $D(U_n D^{(n)}_\omega)$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix

$$L_n := D(Y^\circ_n G_n) \cdot D(U_n D^{(n)}_\omega)^{-1} = \begin{pmatrix}
\times & \times & \times & \times & \times & \circ & \circ \\
+ & \times & \times & \times & \times & \circ & \circ \\
\circ & + & \times & \times & \times & \times & \times \\
\circ & \circ & + & \times & \times & \times & \times \\
\circ & \circ & \circ & + & \times & \times & \times \\
\circ & \circ & \circ & \circ & + & \times & \times \\
\circ & \circ & \circ & \circ & \circ & + & \times
\end{pmatrix}.$$
IDR: a Lanczos process with multiple left-hand sides

Inverting the block-diagonal matrix $D(U_n D^{(n)}_\omega)$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix $L_n := D(Y_n G_n) \cdot D(U_n D^{(n)}_\omega)^{-1}$. 

This is the matrix of the underlying BiORes($s, 1$) process.
Inverting the block-diagonal matrix $D(U_n D^{(n)}(\omega))$ gives an algebraic eigenvalue problem with a block-tridiagonal unreduced upper Hessenberg matrix

$$L_n := D(Y^n G_n) \cdot D(U_n D^{(n)}(\omega))^{-1} = \begin{pmatrix}
\times\times\times\times\times\circ\circ\circ \\
+\times\times\times\times\times\circ\circ\circ \\
\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ\circ\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ\circ\circ\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ\circ\circ\circ\circ+\times\times\times\times\times\circ\circ\circ \\
\circ\circ\circ\circ\circ\circ\circ+\times\times\times\times\times\circ\circ\circ
\end{pmatrix}.$$

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

This matrix (in the extended version) satisfies

$$AQ_n = Q_{n+1} L_n,$$

where the reduced residuals $q_{js+k}$, $k = 0, \ldots, s - 1$, $j = 0, 1, \ldots$, are given by

$$\Omega_j(A) q_{js+k} = r_{j(s+1)+k}.$$
The eigenvalues of the pencil \((H_k, U_k)\) are the roots of the residual polynomials and some of these converge to eigenvalues of \(A\).
The eigenvalues of the pencil \((H_k, U_k)\) are the roots of the residual polynomials and some of these converge to eigenvalues of \(A\).

Suppose that \(Q_{k+1}\) has full rank. The pencil \((H_k, U_k)\) arises as a oblique projection of \((A, I_n)\), as

\[
\hat{Q}_k^H(A, I_n)Q_kU_k = \hat{Q}_k^H(AQ_kU_k, Q_kU_k) = \hat{Q}_k^H(Q_{k+1}H_k, Q_kU_k) = (I_k^T H_k, U_k) = (H_k, U_k),
\]

where \(\hat{Q}_k^H := I_k^T Q_k^\dagger\).
The eigenvalues of the pencil \((H_k, U_k)\) are the roots of the residual polynomials and some of these converge to eigenvalues of \(A\).

Suppose that \(Q_k + 1\) has full rank. The pencil \((H_k, U_k)\) arises as a oblique projection of \((A, I_n)\), as

\[
\hat{Q}_k^H A Q_k U_k = \hat{Q}_k^H (A Q_k U_k, Q_k U_k) = \hat{Q}_k^H (Q_{k+1} H_k, Q_k U_k) = (I_k^T H_k, U_k) = (H_k, U_k),
\]

where \(\hat{Q}_k^H := I_k^T Q_{k+1}^\dagger\).

One uses a deflated pencil that only gives the Ritz values. The theory was developed by Martin Gutknecht and Z. (2010), currently we investigate how to select parameters \((s, \omega_j, P)\) to obtain good eigenpair approximations (this is ongoing joint work with Olaf Rendel and Anisa Rizvanolli).
Recently, IDR(s) was generalized by combining ideas from IDR(s) with the higher dimensional minimization underlying BiCGStab(ℓ).
Recently, IDR\((s)\) was generalized by combining ideas from IDR\((s)\) with the higher dimensional minimization underlying BiCGStab\((\ell)\).

The first paper was a Japanese two-sided sketch of a method named GIDR\((s, L)\) by Masaaki Tanio and Masaaki Sugihara, followed independently by a joint paper by Gerard Sleijpen and Martin van Gijzen.
Recently, IDR(s) was generalized by combining ideas from IDR(s) with the higher dimensional minimization underlying BiCGStab(ℓ).

The first paper was a japanese two-sided sketch of a method named GIDR(s, L) by Masaaki Tanio and Masaaki Sugihara, followed independently by a joint paper by Gerard Sleijpen and Martin van Gijzen.

IDRStab is based on the computation of a Hessenberg matrix of basis matrices and a linear combination of the last column with polynomial coefficients to circumvent the need for the roots $\omega_j$. 
Recently, IDR\((s)\) was generalized by combining ideas from IDR\((s)\) with the higher dimensional minimization underlying BiCGStab\((\ell)\).

The first paper was a Japanese two-sided sketch of a method named GIDR\((s,L)\) by Masaaki Tanio and Masaaki Sugihara, followed independently by a joint paper by Gerard Sleijpen and Martin van Gijzen.

IDRStab is based on the computation of a Hessenberg matrix of basis matrices and a linear combination of the last column with polynomial coefficients to circumvent the need for the roots \(\omega_j\).

IDRStab and the eigenvalue approximations of the resulting Sonneveld pencils are currently analyzed („Studienarbeit“ of Anisa Rizvanolli).
MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

\[
x_k := Q_k z_k, \quad z_k := H_k^\dagger e_1 \|r_0\|.
\]
MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

\[ x_k := Q_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \]

In IDR based methods we have to extend the MR framework to generalized Hessenberg decompositions:

\[ x_k := Q_k U_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \]
MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

$$x_k := Q_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \quad \quad \quad (1)$$

In IDR based methods we have to extend the MR framework to generalized Hessenberg decompositions:

$$x_k := Q_k U_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \quad \quad \quad (2)$$

The implementation has many parameters that we should select “optimal”. Extensive numerical tests are currently done by Olaf Rendel. As an example we show the convergence curves (the true residuals) for the matrix `add20` from Matrix Market.
MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

\[ x_k := Q_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \]

In IDR based methods we have to extend the MR framework to generalized Hessenberg decompositions:

\[ x_k := Q_k U_k z_k, \quad z_k := H_k^\dagger e_1 \| r_0 \|. \]

The implementation has many parameters that we should select “optimal”. Extensive numerical tests are currently done by Olaf Rendel. As an example we show the convergence curves (the true residuals) for the matrix add20 from Matrix Market.

Ongoing joint work with Olaf Rendel, Gerard Sleijpen, and Martin van Gijzen.
$s = 8$; $\omega_j$ local minimization; next by maximal last; various orthogonalizations
\( s = 8; \ \omega_j \) local minimization; various expansions; MGS orthogonalization
$s = 8; \omega_j$ various strategies; GS expansion; stable basis vectors
Residuals of add20

various $s$; $\omega_j$ inverse Rayleigh; stable expansion; GS expansion
QMRIDR: add20

Residuals of add20

various $s$; $\omega_j$ local minimization; stable expansion; MGS expansion
Residuals of add20

\[ ||r||/||b|| \]

Various \( s \); \( \omega_j \) local minimization; stable expansion; GS expansion
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

We related convergence to Ritz values.
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

We related convergence to Ritz values.

We sketched IDR and IDR(s).

We hopefully convinced you that IDR is an interesting Krylov subspace method and offers lots of even more interesting problems in the design and analysis of new IDR based methods.

What about inexact IDR/IDREig/IDRStab/QMRIDR?
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

We related convergence to Ritz values.

We sketched IDR and IDR(s).

We introduced the framework of generalized Hessenberg decompositions.
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

We related convergence to Ritz values.

We sketched IDR and IDR\(^{(s)}\).

We introduced the framework of generalized Hessenberg decompositions.

We briefly touched generalizations of IDR\(^{(s)}\), namely, IDREig, IDRStab, and QMRIDR.
We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

We related convergence to Ritz values.

We sketched IDR and IDR(s).

We introduced the framework of generalized Hessenberg decompositions.

We briefly touched generalizations of IDR(s), namely, IDREig, IDRStab, and QMRIDR.

We hopefully convinced you that IDR is an interesting Krylov subspace method and offers lots of even more interesting problems in the design and analysis of new IDR based methods.
Conclusion and Overview

- We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.
- We related convergence to Ritz values.
- We sketched IDR and IDR\textsuperscript{s}.
- We introduced the framework of generalized Hessenberg decompositions.
- We briefly touched generalizations of IDR\textsuperscript{s}, namely, IDREig, IDRStab, and QMRIDR.
- We hopefully convinced you that IDR is an interesting Krylov subspace method and offers lots of even more interesting problems in the design and analysis of new IDR based methods.
- What about inexact IDR/IDREig/IDRStab/QMRIDR?
Thank you for your attention!
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).

AGS-IDR-CGS-BiCGSTAB-IDR(s): The circle closed. A case of serendipity.

Numerical experiments with a multiple grid and a preconditioned Lanczos type method.