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

UNIVERSITY OF BATH

Outline

Basics

Outline

Basics

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

IDR(s)

- IDR
- IDR(s)
- IDREig
- IDR(s)Stab(ℓ)
- QMRIDR

Outline

Basics

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

IDR(s)

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

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

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

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.

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,

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,

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.

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

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,

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,
- based on tools from a variety of areas that do not seem to be related to Krylov subspace methods at all,

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,
- 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.

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.

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:

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),

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,

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,

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 . . .

What are you going to talk about?

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,

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,

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.

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.

Background

Large linear systems are solved by projection onto smaller subspaces,

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

Background

Large linear systems are solved by projection onto smaller subspaces,

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

Galërkin method:

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

Background

Large linear systems are solved by projection onto smaller subspaces,

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

Galërkin method:

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

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

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

Background

Large linear systems are solved by projection onto smaller subspaces,

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

Galërkin method:

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

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

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

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,

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

Galërkin method:

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

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

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

Compute residual: $\mathbf{r}_1 := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_1 = \mathbf{Q}_1\mathbf{e}_1\|\mathbf{r}_0\| - \mathbf{A}\mathbf{Q}_1\mathbf{z}_1$. Both steps involve $\mathbf{A}\mathbf{q}_1$.
Expand space:

$$\mathcal{K}_2 := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0\} = \text{span}\{\mathbf{q}_1, \mathbf{q}_2\}.$$

Krylov subspaces

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

$$\mathcal{K}_k := \text{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\} = \text{span}\,\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}.$$

Krylov subspaces

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

$$\mathcal{K}_k := \text{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\} = \text{span}\,\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}.$$

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

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

Krylov subspaces

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

$$\mathcal{K}_k := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\} = \text{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}.$$

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

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

Residual polynomials are polynomials that

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

Krylov subspaces

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

$$\mathcal{K}_k := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\} = \text{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}.$$

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

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

Residual polynomials are polynomials that

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

Residual polynomials arise because

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

Krylov subspace methods

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.

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.

$$
\begin{array}{l}
\mathbf{r} = \mathbf{r}_0,\ \mathbf{q} = \mathbf{r}/\|\mathbf{r}\| \\
\mathbf{Q} = \mathbf{q},\ \underline{\mathbf{H}} = () \\
\text{for } k = 1, \dots \\
\qquad \mathbf{r} = \mathbf{A}\mathbf{q} \\
\qquad \mathbf{c} = \mathbf{Q}^{\mathsf{H}}\mathbf{r} \\
\qquad \mathbf{r} = \mathbf{r} - \mathbf{Q}\mathbf{c} \\
\qquad \underline{\mathbf{H}} = \left(\underline{\mathbf{H}}, \mathbf{c}; \mathbf{o}^{\mathsf{T}}, \|\mathbf{r}\|\right) \\
\qquad \mathbf{q} = \mathbf{r}/\|\mathbf{r}\| \\
\qquad \mathbf{Q} = \left(\mathbf{Q}, \mathbf{q}\right) \\
\text{end}
\end{array}
$$

Hessenberg decompositions

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

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

where

- $\mathbf{Q}_{k+1} = \left(\mathbf{Q}_k, \mathbf{q}_{k+1}\right) \in \mathbb{C}^{n\times(k+1)}$ collects the basis vectors,
- $\underline{\mathbf{H}}_k \in \mathbb{C}^{(k+1)\times k}$ is an unreduced extended Hessenberg matrix.

Hessenberg decompositions

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

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

where

- $\mathbf{Q}_{k+1} = \left(\mathbf{Q}_k, \mathbf{q}_{k+1}\right) \in \mathbb{C}^{n\times(k+1)}$ collects the basis vectors,
- $\underline{\mathbf{H}}_k \in \mathbb{C}^{(k+1)\times k}$ is an unreduced extended Hessenberg matrix.

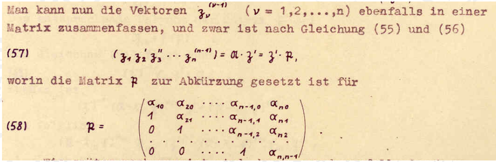
Aspects of perturbed Krylov subspace methods can be captured with perturbed Hessenberg decompositions

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

where $\mathbf{F}_k \in \mathbb{C}^{n\times k}$ accounts for the perturbations.

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 $\mathfrak{z}_\nu^{(\nu-1)}$ $(\nu = 1,2,\ldots,n)$ ebenfalls in einer Matrix zusammenfassen, und zwar ist nach Gleichung (55) und (56)

$$(\mathfrak{z}_1\, \mathfrak{z}_2'\, \mathfrak{z}_3'' \cdots \mathfrak{z}_n^{(n-1)}) = \mathfrak{A}\cdot\mathfrak{Z}' = \mathfrak{Z}'\cdot\mathfrak{P}, \tag{57}$$

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

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

- 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 $\mathbf{x}_k := \mathbf{Q}_k\mathbf{z}_k$ and the MR approximation $\underline{\mathbf{x}}_k := \mathbf{Q}_k\underline{\mathbf{z}}_k$ with coefficient vectors

$$\mathbf{z}_k := \mathbf{H}_k^{-1}\mathbf{e}_1\|\mathbf{r}_0\| \quad \text{and} \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$$

satisfy

$$\mathbf{r}_k := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_k = \mathcal{R}_k(\mathbf{A})\mathbf{r}_0 \quad \text{and} \quad \underline{\mathbf{r}}_k := \mathbf{r}_0 - \mathbf{A}\underline{\mathbf{x}}_k = \underline{\mathcal{R}}_k(\mathbf{A})\mathbf{r}_0$$

with residual polynomials $\mathcal{R}_k$ and $\underline{\mathcal{R}}_k$ given by

$$\mathcal{R}_k(z) := \det\left(\mathbf{I}_k - z\mathbf{H}_k^{-1}\right) \quad \text{and} \quad \underline{\mathcal{R}}_k(z) := \det\left(\mathbf{I}_k - z\underline{\mathbf{H}}_k^{\dagger}\underline{\mathbf{I}}_k\right).$$

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 $\mathbf{x}_k := \mathbf{Q}_k\mathbf{z}_k$ and the MR approximation $\underline{\mathbf{x}}_k := \mathbf{Q}_k\underline{\mathbf{z}}_k$ with coefficient vectors

$$\mathbf{z}_k := \mathbf{H}_k^{-1}\mathbf{e}_1\|\mathbf{r}_0\| \quad \text{and} \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger}\underline{\mathbf{e}}_1\|\mathbf{r}_0\|$$

satisfy

$$\mathbf{r}_k := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_k = \mathcal{R}_k(\mathbf{A})\mathbf{r}_0 \quad \text{and} \quad \underline{\mathbf{r}}_k := \mathbf{r}_0 - \mathbf{A}\underline{\mathbf{x}}_k = \underline{\mathcal{R}}_k(\mathbf{A})\mathbf{r}_0$$

with residual polynomials $\mathcal{R}_k$ and $\underline{\mathcal{R}}_k$ given by

$$\mathcal{R}_k(z) := \det\left(\mathbf{I}_k - z\mathbf{H}_k^{-1}\right) \quad \text{and} \quad \underline{\mathcal{R}}_k(z) := \det\left(\mathbf{I}_k - z\underline{\mathbf{H}}_k^{\dagger}\underline{\mathbf{I}}_k\right).$$

The convergence of OR and MR depends on the Ritz and harmonic Ritz values, respectively.

Perturbed OR methods

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

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

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

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

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

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

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

where

$$\mathcal{R}_{\ell+1:k}(z) := \det\left(\mathbf{I}_{k-\ell} - z\mathbf{H}_{\ell+1:k}^{-1}\right).$$

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

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

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

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

where

$$\mathcal{R}_{\ell+1:k}(z) := \det\left(\mathbf{I}_{k-\ell} - z\mathbf{H}_{\ell+1:k}^{-1}\right).$$

We can expect convergence when $\mathbf{F}_k\mathbf{z}_k$ remains bounded (inexact methods) and all $\mathcal{R}_{\ell+1:k}(\mathbf{A})$ are “small”.

Outline

Birth of a method

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

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

Birth of a method

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

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

This recurrence (almost) always results in the zero vector after $2n$ steps, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{r}_0 \in \mathbb{C}^n$, $\mathbf{r}_1 = \mathbf{A}\mathbf{r}_0$, and $\mathbf{p} \in \mathbb{C}^n$ are arbitrarily chosen.

Birth of a method

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

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

This recurrence (almost) always results in the zero vector after $2n$ steps, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{r}_0 \in \mathbb{C}^n$, $\mathbf{r}_1 = \mathbf{A}\mathbf{r}_0$, and $\mathbf{p} \in \mathbb{C}^n$ are arbitrarily chosen.

He realized that the recurrence constructs vectors in spaces $\mathcal{G}_j$ of shrinking dimensions:

$$\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0) = \text{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots\}$$
$$\mathcal{G}_j := (\mathbf{I}-\mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad \mathcal{S} = \text{span}\,\{\mathbf{p}\}^{\perp}, \quad j = 1, \ldots$$

Birth of a method

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

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

This recurrence (almost) always results in the zero vector after $2n$ steps, where $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{r}_0 \in \mathbb{C}^n$, $\mathbf{r}_1 = \mathbf{A}\mathbf{r}_0$, and $\mathbf{p} \in \mathbb{C}^n$ are arbitrarily chosen.

He realized that the recurrence constructs vectors in spaces $\mathcal{G}_j$ of shrinking dimensions:

$$\mathcal{G}_0 := \mathcal{K}(\mathbf{A}, \mathbf{r}_0) = \text{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots\}$$
$$\mathcal{G}_j := (\mathbf{I} - \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad \mathcal{S} = \text{span}\,\{\mathbf{p}\}^{\perp}, \quad j = 1, \ldots$$

More precisely,

$$\mathbf{r}_{2j}, \mathbf{r}_{2j+1} \in \mathcal{G}_j, \quad j = 0, 1, \ldots$$

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 $k = 1, 2, \dots$ 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

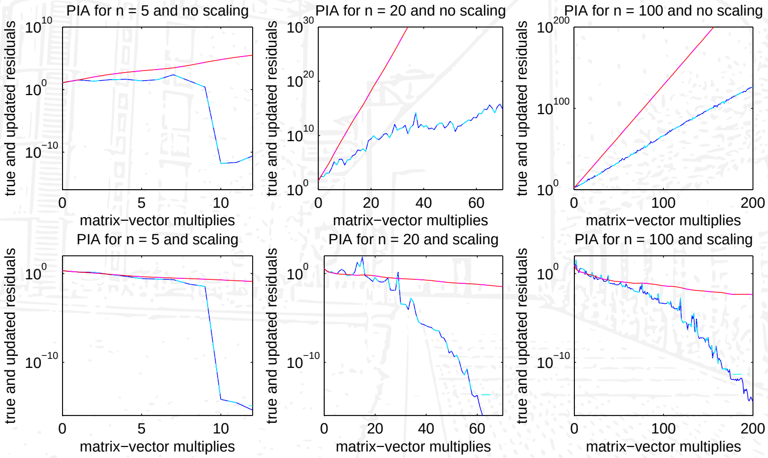
$$\gamma = \mathbf{p}^\mathsf{T}\mathbf{r}_\text{new} / \mathbf{p}^\mathsf{T}(\mathbf{r}_\text{old} - \mathbf{r}_\text{new})$$
$$\mathbf{s} = \mathbf{r}_\text{new} + \gamma(\mathbf{r}_\text{new} - \mathbf{r}_\text{old})$$
$$\mathbf{x}_\text{tmp} = \mathbf{x}_\text{new} + \gamma(\mathbf{x}_\text{new}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

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.

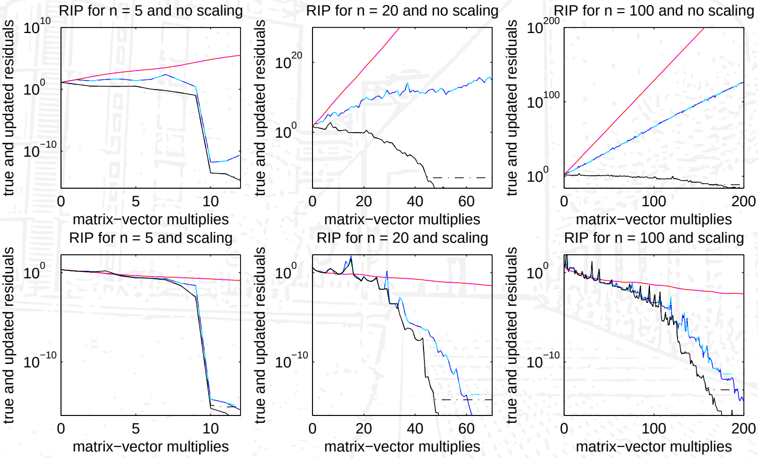
This part of the story took place “in the background” in the year 1976.

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.



IDR: 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 explicitely constructs the coefficients of the underlying Lanczos recurrence.

IDR: 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 explicitely constructs the coefficients of the underlying Lanczos recurrence.

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

IDR: 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 explicitely 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(s)

IDR can be generalized: instead of using one hyperplane $(\operatorname{span}\{\mathbf{p}\})^{\perp}$, one uses the intersection of s hyperplanes. This makes the dimension reduction step less frequent but the reduction a larger one.

IDR(s)

IDR can be generalized: instead of using one hyperplane $(\operatorname{span}\{\mathbf{p}\})^\perp$, 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(s)

IDR can be generalized: instead of using one hyperplane $(\text{span}\,\{\mathbf{p}\})^\perp$, 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.

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

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

  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 $\rho_k(0) = 1$.

Generalized Hessenberg decomposition

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

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

Generalized Hessenberg decomposition

IDR(s) can be captured using 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.

Generalized Hessenberg decomposition

IDR(s) can be captured using 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.

OR based IDR methods use

$$\mathbf{x}_k := \mathbf{Q}_k\mathbf{U}_k\mathbf{z}_k, \quad \mathbf{z}_k := \mathbf{H}_k^{-1}\mathbf{e}_1\|\mathbf{r}_0\|,$$

Generalized Hessenberg decomposition

IDR(s) can be captured using 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.

OR based IDR methods use

$$\mathbf{x}_k := \mathbf{Q}_k\mathbf{U}_k\mathbf{z}_k, \quad \mathbf{z}_k := \mathbf{H}_k^{-1}\mathbf{e}_1\|\mathbf{r}_0\|,$$

the residual is described by

$$\begin{aligned}
\mathbf{r}_k := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_k &= \mathbf{r}_0 - \mathbf{A}\mathbf{Q}_k\mathbf{U}_k\mathbf{z}_k = \mathbf{r}_0 - \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k\mathbf{z}_k \\
&= \mathbf{Q}_k(\mathbf{e}_1\|\mathbf{r}_0\| - \mathbf{H}_k\mathbf{z}_k) - \mathbf{q}_{k+1}h_{k+1,k}\mathbf{e}_k^{\mathsf{T}}\mathbf{z}_k \\
&= \mathcal{R}_k(\mathbf{A})\mathbf{r}_0, \quad \mathcal{R}_k(z) := \det\left(\mathbf{I}_k - z\mathbf{U}_k\mathbf{H}_k^{-1}\right).
\end{aligned}$$

Generalized Hessenberg decomposition

IDR(s) can be captured using 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.

OR based IDR methods use

$$\mathbf{x}_k := \mathbf{Q}_k\mathbf{U}_k\mathbf{z}_k, \quad \mathbf{z}_k := \mathbf{H}_k^{-1}\mathbf{e}_1\|\mathbf{r}_0\|,$$

the residual is described by

$$\begin{aligned}\mathbf{r}_k := \mathbf{r}_0 - \mathbf{A}\mathbf{x}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{Q}_k\mathbf{U}_k\mathbf{z}_k &= \mathbf{r}_0 - \mathbf{Q}_{k+1}\underline{\mathbf{H}}_k\mathbf{z}_k \\ &= \mathbf{Q}_k(\mathbf{e}_1\|\mathbf{r}_0\| - \mathbf{H}_k\mathbf{z}_k) - \mathbf{q}_{k+1}h_{k+1,k}\mathbf{e}_k^{\mathsf{T}}\mathbf{z}_k \\ &= \mathcal{R}_k(\mathbf{A})\mathbf{r}_0, \quad \mathcal{R}_k(z) := \det\left(\mathbf{I}_k - z\mathbf{U}_k\mathbf{H}_k^{-1}\right).\end{aligned}$$

Tacitly assuming $\|\mathbf{q}_{k+1}\| = 1$, we have $\|\mathbf{r}_k\| = |h_{k+1,k}z_k|$.

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

$$
\left(\begin{array}{cccccccccccc}
\times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
+ & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & + & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & + & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & + & \times & \times & \times & \times & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & + & \times & \times & \times & \times & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & + & \times & \times & \times & \times & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & + & \times & \times & \times & \times & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & + & \times & \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 \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & + & \times
\end{array}\right),
\left(\begin{array}{cccccccccccc}
\times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \times & \times & \times & \times & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \times & \times & \times & \times & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \times & \times & \times & \times & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \times & \times & \times & \times & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \times & \times & \times & \times & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \times & \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 \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \times
\end{array}\right).
$$

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

$$
\left(\begin{array}{cccccccccccc}
\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
+&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&+&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&+&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&+&\times&\times&\times&\times&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&+&\times&\times&\times&\times&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&+&\times&\times&\times&\times&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&+&\times&\times&\times&\times&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&+&\times&\times&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&+&\times&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&+&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&+&\times
\end{array}\right),
\left(\begin{array}{cccccccccccc}
\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\times&\times&\times&\times&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\times&\times&\times&\times&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\times&\times&\times&\times&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&\times&\times&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&\times&\times&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times
\end{array}\right).
$$

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

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

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

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

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

Understanding IDR: 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

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

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

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

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

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

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

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

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

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

IDREig

The eigenvalues of the pencil $(\mathbf{H}_k, \mathbf{U}_k)$ are the roots of the residual polynomials and some of these converge to eigenvalues of $\mathbf{A}$.

IDREig

The eigenvalues of the pencil $(\mathbf{H}_k, \mathbf{U}_k)$ are the roots of the residual polynomials and some of these converge to eigenvalues of $\mathbf{A}$.

Suppose that $\mathbf{Q}_{k+1}$ has full rank. The pencil $(\mathbf{H}_k, \mathbf{U}_k)$ arises as a oblique projection of $(\mathbf{A}, \mathbf{I}_n)$, as

$$\begin{aligned} \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{A}, \mathbf{I}_n)\mathbf{Q}_k\mathbf{U}_k &= \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{A}\mathbf{Q}_k\mathbf{U}_k, \mathbf{Q}_k\mathbf{U}_k) \\ &= \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{Q}_{k+1}\underline{\mathbf{H}}_k, \mathbf{Q}_k\mathbf{U}_k) = (\underline{\mathbf{I}}_k^{\mathsf{T}}\underline{\mathbf{H}}_k, \mathbf{U}_k) = (\mathbf{H}_k, \mathbf{U}_k), \end{aligned} \tag{1}$$

where $\widehat{\mathbf{Q}}_k^{\mathsf{H}} := \underline{\mathbf{I}}_k^{\mathsf{T}}\mathbf{Q}_{k+1}^{\dagger}$.

IDREig

The eigenvalues of the pencil $(\mathbf{H}_k, \mathbf{U}_k)$ are the roots of the residual polynomials and some of these converge to eigenvalues of $\mathbf{A}$.

Suppose that $\mathbf{Q}_{k+1}$ has full rank. The pencil $(\mathbf{H}_k, \mathbf{U}_k)$ arises as a oblique projection of $(\mathbf{A}, \mathbf{I}_n)$, as

$$\begin{aligned} \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{A}, \mathbf{I}_n)\mathbf{Q}_k\mathbf{U}_k &= \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{A}\mathbf{Q}_k\mathbf{U}_k, \mathbf{Q}_k\mathbf{U}_k) \\ &= \widehat{\mathbf{Q}}_k^{\mathsf{H}}(\mathbf{Q}_{k+1}\underline{\mathbf{H}}_k, \mathbf{Q}_k\mathbf{U}_k) = (\underline{\mathbf{I}}_k^{\mathsf{T}}\underline{\mathbf{H}}_k, \mathbf{U}_k) = (\mathbf{H}_k, \mathbf{U}_k), \end{aligned} \tag{1}$$

where $\widehat{\mathbf{Q}}_k^{\mathsf{H}} := \underline{\mathbf{I}}_k^{\mathsf{T}}\mathbf{Q}_{k+1}^{\dagger}$.

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, ω_j, $\mathbf{P}$) to obtain good eigenpair approximations (this is ongoing joint work with Olaf Rendel and Anisa Rizvanolli).

IDRStab

Recently, IDR(s) was generalized by combining ideas from IDR(s) with the higher dimensional minimization underlying BiCGStab(ℓ).

IDRStab

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

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 ω_j.

IDRStab

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 ω_j.

IDRStab and the eigenvalue approximations of the resulting Sonneveld pencils are currently analyzed („Studienarbeit“ of Anisa Rizvanolli).

QMRIDR

MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

$$\mathbf{x}_k := \mathbf{Q}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

QMRIDR

MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

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

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \mathbf{U}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

QMRIDR

MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

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

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \mathbf{U}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

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.

QMRIDR

MR methods use the extended Hessenberg matrix to compute the coefficients of the vector in the Krylov subspace, i.e.,

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

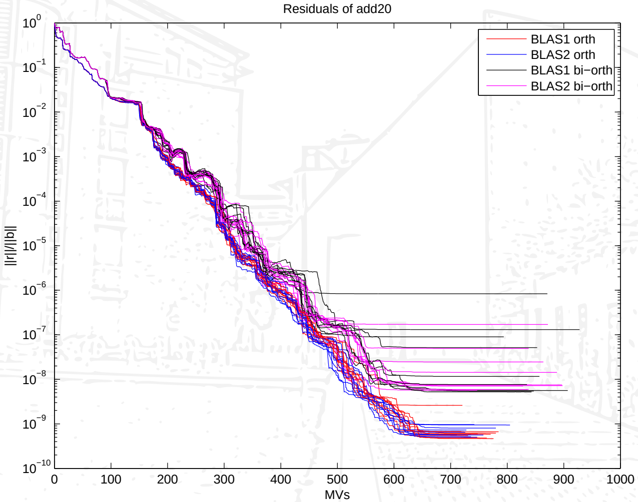
In IDR based methods we have to extend the MR framework to generalized Hessenberg decompositions:

$$\underline{\mathbf{x}}_k := \mathbf{Q}_k \mathbf{U}_k \underline{\mathbf{z}}_k, \quad \underline{\mathbf{z}}_k := \underline{\mathbf{H}}_k^{\dagger} \underline{\mathbf{e}}_1 \|\mathbf{r}_0\|.$$

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.

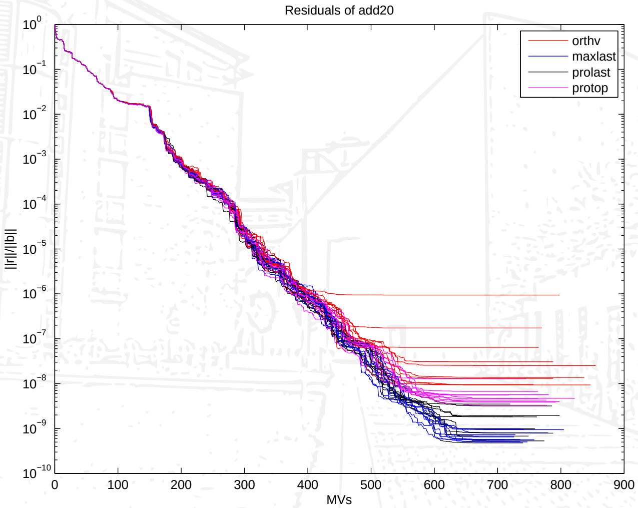
Ongoing joint work with Olaf Rendel, Gerard Sleijpen, and Martin van Gijzen.

QMRIDR: add20



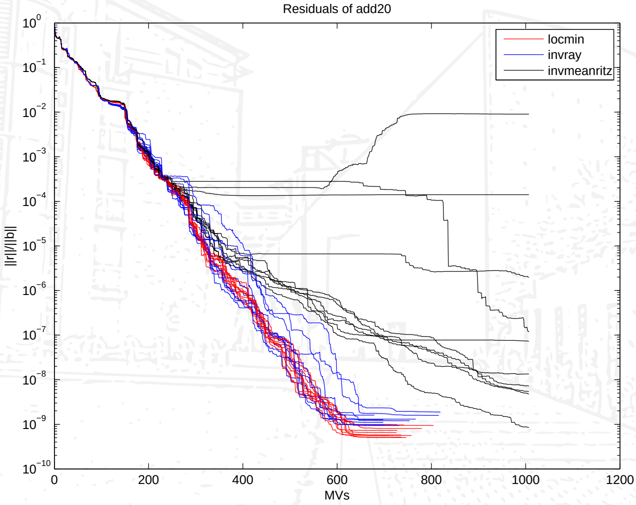
$s = 8$; ω_j local minimization; next by maximal last; various orthogonalizations

QMRIDR: add20



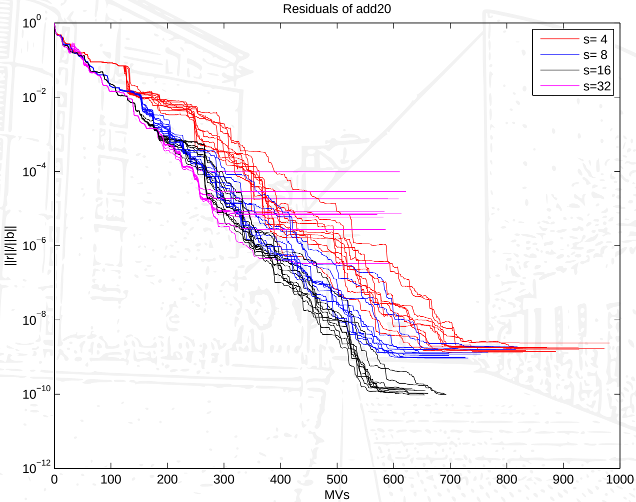
$s = 8$; ω_j local minimization; various expansions; MGS orthogonalization

QMRIDR: add20



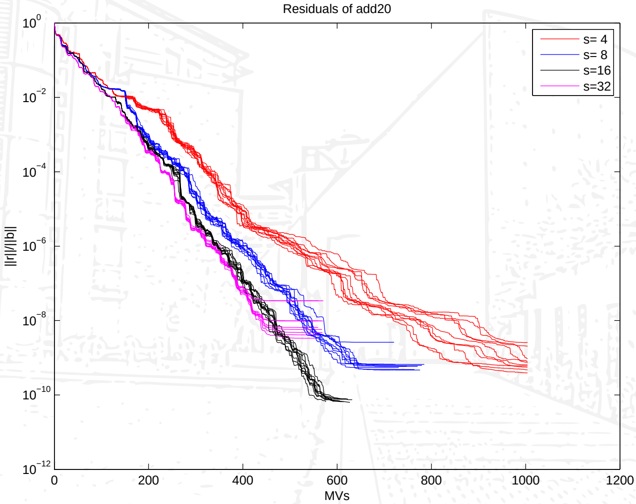
$s = 8$; ω_j various strategies; GS expansion; stable basis vectors

QMRIDR: add20



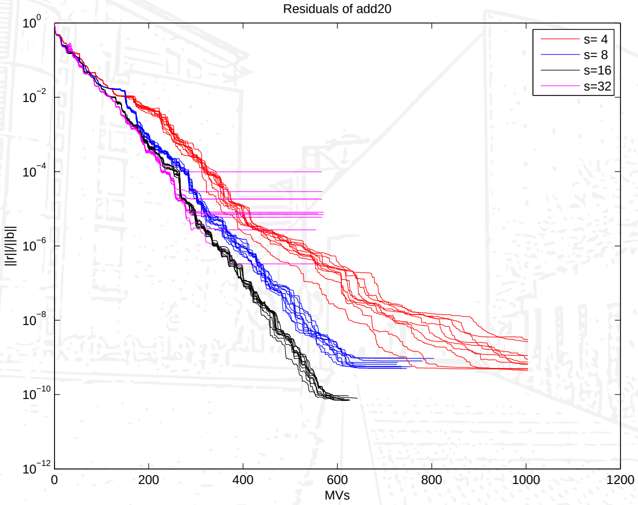
various s; ω_j inverse Rayleigh; stable expansion; GS expansion

QMRIDR: add20



various s; ω_j local minimization; stable expansion; MGS expansion

QMRIDR: add20



various s; ω_j local minimization; stable expansion; GS expansion

Conclusion and Outview

- We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.

Conclusion and Outview

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

Conclusion and Outview

- We sketched some basic facts about Krylov subspace methods and Hessenberg decompositions.
- We related convergence to Ritz values.
- We sketched IDR and IDR(s).

Conclusion and Outview

- 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.

Conclusion and Outview

- 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.

Conclusion and Outview

- 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 conviced 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 Outview

- 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 conviced 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!

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.