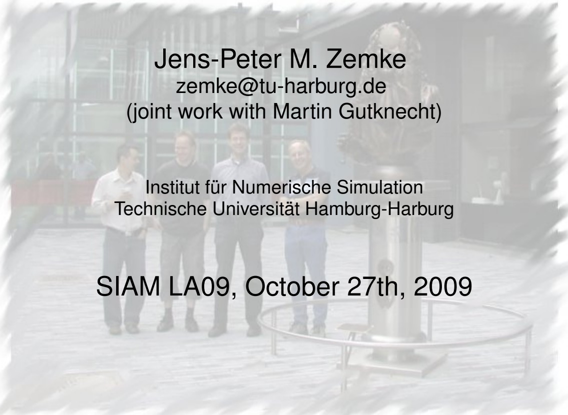
Eigenvalue Perspectives of the IDR Family



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

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

SIAM LA09, October 27th, 2009

Outline

Krylov subspace methods

Hessenberg decompositions

QOR/QMR/Ritz-Galërkin

OrthoRes-type methods

LTPM

IDR

IDR(s)ORes

Sonneveld pencil and Sonneveld matrix

Purified pencil

Deflated pencil and deflated matrix

BiORes($s, 1$)

Numerical Examples

Conclusion

Hessenberg decompositions

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

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

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

Hessenberg decompositions

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

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

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

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

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

Hessenberg decompositions

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

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

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

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

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

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

Generalized Hessenberg decompositions

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

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

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

Generalized Hessenberg decompositions

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

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

and perturbed generalized Hessenberg decompositions

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

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

Generalized Hessenberg decompositions

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

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

and perturbed generalized Hessenberg decompositions

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

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

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions.

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

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

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

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

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

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

These approaches extend easily to generalized Hessenberg decompositions.

QOR/QMR/Ritz-Galërkin

There are three well-known approaches based on such Hessenberg decompositions, namely

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

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

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

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

These approaches extend easily to generalized Hessenberg decompositions.

IDR is of type QOR.

OrthoRes-type methods

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

OrthoRes-type methods

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

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

OrthoRes-type methods

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

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

OrthoRes-type methods have a Hessenberg decomposition

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

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

OrthoRes-type methods

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

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

OrthoRes-type methods have a Hessenberg decomposition

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

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

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

is diagonally scaled to be the matrix of residual vectors.

OrthoRes-type methods

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

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

OrthoRes-type methods have a generalized Hessenberg decomposition

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

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

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

is diagonally scaled to be the matrix of residual vectors.

OrthoRes-type methods

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

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

OrthoRes-type methods have a generalized Hessenberg decomposition

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

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

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

is diagonally scaled to be the matrix of residual vectors.

IDR is of type OrthoRes.

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos $\approx$ CG $\approx$ MinRes are based on short-term recurrences, whereas Arnoldi $\approx$ GMRes are based on long-term recurrences.

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos $\approx$ CG $\approx$ MinRes are based on short-term recurrences, whereas Arnoldi $\approx$ GMRes are based on long-term recurrences.

A large class of short-term recurrences is obtained by multiplication of (simple, block, any number of left- and right-hand sides) Lanczos polynomials with another polynomial.

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos $\approx$ CG $\approx$ MinRes are based on short-term recurrences, whereas Arnoldi $\approx$ GMRes are based on long-term recurrences.

A large class of short-term recurrences is obtained by multiplication of (simple, block, any number of left- and right-hand sides) Lanczos polynomials with another polynomial. At the same time the need for the transpose is eliminated.

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos $\approx$ CG $\approx$ MinRes are based on short-term recurrences, whereas Arnoldi $\approx$ GMRes are based on long-term recurrences.

A large class of short-term recurrences is obtained by multiplication of (simple, block, any number of left- and right-hand sides) Lanczos polynomials with another polynomial. At the same time the need for the transpose is eliminated.

Methods of this class are, e.g., (original) IDR, CGS, BiCGStab, BiCGStab2, BiCGStab(ℓ), ML(k)BiCGStab, and IDR(s).

Lanczos-type Product Methods

Krylov subspace methods can roughly be divided into the classes of short-term and long-term recurrences.

Lanczos $\approx$ CG $\approx$ MinRes are based on short-term recurrences, whereas Arnoldi $\approx$ GMRes are based on long-term recurrences.

A large class of short-term recurrences is obtained by multiplication of (simple, block, any number of left- and right-hand sides) Lanczos polynomials with another polynomial. At the same time the need for the transpose is eliminated.

Methods of this class are, e.g., (original) IDR, CGS, BiCGStab, BiCGStab2, BiCGStab(ℓ), ML(k)BiCGStab, and IDR(s).

We show how IDR fits into the LTPM framework.

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

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

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

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

end while

A few remarks:

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

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

A few remarks:

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

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

A few remarks:

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

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

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

A few remarks:

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

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

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

A few remarks:

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

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

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

A few remarks:

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

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

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

A few remarks:

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

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

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

end while

A few remarks:

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

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

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

A few remarks:

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

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

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

A few remarks:

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

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

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

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

A few remarks:

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

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

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

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

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

A few remarks:

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

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

IDR(s)ORes is based on oblique projections.

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

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

A few remarks:

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

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

IDR(s)ORes is based on oblique projections.

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

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

$\Rightarrow \mathbf{r}_n = (\mathbf{I} - \omega_j\mathbf{A})\mathbf{v}_{n-1}$

A few remarks:

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

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

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

The underlying Hessenberg decomposition

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

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

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

The underlying Hessenberg decomposition

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

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

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

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

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

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

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

Sonneveld pencil and Sonneveld matrix

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

$$
\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 & \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{pmatrix}.
$$

Sonneveld pencil and Sonneveld matrix

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

$$
\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 & \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{pmatrix}.
$$

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.

Purification

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

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

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

$$
\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&{\color{blue}\circ}&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\times&\times&{\color{blue}\circ}&{\color{blue}\circ}&\circ&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\times&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&\circ&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&\circ&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\times&\times&\times&{\color{blue}\circ}&\circ&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\times&\times&{\color{blue}\circ}&{\color{blue}\circ}&\circ&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\times&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&\circ&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&{\color{blue}\circ}&\circ\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&\times&{\color{blue}\circ}\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&\times&{\color{blue}\circ}\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\times&{\color{blue}\circ}\\
\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&\circ&{\color{blue}\circ}
\end{array}\right).
$$

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

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}.
$$

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.

Deflation

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

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

Deflation

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

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

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

Deflation

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

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

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

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.

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

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

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.

A Lanczos process with multiple left-hand sides

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

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

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

This matrix (in the extended version) satisfies

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

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

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

A Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

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

A Lanczos process with multiple left-hand sides

The reduced residuals are defined by

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

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

Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathbf{P}$.

A Lanczos process with multiple left-hand sides

The reduced residuals are defined by

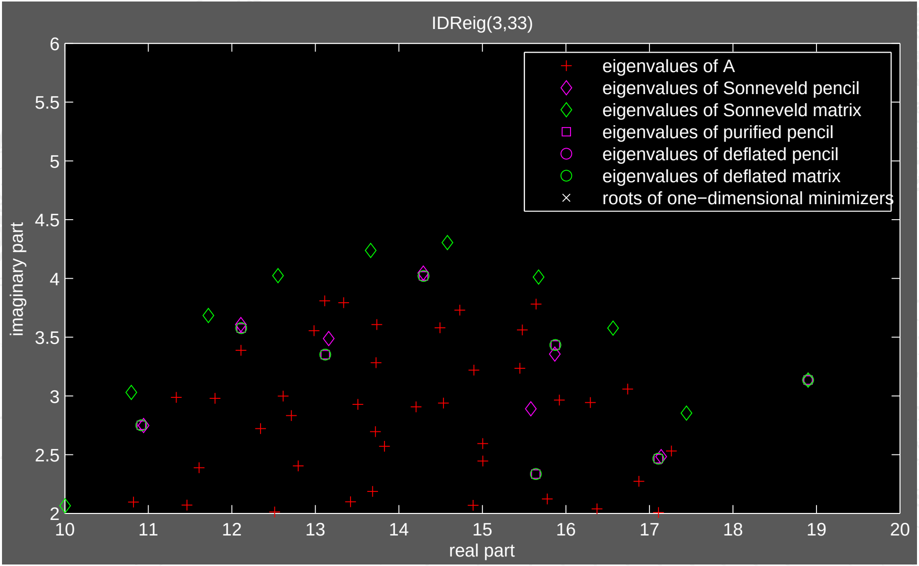
$$\Omega_j(\mathbf{A})\mathbf{q}_{js+k} = \mathbf{r}_{j(s+1)+k} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{j(s+1)+k-1}$$

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

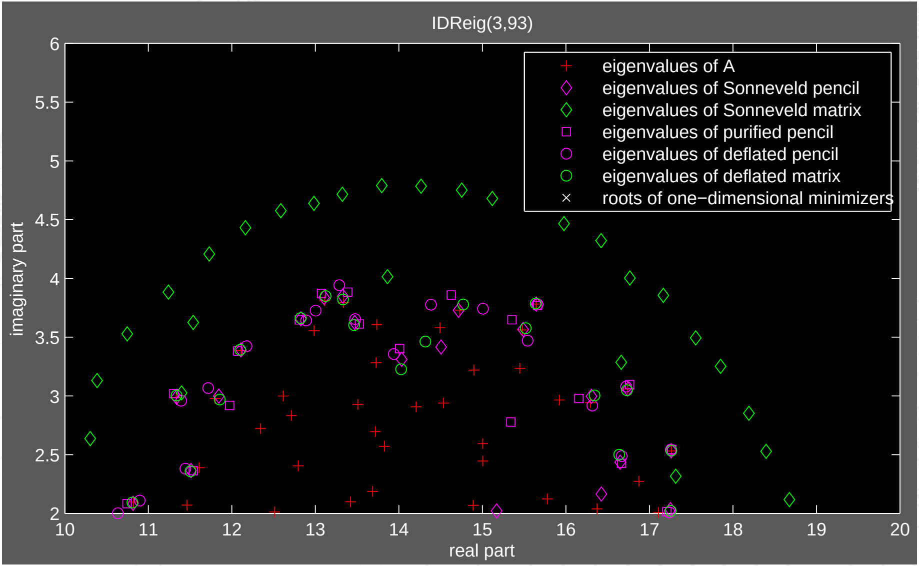
Thus, $\mathbf{q}_{js+k} \perp \Omega_{j-1}(\mathbf{A}^{\mathsf{H}})\mathbf{P}$.

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

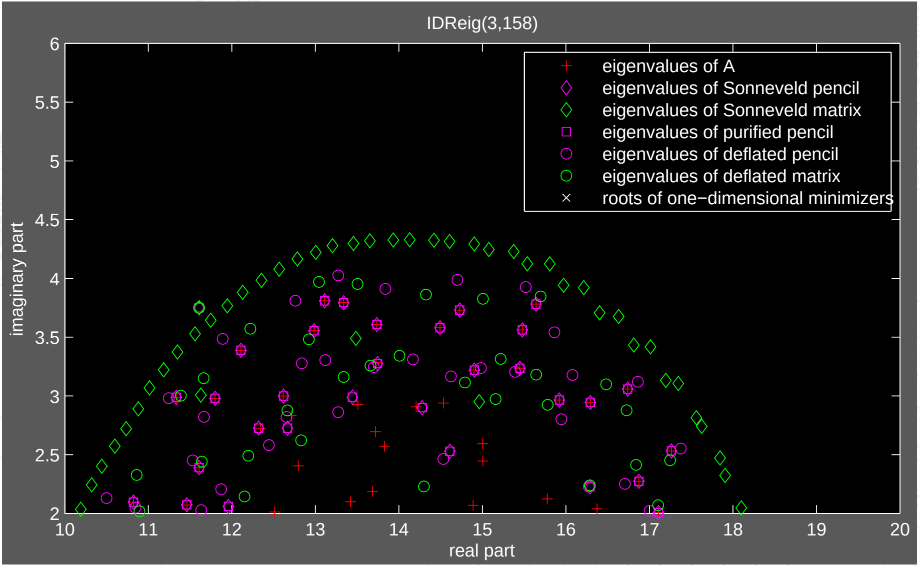
Selected examples for $s = 3$



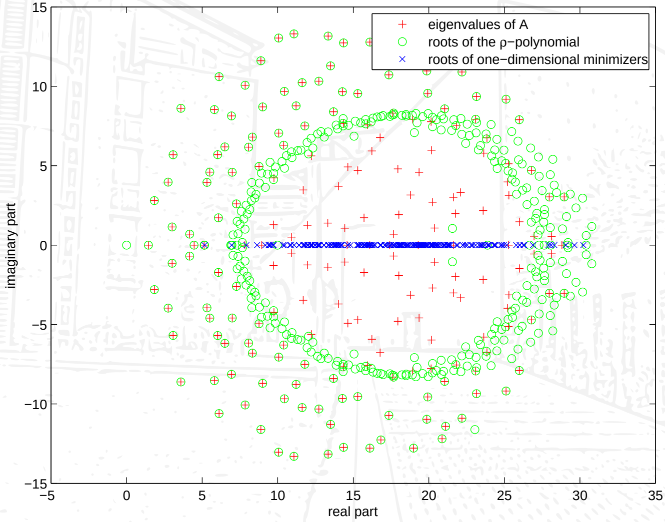
Selected examples for $s = 3$



Selected examples for $s = 3$



600 steps for $s = 2$



Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.
- We have not presented how to extract approximations to eigenvectors. This can be done at all stages.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.
- We have not presented how to extract approximations to eigenvectors. This can be done at all stages.
- The convergence of the Ritz pairs is related to the behavior in finite precision, thus via monitoring the convergence of Ritz pairs we can guess the finite precision behavior.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.
- We have not presented how to extract approximations to eigenvectors. This can be done at all stages.
- The convergence of the Ritz pairs is related to the behavior in finite precision, thus via monitoring the convergence of Ritz pairs we can guess the finite precision behavior.
- The construction of approximate eigenpairs “on the fly” should enable us to construct enhanced IDR(s) family members, e.g., IDR with recycling or deflation.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.
- We have not presented how to extract approximations to eigenvectors. This can be done at all stages.
- The convergence of the Ritz pairs is related to the behavior in finite precision, thus via monitoring the convergence of Ritz pairs we can guess the finite precision behavior.
- The construction of approximate eigenpairs "on the fly" should enable us to construct enhanced IDR(s) family members, e.g., IDR with recycling or deflation.
- The analysis of IDR(s)ORes presented carries over to other family members.

Conclusion & Outlook

- We have shown that IDR(s)ORes is a Lanczos-type product method with an underlying Lanczos process with s left-hand sides and one right-hand side.
- We have shown how to extract approximations to eigenvalues from IDR(s)ORes.
- We have not presented how to extract approximations to eigenvectors. This can be done at all stages.
- The convergence of the Ritz pairs is related to the behavior in finite precision, thus via monitoring the convergence of Ritz pairs we can guess the finite precision behavior.
- The construction of approximate eigenpairs “on the fly” should enable us to construct enhanced IDR(s) family members, e.g., IDR with recycling or deflation.
- The analysis of IDR(s)ORes presented carries over to other family members.
- The understanding gained in analysing IDR(s)ORes should enable us to develop new IDR(s) family members better suited to eigenvalue computations.

Conclusion & Outlook

Thank you for your attention.