# IDR($s$): Finite precision aspects

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

joint work with Martin Gutknecht
(work in progress)

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

03.06.2009

# Outline

## Krylov subspace methods

- origin
- classification
- implementation

## My vision on IDR

- IDR as QOR
- IDR for eigenvalues

## Numerical example

# Origin of Krylov methods

The Krylov matrix $\mathbf{K}_n = \left(\mathbf{q}, \mathbf{A}\mathbf{q}, \mathbf{A}^2\mathbf{q}, \dots, \mathbf{A}^{m-1}\mathbf{q}\right)$ satisfies

$$\left(\mathbf{q}, \mathbf{A}\mathbf{K}_n\right) = \mathbf{K}_{n+1}. \tag{1}$$

# Origin of Krylov methods

The Krylov matrix $\mathbf{K}_n = \left(\mathbf{q}, \mathbf{A}\mathbf{q}, \mathbf{A}^2\mathbf{q}, \dots, \mathbf{A}^{m-1}\mathbf{q}\right)$ satisfies

$$\left(\mathbf{q}, \mathbf{A}\mathbf{K}_n\right) = \mathbf{K}_{n+1}. \tag{1}$$

The (columns of the) $n$th Krylov matrix span(s) a basis of the $n$th Krylov space $\mathcal{K}_n$ iff $n$ is less or equal to the grade of $\mathbf{q}$.

# Origin of Krylov methods

The Krylov matrix $\mathbf{K}_n = \left(\mathbf{q}, \mathbf{A}\mathbf{q}, \mathbf{A}^2\mathbf{q}, \ldots, \mathbf{A}^{m-1}\mathbf{q}\right)$ satisfies

$$\left(\mathbf{q}, \mathbf{A}\mathbf{K}_n\right) = \mathbf{K}_{n+1}. \tag{1}$$

The (columns of the) $n$th Krylov matrix span(s) a basis of the $n$th Krylov space $\mathcal{K}_n$ iff $n$ is less or equal to the grade of $\mathbf{q}$.

Suppose we choose upper triangular basis transformations $\mathbf{K}_n = \mathbf{Q}_n\mathbf{R}_n$,

$$\left(\mathbf{q}, \mathbf{A}\mathbf{Q}_n\mathbf{R}_n\right) = \mathbf{Q}_{n+1}\mathbf{R}_{n+1} \quad \Rightarrow \quad \left(\mathbf{q}, \mathbf{A}\mathbf{Q}_n\right) = \mathbf{Q}_{n+1}\mathbf{R}_{n+1}\begin{pmatrix} 1 & \mathbf{o}^T \\ \mathbf{o} & \mathbf{R}_n \end{pmatrix}^{-1}. \tag{2}$$

# Origin of Krylov methods

Then $\underline{\mathbf{C}}_n$ defined by

$$\begin{pmatrix} \star & \underline{\mathbf{C}}_n \\ \mathbf{o} & \end{pmatrix} := \mathbf{R}_{n+1} \begin{pmatrix} 1 & \mathbf{o}^T \\ \mathbf{o} & \mathbf{R}_n \end{pmatrix}^{-1} \tag{3}$$

is unreduced extended Hessenberg.

# Origin of Krylov methods

Then $\underline{\mathbf{C}}_n$ defined by

$$\begin{pmatrix} \star & \underline{\mathbf{C}}_n \\ \mathbf{o} & \end{pmatrix} := \mathbf{R}_{n+1} \begin{pmatrix} 1 & \mathbf{o}^T \\ \mathbf{o} & \mathbf{R}_n \end{pmatrix}^{-1} \tag{3}$$

is unreduced extended Hessenberg.

We end up with a Hessenberg decomposition

$$\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\underline{\mathbf{C}}_n = \mathbf{Q}_n\mathbf{C}_n + \mathbf{q}_{n+1}c_{n+1,n}\mathbf{e}_n^T, \tag{4}$$

where $\mathbf{C}_n$ is unreduced Hessenberg and measures the “ratio” of the basis transformations.

# Classification of Krylov methods

These Hessenberg decompositions are computed directly (e.g., using the methods of Lanczos or Arnoldi), split (e.g., (Bi)CG-Omin, i.e., using an LDMT decomposition), or implicitly (so-called Lanczos-type product methods, LTPM; e.g., CGS, BiCGStab).

# Classification of Krylov methods

These Hessenberg decompositions are computed directly (e.g., using the methods of Lanczos or Arnoldi), split (e.g., (Bi)CG-Omin, i.e., using an LDMT decomposition), or implicitly (so-called Lanczos-type product methods, LTPM; e.g., CGS, BiCGStab).

There are (basically) three well-known approaches based on Hessenberg decompositions, namely

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

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

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

# Classification of Krylov methods

These Hessenberg decompositions are computed directly (e.g., using the methods of Lanczos or Arnoldi), split (e.g., (Bi)CG-Omin, i.e., using an LDMT decomposition), or implicitly (so-called Lanczos-type product methods, LTPM; e.g., CGS, BiCGStab).

There are (basically) three well-known approaches based on Hessenberg decompositions, namely

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

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

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

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

# Classification of Krylov methods

Krylov subspace methods compute elements from the polynomial Krylov space.

# Classification of Krylov methods

Krylov subspace methods compute elements from the polynomial Krylov space.

The three classes of methods can be described using polynomials and polynomial interpolation:

QOR: $\mathbf{r}_n = \mathcal{R}_n(\mathbf{A})\mathbf{r}_0$, where $\mathcal{R}_n(z) := \det\left(\mathbf{I}_n - z\mathbf{C}_n^{-1}\right)$,

$\mathbf{x}_n = \mathcal{L}_n[z^{-1}](\mathbf{A})\mathbf{r}_0$, where $\mathcal{L}_n[z^{-1}](z) := \frac{\chi_n(0)-\chi_n(z)}{\chi_n(0)} z^{-1}$, $z \neq 0$,

QMR: $\underline{\mathbf{r}}_n = \underline{\mathcal{R}}_n(\mathbf{A})\mathbf{r}_0$, where $\underline{\mathcal{R}}_n(z) := \det\left(\mathbf{I}_n - z\underline{\mathbf{C}}_n^{\dagger}\underline{\mathbf{I}}_n\right)$,

$\underline{\mathbf{x}}_n = \underline{\mathcal{L}}_n[z^{-1}](\mathbf{A})\mathbf{r}_0$, where $\underline{\mathcal{L}}_n[z^{-1}](z)$ interpolates the function $z^{-1}$ at the harmonic Ritz values,

Ritz-Galërkin: $\mathbf{A}\mathbf{V}_n - \mathbf{V}_n\mathbf{J}_n = \frac{\chi_n(\mathbf{A})}{c_{1:n-1}}\mathbf{q}_1\mathbf{e}_n^T\mathbf{S}_n$ (for a specially chosen $\mathbf{S}_n$),

$\mathbf{v}_j^{(m)} = \mathcal{A}_n(\theta, \mathbf{A})\mathbf{q}_1$, where $\mathcal{A}_n(\theta, z) := \frac{\chi_n(\theta)-\chi_n(z)}{\theta - z}$, $\theta \neq z$.

# Implementation of Krylov methods

These Hessenberg decompositions are (more or less explicitely) constructed using linear algebra techniques (e.g., orthogonal and oblique projectors).

# Implementation of Krylov methods

These Hessenberg decompositions are (more or less explicitely) constructed using linear algebra techniques (e.g., orthogonal and oblique projectors).

In finite precision the recurrence will only approximately be satisfied,

$$\mathbf{A}\mathbf{Q}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{C}}_n = \mathbf{Q}_n\mathbf{C}_n + \mathbf{q}_{n+1}c_{n+1,n}\mathbf{e}_n^T, \tag{5}$$

where the perturbation term $\mathbf{F}_n$ is in some sense “small” and/or structured.

# Implementation of Krylov methods

These Hessenberg decompositions are (more or less explicitely) constructed using linear algebra techniques (e.g., orthogonal and oblique projectors).

In finite precision the recurrence will only approximately be satisfied,

$$\mathbf{A}\mathbf{Q}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{C}}_n = \mathbf{Q}_n\mathbf{C}_n + \mathbf{q}_{n+1}c_{n+1,n}\mathbf{e}_n^T, \tag{5}$$

where the perturbation term $\mathbf{F}_n$ is in some sense “small” and/or structured.

All the given polynomial relations extend to the perturbed case. Additional error terms indicate the obtainable level of accuracy (Z, 2007).

# Implementation of Krylov methods

These Hessenberg decompositions are (more or less explicitely) constructed using linear algebra techniques (e.g., orthogonal and oblique projectors).

In finite precision the recurrence will only approximately be satisfied,

$$\mathbf{A}\mathbf{Q}_n + \mathbf{F}_n = \mathbf{Q}_{n+1}\underline{\mathbf{C}}_n = \mathbf{Q}_n\mathbf{C}_n + \mathbf{q}_{n+1}c_{n+1,n}\mathbf{e}_n^T, \tag{5}$$

where the perturbation term $\mathbf{F}_n$ is in some sense “small” and/or structured.

All the given polynomial relations extend to the perturbed case. Additional error terms indicate the obtainable level of accuracy (Z, 2007).

To analyze the convergence behavior of a perturbed QOR Krylov method one has to figure out the behavior of the Ritz values, i.e., the eigenvalues of the Hessenberg matrices $\mathbf{C}_n$.

# IDR as Krylov subspace method

IDR is a Krylov subspace method, as every step is based on a multiplication with the matrix $\mathbf{A}$ (the iterates lie in a Krylov subspace) and certain linear combinations of previous basis vectors are used to compute new vectors (which defines an upper triangular basis transformation).

# IDR as Krylov subspace method

IDR is a Krylov subspace method, as every step is based on a multiplication with the matrix $\mathbf{A}$ (the iterates lie in a Krylov subspace) and certain linear combinations of previous basis vectors are used to compute new vectors (which defines an upper triangular basis transformation).

It turns out that IDR is a QOR method.

# IDR as Krylov subspace method

IDR is a Krylov subspace method, as every step is based on a multiplication with the matrix $\mathbf{A}$ (the iterates lie in a Krylov subspace) and certain linear combinations of previous basis vectors are used to compute new vectors (which defines an upper triangular basis transformation).

It turns out that IDR is a QOR method.

As the residuals are used as "basis" vectors (they need not be a basis, but that doesn't matter), the Krylov subspace method IDR will be what I refer to as 'of type ORTHORES'.

# IDR as Krylov subspace method

IDR is a Krylov subspace method, as every step is based on a multiplication with the matrix $\mathbf{A}$ (the iterates lie in a Krylov subspace) and certain linear combinations of previous basis vectors are used to compute new vectors (which defines an upper triangular basis transformation).

It turns out that IDR is a QOR method.

As the residuals are used as “basis” vectors (they need not be a basis, but that doesn’t matter), the Krylov subspace method IDR will be what I refer to as ‘of type ORTHORES’.

In these methods the columns of the resulting extended Hessenberg matrix sum to zero.

# IDR as Krylov subspace method

The IDR recurrences of the prototype IDR($s$) algorithm can be summarized by

$$\begin{aligned} \mathbf{r}_n &:= (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_{n-1}\,, \\ \mathbf{v}_n &:= \mathbf{r}_n - \widetilde{\mathbf{R}}_n \Delta \mathbf{c}_n = \widetilde{\mathbf{R}}_n \mathbf{y}_n \\ &= (1 - \gamma_1^{(n)})\mathbf{r}_n + \textstyle\sum_{\ell=1}^{s-1} (\gamma_\ell^{(n)} - \gamma_{\ell+1}^{(n)})\mathbf{r}_{n-\ell} + \gamma_s^{(n)} \mathbf{r}_{n-s}\,. \end{aligned} \tag{6}$$

# IDR as Krylov subspace method

The IDR recurrences of the prototype IDR($s$) algorithm can be summarized by

$$\begin{aligned} \mathbf{r}_n &:= (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_{n-1}\,, \\ \mathbf{v}_n &:= \mathbf{r}_n - \widetilde{\mathbf{R}}_n \Delta \mathbf{c}_n = \widetilde{\mathbf{R}}_n \mathbf{y}_n \\ &= (1 - \gamma_1^{(n)})\mathbf{r}_n + \textstyle\sum_{\ell=1}^{s-1} (\gamma_\ell^{(n)} - \gamma_{\ell+1}^{(n)})\mathbf{r}_{n-\ell} + \gamma_s^{(n)} \mathbf{r}_{n-s}\,. \end{aligned} \tag{6}$$

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

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

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

# IDR as Krylov subspace method

The IDR recurrences of the prototype IDR($s$) algorithm can be summarized by

$$\boxed{\begin{aligned} \mathbf{r}_n &:= (\mathbf{I} - \omega_j \mathbf{A})\, \mathbf{v}_{n-1}\,, \\ \mathbf{v}_n &:= \mathbf{r}_n - \widetilde{\mathbf{R}}_n \Delta \mathbf{c}_n = \widetilde{\mathbf{R}}_n \mathbf{y}_n \\ &= (1 - \gamma_1^{(n)}) \mathbf{r}_n + \textstyle\sum_{\ell=1}^{s-1} (\gamma_\ell^{(n)} - \gamma_{\ell+1}^{(n)}) \mathbf{r}_{n-\ell} + \gamma_s^{(n)} \mathbf{r}_{n-s}\,. \end{aligned}} \qquad (6)$$

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

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

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

Removing $\mathbf{v}_n$ from the recurrence we obtain the perturbed generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{R}_n \mathbf{Y}_n \mathbf{D}_\omega + \mathbf{F}_n = \mathbf{R}_{n+1} \underline{\mathbf{Y}}_n^\circ . \qquad (7)$$

# IDR as Krylov subspace method

By inspection, the banded Hessenberg matrix $\underline{\mathbf{Y}}_n^\circ$ has zero column sums. Inverting the upper triangular banded matrix $\mathbf{Y}_n\mathbf{D}_\omega$, we obtain the Hessenberg decomposition

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

Here, the Sonneveld matrix $\underline{\mathbf{S}}_n^\circ$ is defined as long as all $\omega_j \neq 0$ and all $\gamma_1^{(k)} \neq 1$.

# IDR as Krylov subspace method

By inspection, the banded Hessenberg matrix $\underline{\mathbf{Y}}_n^\circ$ has zero column sums. Inverting the upper triangular banded matrix $\mathbf{Y}_n\mathbf{D}_\omega$, we obtain the Hessenberg decomposition

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

Here, the Sonneveld matrix $\underline{\mathbf{S}}_n^\circ$ is defined as long as all $\omega_j \neq 0$ and all $\gamma_1^{(k)} \neq 1$.

The Sonneveld matrix has zero column sums and is thus of type ORTHORES. This implies that IDR follows the QOR approach.

# IDR as Krylov subspace method

By inspection, the banded Hessenberg matrix $\underline{\mathbf{Y}}_n^\circ$ has zero column sums. Inverting the upper triangular banded matrix $\mathbf{Y}_n\mathbf{D}_\omega$, we obtain the Hessenberg decomposition

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

Here, the Sonneveld matrix $\underline{\mathbf{S}}_n^\circ$ is defined as long as all $\omega_j \neq 0$ and all $\gamma_1^{(k)} \neq 1$.

The Sonneveld matrix has zero column sums and is thus of type ORTHORES. This implies that IDR follows the QOR approach.

By well-known results the residuals can be expressed in terms of the leading submatrices of the Sonneveld matrix,

$$\mathbf{r}_n = \mathcal{S}_n(\mathbf{A})\mathbf{r}_0, \quad \mathcal{S}_n(z) := \det\left(\mathbf{I}_n - z(\mathbf{S}_n^\circ)^{-1}\right) = \frac{\det\left(\mathbf{S}_n^\circ - z\mathbf{I}_n\right)}{\det\left(\mathbf{S}_n^\circ\right)}.$$

# IDR for eigenvalues

In unperturbed IDR the generalized Hessenberg decomposition is given by

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ \quad \Rightarrow \quad \mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\underline{\mathbf{S}}_n^\circ. \tag{9}$$

# IDR for eigenvalues

In unperturbed IDR the generalized Hessenberg decomposition is given by

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ \quad \Rightarrow \quad \mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\underline{\mathbf{S}}_n^\circ. \tag{9}$$

We can use the leading submatrices of the Sonneveld matrix $\mathbf{S}_n^\circ$ for the computation of Ritz values, the Ritz vectors are the eigenvectors prolonged by the “basis” given by $\mathbf{R}_n$. We can estimate the accuracy similar to Lanczos' method by looking at the last element of the eigenvector and the size of the current residual.

# IDR for eigenvalues

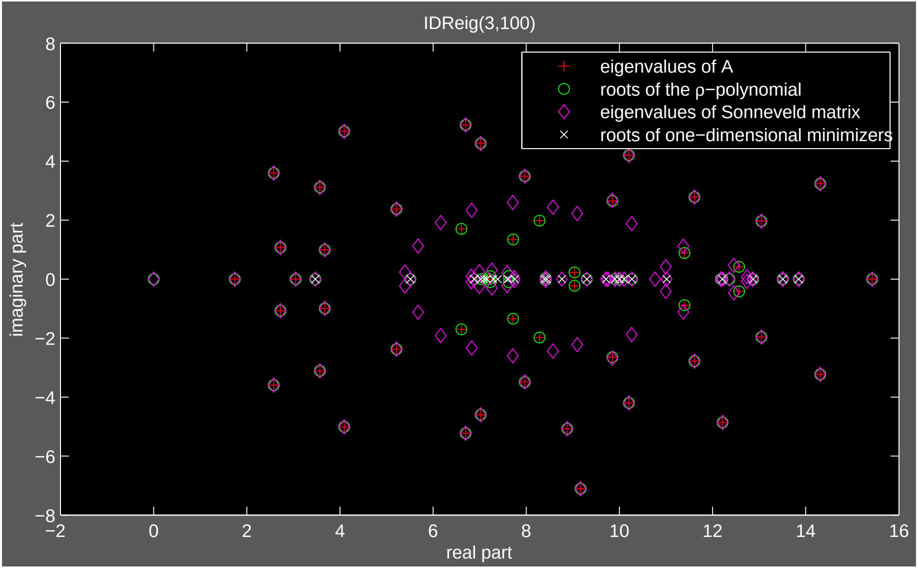
In unperturbed IDR the generalized Hessenberg decomposition is given by

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^\circ \quad \Rightarrow \quad \mathbf{A}\mathbf{R}_n = \mathbf{R}_{n+1}\underline{\mathbf{S}}_n^\circ. \tag{9}$$

We can use the leading submatrices of the Sonneveld matrix $\mathbf{S}_n^\circ$ for the computation of Ritz values, the Ritz vectors are the eigenvectors prolonged by the “basis” given by $\mathbf{R}_n$. We can estimate the accuracy similar to Lanczos' method by looking at the last element of the eigenvector and the size of the current residual.

Numerically more stable and more efficient is the use of the Sonneveld pencil $(\mathbf{Y}_n^\circ, \mathbf{Y}_n\mathbf{D}_\omega)$. The stability comes from the fact that we need not be afraid of a large condition of $\mathbf{Y}_n$ and/or $\mathbf{D}_\omega$. The efficiency is due to the structure: The Sonneveld *matrix* is a *full* unreduced Hessenberg matrix, the Sonneveld *pencil* is *banded* upper Hessenberg/triangular and QZ is the method of choice.

# IDR for eigenvalues



# IDR for eigenvalues

As the example reveals, it is quite troublesome to distinguish the wanted unknown zeros of the residual polynomials (corresponding to a two-sided Lanczos' process we are interested in) from the *known* inverse local minimizers $1/\omega_j$.

# IDR for eigenvalues

As the example reveals, it is quite troublesome to distinguish the wanted unknown zeros of the residual polynomials (corresponding to a two-sided Lanczos' process we are interested in) from the *known* inverse local minimizers $1/\omega_j$.

There seems to be a more stable way to compute approximations to $\mathbf{A}$'s eigenvalues, indicated by the green circles ;-)

# IDR for eigenvalues

As the example reveals, it is quite troublesome to distinguish the wanted unknown zeros of the residual polynomials (corresponding to a two-sided Lanczos' process we are interested in) from the *known* inverse local minimizers $1/\omega_j$.

There seems to be a more stable way to compute approximations to $\mathbf{A}$'s eigenvalues, indicated by the green circles ;-)

Indeed, in LTPM we only want to compute the zeros of *one* of the residual polynomial factors of the residual polynomial.

# IDR for eigenvalues

As the example reveals, it is quite troublesome to distinguish the wanted unknown zeros of the residual polynomials (corresponding to a two-sided Lanczos' process we are interested in) from the *known* inverse local minimizers $1/\omega_j$.

There seems to be a more stable way to compute approximations to $\mathbf{A}$'s eigenvalues, indicated by the green circles ;-)

Indeed, in LTPM we only want to compute the zeros of *one* of the residual polynomial factors of the residual polynomial. This can be done.

# IDR for eigenvalues

As the example reveals, it is quite troublesome to distinguish the wanted unknown zeros of the residual polynomials (corresponding to a two-sided Lanczos' process we are interested in) from the *known* inverse local minimizers $1/\omega_j$.

There seems to be a more stable way to compute approximations to $\mathbf{A}$'s eigenvalues, indicated by the green circles ;-)

Indeed, in LTPM we only want to compute the zeros of *one* of the residual polynomial factors of the residual polynomial. This can be done.

By the manner of construction of the residuals, which is based on the mappings $(\mathbf{I} - \omega_j \mathbf{A}) : \mathcal{G}_{j-1} \to \mathcal{G}_j$, we know that for some $\mathbf{w}_n \in \mathcal{G}_0 = \mathcal{K}(\mathbf{A}, \mathbf{r}_0)$

$$\begin{aligned} \mathbf{r}_n &= \Omega_j(\mathbf{A})\mathbf{w}_n, \\ \mathbf{v}_{n-1} &= \Omega_{j-1}(\mathbf{A})\mathbf{w}_n, \end{aligned} \qquad \Omega_j(z) = \prod_{\ell=1}^{j}(1 - \omega_\ell z), \quad j = \left\lfloor \frac{n}{s+1} \right\rfloor. \tag{10}$$

# IDR for eigenvalues

The polynomials defined by dividing the residual polynomials by the polynomials $\Omega_j$ are *residual polynomials* in the Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{r}_0)$. The polynomials $\Omega_j$ are also residual polynomials, since $\Omega_j(0) = 1$.

# IDR for eigenvalues

The polynomials defined by dividing the residual polynomials by the polynomials $\Omega_j$ are *residual polynomials* in the Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{r}_0)$. The polynomials $\Omega_j$ are also residual polynomials, since $\Omega_j(0) = 1$.

We derive the Hessenberg decomposition corresponding to the basis of the purified residuals $\mathbf{w}_n$. This way we only have to compute the unknown eigenvalue approximations instead of computing again the local minimizers.

# IDR for eigenvalues

The polynomials defined by dividing the residual polynomials by the polynomials $\Omega_j$ are *residual polynomials* in the Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{r}_0)$. The polynomials $\Omega_j$ are also residual polynomials, since $\Omega_j(0) = 1$.

We derive the Hessenberg decomposition corresponding to the basis of the purified residuals $\mathbf{w}_n$. This way we only have to compute the unknown eigenvalue approximations instead of computing again the local minimizers.

Some thinking results in the wanted purified generalized Hessenberg decomposition

$$\mathbf{A}\mathbf{W}_n\mathbf{U}_n\mathbf{D}_\omega = \mathbf{W}_{n+1}\underline{\mathbf{Y}}_n^\circ, \tag{11}$$

where the change from the original residuals $\mathbf{r}_n$ to the purified residuals $\mathbf{w}_n$ is reflected in the construction of the matrix $\mathbf{U}_n$ from $\mathbf{Y}_n$ by cutting out lower triangles from the band such that $\mathbf{U}_n$ is block-diagonal with alternating $s \times s$ upper triangular blocks and single zero elements at every multiple of $s + 1$.

# IDR for eigenvalues

The pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega)$ is regular and has $j = \lfloor n/(s+1) \rfloor$ infinite eigenvalues.

# IDR for eigenvalues

The pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega)$ is regular and has $j = \lfloor n/(s+1) \rfloor$ infinite eigenvalues.

We can show based on the properties of unreduced Hessenberg pencils (Z, 2006) and manipulation of equation (11) (Z, 2007) that scalar multiples of the leading determinants of the Hessenberg pencil ${}^z\mathbf{H}_n := (z\mathbf{U}_n\mathbf{D}_\omega - \mathbf{Y}_n^\circ)$ define the purified residual polynomials.

# IDR for eigenvalues

The pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega)$ is regular and has $j = \lfloor n/(s+1) \rfloor$ infinite eigenvalues.

We can show based on the properties of unreduced Hessenberg pencils (Z, 2006) and manipulation of equation (11) (Z, 2007) that scalar multiples of the leading determinants of the Hessenberg pencil ${}^z\mathbf{H}_n := (z\mathbf{U}_n\mathbf{D}_\omega - \mathbf{Y}_n^\circ)$ define the purified residual polynomials.

This suggests to use the leading submatrices of the banded pencil to compute the wanted eigenvalue approximations. Using QZ this can effectively be done.

# IDR for eigenvalues

The pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega)$ is regular and has $j = \lfloor n/(s+1) \rfloor$ infinite eigenvalues.

We can show based on the properties of unreduced Hessenberg pencils (Z, 2006) and manipulation of equation (11) (Z, 2007) that scalar multiples of the leading determinants of the Hessenberg pencil ${}^z\mathbf{H}_n := (z\mathbf{U}_n\mathbf{D}_\omega - \mathbf{Y}_n^\circ)$ define the purified residual polynomials.

This suggests to use the leading submatrices of the banded pencil to compute the wanted eigenvalue approximations. Using QZ this can effectively be done.

This has two drawbacks: We make no use of the known $\lfloor n/(s+1) \rfloor$ infinite eigenvalues and this does not reveal the structure of the underlying two-sided Lanczos' process.

# IDR for eigenvalues

The pencil $(\mathbf{Y}_n^\circ, \mathbf{U}_n\mathbf{D}_\omega)$ is regular and has $j = \lfloor n/(s+1) \rfloor$ infinite eigenvalues.

We can show based on the properties of unreduced Hessenberg pencils (Z, 2006) and manipulation of equation (11) (Z, 2007) that scalar multiples of the leading determinants of the Hessenberg pencil ${}^z\mathbf{H}_n := (z\mathbf{U}_n\mathbf{D}_\omega - \mathbf{Y}_n^\circ)$ define the purified residual polynomials.

This suggests to use the leading submatrices of the banded pencil to compute the wanted eigenvalue approximations. Using QZ this can effectively be done.

This has two drawbacks: We make no use of the known $\lfloor n/(s+1) \rfloor$ infinite eigenvalues and this does not reveal the structure of the underlying two-sided Lanczos' process.

Both drawbacks can be removed utilizing Schur's determinant formula.

# IDR for eigenvalues

Block-Gauß elimination applied to a typical block of the pencil results in

$$\begin{pmatrix} {}^{z}\mathbf{H}^{\star} & \mathbf{h}_c & \mathbf{L}^{\star} \\ \mathbf{e}_s^T & (\gamma^{\star}-1) & \mathbf{h}_r^T \\ \mathbf{O} & \mathbf{e}_1 & {}^{z}\mathbf{H}_{\star} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{o} & \mathbf{O} \\ -\mathbf{e}_s^T/(\gamma^{\star}-1) & 1/(\gamma^{\star}-1) & -\mathbf{h}_r^T/(\gamma^{\star}-1) \\ \mathbf{O} & \mathbf{o} & \mathbf{I} \end{pmatrix} =$$

$$\begin{pmatrix} {}^{z}\mathbf{H}^{\star} - \mathbf{h}_c\mathbf{e}_s^T/(\gamma^{\star}-1) & \mathbf{h}_c^T/(\gamma^{\star}-1) & \mathbf{L}^{\star} - \mathbf{h}_c\mathbf{h}_r^T/(\gamma^{\star}-1) \\ \mathbf{o}^T & 1 & \mathbf{o}^T \\ -\mathbf{e}_1\mathbf{e}_s^T/(\gamma^{\star}-1) & \mathbf{e}_1^T/(\gamma^{\star}-1) & {}^{z}\mathbf{H}_{\star} - \mathbf{e}_1\mathbf{h}_r^T/(\gamma^{\star}-1) \end{pmatrix}. \quad (12)$$

# IDR for eigenvalues

Block-Gauß elimination applied to a typical block of the pencil results in

$$\begin{pmatrix} {}^z\mathbf{H}^\star & \mathbf{h}_c & \mathbf{L}^\star \\ \mathbf{e}_s^T & (\gamma^\star - 1) & \mathbf{h}_r^T \\ \mathbf{O} & \mathbf{e}_1 & {}^z\mathbf{H}_\star \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{o} & \mathbf{O} \\ -\mathbf{e}_s^T/(\gamma^\star - 1) & 1/(\gamma^\star - 1) & -\mathbf{h}_r^T/(\gamma^\star - 1) \\ \mathbf{O} & \mathbf{o} & \mathbf{I} \end{pmatrix} =$$
$$\begin{pmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{h}_c^T/(\gamma^\star - 1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^T/(\gamma^\star - 1) \\ \mathbf{o}^T & 1 & \mathbf{o}^T \\ -\mathbf{e}_1\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{e}_1^T/(\gamma^\star - 1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^T/(\gamma^\star - 1) \end{pmatrix}. \quad (12)$$

This shows that we can work on a deflated pencil, here depicted block-wise,

$$\begin{pmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^T/(\gamma^\star - 1) \\ -\mathbf{e}_1\mathbf{e}_s^T/(\gamma^\star - 1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^T/(\gamma^\star - 1) \end{pmatrix}. \quad (13)$$

# IDR for eigenvalues

Block-Gauß elimination applied to a typical block of the pencil results in

$$\begin{pmatrix} {}^z\mathbf{H}^\star & \mathbf{h}_c & \mathbf{L}^\star \\ \mathbf{e}_s^T & (\gamma^\star - 1) & \mathbf{h}_r^T \\ \mathbf{O} & \mathbf{e}_1 & {}^z\mathbf{H}_\star \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{o} & \mathbf{O} \\ -\mathbf{e}_s^T/(\gamma^\star - 1) & 1/(\gamma^\star - 1) & -\mathbf{h}_r^T/(\gamma^\star - 1) \\ \mathbf{O} & \mathbf{o} & \mathbf{I} \end{pmatrix} =$$
$$\begin{pmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{h}_c^T/(\gamma^\star - 1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^T/(\gamma^\star - 1) \\ \mathbf{o}^T & 1 & \mathbf{o}^T \\ -\mathbf{e}_1\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{e}_1^T/(\gamma^\star - 1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^T/(\gamma^\star - 1) \end{pmatrix}. \qquad (12)$$

This shows that we can work on a deflated pencil, here depicted block-wise,

$$\begin{pmatrix} {}^z\mathbf{H}^\star - \mathbf{h}_c\mathbf{e}_s^T/(\gamma^\star - 1) & \mathbf{L}^\star - \mathbf{h}_c\mathbf{h}_r^T/(\gamma^\star - 1) \\ -\mathbf{e}_1\mathbf{e}_s^T/(\gamma^\star - 1) & {}^z\mathbf{H}_\star - \mathbf{e}_1\mathbf{h}_r^T/(\gamma^\star - 1) \end{pmatrix}. \qquad (13)$$

This pencil again is of ORTHORES-type as the column sums of the deflated Hessenberg matrix are zero.

# IDR for eigenvalues

As we did remove the infinite eigenvalues, i.e., the zero blocks from the block-diagonal upper triangular matrix $\mathbf{U}_n$, we can now invert the deflated matrix $D(\mathbf{Y}_n\mathbf{D}_\omega\mathbf{G}_n)$ and multiply it from the right to the deflated Hessenberg matrix $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$.

# IDR for eigenvalues

As we did remove the infinite eigenvalues, i.e., the zero blocks from the block-diagonal upper triangular matrix $\mathbf{U}_n$, we can now invert the deflated matrix $D(\mathbf{Y}_n\mathbf{D}_\omega\mathbf{G}_n)$ and multiply it from the right to the deflated Hessenberg matrix $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$.

Here, $\mathbf{G}_n$ denotes the block-Gauß eliminator and $D$ denotes the deflation operator $D(\mathbf{M}) = \mathbf{M}(\mathrm{ind}, \mathrm{ind})$, where ind denotes the set of indices to remain. We remark that $\mathbf{Y}_n\mathbf{D}_\omega$ is not altered by application of $\mathbf{G}_n$.

# IDR for eigenvalues

As we did remove the infinite eigenvalues, i.e., the zero blocks from the block-diagonal upper triangular matrix $\mathbf{U}_n$, we can now invert the deflated matrix $D(\mathbf{Y}_n\mathbf{D}_\omega\mathbf{G}_n)$ and multiply it from the right to the deflated Hessenberg matrix $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$.

Here, $\mathbf{G}_n$ denotes the block-Gauß eliminator and $D$ denotes the deflation operator $D(\mathbf{M}) = \mathbf{M}(\text{ind}, \text{ind})$, where ind denotes the set of indices to remain. We remark that $\mathbf{Y}_n\mathbf{D}_\omega$ is not altered by application of $\mathbf{G}_n$.

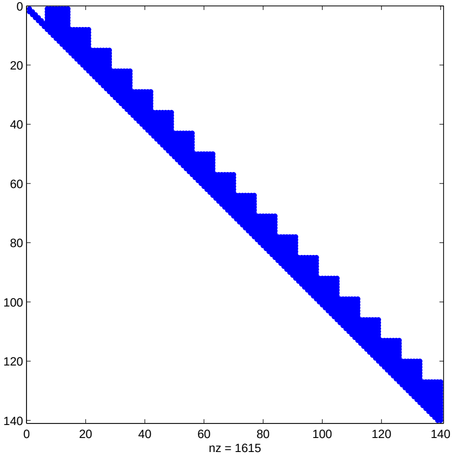
As $D(\mathbf{Y}_n^\circ\mathbf{G}_n)$ is of ORTHORES-type, Hessenberg, and block tridiagonal with blocks of size $s \times s$, and as $D(\mathbf{Y}_n\mathbf{D}_\omega\mathbf{G}_n)$ is block-diagonal upper triangular with blocks of size $s \times s$, the resulting matrix

$$\mathbf{P}_n^\circ := D(\mathbf{Y}_n^\circ\mathbf{G}_n)(D(\mathbf{Y}_n\mathbf{D}_\omega\mathbf{G}_n))^{-1} \tag{14}$$

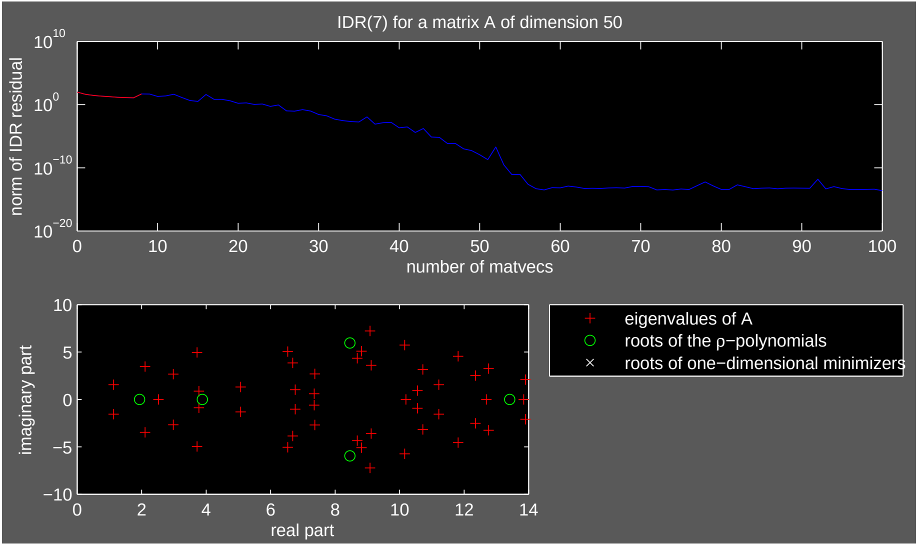
is the matrix of the ORTHORES-form of the underlying two-sided Lanczos' process with $s$ left and one right starting vectors.

# IDR for eigenvalues

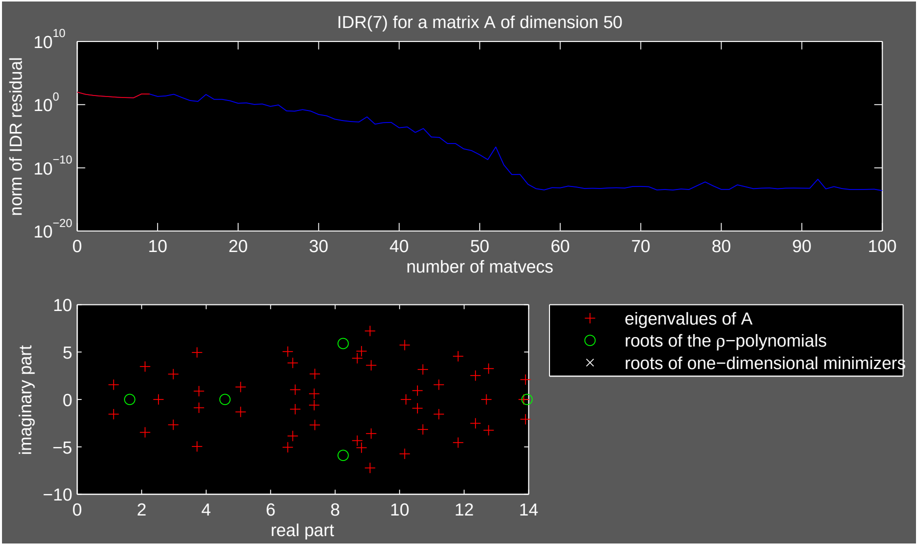
The following picture shows the structure of the resulting matrix $\mathbf{P}_n^\circ$ of the deflated purified process for IDR$(7)$ applied for $160$ steps.



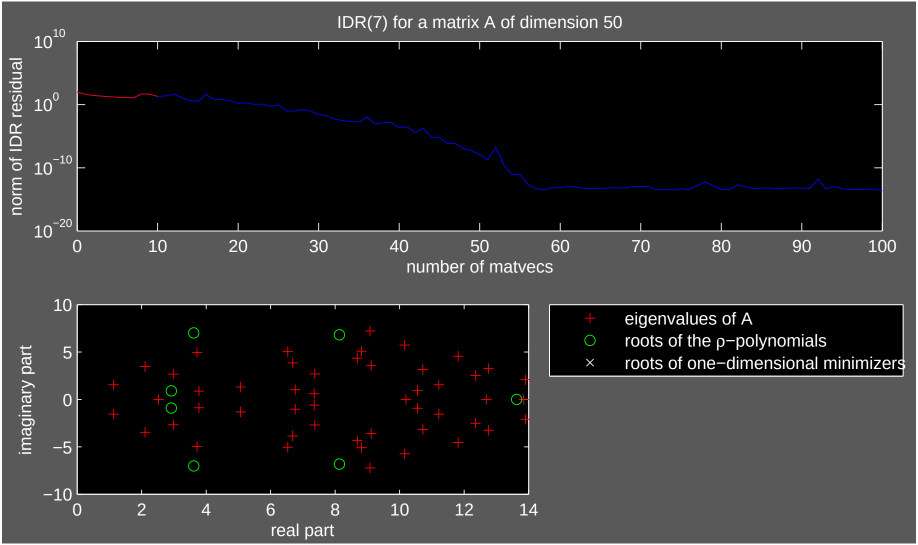
# One run of IDR(7)



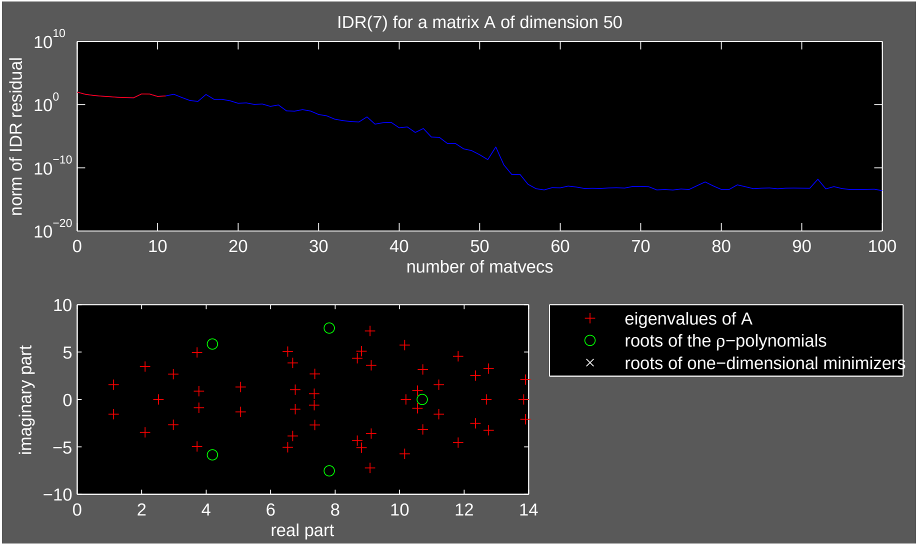
# One run of IDR(7)



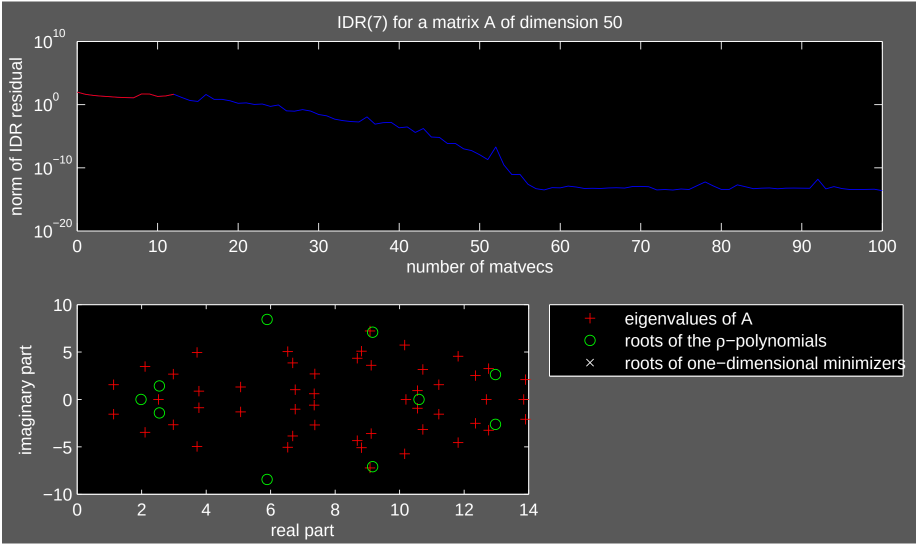
# One run of IDR(7)



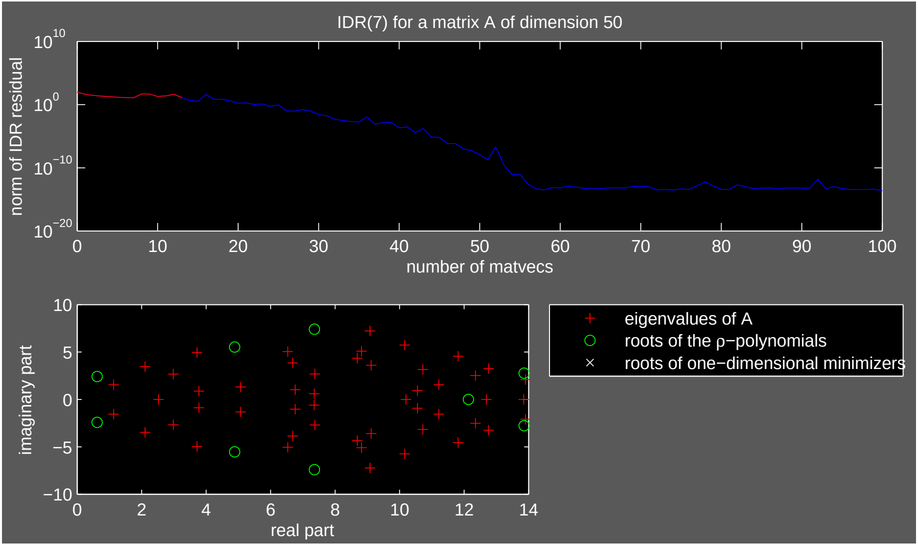
# One run of IDR(7)



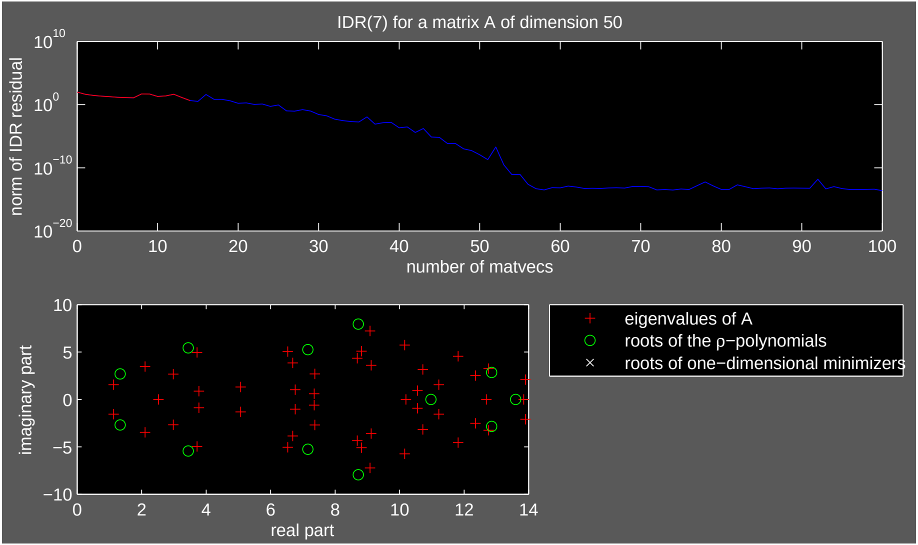
# One run of IDR(7)



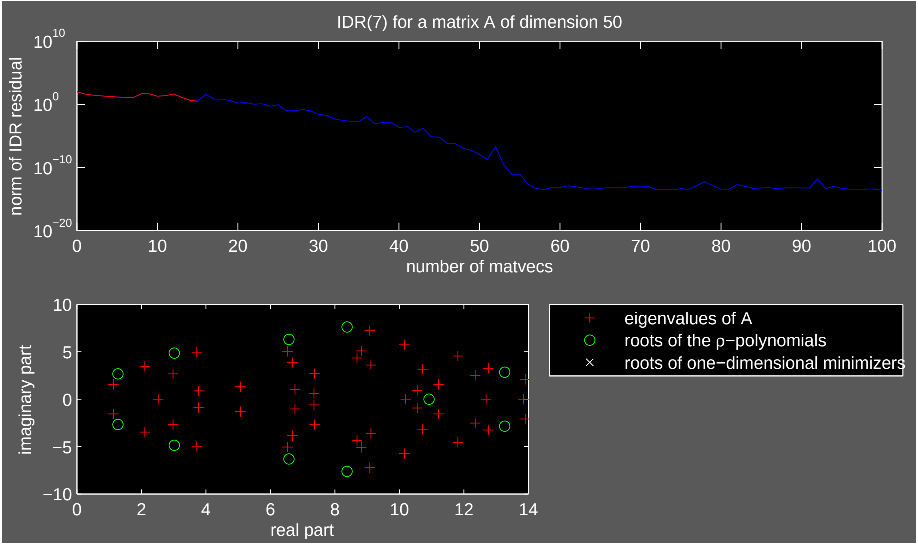
# One run of IDR(7)



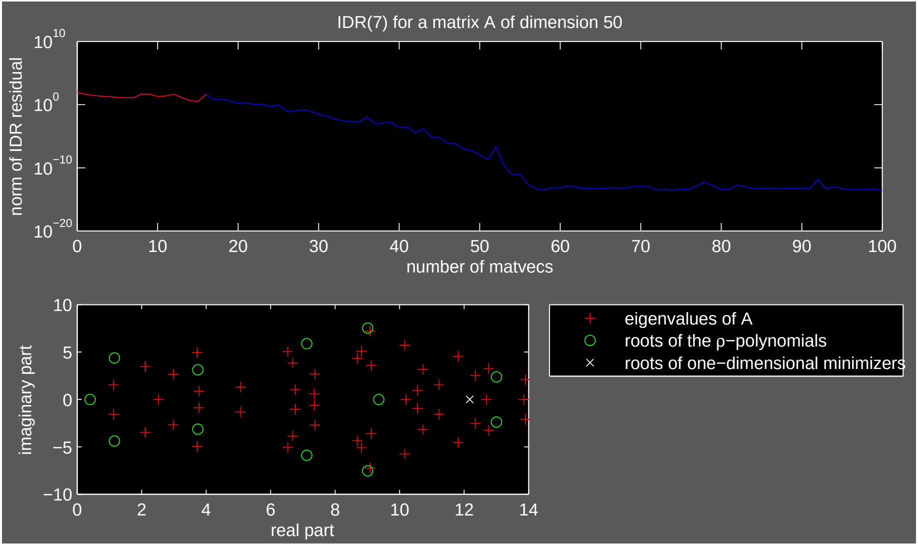
# One run of IDR(7)



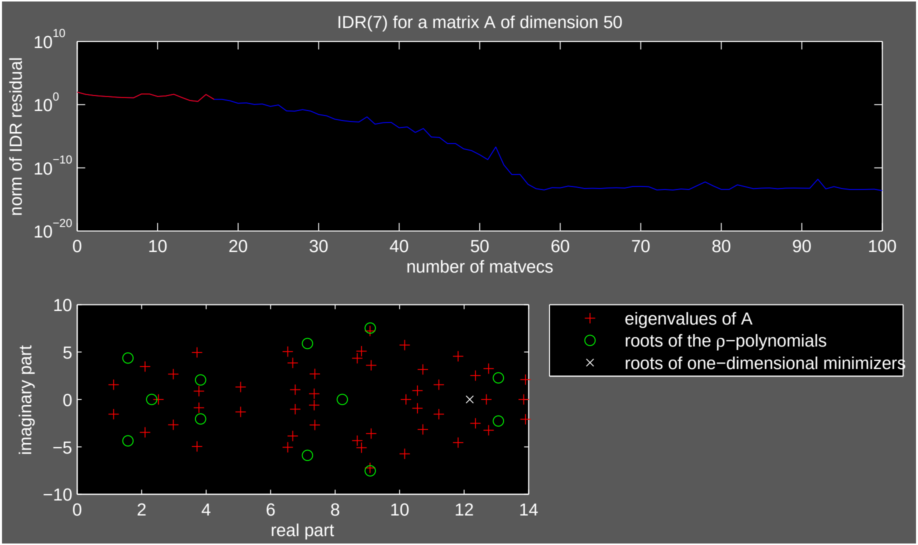
# One run of IDR(7)



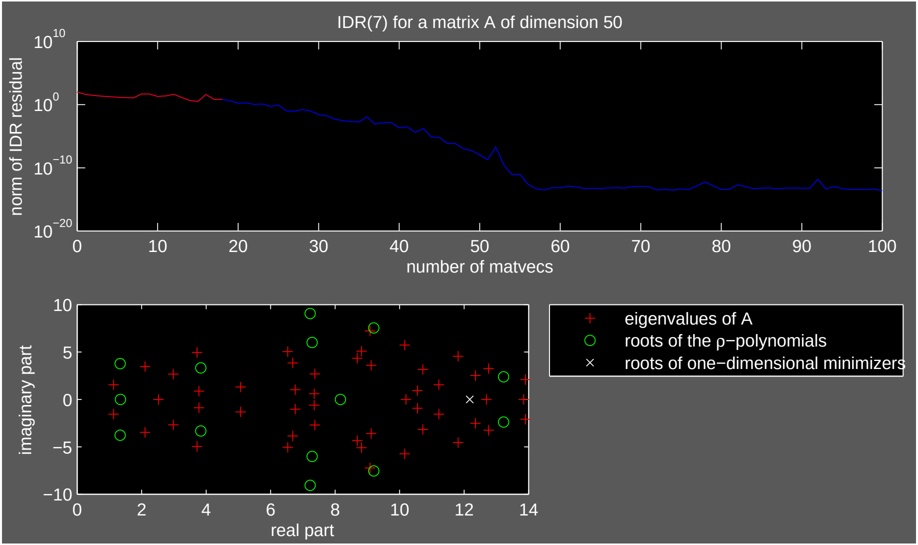
# One run of IDR(7)



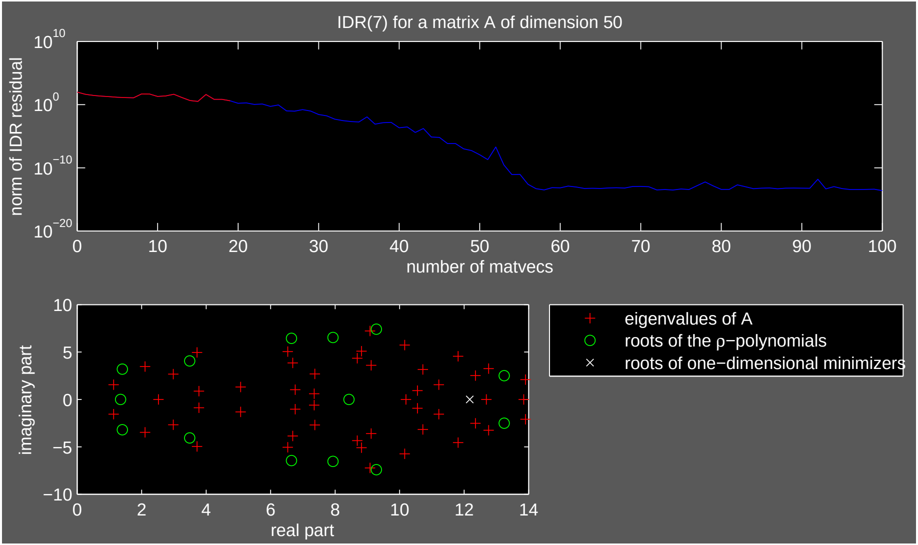
# One run of IDR(7)



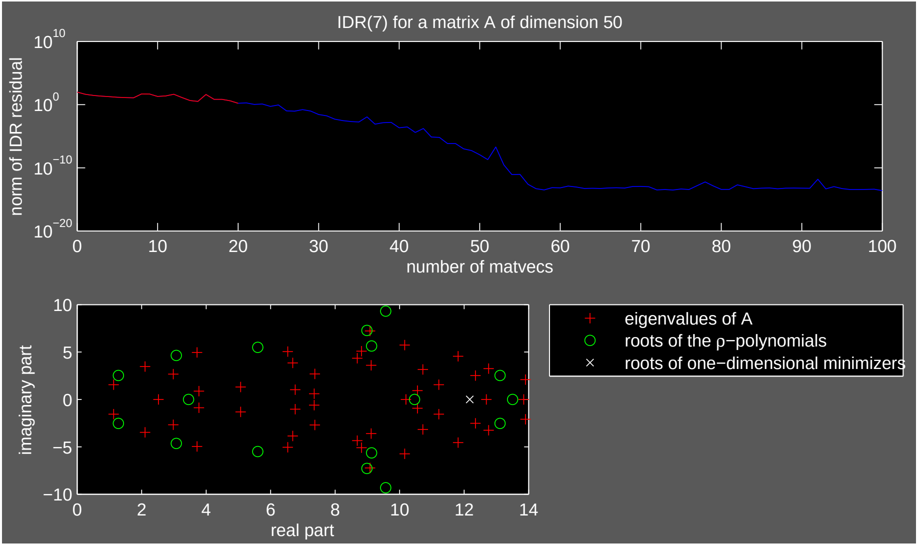
# One run of IDR(7)



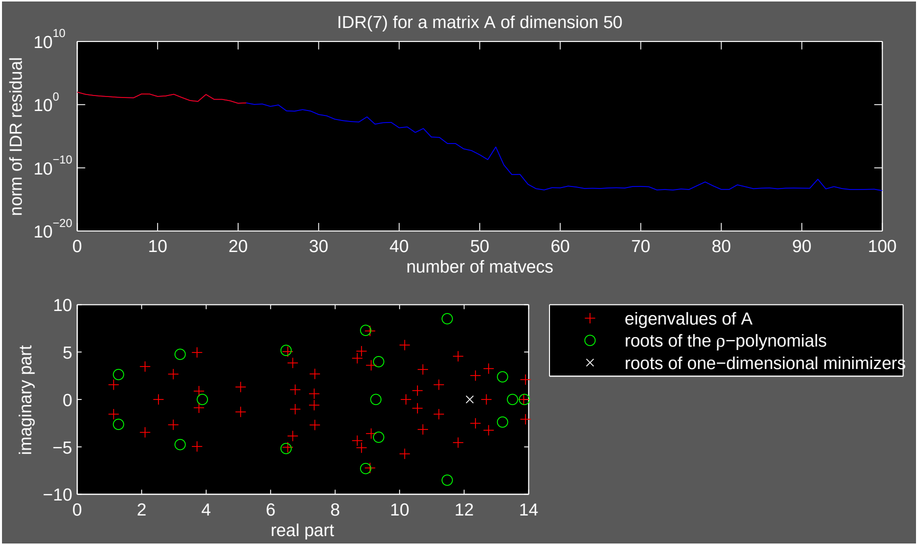
# One run of IDR(7)



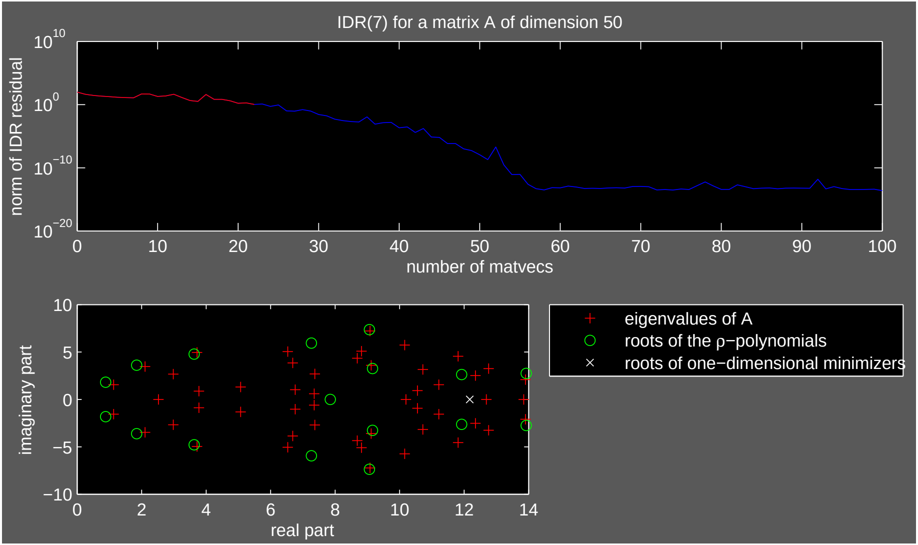
# One run of IDR(7)



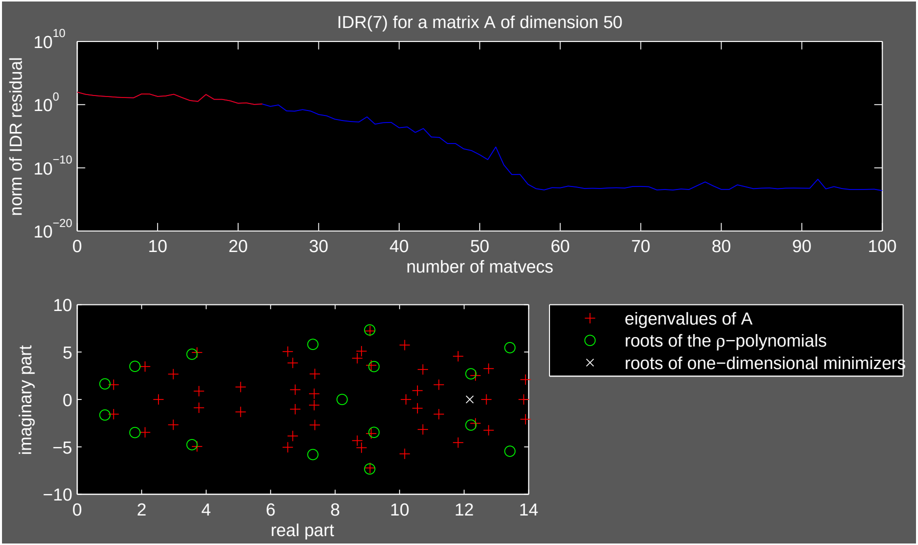
# One run of IDR(7)



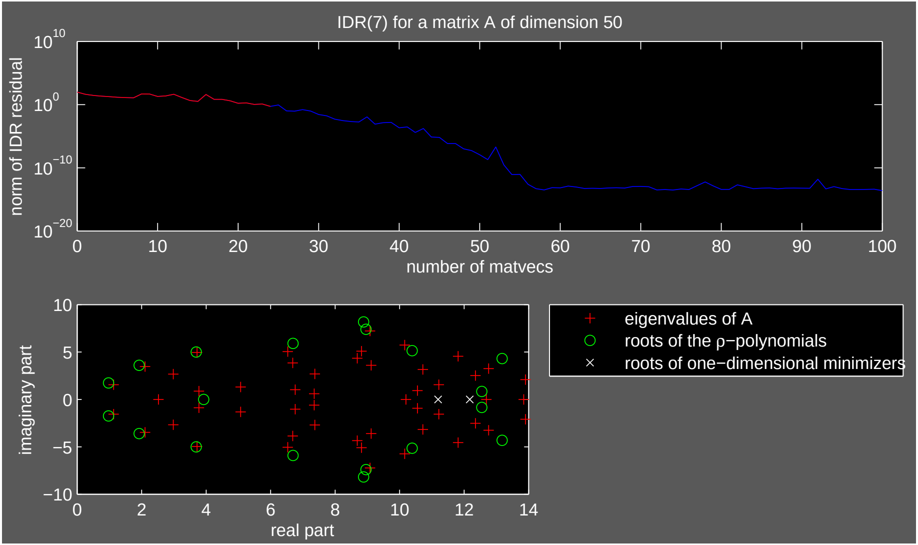
# One run of IDR(7)



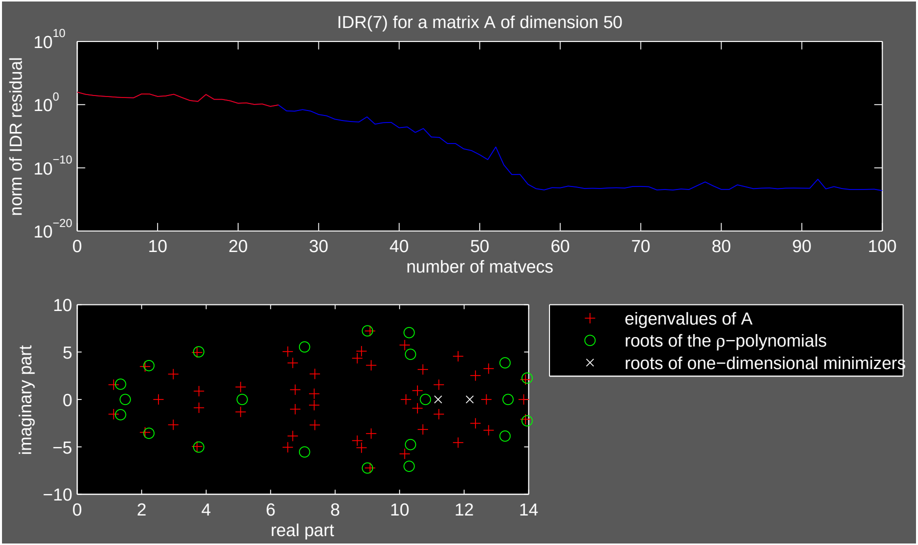
# One run of IDR(7)



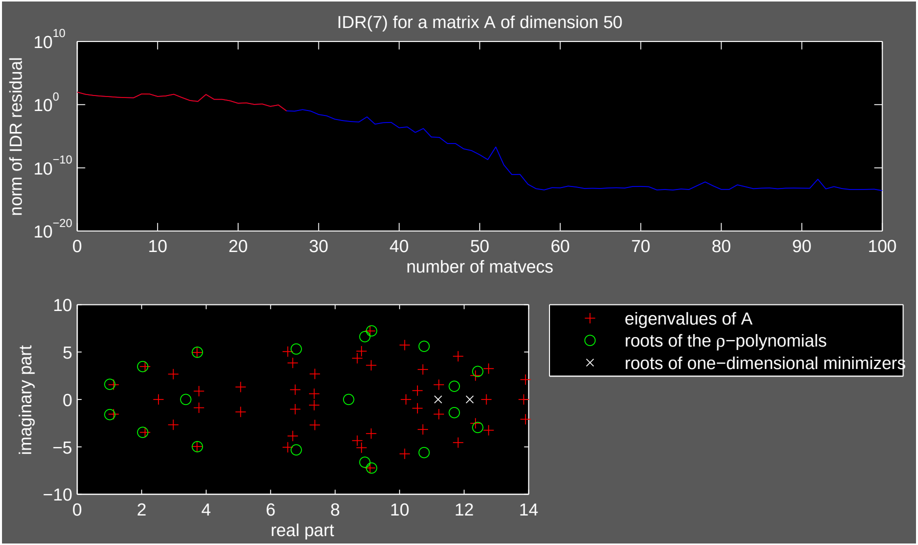
# One run of IDR(7)



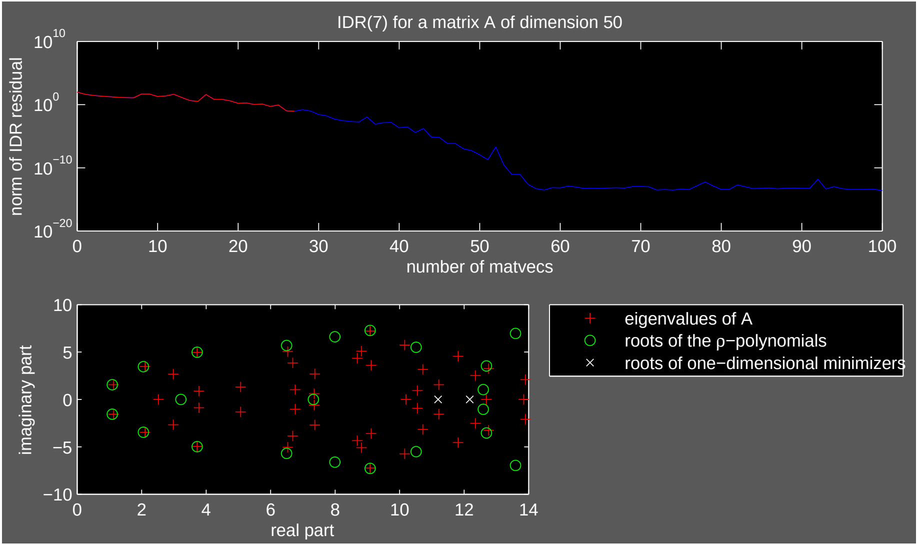
# One run of IDR(7)



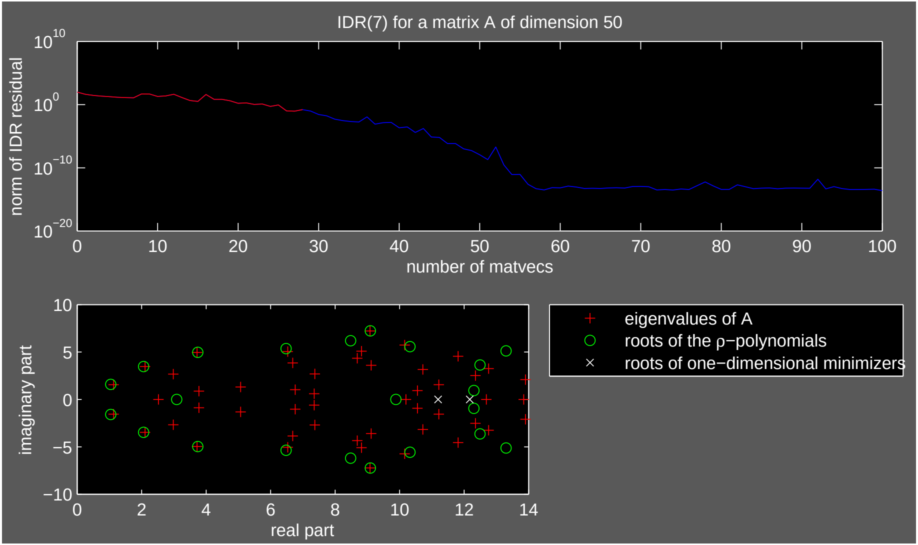
# One run of IDR(7)



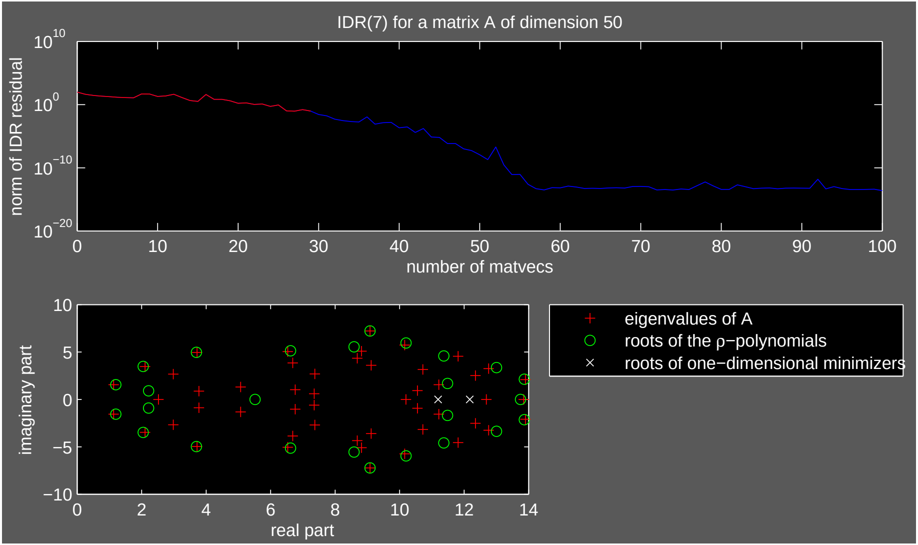
# One run of IDR(7)



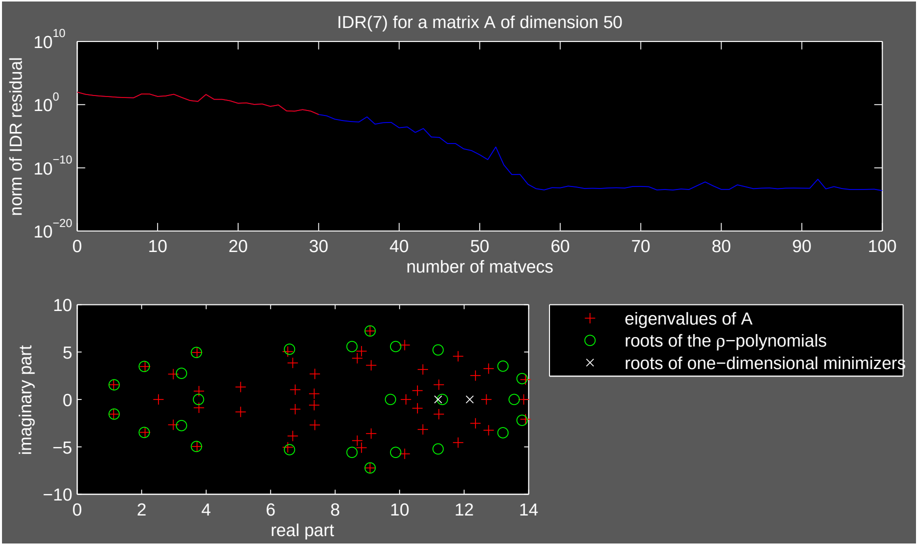
# One run of IDR(7)



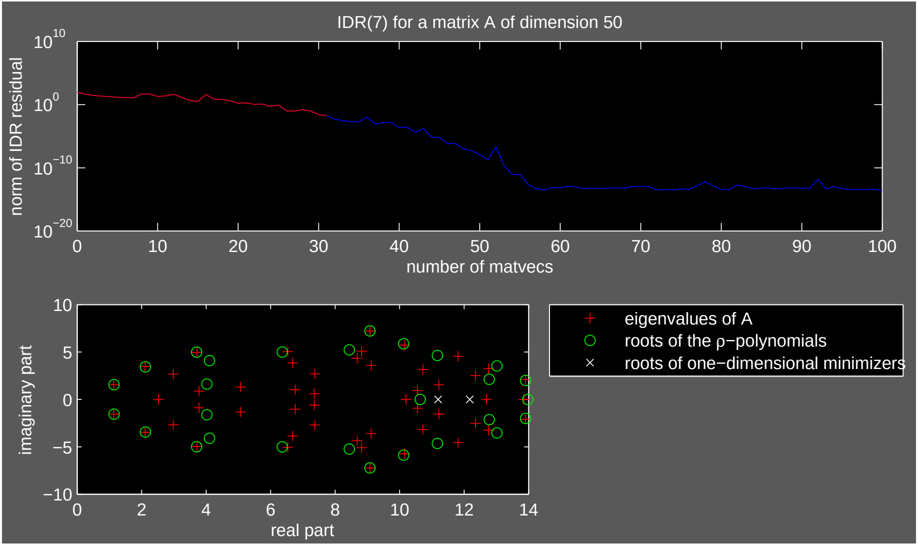
# One run of IDR(7)



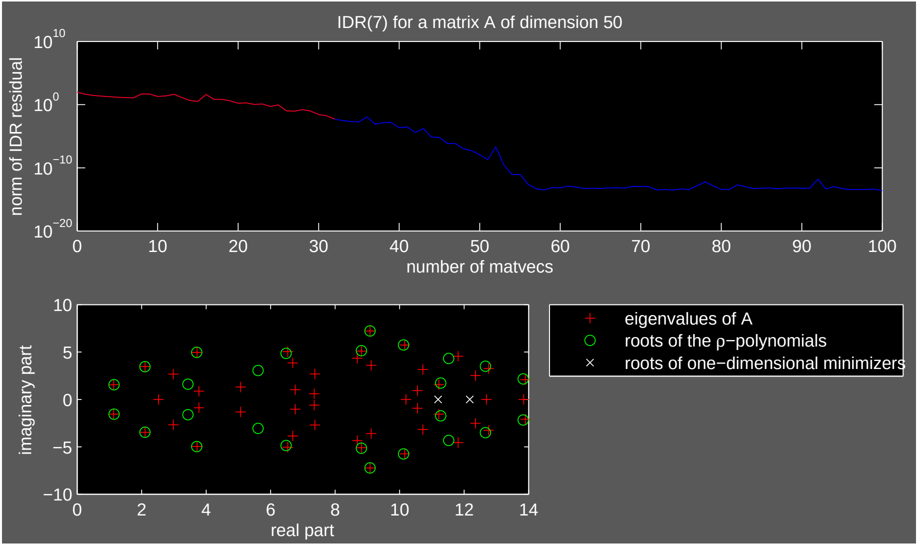
# One run of IDR(7)



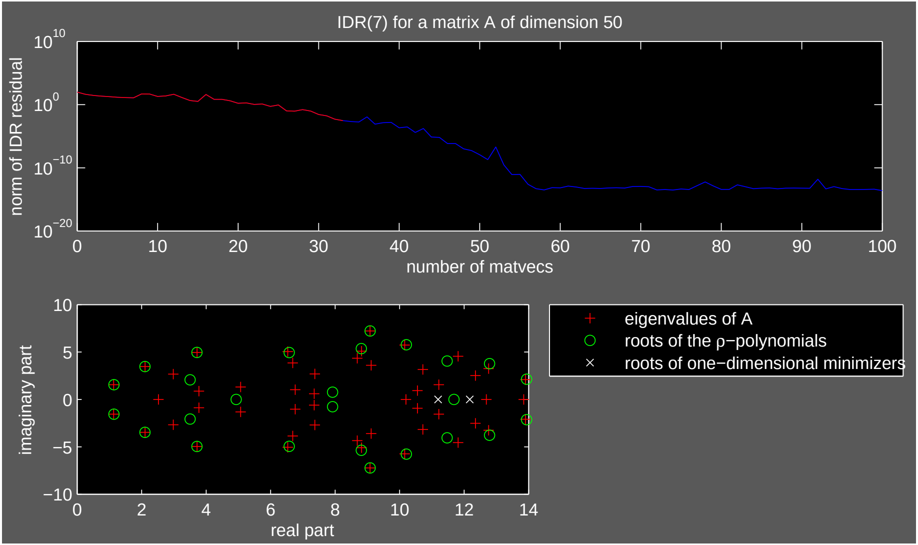
# One run of IDR(7)



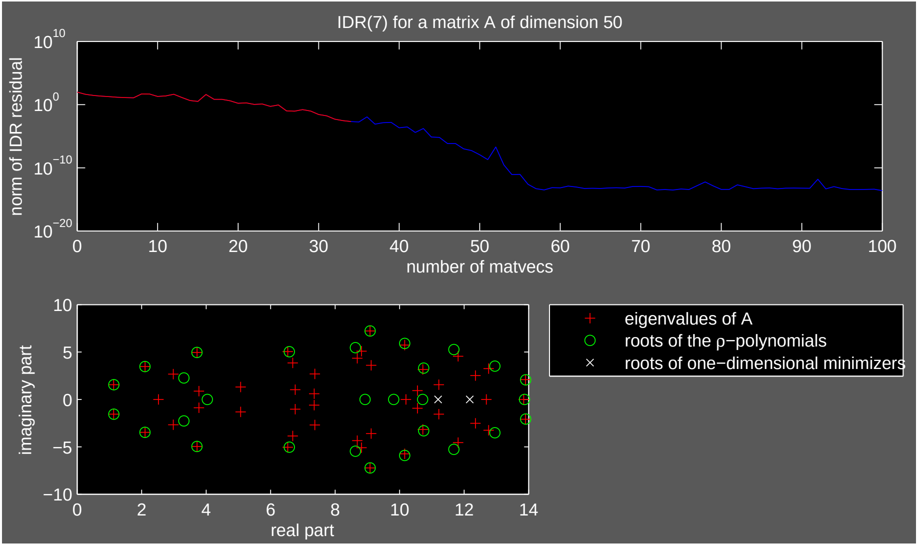
# One run of IDR(7)



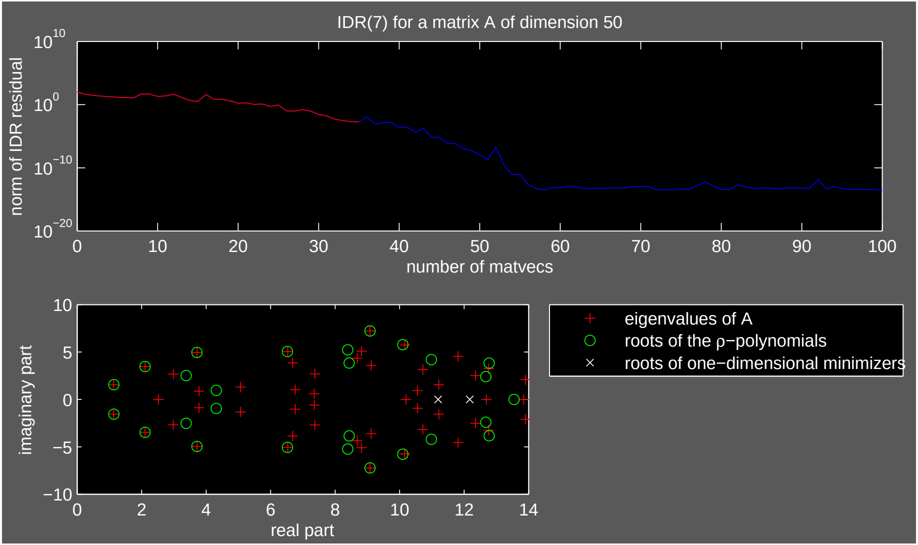
# One run of IDR(7)



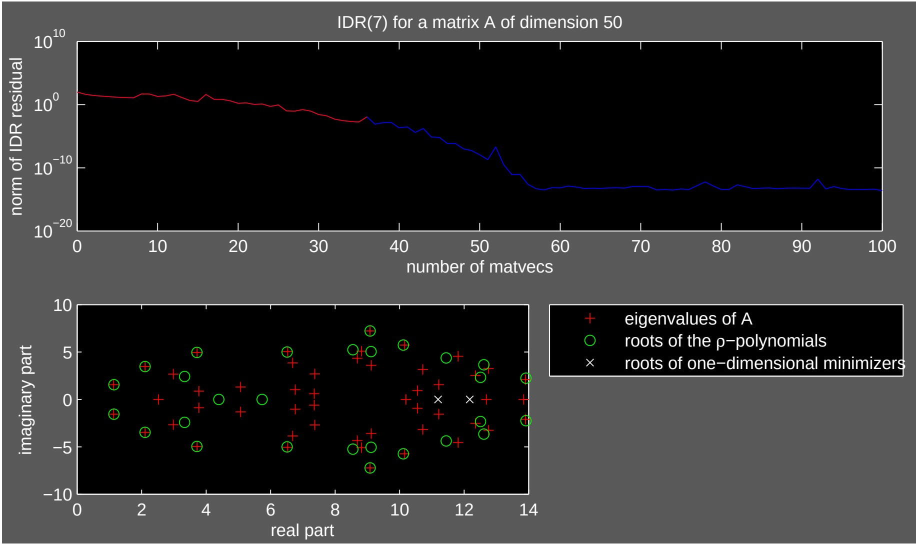
# One run of IDR(7)



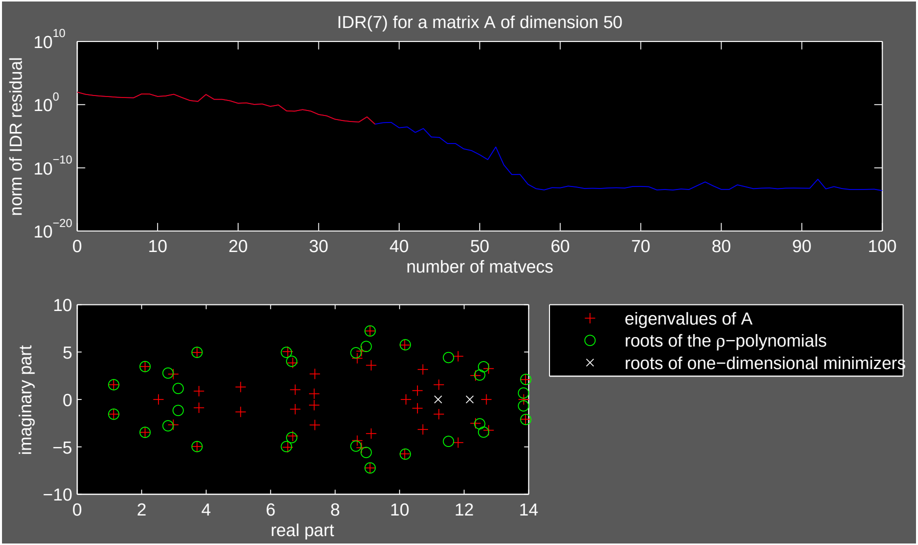
# One run of IDR(7)



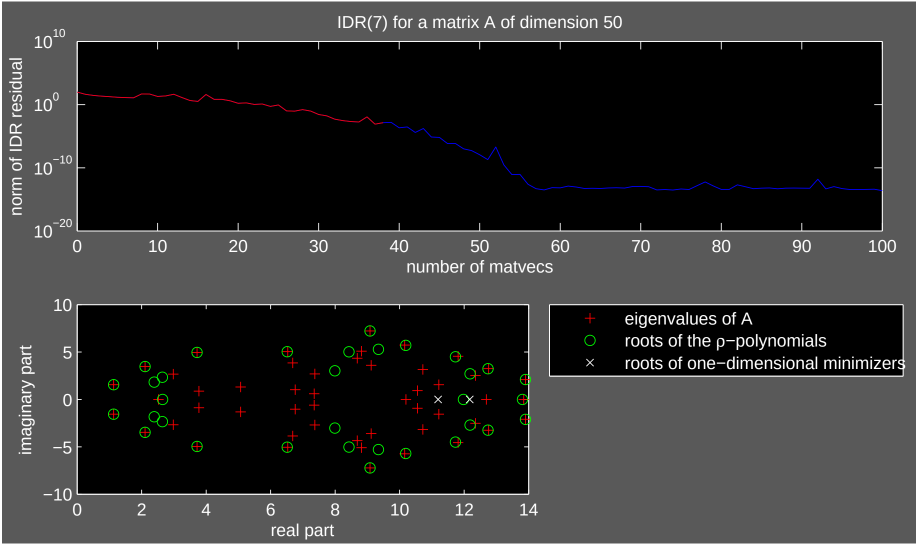
# One run of IDR(7)



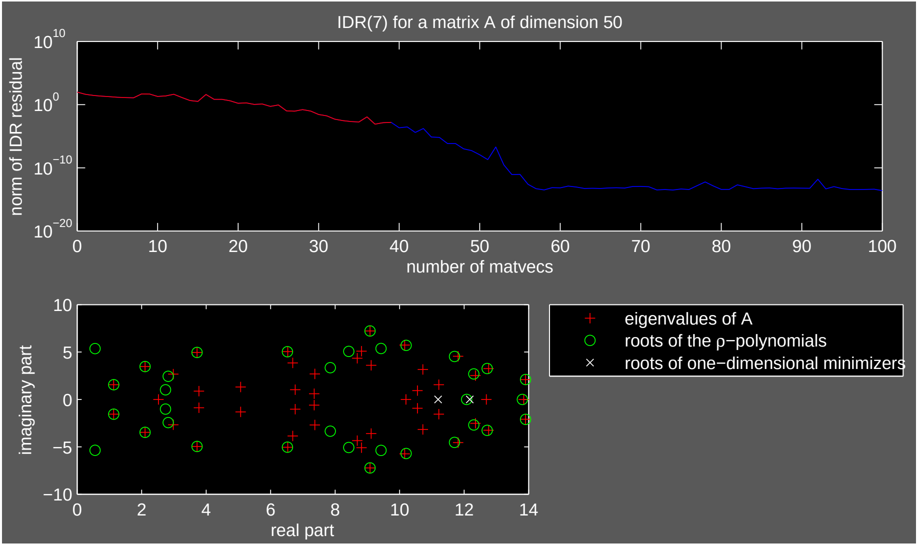
# One run of IDR(7)



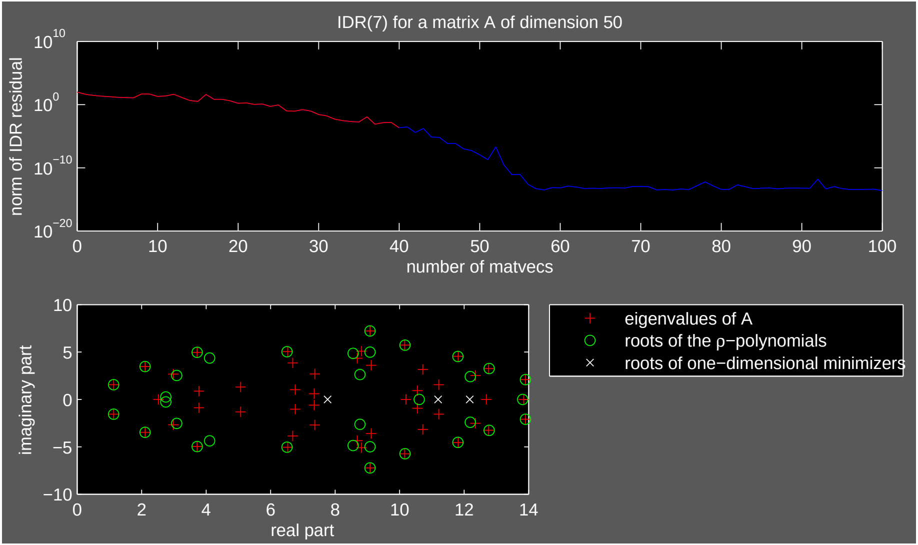
# One run of IDR(7)



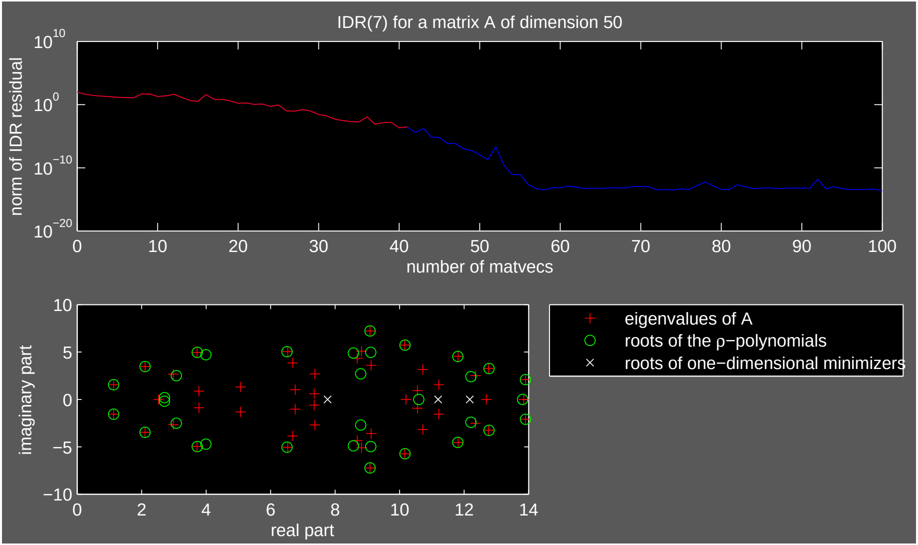
# One run of IDR(7)



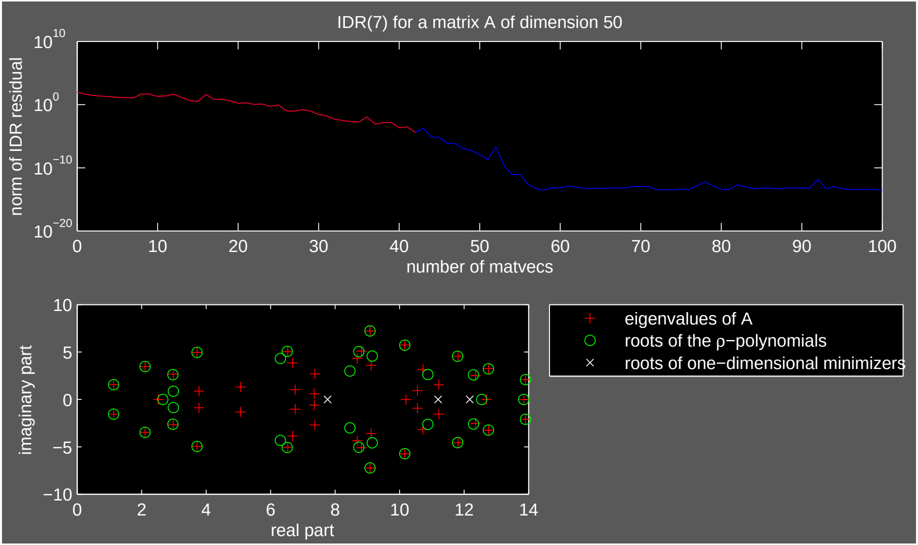
# One run of IDR(7)



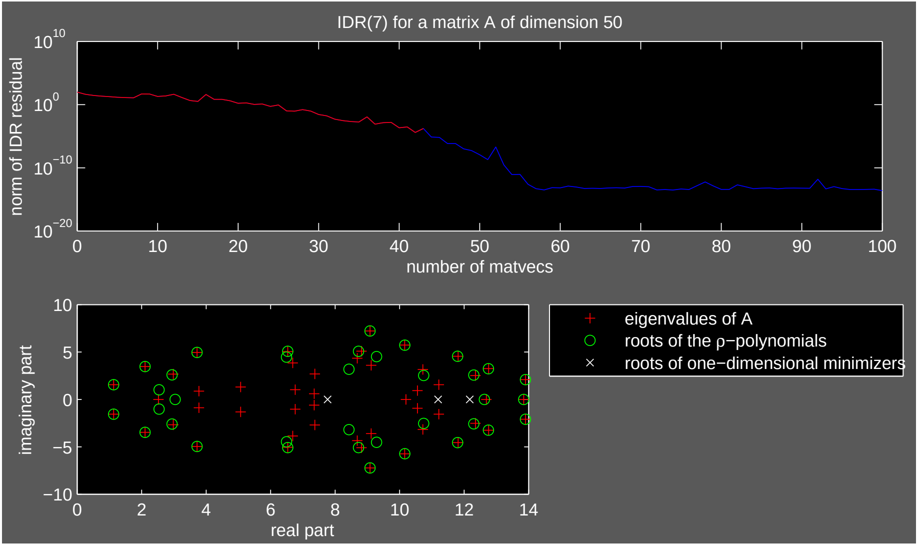
# One run of IDR(7)



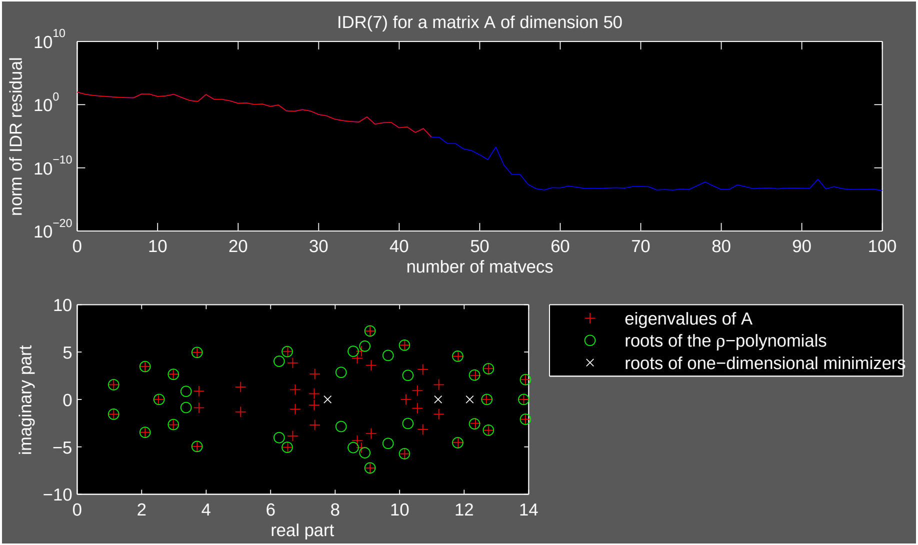
# One run of IDR(7)



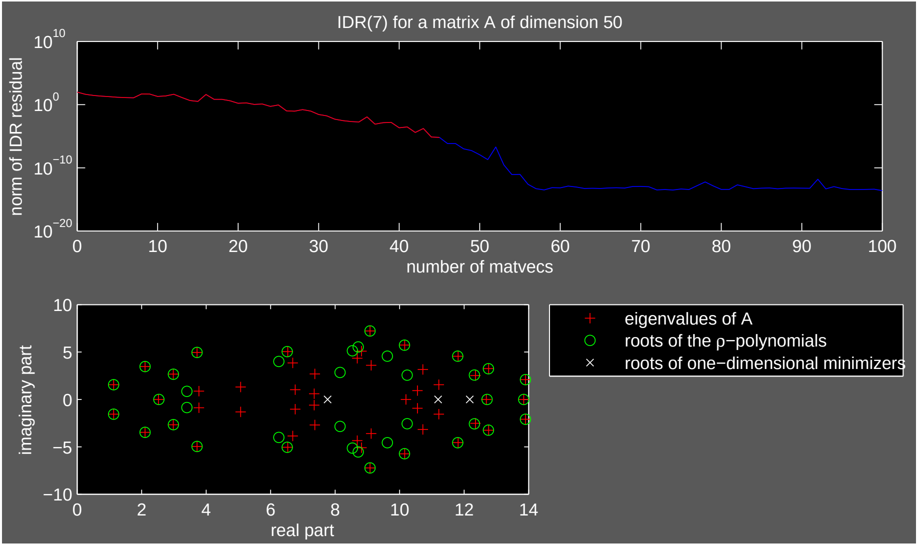
# One run of IDR(7)



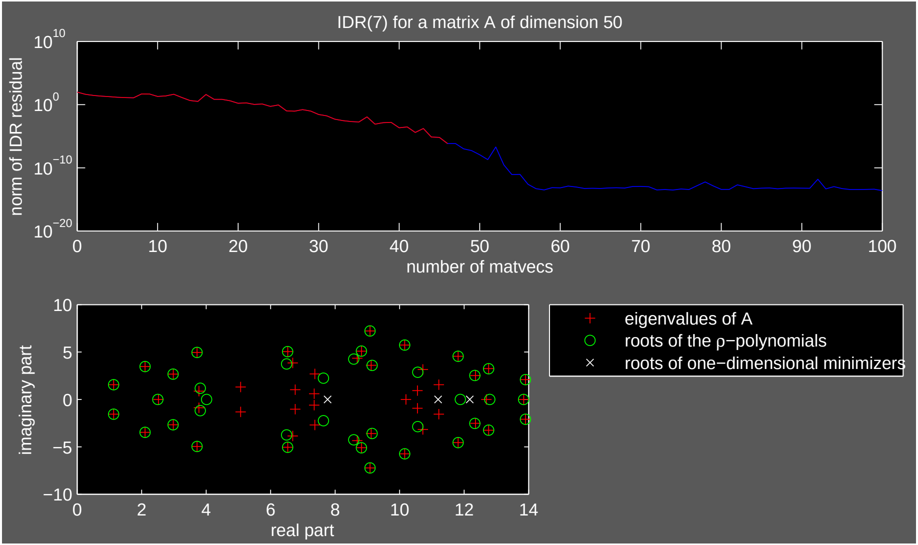
# One run of IDR(7)



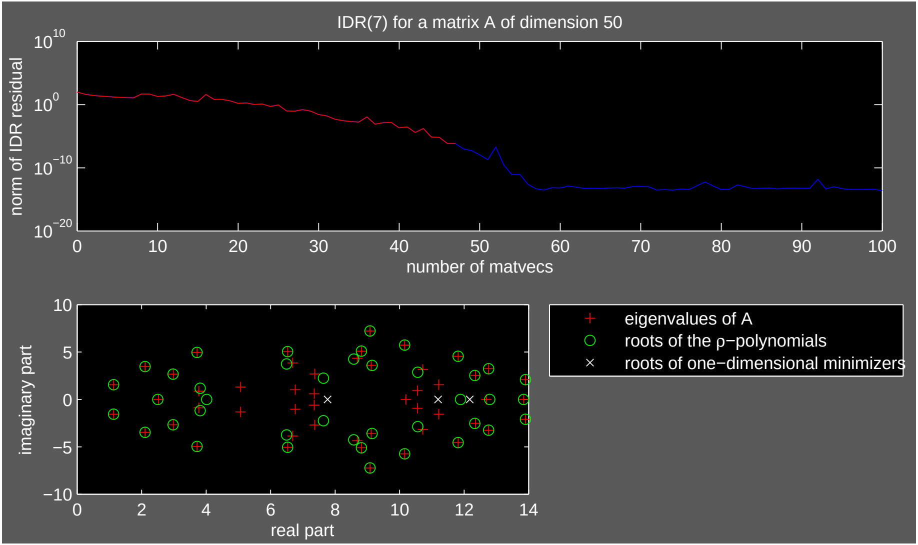
# One run of IDR(7)



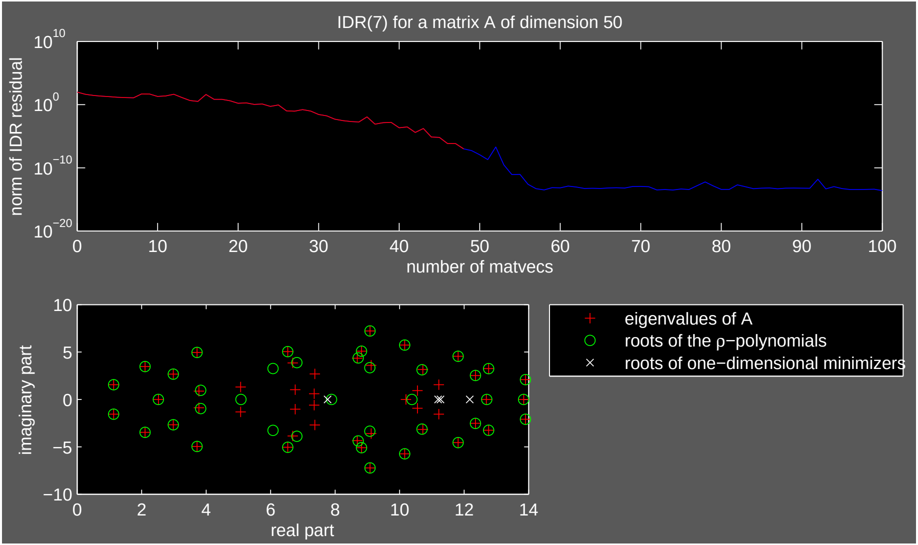
# One run of IDR(7)



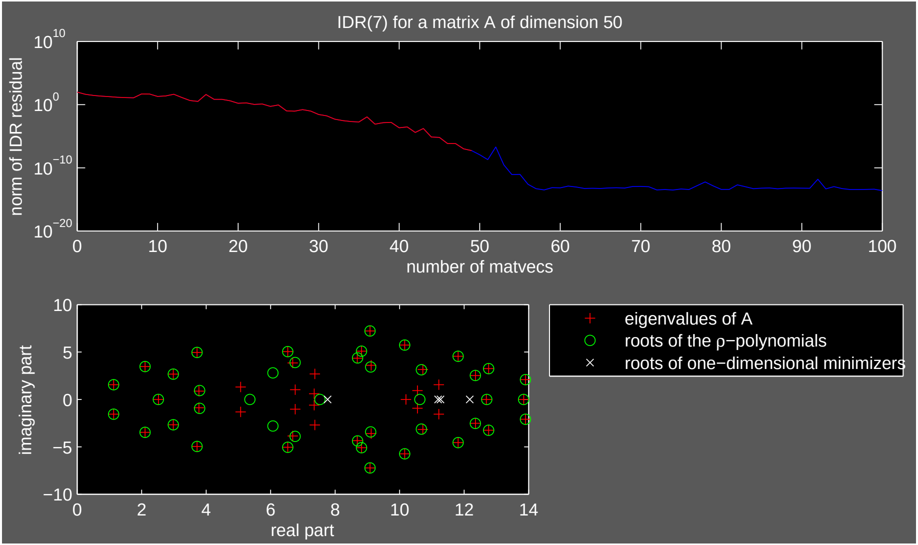
# One run of IDR(7)



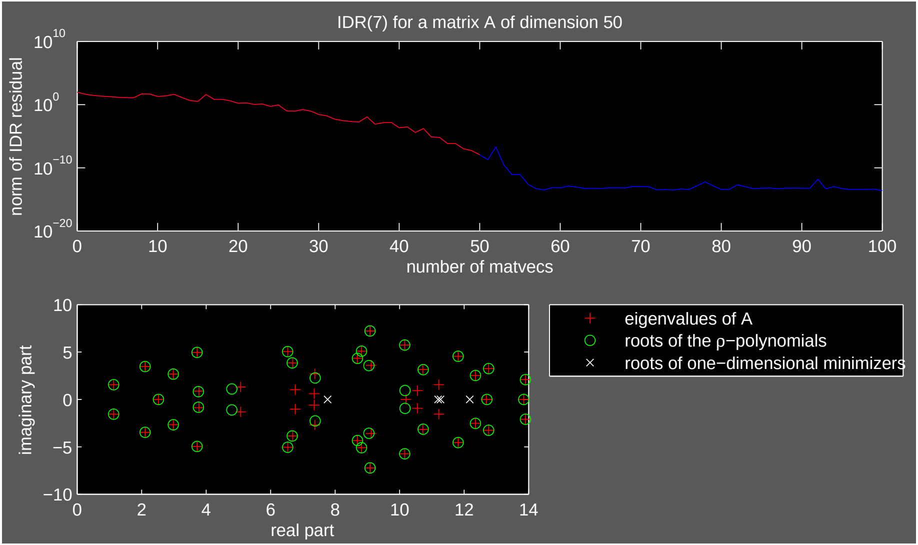
# One run of IDR(7)



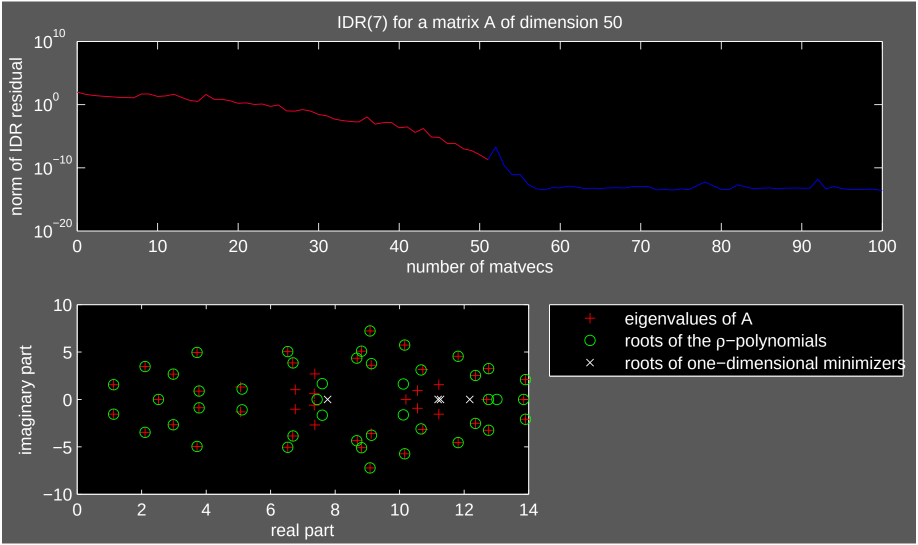
# One run of IDR(7)



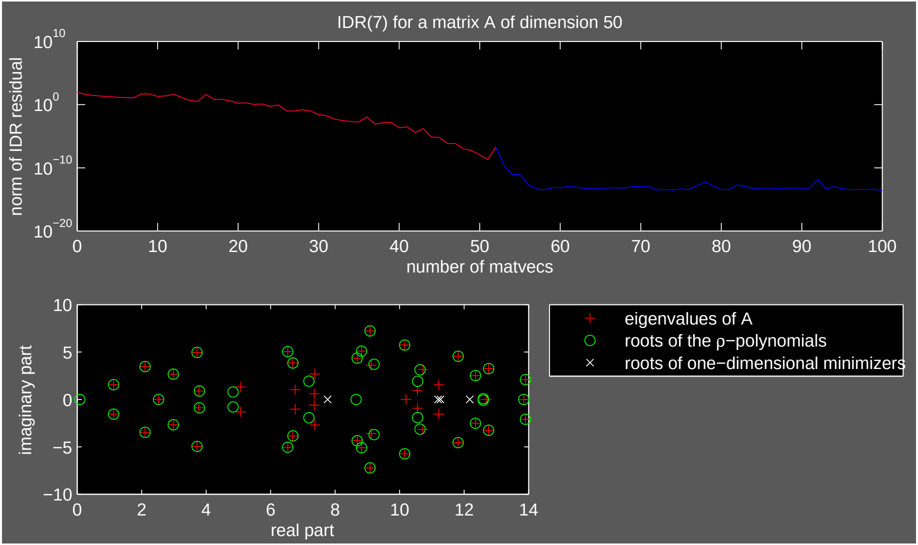
# One run of IDR(7)



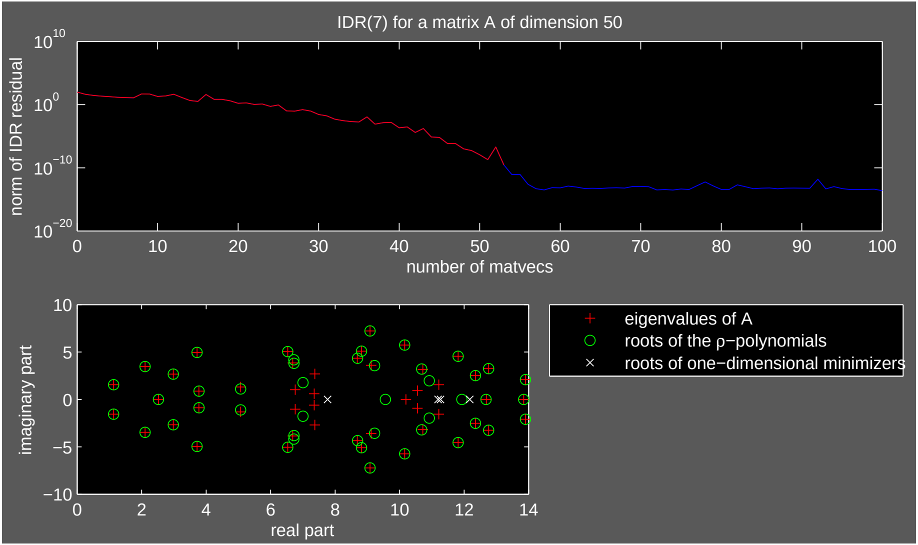
# One run of IDR(7)



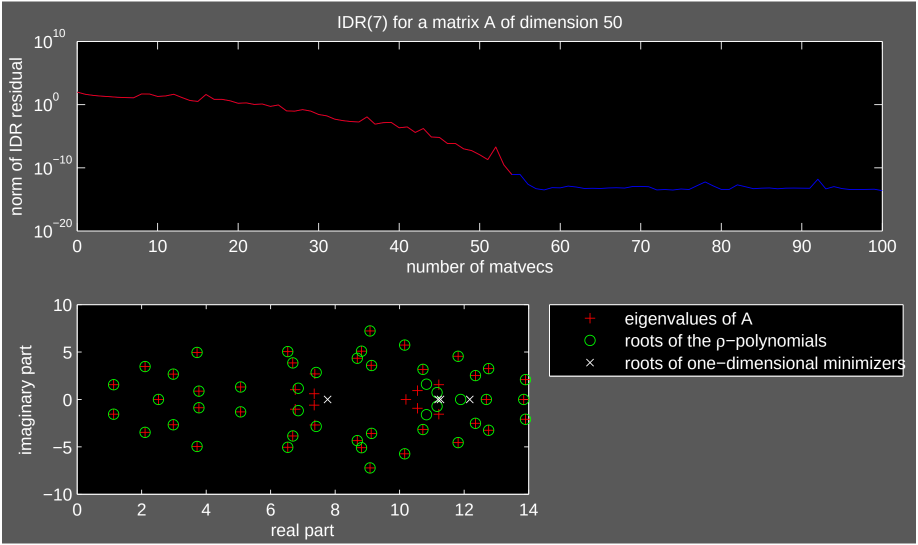
# One run of IDR(7)



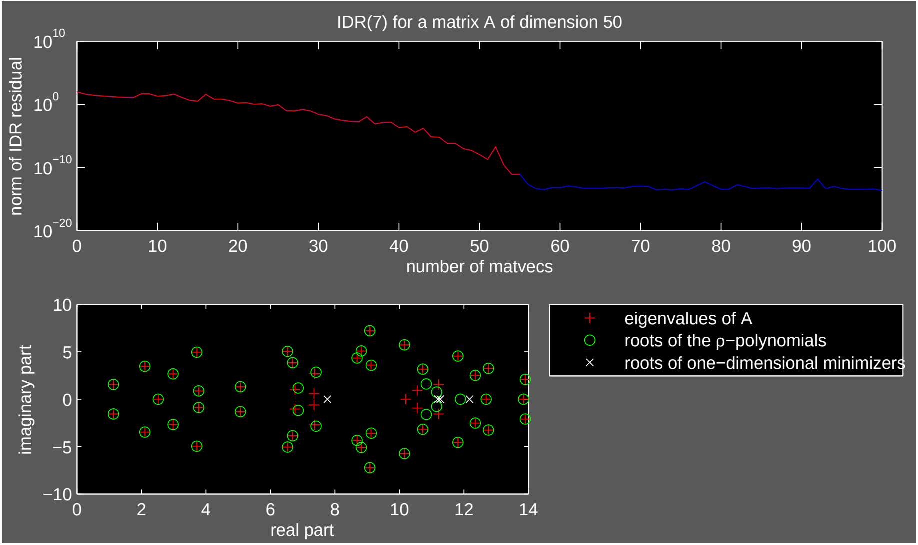
# One run of IDR(7)



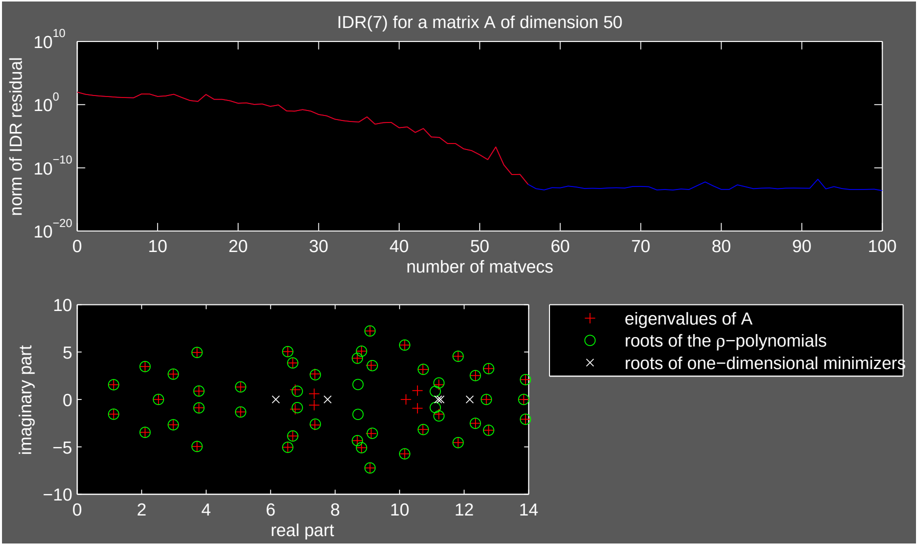
# One run of IDR(7)



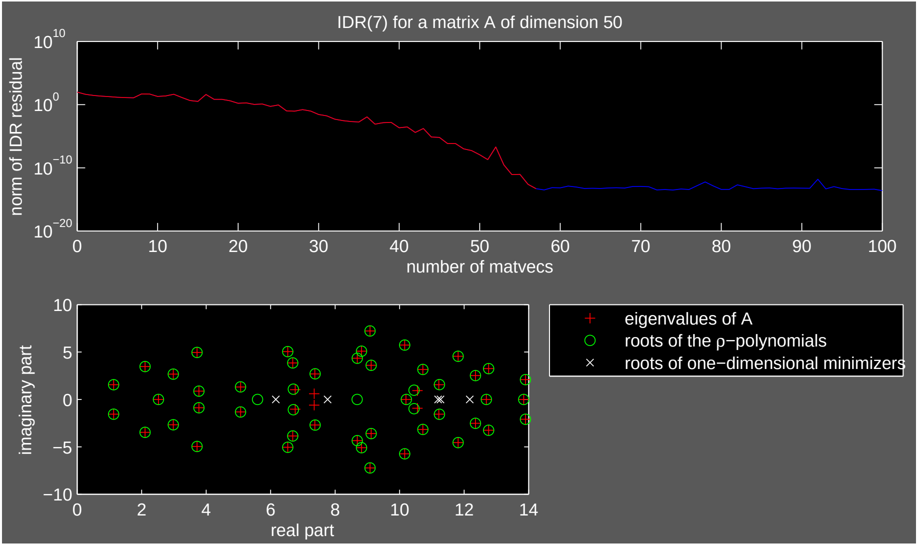
# One run of IDR(7)



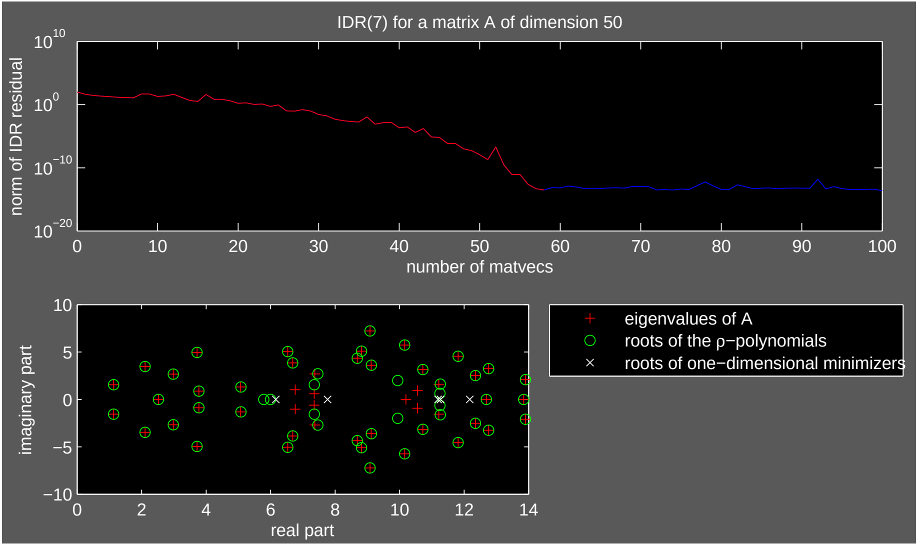
# One run of IDR(7)



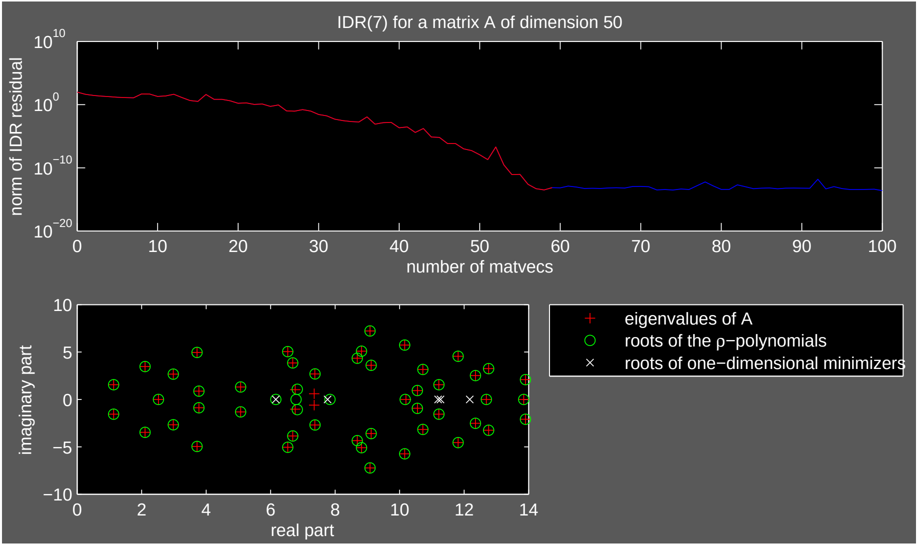
# One run of IDR(7)



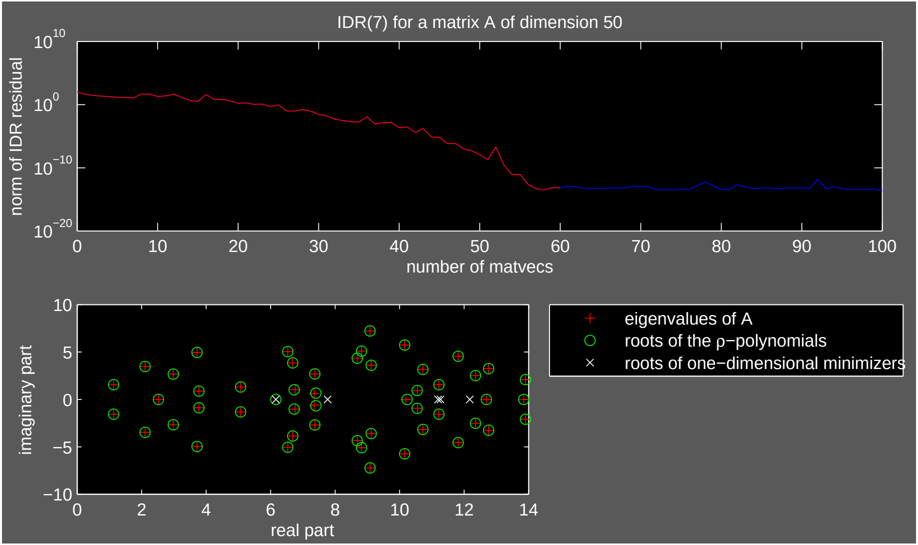
# One run of IDR(7)



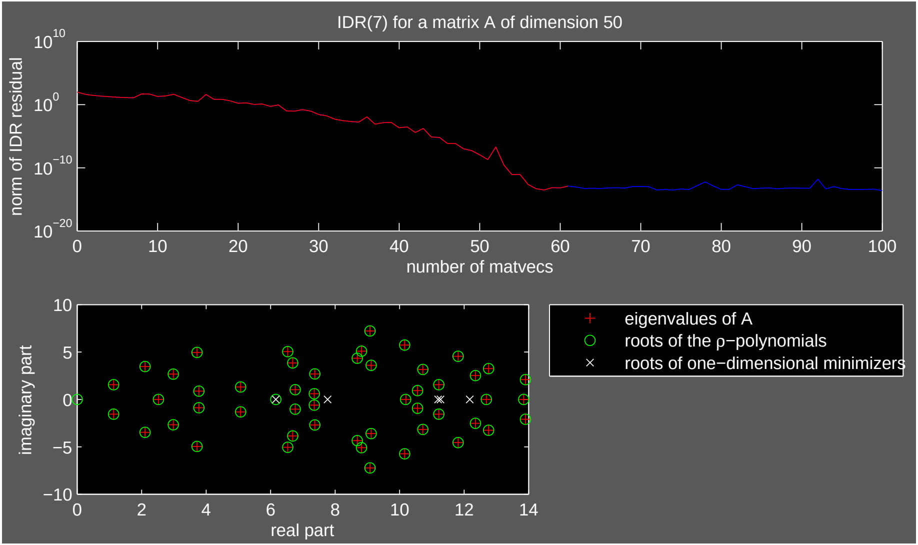
# One run of IDR(7)



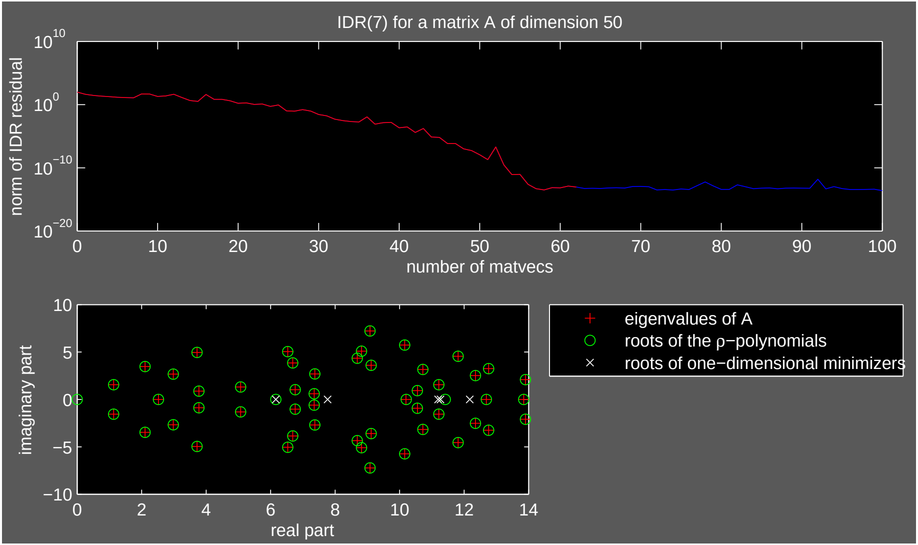
# One run of IDR(7)



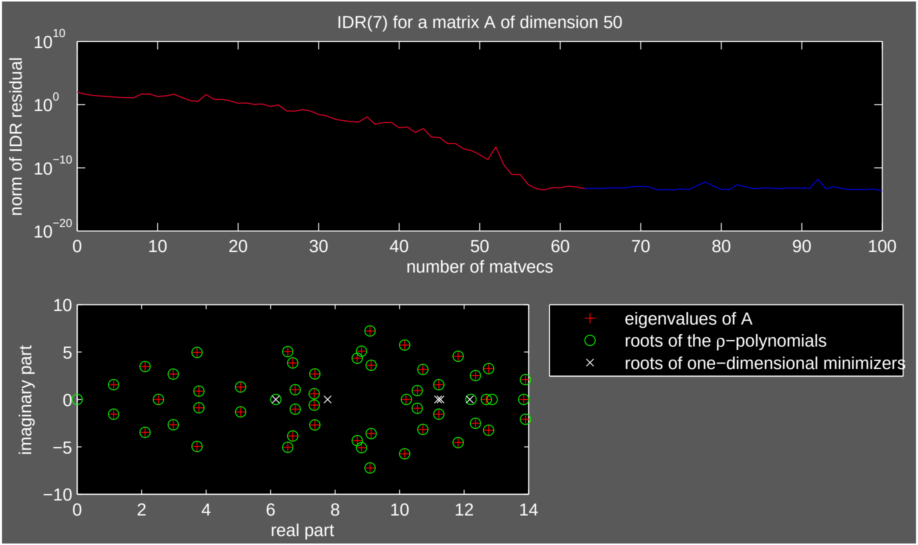
# One run of IDR(7)



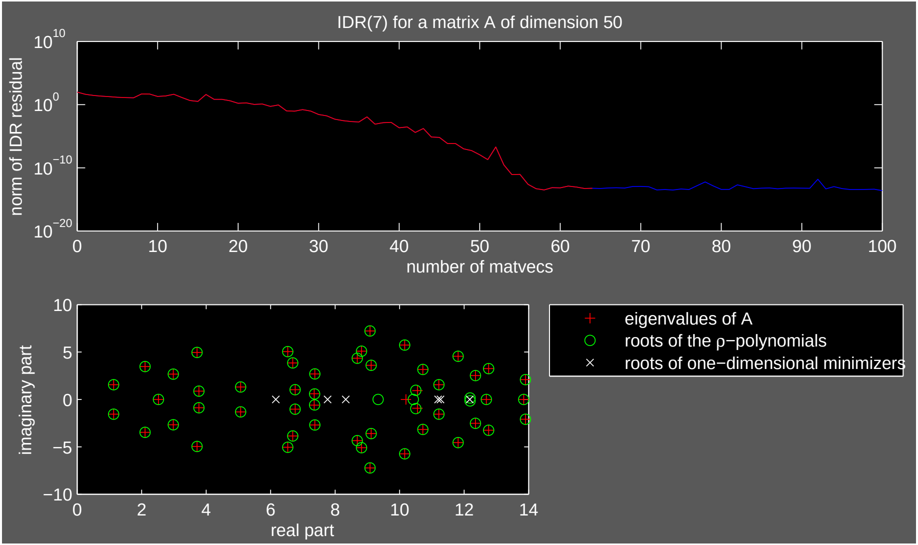
# One run of IDR(7)



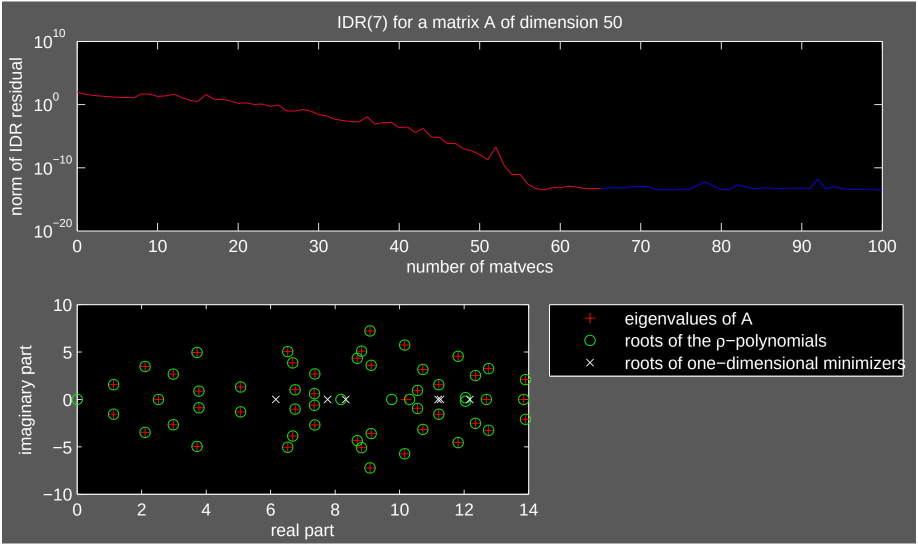
# One run of IDR(7)



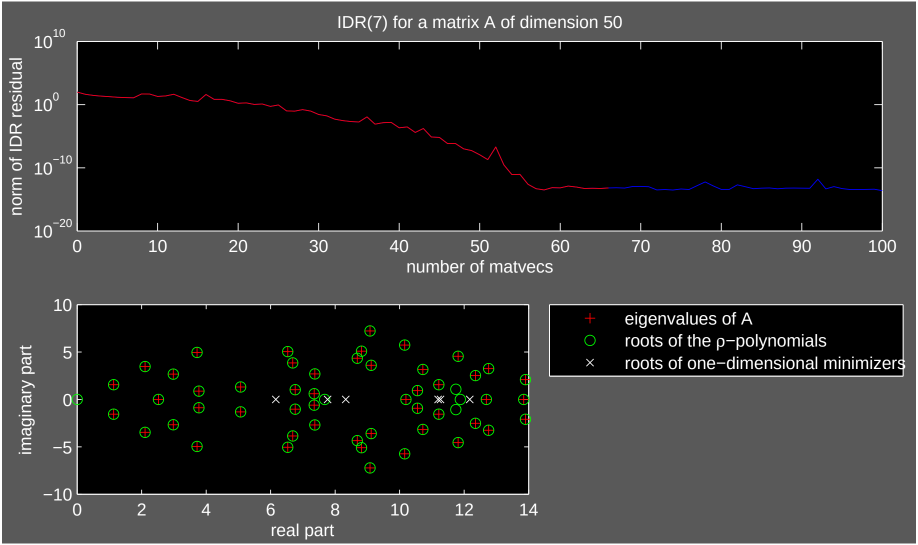
# One run of IDR(7)



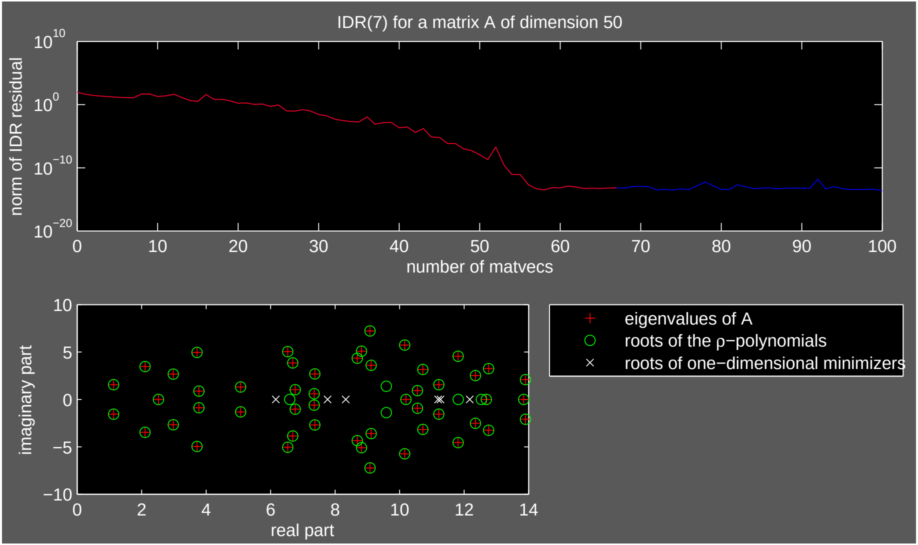
# One run of IDR(7)



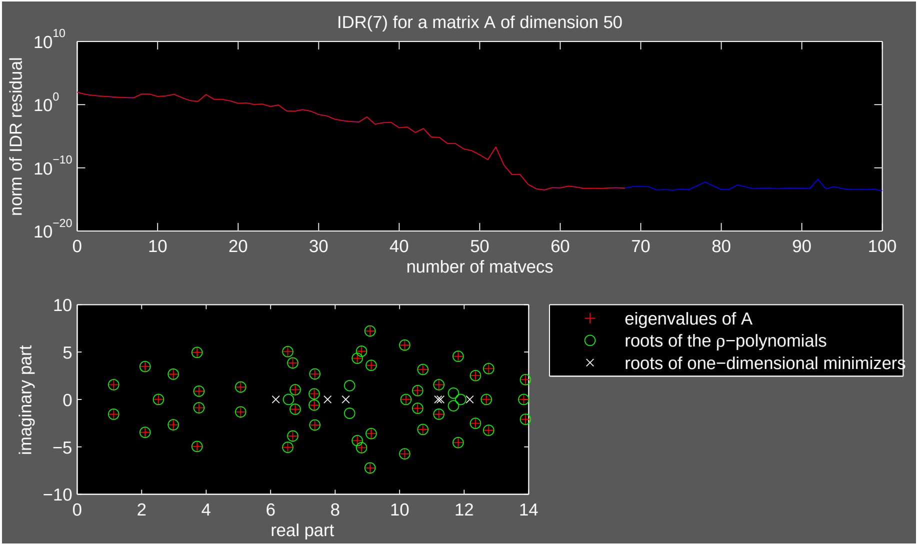
# One run of IDR(7)



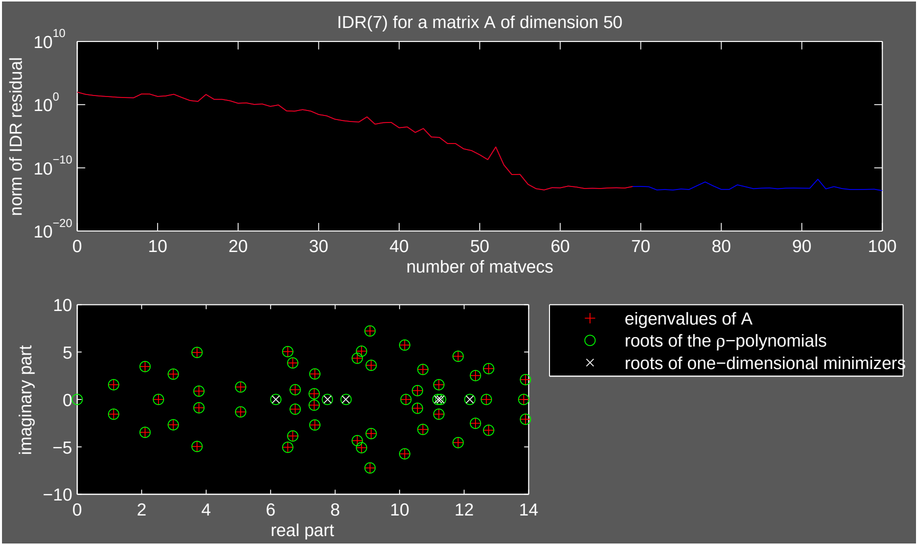
# One run of IDR(7)



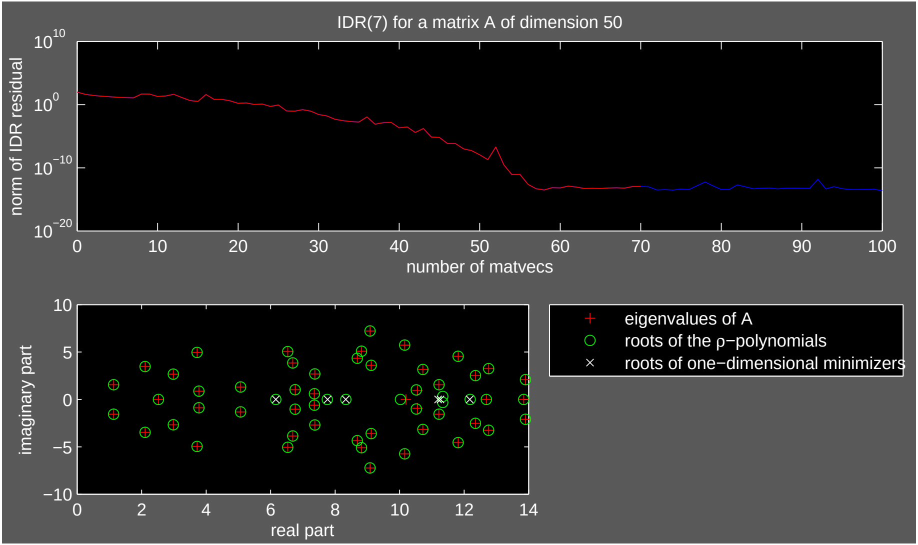
# One run of IDR(7)



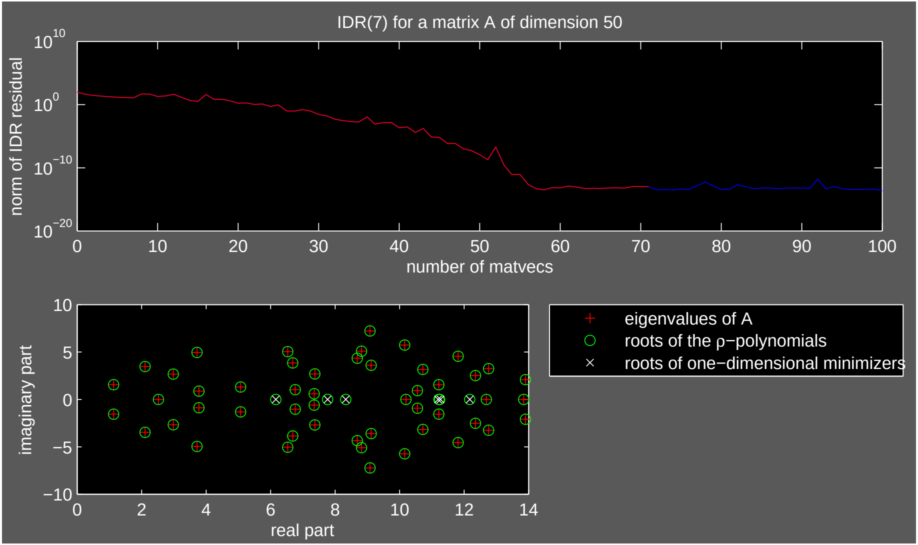
# One run of IDR(7)



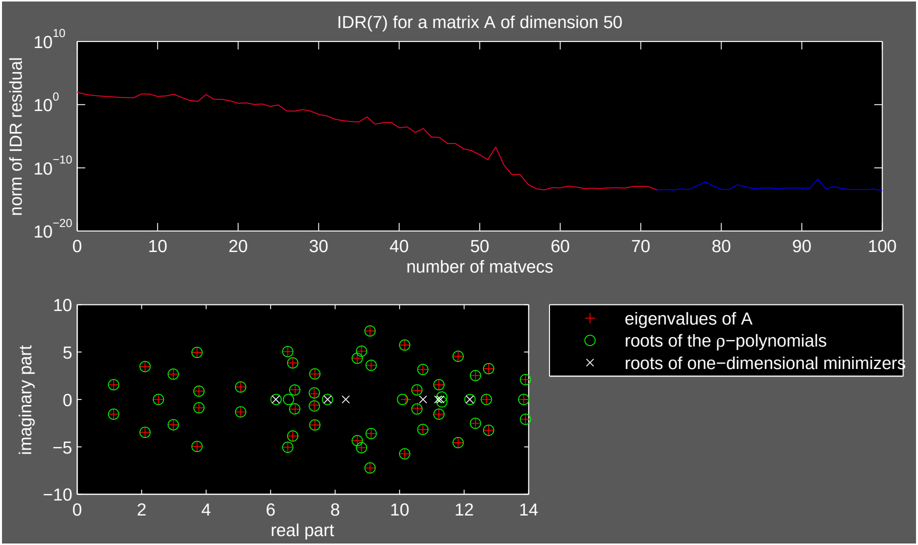
# One run of IDR(7)



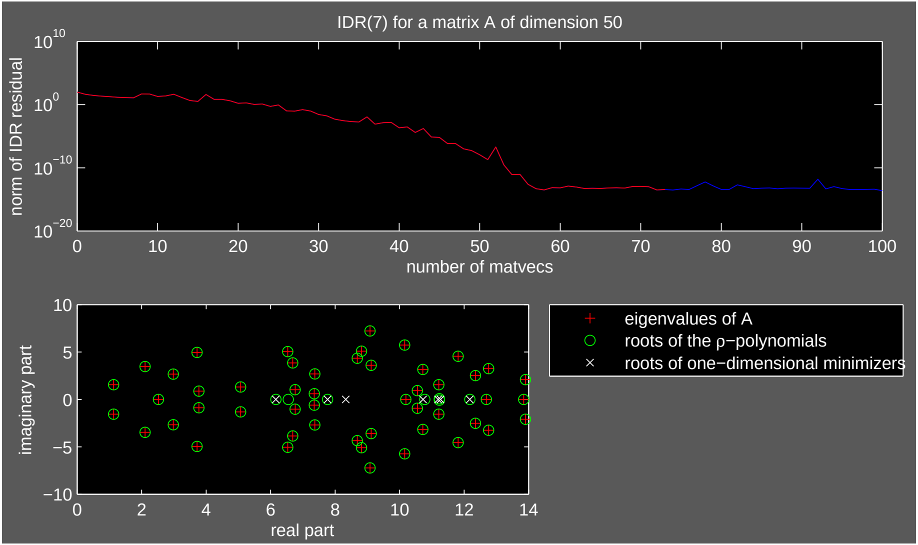
# One run of IDR(7)



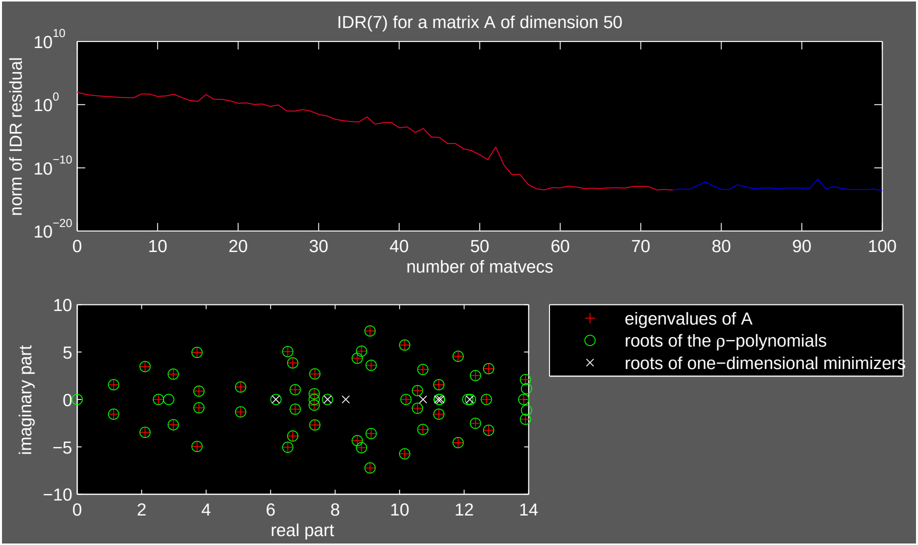
# One run of IDR(7)



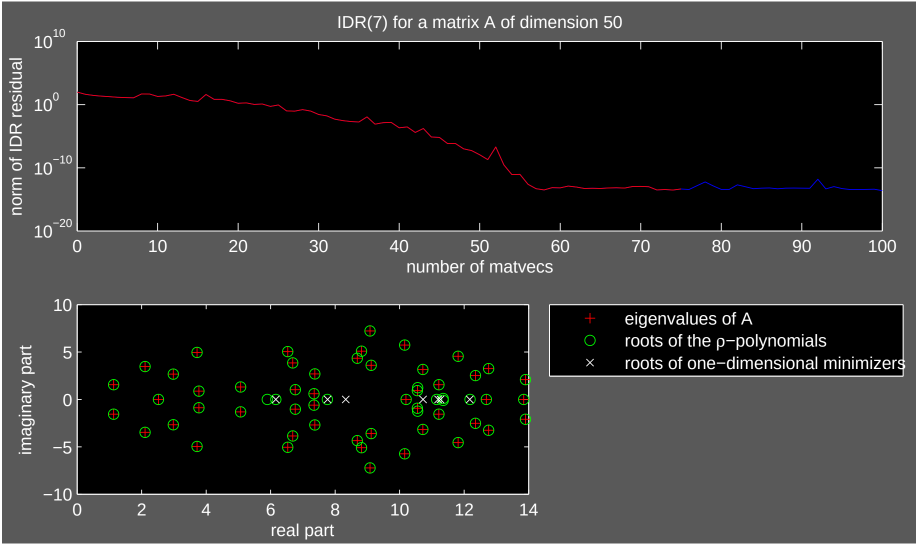
# One run of IDR(7)



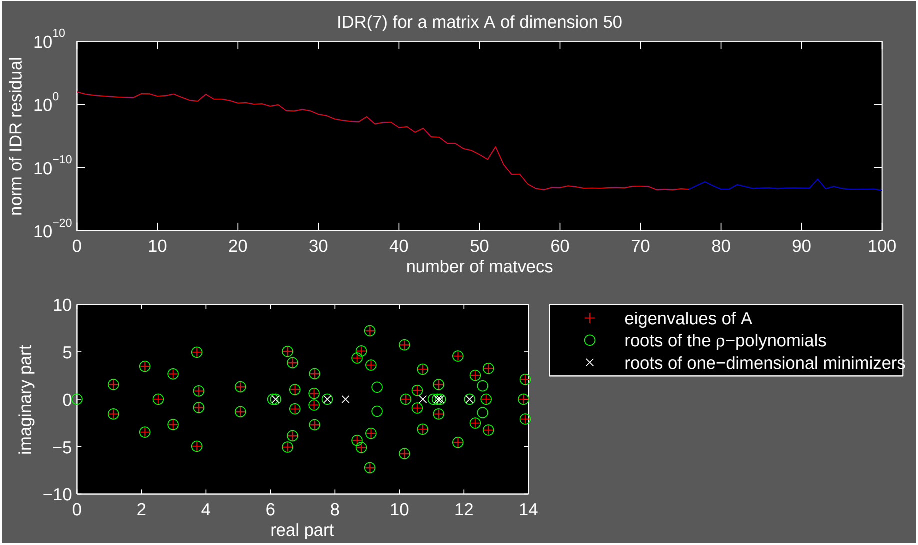
# One run of IDR(7)



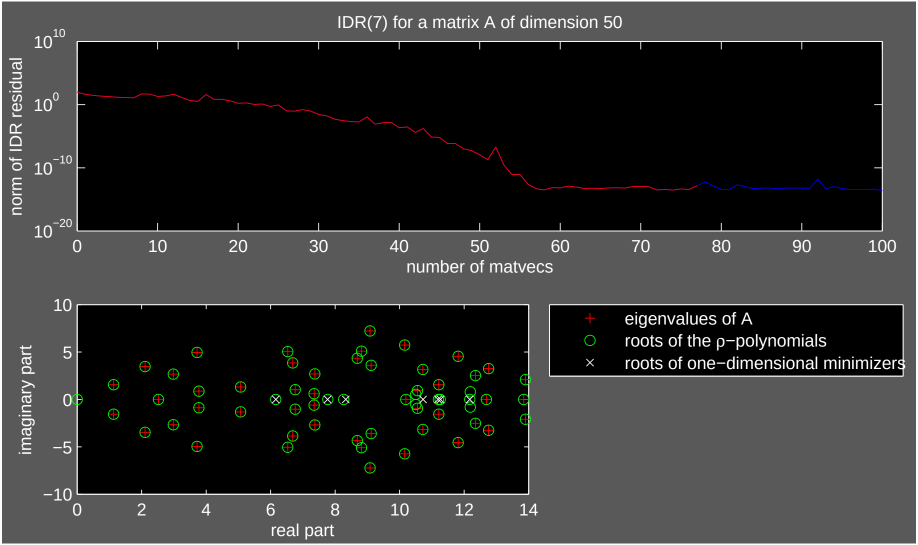
# One run of IDR(7)



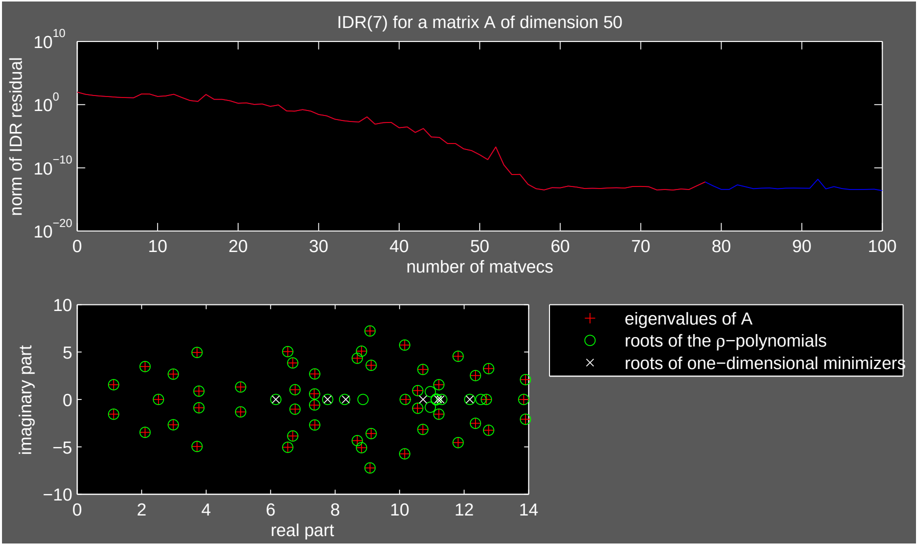
# One run of IDR(7)



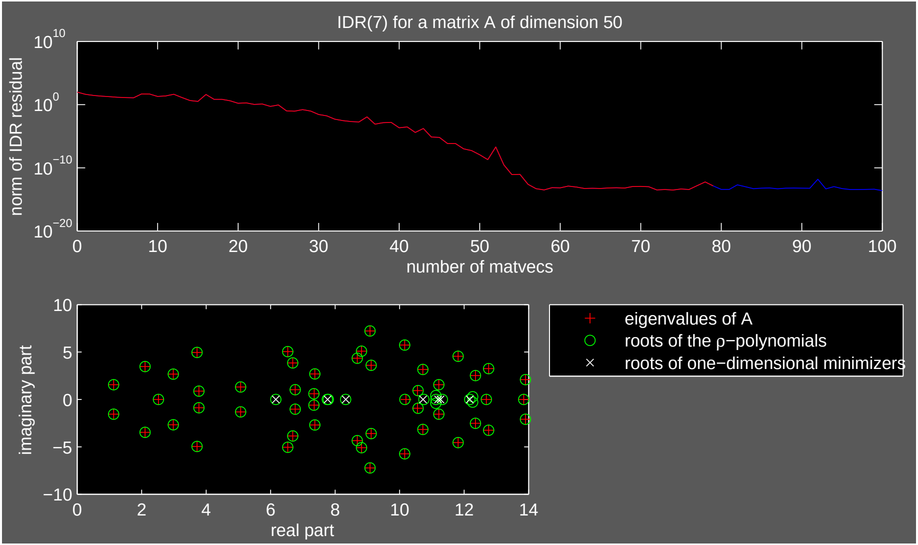
# One run of IDR(7)



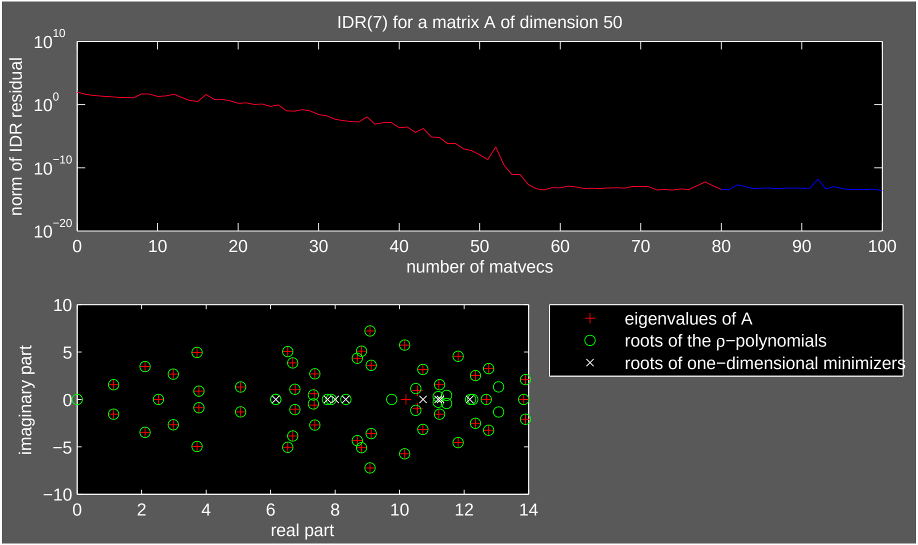
# One run of IDR(7)



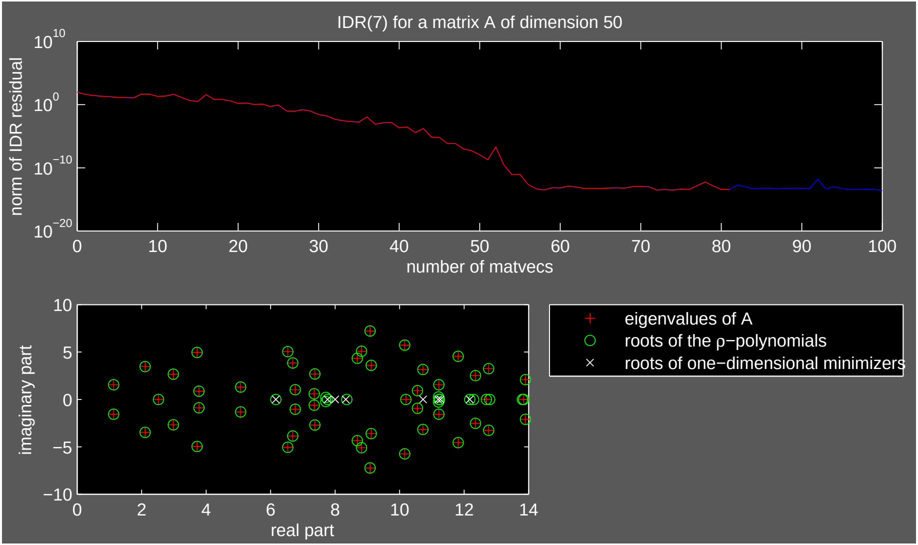
# One run of IDR(7)



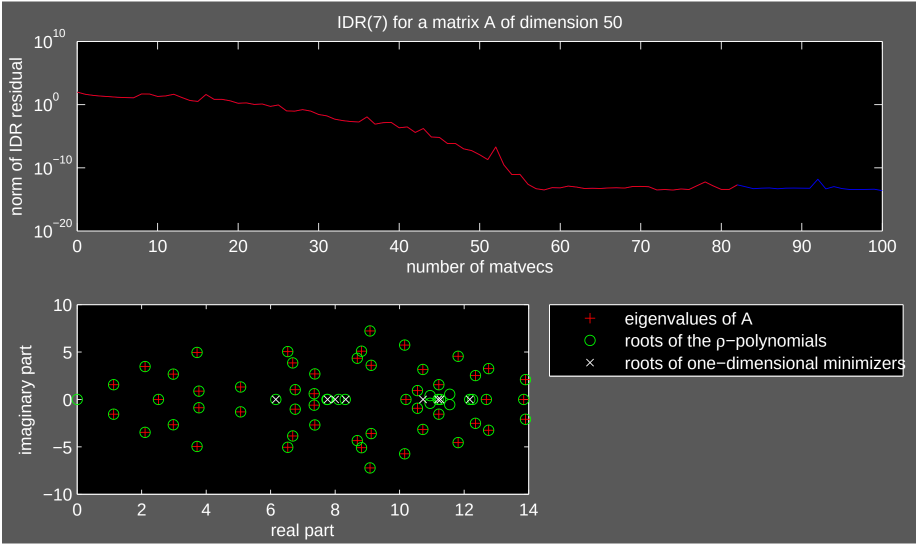
# One run of IDR(7)



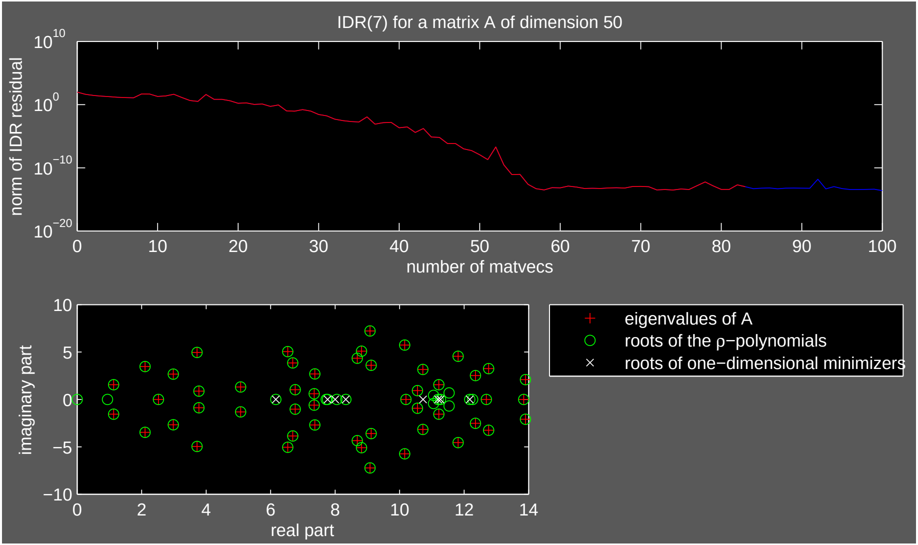
# One run of IDR(7)



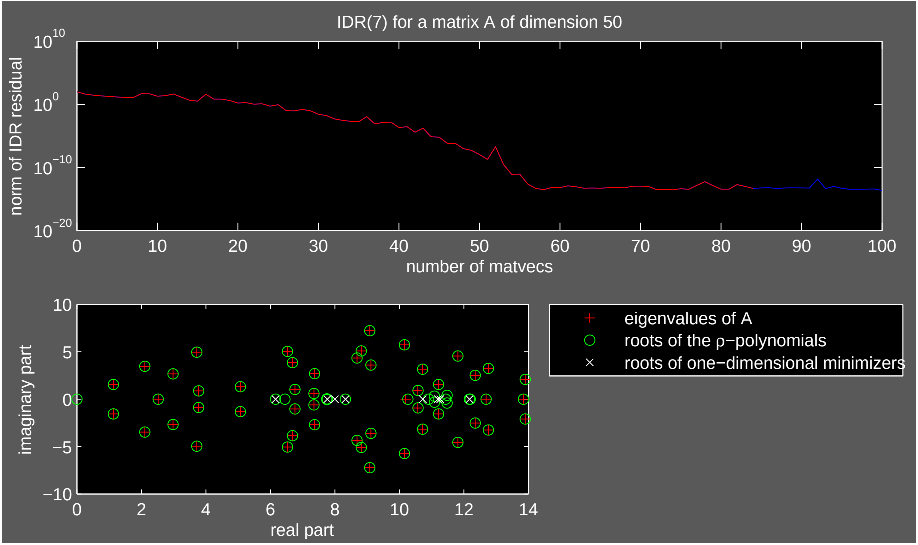
# One run of IDR(7)



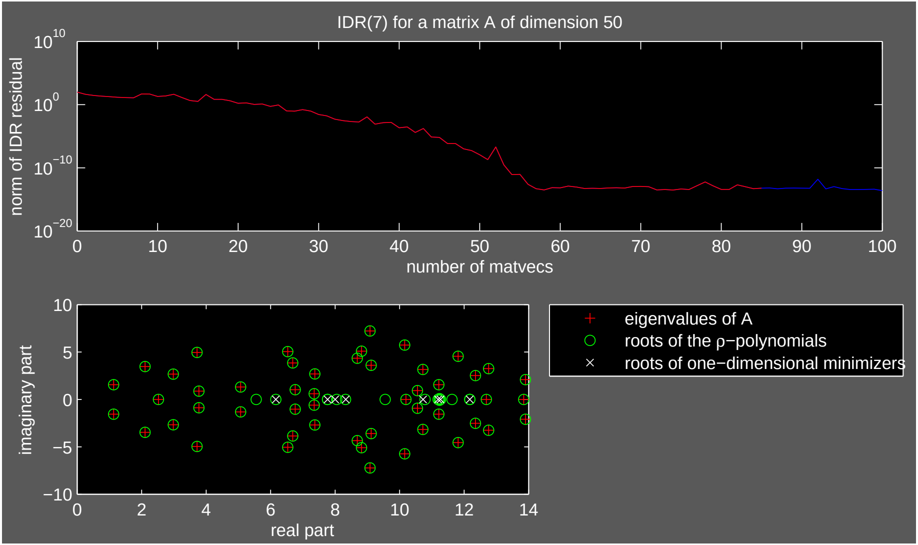
# One run of IDR(7)



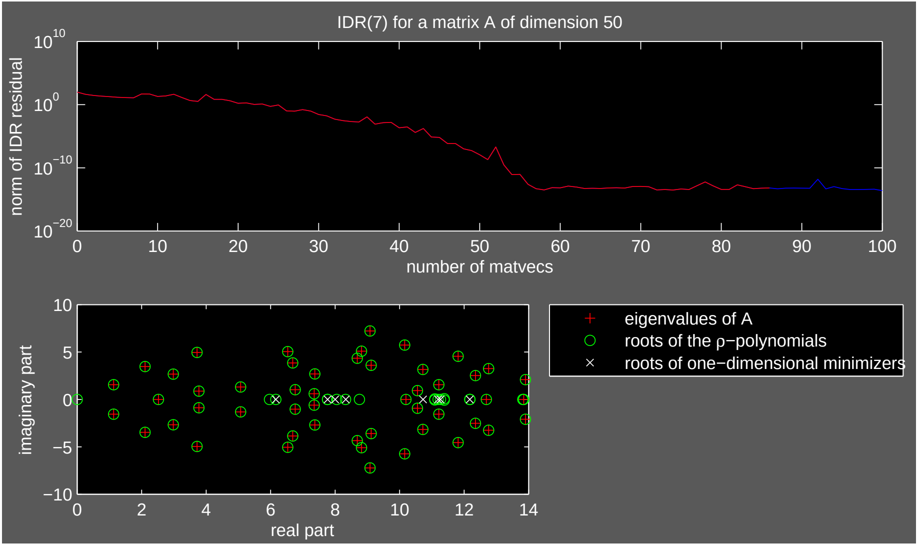
# One run of IDR(7)



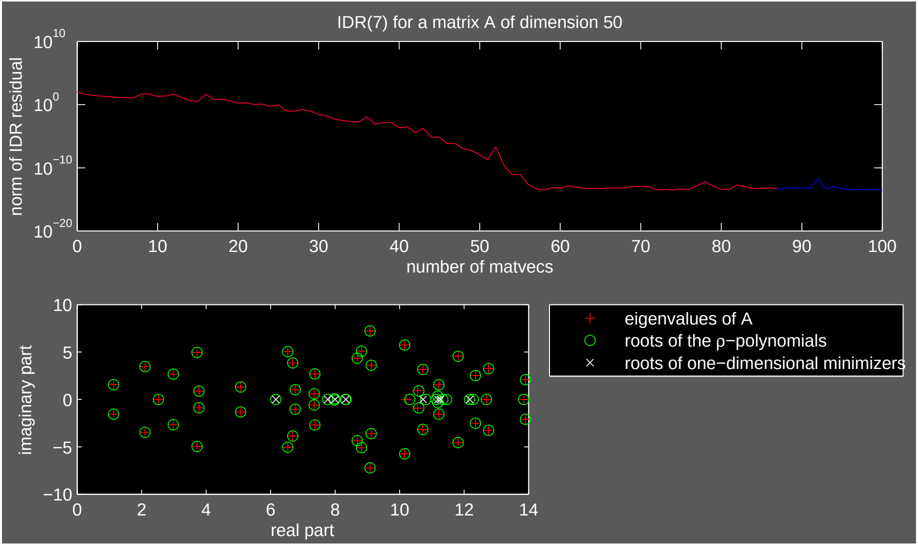
# One run of IDR(7)



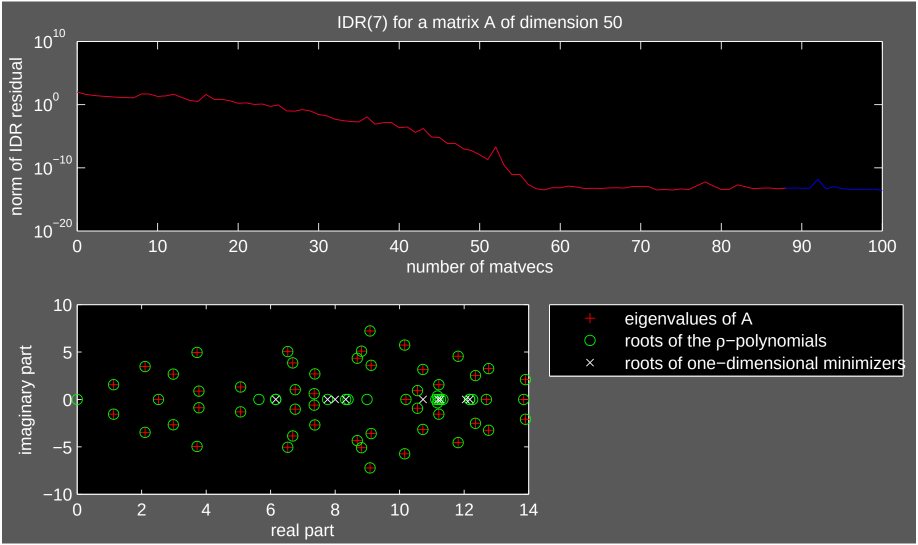
# One run of IDR(7)



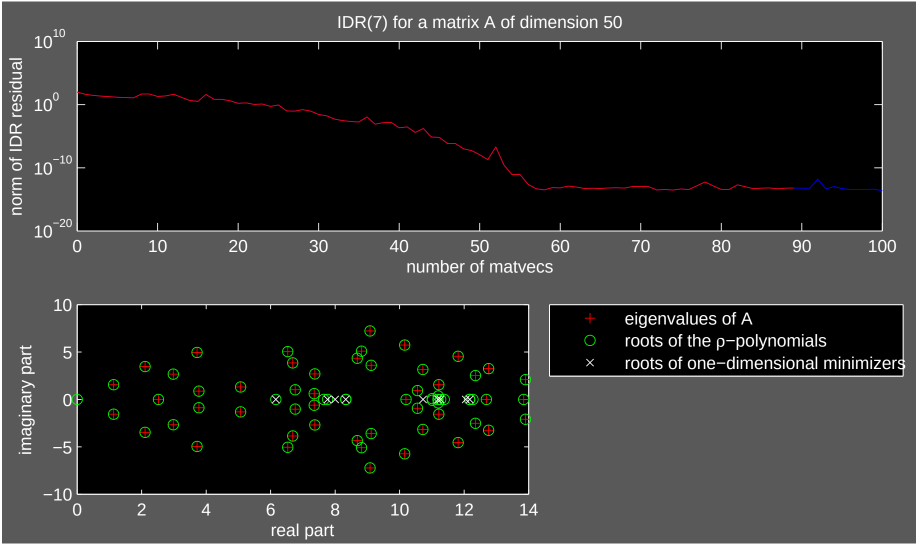
# One run of IDR(7)



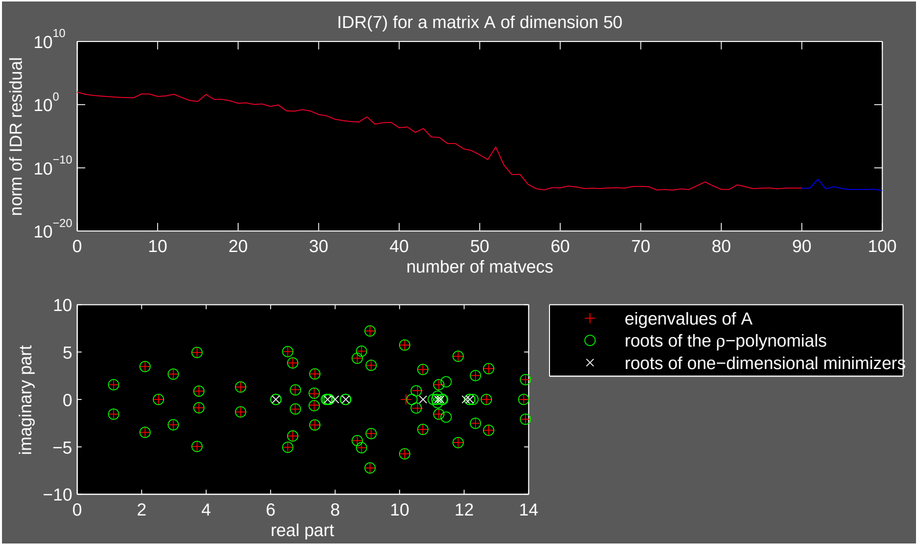
# One run of IDR(7)



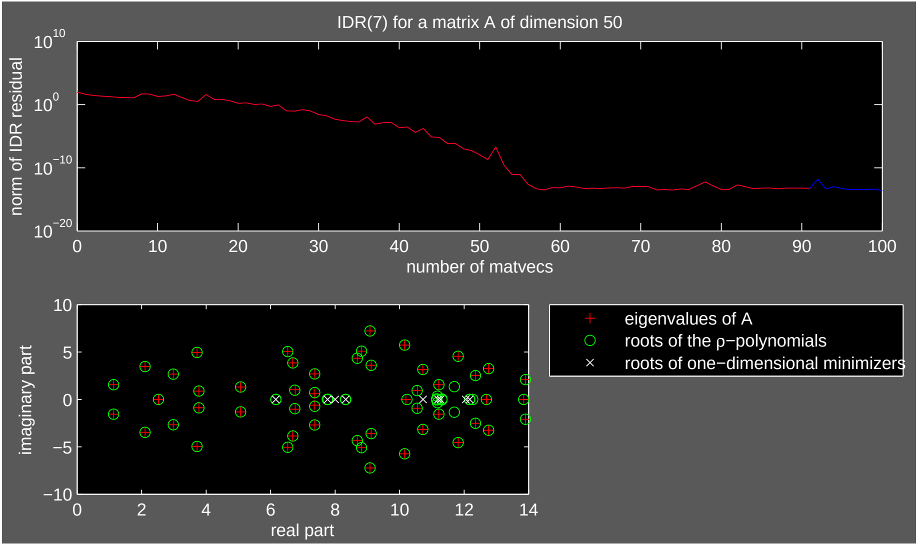
# One run of IDR(7)



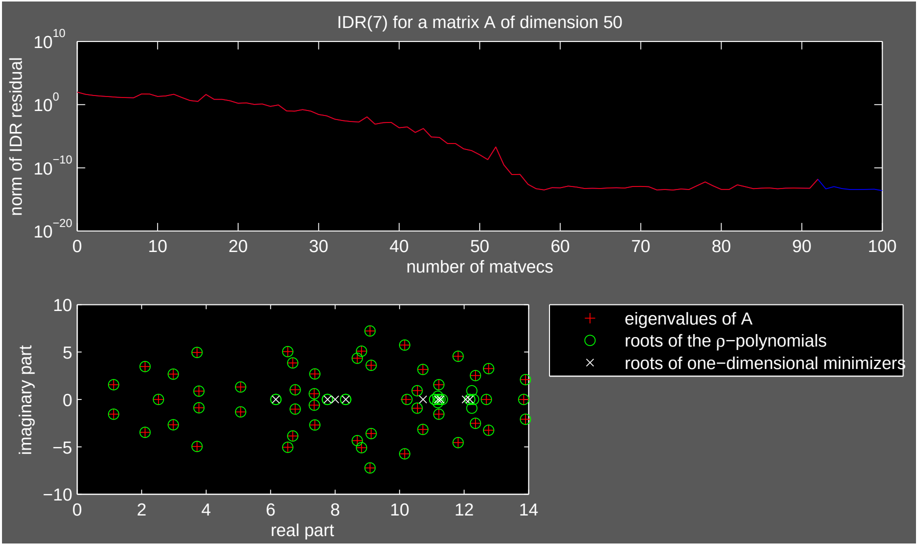
# One run of IDR(7)



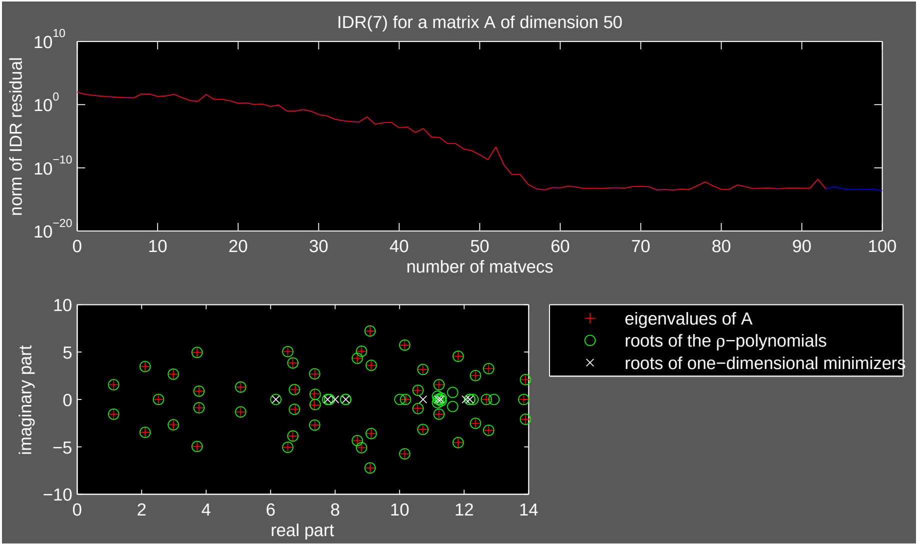
# One run of IDR(7)



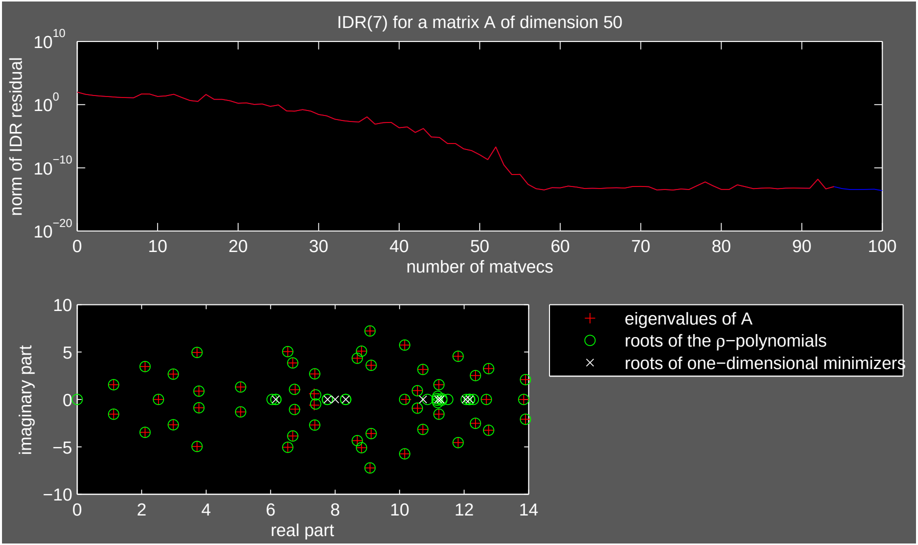
# One run of IDR(7)



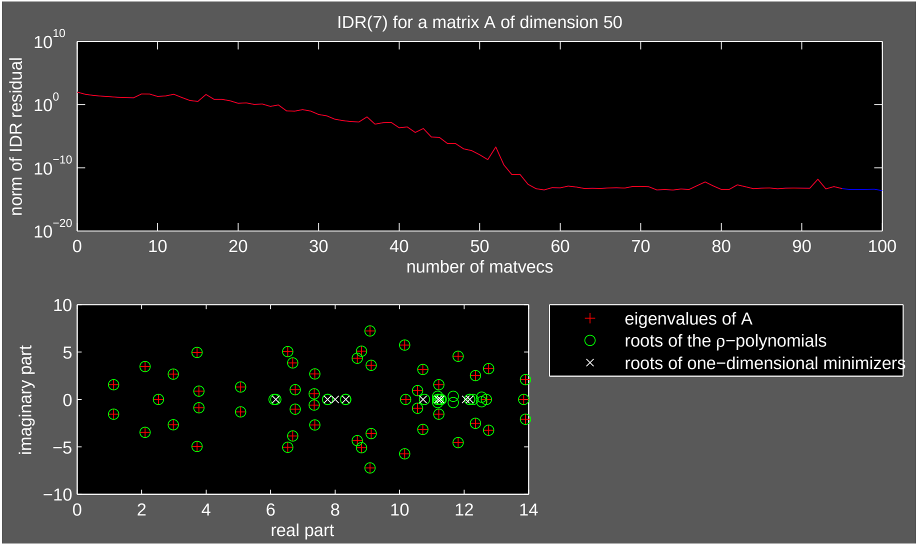
# One run of IDR(7)



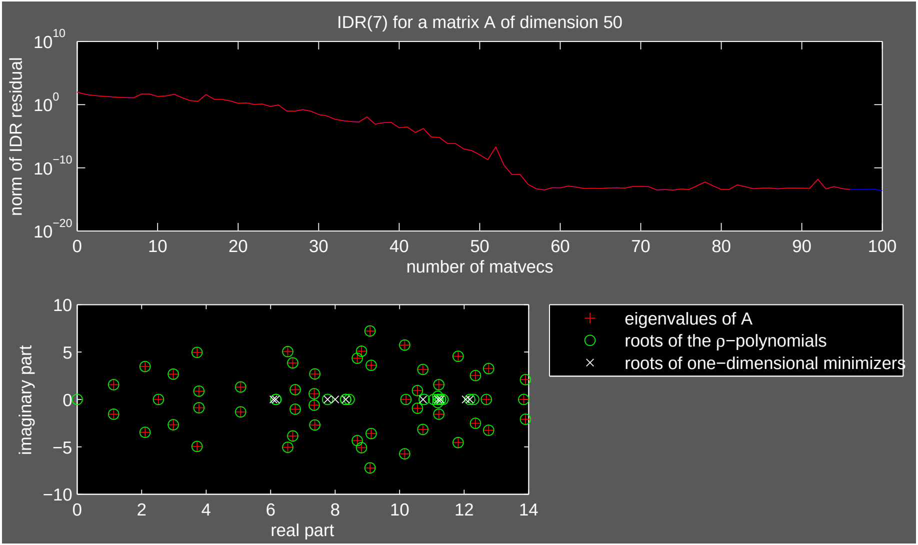
# One run of IDR(7)



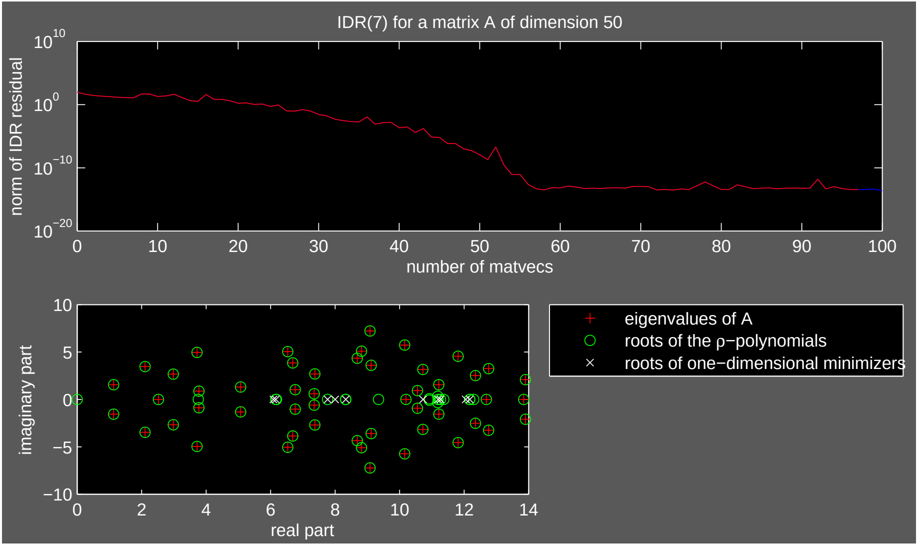
# One run of IDR(7)



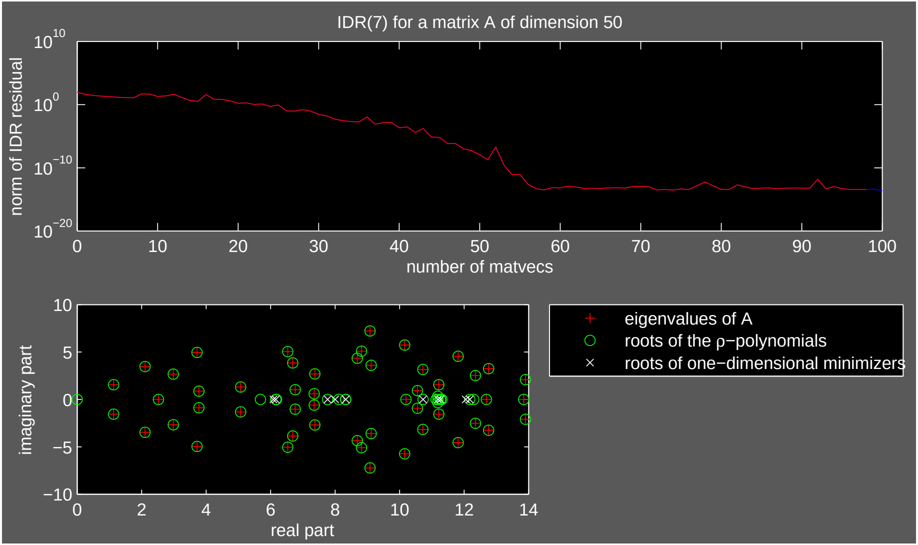
# One run of IDR(7)



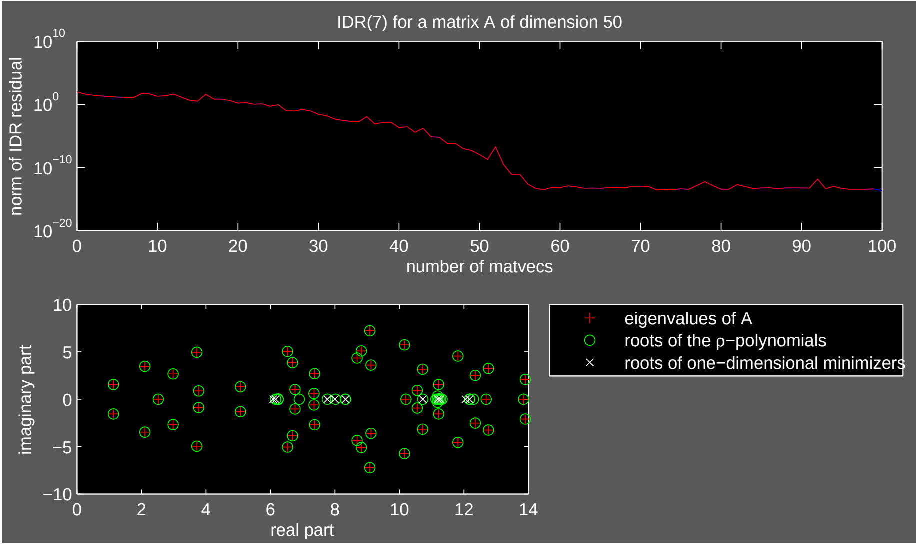
# One run of IDR(7)



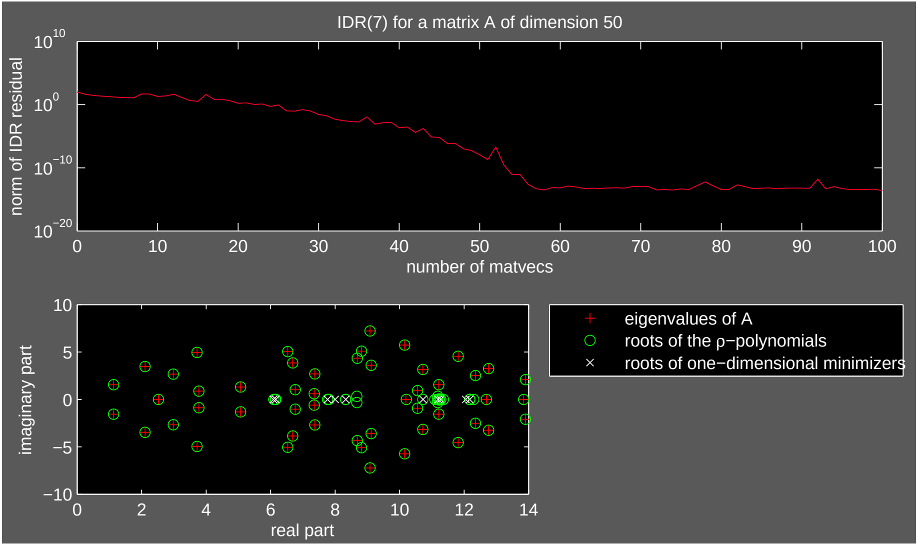
# One run of IDR(7)



# One run of IDR(7)



# One run of IDR(7)



- We sketched our approach of error analysis of IDR.

- We sketched our approach of error analysis of IDR.
- We gave various ways of computing eigenvalues using IDR.

- We sketched our approach of error analysis of IDR.
- We gave various ways of computing eigenvalues using IDR.
- We showed the relation between IDR and the ORTHORES-form of a two-sided Lanczos' process.

- We sketched our approach of error analysis of IDR.
- We gave various ways of computing eigenvalues using IDR.
- We showed the relation between IDR and the ORTHORES-form of a two-sided Lanczos' process.
- We presented a numerical example depicting finite precision aspects.

- We sketched our approach of error analysis of IDR.
- We gave various ways of computing eigenvalues using IDR.
- We showed the relation between IDR and the ORTHORES-form of a two-sided Lanczos' process.
- We presented a numerical example depicting finite precision aspects.

Many questions, some of them partially attacked, remain:

- How do we compute Ritz vectors? How accurate are the Ritz pairs?
- How are the residual and purified residual decomposition related matrix-wise?
- Are all eigenvalues approximated just once?
- Why does the finite precision Lanczos' process re-compute the minimizers and compute spurious eigenvalues close to zero?
- How does the condition grow when the roots $1/\omega_j$ become (almost) multiple (mostly $s$ fold)?
- How does this affect the convergence rate of finite precision IDR?

Thank you for your attention.