

## RKTOOLBOX

The Rational Krylov Toolbox (RKToolbox) is a collection of scientific computing tools based on rational Krylov techniques and it is freely available at <http://rktoolbox.org>. The toolbox provides many methods including:

1. An implementation of Ruhe's (block) rational Krylov sequence method allowing the user to control various options, including non-standard inner product, exploitation of complex-conjugate shifts, orthogonalisation, re-running and parallelism [4].
2. Algorithms for implicit and explicit relocation of the poles of a rational Krylov space [3].
3. An implementation of RKFIT algorithm for rational least approximation [5].
4. The RKFUN class [5], allowing for numerical computations with rational functions.

## THE RATIONAL KRYLOV ALGORITHM

Consider a square matrix  $A \in \mathbb{C}^{N \times N}$ , a starting vector  $b \in \mathbb{C}^N$ , an integer  $m > 0$ , and a nonzero polynomial  $q_m \in \mathcal{P}_m$  with roots disjoint from the spectrum of  $A$ , we define the associated *rational Krylov space of order  $m + 1$*  as

$$\mathcal{Q}_{m+1} = \mathcal{Q}_{m+1}(A, b, q_m) := q_m(A)^{-1} \mathcal{K}_{m+1}(A, b),$$

where  $\mathcal{K}_{m+1} = \mathcal{K}_{m+1}(A, b) := \text{span}\{b, Ab, \dots, A^m b\}$  is a *polynomial Krylov space*. There exists an integer  $M \leq N$ , called the *invariance index* for  $(A, b)$ , such that  $\mathcal{K}_1 \subset \mathcal{K}_2 \subset \dots \subset \mathcal{K}_{M-1} \subset \mathcal{K}_M = \mathcal{K}_{M+1}$ .

Now the rational Krylov method by Ruhe [7] computes an orthonormal basis  $V_{m+1}$  of  $\mathcal{Q}_{m+1}$  which satisfies the rational Arnoldi decomposition

$$AV_{m+1}K_m = V_{m+1}H_m.$$

There are many applications for rational Arnoldi decompositions such as approximating some of  $A$ 's eigenvalues, matrix function approximation, rational quadrature, model order reduction, nonlinear eigenproblems, and rational least squares fitting.

## POLE RELOCATION

There is a direct link between the chosen starting vector  $b$  and the poles  $\xi_j$  of our rational Krylov space  $\mathcal{Q}_{m+1}$ . A change in poles  $\xi_j$  to  $\hat{\xi}_j$  can be interpreted as a change in starting vector from  $b$  to  $\hat{b}$ . Algorithms for moving the poles are provided in the toolbox under the function names `move_poles_exp1` and `move_poles_impl`.

## RKFIT

RKFIT is an iterative Krylov-based algorithm for nonlinear rational approximation. Let  $\{F^{[j]}\}_{j=1}^\ell$  be a family of  $N \times N$  matrices, let  $B$  be an  $N \times n$  block of vectors and  $A$  be an  $N \times N$  matrix, the algorithm finds a family of rational functions  $\{r^{[j]}\}_{j=1}^\ell$  of type  $(m+k, m)$ , all sharing a common denominator  $q_m$ , such that the *relative misfit*

$$\text{misfit} = \sqrt{\frac{\sum_{j=1}^\ell \|F^{[j]}B - r^{[j]}(A)B\|_F^2}{\sum_{j=1}^\ell \|F^{[j]}B\|}} \rightarrow \min$$

is minimal. The algorithm takes an initial guess for  $q_m$  and iteratively tries to improve it by relocating the poles.

This implementation also includes an automated degree reduction and non-diagonal rational approximants are supported as well. The output of RKFIT is an RKFUN object.

## RKFUN

The `rkfun` class is a data type to represent and work with rational functions. A rational function

$$r = p_{m+k}/q_m$$

is of type  $(m+k, m)$ , and such a function can be represented by the matrix pencil  $(H_d, K_d)$ , satisfying  $AV_dK_d = V_{d+1}H_d$  with  $d := \max(m, m+k)$ . With the coefficients  $c = V_{d+1}Fb/\|b\|_2$ , we can now perform calculations with the RKFUN representation

$$r \equiv (H_d, K_d, c).$$

The RKFUN gallery contains many predefined rational functions, or one create them from strings using MATLAB's symbolic interpreter. There are over 20 methods implemented for `rkfun` data type. Most of these methods support MATLAB's Variable Precision Arithmetic (VPA) or the Advanpix Multiple Precision toolbox (MP).

## BLOCK RATIONAL ARNOLDI

Consider a square matrix  $A \in \mathbb{C}^{N \times N}$ , a linearly independent set  $\{b_1, \dots, b_s\}$  of starting vectors which form the columns of a block vector  $\mathbf{b} \in \mathbb{C}^{N \times s}$  (of full rank), and an integer  $m$  such that  $ms < N$ , then the *block Krylov space of order  $m+1$*  is defined as

$$\mathcal{B}_{m+1}^\square(A, \mathbf{b}, q_m) := q_m(A)^{-1} \text{blockspan}\{\mathbf{b}, A\mathbf{b}, \dots, A^m \mathbf{b}\}.$$

The notation used here is not the span of the block vectors, it is the *block span*. This can be simply written as

$$\mathcal{B}_{m+1}^\square(A, \mathbf{b}, q_m) = q_m(A)^{-1} \left\{ \sum_{k=0}^m A^k \mathbf{b} C_k : C_k \in \mathbb{C}^{s \times s} \right\}.$$

We could also consider  $\text{span}\{\mathbf{b}, \dots, A^m \mathbf{b}\}$  which leads to global Arnoldi methods. The block rational Arnoldi algorithm computes an orthonormal basis  $V_{m+1}$ , which satisfies the block rational Arnoldi decomposition (BRAD)

$$AV_{m+1}K_m = V_{m+1}H_m.$$

### Block rational Arnoldi algorithm without deflation

**input** :  $A \in \mathbb{C}^{N \times N}$ ,  $\mathbf{b} \in \mathbb{C}^{N \times s}$  of full rank, poles  $\{\xi\}_{j=1}^m \subset \mathbb{C} \setminus \Lambda(A)$ .

**output**: Decomposition  $AV_{m+1}K_m = V_{m+1}H_m$ .

- 1 For every  $j = 1, \dots, m$ , take any  $(\nu_j, \mu_j, \rho_j, \eta_j) \in \mathbb{C}^4$  such that  $\mu_j/\nu_j = \xi_j$  and  $\rho_j \mu_j \neq \eta_j \nu_j$ .
- 2 Compute the QR decomposition  $\mathbf{v}_1 R := \mathbf{b}$ .
- 3 **for**  $j = 1, \dots, m$  **do**
- 4 **Choose a continuation block vector**  $\mathbf{t}_j \in \mathbb{C}^{s \times s}$ .
- 5 Compute  $\mathbf{w} := (\nu_j A - \mu_j I)^{-1} (\rho_j A - \eta_j I) \mathbf{v}_j \mathbf{t}_j$ .
- 6 Project  $\mathbf{c}_j := \mathbf{V}_j^* \mathbf{w}$ .
- 7 Compute  $\mathbf{w} := \mathbf{w} - \mathbf{V}_j \mathbf{c}_j$  orthogonal to  $\mathbf{v}_1, \dots, \mathbf{v}_j$ .
- 8 **Compute the QR decomposition**  $\mathbf{v}_{j+1} \mathbf{C}_{j,j+1} := \mathbf{w}$ .
- 9 **Set**  $\mathbf{k}_j := \nu_j \mathbf{c}_j - \rho_j \mathbf{t}_j$  and  $\mathbf{h}_j := \mu_j \mathbf{c}_j - \eta_j \mathbf{t}_j$ , where  $\mathbf{t}_j = [\mathbf{t}_j^T \mathbf{O}]^T$  and  $\mathbf{c}_j = [\mathbf{c}_j^T, \mathbf{C}_{j,j+1}^T]^T$ .

Some applications of block rational Arnoldi methods are solving linear systems with multiple right-hand sides, model order reduction [1], multiport RLC networks and continuous-time algebraic Ricatti equations.

## DEFLATION

Under certain circumstances, possible linear dependence may occur, the removal of these columns before adding to the space is known as deflation. Many papers either neglect this or make a full rank assumption. Care has to be taken with choosing the **block continuation vector** to avoid premature breakdown of the algorithm. Exact deflation is rare in finite arithmetic so a tolerance must be used. Failure to detect deflation results in the loss of orthogonality or the introduction of vectors which do not span the desired space. Deflation can be detected by either the **rank revealing QR** or Singular Value Decomposition (SVD).

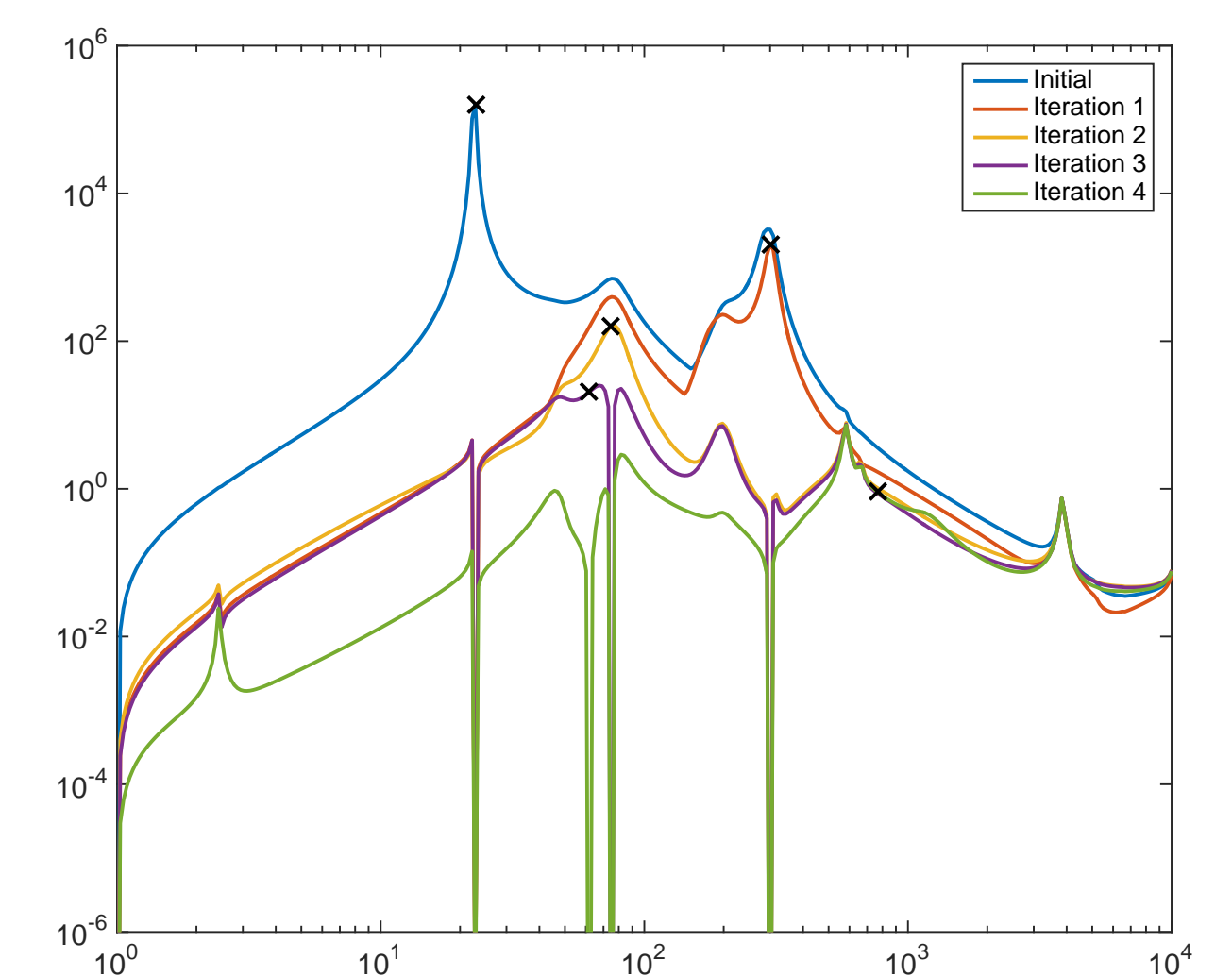
Deflation leads to an **alternative decomposition** where the elements of the block matrices are no longer square. Depending on the choice of decomposition, this leads to either an additional error term or changing column sizes.

## EXAMPLE

Consider a linear time-invariant (LTI) multi-input and multi-output (MIMO) system described by the state-space equations

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t), \end{cases}$$

such that  $A \in \mathbb{R}^{n \times n}$  is large and sparse,  $B, C^T \in \mathbb{R}^{n \times p}$  are tall and skinny,  $x(t) \in \mathbb{R}^n$  denotes the state vector and  $u(t), y(t) \in \mathbb{R}^p$  are the input and output vectors respectively. We can use block rational Arnoldi method for model order reduction [1]. Applying the block rational Arnoldi algorithm to the CD-player matrix using the adaptive shifts defined in [1] proves to be quite effective, as shown in the following figure.



The black crosses indicate the location of the pole for the next iteration chosen using an adaptive selection technique. After an iteration of the block rational Arnoldi algorithm, the error is reduced around that pole. The adaptive selection places a pole around the point of largest error.

## REFERENCES

- [1] O. ABIDI, M. HACHED, K. JBILOU, *Adaptive rational block Arnoldi methods for model reductions in large-scale MIMO dynamical systems* New Trends Math. Sci., 4(2): 227–239, 2016.
- [2] M. BERLJAJA, S. GÜTTEL, *A Rational Krylov Toolbox for MATLAB*, MIMS EPrint 2014.56, Manchester Institute for Mathematical Sciences, The University of Manchester, Manchester, UK, 2014. Available for download at <http://guettel.com/rktoolbox/>.
- [3] M. BERLJAJA, S. GÜTTEL, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., 36(2):894–916, 2015.
- [4] M. BERLJAJA, S. GÜTTEL, *Parallelization of the rational Arnoldi algorithm*, MIMS EPrint 2016.32 (<http://eprints.ma.man.ac.uk/2503/>), Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2016. Accepted for publication in SIAM J. Sci. Comput., 2017.
- [5] M. BERLJAJA, S. GÜTTEL, *The RKFIT algorithm for nonlinear rational approximation*, MIMS EPrint 2015.38 (<http://eprints.ma.man.ac.uk/2530/>), Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2015.
- [6] S. ELSWORTH, S. GÜTTEL, *Block rational Arnoldi*, In preparation.
- [7] A. RUHE, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19(5):1535–1551, 1998.