A Randomized Numerical Method for Joint Eigenvalues of Commuting Matrices

Haoze He, Daniel Kressner, Bor Plestenjak

Abstract

Let $\mathcal{A} = \{A_1, \ldots, A_d\}$ be a *commuting family* of $n \times n$ complex matrices, i.e., $A_j A_k = A_k A_j$ for all $1 \le j, k \le d$. Then there exists a unitary matrix U such that all matrices $U^* A_1 U, \ldots, U^* A_d U$ are upper triangular and the n d-tuples containing the diagonal elements of $U^* A_1 U, \ldots, U^* A_d U$ are called the *joint eigenvalues* of $\mathcal{A}$. For every joint eigenvalue $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$ of $\mathcal{A}$ there exists a nonzero *common eigenvector* x, such that $A_i x = \lambda_i x$ for $i = 1, \ldots, d$.

The task of numerical computation of joint eigenvalues for a commuting family arises, e.g., in solvers for multiparameter eigenvalue problems and systems of multivariate polynomials. We propose and analyze a simple approach, summarized in Algorithm 1, that computes eigenvalues as one-sided or two-sided Rayleigh quotients from eigenvectors of a random linear combination

$$A(\mu) = \mu_1 A_1 + \mu_2 A_2 + \cdots + \mu_d A_d, \qquad (1)$$

where $\mu = [\mu_1 \ \cdots \ \mu_d]^T$ is a random vector from the uniform distribution on the unit sphere in $\mathbb{C}^d$.

We show that Algorithm 1, in particular the use of two-sided Rayleigh quotients, accurately computes well-conditioned semisimple joint eigenvalues with high probability. It still works satisfactorily in the presence of defective eigenvalues. Experiments show that the method can be efficiently used in solvers for multiparameter eigenvalue problems and roots of systems of multivariate polynomials.

Algorithm 1 Randomized **J**oint **E**igenvalue **A**pproximation

Input: A nearly commuting family $\mathcal{A} = \{A_1, \ldots, A_d\}$, opt $\in \{\text{RQ1}, \text{RQ2}\}$.
Output: Approximations of joint eigenvalues of $\mathcal{A}$.

```
1: Draw μ ∈ C^d from the uniform distribution on the unit sphere.
2: Compute A(μ) = μ_1 A_1 + ··· + μ_d A_d.
3: Compute invertible matrices X, Y such that the columns of X have norm 1, Y*X = I,
   and Y* A(μ) X is diagonal.
4: if opt = RQ1 then
5:     return λ^(i)_RQ1 = (x_i* A_1 x_i, ..., x_i* A_d x_i),   i = 1, ..., n.
6: else if opt = RQ2 then
7:     return λ^(i)_RQ2 = (y_i* A_1 x_i, ..., y_i* A_d x_i),   i = 1, ..., n.
8: end if
```

The idea of using a random linear combination like (1) is not new. For example, in [1, 4] the unitary matrix U from the Schur decomposition $A(\mu) = U^* R U$ is used to transform all matrices from $\mathcal{A}$ to *block* upper triangular form. Using the Schur decomposition, however, requires clustering to group multiple eigenvalues together, and this is a numerically subtle task. On the other hand, Algorithm 1 does not require clustering and in practice often leads to equally good or even better results for, e.g., multiparameter eigenvalue problems [5] and multivariate root finding problems.

For a significantly simpler situation of commuting *Hermitian* matrices, where a unitary matrix exists that jointly diagonalizes all matrices, randomized methods based on (1) have recently been analyzed in [2], establishing favorable robustness and stability properties.

An important source of joint eigenvalue problems are eigenvector-based methods for solving systems of multivariate polynomial equations. If we are looking for roots of a set of polynomials

$$p_i(x_1,\ldots,x_d)=0,\quad i=1,\ldots,m, \tag{2}$$

such that the solution consists of finitely many points, then a common feature of these methods is that they construct so called *multiplication matrices* $M_{x_1},\ldots,M_{x_d}$ that commute and their joint eigenvalues are the roots $(x_1,\ldots,x_d)$ of (2). Many techniques that use symbolic and/or numerical computation, including Gröbner basis, various resultants, and Macaulay matrices, are used to construct the multiplication matrices, see, e.g., [6].

Another source are *multiparameter eigenvalue problems*. A d-parameter version has the form

$$A_{i0}x_i=\lambda_1 A_{i1}x_i+\cdots+\lambda_d A_{id}x_i,\quad i=1,\ldots,d, \tag{3}$$

where A_{ij} is an $n_i\times n_i$ complex matrix and $x_i\neq 0$ for $i=1,\ldots,d$. When (3) is satisfied, a d-tuple $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_d)\in\mathbb{C}^d$ is called an *eigenvalue* and $x_1\otimes\cdots\otimes x_d$ is a corresponding eigenvector. Generically, a multiparameter eigenvalue problem (3) has $N=n_1\cdots n_d$ eigenvalues. The problem (3) is related to a system of d generalized eigenvalue problems

$$\Delta_i z=\lambda_i\Delta_0 z,\quad i=1,\ldots,d,$$

with $z=x_1\otimes\cdots\otimes x_d$ and the $N\times N$ matrices (that are called *operator determinants*)

$$\Delta_0=\begin{vmatrix}A_{11} & \cdots & A_{1d}\\ \vdots & & \vdots\\ A_{d1} & \cdots & A_{dd}\end{vmatrix}_\otimes,\quad \Delta_i=\begin{vmatrix}A_{11} & \cdots & A_{1,i-1} & A_{10} & A_{1,i+1} & \cdots & A_{1d}\\ \vdots & & \vdots & \vdots & \vdots & & \vdots\\ A_{d1} & \cdots & A_{d,i-1} & A_{d0} & A_{d,i+1} & \cdots & A_{dd}\end{vmatrix}_\otimes,\quad i=1,\ldots,d.$$

If Δ_0 is invertible, then the matrices $\Gamma_i:=\Delta_0^{-1}\Delta_i$ for $i=1,\ldots,d$ commute. If N is not too large, then a standard approach to solve (3) is to explicitly compute the matrices $\Gamma_1,\ldots,\Gamma_d$ and then solve the joint eigenvalue problem.

References

[1] R.M. Corless, P.M Gianni, B.M. Trager, *A reordered Schur factorization method for zero dimensional polynomial systems with multiple roots*, in Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI) 133–140, ACM, New York, 1997.

[2] H. He, D. Kressner, *Randomized joint diagonalization of symmetric matrices*, SIAM J. Matrix Anal. Appl. 45 (2024) 661–684.

[3] H. He, D. Kressner, B. Plestenjak, *Randomized methods for computing joint eigenvalues, with applications to multiparameter eigenvalue problems and root finding*, arXiv:2409.00500 (2024), to appear in Numer. Algorithms.

[4] D. Manocha, J. Demmel, *Algorithms for intersecting parametric and algebraic curves I: simple intersections*, ACM Trans. Graph. 13 (1994) 73–100.

[5] B. Plestenjak, *MultiParEig. Toolbox for multiparameter and singular eigenvalue problems*, www.mathworks.com/matlabcentral/fileexchange/47844-multipareig, MATLAB Central File Exchange.

[6] H. J. Stetter, *Numerical Polynomial Algebra*, SIAM, Philadelphia, PA, 2004.