Computing Eigenvalue Decomposition of Arrowhead and Diagonal-Plus-Rank-k Matrices of Quaternions

by Thaniporn Chaysri, Nevena Jakovčević Stor and Ivan Slapničar

from the University of Split, FESB

at the 8th Croatian Mathematical Congress, Osijek, July 2-5, 2024.

This work has been fully supported by the Croatian Science Foundation under the project IP-2020-02-2240 - http://manaa.fesb.unist.hr.

HRZZ Croatian Science Foundation

Quaternions

Quaternions are a non-commutative associative number system that extends complex numbers, introduced by Hamilton (1853, 1866). For basic quaternions $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$, the quaternions have the form

$$q = a + b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k}, \quad a, b, c, d, \in \mathbb{R}.$$

The multiplication table of basic quaternions is the following:

$\times$	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
$\mathbf{i}$	-1	$\mathbf{k}$	$-\mathbf{j}$
$\mathbf{j}$	$-\mathbf{k}$	-1	$\mathbf{i}$
$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	-1

Conjugation is given by

$$\bar{q} = a - b\,\mathbf{i} - c\,\mathbf{j} - d\,\mathbf{k}.$$

Then,

$$\bar{q}q = q\bar{q} = |q|^2 = a^2 + b^2 + c^2 + d^2.$$

Let $f(x)$ be a complex analytic function. The value $f(q)$, where $q \in \mathbb{Q}$, is computed by evaluating the extension of f to the quaternions at q, see (Sudbery,1979), for example,

$$\sqrt{q} = \pm \left(\sqrt{\frac{\|q\| + a_1}{2}} + \frac{\operatorname{imag}(q)}{\|\operatorname{imag}(q)\|} \sqrt{\frac{\|q\| - a_1}{2}} \right).$$

Basic operations with quaternions and computation of the functions of quaternions are implemented in the package Quaternions.jl.

Standard form

Quaternions p and q are **similar** if

$$\exists x \quad \mathrm{s.\,t.} \quad p = x^{-1} q x.$$

This is *iff*

$$\mathrm{real}(p) = \mathrm{real}(q) \quad \text{and} \quad \|p\| = \|q\|.$$

The **standard form** of the quaternion q is the (unique) similar quaternion q_s:

$$q_s = x^{-1} q x = a + \hat{b}\,\mathbf{i}, \quad \|x\| = 1, \quad \hat{b} \geq 0,$$

where x is computed as follows:

if $c = d = 0$, then $x = 1$,

if $b < 0$, then $x = -\mathbf{j}$, ortherwise,

if $c^2 + d^2 > 0$, then $x = \hat{x}/\|\hat{x}\|$, where $\hat{x} = \|\,\mathrm{imag}(q)\| + b - d\,\mathbf{j} + c\,\mathbf{k}$.

Homomorphism

Quaternions are homomorphic to $\mathbb{C}^{2\times 2}$:

$$q \to \begin{bmatrix} a + b\,\mathbf{i} & c + d\,\mathbf{i} \\ -c + d\,\mathbf{i} & a - b\,\mathbf{i} \end{bmatrix} \equiv C(q),$$

with eigenvalues q_s and $\bar{q}_s$. It holds

$$C(p+q) = C(p) + C(q), \quad C(pq) = C(p)C(q).$$

Matrices

Arrowhead matrix (**Arrow**) is a matrix of the form

$$A = \begin{bmatrix} D & u \\ v^* & \alpha \end{bmatrix},$$

where

$$\operatorname{diag}(D), u, v \in \mathbb{Q}^{n-1}, \quad \alpha \in \mathbb{Q},$$

or any symmetric permutation of such a matrix.

Diagonal-plus-rank-one matrix (**DPR1**) is a matrix of the form

$$A = \Delta + x\rho y^*$$

where

$$\operatorname{diag}(\Delta), x, y \in \mathbb{Q}^n, \quad \rho \in \mathbb{Q}.$$

Matrix × vector

Products

$$w = Az$$

are computed in $O(n)$ operations.

Let $A = \mathrm{Arrow}(D, u, v, \alpha)$. Then

$$\begin{aligned}
w_j &= d_j z_j + u_j z_i, \quad j = 1, 2, \cdots, i-1 \\
w_i &= v_{1:i-1}^* z_{1:i-1} + \alpha z_i + v_{i:n-1}^* z_{i+1:n} \\
w_j &= u_{j-1} z_i + d_{j-1} z_j, \quad j = i+1, i+2, \cdots, n.
\end{aligned}$$

Let $A = \mathrm{DPRk}(\Delta, x, y, \rho)$ and let $\beta = \rho(y^* z)$. Then

$$w_i = \delta_i z_i + x_i \beta, \quad i = 1, 2, \cdots, n.$$

Inverses (Arrowhead)

Inverses are computed in $O(n)$ operations.

Let $A = \mathrm{Arrow}(D, u, v, \alpha)$ be nonsingular.

Let P be the permutation matrix of the permutation $p = (1, 2, \cdots, i-1, n, i, i+1, \cdots, n-1)$.

If all $d_j \neq 0$, the inverse of A is a DPRk (DPR1) matrix

$$A^{-1} = \Delta + x\rho y^*,$$

where

$$\Delta = P \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^T, \quad x = P \begin{bmatrix} D^{-1}u \\ -1 \end{bmatrix} \rho, \quad y = P \begin{bmatrix} D^{-*}v \\ -1 \end{bmatrix}, \quad \rho = (\alpha - v^* D^{-1} u)^{-1}.$$

Inverses (Arrowhead, cont.)

If $d_j = 0$, the inverse of A is an Arrow with the tip of the arrow at position (j,j) and zero at position A_{ii} (the tip and the zero on the shaft change places). Let $\hat{P}$ be the permutation matrix of the permutation $\hat{p} = (1, 2, \cdots, j-1, n, j, j+1, \cdots, n-1)$. Partition D, u and v as

$$D = \begin{bmatrix} D_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_2 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_j \\ u_2 \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_j \\ v_2 \end{bmatrix}.$$

Then

$$A^{-1} = P \begin{bmatrix} \hat{D} & \hat{u} \\ \hat{v}^* & \hat{\alpha} \end{bmatrix} P^T,$$

where

$$\hat{D} = \begin{bmatrix} D_1^{-1} & 0 & 0 \\ 0 & D_2^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{u} = \begin{bmatrix} -D_1^{-1} u_1 \\ -D_2^{-1} u_2 \\ 1 \end{bmatrix} u_j^{-1}, \quad \hat{v} = \begin{bmatrix} -D_1^{-*} v_1 \\ -D_2^{-*} v_2 \\ 1 \end{bmatrix} v_j^{-1},$$
$$\hat{\alpha} = v_j^{-*} \left(-\alpha + v_1^* D_1^{-1} u_1 + v_2^* D_2^{-1} u_2\right) u_j^{-1}.$$

Inverses (DPRk)

Let $A = \mathrm{DPRk}(\Delta, x, y, \rho)$ be nonsingular.

If all $\delta_j \neq 0$, the inverse of A is a DPRk matrix

$$A^{-1} = \hat{\Delta} + \hat{x}\hat{\rho}\hat{y}^*,$$

where

$$\hat{\Delta} = \Delta^{-1}, \qquad \hat{x} = \Delta^{-1}x, \quad \hat{y} = \Delta^{-*}y, \quad \hat{\rho} = -\rho(I + y^*\Delta^{-1}x\rho)^{-1}.$$

Inverses (DPRk, cont.)

If $k=1$ and $\delta_j=0$, the inverse of A is an arrowhead matrix with the tip of the arrow at position (j,j). In particular, let P be the permutation matrix of the permutation $p=(1,2,\cdots,j-1,n,j,j+1,\cdots,n-1)$. Partition Δ, x and y as

$$\Delta=\begin{bmatrix}\Delta_1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \Delta_2\end{bmatrix},\quad x=\begin{bmatrix}x_1\\ x_j\\ x_2\end{bmatrix},\quad y=\begin{bmatrix}y_1\\ y_j\\ y_2\end{bmatrix}.$$

Then,

$$A^{-1}=P\begin{bmatrix}D & u\\ v^* & \alpha\end{bmatrix}P^T,$$

where

$$D=\begin{bmatrix}\Delta_1^{-1} & 0\\ 0 & \Delta_2^{-1}\end{bmatrix},\quad u=\begin{bmatrix}-\Delta_1^{-1}x_1\\ -\Delta_2^{-1}x_2\end{bmatrix}x_j^{-1},\quad v=\begin{bmatrix}-\Delta_1^{-*}y_1\\ -\Delta_2^{-*}y_2\end{bmatrix}y_j^{-1},$$
$$\alpha=(y_j^{-1})^*\left(\rho^{-1}+y_1^*\Delta_1^{-1}x_1+y_2^*\Delta_2^{-1}x_2\right)x_j^{-1}.$$

Eigenvalue decomposition

Right eigenpairs (λ, x) satisfy

$$Ax = x\lambda, \quad x \neq 0.$$

Usually, x is chosen such that λ is the standard form.

Eigenvalues are invariant under similarity.

> Eigenvalues are **NOT** shift invariant, that is, eigenvalues of the shifted matrix are **NOT** the shifted eigenvalues. (In general, $X^{-1}qX \neq qX^{-1}X = qI$)

If λ is in the standard form, it is invariant under similarity with complex numbers.

A Quaternion QR algorithm

by Angelika Bunse-Gerstner, Ralph Byers, and Volker Mehrmann, Numer. Math 55, 83-95 (1989)

Given $A \in \mathbb{Q}^{n\times n}$, the algorithm has four steps, as usual:

1. Reduce A to Hessenberg form by Householder reflectors:

$$X^*AX = H,$$

where X is unitary and H is an upper Hessenberg matrix.

2. Compute the Schur decomposition of H,

$$Q^*HQ = T,$$

where Q is unitary and T is upper triangular with eigenvalues of A on the diagonal.

3. Compute the eigenvectors V of T by solving the Sylvester equation:

$$TV - V\Lambda = 0.$$

Then $V^{-1}TV = \Lambda$.

4. Multiply

$$U = X * Q * V.$$

Then $U^{-1}AU = \Lambda$ is the eigenvalue decomposition of A.

The algorithm is derived for general matrices and requires $O(n^3)$ operations. The algorithm is stable and we use it for comparison.

Computing the Schur decomposition

Given the upper Hessenberg matrix $A \in \mathbb{Q}^{n\times n}$, the method applies complex shift μ to A by using Francis standard double shift on the matrix

$$M = A^2 - (\mu + \bar{\mu})A + \mu\bar{\mu}I$$

and applying it implicitly on A.

If $Ax = x\lambda$, then

$$\begin{aligned} Mx &= (A^2 - (\mu + \bar{\mu})A + \mu\bar{\mu}I)x = x\lambda^2 - x(\mu + \bar{\mu})\lambda + x\mu\bar{\mu} \\ &= x(\lambda^2 - (\mu + \bar{\mu})\lambda + \mu\bar{\mu}) \end{aligned}$$

For the perfect shift, $\mu = \lambda$, it holds

$$\lambda^2 - (\mu + \bar{\mu})\lambda + \mu\bar{\mu} = \lambda^2 - (\lambda + \bar{\lambda})\lambda + \lambda\bar{\lambda} = 0.$$

Details are given in Algorithm 4 in the Appendix of [BGBM89].

RQI with double shifts

We can apply the double shift μ and $\bar\mu$ similarly as in the [BGGM89] method.

The **Rayleigh Quotient Iteration with Double Shifts** (RQIds) produces sequences of shifts and vectors

$$\mu_k=\frac{1}{x_k^* x_k}x_k^*Ax_k,\quad y_k=(A^2-(\mu_k+\bar\mu_k)A+\mu_k\bar\mu_k I)^{-1}x_k,\quad x_{k+1}=\frac{y_k}{\|y_k\|},\quad k=0,1,2,\ldots$$

Due to the arrowhead or DPRk structure of A, one step of the method requires $O(n)$ operations:

Here y_k is the solution of the system

$$(A^2-(\mu_k+\bar\mu_k)A+\mu_k\bar\mu_k I)y_k=x_k.$$

RQIds for Arrowhead

Let

$$y_k = y, \quad x_k = \begin{bmatrix} x \\ \xi \end{bmatrix}, \quad \hat{\alpha} = \mu_k + \bar{\mu}_k, \quad \beta = \mu_k \bar{\mu}_k.$$

Notice that $\hat{\alpha}$ and β are real. Then:

$$\left(\begin{bmatrix} D & u \\ v^* & \alpha \end{bmatrix} \begin{bmatrix} D & u \\ v^* & \alpha \end{bmatrix} - \hat{\alpha} \begin{bmatrix} D & u \\ v^* & \alpha \end{bmatrix} + \beta \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \right) y = \begin{bmatrix} x \\ \chi \end{bmatrix}.$$

Therefore,

$$\left(\begin{bmatrix} D^2 + uv^* & Du + u\alpha \\ v^* D + \alpha v^* & v^* u + \alpha^2 \end{bmatrix} - \hat{\alpha} \begin{bmatrix} D & u \\ v^* & \alpha \end{bmatrix} + \beta \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \right) y = \begin{bmatrix} x \\ \chi \end{bmatrix},$$

so

$$\begin{bmatrix} D^2 - \hat{\alpha} D + \beta I + uv^* & Du + u(\alpha - \hat{\alpha}) \\ v^* D + (\alpha - \hat{\alpha}) v^* & v^* u + (\alpha - \hat{\alpha})\alpha + \beta \end{bmatrix} y = \begin{bmatrix} x \\ \chi \end{bmatrix}. \tag{1}$$

RQIds for Arrowhed (cont.)

The matrix $C=D^2-\hat{\alpha}D+\beta I+uv^*$ is a DPRk (DPR1) matrix,

$$C=\operatorname{DPRk}(D^2-\hat{\alpha}D+\beta I,u,v,1).$$

Multiplying of (1) by the block matrix $\begin{bmatrix} C^{-1} & \\ & 1\end{bmatrix}$ from the left yields

$$My\equiv\begin{bmatrix} I & C^{-1}(Du+u(\alpha-\hat{\alpha}))\\ v^*D+(\alpha-\hat{\alpha})v^* & v^*u+(\alpha-\hat{\alpha})\alpha+\beta\end{bmatrix}y=\begin{bmatrix}C^{-1}x\\ \xi\end{bmatrix}. \tag{2}$$

where M is an arrowhead matrix. Finally, $y=M^{-1}z$, where $z=\begin{bmatrix}C^{-1}x\\ \xi\end{bmatrix}$.

Due to the fast multiplication and computation of inverses, one step requires $O(n)$ operations.

Wielandt's deflation

- Let A be a (real, complex, or quaternionic) matrix.
- Let (λ, u) be a right eigenpair of A.
- Choose z such that $z^*u = 1$, say $z^* = [1/u_1 \quad 0 \quad \cdots \quad 0]$.
- Compute the deflated matrix $\tilde{A} = (I - uz^*)A$.
- Then $(0, u)$ is an eigenpair of $\tilde{A}$.
- Further, if (μ, v) is an eigenpair of A, then $(\mu, \tilde{v})$, where $\tilde{v} = (I - uz^*)v$ is an eigenpair of $\tilde{A}$.

Proofs: Using $Au = u\lambda$ and $z^*u = 1$, the first statement holds since

$$\tilde{A}u = (I - uz^*)Au = Au - uz^*Au = u\lambda - uz^*u\lambda = 0.$$

Further,

$$\begin{aligned}\tilde{A}\tilde{v} &= (I - uz^*)A(I - uz^*)v \\ &= (I - uz^*)Av - Auz^*v + uz^*Auz^*v \\ &= (I - uz^*)v\mu - u\lambda z^*v + uz^*u\lambda z^*v \\ &= \tilde{v}\mu\end{aligned}.$$

Deflation for Arrowhead

Lemma 1. Let A be an arrowhead matrix partitioned as

$$A = \begin{bmatrix} \delta & 0 & \chi \\ 0 & \Delta & x \\ \bar{\upsilon} & y^* & \alpha \end{bmatrix},$$

where χ, υ and α are scalars, x and y are vectors, and Δ is a diagonal matrix.

Let $\left(\lambda, \begin{bmatrix} \nu \\ u \\ \psi \end{bmatrix}\right)$, where ν and ψ are scalars, and u is a vector, be an eigenpair of A. Then,

$$\tilde{A} = \begin{bmatrix} 0 & 0^T \\ w & \hat{A} \end{bmatrix}, \qquad w = \begin{bmatrix} -u\frac{1}{\nu}\delta \\ -\psi\frac{1}{\nu}\delta + \bar{\upsilon} \end{bmatrix}, \tag{0}$$

and $\hat{A}$ is an arrowhead matrix

$$\hat{A} = \begin{bmatrix} \Delta & -u\frac{1}{\nu}\chi + x \\ y^* & -\psi\frac{1}{\nu}\chi + \alpha \end{bmatrix}. \tag{1}$$

Deflation for Arrowhead (cont.)

Lemma 2. Let A and $\hat{A}$ be as in Lemma 1. If $\left(\mu, \begin{bmatrix} \hat{z} \\ \hat{\xi} \end{bmatrix}\right)$ is an eigenpair of $\hat{A}$, then the eigenpair of A is

$$\left(\mu, \begin{bmatrix} \zeta \\ \hat{z} + u\frac{1}{\nu}\zeta \\ \hat{\xi} + \psi\frac{1}{\nu}\zeta \end{bmatrix}\right), \tag{2}$$

where ζ is the solution of the scalar Sylvester equation

$$\left(\delta + \chi\psi\frac{1}{\nu}\right)\zeta - \zeta\mu = -\chi\hat{\xi}. \tag{3}$$

Computing the eigenvectors

Let $\left(\lambda, \begin{bmatrix} \nu \\ u \\ \psi \end{bmatrix}\right)$ be an eigenpair of the matrix A, that is

$$\begin{bmatrix} \delta & 0 & \chi \\ 0 & \Delta & x \\ \bar{v} & y^* & \alpha \end{bmatrix} \begin{bmatrix} \nu \\ u \\ \psi \end{bmatrix} = \begin{bmatrix} \nu \\ u \\ \psi \end{bmatrix} \lambda.$$

If λ and ψ are known, then the other components of the eigenvector are solutions of scalar Sylvester equations

$$\begin{aligned} \delta\nu - \nu\lambda &= -\chi\psi, \\ \Delta_{ii}u_i - u_i\lambda &= -x_i\psi, \quad i = 1, \ldots, n-2. \end{aligned} \tag{4}$$

By setting

$$\gamma = \delta + \chi\psi\frac{1}{\nu}$$

the Sylvester equation (3) becomes

$$\gamma\zeta - \zeta\mu = -\chi\hat{\xi}. \tag{5}$$

Dividing (4) by ν from the right gives

$$\gamma = \nu\lambda\frac{1}{\nu}. \tag{6}$$

Algorithm

In the first (forward) pass, in each step the absolutely largest eigenvalue and its eigenvector are computed by the RQIds. The first element of the current vector x and the first and the last elements of the current eigenvector are stored. The current value γ is computed using (6) and stored. The deflation is then performed according to Lemma 1.

The eigenvectors are reconstructed bottom-up, that is from the smallest matrix to the original one (a backward pass). In each iteration, we need access to:

- the first element of the vector x which was used to define the current Arrow matrix,
- its absolutely largest eigenvalue, and
- the first and the last elements of the corresponding eigenvector.

In the ith step, for each $j = i + 1, \ldots, n$ the following steps are performed:

1. The equation (5) is solved for ζ (the first element of the eigenvector of the larger matrix). The quantity $\hat{\xi}$ is the last element of the eigenvectors and was stored in the forward pass.
2. The first element of the eigenvector of super-matrix is updated (set to ζ).
3. The last element of the eigenvectors of the super-matrix is updated using (2).

Iterations are completed in $O(n^2)$ operations.

After all iterations are completed, we have:

- the computed eigenvalue and its eigenvector (unchanged from the first run of the RQIds),
- all other eigenvalues and the last elements of their corresponding eigenvectors.

The rest of the elements of the remaining eigenvectors are computed using the procedure described above. This step also requires $O(n^2)$ operations.

Corrections

Due to floating-point error in operations with Quaternions, the computed eigenpairs have larger residuals than required. This is successfully remedied by running a few steps of the RQIds, starting from the computed eigenvectors. This has the effect of using nearly perfect shifts, so typically just a few additional iterations are needed to attain the desired accuracy. This step also requires $O(n^2)$ operations.

Pseudocode

Computing all eigenpairs of an Arrow matrix

Require: an Arrow matrix $A \in \mathbb{H}^{n\times n}$

Compute and store the first eigenpair (λ_1, u) using RQIds

Compute the deflated matrix $\hat{A}$ according to Lemma 1

Compute $\gamma = \nu\lambda\frac{1}{\nu}$

Compute and store ν, χ, ψ according to Lemma 1

for $i = 2, 3, \ldots, n-1$ **do**

Compute $g = \frac{1}{\nu}\chi$

Compute w from (0): $w = x - ug$

Compute the new matrix $\hat{A}$ from (1)

Compute and store an eigenpair (λ_i, u) of $\hat{A}$ using RQIds

Update and store $\gamma_i, \nu_i, \chi_i, \psi_i$ according to Lemma 1

end for

Compute and store the last eigenvalue

for $i = n-1, n-2, \ldots, 1$ **do**

for $j = i+1, i+2, \ldots, n$ **do**

Solve the Sylvester equation $\gamma_i\zeta - \zeta\lambda_j = -\chi_i\psi_j$ for ζ

Update ν_j and ψ_j, the first and the last element of the eigenvector of the super-matrix, respectively:

$\nu_j = \zeta, \quad \psi_j = \psi_j + \psi_i\frac{\zeta}{\nu_i}$

end for

end for

Reconstruct all eigenvectors from the computed eigenvalues and respective first and last elements using (4)

Correct the computed eigenpairs by running few steps of RQIds with nearly perfect shifts.

DPRk matrices

For DPRk matrices there are analogous results:

- RQIds for DPRk (multiplying by one DPRk matrix ad solving the system with another DPRk matrix with I on the diagonal) - $O(n)$
- Deflation for DPRk - $O(n)$
- Computing the eigenvectors of a DPRk - $O(n^2)$
- Algorithm for DPRk - $O(n^2)$

Perturbation theory

We have the following Bauer-Fike type theorem from *Sk. Safique Ahmad, Istkhar Ali, and Ivan Slapničar, Perturbation analysis of matrices over a quaternion division algebra, ETNA, Volume 54, pp. 128-149, 2021.*

Theorem 1 Let $A \in \mathbb{H}^{n\times n}$ be a diagonalizable matrix, with $A = XDX^{-1}$, where $X \in \mathbb{H}^{n\times n}$ is invertible and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with λ_i being the standard right eigenvalues of A. If μ is a standard right eigenvalue of $A + \Delta A$, then

$$\operatorname{dist}(\mu, \Lambda_s(A)) = \min_{\lambda_i \in \Lambda_s(A)} \{|\lambda_i - \mu|\} \leq \kappa(X) \|\Delta A\|_2.$$

Moreover, we have

$$\operatorname{dist}(\xi, \Lambda(A)) = \inf_{\eta_j \in \Lambda(A)} \{|\eta_j - \xi|\} \leq \kappa(X) \|\Delta A\|_2,$$

where $\xi \in \Lambda(A + \Delta A)$ and $\kappa(\cdot)$ is the condition number with respect to the matrix 2-norm.

Residual bounds

Theorem 2 Let $(\tilde{\lambda}, \tilde{x})$ be the approximate eigenpair of the matrix A, where $\|\tilde{x}\|_2 = 1$. Let

$$r = A\tilde{x} - \tilde{x}\tilde{\lambda}, \quad \Delta A = -r\tilde{x}^*.$$

Then, $(\tilde{\lambda}, \tilde{x})$ is the eigenpair of the matrix $A + \Delta A$ and $\|\Delta A\|_2 \leq \|r\|_2$.

Theorem 3 Let $(\tilde{\lambda}_i, \tilde{x}_i)$, $i = 1, \ldots, m$, be approximate eigenpairs of the matrix A, where $\|\tilde{x}_i\|_2 = 1$. Set $\tilde{\Lambda} = \operatorname{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_m)$ and $\tilde{X} = [\tilde{x}_1 \quad \cdots \quad \tilde{x}_m]$. We assume that eigenvectors are linearly independent. Let

$$R = A\tilde{X} - \tilde{X}\tilde{\Lambda}, \quad \Delta A = -R(\tilde{X}^*\tilde{X})^{-1}\tilde{X}^*.$$

Then, $(\tilde{\lambda}_i, \tilde{x}_i)$, $i = 1, \ldots, m$ are the eigenpairs of the matrix $A + \Delta A$ and

$$\|\Delta A\|_2 \leq \|R\|_2 \|(\tilde{X}^*\tilde{X})^{-1}\tilde{X}^*\|_2.$$

Error analysis

An error of the product of two quaternions is bounded as follows (see *Joldes, M.; Muller, J. M., Algorithms for manipulating quaternions in floating-point arithmetic. In IEEE 27th Symposium on Computer Arithmetic (ARITH), Portland, OR, USA, 2020, pp. 48-55*)

Lemma 3 Let $p, q \in \mathbb{H}$. Then

$$|fl(pq) - pq| \leq (5.75\varepsilon + \varepsilon^2)|p||q|.$$

Lemma 4 Let $p, q \in \mathbb{H}^n$, that is, $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$, where $p_i, q_i \in \mathbb{H}$ for $i = 1, \ldots, n$. Let $|p| \equiv (|p_1|, |p_2|, \ldots, |p_n|)$ and $|q| \equiv (|q_1|, |q_2|, \ldots, |q_n|)$ denote the corresponding vectors of component-wise absolute values. Then

$$|fl(p \cdot q) - p \cdot q| \leq (2n + 5.75)\varepsilon|p| \cdot |q| + \mathcal{O}(\varepsilon^2).$$

Corollary Let $A, B \in \mathbb{H}^{n \times n}$ be matrices of quaternions and ε, and let $|A|$ and $|B|$ denote the corresponding matrices of component-wise absolute values. Then

$$|fl(A \cdot B) - A \cdot B| \leq (2n + 5.75)\varepsilon|A| \cdot |B| + \mathcal{O}(\varepsilon^2).$$

Error bounds

For example, let $(\tilde{\mu}, \tilde{x})$ be the computed eigenpair of the matrix A, where $\tilde{\mu}$ is in the standard form and $\|\tilde{x}\|_2 = 1$. Then we can compute the residual r as in Theorem 2, and Theorem 1 implies that

$$\min_{\lambda_i \in \Lambda_s(A)} \{|\lambda_i - \tilde{\mu}|\} \leq \kappa(X)\|r\|_2.$$

We can use the bound effectively if the matrix is diagonalizable and we can approximate the condition of the eigenvector matrix $\kappa(X)$ by the condition of the computed eigenvector matrix $\tilde{X}$.

If we computed all eigenvalues and all eigenvectors of a diagonalizable matrix, $\tilde{\Lambda} = \operatorname{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$ and $\tilde{X}$, respectively, then we can compute the residual R as in Theorem 2. Inserting the bound for $\|\Delta A\|_2$ from Theorem 3 into Theorem 1, yields

$$\max_j \min_{\lambda_i \in \Lambda_s(A)} \{|\lambda_i - \tilde{\lambda}_j|\} \leq \kappa(\tilde{X})\|R\|_2 \|(\tilde{X}^* \tilde{X})^{-1} \tilde{X}^*\|_2.$$

If the matrix is normal or Hermitian, then $\kappa(X) = 1$, so the bounds are sharper.

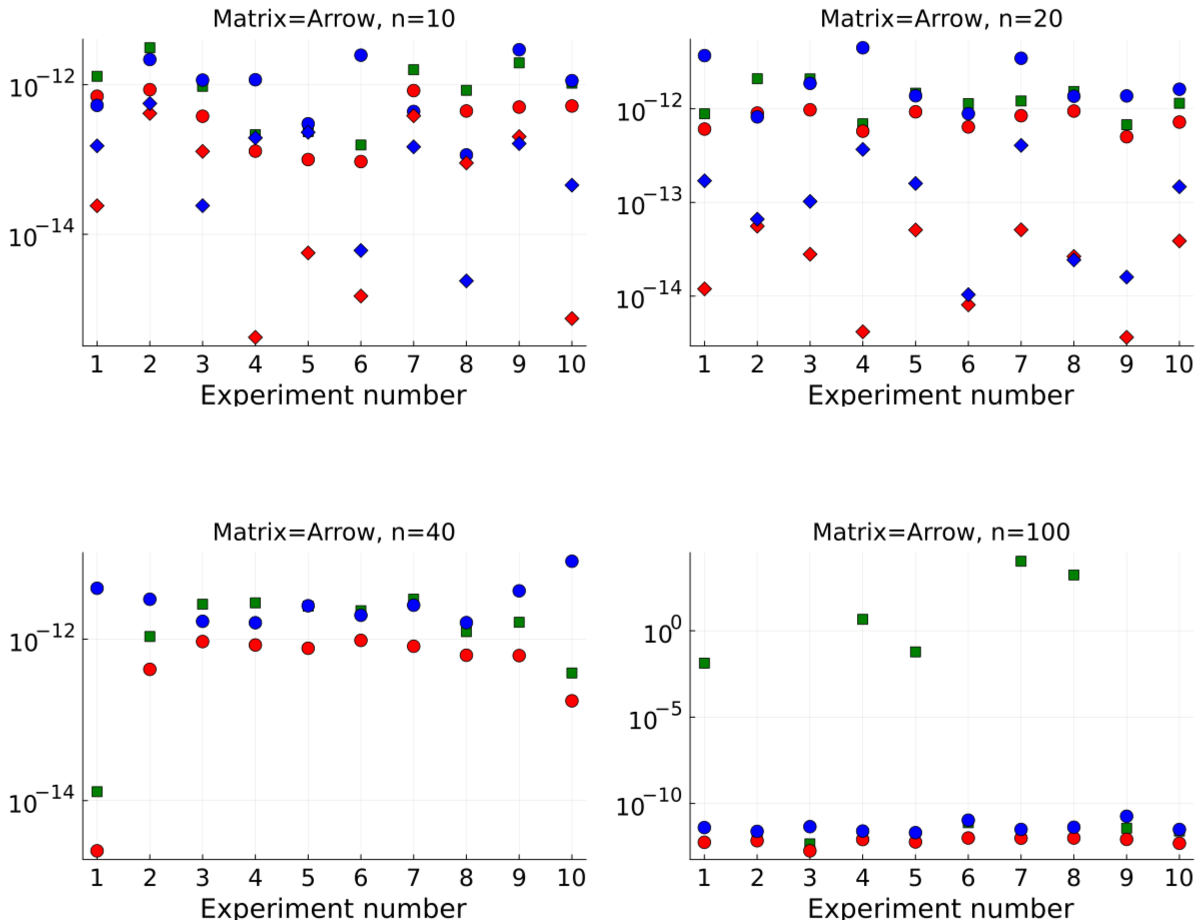
Codes and reference

The Julia codes are available at https://github.com/ivanslapnicar/MANAA

Details, including proofs, are in Fast Eigenvalue Decomposition of Arrowhead and Diagonal-Plus-Rank-k Matrices of Quaternions, Mathematics 2024, 12(9), 1327.

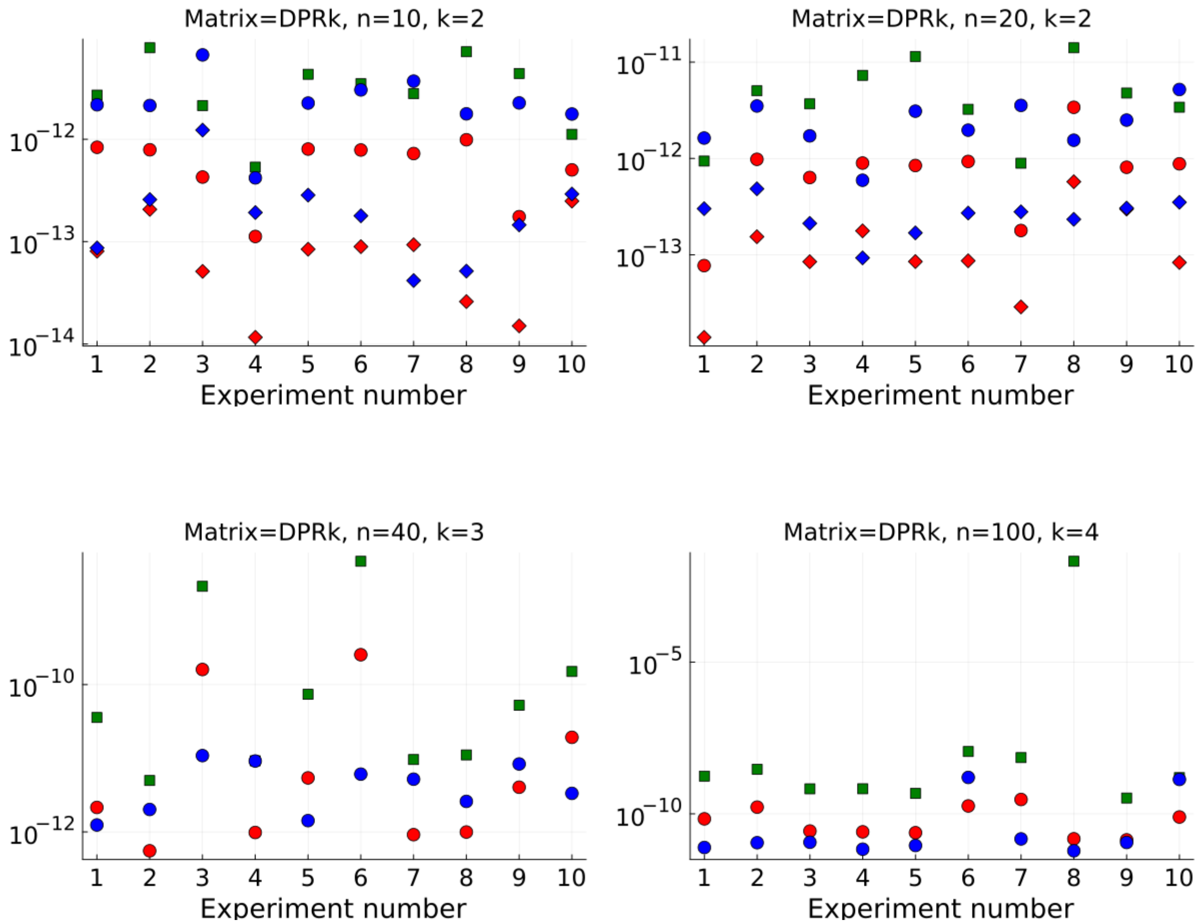
Example 1

Error bounds (green squares), residuals, and actual errors (using `BigFloat`) computed by RQIds (red dots and diamonds, respectively), and residuals and actual errors computed by QR (blue dots and diamonds, respectively). The actual errors are not computed for $n=40$ and $n=100$.



Example 2

Error bounds (green squares), residuals, and actual errors (using `BigFloat`) computed by RQIds (red dots and diamonds, respectively), and residuals and actual errors computed by QR (blue dots and diamonds, respectively). The actual errors are not computed for $n = 40$ and $n = 100$.



Iterations and running times

Mean number of iterations per eigenvalue and mean total running times for Arrow and DPRk matrices of orders $n = 10, 20, 40, 100$, using RQIds and QR, respectively.

n	# iters Arrow RQIds	Time RQIds	Time QR
10	8	0.00081	0.00079
20	9	0.0026	0.011
40	16	0.014	0.039
100	32	0.17	0.47

n	k	# iters DPRk RQIds	Time RQIds	Time QR
10	2	7	0.0018	0.00075
20	2	9	0.0077	0.011
40	3	16	0.031	0.071
100	4	27	0.25	0.85

Conclusions

The key contributions are the following:

- efficient algorithms for computing eigenvalue decompositions of Arrow and DPRk matrices of quaternions,
- the algorithms require $O(n^2)$ arithmetic operations, n being the order of the matrix,
- algorithms have proven error bounds,
- the computable residual is a good estimate of actual errors,
- actual errors are even smaller than predicted by the residuals,
- in all experiments errors and residuals are of the order of tolerance from respective algorithms,
- Rayleigh Quotient Iteration with double-shifts is efficient for non-Hermitian matrices,
- RQIds algorithms compare favorably in accuracy and speed to the quaternion QR method for general matrices.

Thank you!