Chapter 0 Preliminaries

If you are familiar with QR, eigen and Schur decomposition of matrices (especially tridiagonal and Hessenberg matrices), or if you do not care about these algorithms that have little to do with Krylov subspace, you can skip this chapter other than Section 1.

Section 1 Notations

1.1 Scalars, Vectors and Matrices

Sets are represented by capital squiggles, e.g., $\mathcal{L}$.

Scalars are represented by italic lowercase Greek letters, e.g., α. Its magnitude is denoted as $|\alpha|$; if α is a real number, it is its absolute value; if α is a complex number, it is its length in complex space.

Vectors are represented by italic lowercase English or Greek letters with arrows, e.g., $\vec{x}$. Its elements are represented by italic corresponding lowercase letters without arrows and with a subscript, e.g., x_i. The notation $\vec{x} \in \mathbb{F}^m$ is equivalent to saying that $\vec{x}$ has m elements:

$$\vec{x} \in \mathbb{F}^m \Leftrightarrow \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}; \quad x_1, \dots, x_m \in \mathbb{F}$$

Matrices are represented by uppercase italic English or Greek letters, e.g., A and Λ. Its elements are represented by the corresponding lowercase italic letters with subscripts, such as $a_{i,j}$. The notation $A \in \mathbb{F}^{m \times n}$ is equivalent to saying that A has m rows and n columns:

$$A \in \mathbb{F}^{m \times n} \Leftrightarrow A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}; \quad a_{1,1}, \dots, a_{m,n} \in \mathbb{F}$$

The submatrix of a matrix A is denoted as $A_{i_1:i_2, j_1:j_2}$, which means that the new matrix is composed of the i_1 to i_2 rows and j_1 to j_2 columns of matrix A (both inclusive).

The element-wise conjugates of scalars, vectors, and matrices are denoted as $\bar{\alpha}$, $\bar{\vec{x}}$ and $\bar{A}$, respectively. The transposed conjugation of vectors and matrices are denoted as $\vec{x}^*$ and A^*, respectively. If $A = A^*$ then we say matrix A is Hermitian.

For example, the vector $\vec{x} \in \mathbb{R}^3$ can be

$$\vec{x} = [0,1,2]^T$$

The matrix $A \in \mathbb{C}^{2\times 3}$ can be

$$A = \begin{bmatrix} 0 & 1 & \mathrm{i} \\ -1+2\mathrm{i} & -\mathrm{i} & 0 \end{bmatrix}$$

The diagonal of matrix A is the vector consists of $\{a_{i,i}\}$, denoted as $\operatorname{diag} A$. Its k-diagonal is the vector consists of $\{a_{i,i+k}\}$, denoted as $\operatorname{diag}_k A$. The notation "diag " can also be used in reverse, to form diagonal matrices.

The diagonal and 1-diagonal of the matrix in the example above are

$$\operatorname{diag} A = [0, -\mathrm{i}]^T, \qquad \operatorname{diag}_{-1} A = [1,0]^T$$

respectively. Correspondingly,

$$\operatorname{diag}\{1,2,3\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

1.2 Inner Product and Norm

The inner product of the two vectors in $\mathbb{F}^m$, namely $\vec{x}$ and $\vec{y}$, are denoted as $\langle \vec{x}, \vec{y} \rangle$:

$$\langle \vec{x}, \vec{y} \rangle := \sum_{i=1}^{n} \bar{x}_i y_i$$

it is not difficult to find that $\langle \vec{x}, \vec{y} \rangle = \vec{x}^* \vec{y}$.

The norm of the vector $\vec{x}$ is denoted as $\|\vec{x}\|$:

$$\|\vec{x}\| := \sqrt{\langle \vec{x}, \vec{x} \rangle}$$

The inner product of the vector $\vec{x} = [\mathrm{i}, 1+\mathrm{i}, -1]^T$ and $\vec{y} = [1,2,3]^T$ is

$$\langle \vec{x}, \vec{y} \rangle = -\mathrm{i} + 2(1-\mathrm{i}) - 3 = -1 - 3\mathrm{i}$$

The norm of $\vec{x}$ is $\|\vec{x}\| = \sqrt{1+2+1} = 2$. If $\|\vec{x}\| = 1$, we say that the vector $\vec{x}$ is a unit vector, such as $\vec{x} = \frac{1}{2}[\mathrm{i}, 1+\mathrm{i}, -1]^T$ is a unit vector.

The norm of matrix A is defined as

$$\|A\| = \max_{\|\vec{x}\|=1} \|A\vec{x}\|$$

It is relatively easy to prove that $\|A\|$ is the magnitude of the one with the largest magnitude among all the eigenvalues of $\|A\|$: simply decompose $\vec{x}$ with the eigenvectors of A.

The norm of matrix

$$A = \begin{bmatrix} 2 & -3 & 0 \\ 4 & 2 & 0 \\ -5 & 0 & 4 \end{bmatrix}$$

is $\|A\| = \left|2 + 2\sqrt{3}\mathrm{i}\right| = 4$.

1.3 Eigen Decomposition

Eigen decomposition of a matrix $A \in \mathbb{F}^{m \times m}$ is

$$AU = U\Lambda$$

Λ is a diagonal matrix $\Lambda = \operatorname{diag}\{\lambda_1, \dots, \lambda_m\}$ and U is a column vector matrix consisting of $U = [\vec{u}_i, \dots, \vec{u}_m]$ where

$$A\vec{u} = \lambda\vec{u}$$

for all eigenvalues λ_i and unit eigenvectors. For Hermitian matrices, you should have learned in a linear algebra course that $U^{-1} = U^*$, that U is a unitary matrix. Of course, for non-Hermitian matrices, there may be eigenvectors that follows $\vec{v}^* A = \lambda \vec{v}^*$ while $\vec{v} \neq \vec{u}$. The column vector matrix $V = [\vec{v}_i, \dots, \vec{v}_m]$ is therefore called the left eigenvectors, and U is called the right eigenvectors. In this not, the left eigenvectors are generally not covered: if you need them instead of right eigenvectors or solve $\vec{x}^* A = \vec{b}^*$, you can simply conjugate transpose A.

For example, the eigenvalues and eigenvectors of a Hermitian matrix

$$A=\begin{bmatrix}1 & 1-\mathrm{i} & 2\mathrm{i}\\ 1+\mathrm{i} & 2 & 1+\mathrm{i}\\ -2\mathrm{i} & 1-\mathrm{i} & 3\end{bmatrix}$$

are

$$\Lambda=\operatorname{diag}\begin{bmatrix}5\\2\\-1\end{bmatrix};\quad U=\frac{1}{3}\begin{bmatrix}\frac{1}{\sqrt{5}}(1+3\mathrm{i}) & -\sqrt{2} & \sqrt{-4-3\mathrm{i}}\\ \frac{1}{\sqrt{5}}(1+3\mathrm{i}) & \sqrt{-4-3\mathrm{i}} & -\sqrt{2}\\ \sqrt{5} & \sqrt{2} & \sqrt{2}\end{bmatrix}$$

respectively. The eigenvalues and eigenvectors of a non-Hermitian real matrix

$$A=\begin{bmatrix}2 & -3 & 0\\ 4 & 2 & 0\\ -5 & 0 & 4\end{bmatrix}$$

are

$$\Lambda=\operatorname{diag}\begin{bmatrix}4\\2+2\sqrt{3}\mathrm{i}\\2-2\sqrt{3}\mathrm{i}\end{bmatrix};\quad U=\frac{1}{\sqrt{187}}\begin{bmatrix}0 & 0 & 1\\ 2(\sqrt{3}-3\mathrm{i}) & -4(\sqrt{3}+\mathrm{i}) & 5\sqrt{3}\\ 2(\sqrt{3}+3\mathrm{i}) & -4(\sqrt{3}-\mathrm{i}) & 5\sqrt{3}\end{bmatrix}$$

respectively. It is not difficult to find that even if there are only real numbers, a non-Hermitian real matrix may have complex eigenvalues and eigenvectors. Yet, they must appear as mutually conjugate pairs. Using the conjugate complex root theorem of the characteristic equation $\det(xI-A)=0$ with real coefficients, you can directly prove this conclusion. Thus, I will not repeat it here.

The null space / kernel of a matrix or linear map A refers to the space spanned by all the vectors $\vec{x}$ of linear system $A\vec{x}=\vec{0}$, which is also the space spanned by all (right) eigenvectors with eigenvalue 0 of A.

1.4 Iteration Notation

The iteration number will be put to the subscript without parentheses, such as $\vec{q}_n$. Sometimes, since the subscript is reserved for the corresponding element or sub-matrix and vector notation, it will be placed in the superscript and the parentheses will be added to differentiate from powers, such as $\vec{q}^{(n)}$.

Section 2 QR Factorization

2.1 Definition

QR factorization refers to decompose a general matrix $A\in\mathbb{F}^{m\times n}$ (usually $m\geq n$) into a product of a unitary matrix Q and an upper triangular matrix R, where R is the abbreviation of right, which means that the right half of the triangular matrix is non-zero. The purpose of QR factorization is the same as Gaussian elimination, which is to turn a general matrix into a (upper) triangular matrix. But in QR, one shall multiply the upper triangular matrix and a unary matrix to restore the original matrix instead of a series of elementary row reductions.

The simplest method to implement the QR factorization is to perform the Schmidt orthogonalization on the column vectors $\{\vec{a}_1,\dots,\vec{a}_n\}$ of the matrix A to be decomposed:

$$\begin{cases} \vec{q}_1 = \dfrac{\vec{a}_1}{\|\vec{a}_1\|} = \tau_{1,1}\vec{a}_1 \\ \vec{q}_2 = \dfrac{\vec{a}_2 - \vec{q}_1\langle\vec{a}_2, \vec{q}_1\rangle}{\|\vec{a}_2 - \vec{q}_1\langle\vec{a}_2, \vec{q}_1\rangle\|} = \tau_{1,2}\vec{a}_1 + \tau_{2,2}\vec{a}_2 \\ \qquad\qquad \vdots \\ \vec{q}_n = \dfrac{\vec{a}_n - \sum_{i=1}^{n-1}\vec{q}_i\langle\vec{a}_i, \vec{q}_i\rangle}{\left\|\vec{a}_n - \sum_{i=1}^{n-1}\vec{q}_i\langle\vec{a}_i, \vec{q}_i\rangle\right\|} = \tau_{1,n}\vec{a}_1 + \cdots + \tau_{n,n}\vec{a}_n \end{cases}$$

which leads to

$$[\vec{q}_1, \dots, \vec{q}_n] = [\vec{a}_1, \dots, \vec{a}_n] \begin{bmatrix} \tau_{1,1} & \tau_{1,2} & \cdots & \tau_{1,n} \\ & \tau_{2,2} & \cdots & \tau_{2,n} \\ & & \ddots & \vdots \\ & & & \tau_{n,n} \end{bmatrix} \Leftrightarrow Q = AR^{-1}$$

The Schmidt orthogonalization QR result of matrix

$$A = \begin{bmatrix} 0.13909 & 0.577394 & 0.867797 \\ 0.276491 & 0.00213917 & 0.691662 \\ 0.193597 & 0.349887 & 0.798548 \end{bmatrix}$$

is

$$Q = \begin{bmatrix} 0.381 & 0.784086 & 0.489948 \\ 0.757373 & -0.568619 & 0.321028 \\ 0.530307 & 0.248762 & -0.810489 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} 2.73924 & -2.07092 & -690323 \\ 0 & 1.85684 & -249615 \\ 0 & 0 & 276728 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.365065 & 0.407155 & 1.27795 \\ 0 & 0.538549 & 0.485784 \\ 0 & 0 & 3.61366 \times 10^{-6} \end{bmatrix}$$

One may notice that the number of calculations required by the Schmidt orthogonalization method is relatively large. And for ill-conditioned matrices like the example above, due to the dramatic difference between different columns of R^{-1}, the inversion process will introduce excessive numerical error. Therefore, we often tend to use another method -- Householder reflection transformation.

2.2 Householder QR Factorization

The Householder QR factorization is a QR factorization that directly obtains the upper triangular matrix using the Householder reflection transformation

$$H = I - 2\vec{v}\vec{v}^*$$

where $\|\vec{v}\| = 1$.

The product of the Householder reflection transformation and the original matrix is

$$HA = A - \frac{1}{\gamma}\vec{v}\vec{v}^*A = \begin{bmatrix} a_{1,1} - v_1\langle\vec{v}, \vec{a}_1\rangle/\gamma & \cdots & \times \\ a_{2,1} - v_2\langle\vec{v}, \vec{a}_1\rangle/\gamma & \cdots & \times \\ \vdots & \ddots & \vdots \\ a_{m,1} - v_m\langle\vec{v}, \vec{a}_1\rangle/\gamma & \cdots & \times \end{bmatrix}$$

where $\vec{v}$ is not unit, hence, a unitization coefficient γ is introduced. To make the new elements $a_{2,1}, \dots, a_{m,1}$ zeros, we shall have

$$\begin{cases}\langle\vec{v},\vec{a}_1\rangle=\gamma/\alpha\\ v_1\neq\alpha a_{1,1}\\ v_2=\alpha a_{2,1},v_3=\alpha a_{3,1},\dots\end{cases}\Rightarrow\begin{cases}\vec{v}=[a_{1,1}\pm\|\vec{a}_1\|,a_{2,1},\dots]^T\\ \gamma=\|\vec{a}_1\|^2+\|\vec{a}_1\|\left|a_{1,1}\right|\end{cases}$$

Meanwhile, in order to ensure that the subsequent reflection transformations does not destroy these zeros, all subsequent $\vec{v}$'s first elements must be 0. By mathematical induction, at iteration k, the previous $(k-1)$ elements of $\vec{v}_k$ are all 0. Using a series of such reflection transformations, we can map the non-triangular part of the column vectors of A to the unit vectors, that is, there is $H_1,H_2,\dots,H_n$ such that

$$\begin{bmatrix}R\\O\end{bmatrix}=H_n\cdots H_2H_1A:=Q^*A$$

For the matrix in the above example, $\vec{v}_1$ should be the unitized result of $[0.13909+0.365065,0.276491,0.193597]^T$

Now,

$$H_1A=(I-2\vec{v}_1\vec{v}_1^T)A=\begin{bmatrix}-0.365065&-0.407155&-1.27795\\0&-0.537811&-0.485118\\0&-0.0281816&-0.0254241\end{bmatrix}$$

transforms the first column of A into the required upper triangular form.

$\vec{v}_2$ should be the unitized result of $[-0.537811-0.538549,-0.0281816]^T$, and so on. Finally,

$$R=H_3H_2H_1A=\begin{bmatrix}-0.365065&-0.407155&-1.27795\\0&0.538549&0.485784\\0&0&-3.61366\times10^{-6}\end{bmatrix}$$

The upper triangular matrix is directly available now, instead of inversion after Schmidt orthogonalization (only the signs of the columns are different).

Finally, you may still have question: is $Q=H_1H_2\cdots H_n$ a unitary matrix? According to the definition $H=I-2\vec{v}\vec{v}^*$, one can find that it H is a Hermitian, reflexive unitary matrix. Therefore, their product is also a unitary matrix. At the same time, the product of a Householder reflection transformation and a vector $H\vec{x}=\vec{x}-2\vec{v}(\vec{v}^*\vec{x})$ is quite easy to calculate: the number of floating-point operations is only 3 times the number of vector elements. Thus, we have the Householder QR factorization with time complexity of $\Theta(mn^2)$:

1. **Input: matrix to be decomposed** $A\in\mathbb{F}^{m\times n}$ $(m\geq n)$
2. **Output: Decomposition result matrix** $Q\in\mathbb{F}^{m\times m}$, $R\in\mathbb{F}^{n\times n}$
3. **For** $k=1,\dots,n$ **Do**
4. $\vec{y}\leftarrow A_{k:m,k}$
5. $\vec{w}\leftarrow\vec{y}+\operatorname{sign}(y_1)\|\vec{y}\|\vec{e}_1$
6. $\vec{v}_k\leftarrow\vec{w}/\|\vec{w}\|$
7. $A_{k:m,k:n}\leftarrow A_{k:m,k:n}-2\vec{v}_k(\vec{v}_k^*A_{k:m,k:n})$
8. **End For**
9. **Return** $R\equiv A_{1:n,1:n}$
10. **Return** $Q\equiv\prod_{k=1}^{n}\left(I-2\left[\vec{0}_k^T,\vec{v}_k^T\right]^T\left[\vec{0}_k^T,\vec{v}_k^*\right]\right)$

Algorithm 1 The Householder QR factorization

In practice, it is not necessary to store Q explicitly: $\vec{v}_k$ will be stored in the lower triangular part of A and the actual diagonal element (or the first elements of $\vec{v}_k$) will be stored in an additional vector. Later, if one needs to multiply Q or Q^* with other matrices, one can simply extract $\vec{v}_k$ and use $Q = H_1 H_2 \cdots H_n$ to calculate the multiplication, which is also the design of LAPACK's QR factorization APIs.

2.3 QR Factorization of Special Matrices

A tridiagonal matrix refers to a matrix of form like

$$A = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \gamma_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{F}^{n\times n}$$

It is not difficult to notice that since the lower half matrix only exists -1-diagonal elements, so $\vec{v}_1$ only has 2 elements. Hence, when β_1 is changed to 0 by $H_1 = I - 2\vec{v}_1\vec{v}_1^T$, β_2 will not be affected; and so on. Thus, $\vec{v}_k$ also only has 2 elements. On the other hand, because there are only 1-diagonal elements in the upper half of the matrix, one shall only compute a constant number of operations in each calculation $H_k A$. Therefore, the time complexity of the whole QR of the tridiagonal matrix is $\Theta(n)$.

A Hessenberg matrix refers to a matrix of the form like

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ \beta_1 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ & \beta_2 & a_{3,3} & \ddots & \vdots \\ & & \ddots & \ddots & a_{n-1,n} \\ & & & \beta_{n-1} & a_{n,n} \end{bmatrix} \in \mathbb{F}^{n\times n}$$

Same as above, $\vec{v}_k$ only has 2 elements. Yet, since the upper half matrix is (possibly) non-zero, each calculation of $H_k A$ requires updating the elements of k and $k+1$ row/column. Therefore, the time complexity of the entire QR of the Hessenberg matrix is $\Theta(n^2)$.

By analogy, if a matrix's $-\nu$-diagonal elements are non-zero at most and the upper half of the matrix is non-zero, then the time complexity of its entire QR is $\Theta(\nu^2 n^2)$. Furthermore, if its ν-diagonal elements are non-zero at most, the time complexity of the entire QR is $\Theta(\nu^2 n)$.

Section 3 Eigen Decomposition of Symmetric Tridiagonal Matrices

A real symmetric tridiagonal matrix refers to a matrix of the form

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{R}^{n\times n}$$

where $\beta_i \neq 0$ (otherwise A can be decomposed into three or more block

diagonal matrices).

Since methods such as QR or the Lanczos iteration (which will be introduced later) can transform a Hermitian matrix into a real symmetric tridiagonal matrix, we can assume $A \in \mathbb{F}^{n\times n}$ to be a real symmetric tridiagonal matrix without loss of generality, and we can focus on its eigen decomposition.

Note that when the tridiagonal matrix is QR factorized, $\vec{v}_k$ only has two elements that are non-zero, which are

$$\left[\alpha_k \pm \sqrt{\alpha_k^2+\beta_k^2},\quad \beta_k\right];\quad \gamma = \alpha_k^2+\beta_k^2+|\alpha_k|\sqrt{\alpha_k^2+\beta_k^2}$$

Furthermore, the R matrix is still a band matrix. We already know that the complexity of the entire QR is $\Theta(n)$ which is much smaller than the general matrix's $\Theta(n^3)$. This leads us to think what is the effect of QR factorization when solving eigen problems?

Note that not only R matrix has a special form

$$\begin{bmatrix} \times & \times & \times & & \\ & \times & \times & \ddots & \\ & & \ddots & \ddots & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

but also Q matrix has a special form

$$\begin{bmatrix} \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ & \times & \times & \cdots & \times \\ & & \ddots & \ddots & \vdots \\ & & & \times & \times \end{bmatrix}$$

which is a real Hessenberg matrix. For example, the result of the QR factorization of

$$A = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 0 & 5 & 9 & 7 \\ 0 & 0 & 7 & 16 \end{bmatrix}$$

is

$$Q = \begin{bmatrix} -0.316228 & 0.286039 & 0.467099 & 0.774597 \\ -0.948683 & -0.0953463 & -0.1557 & -0.258199 \\ 0 & 0.953463 & -0.1557 & -0.258199 \\ 0 & 0 & -0.856349 & 0.516398 \end{bmatrix}$$

$$R = \begin{bmatrix} -3.16228 & -4.74342 & -4.74342 & 0 \\ 0 & 5.24404 & 8.10443 & 6.67424 \\ 0 & 0 & -8.17424 & -14.7915 \\ 0 & 0 & 0 & 6.45497 \end{bmatrix}$$

Thus we have

$$RQ = \begin{bmatrix} 5.5 & -4.97494 & 0 & 0 \\ -4.97494 & 7.22727 & -7.79383 & 0 \\ 0 & -7.79383 & 13.9394 & -5.52771 \\ 0 & 0 & -5.52771 & 3.33333 \end{bmatrix}$$

This example shows that $A' := RQ = Q^TAQ$ is still a real symmetric tridiagonal matrix, so we might call this orthogonal similarity transformation QR transformation. To understand its effect on eigen decomposition, we need to know the following theorem: Let the eigen decomposition of A be $A = U^T\Lambda U$ like above, then for fixed point iteration

$$A_k \to Q_kR_k, \qquad A_{k+1} \leftarrow Q_k^TA_kQ_k$$

we have $\lim_{k\to\infty} A_k = \Lambda$, thus $U = \prod_{k=1}^{\infty} Q_k$. The proof of this theorem is long, so I won't repeat it here. If you want to learn more about it, please refer to the numerical analysis textbook e.g., (Stoer & Bulirsch, 1993).

The simplest eigen decomposition method is simply to repeat the above QR orthogonal transformations. But if there are adjacent eigenvalues with small gaps, the convergence speed may be very slow (see why the Lanczos algorithm can converge). Therefore, in practice, the eigenvalue shift method is used to accelerate the convergence:

$$A_k - s_k I \to Q_k R_k, \qquad A_{k+1} \leftarrow Q_k^T A_k Q_k + s_k I$$

For example, for matrix

$$A = \begin{bmatrix} 8.90947 & 1.68161 & 0 & 0 \\ 1.68161 & 9.03046 & 1.41927 & 0 \\ 0 & 1.41927 & 6.91227 & 4.06688 \\ 0 & 0 & 4.06688 & 2.16266 \end{bmatrix}$$

We know that its largest eigenvalue is about 11.1414. After 3 QR transformations without displacement,

$$A_3 = \begin{bmatrix} 10.4038 & 1.19198 & 0 & 0 \\ 1.19198 & 8.70635 & 1.23694 & 0 \\ 0 & 1.23694 & 8.13356 & 0.000103396 \\ 0 & 0 & 0.000103396 & -0.228824 \end{bmatrix}$$

It can be seen that the eigenvalues in the lower right corner does not completely converge. You would need to go through 7 iterations to make it converged. After that, if you wanted all the eigenvalues to converge, you would need an additional 100 iterations. At this point,

$$A_{110} \approx \text{diag}\{11.1414, 9.12895, 6.97338, -0.228824\}$$

The eigenvalues are sorted in descending order.

However, if we perform 3 QR transformations with a shift of 11.1, we have

$$A_3' = \begin{bmatrix} 5.54795 & 2.90074 & 0 & 0 \\ 2.90074 & 1.27316 & 0.649228 & 0 \\ 0 & 0.649228 & 9.0524 & 0.0000399188 \\ 0 & 0 & 0.0000399188 & 11.1414 \end{bmatrix}$$

You can see that the value in the lower right corner is very close to 11.1414 and the off-diagonal element is close to 0, which means that this matrix can be divided into separate block diagonal matrices. Then we say that the value in the lower right corner is close to convergence.

This also gives us a very simple but effective convergence criterion: if A_k's $|\beta_i| \ll |\alpha_i| + |\alpha_{i+1}|$, then it can be divided into two block diagonal matrices. Furthermore, if a block diagonal matrix is of size 1, then it is a converged eigenvalue.

The shift of the k step s_k is often chosen to be eigenvalue of the 2×2 lower right corner of A_k that is closer to α_{i+1} (this is only an experience setup). The purpose is to make the eigenvalue closer to s_k converges first, while this eigenvalue often appears in the lower right corner, which can be seen in the example above. In the iterations, when the 2×2 lower right corner of A_k has $|\beta_i| \ll |\alpha_i| + |\alpha_{i+1}|$, the eigenvalue has then converged. Repeating this process until all eigenvalues converge, we have obtained the eigen decomposition of A.

For the same matrix A as the above example, the eigenvalues of the lower right 2×2 matrix closer to α_4 is approximately -0.172021. After a QR transformation, we get

$$A_1 = \begin{bmatrix} 9.5156 & 1.61262 & 0 & 0 \\ 1.61262 & 8.82698 & 1.26542 & 0 \\ 0 & 1.26542 & 8.90097 & 0.0340491 \\ 0 & 0 & 0.0340491 & -0.228694 \end{bmatrix}$$

The eigenvalues of the lower right 2×2 matrix that are closer to α_4 is approximately -0.228821. Therefore,

$$A_2 = \begin{bmatrix} 10.0168 & 1.43072 & 0 & 0 \\ 1.43072 & 8.6946 & 1.27857 & 0 \\ 0 & 1.27857 & 8.53231 & 9.94368 \times 10^{-9} \\ 0 & 0 & 9.94368 \times 10^{-9} & -0.228824 \end{bmatrix}$$

It can be seen that the accuracy has far exceeded the above example's A_3'. Similarly,

$$A_3 = \begin{bmatrix} 10.3834 & 1.2079 & 0 & 0 \\ 1.2079 & 8.69909 & 1.2416 & 0 \\ 0 & 1.2416 & 8.16118 & 0 \\ 0 & 0 & 0 & -0.228824 \end{bmatrix}$$

whose off-diagonal elements is 0 relative to the $-0.228824 + 8.16118$ in machine precision, that is, the eigenvalue -0.228824 has converged. Next, we divide the matrix, let $A \leftarrow A^{(3)}_{1:3,1:3}$, repeat the above process. This time, after 3 iterations, we have

$$A_3 = \begin{bmatrix} 11.1281 & 0.162862 & 0 \\ 0.162862 & 9.14221 & 0 \\ 0 & 0 & 6.97338 \end{bmatrix}$$

The eigenvalue 6.97338 converges. Finally, divide the matrix once more and iterate for once, all eigenvalues converge.

I believe you can see that the eigenvalue shifts greatly accelerates the convergence, but it also brings about a minor problem that the eigenvalues are not mutually sorted. Of course, we can solve it by exchanging eigenvalues and eigenvectors at the same time. The average time complexity of quick sorting them is then $O(n^2 \log n)$.

You may also consider that due to the limitation of machine precision, if s_k is very large, $A_k - s_k I$ may lead to loss of accuracy. It is true. Therefore, in practice, implicit shifts are generally used instead of explicit ones. The principle of implicit shifts requires the proof of the following theorem: if matrix A is symmetric and non-singular and $B = Q^T A Q$ where Q is an orthogonal matrix and a B is a tridiagonal matrix whose off-diagonal elements are all greater than 0, then the first column of the Q is completely determines Q and B. The proof of this theorem is not difficult, but to lighten your burden, I won't introduce it here while you can still refer to the numerical analysis textbook such as (Stoer & Bulirsch, 1993).

By this theorem, the first column of Q_k is required to be exactly the first column of the QR factorization's Q matrix of $A_k - s_k I$. Also, $\vec{v}_i$ only has two non-zero elements in QR factorization, so the first column of matrix Q is decided only by $\vec{v}_1$. Thus, Q_k and A_{k+1} can be calculated in turn.

For example, for matrix

$$A = \begin{bmatrix} 5.93109 & 3.64299 & 0 & 0 & 0 \\ 3.64299 & 3.46141 & 4.04806 & 0 & 0 \\ 0 & 4.04806 & 7.23658 & 2.15139 & 0 \\ 0 & 0 & 2.15139 & 7.57971 & 2.29058 \\ 0 & 0 & 0 & 2.29058 & 1.52605 \end{bmatrix} \equiv A_1$$

We can determine $s_1 \approx 0.757031$, then, the first Householder reflection vector for QR factorizing $A_1 - s_1 I$ is

$$\vec{v}_1^{(1)} = [0.953325, 0.301945, 0, 0, 0]^T$$

whose corresponding reflector is $H_1^{(1)} = I - 2\vec{v}_1^{(1)}\vec{v}_1^{(1)^T}$, which will transform A_1 into

$$H_1^{(1)} A_1 H_1^{(1)} = \begin{bmatrix} 8.54228 & -0.065616 & -2.33048 & 0 & 0 \\ -0.065616 & 0.850224 & 3.30993 & 0 & 0 \\ -2.33048 & 3.30993 & 7.23658 & 2.15139 & 0 \\ 0 & 0 & 2.15139 & 7.57971 & 2.29058 \\ 0 & 0 & 0 & 2.29058 & 1.52605 \end{bmatrix}$$

Non-zero element appears in the 2- and -2-diagonals. According to QR Factorization of Special Matrices, $\vec{v}_2^{(1)}$ should be generated by $\vec{a}_{2:3,1}^{(1)}$ instead $\vec{a}_{2:3,2}^{(1)}$ to eliminate redundant non-zero off-diagonal elements, i.e.,

$$\vec{v}_2^{(1)} = [0, -0.716988, -0.697085, 0, 0]^T$$

After the reflection transformation,

$$H_2^{(1)} H_1^{(1)} A_1 H_1^{(1)} H_2^{(1)} = \begin{bmatrix} 8.54228 & 2.33141 & 0 & 0 & 0 \\ 2.33141 & 7.41776 & 3.12502 & -2.15054 & 0 \\ 0 & 3.12502 & 0.669045 & 0.0605497 & 0 \\ 0 & -2.15054 & 0.0605497 & 7.57971 & 2.29058 \\ 0 & 0 & 0 & 2.29058 & 1.52605 \end{bmatrix}$$

Similarly, by using $\vec{v}_3^{(1)}$ and $\vec{v}_4^{(1)}$ we finally obtain A_2:

$$H_4^{(1)} \cdots H_1^{(1)} A_1 H_1^{(1)} \cdots H_4^{(1)} = \begin{bmatrix} 8.54228 & 2.33141 & 0 & 0 & 0 \\ 2.33141 & 7.41776 & -3.79349 & 0 & 0 \\ 0 & -3.79349 & 2.83344 & -3.45871 & 0 \\ 0 & 0 & -3.45871 & 6.18033 & -0.0016244 \\ 0 & 0 & 0 & -0.0016244 & 0.761041 \end{bmatrix}$$

The implicit shift method avoids adding or subtracting s_k that may be too large while maintaining the same convergency as the explicit shift method. In addition, to avoid storing the entire matrix explicitly, we need to know how to simply compute the transformation of $H_{j-1}^{(k)} \cdots H_1^{(k)} A_k H_1^{(k)} \cdots H_{j-1}^{(k)}$ under $H_j^{(k)}$.

In general, $\vec{v}_j^{(k)} = \left[0, \ldots, 0, \beta_{j-1} \pm \sqrt{\beta_{j-1}^2 + c^2}, c, 0, \ldots, 0\right]^T$ and

$$H_j^{(k)} = I - \frac{1}{\gamma}\vec{v}_j^{(k)}\vec{v}_j^{(k)^T} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & h_1 & h_3 & 0 \\ 0 & h_3 & h_2 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, h_1 = -h_2 = c^2/\gamma - 1$$

where $\gamma = \beta_{j-1}^2 + c^2 \pm \beta_{j-1}\sqrt{\beta_{j-1}^2 + c^2}$, $h_3 = -c\left(\beta_{j-1} \pm \sqrt{\beta_{j-1}^2 + c^2}\right)/\gamma$. The previous transformation result $H_{j-1}^{(k)} \cdots H_1^{(k)} A_k H_1^{(k)} \cdots H_{j-1}^{(k)}$ has the following form:

$$\begin{bmatrix}
\alpha_1 & \beta_1 & & & & & & \\
\beta_1 & \ddots & \ddots & & & & & \\
 & \ddots & \alpha_{j-2} & \beta_{j-2} & & & & \\
 & & \beta_{j-2} & \alpha_{j-1} & \beta_{j-1} & c & & \\
 & & & \beta_{j-1} & \alpha_j & \beta_j & & \\
 & & & c & \beta_j & \alpha_{j+1} & \ddots & \\
 & & & & & \ddots & \ddots & \beta_{n-1} \\
 & & & & & & \beta_{n-1} & \alpha_n
\end{bmatrix}$$

So, we have

$$\begin{bmatrix} \beta_{j-1} & \alpha_j & \beta_j & c \\ 0 & \beta_j & \alpha_{j+1} & \beta_{j+1} \end{bmatrix} \leftarrow \begin{bmatrix} h_1 & h_3 \\ h_3 & h_2 \end{bmatrix} \begin{bmatrix} \beta_{j-1} & \alpha_j & \beta_j & 0 \\ c & \beta_j & \alpha_{j+1} & \beta_{j+1} \end{bmatrix}$$
$$\begin{bmatrix} \alpha_j & \beta_j \\ \beta_j & \alpha_{j+1} \end{bmatrix} \leftarrow \begin{bmatrix} \alpha_j & \beta_j \\ \beta_j & \alpha_{j+1} \end{bmatrix} \begin{bmatrix} h_1 & h_3 \\ h_3 & h_2 \end{bmatrix} \tag{0.1}$$

In this way, the only thing we need is to calculate and store h_1, h_2, h_3 and c to complete the whole process.

Furthermore, if eigenvectors are desired, calculate $\prod_{k=1} H_1^{(k)} \cdots H_{n-1}^{(k)}$ (multiply in the right direction). Finally, we obtain the eigen decomposition of the real symmetric tridiagonal matrix by the QR transformation with implicit shifts:

1. **Input: real symmetric tridiagonal matrix** $\{\alpha_i\},\{\beta_i\}$**;** **matrix** U **which eigenvectors need to left-multiply**
2. **Output: The eigenvalues of the matrix stored in** $\{\alpha_i\}$**;** **the eigenvectors left-multiply** U **stored in** U
3. **For** $k = n, \ldots, 2$ **Do**
4. **For** $i = k-1, \ldots, 1$ **Do**
5. **If** $|\beta_i| < \varepsilon(|\alpha_i| + |\alpha_{i+1}|)$ **Then Break** i
6. **End For**
7. **If** $i = k-1$ **Then Skip** k **/*Eigenvalue converged*/**
8. $s \leftarrow \underset{\lambda}{\operatorname{argmin}} |\lambda - \alpha_k|, \lambda = \frac{1}{2}(\alpha_{k-1} + \alpha_k \pm \sqrt{(\alpha_{k-1} - \alpha_k)^2 + 4\beta_{k-1}^2})$
9. **For** $j = i, \ldots, k$ **Do**
10. **If** $j = i$ **Then**
11. $\|\vec{y}\| \leftarrow \sqrt{(\alpha_i - s)^2 + \beta_i^2}$**;** $b \leftarrow \alpha_i - s$
12. $c \leftarrow \beta_i$
13. **Else**
14. $\|\vec{y}\| \leftarrow \sqrt{c^2 + \beta_{j-1}^2}$**;** $b \leftarrow \beta_{j-1}$
15. **End If**
16. $\gamma \leftarrow \|\vec{y}\|^2 + |b|\|\vec{y}\|$
17. $h_1 \leftarrow c^2/\gamma - 1$
18. $h_2 \leftarrow 1 - c^2/\gamma$
19. $h_3 \leftarrow -c(b + \operatorname{sign} b\, \|\vec{y}\|)/\gamma$
20. **Updating** $\{\alpha_i\}$**,** $\{\beta_i\}$ **and** c **with equation**(0.1)

Divide matrix by small off-diagonal element β_i.

Implicit shift QR: update the tridiagonal matrix A and the eigenvector U. The shift s is chosen to be the one stated above.

```
    U_{:,j:j+1} ← U_{:,j:j+1} [h1 h3; h3 h2]
End For
Restart k                    /*Did not converge, restart loop k*/
End For
```

Algorithm 2 Eigen decomposition of real symmetric tridiagonal matrix with implicit shift QR transformations

If the eigenvectors are not desired, the time complexity of this algorithm is on average $O(n^2)$: each eigenvalue requires $O(n)$ reflection transformations to converge, roughly about a $2n$ steps. Otherwise, the average time complexity is $O(n^3)$.

Section 4 Schur Decomposition

4.1 Definition

Schur decomposition of matrix $A \in \mathbb{C}^{n\times n}$ is to unary transform (by unitary matrix $U \in \mathbb{C}^{n\times n}$) it to an upper triangular matrix $T \in \mathbb{C}^{n\times n}$, i.e.,

$$U^* AU = T$$

According to the properties of the determinant of upper triangular matrix, the eigenvalues of T are the eigenvalues of A. Specifically, this is known as the complex Schur decomposition.

Looking at the equation $AU = UT$ column by column, the first k columns are:

$$A\vec{u}_k = \lambda_k \vec{u}_k + \sum_{i=1}^{k-1} t_{i,k}\vec{u}_i \quad (\lambda_k := t_{k,k})$$

that is,

$$A\vec{u}_k \in \text{span}\{\vec{u}_1, \dots, \vec{u}_k\}$$

Hence, the first k Schur vectors $\{\vec{u}_1, \dots, \vec{u}_k\}$ form an invariant subspace of A. The most special one is $\{\vec{u}_1\}$ -- it is the eigenvector of $t_{1,1}$.

Correspondingly, real matrices $A \in \mathbb{R}^{n\times n}$ can also be unary transformed (by $U \in \mathbb{R}^{n\times n}$) to a quasi-upper triangular form unitary matrices, i.e.,

$$U^T AU = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & T_{1,t} \\ & T_{2,2} & \cdots & T_{2,t} \\ & & \ddots & \vdots \\ & & & T_{t,t} \end{bmatrix}$$

$T_{i,i}$ is either a real number meaning that A has that eigenvalue, or a 2×2 matrix $\begin{bmatrix} \alpha & \beta \\ \gamma & \alpha \end{bmatrix}$ ($\beta\gamma < 0$) meaning that A has conjugate eigenvalues $\alpha \pm \mathrm{i}\sqrt{-\beta\gamma}$. This is known as the real Schur decomposition.

For example, the matrix

$$A = \begin{bmatrix} 3.74809 & 3.88764 & 2.10135 & 1.75194 & 0.268092 & 1.37127 \\ 2.4017 & 1.72176 & 0.89634 & 0.585313 & 3.80375 & 3.48008 \\ 1.87596 & 1.30346 & 2.28016 & 3.55031 & 0.234249 & 1.23674 \\ 4.60383 & 3.65656 & 1.50011 & 4.04174 & 1.36608 & 3.30755 \\ 0.57907 & 4.66138 & 1.94013 & 2.84008 & 2.07986 & 4.88629 \\ 1.22869 & 1.64877 & 3.81297 & 4.6123 & 3.28562 & 3.07936 \end{bmatrix}$$

has real Schur decomposition

$$T = \begin{bmatrix} 15.109 & 2.26233 & -0.591732 & -1.39415 & -0.0581278 & -1.59071 \\ 0 & -2.93657 & -1.12939 & 0.929401 & 0.140671 & 0.526973 \\ 0 & 0 & 2.22716 & 3.37144 & 1.39725 & 0.0647171 \\ 0 & 0 & -0.812306 & 2.22716 & -2.56696 & 2.23815 \\ 0 & 0 & 0 & 0 & -0.898346 & -0.971017 \\ 0 & 0 & 0 & 0 & 0 & 1.22254 \end{bmatrix}$$

and complex Schur decomposition

$$T' = \begin{bmatrix} 15.109 & 0.275098 & 0.734986 & 1.24614 & 0.0519061 & 1.42045 \\ 0 & -2.93657 & -0.687932 & 0.374949 & 0.047575 & 0.178222 \\ 0 & 0 & 2.22716 & 2.3174 & 1.30758 & 0.0102787 \\ 0 & 0 & 0 & 2.22716 & -2.32477 & 1.76264 \\ 0 & 0 & 0 & 0 & -0.898346 & -0.971017 \\ 0 & 0 & 0 & 0 & 0 & 1.22254 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 2.24555 & 0.345444 & 0.285301 & -0.0261648 & -0.716017 \\ 0 & 0 & -0.849901 & 0.896181 & 0.132382 & 0.49592 \\ 0 & 0 & 1.65488 & 1.08572 & 1.06903 & -0.987859 \\ 0 & 0 & 0 & -1.65488 & 0.533285 & -0.964689 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathrm{i}$$

4.2 Schur Decomposition of Hessenberg Matrices

Since QR or the Arnoldi iterations (which will be introduced later) can unary transform a general square matrix into a Hessenberg matrix, we can assume $A \in \mathbb{F}^{n\times n}$ to be a Hessenberg matrix without loss of generality, and we can focus on its Schur and eigen decomposition.

Same as the idea in the previous section, QR transformation is continuously applied to the original matrix A, and the final result matrix A_s is basically the Schur decomposition: if $A \in \mathbb{C}^{n\times n}$, A_s is an upper triangular matrix, and its diagonal elements are the eigenvalues of A; if $A \in \mathbb{R}^{n\times n}$, A_s is a block upper triangular matrix whose each diagonal matrix $T_{i,i}$ is either a real number indicating that A has the eigenvalue, or a 2×2 real matrix (the two diagonal elements are not necessarily the same) indicating that A has a conjugate pair of eigenvalues. The proof of this theorem can also be found in numerical analysis textbooks and will not be repeated here. Likewise, we have the theorem: if A is not singular and $B = Q^* AQ$ where Q is a unary matrix and B is a Hessenberg matrix, then the first column of Q completely determines Q and B. This allows us to use implicit shifts QR transformation method to speed up the convergence.

In addition, for real matrices, since the shifts may be a complex number, which will introduce complex arithmetic that we do not desire, method named two-step QR algorithm shall be applied. This means to calculate

$$A_{k+2} = Q_{k+1} Q_k A_k Q_k^T Q_{k+1}^T$$

once. The two shifts are taken as s and $\bar{s}$, since the complex eigenvalues of real matrix always appear in the form of conjugate complex pairs, avoiding complex arithmetic. Now,

$$\begin{aligned} A_k - sI &= Q_k^T R_k \\ A_{k+1} &= Q_k A_k Q_k^T \\ A_{k+1} - \bar{s}I &= Q_{k+1}^T R_{k+1} \end{aligned}$$

Hence,

$$Q_k A_k Q_k^T - \bar{s}I = Q_{k+1}^T R_{k+1} \Leftrightarrow A_k - \bar{s}I = Q_k^T Q_{k+1}^T R_{k+1} Q_k$$

If we define

$$M = (A_k - sI)(A_k - \bar{s}I)$$

Then

$$R = QM$$

where $R = R_{k+1} R_k$ and $Q = Q_{k+1} Q_k$. Therefore, $A_{k+2} = Q_{k+1} Q_k A_k Q_k^T Q_{k+1}^T$ becomes

$$A_{k+2} = Q A_k Q^T$$

In this way, we only need to determine the first column of the matrix Q of the QR factorization of M, and then gradually restore the matrix to Hessenberg form.

Note that

$$M = (A_k - sI)(A_k - \bar{s}I) = A_k^2 - (s + \bar{s})A_k + s\bar{s}I$$

only A_k^2 contributes the lower 2-diagonal elements of M. After calculation, it can be found that the non-zero part of the first column of M is

$$\overrightarrow{m}_{1:3,1} = \begin{bmatrix} s\bar{s} - (s + \bar{s})a_{1,1} + a_{1,1}^2 + a_{1,2}a_{2,1} \\ a_{2,1}\left(a_{1,1} + a_{2,2} - (s + \bar{s})\right) \\ a_{2,1}a_{3,2} \end{bmatrix}$$

Therefore, the first column of Q is generated by the reflection vector $\vec{v}_1^{(k)} = [m_{1,1} \pm \|\overrightarrow{m}_{1:3,1}\|, m_{2,1}, m_{3,1}, 0, \ldots, 0]^T$ (not unitized).

For example, if we want to Schur decompose a matrix

$$A = \begin{bmatrix} -0.260662 & -0.204006 & 0.13533 & -0.408664 & 0.179327 & -0.543618 \\ 0.60462 & -0.46534 & -0.996806 & -0.352452 & 0.372393 & 0.698925 \\ 0 & 0.806209 & 0.186408 & 0.234502 & -0.613998 & -0.23159 \\ 0 & 0 & -0.0570047 & -0.190926 & -0.196383 & -0.0797846 \\ 0 & 0 & 0 & 0.321647 & -0.916313 & 0.972468 \\ 0 & 0 & 0 & 0 & -0.399808 & -0.956982 \end{bmatrix}$$

In the first step, likewise, we take the shift as the eigenvalue of the lower right 2×2 submatrix of A_k

$$s + \bar{s} = a_{n,n} + a_{n-1,n-1}, \qquad s\bar{s} = a_{n,n}a_{n-1,n-1} - a_{n-1,n}a_{n,n-1}$$

which is $s, \bar{s} = -0.9366479 \pm 0.623207\mathrm{i}$. Substituting the above formula into $\vec{v}_1^{(k)}$ (scaled)

$$\begin{bmatrix} [(a_{n,n} - a_{1,1})(a_{n-1,n-1} - a_{1,1}) - a_{n-1,n}a_{n,n-1}]/a_{2,1} + a_{1,2} \\ a_{1,1} + a_{2,2} - (a_{n-1,n-1} + a_{n,n}) \\ a_{3,2} \end{bmatrix}$$

we have $\vec{v}_1^{(1)} = [0.907842, 0.343078, 0.241083, 0, 0, 0]^T$. The corresponding transformation result is ($A_1 = A$)

$$H_1^{(1)} A_1 H_1^{(1)}$$
$$= \begin{bmatrix} -0.106187 & 0.107572 & 0.478533 & 0.38186 & -0.0794742 & 0.0184555 \\ 0.204562 & -0.520838 & -0.988849 & -0.0537095 & 0.274591 & 0.911336 \\ -0.674894 & 0.618402 & 0.0874315 & 0.44443 & -0.682724 & -0.0823285 \\ 0.0249527 & 0.00942974 & -0.0503784 & -0.190926 & -0.196383 & -0.0797846 \\ 0 & 0 & 0 & 0.321647 & -0.916313 & 0.972468 \\ 0 & 0 & 0 & 0 & -0.399808 & -0.956982 \end{bmatrix}$$

Same as above, the QR transformation of $\vec{v}_{j\geq 2}^{(1)}$ must be generated from $\vec{a}_{j:,j-1}$, e.g.,

$$\vec{v}_2^{(1)} = [0, 0.803084, -0.595459, 0.0220157, 0, 0]^T$$

Then,

$$H_2^{(1)} H_1^{(1)} A_1 H_1^{(1)} H_2^{(1)}$$
$$= \begin{bmatrix} -0.106187 & 0.412985 & 0.252079 & 0.390232 & -0.0794742 & 0.0184555 \\ -0.705656 & 0.124894 & 0.829691 & 0.430457 & -0.725618 & -0.340104 \\ 0 & -0.791843 & -0.570308 & 0.0599027 & 0.0588961 & 0.84557 \\ 0 & -0.0190053 & 0.033684 & -0.178918 & -0.223802 & -0.114091 \\ 0 & -0.0113738 & 0.00843324 & 0.321335 & -0.916313 & 0.972468 \\ 0 & 0 & 0 & 0 & -0.399808 & -0.956982 \end{bmatrix}$$

Finally,

$$H_5^{(1)} \cdots H_1^{(1)} A_1 H_1^{(1)} \cdots H_5^{(1)}$$
$$= \begin{bmatrix} -0.106187 & 0.412985 & -0.260202 & -0.341625 & 0.127396 & 0.147782 \\ -0.705656 & 0.124894 & -0.829276 & -0.241575 & 0.867168 & 0.133552 \\ 0 & 0.792153 & -0.56691 & -0.0345199 & 0.445899 & -0.735743 \\ 0 & 0 & 0.0417044 & -0.227389 & -0.205796 & -0.380883 \\ 0 & 0 & 0 & -0.012747 & -0.674361 & -0.81885 \\ 0 & 0 & 0 & 0 & 0.589433 & -1.15386 \end{bmatrix}$$

which is A_2.

Performing 3 more QR transformations in this way, we have

$$A_5 = \begin{bmatrix} -0.252699 & 0.425771 & 0.00357906 & -0.26819 & -0.511788 & -0.527744 \\ -0.925479 & -0.407302 & -0.872757 & 0.289819 & -0.741294 & 0.495542 \\ 0 & 0.594922 & 0.154857 & 0.0279357 & 0.307838 & -0.403065 \\ 0 & 0 & 0.0166419 & -0.264934 & 0.137308 & 0.321643 \\ 0 & 0 & 0 & 0 & -0.884088 & 0.440563 \\ 0 & 0 & 0 & 0 & -0.970652 & -0.949648 \end{bmatrix}$$

One can see that the eigenvalue pair in the lower right corner has converged, but not yet been transformed into standard Schur form.

After convergence, we can ignore this part and perform QR transformation another 4 times. It can be found that two more real eigenvalues converge together, i.e.,

$$A_9 = \begin{bmatrix} -0.0284479 & 1.29193 & -0.0800464 & -0.286915 & -0.976772 & 0.0595435 \\ -0.679537 & -0.416318 & -0.311482 & -0.0007766 & 0.170012 & 0.583059 \\ 0 & 0 & -0.0462353 & -0.252171 & 0.155548 & 0.508649 \\ 0 & 0 & 0 & -0.279077 & 0.155469 & -0.285434 \\ 0 & 0 & 0 & 0 & -0.916868 & 0.440845 \\ 0 & 0 & 0 & 0 & -0.971273 & -0.916868 \end{bmatrix}$$

For the matrix in the upper left 2×2 corner, the eigenvalues are found to be complex numbers after calculation, hence, the calculation completed.

During calculation, we noticed that any reflection transformation matrix is

$$H_j^{(k)} = \begin{bmatrix} I & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & I \end{bmatrix}, H = \begin{bmatrix} h_1 & h_4 & h_5 \\ h_4 & h_2 & h_6 \\ h_5 & h_6 & h_3 \end{bmatrix}$$

Therefore, calculating the product of the reflection transformation matrix and the original matrix is easy: simply act H on the rows and columns respectively.

The converged diagonal element may not be in standard Schur form, but it is not difficult to transform it into a standard form: let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbb{R}^{2\times 2}$, the goal is to use the Householder unary transformation to transform it into $\begin{bmatrix} (a+d)/2 & e \\ f & (a+d)/2 \end{bmatrix}$, where e and f have different sign. Without losing generality, let $\vec{v} = \left[p, \sqrt{1-p^2}\right]^T$, then the solution is

$$2p^2 - 1, -2p\sqrt{1-p^2} = \frac{\sqrt{2}}{2}\sqrt{1 \pm \frac{|b+c|}{\sqrt{(b+c)^2 + (a-d)^2}}} \tag{0.2}$$

Thus $H = I - 2\vec{v}\vec{v}^T = \begin{bmatrix} 1-2p^2 & -2p\sqrt{1-p^2} \\ -2p\sqrt{1-p^2} & 2p^2-1 \end{bmatrix}$. Here, the positive sign is in front when $a > d$, and otherwise the negative sign is in front (obviously, no operation is required when $a = d$).

In addition, it is also possible that A has two real eigenvalues when $(a-d)^2 + 4bc \geq 0$. Now,

$$2p^2 - 1, -2p\sqrt{1-p^2} = \sqrt{\frac{2\alpha(b+c) + (a-d)\left[a - d \pm \sqrt{4bc + (a-d)^2}\right]}{2[(b+c)^2 + (a-d)^2]}} \tag{0.3}$$

where α is b for the first one and c for the second one, and the positive sign in the formula comes first.

Finally, we obtain the Schur decomposition of Hessenberg matrix by QR transformation with implicit shifts:

```
Input: real Hessenberg matrix A ∈ R^{n×n};
   matrix U which Schur vectors right multiplies
Output: Eigenvalues of this matrix {λ_i};
   Schur form is stored in A;
   Schur vector right multiply U stored in U
  For k = n, ... ,2 Do
    For i = k, ... ,2 Do
        If |a_{i,i-1}| < ε(|a_{i,i}| + |a_{i-1,i-1}|) Then Break i
    End For
    If i = k Then                    /*Real eigenvalue converges*/
        λ_k = a_{i,i}; Skip k
    End If
    Ifi = k − 1 Then                      /*Eigen pair converge*/
        H ← The symmetric orthogonal transformation
    matrix obtained by equation(0.2)and(0.3) from A_{i:k,i:k}
        A_{i:k,:} ← HA_{i:k,:}; A_{:,i:k} ← A_{:,i:k}H; U_{:,i:k} ← U_{:,i:k}H
        If |a_{i,k}a_{k,i}| < εa²_{i,i} Then a_{i,k} ← a_{k,i} ← 0
        λ_i, λ_k = α ± i√(−a_{i,k}a_{k,i})
        Skip k, k + 1
    End If
```

Find the lower diagonal element with less machine precision than the diagonal element to split the matrix

17. **For** $j = i, \dots, k-1$ **Do**
18. **If** $j = i$ **Then**
19. $\vec{v} \leftarrow \begin{bmatrix} [(a_{k,k} - a_{i,i})(a_{k-1,k-1} - a_{i,i}) - a_{k-1,k}a_{k,k-1}]/a_{i+1,i} + a_{i,i+1} \\ a_{i,i} + a_{i+1,i+1} - (a_{k-1,k-1} + a_{k,k}) \\ a_{i+2,i+1} \end{bmatrix}$
20. **Else**
21. $\vec{v} \leftarrow \vec{a}_{j:j+2,j-1}$
22. **End If**
23. $v_1 \leftarrow v_1 + \text{sign}(v_1)\,\|\vec{v}\|$; $H \leftarrow I - 2\vec{v}\vec{v}^T/\|\vec{v}\|^2$
24. $A_{j:j+2,:} \leftarrow HA_{j:j+2,:}$; $A_{:,j:j+2} \leftarrow A_{:,j:j+2}H$
25. $U_{:,j:j+2} \leftarrow U_{:,j:j+2}H$
26. **End For**
27. **If** $j \neq i$ **Then** $\vec{a}_{j+1:,j-1} \leftarrow \vec{0}$

/*Restored A's zeros to avoid accumulating errors*/

28. **End For**
29. **Restart** k /*Did not converge, restart loop k*/
30. **End For**

Implicit shift QR: Update the Hessenberg matrix A and Schur vectors U, the shifts are chosen to the ones stated above.

Algorithm 3 Schur decomposition of Hessenberg matrix by QR transformation with implicit shifts

Regardless of whether the eigenvectors are calculated or not, the time complexity of the algorithm is on average $O(n^3)$: it takes $O(n^2)$ steps for each eigenvalue to converge, roughly about $2n^2$ steps, i.e., 2 QR transformations.

4.3 Reorder of Schur Decomposition

The simplest and widely used Schur decomposition reorder method is derived from (Bait & Demmel, 1993). The original algorithm is only used to swap two adjacent 1×1 or 2×2 Schur diagonal blocks, but with bubble sort, we can rearrange the entire Schur decomposition.

Assuming that the eigenvalues of the adjacent Schur diagonal blocks to be exchanged are different (they cannot be exchanged if they are the same), they form something like

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{bmatrix}$$

where $A_{2,2} \in \mathbb{R}^{q\times q}$, $A_{1,1} \in \mathbb{R}^{p\times p}$, $p, q \in \{1,2\}$. A can be block diagonalized:

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{bmatrix} = \begin{bmatrix} I_p & -X \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A_{1,1} & 0 \\ 0 & A_{2,2} \end{bmatrix} \begin{bmatrix} I_p & X \\ 0 & I_q \end{bmatrix}$$

where $X \in \mathbb{R}^{p\times q}$ is the solution of equation

$$A_{1,1}X - XA_{2,2} = A_{1,2}$$

This equation can be rewritten as a system of linear equations:

$$\begin{bmatrix} \alpha_1 - \alpha_2 & -\gamma_2 & \beta_1 & 0 \\ -\beta_2 & \alpha_1 - \delta_2 & 0 & \beta_1 \\ \gamma_1 & 0 & \delta_1 - \alpha_2 & -\gamma_2 \\ 0 & \gamma_1 & -\beta_2 & \delta_1 - \delta_2 \end{bmatrix} \vec{x} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} \tag{0.4}$$

where $\vec{x} = [x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}]^T$. For the other three cases, the linear systems

will be simpler:

$$\begin{bmatrix} \alpha_1 - \alpha_2 & -\gamma_2 \\ -\beta_2 & \alpha_1 - \delta_2 \end{bmatrix} \vec{x} = \begin{bmatrix} t_3 \\ t_4 \end{bmatrix} \tag{0.5}$$

$$\begin{bmatrix} \alpha_1 - \alpha_2 & \beta_1 \\ \gamma_1 & \delta_1 - \alpha_2 \end{bmatrix} \vec{x} = \begin{bmatrix} t_1 \\ t_3 \end{bmatrix} \tag{0.6}$$

$$(\alpha_1 - \alpha_2)x = t_3 \tag{0.7}$$

QR factorization of X

$$Q^T \begin{bmatrix} -X \\ I_q \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

will be done later, so that

$$Q^T \begin{bmatrix} -X & I_p \\ I_q & 0 \end{bmatrix} = \begin{bmatrix} R & Q_{1,1}^T \\ 0 & Q_{1,2}^T \end{bmatrix}$$

Then,

$$Q^T A Q = \begin{bmatrix} RA_{2,2}R^{-1} & A'_{1,2} \\ 0 & Q_{1,2}^T A_{1,1} Q_{1,2}^{-T} \end{bmatrix}$$

which has same block upper triangular form and $RA_{2,2}R^{-1} \sim A_{1,1}$, $Q_{1,2}^T A_{1,1} Q_{1,2}^{-T} \sim A_{2,2}$. This completes the exchange. However, it is noted that if $RA_{2,2}R^{-1}$ or $Q_{1,2}^T A_{1,1} Q_{1,2}^{-T}$ are 2×2 matrices, they are not necessarily of standard Schur form, and require unary transformation(s). Combine them together, one can simply perform unary transformation to A_s and right multiply the Schur vector by Q':

$$A_s \leftarrow \begin{bmatrix} I_1 & 0 & 0 \\ 0 & Q'^* & 0 \\ 0 & 0 & I_2 \end{bmatrix} A_s \begin{bmatrix} I_1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & I_2 \end{bmatrix}, \qquad U \leftarrow U \begin{bmatrix} I_1 & 0 & 0 \\ 0 & Q' & 0 \\ 0 & 0 & I_2 \end{bmatrix}$$

The result of the Schur decomposition of the above example is

$$T = \begin{bmatrix} -0.222383 & 1.34818 & -0.00988929 & 0.275342 & -0.715743 & 0.685518 \\ -0.623287 & -0.222383 & 0.321451 & 0.0806696 & 0.37062 & 0.454874 \\ 0 & 0 & -0.0462354 & -0.252171 & -0.489121 & -0.208997 \\ 0 & 0 & 0 & -0.279077 & 0.117979 & 0.30286 \\ 0 & 0 & 0 & 0 & -0.916868 & 0.438544 \\ 0 & 0 & 0 & 0 & -0.972671 & -0.916868 \end{bmatrix}$$

If we want to sort according to the magnitude of the eigenvalues to make the largest one on the upper left corner, we need to move the diagonal block in the lower right corner to the upper left corner, and move -0.222383 and -0.0462354 to the lower right corner.

First of all, let

$$A = \begin{bmatrix} [-0.279077] & [0.117979 & 0.30286] \\ 0 & \begin{bmatrix} -0.916868 & 0.438544 \\ -0.972671 & -0.916868 \end{bmatrix} \end{bmatrix}$$

Solve equation $A_{1,1}X - XA_{2,2} = A_{1,2}$, i.e.,

$$\begin{bmatrix} \alpha_1 - \alpha_2 & -\gamma_2 \\ -\beta_2 & \alpha_1 - \alpha_2 \end{bmatrix} \vec{x} = \begin{bmatrix} t_3 \\ t_4 \end{bmatrix}$$

Substitute values into it, we have $\begin{bmatrix} 0.637791 & 0.972671 \\ -0.438544 & 0.637791 \end{bmatrix} \vec{x} = \begin{bmatrix} 0.117979 \\ 0.30286 \end{bmatrix}$. Therefore, we have $X = [-0.263204, 0.293879]$. Then, the full QR factorization yields the Q matrix

$$Q = \begin{bmatrix} -0.254535 & 0.26437 & 0.930226 \\ -0.967064 & -0.0695831 & -0.244839 \\ 0 & -0.961908 & 0.273374 \end{bmatrix}$$

and

$$Q^T AQ = \begin{bmatrix} -0.846506 & 0.441268 & -0.280672 \\ -0.977885 & -0.98723 & -0.0663183 \\ 0 & 0 & -0.279077 \end{bmatrix}$$

We now successfully exchanged the two blocks in the lower right corner. It is not difficult to complete the whole operation following this procedure. When all the exchanges are completed, the unary transformations can applied to make standard Schur form.

Since $\sup(p+q) = 4$ is a constant, the time complexity of this process is $O(n)$, which is about $4n$ floating point operations per column at most. Since the time complexity of bubble sort is $O(n^2)$, the complexity of the entire sorting process is $O(n^3)$. Finally, we have the algorithm for reordering the entire Schur decomposition:

1. **Input: Schur form** A**; Schur vector** U**;**
   Eigenvalue $\{\lambda_i\}$**; Sort by** $\{key_i\}$
2. **Output: Sorted eigenvalues** $\{\lambda_i\}$**,**
   Schur form A **and Schur vector** U
3. **For** $k = 2, \dots, n$ **Do**
4. **For** $i = n, \dots, k$ **Do**
5. $q \leftarrow \left|\operatorname{sign} a_{i,i-1}\right| + 1$
6. $p \leftarrow \left|\operatorname{sign} a_{i-q,i+q-1}\right| + 1$
7. **If** $key_i \geq key_{i-q}$ **Then Skip** i **To** $i-q$
8. **Exchange** $\{key_i, key_{i+q-1}\}$ **and** $\{key_{i-1}, key_{i-p}\}$
   Exchange $\{\lambda_i, \lambda_{i+q-1}\}$ **and** $\{\lambda_{i-1}, \lambda_{i-p}\}$
9. $\begin{bmatrix} T_{1,1} & T_{1,2} \\ 0 & T_{2,2} \end{bmatrix} \leftarrow A_{i-p:i+q-1,i-p:i+q-1}$
10. $X \leftarrow$ **Solve** $T_{1,1}X - XT_{2,2} = T_{1,2}$ **by equation(0.4)~(0.7)**
11. $Q \xleftarrow{\text{Complete QR}} \begin{bmatrix} -X \\ I_q \end{bmatrix}$
12. $A_{i-p:i+q-1,:} \leftarrow Q^T A_{i-p:i+q-1,:}$
13. $A_{:,i-p:i+q-1} \leftarrow A_{:,i-p:i+q-1} Q$
14. $U_{:,i-p:i+q-1} \leftarrow U_{:,i-p:i+q-1} Q$
15. **Skip** i**To**$i-p$
16. **End For**
17. **If** $\left|a_{k,k-1}\right| < \varepsilon(\left|a_{k-1,k-1}\right| + \left|a_{k,k}\right|)$ **Then**
18. **Transform** $A_{k-1:k,k-1:k}$ **to standard Schur form similarly to Algorithm 3**
19. **Skip** k **To** $k+1$
20. **End If**
21. **End For**

Exchange the blocks of i and $i-1$

Transforms the elements just ordered by bubble sort to standard Schur form

Algorithm 4 Reorder of Schur decomposition

Section 5 Eigen Decomposition of Hessenberg Matrices

In the previous section, we obtained the Schur decomposition of the Hessenberg matrix. During the decomposition, we have obtained the eigenvalues

of the matrix A, the corresponding Schur form T and Schur vectors U. The easiest way to calculate the eigenvectors is to solve a series of linear equations of upper triangular matrices, that is,

$$(T - \lambda_i I)\vec{u} = \vec{0}$$

Using back-substitution algorithm, the time complexity of a single eigenvector is $\Theta(n^2)$, and all eigenvectors can be obtained by a total of $\Theta(n^3)$ floating point calculations. However, this method has certain limitations: (1) if λ_i is a complex number, it will introduce undesired complex arithmetic; (2) if λ_i has multiplicity, the simple back-substitution algorithm cannot find the basis of the corresponding eigenspace of λ_i. To resolve these problems, we need to make some changes.

First recall the method for computing the basis of null space of a matrix. For a square matrix $A \in \mathbb{F}^{n\times n}$, after row reduction, it becomes

$$\ker A = \ker T \equiv \ker \begin{bmatrix} d_1 & t_{1,2} & \cdots & t_{1,n} \\ 0 & d_2 & \cdots & t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

where $d_i = 0$ or 1 and when $d_i = 0$, its column are all 0. The null space is the solution set of the homogeneous linear equation system $T\vec{x} = \vec{0}$. Let $\mathcal{P} = \{i | d_i = 0\}$ be free variables. Take $x_{i\in\mathcal{P}}\vec{z} \coloneqq \{x_{i\in\mathcal{P}}\} = \vec{e}_1, \ldots, \vec{e}_{n-r(A)}$, then, we only need to solve $n - r(A)$ linear systems to get the basis of null space.

For example, the Schur form

$$A = \begin{bmatrix} 0.815373 & 1.42317 & -0.253261 & 0.0073923 & 0.599984 & -0.829227 \\ -0.373364 & 0.815373 & -0.153782 & 0.263274 & -0.144811 & 0.0478312 \\ 0 & 0 & -0.88416 & 1.18541 & -0.0390394 & 0.717003 \\ 0 & 0 & -1.57779 & -0.88416 & -0.919224 & 0.668787 \\ 0 & 0 & 0 & 0 & -0.806528 & -0.0750158 \\ 0 & 0 & 0 & 0 & 0 & -0.806528 \end{bmatrix}$$

shows that its eigenvalue -0.806528 is twofold. Perform forward row reduction to $A + 0.806528I$, we have

$$A + 0.806528I \to \begin{bmatrix} 1 & 0.877469 & -0.156151 & 0.0045578 & 0.369926 & -0.511268 \\ 0 & 1 & -0.108788 & 0.135918 & -0.00343388 & -0.0733813 \\ 0 & 0 & 1 & -15.2696 & 0.502879 & -9.23596 \\ 0 & 0 & 0 & 1 & 0.00520423 & 0.575246 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It can be seen that its rank is $6 - 1 = 5$, so the linear equation system $(A + 0.806528I)\vec{x} = \vec{0}$ has exactly one solution, and the free variable is x_5. Set it to 1, we have

$$\vec{x} = [-0.408881, -0.0592108, -0.582346, -0.00520423, 1, 0]^T$$

It is not difficult to verify that

$$A\vec{x} = -0.806528\vec{x}$$

Furthermore, we don't actually have to solve these equations: because the submatrix formed by the columns of $d_{i\in\mathcal{P}}$ and preceding $r(A)$ rows (denoted as $\{\vec{b}_j\}$) and the identity matrix naturally form a set of basis of the null space:

$$\ker A = \operatorname{span}\{[\vec{b}_j^T, -\vec{e}_j^T]\}$$

One can simply normalize and orthogonalize them to obtain the desired eigenvector(s).

For the matrix of the above example, we only need to backward row reduced $A+0.806528I$:

$$A+0.806528I \to \begin{bmatrix} 1 & 0 & 0 & 0 & 0.408881 & 0 \\ 0 & 1 & 0 & 0 & 0.0592108 & 0 \\ 0 & 0 & 1 & 0 & 0.582346 & 0 \\ 0 & 0 & 0 & 1 & 0.00520423 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Comparing the solution to be one above, it is easy to find that $\vec{x}$ is the combination of the column with zero diagonal element and the identity matrix, i.e.,

$$\ker(A-\lambda I) = \operatorname{span}\{\,[\vec{b}_j^T, -\vec{e}_j^T]\,\}$$

Now, for real eigenvalues, one only needs to know how to reduce Schur form into row echelon form to solve problem (2). After Algorithm 3 and Algorithm 4 the diagonal elements of Schur form A_s with the same eigenvalue must be adjacent, therefore, after forward row reduction,

$$A_s-\lambda_i I \to \begin{bmatrix} 1 & t_{1,2} & \cdots & t_{1,p-1} & t_{1,p} & \cdots & t_{1,q} & \cdots \\ & 1 & \cdots & t_{2,p-1} & t_{2,p} & \cdots & t_{2,q} & \cdots \\ & & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & & 1 & t_{p-1,p} & \cdots & t_{p-1,q} & \cdots \\ & & & & 0 & \ddots & \vdots & \vdots \\ & & & & & \ddots & t_{q-1,q} & \cdots \\ & & & & & & 0 & \cdots \\ & & & & & & & \ddots \end{bmatrix} \tag{0.8}$$

It is not difficult to find that the submatrix

$$\begin{bmatrix} 0 & t_{p,p+1} & \cdots & t_{p,q} \\ & 0 & \ddots & \vdots \\ & & \ddots & t_{q-1,q} \\ & & & 0 \end{bmatrix}$$

is not affected by the row echelon form of the preceding and exceeding rows. Therefore, if and only if there are l columns of the matrix are all 0, the dimension of the corresponding eigen space is l. Obviously, $1 \le l \le q-p+1$. Therefore,

$$[\vec{b}_1, \dots] = \begin{bmatrix} t_{1,p} & \cdots & t_{1,q} \\ \vdots & \ddots & \vdots \\ t_{p,p} & \cdots & t_{p,q} \end{bmatrix}$$

Compared to A_s, it is only affected by the preceding 2×2 diagonal block row reduction and diagonal scaling. Thus, after backward row reduction, the corresponding l columns $\vec{b}_l$ that are not all 0 will be finally reduced to 0, and the remaining columns will be backward reduced in turn. I won't go into details here since it is easier to learn through my codes yourself.

As for problem (1), we need to occupy additional space to process complex numbers, which will not be discussed complicatedly for the same reason as

above. The concrete algorithm is then:

Input: Schur form A; eigenvalues $\{\lambda_i\}$; Schur vectors U
Output: Eigenvectors of A left multiply U stored in V
For $k = 1, \dots, n$ **Do**
  $l \leftarrow \max\{l | \lambda_l = \lambda_k \text{ or } \lambda_l = \overline{\lambda_k}\}$
  $V_{:,k:l} \leftarrow A_{:,k:l}$
  $V_{k:n,k:l} \leftarrow 0$

Set the values beyond the range to zero.

  $\vec{\beta} \leftarrow -\operatorname{diag}_{-1} A_{0:k-1,0:k-1} / (\operatorname{diag} A_{0:k-2,0:k-2} - \lambda_k)$
  $\vec{\alpha} \leftarrow 1/\left(\operatorname{diag} A_{1:k-1,1:k-1} + \vec{\beta} \odot \operatorname{diag}_1 A_{0:k-1,0:k-1} - \lambda_k\right)$
  $\vec{\beta} \leftarrow \vec{\alpha} \odot \vec{\beta}$ /*Multiply element by element*/

Real eigenvalues

  If $\operatorname{Im} \lambda_k = 0$ **Then**
    $\mathcal{J} \leftarrow \left\{i \middle| \vec{a}_{k:i-1,i} = \vec{0}, k < i \le l\right\} \cup \{k\}$ /*Find the 0 columns*/

Complex conjugate pair: Now, $\vec{\alpha}$, $\vec{\beta}$, T are all complex.

  Else
    $\mathcal{J} \leftarrow \left\{i \middle| \vec{a}_{k:i-2,i} = \vec{0}, k+1 < i \le l, i - k \bmod 2 \equiv 0\right\} \cup \{k\}$
    /*Find the column with all 0*/
    /*In practice, the matrix $V_{:,k:l}$ will always be real: the real parts are stored in $V_{:,i}$ and imaginary parts are stored in $V_{:,i+1}$, which requires special processing in the following lines*/
  End If

row reduction

  $V_{1:k-1,\mathcal{J}} \leftarrow \operatorname{diag} \vec{\beta} \cdot V_{0:k-2,\mathcal{J}} + \operatorname{diag} \vec{\alpha} \cdot V_{1:k-1,\mathcal{J}}$ /*Convert to (0.6)*/
  For $j = k-2, \dots, 1$ **Do**
    $V_{1:j,\mathcal{J}} \leftarrow V_{1:j,\mathcal{J}} - \left(\vec{\beta}_{1:j} \odot \vec{a}_{0:j-1,j+1} + \vec{\alpha}_{1:j} \odot \vec{a}_{1:j,j+1}\right) \otimes \vec{v}_{k-1,\mathcal{J}}$
  End For /*Reduced to the row echelon from*/

get feature vector

  $V_{k:n,\mathcal{J}} \leftarrow -[\vec{e}_1, \dots, \vec{e}_{|\mathcal{J}|}]$
  $V_{:,\mathcal{J}} \leftarrow \textbf{Orthogonalize}(V_{:,\mathcal{J}})$
  $V_{:,\bar{\mathcal{J}}} \leftarrow 0$ /* $\bar{\mathcal{J}} = \{i | i \notin \mathcal{J}, k \le i \le l\}$*/
  $V_{:,\mathcal{J}} \leftarrow U \cdot V_{:,\mathcal{J}}$
  /*Left multiply by Schur vectors to get eigenvectors*/
  Skip k **to** $l+1$
End For

Algorithm 1 Solving Eigenvectors in Schur Form Using Row Reduction

In this algorithm, $\vec{\beta}$ represents the coefficient k of the elementary row transformation matrix $E_{i,i+1}(k)$ required to eliminate 1-diagonal elements in the forward row reduction. $\vec{\alpha}$ represents the coefficient k of the elementary row transformation $E_i(k)$ required to scale the diagonal elements to 1. The operations $\vec{\beta} \leftarrow \vec{\alpha} \odot \vec{\beta}$ is used to simplify

$$V_{1:k-1,\mathcal{J}} \leftarrow \operatorname{diag} \vec{\beta} \cdot \operatorname{diag} \vec{\alpha} \cdot V_{0:k-2,\mathcal{J}} + \operatorname{diag} \vec{\alpha} \cdot V_{1:k-1,\mathcal{J}}$$

which is, in practice, implemented by element-wise multiplication of vectors.

Section 6 Other Proofs

6.1 Rayleigh Quotient

$$\rho(\vec{x}, A) := \frac{\vec{x}^* A \vec{x}}{\vec{x}^* \vec{x}}$$

is called the Rayleigh quotient of matrix $A \in \mathbb{F}^{m\times m}$ to vector $\vec{x}$.

Since $\rho(\alpha\vec{x}, A) = \rho(\vec{x}, A)$, we can only study unit vector $\vec{x}$. Now, the Rayleigh quotient becomes

$$\rho(\vec{x}, A) = \vec{x}^* A \vec{x} \equiv \|\vec{x}\|_A$$

the norm of vector of $\vec{x}$ in the space spanned by the column vectors of A.

For a Hermitian matrix A, the extremes of $\rho(\vec{x}, A)$ are the largest and smallest eigenvalues. The proof is as follow: Let the eigen decomposition of A be $A = V\Lambda V^*$, then we have

$$\rho(\vec{x}, A) = \vec{x}^* A \vec{x} = \vec{x}^* V \Lambda V^* \vec{x} = \sum_{i=1}^{m} \lambda_i |\vec{u}_i^* \vec{x}|^2$$

Now we perform up scaling. Assuming that the eigenvalues has been sorted, that is, $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Then,

$$\lambda_1 \sum_{i=1}^{m} |\vec{u}_i^* \vec{x}|^2 = \lambda_1 \le \sum_{i=1}^{m} \lambda_i |\vec{u}_i^* \vec{x}|^2 \le \lambda_m \sum_{i=1}^{m} |\vec{u}_i^* \vec{x}|^2 = \lambda_n$$

Hence,

$$\lambda_1 \le \rho(\vec{x}, A) \le \lambda_m, \qquad \rho(\vec{u}_i, A) = \lambda_i$$

Therefore,

$$\min_{\vec{x}} \rho(\vec{x}, A) = \lambda_1, \max_{\vec{x}} \rho(\vec{x}, A) = \lambda_m$$

Q.E.D.■

Next, let me prove the monotonicity theorem that will be used in the main text: if a Hermitian matrix $A \in \mathbb{F}^{m\times m}$ and $\vec{q}_1, \ldots, \vec{q}_n$ is a set of unit orthonormal bases. Let $Q := [\vec{q}_1, \ldots, \vec{q}_n]$, then $A' := Q^* A Q \in \mathbb{F}^{n\times n}$. Now, the eigenvalues $\lambda'_1 \le \cdots \le \lambda'_n$ of A' have relations

$$\lambda'_i \le \lambda_i, \qquad \forall 1 \le i \le n$$

Prove: Let $\vec{w}_1, \ldots, \vec{w}_n \in \mathbb{F}^n$ be the eigenvectors of A', i.e., $A' = W\Lambda' W^*$. Therefore, $Q\vec{w}_1, \ldots, Q\vec{w}_k$ (with k being the index of interest) are mutually orthogonal. Then, we can use them as basis to construct a unit vector $\vec{x}_0$:

$$\vec{x}_0 := \sum_{i=1}^{k} a_i Q \vec{w}_i = Q \sum_{i=1}^{k} a_i \vec{w}_i := Q\vec{a}$$

and make sure that it is perpendicular to all the eigenvectors $\vec{u}_i$ of A whose indexes are less than k:

$$\langle \vec{x}_0, \vec{u}_i \rangle = 0, \qquad \forall 1 \le i \le k-1$$

Since λ_k is the k^{th} smallest eigenvalue, according to the previous theorem, λ_k is the smallest Rayleigh quotient in the space other than the one composed of the previous $k-1$ eigenvectors. This is

$$\lambda_k = \min_{\vec{x}, \langle \vec{x}_0, \vec{u}_i \rangle = 0, \forall i < k} \rho(\vec{x}, A)$$

Thus,

$$\lambda_k \le \rho(\vec{x}_0, A) = \vec{x}_0^* A \vec{x}_0 = \vec{a}^*(Q^* A Q)\vec{a} = \vec{a}^* A' \vec{a} = \vec{a}^* W \Lambda' W^* \vec{a} = \sum_{i=1}^{m} \lambda_i' |\vec{w}_i^* \vec{a}|^2$$
$$= \sum_{i=1}^{k} \lambda_i' |a_i|^2 \le \lambda_k'$$

Q.E.D.■

6.2 Multiplicity of Real Symmetric Matrices

Suppose a symmetric tridiagonal matrix

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{R}^{n\times n}$$

Then $A - \lambda I$ (with λ being a certain eigenvalue of A) becomes

$$\begin{bmatrix} \beta_2 & \alpha_2 - \lambda & \beta_3 & & \\ & \beta_3 & \alpha_3 - \lambda & \ddots & \\ & & \ddots & \ddots & \beta_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} - \lambda \\ & & & & \beta_{n-1} \end{bmatrix}$$

after deleting the first row and the last column. It is an upper triangular matrix, and the diagonal elements are not all 0, so it is of full rank $(n-1)$. Thus $\operatorname{rank}(A - \lambda I) \ge n - 1$. So, the multiplicity of root 0 of $\det(A - \lambda I)$ is at most 1 (because the rank of the matrix is equal to the characteristic polynomial's order minus the multiplicity of 0). Therefore, the multiplicity of all eigenvalues of A is 1. Q.E.D.■

Similarly, a banded symmetric matrix whose bandwidth is $2\nu + 1$ and off-diagonal elements are not 0

$$T = \begin{bmatrix} \mathrm{A}_1 & \mathrm{B}_1^T & & & \\ \mathrm{B}_1 & \mathrm{A}_2 & \mathrm{B}_2^T & & \\ & \mathrm{B}_2 & \mathrm{A}_3 & \ddots & \\ & & \ddots & \ddots & \mathrm{B}_{j-1}^T \\ & & & \mathrm{B}_{j-1} & \mathrm{A}_j \end{bmatrix} \in \mathbb{R}^{n\times n}$$

where $\mathrm{A}^* = \mathrm{A}^T = \mathrm{A} \in \mathbb{R}^{\nu\times\nu}$ and $\mathrm{B} \in \mathbb{R}^{\nu\times\nu}$ is an upper triangular matrix. We only need to delete the first ν rows and the last ν columns of $\hat{T} - \lambda I$ to get an upper triangular matrix whose diagonal elements are not 0:

$$\begin{bmatrix} \mathrm{B}_2 & \mathrm{A}_2 - \lambda I & \mathrm{B}_3^T & & \\ & \mathrm{B}_3 & \mathrm{A}_3 - \lambda I & \ddots & \\ & & \ddots & \ddots & \mathrm{B}_{j-2}^T \\ & & & \mathrm{B}_{j-2} & \mathrm{A}_{j-1} - \lambda I \\ & & & & \mathrm{B}_{j-1} \end{bmatrix}$$

We have $\operatorname{rank}\big(\hat{T} - \lambda I\big) \ge (j-1)\nu = n - \nu$, therefore, the multiplicity of all eigenvalues of T is at most ν. Q.E.D.■