1 Computing eigenvalues with the QR method

Consider the algebraic eigenvalue problem

$$\text{Find } \lambda \text{ and } x \neq 0 \text{ such that } Ax = \lambda x. \tag{1}$$

If A is triangular (or even diagonal), the eigenvalues are readily available as they are the diagonal entries of A.

In the more general case the situation is trickier as the numerical solution of an algebraic eigenvalue problem of size five or greater is, unlike numerical solution of a system of liner equations, always an iterative process. Namely, the polynomial

$$p(\xi) = (-1)^n(\xi^n - \alpha_{n-1}\xi^{n-1} - ... - \alpha_0)$$

is the characteristic polynomial of the matrix

$$C_p = \begin{bmatrix} \alpha_{n-1} & \alpha_{n-2} & \dots & \alpha_1 & \alpha_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

If there would be a "direct" method to compute eigenvalues of the matrix C_p, then we could (in exact arithmetic) compute the roots of an arbitrary polynomial in closed form!

1.1 QR method of Francis (and Kublanovskaya)

The most popular method to compute all eigenvalues of a dense matrix is the QR method. The basic idea is very simple. Let us define the sequence $\{A_k\}_{k=0}^{\infty}$ by

$$\begin{cases} A_0 := A \\ A_k =: Q_k R_k, \quad A_{k+1} := R_k Q_k, \quad k = 0, 1, ... \end{cases}$$

Theorem 1. *The matrix A_k is unitarily similar to the matrix A_0 for all k.*

Proof:

$$\begin{aligned} A_{k+1} &= R_k Q_k = {Q_k}^H A_k Q_k \\ &= {Q_k}^H {Q_{k-1}}^H A_{k-1} Q_{k-1} Q_k = \dots = {Q_k}^H \dots {Q_0}^H A_0 Q_0 \dots Q_k. \end{aligned}$$

□

Example 1. Consider applying the QR iteration to the matrix

$$A = \begin{bmatrix} 4 & -2 & -1 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{bmatrix}.$$

$$A_1 = \begin{bmatrix} 5.6667 & -0.5774 & -0.4714 \\ 0.0000 & 6.0000 & 0.0000 \\ 0.2357 & 0.4082 & 0.3333 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 5.6477 & -0.6082 & 0.6505 \\ 0.0011 & 6.0018 & 0.4067 \\ -0.0146 & -0.0252 & 0.3505 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 5.6459 & -0.6110 & -0.6615 \\ 0.0001 & 6.0001 & -0.4318 \\ 0.0009 & 0.0016 & 0.3540 \end{bmatrix}, \qquad A_4 \approx \begin{bmatrix} 5.6458 & -0.6112 & 0.6622 \\ 0 & 6.0000 & 0.4333 \\ 0 & 0 & 0.3542 \end{bmatrix}.$$

During the QR iteration the lower triangular entries of matrices A_k tend to zero and the vector of diagonal entries tends to the vector

$$(5.6458,\ 6.0000,\ 0.3542)^T.$$

The eigenvalues of an upper triangular matrix are its diagonal entries. As A_4 is similar to A, then $a_{ii}^{(4)} \approx \lambda_i$. In what follows we prove this is in a more general setting.

Theorem 2. *A QR factorization of A_0^k is $P_k U_k$, where $P_k := Q_0 \dots Q_k$ and $U_k := R_k \dots R_0$.*

Proof: Using Thorem 1 we get

$$\begin{aligned} P_k U_k &= (Q_0 \dots Q_k)(R_k \dots R_0) = (Q_0 \dots Q_{k-1}) A_k (R_{k-1} \dots R_0) \\ &= A_0 (Q_0 \dots Q_{k-1})(R_{k-1} \dots R_0) = A_0 (Q_0 \dots Q_{k-2}) A_{k-1} (R_{k-2} \dots R_0) \\ &= A_0^2 (Q_0 \dots Q_{k-2})(R_{k-2} \dots R_0) = \dots = A_0^k. \end{aligned}$$

□

In the sequel, we assume (for simplicity) that the eigenvalues of A satisfy $0 < |\lambda_n| < \dots < |\lambda_1|$. Then A is invertible and diagonalisable. Denote

$$D = \operatorname{diag}(\lambda_1, ..., \lambda_n).$$

Theorem 3. *Let $A = XDX^{-1} =: XDY$ and let Y have an LU factorization $Y = LU$. Then in the QR iteration matrices Q_k and R_k satisfy*

$$\begin{aligned} &S_{k-1}^H Q_k S_k \to I \\ &{S_k}^H R_k S_{k-1} \to T \in Upp(n), \end{aligned}$$

where $S_k \in Unit(n) \cap Diag(n) \quad \forall k \geq 1$.

Proof: is messy ;-) □

Theorem 4. *In the QR iteration matrices A_k satisfy*

$$\lim_{k \to \infty} a_{ij}^{(k)} = \begin{cases} \lambda_i, & \text{if } i = j \\ 0, & \text{if } i > j. \end{cases}$$

Proof: It holds

$$\hat{A}_k := S_k^H A_{k+1} S_k = S_k^H R_k Q_k S_k = S_k^H R_k S_{k-1} S_{k-1}^H Q_k S_k \to TI = T, \quad \text{as } k \to \infty,$$

where matrices S_k are as above. Now we have

$$\hat{a}_{ij}^{(k)} = \bar{s}_{ii}^{(k)} a_{ij}^{(k+1)} s_{jj}^{(k)} = \begin{cases} a_{ii}^{(k+1)}, & \text{if } i = j, \\ \bar{s}_{ii} a_{ij}^{(k+1)} s_{jj}, & \text{if } i \neq j. \end{cases}$$

The claim follows by letting $k \to \infty$. □

1.2 Practical implentation of the QR method

Each iteration of the QR method involves computing the QR factorization which is quite expensive for a full matrix. Thus, in practise, the QR iteration is never applied to the original full matrix. Instead, the matrix can be cheaply transformed into "almost triangular" form, namely the Hessenberg form, i.e. $a_{ij} = 0$ whenever $i > j + 1$. The Hessenberg form of a symmetric matrix is a tridiagonal matrix. Applying one QR iteration to a Hessenberg matrix is cheap. On the other hand, the QR iteration maintains the Hessenberg form.

1.2.1 Transformation into Hessenberg form

A matrix A can be transformed into similar Hessenberg matrix by multiplications from left and right by a suitable Householder transformations. First we multiply from left by the matrix

$$P_1 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{P}_1 \\ & \end{array}\right],$$

which nullifies the first column starting from third row. Then we multiply from right with the same matrix (note that $P_1^H = P_1$).

$$P_1AP_1 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{P}_1 \end{array}\right] \left[\begin{array}{c|c} a_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{P}_1 \end{array}\right]$$

$$= \left[\begin{array}{c|c} a_{11} & A_{12} \\ \hline \hat{P}_1A_{21} & \hat{P}_1A_{22} \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{P}_1 \end{array}\right]$$

$$= \left[\begin{array}{c|c} a_{11} & A_{12}\hat{P}_1 \\ \hline \hat{P}_1A_{21} & \hat{P}_1A_{22}\hat{P}_1 \end{array}\right] = \left[\begin{array}{c|c} a_{11} & A_{12}\hat{P}_1 \\ \hline \begin{array}{c} \beta \\ 0 \\ \vdots \\ 0 \end{array} & \hat{P}_1A_{22}\hat{P}_1 \end{array}\right] \tag{2}$$

When we multiply from right by P_1, the first column remains untouched so that its zero part remains zero. We continue this process by multiplying by P_2 to nullify the second column starting from fourthe row etc. Finally the matrix PAP^H, $P = P_1P_2 ... P_{n-2} = P^H$ is in Hessenberg form. The vectors v needed to construct matrices P_i can be partially stored into the nullified parts of A.

1.2.2 QR iteration for a Hessenberg matrix (including a shift)

Let H be a Hessenberg matrix obtained from the original matrix A. In what follows, the role of the set of scalar parameters $\{\eta_i\}$ (shifts) is to accelerate the convergence.

0. Set $H_0 := H$ and $i := 1$.

1. Set $\tilde{H} := H_{i-1} - \eta_i I$.

2. Find $Q_i \in Unit(n)$ and $R_i \in Upp(n)$ such that
$$\tilde{H} = Q_iR_i.$$

3. Set
$$H_i := R_iQ_i + \eta_i I.$$

4. If H_i is an upper triangular matrix (up to the precision specified) then stop. Otherwise set $i := i + 1$ and goto step 1.

As $R_i = Q_i^H(H_{i-1} - \eta_i I)$, then

$$H_i = R_iQ_i + \eta_i I = Q_i^H H_{i-1} Q_i - \eta_i Q_i^H Q_i + \eta_i I = Q_i^H H_{i-1} Q_i.$$

Thus, the next matrix H is similar to the previous one, so they have the same eigenvalues.

Consider now the QR factorization of a Hessenberg matrix H. We should find unitary Q and upper triangular R such that $H = QR$ or $Q^H H = R$. Define a Givens rotation matrix $J(1,2,\theta_1)$ such that the entry (2,1) of the matrix $J(1,2,\theta_1)H$ is zero. Then we multiply by matrix $J(2,3,\theta_2)$ such that the entry (3,2) becomes zero, etc. Thus

$$J(n-1,n,\theta_{n-1})\dots J(2,3,\theta_2)\,J(1,2,\theta_1)\,H = R \in Upp(n).$$

Denote $Q^H = J(n-1,n,\theta_{n-1})\dots J(1,2,\theta_1)$. Then $Q^H H = R$ from which we obtain $H = QR$.

The shifted QR for a Hessenberg matrix gets now the following form (H contains the latest H_i):

1. Set $H := H - \eta_i I$.
2. $H := J(n-1,n,\theta_{n-1})\dots J(1,2,\theta_1)H$.
3. $H := HJ(1,2,\theta_1)^H \dots J(n-1,n,\theta_{n-1})^H$.
4. $H := H + \eta_i I$.
5. If H is an upper triangular matrix (up to the precision specified) then stop. Otherwise set $i := i+1$ and goto step 1.

Note that it would be tempting to perform the multiplications from right immediately on step 2. Doing this, however, would not maintain the similarity!

Theorem 5. *If H_{i-1} ia an Hessenberg matrix, then H_i is too.*

Proof: (Exercise). □

If, during the QR iteration, some lower diagonal entry becomes $h_{i+1,i} \approx 0$, then H is approximately a block diagonal matrix

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}.$$

The eigenvalue problem then splits into two smaller eigenvalue problems involving matrices H_{11} and H_{22}.

1.2.3 Choosing the shifts

The aim is to make $h_{n,n-1}$ close to zero. When it is close zero (up to precision scpecified) we can apply the QR iteration to smaller matrix excluding the last row and column of the current H. This process (deflation) further accelerates the convergence.

A good choice for the shift is $\eta = h_{mm}$. Before the application of the last Givens transformation, the matrix $H - \eta I$ is of the form

$$H - \eta I = \begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & & x & y \\ & & & \epsilon & 0 \end{bmatrix}.$$

The last Givens matrix is of the form

$$J(m-1, m, \theta_{m-1}) = \begin{bmatrix} c & \bar{s} \\ -s & c \end{bmatrix}, \quad c \in \mathbb{R},\ s \in \mathbb{C},\ c = |x|/\sqrt{|x|^2 + |\epsilon|^2},\ s = c\epsilon/x.$$

Now

$$\begin{bmatrix} c & \bar{s} \\ -s & c \end{bmatrix} \begin{bmatrix} x & y \\ \epsilon & 0 \end{bmatrix} = \begin{bmatrix} \dots & \dots \\ 0 & -c\epsilon y/x \end{bmatrix}.$$

When the last multiplication from the right has been done, we get

$$\begin{bmatrix} \dots & \dots \\ 0 & -\epsilon y c/x \end{bmatrix} \begin{bmatrix} c & -\bar{s} \\ s & c \end{bmatrix} = \begin{bmatrix} \dots & \dots \\ -\epsilon^2 c^2 y/x^2 & \dots \end{bmatrix}.$$

It holds

$$|h_{m,m-1}| = |\epsilon^2 c^2 y/x^2| = |\epsilon|^2 |y|/(|x|^2 + |\epsilon|^2).$$

Thus if $|\epsilon| \ll |x|$ then $|h_{m,m-1}|$ dimishes quadratically.

1.2.4 QR algorithm for a tridiagonal matrix

If A is symmetric, then the transformation into Hessenberg form actually produces a tridiagonal matrix. For a tridiagonal matrix the QR iteration can be done in a very efficient way.

We can apply the previous nonsymmetric QR method to the tridiagonal matrix

$$T = \begin{bmatrix} a_1 & b_2 & & & \\ b_2 & a_2 & b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & b_n \\ & & & b_n & a_n \end{bmatrix},$$

i.e. we form orthogonal matrices Q_1, Q_2, ..., Q_k such that

$$(Q_1 \dots Q_k)^T T (Q_1 \dots Q_k) \approx D,$$

where D is a diagonal matrix containing the approximate eigenvalues.

1.2.5 Implicitly shifted QR iteration

It is possible to derive a variant of the QR iteration for a tridiagonal matrix where matrices $Q_1, ..., Q_k$ need not to be stored, i.e. the multiplication from right is done immediately. This variant is based on "Implicit Q Theorem" [Golub & van Loan]. The theorem essentially says the following: If the first columns of orthogonal matrices U are V the same, and $H = U^TAU$ and $G = V^TAV$ are Hessenberg matrices. Then H and G are "essentially same", i.e. $G = E^{-1}HE$, where $E = \text{diag}(\pm 1, \pm 1, ..., \pm 1)$. Note that in any case, there is always some freedon in the QR iteration as the QR factorization is not unique.

Let η be the shift. In the previously shown QR algorithm the shift was first substracted from the diagonal entries. After that we multiply from left using Givens rotations $J_1, ..., J_{n-1}$ to obtain upper triangular matrix. After that we multiply from right with matrices $J_1^T, ..., J_{n-1}^T$ resulting once again a tridiagonal matrix. After that η is added to the diagonal entries. Thus the initial matrix T was replaced by U^TTU, where $U = J_1^TJ_2^T \dots J_{n-1}^T$ is orthogonal. It is easy to see that the first row of the matrix $U^T = J_1J_2 \dots J_{n-1}$ is the same as the first row of the matrix J_1.

When one uses *implicitly shifted QR algorithm for a tridiagonal matrix*, the matrix T (not $T - \eta I$) is multiplied from left by J_1:lla and *immediately* from right byJ_1^T. The matrix $J_1TJ_1^T$ is of the form

$$J_1TJ_1^T = \begin{bmatrix} * & * & \boxed{*} & & \\ * & * & * & & \\ \boxed{*} & * & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix},$$

as multiplication by J_1 replaces rows 1 and 2 of T by their linear combinations. Multiplication by J_1^T replaces columns by their linear combinations. As a result there are "extra" nonzero entries at positions (3,1) and (1,3). One should get rid of these. Form Givens rotation $\tilde{J}_2(2,3,\theta_2)$ such that the entry at (3,1) becomes zero. Due to symmetry the entry at (1,3) becomes zero when we multiply from right by $\tilde{J}_2^T$:lla. However, now we have again "extra" nonzeros at positions (4,2) and (2,4). The aim is now to "chase" these extra entries "out" of the matrix by continuing the multiplication with Givens matrices $\tilde{J}_3, ...$

$$\tilde{J}_2J_1TJ_1^T\tilde{J}_2^T = \begin{bmatrix} * & * & & & \\ * & * & * & \boxed{*} & \\ & * & * & * & \\ & \boxed{*} & * & * & * \\ & & & * & * \end{bmatrix} \tag{3}$$

$$\tilde{J}_3\tilde{J}_2J_1TJ_1^T\tilde{J}_2^T\tilde{J}_3^T = \begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & * & \boxed{*} \\ & & * & * & * \\ & & \boxed{*} & * & * \end{bmatrix} \tag{4}$$

Finally we have again tridiagonal matrix V^TTV. The first rows of matrices V^T and J_1^T are the same. Therefore the first rows of V^T and U^T are the same, and thus the first columns of V and U are equal. The Impcit Q Theorem now says that the matrices V^TTV and are essentially same U^TTU.

The advantage of the implicitly shifted variant is that the Givens transformations need not to be stored.

Remark 1. For a symmetric matrix, an efficient shift strategy due to Wilkinson can also be used. Instead of the last diagonal entry a_n, the eigenvalue of 2×2 matrix

$$\begin{bmatrix} a_{n-1} & b_n \\ b_n & a_n \end{bmatrix}$$

closer to a_n is chosen as the shift. The eigenvalues are

$$\lambda = \frac{1}{2}(a_{n-1} + a_n) \pm \sqrt{\left(\frac{a_{n-1} - a_n}{2}\right)^2 + b_n^2}.$$

The one closer to a_n is

$$\lambda^* = a_n + d - \operatorname{sign}(d)\sqrt{d^2 + b_n^2}, \quad d = (a_{n-1} - a_n)/2.$$

Remark 2. The practical implementation of the implicitly shifter QR iteration can be done as follows. The tridiagonal matrix T is stored in two vectors b (co-diagonal) and a (diagonal). When we multiply from right by J_k and from left by J_k^T the following calculations are done:

$$J_k^T T J_k = \begin{array}{c} \\ \\ \\ k \\ \\ \\ \end{array}\!\!\begin{array}{c} \qquad k \\ \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & c & s & \\ & & -s & c & \\ & & & & 1 \end{pmatrix} \end{array} \begin{bmatrix} & & & & \\ & & x & \boxed{z} & \\ & x & u & v & \\ & \boxed{z} & v & w & r \\ & & & r & \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & s & c & \\ & & & & 1 \end{bmatrix}$$

After multiplication by J_k^T:

$$\begin{bmatrix} & k-1 & k & k+1 & k+2 \\ \text{row } k: & cx + sz & cu + sv & cv + sw & sr \\ \text{row } k+1: & 0 & -su + cv & -sv + cw & cr \end{bmatrix}$$

After multiplication by J_k:

$$\begin{bmatrix} & k-1 & k & k+1 & k+2 \\ \text{row } k-1: & & cx + sz & 0 & \\ \text{row } k: & cx + sz & c^2u + 2csv + s^2w & cs(w-u) + (c^2 - s^2)v & \boxed{sr} \\ \text{row } k+1: & & cs(w-u) + (c^2 - s^2)v & s^2u - 2csv + c^2w & cr \\ \text{row } k+2: & & \boxed{sr} & cr & \end{bmatrix}$$

1.3 Eigenvector computation

Up to this point, nothing was said about computation of eigenvectors. Let T be a symmetric tridiagonal matrix. During the QR iteration T is multiplied by Givens matrices until

$$Q^T AQ \approx \operatorname{diag}(\lambda_1, ..., \lambda_n),$$

where orthogonal matrix $Q = J_1 J_2 ... J_m$ is the product of aplied Givens matrices. Let $Q = [q_1, ..., q_n]$. Then $Tq_i = \lambda_i q_i$, i.e. the columns of Q are the eigenvectors of T. The matrix Q is formed during the computations by first initializing as the identity matrix and then multiplying it my right by the Givens matrices applied.

Let A be a general symmetric matrix that is transformed to tridiagonal matrix T using product of Householder transformations. Then

$$P^T AP = T,$$

where $P = P_1 ... P_{n-2}$. If (λ, x) is en eigenpair of T, then (λ, Px) is en eigenpair of A.

If the matrix A is nonsymmetric or one only needs a few eigenvectors the inverse iteration can be applied.

Theorem 6. *If (λ, x) is an eigenpair of A, then*

- *$(\lambda - \eta, x)$ is an eigenpair of $A - \eta I$*
- *(λ^{-1}, x) is an eigenpair of A^{-1} (provided that $\lambda \neq 0$)*
- *(λ^k, x) is an eigenpair of A^k (if $k < 0$ assume $\lambda \neq 0$).*

Proof: is a simple exercise ;-) □

Assume that $A \in \mathbb{R}^{n \times n}$ has n linearly independent eignvectors. Let $0 \neq v \in \mathbb{R}^n$ be arbitrary. Let η be an eigenvalue approximation computed e.g. using the QR method. The vector v can be represented by as a linear combination of the eigenvectors $x_1, ..., x_n$ of A:

$$v = \sum_{i=1}^{n} \alpha_i x_i, \qquad Ax_i = \lambda_i x_i.$$

Let $x = (A - \eta I)^{-1} v$. Then

$$x = \sum_{i=1}^{n} \alpha_i (A - \eta I)^{-1} x_i = \sum_{i=1}^{n} \alpha_i x_i (\lambda_i - \eta)^{-1}.$$

Moreoever

$$(A - \eta I)^{-k} v = \sum_{i=1}^{n} \alpha_i (A - \eta I)^{-k} x_i = \sum_{i=1}^{n} \alpha_i x_i (\lambda_i - \eta)^{-k}.$$

If $\eta \approx \lambda_m$, then for large k it holds that the vector $(A - \eta I)^{-k} v$ is in the direction x_m.

The idea above can be formulated as an algorithm that computes one eigenvector estimate.

0. Set $k := 0$ and choose arbitrary $x^{(0)} \neq 0$ and the stopping criterion $\tau > 0$. Perform LU factorization with pivoting for $A - \eta I$.
1. Solve
$$(A - \eta I)x^{(k+1)} = x^{(k)}$$
using the LU factorization.
2. Normalize $x^{(k+1)} := x^{(k+1)}/\|x^{(k+1)}\|$.
3. If $\|x^{(k+1)} - x^{(k)}\| \leq \tau$ then stop. Otherwise set $k := k + 1$ and goto step 1.

In case of zero pivot in LU factorization (due to multiple eigenvalues, or "too good" eigenvalue estimate) the simple algorithm above obviously needs some additional steps.

2 Singular value decomposition and solution of overdetermined system of linear equations

The singular value decomposition is the most general matrix decomposition. It can be used for many purposes.

Theorem 7. *Consider the matrix $A \in \mathbb{C}(m, n)$. Then there exist unitary matrices*
$$U = [u_1, u_2, ..., u_m] \in Unit(m) \quad and \quad V = [v_1, v_2, ..., v_n] \in Unit(n)$$
such that
$$U^H AV = diag(\sigma_1, \sigma_2, ..., \sigma_p), \quad p = \min\{m, n\},$$
where the numbers $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$ are the singular values *of A.*

Proof: [Golub & van Loan]. □

Theorem 8. *Let the matrix A have r nonzero singular values. Then A can be represented as follows*
$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^H.$$
This form is called the singular value decomposition, SVD *of A. Moreover*
$$\begin{aligned} \ker(A) &= \text{span}\,\{v_{r+1}, ..., v_n\}, \\ R(A) &= \text{span}\,\{u_1, ..., u_r\}, \\ \text{rank}(A) &= n - \dim(\ker(A)) = r. \end{aligned}$$

Proof: Exercise. □

2.1 Numerical computation of singular values

Singular values and eigenvalues are related as follows. Let $A \in \mathbb{C}^{m\times n}$. If $U^HAV = \mathrm{diag}(\sigma_1, ..., \sigma_n)$, then also $V^HA^HU = \mathrm{diag}(\sigma_1, ..., \sigma_n)$. By multiplying the previous equations we get

$$V^HA^HUU^HAV = [\mathrm{diag}(\sigma_1, ..., \sigma_n)]^2 = \mathrm{diag}(\sigma_1^2, ..., \sigma_n^2).$$

As $UU^H = I$, it follows that

$$V^H(A^HA)V = \mathrm{diag}(\sigma_1^2, ..., \sigma_n^2).$$

Thus the singular values are the square roots of the eigenvalues of the Hermitian matrix A^HA. This matrix is, however, never explicitly formed in practise. Instead the shifted QR method is used in an implicit way to compute the eigenvalues of A^HA.

2.1.1 Transforming a matrix into a bidiagonal form

Like in the case of an eigenvalue problem, we first tranform the matrix $A \in \mathbb{C}^{m\times n}$ into a simpler form having the same singular values. We transform the matrix A into *bidiagonal form* by multiplying it from left and from right by suitable Householder matrices.

At the end of this process we have

$$(U_n^H ... U_1^H)A(V_1 ... V_{n-2}) = \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{array}{l} n \\ m-n \end{array},$$

where

$$B = \begin{bmatrix} d_1 & f_2 & & & \\ & d_2 & f_3 & & \\ & & & \ddots & \\ & & & d_{n-1} & f_n \\ & & & & d_n \end{bmatrix}$$

is bidiagonal. All matrices $U_n, ..., U_1$ and $V_1, ..., V_{n-2}$ are unitary. Thus there exist unitary matrices U and V such that UAV is bidiagonal.

Theorem 9. *Let $B = U^HAV$, where U and V are Hermitian matrices. The singular values of B and A are the same.*

Proof: We immediately see that

$$B^HB = (V^HA^HU)U^HAV = V^HA^HAV,$$

implying that matrices B^HB and A^HA are similar. □

2.1.2 QR algorithm for singular values

Let us assume a real matrix B. In principle the shifted QR algorithm is applied to the matrix B^TB, where B is bidiagonal. Its eigenvalues are the squares of the singular values of B. One QR iteration replaces the matrix B^TB with matrix $V^T(B^TB)V$, where V is orthogonal. This is repeated until we get an approximately diagonal matrix.

In the *Golub–Kahan SVD algorithm* the matrix B^TB is not explicitly formed. The QR iteration is done working with bidiagonal matrices, quite analogously to the implicitly shifted QR method for tridiagonal matrices.

2.2 Solution of an overdetermined system of linear equations

Let $A \in \mathbb{R}^{m\times n}$, $m > n$, and $b \in \mathbb{R}^m$. Consider the system

$$Ax = b. \tag{5}$$

As there are more equations than unknowns, the system is said *overdetermined*. The system has a solution only if $b \in R(A)$. As this is not the usual case, we need to consider the least squares solution.

Example 2. Consider the following simple data fitting problem. Find coefficients $w_1, ..., w_n$ such that the function

$$f(w; x) = \sum_{j=1}^{n} w_j \varphi_j(x)$$

approximates the data $(x_1, y_1), ..., (x_m, y_m)$ in least squares sense, i.e. the coefficients solve the minimization problem

$$\min_w \left\{ F(w) := \sum_{i=1}^{m} (f(w; x_i) - y_i)^2 \right\}. \tag{6}$$

The optimality conditions of (6) imply *normal equations*:

$$\frac{\partial F(w)}{\partial w_k} = \sum_{i=1}^{m} 2\left(f(w; x_i) - y_i\right) \frac{\partial f(w; x_i)}{\partial w_k} = 0, \quad k = 1, ..., n. \tag{7}$$

Let $B \in \mathbb{R}^{m\times n}$ be a matrix with entries $b_{ij} = \varphi_j(x_i)$. Then equation (7) can be written in matrix form

$$B^TBw = B^Ty.$$

Example 3. Consider fitting the linear model

$$f(x) = w_1x_1 + w_2x_2 + ... + w_nx_n.$$

into measured data $(y_1, x_1^{(1)}, ..., x_n^{(1)}), ..., (y_m, x_1^{(m)}, ..., x_n^{(m)})$, $m > n$. For the unknown coefficients $w_1, ..., w_n$ we get the overdetermined system

$$\begin{aligned} y_1 &= w_1 x_1^{(1)} + ... + w_n x_n^{(1)} \\ &\vdots \\ y_m &= w_1 x_1^{(m)} + ... + w_n x_n^{(m)}. \end{aligned}$$

If we denote $A = [a_{ij}]$, $a_{ij} = x_j^{(i)}$ we can write it in matrix form

$$Aw = y.$$

Definition 1. The vector $\tilde{x} \in \mathbb{R}^n$ is the *least squares solution* of the system (5) if it solves the minimization problem

$$\min_x \|Ax - b\|^2. \tag{8}$$

Theorem 10. *Let $X \subset \mathbb{R}^n$ be the set of solutions to problem* (8). *Then*

$$\begin{aligned} &x \in X \quad \Leftrightarrow \quad A^T(b - Ax) = 0, \\ &X \quad \textit{is convex}, \\ &\exists \quad \textit{unique } x^* \in X \textit{ such that } \|x^*\| = \min, \\ &X = \{\tilde{x}\} \Leftrightarrow \operatorname{rank}(A) = n. \end{aligned}$$

Proof: Exercise. □

2.2.1 Solving overdetermined system using QR factorization

If $\operatorname{rank}(A) = n$, the the least squares problem (8) can be solved using a *direct method*. First, compute the QR factorization $A = QR$, where

$$R = \begin{array}{cc} \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array}.$$

Using the QR factorization the sum of squares can be written as

$$\|Ax - b\|^2 = \|Q^T(Ax - b)\|^2 = \|Rx - Q^Tb\|^2.$$

Denote

$$Q^Tb = \begin{array}{cc} \begin{bmatrix} c \\ d \end{bmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array}.$$

Now

$$\|Rx - Q^Tb\|^2 = \left\| \begin{bmatrix} \hat{R}x - c \\ d \end{bmatrix} \right\|^2 = \|\hat{R}x - c\|^2 + \|d\|^2.$$

The RHS obtains its minimum value $\|d\|^2$ when

$$\hat{R}x = c. \tag{9}$$

As $\hat{R}$ an invertible triangular matrix, equation (9) can be solved using backward substitutions.

Example 4. Consider fitting quadratic polynomial

$$p(\xi) = x_1 + x_2\xi + x_3\xi^2$$

into data (1,2), (2,2), (3,3), (3,5), (4,6) in least squares sense. Thus we need to solve overdetermined system $Ax = b$, where

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 5 \\ 6 \end{bmatrix}.$$

By computing the QR factorization of A we get

$$Q = \begin{bmatrix} -0.4472 & 0.7016 & 0.4931 & -0.0019 & -0.2540 \\ -0.4472 & -0.2631 & -0.3875 & 0.0056 & 0.7620 \\ -0.4472 & 0.1754 & -0.3522 & -0.7099 & -0.3758 \\ -0.4472 & 0.1754 & -0.3522 & 0.7043 & -0.3862 \\ -0.4472 & 0.6139 & 0.5988 & 0.0019 & 0.2540 \end{bmatrix}$$

and

$$R = \begin{bmatrix} -2.2361 & -5.8138 & -17.4413 \\ 0 & 2.2804 & 11.2263 \\ 0 & 0 & 2.1839 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Solving $\hat{R}x = c$ we get

$$x = (2.2581, -0.8387, 0.4516)^T.$$

The fitted polynomial is then

$$p(\xi) = 2.2581 - 0.8387\,\xi + 0.4516\,\xi^2.$$

2.2.2 The more general case

Even if A is of full rank, the diagonal entries of R may become very small in magnitude yielding numerical instability. A more general (and more expensive) way to solve least squares problem is by using the singular value decomposition. It is an iterative method as singular value approximations are computed by iteration.

Theorem 11. *Let A have the SVD*

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U\Sigma V^T.$$

Then the least square solution of $Ax = b$ is Va, where $a_i = (u_i^T b/\sigma_i)$, $i = 1, ..., r$, and a_i, $i = r+1, ..., n$ are arbitrary. If one wants solution with minimal $\|x\|_2$, then set $a_{r+1} = ... = a_n = 0$. The value of the sum of squares is

$$\rho = \|Ax - b\|^2 = \sum_{i=r+1}^{m} (u_i^T b)^2.$$

Proof: For all $x \in \mathbb{R}^n$ it holds

$$\rho = \|Ax - b\|^2 = \|U^T(Ax - b)\|^2 = \|U^T AV(V^T x) - U^T b\|^2,$$

as the vector length is preserved when multiplied by an orthogonal matrix. Denote $a = V^T x$ and substitute $\Sigma = U^T AV$ yielding

$$\rho = \|\Sigma a - U^T b\|^2 = \sum_{i=1}^{m} (\sigma_i a_i - u_i^T b)^2 = \sum_{i=1}^{r} (\sigma_i a_i - u_i^T b)^2 + \sum_{i=r+1}^{m} (u_i^T b)^2.$$

The minimum is obtained when $a_i = (u_i^T b)/\sigma_i,\ i = 1, ..., r.$ $\square$

Example 5. Consider again the linear system of Example 4. A SVD of A is $A = U\Sigma V^T$, where

$$U = \begin{bmatrix} 0.0607 & -0.6228 & 0.7375 & 0.0127 & -0.2537 \\ 0.2050 & -0.5679 & -0.2340 & -0.0381 & 0.7610 \\ 0.4370 & -0.2111 & -0.3454 & -0.6872 & -0.4159 \\ 0.4370 & -0.2111 & -0.3454 & 0.7253 & -0.3452 \\ 0.7566 & 0.4477 & 0.4033 & -0.0127 & 0.2537 \end{bmatrix},$$

$$\Sigma = \begin{bmatrix} 21.8132 & 0 & 0 \\ 0 & 1.7611 & 0 \\ 0 & 0 & 0.2899 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 0.0869 & -0.6617 & 0.7447 \\ 0.2805 & -0.7011 & -0.6556 \\ 0.9559 & 0.2659 & 0.1247 \end{bmatrix}$$

As $r = n$ the vectpr a reads

$$a = [0.3927, -0.7860, 2.2878]^T,$$

and the final least squares solution

$$x = Va = [2.2581, -0.8387, 0.4516]^T.$$