## 1. NUMERICAL LINEAR ALGEBRA

1.1. The LU Factorization. Recall from linear algebra that Gaussian elimination is a method for solving linear systems of the form

$$Ax = b$$
,

where  $A \in \mathbb{R}^{m \times n}$  and  $b \operatorname{Ran}(A)$ . In this method one first forms the augmented system

and then uses the three elementary row operations to put this system into row echelon form (or upper triangular form). A solution x is then obtained by back substitution, or back solving, starting with the component  $x_n$ . We now show how the process of bringing a matrix to upper triangular form can be performed by left matrix multiplication.

The key step in Gaussian elimination is to transform a vector of the form

$$\left[\begin{array}{c}a\\\alpha\\b\end{array}\right],$$

where  $a \in \mathbb{R}^k$ ,  $0 \neq \alpha \in \mathbb{R}$ , and  $b \in \mathbb{R}^{n-k-1}$ , into one of the form

 $\left[\begin{array}{c} a\\ \alpha\\ 0\end{array}\right] \ .$ 

This can be accomplished by left matrix multiplication as follows:

$$\begin{bmatrix} I_{k\times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\alpha^{-1}b & I_{(n-k-1)\times(n-k-1)} \end{bmatrix} \begin{bmatrix} a \\ \alpha \\ b \end{bmatrix} = \begin{bmatrix} a \\ \alpha \\ 0 \end{bmatrix}$$

The matrix

$$\begin{bmatrix} I_{k \times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\alpha^{-1}b & I_{(n-k-1) \times (n-k-1)} \end{bmatrix}$$

is called a Gaussian elimination matrix. This matrix is invertible with inverse

$$\begin{bmatrix} I_{k \times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \alpha^{-1}b & I_{(n-k-1) \times (n-k-1)} \end{bmatrix}$$

We now use this basic idea to show how a matrix can be put into upper triangular form. Suppose

$$A = \begin{bmatrix} a_1 & v_1^T \\ u_1 & \widetilde{A}_1 \end{bmatrix} \in \mathbb{C}^{n \times m},$$

with  $0 \neq a_1 \in \mathbb{C}$ ,  $u_1 \in \mathbb{C}^{m-1}$ ,  $v_1 \in \mathbb{C}^{n-1}$ , and  $\widetilde{A}_1 \in \mathbb{C}^{(m-1)\times(n-1)}$ . Then using the first row to zero out  $u_1$  amounts to left multiplication of the matrix A by the matrix

$$\begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_1} & I \\ 1 \end{bmatrix}$$

2

to get

(\*) 
$$\begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_1} & I \end{bmatrix} \begin{bmatrix} a_1 & v_1^T \\ u_1 & \widetilde{A}_1 \end{bmatrix} \in \mathbb{C}^{n \times m} = \begin{bmatrix} a_1 & v_1^T \\ 0 & A_1 \end{bmatrix} ,$$

where

$$A_1 = \widetilde{A}_1 - u_1 v_1^T / a_1 \; .$$

Define

$$L_1 = \begin{bmatrix} 1 & 0\\ \frac{u_1}{a_1} & I \end{bmatrix} \in \mathbb{C}^{m \times m} \quad \text{and} \quad U_1 = \begin{bmatrix} a_1 & v_1^T\\ 0 & A_1 \end{bmatrix} \in \mathbb{C}^{m \times n}$$

and observe that

$$L_1^{-1} = \left[ \begin{array}{cc} 1 & 0 \\ -\frac{u_1}{a_1} & I \end{array} \right] \ .$$

Hence (\*) becomes

$$L_1^{-1}A = U_1$$
, or equivalently,  $A = L_1U_1$ .

Note that  $L_1$  is unit lower triangular (ones on the mail diagonal) and  $U_1$  is block uppertriangular with one  $1 \times 1$  block and one  $(m-1) \times (n-1)$  block on the block diagonal. The multipliers are usually denoted

$$u/a = [\mu_{21}, \ \mu_{31}, \ \dots, \ \mu_{m1}]^T$$

If the (1, 1) entry of  $A_1$  is not 0, we can apply the same procedure to  $A_1$ : if

$$A_1 = \begin{bmatrix} a_2 & v_2^T \\ u_2 & \widetilde{A}_2 \end{bmatrix} \in \mathbb{C}^{(m-1) \times (n-1)}$$

with  $a_2 \neq 0$ , letting

$$\widetilde{L}_2 = \begin{bmatrix} I & 0\\ \frac{u_2}{a_2} & I \end{bmatrix} \in \mathbb{C}^{(m-1) \times (m-1)},$$

and forming

$$\widetilde{L}_2^{-1}A_1 = \begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_2} & I \end{bmatrix} \begin{bmatrix} a_2 & v_2^T \\ u_2 & \widetilde{A}_1 \end{bmatrix} = \begin{bmatrix} a_2 & v_2^T \\ 0 & A_2 \end{bmatrix} \equiv \widetilde{U}_2 \in \mathbb{C}^{(m-1) \times (n-1)},$$

where  $A_2 \in \mathbb{C}^{(m-2)\times(n-2)}$ . This process amounts to using the second row to zero out elements of the second column below the diagonal. Setting

$$L_2 = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{L}_2 \end{bmatrix} \quad \text{and} \quad U_2 = \begin{bmatrix} a & v^T \\ 0 & \widetilde{U}_2 \end{bmatrix},$$

we have

$$L_2^{-1}L_1^{-1}A = \begin{bmatrix} 1 & 0\\ 0 & \widetilde{L}_2^{-1} \end{bmatrix} \begin{bmatrix} a & v^T\\ 0 & A_1 \end{bmatrix} = U_2,$$

or equivalently,

$$A = L_2 L_1 U_2.$$

Here  $U_2$  is block upper triangular with two  $1 \times 1$  blocks and one  $(m-2) \times (n-2)$  block on the diagonal, and again  $L_2$  is unit lower triangular. We can continue in this fashion at most  $\tilde{m} - 1$  times, where

$$\tilde{m} = \min\{m, n\}.$$

If we can proceed  $\tilde{m} - 1$  times, then

$$L_{\tilde{m}-1}^{-1}\cdots L_2^{-1}L_1^{-1}A = U_{\tilde{m}-1} = U$$

is upper triangular provided that along the way that the (1, 1) entries of

$$A, A_1, A_2, \ldots, A_{\tilde{m}-2}$$

are nonzero so the process can continue. Define

$$L = (L_{\tilde{m}-1}^{-1} \cdots L_1^{-1})^{-1} = L_1 L_2 \cdots L_{\tilde{m}-1}.$$

The matrix L is square unit lower triangular, and so is invertable. Moreover, A = LU, where the matrix U is the so called *row echelon form* of A. In general, a matrix  $T \in \mathbb{C}^{m \times n}$  is said to be in row echelon form if for each  $i = 1, \ldots, m-1$  the first non-zero entry in the  $(i+1)^{\text{st}}$ row lies to the right of the first non-zero row in the  $i^{\text{th}}$  row.

Let us now suppose that m = n and  $A \in \mathbb{C}^{n \times n}$  is invertible. Writing A = LU as a product of a unit lower triangular matrix  $L \in \mathbb{C}^{n \times n}$  (necessarily invertible) and an upper triangular matrix  $U \in \mathbb{C}^{n \times n}$  (also nessecarily invertible in this case) is called the *LU factorization* of *A*.

## Remarks

- (1) If  $A \in \mathbb{C}^{n \times n}$  is invertible and has an LU factorization, it is unique.
- (2) One can show that  $A \in \mathbb{C}^{n \times n}$  has an LU factorization iff for  $1 \leq j \leq n$ , the upper left  $j \times j$  principal submatrix

$$\left[\begin{array}{cccc}a_{11}&\cdots&a_{ij}\\\vdots\\a_{j1}&\cdots&a_{jj}\end{array}\right]$$

is invertible.

(3) Not every invertible 
$$A \in \mathbb{C}^{n \times n}$$
 has an LU-factorization  
Example:  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 

Typically, one must permute the rows of A to move nonzero entries to the appropriate spot for the elimination to proceed. Recall that a permutation matrix  $P \in \mathbb{C}^{n \times n}$  is the identity I with its rows (or columns) permuted: so

 $P \in \mathbb{R}^{n \times n}$  is orthogonal, and  $P^{-1} = P^T$ .

Permuting the rows of A amounts to left multiplication by a permutation matrix  $P^T$ ; then  $P^T A$  has an LU factorization, so A = PLU (called the PLU factorization of A).

- (4) Fact: Every invertible  $A \in \mathbb{C}^{n \times n}$  has a (not necessarily unique) PLU factorization.
- (5) The LU factorization can be used to solve linear systems Ax = b (where  $A = LU \in \mathbb{C}^{n \times n}$  is invertible). The system Ly = b can be solved by forward substitution (1<sup>st</sup> equation gives  $x_1$ , etc.), and Ux = y can be solved by back-substitution ( $n^{\text{th}}$  equation gives  $x_n$ , etc.), giving the solution to? Ax = LUx = b.

4

**Example:** We now use the procedure outlined above to compute the LU factorization of the matrix

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 4 & 2 \\ -1 & 1 & 3 \end{bmatrix}.$$
$$L_1^{-1}A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 2 & 4 & 2 \\ -1 & 1 & 3 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 2 & 5 \end{bmatrix}$$

$$\begin{aligned} L_2^{-1}L_1^{-1}A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 2 & 5 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 8 \end{bmatrix} \end{aligned}$$

We now have

$$U = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 8 \end{bmatrix}$$

,

and

$$L = L_1 L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}.$$

1.2. The Cholesky Factorization. We now consider the application of the LU factorization to a symmetric positive definite matrix, but with a twist. Suppose the  $n \times n$  matrix H is symmetric and we have performed the first step in the procedure for computing the LU factorization of H so that

$$L_1^{-1}H = U_1$$
.

Clearly,  $U_1$  is no-longer symmetric (assuming  $L_1$  is not the identity matrix). To recover symmetry we could multiply  $U_1$  on the right by the upper triangular matrix  $L_1^{-T}$  so that

$$L_1^{-1}HL_1^{-T} = U_1L_1^{-T} = H_1 .$$

We claim that  $H_1$  necessarily has the form

$$H_1 = \left[ \begin{array}{cc} h_{(1,1)} & 0 \\ 0 & \hat{H}_1 \end{array} \right],$$

where  $h_{(1,1)}$  is the (1,1) element of H and  $\hat{H}_1$  is an  $(n-1) \times (n-1)$  symmetric matrix. For example, consider the matrix

$$H = \left[ \begin{array}{rrrr} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{array} \right].$$

In this case, we get

$$L_{1}^{-1}HL_{1}^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix}.$$

If we now continue this process with the added feature of multiplying on the right by  $L_j^{-T}$  as we proceed, we obtain

$$L^{-1}HL^{-T} = D,$$

or equivalently,

$$H = LDL^T,$$

where L is a unit lower triangular matrix and D is a diagonal matrix. Note that the entries on the diagonal of D are not necessarily the eigenvalues of H since the transformation  $L^{-1}HL^{-T}$  is not a similarity transformation.

Observe that if it is further assumed that H is positive definite, then the diagonal entries of D are necessarity all positive and the factorization  $H = LDL^T$  can always be obtained, i.e. no zero pivots can arise in computing the LU factorization (see exercises).

Let us apply this approach by continuing the computation of the LU factorization for the matrix given above. Thus far we have

$$L_1^{-1}HL_1^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix}.$$

Next

$$L_{2}^{-1}L_{1}^{-1}HL_{1}^{-T}L_{2}^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -7 \end{bmatrix},$$

giving the desired factorization

$$H = LDL^{T} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -7 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that this implies that the matrix H is not positive definite.

We make one final comment on the positive definite case. When H is symmetric and positive definite, an LU factorization always exists and we can use it to obtain a factorization of the form  $H = LDL^T$ , where L is unit lower triangular and D is diagonal with positive diagonal entries. If  $D = \text{diag}(d_1, d_2, \ldots, d_n)$  with each  $d_i > 0$ , the  $D^{1/2} = \text{diag}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$ . Hence we can write

$$H = LDL^T = LD^{1/2}D^{1/2}L^T = \hat{L}\hat{L}^T,$$

where  $\hat{L} = LD^{1/2}$  is a non-singular lower triangular matrix. The factorization  $H = \hat{L}\hat{L}^T$ where  $\hat{L}$  is a non-singular lower triangula matrix is called the *Cholesky factorization* of H. In this regard, the process of computing a Chlesky factorization is an effective means for determining is a symmetric matrix is positive definite.

1.3. The QR Factorization. Recall the Gram-Schmidt orthogonalization process for a sequence of linearly independent vectors  $a_1, \ldots, a_n \in \mathbb{R}^n$ . Define  $q_1, \ldots, q_n$  inductively, as follows: set

 $p_1 = a_1, \qquad q_1 = p_1 / \|p_1\|,$ 

$$p_j = a_j - \sum_{i=1}^{j-1} \langle a_j, q_j \rangle q_i$$
 for  $2 \le j \le n$ , and

$$q_j = p_j / \|p_j\| .$$

For  $1 \leq j \leq n$ ,

$$q_j \in \operatorname{Span}\{a_1,\ldots,a_j\},\$$

so each  $p_j \neq 0$  by the lin. indep. of  $\{a_1, \ldots, a_n\}$ . Thus each  $q_j$  is well-defined. We have  $\{q_1, \ldots, q_n\}$  is an orthonormal basis for  $\text{Span}\{a_1, \ldots, a_n\}$ . Also

$$a_k \in \operatorname{Span}\{q_1, \dots, q_k\} \qquad 1 \le k \le n,$$

so  $\{q_1, \ldots, q_k\}$  is an orthonormal basis of  $\text{Span}\{a_1, \ldots, a_k\}$ . Define

$$r_{jj} = ||p_j||$$
 and  $r_{ij} = \langle a_j, q_i \rangle$  for  $1 \le i < j \le n$ ,

we have:

$$a_{1} = r_{11} q_{1},$$

$$a_{2} = r_{12} q_{1} + r_{22} q_{2},$$

$$a_{3} = r_{13} q_{1} + r_{23} q_{2} + r_{33} q_{3},$$

$$\vdots$$

$$a_{n} = \sum_{i=1}^{n} r_{in} q_{i}.$$

Set

 $A = [a_1 \ a_2 \ \dots \ a_n], \quad R = [r_{ij}], \text{ and } Q = [q_1 \ q_2 \ \dots \ q_n],$ 

where  $r_{ij} = 0$ , i > j. Then

$$A = QR$$
,

where Q is unitary and R is upper triangular.

## Remarks

- (1) If  $a_1, a_2, \cdots$  is a linearly independent sequence, apply Gram-Schmidt to obtain an orthonormal sequence  $q_1, q_2, \ldots$  such that  $\{q_1, \ldots, q_k\}$  is an orthonormal basis for  $\text{Span}\{a_1, \ldots, a_k\}, k \ge 1$ .
- (2) If the  $a_j$ 's are linearly dependent, for some value(s) of k,

$$a_k \in \operatorname{Span}\{a_1, \dots, a_{k-1}\}, \text{ so } p_k = 0.$$

The process can be modified by setting  $r_{kk} = 0$ , not defining a new  $q_k$  for this iteration and proceeding. We end up with orthogonal  $q_j$ 's. Then for  $k \ge 1$ , the vectors  $\{q_1, \ldots, q_k\}$  form an orthonormal basis for  $\text{Span}\{a_1, \ldots, a_{\ell+k}\}$  where  $\ell$  is the number of  $r_{jj} = 0$ . Again we obtain A = QR, but now Q may not be square.

(3) The classical Gram-Schmidt algorithm as described does not behave well computationally. This is due to the accumulation of round-off error. The computed  $q_j$ 's are not orthogonal:  $\langle q_j, q_k \rangle$  is small for  $j \neq k$  with j near k, but not so small for  $j \ll k$ or  $j \gg k$ .

An alternate version, "Modified Gram-Schmidt," is equivalent in exact arithmetic, but behaves better numerically. In the following "pseudo-codes," p denotes a temporary storage vector used to accumulate the sums defining the  $p_j$ 's.

Classic Gram-Schmidt  
For 
$$j = 1, \dots, n \ do$$
Modified Gram-Schmidt  
For  $j = 1, \dots, n \ do$  $p := a_j$   
For  $i = 1, \dots, j - 1 \ do$   
 $\left[ \begin{array}{c} r_{ij} = \langle a_j, q_i \rangle \\ p := p - r_{ij}q_i \\ r_{jj} := \|p\| \\ q_j := p/r_{jj} \end{array} \right]$ For  $i = 1, \dots, j - 1 \ do$   
 $\left[ \begin{array}{c} r_{ij} = \langle p, q_i \rangle \\ p := p - r_{ij}q_i \\ r_{jj} := \|p\| \\ q_j := p/r_{jj} \end{array} \right]$ 

The only difference is in the computation of  $r_{ij}$ : in Modified Gram-Schmidt, we orthogonalize the accumulated partial sum for  $p_i$  against each  $q_i$  successively.

**Theorem 1.1.** Suppose  $A \in \mathbb{R}^{m \times n}$  with  $m \ge n$ . Then

$$\exists \quad unitary \quad Q \in \mathbb{R}^{m \times m} \quad upper \ triangular \quad R \in \mathbb{R}^{m \times n}$$

for which

A = QR.

If  $\widetilde{Q} \in \mathbb{R}^{m \times n}$  denotes the first *n* columns of *Q* and  $\widetilde{R} \in \mathbb{R}^{n \times n}$  denotes the first *n* rows of *R*, then

$$A = QR = \begin{bmatrix} \widetilde{Q} & * \end{bmatrix} \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} = \widetilde{Q}\widetilde{R}.$$

Moreover

- (a) We may choose an R to have nonnegative diagonal entries.
- (b) If A is of full rank, then we can choose R with positive diagonal entries, in which case the condensed factorization A = Q R is unique factorization A = Q R is unique since then Q = Q and R = R.

Proof. If A has full rank, apply the Gram-Schmidt. Define

$$\widetilde{Q} = [q_1, \dots, q_n] \in \mathbb{R}^{m \times n}$$
 and  $\widetilde{R} = [r_{ij}] \in \mathbb{R}^{n \times n}$ 

as above, so

$$A = \widetilde{Q}\widetilde{R}$$

Extend  $\{q_1, \ldots, q_n\}$  to an orthonormal basis  $\{q_1, \ldots, q_m\}$  of  $\mathbb{R}^m$ , and set

$$Q = [q_1, \dots, q_m]$$
 and  $R = \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} \in \mathbb{C}^{m \times n}$ , so  $A = QR$ .

As  $r_{jj} > 0$  in the G-S process, we have (b). Uniqueness follows by induction passing through the G-S process again, noting that at each step we have no choice.

#### Remarks

- (1) In practice, there are more efficient and better computationally behaved ways of calculating the Q and R factors. The idea is to create zeros below the diagonal (successively in columns  $1, 2, \ldots$ ) as in Gaussian Elimination, except we now use Householder transformations (which are unitary) instead of the unit lower triangular matrices  $L_j$ .
- (2) A QR factorization is also possible when m < n.

$$A = Q[R_1 \ R_2] ,$$

where  $Q \in \mathbb{C}^{m \times m}$  is unitary and  $R_1 \in \mathbb{C}^{m \times m}$  is upper triangular.

Every  $A \in \mathbb{R}^{m \times n}$  has a QR-factorization, even when m < n. Indeed, if

$$\operatorname{rank}(A) = k$$
.

there always exist

 $Q \in \mathbb{R}^{m \times k}$  with orthonormal columns,

 $R \in \mathbb{R}^{k \times n}$  full rank upper triangular,

and a permutation matrix  $P \in \mathbb{R}^{n \times n}$  such that

$$(*) AP = QR.$$

Moreover, if A has rank n (so  $m \ge n$ ), then  $R \in \mathbb{R}^{n \times n}$  is nonsingular. On the other hand, if m < n, then

$$R = [R_1 \ R_2],$$

where  $R_1 \in \mathbb{R}^{k \times k}$  is nonsingular. Finally, if  $A \in \mathbb{R}^{m \times n}$ , then the same facts hold, but now both Q and R can be chosen to be real matrices.

## **QR-Factorization and Orthogonal Projections**

Let  $A \in \mathbb{R}^{m \times n}$  have condensed QR-factorization

$$A = \widetilde{Q}\widetilde{R}$$
.

Then by construction the columns of  $\widetilde{Q}$  form an orthonormal basis for the range of A. Hence  $P = \widetilde{Q}\widetilde{Q}^T$  is the orthogonal projector onto the range of A. Similarly, if the condensed QR-factorization of  $A^T$  is

$$A^T = \widetilde{Q}_1 \widetilde{R}_1 ,$$

then

$$P_1 = \widetilde{Q}_1 \widetilde{Q}_1^T$$

is the orthogonal projector onto  $\operatorname{Ran}(A^T) = \ker(A)^{\perp}$ , and so

$$I - \widetilde{Q}_1 \widetilde{Q}_1^T$$

is the orthogonal projector onto  $\ker(A)$ .

The QR-factorization can be computed using either Givens rotations or Householder reflections. Although, the approach via rotations is arguably more stable numerically, it is more difficult to describe so we only illustrate the approach using Householder reflections.

## QR using Householder Reflections

10

Given  $w \in \mathbb{R}^n$  we can associate the matrix

$$U = I - 2\frac{ww^T}{w^T w}$$

which reflects  $\mathbb{R}^n$  across the hyperplane  $\text{Span}\{w\}^{\perp}$ . The matrix U is call the Householder reflection across this hyperplane.

Given a pair of vectors x and y with

$$||x||_2 = ||y||_2$$
, and  $x \neq y$ 

there is a Householder reflection such that y = Ux:

$$U = I - 2\frac{(x-y)(x-y)^{T}}{(x-y)^{T}(x-y)}$$

Proof.

$$Ux = x - 2(x - y) \frac{\|x\|^2 - y^T x}{\|x\|^2 - 2y^T x + \|y\|^2}$$
$$= x - 2(x - y) \frac{\|x\|^2 - y^T x}{2(\|x\|^2 - y^T x)}$$

= y

since ||x|| = ||y||.

# QR using Householder Reflections

We now describe the basic *deflation* step in the QR-factorization.

$$A_0 = \begin{bmatrix} \alpha_0 & a_0^T \\ b_0 & A_0 \end{bmatrix}.$$

 $\operatorname{Set}$ 

$$\nu_0 = \left\| \begin{pmatrix} \alpha_0 \\ b_0 \end{pmatrix} \right\|_2.$$

Let  $H_0$  be the Householder transformation that maps

$$\begin{pmatrix} \alpha_0 \\ b_0^T \end{pmatrix} \mapsto \nu_0 \ e_1 \quad :$$

$$H_0 = I - 2\frac{ww^T}{w^T w} \quad \text{where} \quad w = \begin{pmatrix} \alpha_0 \\ b_0 \end{pmatrix} - \nu_0 e_1 = \begin{pmatrix} \alpha_0 - \nu_0 \\ b_0 \end{pmatrix}.$$

Thus,

$$H_0 A = \begin{bmatrix} \nu_0 & a_1^T \\ 0 & A_1 \end{bmatrix}$$

A problem occurs if  $\nu_0 = 0$  or

$$\begin{pmatrix} \alpha_0 \\ b_0 \end{pmatrix} = 0 .$$

In this case, permute the offending column to the right bringing in the column of greatest magnitude. Now repeat with  $A_1$ .

If the above method if implemented by always permuting the column of greatest magnitude into the current pivot column, then

$$AP = QR$$

gives a QR-factorization with the diagonal entries of R nonnegative and listed in the order of descending magnitude.