Range of $A$

Let $A \in \mathbb{R}^{m\times n}$ and full rank, with $m \geq n$

(see overdetermined linear systems).

Goal: Find an orthogonal basis of $\mathcal{R}_n = \text{range}(A)$

Idea: Produce a chain of subspaces and a basis

$$\text{span}(a_1) \subseteq \text{span}(a_1, a_2) \subseteq \cdots \subseteq \text{span}(a_1, a_2, \ldots, a_n)$$

$$=: \mathcal{R}_1 \quad =: \mathcal{R}_2 \quad =: \mathcal{R}_n$$
Range of $A$: orthogonal basis

$\text{span}(a_1) \subseteq \text{span}(a_1, a_2) \subseteq \cdots \subseteq \text{span}(a_1, a_2, \ldots, a_n) =: R_n$

We aim for an orthogonal basis $R_j$ with:

$R_j = \text{span}(q_1, q_2, \ldots, q_j) = \text{span}(a_1, a_2, \ldots, a_j)$

\[
\begin{pmatrix}
a_1 & a_2 & \cdots & a_n \\
\end{pmatrix} = A = \begin{pmatrix} q_1 & q_2 & \cdots & q_n \end{pmatrix} =: \hat{Q}
\]

\[
\begin{pmatrix}
r_{11} & r_{12} & \cdots & r_{1n} \\
r_{22} & \cdots & r_{2n} \\
\vdots & \ddots & \vdots \\
r_{nn} & & & \\
\end{pmatrix} =: \hat{R}
\]
Range of $A$: reduced $QR$ factorization

\[
A = \hat{Q}\hat{R}
\]

\[
\begin{align*}
    a_1 &= r_{11}q_1 \\
    a_2 &= r_{12}q_1 + r_{22}q_2 \\
    \vdots \\
    a_n &= r_{1n}q_1 + r_{2n}q_2 + \cdots + r_{nn}q_n
\end{align*}
\]
Range of $A$: full $QR$ factorization

Complete $\hat{Q}$ by $m - n$ additional orthogonal columns and $\hat{R}$ by a $(m - n) \times n$ zero matrix:

$$A = \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} Q \end{bmatrix} \begin{bmatrix} R \end{bmatrix}$$
Gram-Schmidt orthogonalization

Algorithm to compute $\hat{Q}$ and $\hat{R}$:

Step j: Given: $q_1, \ldots, q_{j-1}$ orthonormal basis of $\text{span}(a_1, \ldots, a_{j-1})$.
Find: $q_j \in \text{span}(a_1, \ldots, a_j)$ with $q_j \perp q_i$

$$v_j = a_j - q_1^T a_j q_1 - q_2^T a_j q_2 - \cdots - q_{j-1}^T a_j q_{j-1}$$

and

$$q_j = \frac{v_j}{\|v_j\|_2}$$
Gram-Schmidt orthogonalization (Cont.)

Set $r_{ij} = q_i^T a_j$ for $i \neq j$ and $r_{ii} = \|v_i\|_2$

Compute $q_1, \ldots, q_j$ from

\[
\begin{align*}
v_1 & = a_1 \\
v_2 & = a_2 - r_{12} q_1 \\
& \vdots \\
v_n & = a_n - \sum_{i=1}^{n-1} r_{in} q_i
\end{align*}
\]

and

\[
q_i = \frac{v_i}{r_{ii}}
\]
Gram-Schmidt orthogonalization - Projections

Note, $r_{ij}q_i = q_iq_i^Ta_j$ is a projection of $a_j$ onto $\text{span}(q_i)$.

Thus, Gram-Schmidt orthogonalization can be written as a sequence of projections:

$$v_j = a_j - P_{q_1}a_j - P_{q_2}a_j - \ldots P_{q_{j-1}}a_j = (I - \hat{Q}_{j-1}\hat{Q}_{j-1}^T)a_j =: P_j$$

Problem: Not a very good method (see assignment). Look for alternative idea!
Householder reflection in 2D

Reflection across $\text{span}(v)^\perp$
The idea:
Select a vector $v$ so that $H$ reflects $a$ onto the coordinate axis!

$$H a = H \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} \lVert a \rVert_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$  

This leads to $v = a - \hat{a}$.

(see also the figure in the next slide)  
Check that $H a = \lVert a \rVert e_1$.  

Numerical Analysis, Lund University, 2017
Householder triangularization (Cont.)

Householder transformation for annihilating (nollställa) entries in a vector
function [Q]=householder(A)
% [Q]=function house(a)
% computes a Householder matrix to
% introduce zeros in the first column of A
% by an orthogonal transformation
%
m = sizedim(A,1)
a=A[:,1]  
la=norm(a)
ahat=[la;zeros(n-1,1)]
v=a-ahat;
v=v/norm(v)
Q=eye(n)-2*(v*v')/(v'*v);
from numpy import array, eye, outer
from scipy.linalg import norm

def Householder_step(A):
    
    """
    One Householder step
    Example code for
    FMNN01/NUMA11 Numerical Linear Algebra
    """
    a=A[:,0]
    m=A.shape[0]
    ahat = array([norm(a)]+(m-1)*[0.])
    v=a-ahat
    v=v/norm(v)
    Q=eye(m)-2*outer(v,v)
    return Q
**Multiplication with a Householder matrix**

Multiplying with a Householder matrix is cheap:

\[
Ha = (I - 2 \frac{vv^T}{v^Tv})a = a - \frac{2}{v^Tv}(v^Ta)v
\]

Thus we never need to form and store this matrix!
First step

>> A=[1,2;4,5;7,8]
>> H1=housholder(A); A1=H1*A
A1 =

-8.1240   -9.6011
  0.0000   -0.0860
  0.0000   -0.9004
Second step

We look now for a matrix $H_2$, which transforms $A_1(2 :, 2)$ into the vector

$$
\begin{pmatrix}
-9.6011 \\
\mathbf{a}_2 \\
0
\end{pmatrix}
$$
Second step (Cont.)

We form

\[ H_2 = \begin{pmatrix} 1 & H_{22} \\ \end{pmatrix} \]

where \( H_{22} \) is a 2 \times 2 matrix which reflects

\[
\begin{pmatrix} -0.0860 \\ 0.9004 \end{pmatrix}
\]

onto

\[
\begin{pmatrix} a_2 \\ 0 \end{pmatrix}
\]

The same procedure as before but in one dimension less!
Second step (Cont.)

>> A1=H1*A % first step
A1 =
    -8.1240    -9.6011
     0.0000    -0.0860
     0.0000    -0.9004

>> H22=householder(A1(2:end,2:end));
H22 =
    -0.0950    -0.9955
    -0.9955    0.0950

>> H2=[1 0 0 ;[0 0]’,H22]
H2 =
    1.0000     0     0
     0 -0.0950   -0.9955
     0 -0.9955    0.0950

>> A2=H2*A1 % second step
A2 =
    -8.1240    -9.6011
     0     0.9045
     0     0.0000
    -0.0000    0.0000
The rank of a matrix is the number of non-zero diagonal elements in \( R \):

\[
\begin{pmatrix}
1 & 2 & 3 \\
1 & 2 & 3 \\
2 & 5 & 7
\end{pmatrix}
\]

\[
\begin{pmatrix}
-0.4082 & -0.5774 & -0.7071 \\
-0.4082 & -0.5774 & 0.7071 \\
-0.8165 & 0.5774 & 0.0000
\end{pmatrix}
\]

Due to round-off errors, the rank could be computed as:

\[
\text{nnz(diag(R))}
\]

\[
\text{nnz(find(abs(diag(R))>1.e-15))}
\]

3 % due to round-off

2
Null space of a matrix

Definition
Let $A$ be a $n \times m$ matrix. The set of all $x \in \mathbb{R}^m$ with $Ax = 0$ is called the null space (or kernel) of $A$:

$$\mathcal{N}(A) := \{x \in \mathbb{R}^m | Ax = 0\}$$

In MATLAB

$$\gg \text{null}(A)$$
Null space of a matrix (Cont.)

\[
A = \begin{bmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8
\end{bmatrix}
\]

\[
\text{null}(A)
\]

\[
\text{ans} = \begin{bmatrix}
-0.5026 & 0.2178 \\
0.8324 & 0.0842 \\
-0.1571 & -0.8218 \\
-0.1727 & 0.5198
\end{bmatrix}
\]

\[
\text{ans} \cdot \text{ans}
\]

\[
\begin{bmatrix}
0.0333 & -0.0444 \\
0.0333 & -0.0444 \\
0.0333 & -0.0444 \\
0.3997 & -0.0888
\end{bmatrix}
\]
Null space of a matrix with $QR$

$Ax = QRx = 0.$

If $A$ has rank $k < n$ then $R$ takes the following form

$$R = \begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix},$$

(possibly after a previous permutation of the columns of $A$)

$R_1$ is a $k \times k$ upper triangular matrix with no zeros on its diagonal and $S$ a $k \times (n - k)$-matrix.
Null space of a matrix with $QR$ (Cont.)

This gives

$$Ax = QRx = (Q_1, Q_2) \begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

first $k$ columns of $Q$ denoted by $Q_1$
first $k$ rows of $x$ denoted by $x_1$. 

Numerical Analysis, Lund University, 2017
Null space of a matrix with $QR$ (Cont.)

Multiplication gives

$$Q_1 R_1 x_1 + Q_1 S x_2 = 0$$

or due to the orthogonality of $Q_1$

$$x_1 = -R_1^{-1} S x_2.$$  

Thus, the general solution of $Ax = 0$ is given by

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -R_1^{-1} S x_2 \\ x_2 \end{pmatrix} = \begin{pmatrix} -R_1^{-1} S \\ I \end{pmatrix} x_2.$$
Null space of a matrix with $QR$ (Cont.)

Components of $x_2$ are the free parameters in the solution of the homogenous problem and the columns of

$$V = \begin{pmatrix} -R_1^{-1}S \\ I \end{pmatrix}$$

form a basis of the null space.
(Note again, we did not consider any permutations in the derivation)

The column space (or range space) is given by the columns of $Q_1$. 
Null space of a matrix in Python

```
n=3
A=array([[1,2,3],[4,5,6],[6,7,8]]
# QR factorization with permutations
[Q,R,P]=qr(A,pivoting=True)
rankA=sum(diag(abs(R)>1.e-15))
R1=R[:rankA,:rankA]
S=R[:rankA,rankA:]
V=vstack((-solve(R1,S),eye(n-rankA)))
V=V[P[P],:]  # backpermute
# Test
norm(A@V))  # gives 4.61 e-16
```
Null space of a matrix in mechanics

How many degrees of freedom has this system?