

```
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>> % Vi skal løse Ax=b hvor
>> A = [1 0; 0 1; -1 1]
```

A =

1	0
0	1
-1	1

```
>> b = [5.1; 9.4; 4.1]
```

b =

5.1000
9.4000
4.1000

```
>> % Vi vil bruge QR-faktorisering af A
>> [Q,R]=qr(A)
```

Q =

$$\begin{bmatrix} -0.7071 & -0.4082 \\ 0 & -0.8165 \\ 0.7071 & -0.4082 \end{bmatrix} \begin{matrix} 0.5774 \\ -0.5774 \\ 0.5774 \end{matrix}$$

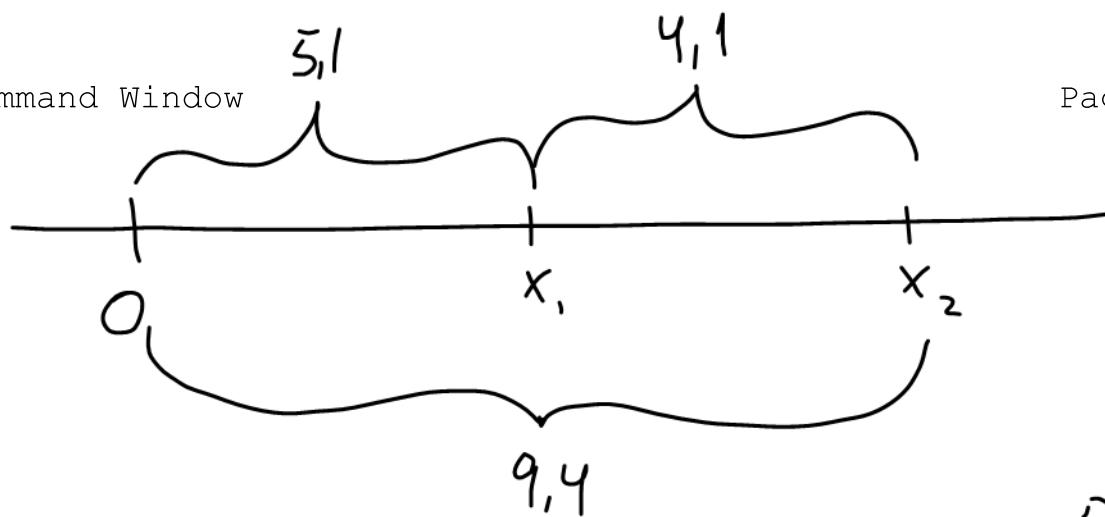
R =

$$\begin{bmatrix} -1.4142 & 0.7071 \\ 0 & -1.2247 \\ 0 & 0 \end{bmatrix}$$

```
>> % MATLAB definerer QR-faktorisering på en anden måde end vores bog.
>> % MATLAB har tilføjet en ekstra søjle til Q så søjlerne er en ortonormal basis.
>> % Tilsvarende er der tilføjet en ekstra 0-række til R.
>> % Vi fjerner disse
>> Q=Q(:,1:2)
```

Q =

-0.7071	-0.4082
0	-0.8165



$$\begin{aligned} x_1 &= s_1 \\ x_2 &= q_1 \\ -x_1 + x_2 &= u_1 \end{aligned}$$

```
0.7071 -0.4082
```

```
>> R=R(1:2,:)
```

```
R =
```

```
-1.4142 0.7071
0 -1.2247
```

```
>> % Vi kontrollerer:
>> Q*R
```

```
ans =
```

```
1.0000 0.0000
0 1.0000
-1.0000 1.0000
```

```
>>
>> % Vi skal nu løse ligningssystemet Rx=Q^T*b.
>> % Vi beregner først Q^T*b
>> c = Q'*b
```

```
c =
```

```
-0.7071
-11.4310
```

```
>> % For løse Rx=c kan vi gange med den inverse til R
>> R^-1
```

```
ans =
```

```
-0.7071 -0.4082
0 -0.8165
```

```
>> % Løsningen til Rx=Q^T*b er
>> d = R^-1*c
```

d =

$$\begin{bmatrix} 5.1667 \\ 9.3333 \end{bmatrix}$$

```
>> % Er dette en løsning til Ax=b
>> A*d
```

```
ans =
```

```
5.1667
9.3333
4.1667
```

```
>> % Nej, men det er tæt på!
>>
```

**Metoden fra vores bog
(anvendelse af Gram-Schmidt)
giver faktoriseringen $A=(-Q)(-R)$.**