In [1]:

```
import numpy as np
import numpy.linalg as lg
import math
```

# Problem 1

## a)

determine if lines are parallel:

$4x - y + 2z = 5$ and $7x - 3y + 4z = 8$

$$\left[\begin{array}{ccc|c} 4 & -1 & 2 & 5 \\ 0 & -5 & 2 & -3 \end{array}\right]$$

Those two rows are independent as well as all of the columns, thus we could say they are not parallel to each other.

$x - 4y + 3z = 2$ and $3x - 12y + 9z = 7$

$$\begin{bmatrix} 1 & -4 & -3 \\ 3 & -12 & -9 \end{bmatrix}$$

The rows are liniearly dependent. That means that these two vectors are parallel

b) perpendicular

$3x - y + z = 0$ and $x + 2z = -1$

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

not parallel

$x - 2y + 3z = 4$ and $-2x + 5y + 4z = -1$

$$\begin{bmatrix} 1 & -2 & 3 \end{bmatrix} \begin{bmatrix} -2 \\ 5 \\ 4 \end{bmatrix} = -2 - 10 + 12 = 0$$

parallel

# Problem 2

## a)

not added yet.

### b)

not added yet.

### c)

not added yet.

## Problem 3

### a)

Find distance between $\frac{x-1}{2} = \frac{y-2}{1} = \frac{z+1}{2}$ and $P = (1, 1, 0)$

$$\begin{vmatrix} i & j & k \\ 0 & -1 & 1 \\ 2 & 1 & 2 \end{vmatrix} = i(-2-1) - j(-2) + k(2) = -3i + 2j + 2k = \sqrt{9+4+4} = \sqrt{17}$$

### b)

The idea is to get projection onto normal vector. Lets see it. Denomenator of formula represents the length of normal vector while nominator is a parallel plane that lies on some distance from the origin.

### c)

Find distance between $2x + 2y - z = 2$ and $P = (1, 0, 1)$

$$d = \frac{|AM_x + BM_y + CM_z + D|}{\sqrt{A^2 + B^2 + C^2}} = \frac{|2 + 0 - 1 - 2|}{\sqrt{2^2 + 2^2 + 1^2}} = \frac{1}{3}$$

## Problem 4

### a)

We could consider finding $w^T v = 0$ and $w^T u = 0$ as cross product obtaining process. Cross product is just a cofactor expention satisfied by formula $(-1)^{i+j} M_{i,j} = A_{i,j}$

We could represent it as:

$$w_1 = \begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix} w_2 = -\begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix} w_3 = \begin{vmatrix} u_1 & u_2 \\ v_1 & v_3 \end{vmatrix}$$

$$w^T u = u_1 \begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix} - u_2 \begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix} + u_3 \begin{vmatrix} u_1 & u_2 \\ v_1 & v_2 \end{vmatrix} = det \begin{vmatrix} u_1 & u_2 & u_3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = 0$$

We could see that we obtain 2 dependent vectors, so determinant is equal to 0, that states (in this case) for orthgonality of two vectors. The same idea is behind the $w^T v$.

### b)

For the $R^n$ case we have the same idea. Any orthogonal vector will have 2 dependent vectors.

# Problem 5

### a)

If $Q_1$ and $Q_2$ are orthogonal matrixes, this means, that they have such a property: $Q^T Q = I$ thus we could state that $Q^T = Q^{-1}$

$Q_1 Q_2$ multiplication of orthogonal matrixes is an orthgonal matrix as well.

Consider two matrixes $Q$ and $R$ such that $Q^T Q = I$ and $R^T R = I$

$(QR)^T (QR) = R^T (Q^T Q) R = R^T R = I$

### b)

If $Q$ is invertable upper triangular matrix, than $Q^{-1}$ is also upper triangular. $Q^T$ lower triangular. But orthogonal matrixes has a property $Q^T = Q^- 1$, thus we could say that to satisfy this property $Q$ must be diagonal.

# Problem 6

not added yet.

# Problem 7

Is there a least squares solution for: $\begin{bmatrix} 1 & -1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ s \end{bmatrix}$

$A^T A x^* = A^T b$

$\begin{bmatrix} 21 & 25 \\ 25 & 35 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ -1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ s \end{bmatrix}$

$\begin{bmatrix} 71 \\ 95 \end{bmatrix} = \begin{bmatrix} 1 + 2 + 4s \\ -1 + 3 + 5s \end{bmatrix}$

$\begin{cases} 4s = 68 \\ 5s = 97 \end{cases}$

$\begin{cases} s = 17 \\ s = 19.4 \end{cases}$

no least square solution found.

# Problem 8

$b = C + Dt$

We need to find best line fit.

$Ax = proj_{C(A)} b$

$C(A)^{\perp} = N(A^T)$

$A^T(Ax^* - b) = 0$

$A^T Ax^* - A^T b = 0$

$x^* = (A^T A)^{-1} A^T b$

## a)

$$\begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 3 & 3 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 4 & 8 \\ 8 & 22 \end{bmatrix}$$

$$(A^T A)^{-1} = \begin{bmatrix} \frac{11}{12} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{1}{6} \end{bmatrix}$$

$$x^* = \begin{bmatrix} \frac{11}{12} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{6} \end{bmatrix}$$

## b)

same idea applied here

$b = C + Dt + Et^2$

$$\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \begin{bmatrix} C \\ D \\ E \end{bmatrix} = \begin{bmatrix} 0 \\ -10 \\ -48 \\ -76 \end{bmatrix}$$

$$\begin{bmatrix} C \\ D \\ E \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ -3 \end{bmatrix}$$

$b = 2 + 5t - 3t^2$

### c)

and here

$b = C + Dt + Et^2 + Ft^3$

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \begin{bmatrix} C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} -14 \\ -5 \\ -4 \\ 1 \\ 22 \end{bmatrix}$$

$$\begin{bmatrix} C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} -5 \\ 3 \\ -4 \\ 2 \end{bmatrix}$$

$b = -5 + 3t - 4t^2 + 2t^3$

## Problem 9

### a)

$A = \begin{bmatrix} \overrightarrow{u_1} & \overrightarrow{u_1 + \varepsilon u_2} \end{bmatrix}$

$x^* = (A^T A)^{-1} A^T b$

$x^*$ explodes if $\varepsilon$ approaches $0$

$$A^T A = \begin{bmatrix} u_1^2 & u_1(u_1 + \varepsilon u_2) \\ u_1(u_1 + \varepsilon u_2) & (u_1 + \varepsilon u_2)^2 \end{bmatrix} = \begin{bmatrix} u_1^2 & u_1^2 \\ u_1^2 & u_1^2 + \varepsilon^2 u_2^2 \end{bmatrix}$$

$$(A^T A)^{-1} = \frac{1}{\varepsilon^2 u_1^2 u_2^2} \begin{bmatrix} u_1^2 + \varepsilon^2 u_2^2 & -u_1^2 \\ -u_1^2 & u_1^2 \end{bmatrix}$$

$$x^* = \frac{1}{\varepsilon^2 u_1^2 u_2^2} \begin{bmatrix} u_1^2 + \varepsilon^2 u_2^2 & -u_1^2 \\ -u_1^2 & u_1^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

having $\varepsilon$ in denominator leads to explosion:

$$\lim_{\varepsilon \to 0} x^* = \begin{bmatrix} \infty \\ \infty \end{bmatrix}$$

### b)

$$\frac{1}{\varepsilon^2 u_1^2 u_2^2}\left[\overrightarrow{u_1} \quad \overrightarrow{u_1 + \varepsilon u_2}\right]\begin{bmatrix} b_1(u_1^2+\varepsilon^2 u_2^2) - b_2 u_1^2 \\ -b_1 u_1^2 + b_2 u_1^2 \end{bmatrix} = \frac{1}{\varepsilon^2 u_1^2 u_2^2}\begin{bmatrix} \overrightarrow{u_1}(b_1(u_1^2+\varepsilon^2 u_2^2) - b_2 u_1^2) \\ \overrightarrow{u_1 + \varepsilon u_2}(-b_1 u_1^2 + b_2 u_1^2) \end{bmatrix} = \dots$$

after multiplication we will eliminate all $\varepsilon$

# Problem 10

## a)

$$\begin{bmatrix} u_1 & u_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & -3 \end{bmatrix}$$

In [2]:

```
u1 = np.array([1, 3])
u2 = np.array([2, -2])

w1 = u1
w2 = u2 - ( np.dot(u2.T, w1) / np.dot(w1.T, w1) ) * w1

e1 = w1 / lg.norm(w1)
e2 = w2 / lg.norm(w2)

np.array([e1, e2])
```

Out[2]:

```
array([[ 0.31622777,  0.9486833 ],
       [ 0.9486833 , -0.31622777]])
```

## b)

In [3]:

```
u1 = np.array([1, 0, 1])
u2 = np.array([1, 3, -2])
u3 = np.array([0, 2, 1])

w1 = u1
w2 = u2 - ( np.dot(u2.T, w1) / np.dot(w1.T, w1) ) * w1
w3 = u3 - ( np.dot(u3.T, w1) / np.dot(w1.T, w1) ) * w1 - ( np.dot(u3.T, w2) / np.do

e1 = w1 / lg.norm(w1)
e2 = w2 / lg.norm(w2)
e3 = w3 / lg.norm(w3)

np.array([e1, e2, e3])
```

Out[3]:

```
array([[ 0.70710678,  0.        ,  0.70710678],
       [ 0.40824829,  0.81649658, -0.40824829],
       [-0.57735027,  0.57735027,  0.57735027]])
```

# Problem 11

In order to make $QR$ fuctorization we need to make Gram–Schmidt process first.

$Q$ must be an orthogonal matrix formed by orthonormal vectors (result of Gram–Schmidt), while $R$ must be an upper triangular matrix such that multiplication on $Q$ will return back $A$.

$Q = \begin{bmatrix} e_1 & e_2 & \dots & e_n \end{bmatrix}$

Q must be orthgonal: $Q^T Q = I$

$A = QR$

$Q^T A = Q^T QR$

$Q^T A = IR$

$Q^T A = R$

**a)** $A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} u_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} u_2 = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$

$w_1 = u1$

$w_2 = u_2 - \frac{u_2^T w_1}{w_1^T w_1} w_1$

$w_2 = \begin{bmatrix} -1 \\ 3 \end{bmatrix} - \frac{\begin{bmatrix} -1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}}{\begin{bmatrix} -1 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \end{bmatrix}} \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

$e_1 = \frac{w_1}{||w_1||} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} e_2 = \frac{w_2}{||w_2||} = \begin{bmatrix} \frac{-2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix}$

$Q = \begin{bmatrix} e_1 & e_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix}$

$Q^T A = R$

$R = Q^T A = \begin{bmatrix} \frac{5}{\sqrt{5}} & \frac{5}{\sqrt{5}} \\ 0 & \frac{5}{\sqrt{5}} \end{bmatrix}$

$A = QR$

$$A = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} \frac{5}{\sqrt{5}} & \frac{5}{\sqrt{5}} \\ 0 & \frac{5}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}$$

In [4]:

```
A = np.array([[1, -1], [2, 3]])
print("A:\n", A)
u1 = A[:,0]
u2 = A[:,1]

w1 = u1
w2 = u2 - ( np.dot(u2.T, w1) / np.dot(w1.T, w1) ) * w1

e1 = w1 / lg.norm(w1)
e2 = w2 / lg.norm(w2)

Q = np.array([e1, e2]).T
print("Q:\n", Q)
print("Q must be orthogonal. Let's check it out:\n", np.dot(Q.T, Q))

R = np.dot(Q.T, A)
print("R:\n", R)

print("Check solution: QR\n", np.dot(Q, R))
```

```
A:
 [[ 1 -1]
 [ 2  3]]
Q:
 [[ 0.4472136  -0.89442719]
 [ 0.89442719  0.4472136 ]]
Q must be orthogonal. Let's check it out:
 [[1. 0.]
 [0. 1.]]
R:
 [[2.23606798 2.23606798]
 [0.         2.23606798]]
Check solution: QR
 [[ 1. -1.]
 [ 2.  3.]]
```

**b)** $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 4 \end{bmatrix} u_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u_2 = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}$

In [5]:

```
A = np.array([[1, 2], [0, 1], [1, 4]])
print("A:\n", A)
u1 = A[:,0]
u2 = A[:,1]

w1 = u1
w2 = u2 - ( np.dot(u2.T, w1) / np.dot(w1.T, w1) ) * w1

e1 = w1 / lg.norm(w1)
e2 = w2 / lg.norm(w2)

Q = np.array([e1, e2]).T
print("Q:\n", Q)
print("Q must be orthogonal. Let's check it out:\n", np.dot(Q.T, Q))

R = np.dot(Q.T, A)
print("R:\n", R)

print("Check solution: QR\n", np.dot(Q, R))
```

```
A:
 [[1 2]
 [0 1]
 [1 4]]
Q:
 [[ 0.70710678 -0.57735027]
 [ 0.          0.57735027]
 [ 0.70710678  0.57735027]]
Q must be orthogonal. Let's check it out:
 [[1. 0.]
 [0. 1.]]
R:
 [[1.41421356 4.24264069]
 [0.         1.73205081]]
Check solution: QR
 [[1. 2.]
 [0. 1.]
 [1. 4.]]
```

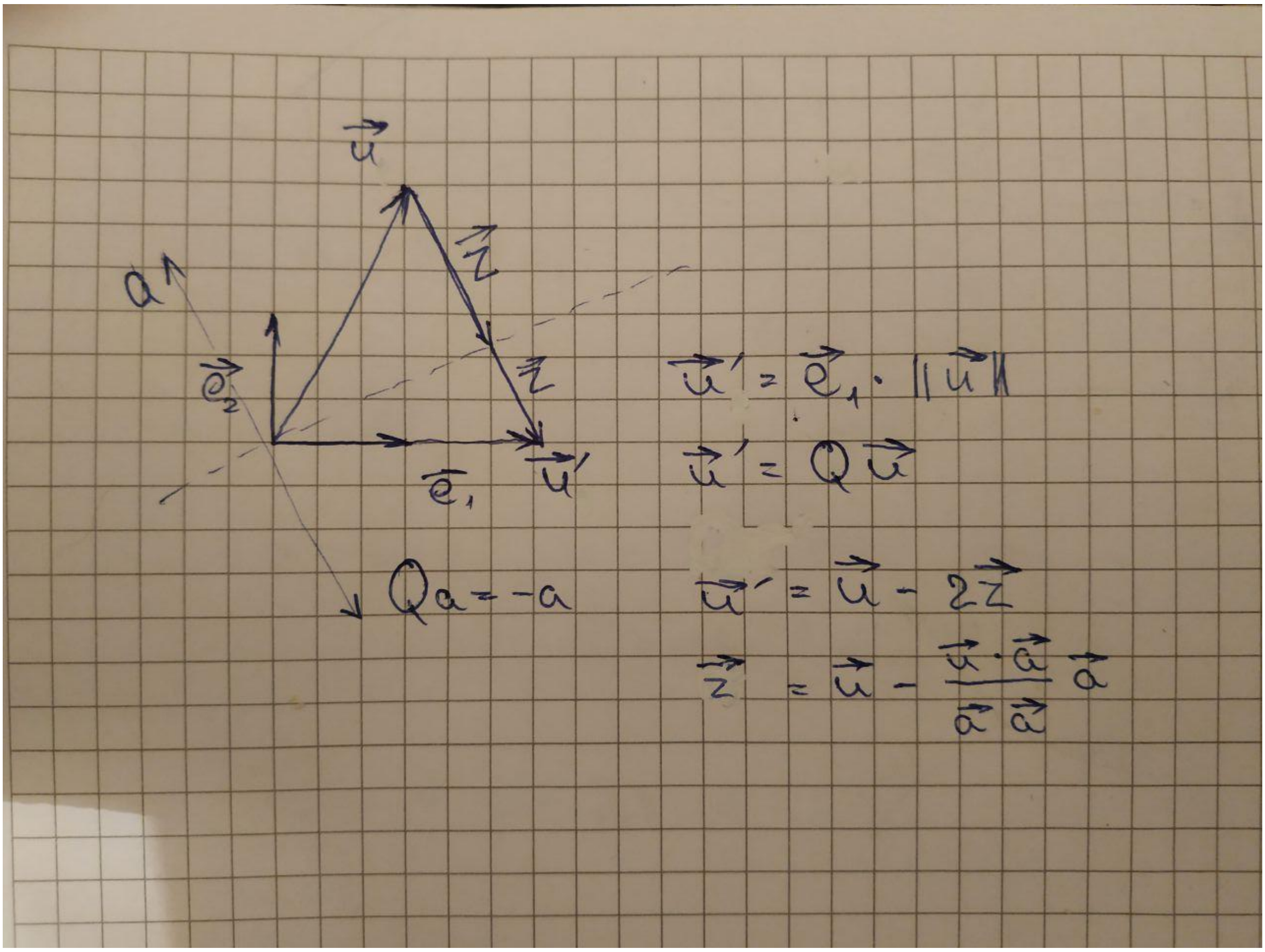
**c)** $A = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} u_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} u_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_3 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$

In [6]:

```
A = np.array([[1, 0, 2], [0, 1, 1], [2, 0, 1]])
print("A:\n", A)
print("\nthere is one dependent column, namely 3rd one\n")
u1 = A[:,0]
u2 = A[:,1]
u3 = A[:,2]

w1 = u1
w2 = u2 - ( np.dot(u2.T, w1) / np.dot(w1.T, w1) ) * w1
w3 = u3 - ( np.dot(u3.T, w1) / np.dot(w1.T, w1) ) * w1 - ( np.dot(u3.T, w2) / np.do

e1 = w1 / lg.norm(w1)
e2 = w2 / lg.norm(w2)
e3 = w3 / lg.norm(w3)

Q = np.array([e1, e2, e3])
print("Q:\n", Q)
print("Q must be orthogonal. Let's check it out:\n", np.dot(Q.T, Q))

R = np.dot(Q.T, A)
print("R:\n", R)

print("Check solution: QR\n", np.dot(Q, R))
```

```
A:
 [[1 0 2]
 [0 1 1]
 [2 0 1]]

there is one dependent column, namely 3rd one

Q:
 [[ 0.4472136   0.          0.89442719]
 [ 0.          1.          0.        ]
 [ 0.89442719  0.         -0.4472136 ]]
Q must be orthogonal. Let's check it out:
 [[ 1.00000000e+00  0.00000000e+00 -5.55111512e-17]
 [ 0.00000000e+00  1.00000000e+00  0.00000000e+00]
 [-5.55111512e-17  0.00000000e+00  1.00000000e+00]]
R:
 [[ 2.23606798e+00  0.00000000e+00  1.78885438e+00]
 [ 0.00000000e+00  1.00000000e+00  1.00000000e+00]
 [-1.11022302e-16  0.00000000e+00  1.34164079e+00]]
Check solution: QR
 [[1. 0. 2.]
 [0. 1. 1.]
 [2. 0. 1.]]
```

# Problem 12




$$u'' = Qu = u - 2\frac{ua^T}{a^Ta}a = (I - 2\frac{aa^T}{a^Ta})u$$

$$a = u - sign(u_{11})||u||e_1$$

## a)

$$v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$||v|| = \sqrt{1^2 + 2^2} = \sqrt{5}$$

$$u = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \sqrt{5}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 - \frac{1}{\sqrt{5}} \\ 2 \end{bmatrix}$$

$$u = \frac{u}{||u||}$$

not finished.

## b)

Lets remind ourselves that from $A = QR$ we could obtain $Q^TA = R$. Using Househloders reflaction we could form $R$ applying new reflectors to $A$. Let's do it.

not finished.