ALI exam 2020

In [4]:

```
# Student number:
```

In [40]:

```
import sympy as sp
import scipy as sc
init_printing()
from IPython.display import display, Latex, HTML, Math
import numpy as np
import pandas as pd
from sympy import *
from scipy.misc import derivative
```

Feel free to add cells if you need to. The easiest way to convert to pdf is to save this notebook as .html (File-->Download as-->HTML) and then convert/print this html file to pdf.

Assignment 1 (15%)

In [ ]:

```
# a)

# Since the matrix is diagonal, the sum of the eigenvalues is the same as the sum of th
e diagonal. The sum of the diagonal
# is 6 which means that the final eigenvalue is 8 since 8 -1 -1 = 6.
```

In [1]:

```
# b)

# A is in reduced form since
    # i)   All nonzero rows are above all rows of zeros.
    # ii)  Each leading entry of a row is in a column to the right of the leading entry
of the row above it.
    # iii) All entries in a column below a leading entry are zeros.
    # iv)  The leading entry in each nonzero row is 1.
    # v)   Each leading 1 is the only nonzero entry in its column.

# B is in echelon form since
    # i)   All nonzero rows are above all rows of zeros.
    # ii)  Each leading entry of a row is in a column to the right of the leading entry
of the row above it.
    # iii) All entries in a column below a leading entry are zeros.

# C is not in any form since
    # Each leading entry of a row is NOT in a column to the right of the leading entry
 of the row above it


# D is not in any form since
    # All nonzero rows are NOT above all rows of zeros.
```

In [ ]:

```
# c)

# AAT is m by m but its rank is not greater than n (all columns of AAT are combinations
of
# columns of A). Since n<m, AAT is singular.
```

In [ ]:

```
# d)

# det A = 0 because two rows are equal hence free variable hence non-invertible hence d
et A = 0.
```

In [ ]:

```
# e)

# A has two pivots which gives us dim Col A = 2 and since dim Col A + dim Nul A = 5 we
 find that dim Nul A = 3.
```

In [ ]:

```
# f)
# An inconsistent linear system in three variables, with a coefficient matrix of rank t
wo.

# A consistent linear system with three equations and two unknowns, with a coefficient
 matrix
# of rank one.

# A consistent linear system with three equations and two unknowns, with a coefficient
 matrix
# of rank larger than one

# A linear system of two equations in three unknowns, with an invertible coefficient ma
trix.

# A linear system in three variables, whose geometrical interpretation is three planes
 intersecting
# in a line.
```

An inconsistent linear system in three variables, with a coefficient matrix of rank two.

Here we need three equations not all proportional, that are incompatible: so for example the following system will do:

$$\begin{aligned} &x+y+z=4 \\ &y+z=5 \\ &x+2y+2z=10 \end{aligned}$$

This system is inconsistent, since the sum of the first two equations gives $x+2y+2z=9$ which is incompatible with the third equation. The coefficient matrix is not rank zero, since it is not zero. It is not rank one, since otherwise all the left-hand sides of the equations would be proportional. It is not rank three, since otherwise the coefficient matrix would be invertible and the system would have a solution. Therefore it is rank two, as required.

A consistent linear system with three equations and two unknowns, with a coefficient matrix of rank one.

The following system will do:

$$\begin{aligned} &x+y=4 \\ &2x+2y=8 \\ &3x+3y=12 \end{aligned}$$

The equations are all proportional, so the coefficient matrix is rank one. The system is consistent, since $x=y=2$ solves it.

A consistent linear system with three equations and two unknowns, with a coefficient matrix of rank larger than one

The following system will do

$$x + y = 4$$
$$x + 2y = 5$$
$$x + 3y = 6$$

The coefficient matrix is not rank one, since the left-hand sides of the equations are not proportional and is clearly not rank zero either. So the rank is larger than one, as required.

A linear system of two equations in three unknowns, with an invertible coefficient matrix.

This is impossible: the coefficient matrix is 2 by 3 so is not square and only square matrices can be invertible.

A linear system in three variables, whose geometrical interpretation is three planes intersecting in a line.

We need pivots in all columns so the following system would work:

$$x - 3y + 2z = 8$$
$$3x - 8y - 5z = 11$$
$$2x - 4y - 18z = -10$$

A simpler version would be the following: $x = 0, y = 0, x + y = 0$. Each of these equations represents a plane through the z-axis, which is their common intersection.

Assignment 2 (15%)

In [37]:

```
# a)

q = symbols('q')

A = Matrix([[3-2*q,1],[4,3+2*q]])
B = Matrix([[29,6],[24,125]])
que = solve(A**2-B, q)
display(Latex('$$q = {}$$'.format(que[0][0])))
```

$q = 4$

In [3]:

```
A = Matrix([[3-2*4,1],[4,3+2*4]])
A.T*(A**-1).T
```

Out[3]:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

b)

$$B^2B^TBB^{-1}\left(B^{-1}\right)^TB\left(B^{-1}\right)^2 = BBB^TBB^{-1}\left(B^{-1}\right)^TBB^{-1}B^{-1} = BBB^T\left(B^{-1}\right)^TB^{-1} = BBB^T(B$$

In all steps $BB^{-1} = I$ and $BB = B^2$ are used, and in the second last step we also use $\left(B^{-1}\right)^\top = \left(B^\top\right)^{-1}$

In [42]:

```
# c)
k = symbols('k')
A = Matrix([[2,3],[-1,1]])
B = Matrix([[1,9],[-3,k]])

kay = solve(A*B - B*A, k)
display(Latex('$$k = {}$$'.format(list(kay.values())[0])))
```

$$k = -2$$

Assignment 3 (10%)

In [4]:

```
# a)

A = Matrix([[1, 2, 5], [3, -2, 1], [2, 4, 10]])
B = A.rref()[0]
display(Latex('$$x_1 = {}$$'.format(B[0,2])))
display(Latex('$$x_2 = {}$$'.format(B[1,2])))
```

$$x_1 = 3/2$$

$$x_2 = 7/4$$

In [5]:

```
# b)
a = symbols('a')

A = Matrix([[1,1,1,-1],[1,2,a,2*a],[1,a,2,-2]])
L, U, perm = A.LUdecomposition()
display(solve(U[2,2], a))
display(solve(U[2,3], a))
display(Latex("When  α ≠ 0  and  α ≠ 2,  the linear system has three basic variables."
              " When  α = 0,  the system has two basic variable and one free variable."
              " (If  α = 2  the linear system has no solution)"))
```

$$[0,\ 2]$$

$$\left[0,\ \frac{1}{2}\right]$$

When $\alpha \neq 0$ and $\alpha \neq 2$, the linear system has three basic variables. When $\alpha = 0$, the system has two basic variable and one free variable. (If $\alpha = 2$ the linear system has no solution)

Assignment 4 (10%)

In [6]:

```
# a)
x, y, z = symbols('x y z')

# Note a required method is not stated, so we simply use the built in method.
A = Matrix([[x,y,z,1],[1,-2,3,1],[2,-3,1,1],[4,-6,3,1]])
display(Latex('det A = ${}$'.format(latex(A.det()))))
```

det A = $-8x - 6y - z - 1$

b)

$\det(2A) = 2^4 \det(A) = 48$ Rule: $\det(cA) = c^n \det(A)$

$\det\left(A^3\right) = (\det(A))^3 = 27$ Rule: Self explanatory

$\det\left(A^{-1}\right) = (\det(A))^{-1} = \frac{1}{3}$ Rule: Self explanatory

$\det\left(A^2B^3\right) = (\det(A))^2(\det(B))^3 = 3^2(-2)^3 = -72$ Rule: Self explanatory

$\det\left(A^3B^{-2}\right) = (\det(A))^3(\det(B))^{-2} = 3^3(-2)^{-2} = \frac{27}{4}$ Rule: Self explanatory

Assignment 5 (10%)

In [7]:

```
A = Matrix([[3,2,-2],[0,2,0],[0,1,3]])
A.eigenvects()
```

Out[7]:

$$\left[\left(2, \ 1, \ \left[\left[\begin{matrix}4\\-1\\1\end{matrix}\right]\right]\right), \ \left(3, \ 2, \ \left[\left[\begin{matrix}1\\0\\0\end{matrix}\right]\right]\right)\right]$$

In [ ]:

```
# A is not diagonalizable since the eigenvalue 3 has algebraic multiplicity 2 but geome
tric multiplicity 1
# (i.e. dim Nul (A-3*I) = 1).
```

In [8]:

```
B = Matrix([[-5,2,-1,3],[0,1,0,2],[0,0,1,2],[0,0,0,3]])
B.eigenvects()
```

Out[8]:

$$\left[\left(-5, \ 1, \ \left[\left[\begin{matrix}1\\0\\0\\0\end{matrix}\right]\right]\right), \ \left(1, \ 2, \ \left[\left[\begin{matrix}\frac{1}{3}\\1\\0\\0\end{matrix}\right], \ \left[\begin{matrix}-\frac{1}{6}\\0\\1\\0\end{matrix}\right]\right]\right), \ \left(3, \ 1, \ \left[\left[\begin{matrix}\frac{1}{2}\\1\\1\\1\end{matrix}\right]\right]\right)\right]$$

In [9]:

```
# We see that B is diagonalisable since the algebraic multiplicity is equal to the geom
etric multiplicity for all eigenvalues

P, D = B.diagonalize()
Pinv = P**-1

display(Math(r'P D P^{-1} = ' + latex(P) + latex(D) + latex(Pinv)))
display(Latex("Test:"))
display(P*D*Pinv)
display(B)
```

$$PDP^{-1} = \left[\begin{matrix}1 & 1 & -1 & 1\\0 & 3 & 0 & 2\\0 & 0 & 6 & 2\\0 & 0 & 0 & 2\end{matrix}\right]\left[\begin{matrix}-5 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 3\end{matrix}\right]\left[\begin{matrix}1 & -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3}\\0 & \frac{1}{3} & 0 & -\frac{1}{3}\\0 & 0 & \frac{1}{6} & -\frac{1}{6}\\0 & 0 & 0 & \frac{1}{2}\end{matrix}\right]$$

Test:

$$\left[\begin{matrix}-5 & 2 & -1 & 3\\0 & 1 & 0 & 2\\0 & 0 & 1 & 2\\0 & 0 & 0 & 3\end{matrix}\right]$$

$$\left[\begin{matrix}-5 & 2 & -1 & 3\\0 & 1 & 0 & 2\\0 & 0 & 1 & 2\\0 & 0 & 0 & 3\end{matrix}\right]$$

Assignment 6 (25%)

In [3]:

```python
# a)
x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
y = np.array([0, 26.4, 89.7, 186.0, 314.1, 477.3, 666.0, 883.5, 1141.2, 1413.3, 1715.1,
1715.1, 2527.6])

X1 = Matrix([ones(len(x), 1)]).row_join(Matrix(x)).row_join(Matrix(x**2)).row_join(Matr
ix(x**3))
X2 = Matrix(x).row_join(Matrix(x**2)).row_join(Matrix(x**3))
X3 = Matrix([ones(len(x), 1)]).row_join(Matrix(x)).row_join(Matrix(x**2))

display(Math(r'X_1 = ' + latex(X1) + r'\quad X_2 = ' + latex(X2) + r'\quad X_3 = ' + la
tex(X3) + r'\quad y = '
             + latex(Matrix(y))))
```

$$X_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \\ 1 & 6 & 36 & 216 \\ 1 & 7 & 49 & 343 \\ 1 & 8 & 64 & 512 \\ 1 & 9 & 81 & 729 \\ 1 & 10 & 100 & 1000 \\ 1 & 11 & 121 & 1331 \\ 1 & 12 & 144 & 1728 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 4 & 8 \\ 3 & 9 & 27 \\ 4 & 16 & 64 \\ 5 & 25 & 125 \\ 6 & 36 & 216 \\ 7 & 49 & 343 \\ 8 & 64 & 512 \\ 9 & 81 & 729 \\ 10 & 100 & 1000 \\ 11 & 121 & 1331 \\ 12 & 144 & 1728 \end{bmatrix} \quad X_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \\ 1 & 10 & 10 \\ 1 & 11 & 12 \\ 1 & 12 & 14 \end{bmatrix}$$

In [4]:

```
# b)
X1tX1 = X1.T*X1
X1ty = X1.T*Matrix(y)
Mat, _ = X1tX1.row_join(X1ty).rref()
B1 = Mat[:,-1]
display(Latex("$$y_1(t) = {}+{}t+{}t^2+{}t^3$$".format(round(B1[0],2), round(B1[1], 4), round(B1[2], 4), round(B1[3], 4))))

X2tX2 = X2.T*X2
X2ty = X2.T*Matrix(y)
Mat, _ = X2tX2.row_join(X2ty).rref()
B2 = Mat[:,-1]
display(Latex("$$y_2(t) = {}t+{}t^2+{}t^3$$".format(round(B2[0],2), round(B2[1], 4), round(B2[2], 4))))

X3tX3 = X3.T*X3
X3ty = X3.T*Matrix(y)
Mat, _ = X3tX3.row_join(X3ty).rref()
B3 = Mat[:,-1]
display(Latex("$$y_3(t) = {}+{}t+{}t^2$$".format(round(B3[0],2), round(B3[1], 4), round(B3[2], 4))))
```

$$y_1(t) = -23.02 + 44.1007t + 10.0985t^2 + 0.2384t^3$$

$$y_2(t) = 30.48t + 12.2699t^2 + 0.1371t^3$$

$$y_3(t) = -7.28 + 24.3108t + 14.3903t^2$$

In [5]:

```
# c)
display(Latex("$$e_1 = {}$$".format(round((Matrix(y)-X1*B1).norm(), 2))))
display(Latex("$$e_2 = {}$$".format(round((Matrix(y)-X2*B2).norm(), 2))))
display(Latex("$$e_3 = {}$$".format(round((Matrix(y)-X3*B3).norm(), 2))))
```

$$e_1 = 338.58$$

$$e_2 = 339.66$$

$$e_3 = 340.31$$

Clearly, the first model has the best fit for this specific data (although it may be overfitting but that is not part of this course!)

In [42]:

```
# d)
y45 = B1[1]+2*B1[2]*4.5+3*B1[3]*4.5**2
display(Latex("$$y_3'(4.5) = {} \\ m/s = {} \\ km/h$$".format(round(y45, 2), round(y45*3.6, 2))))

# or

def f(x):
    return B1[0] + B1[1]*x+B1[2]*x**2+B1[3]*x**3
display(Latex("$$y_3'(4.5) = {} \\ m/s = {} \\ km/h$$".format(round(derivative(f,4.5), 2), round(derivative(f,4.5)*3.6, 2))))
```

$$y_3'(4.5) = 149.47 \ m/s = 538.10 \ km/h$$

$$y_3'(4.5) = 149.71 \ m/s = 538.96 \ km/h$$

In [ ]:

```
# The small difference is due to rounding issues in Python
```

Assignment 7 (15%)

In [15]:

```
# a)
A = Matrix([[2,0,0],[0,2,1],[0,1,2],[0,0,0]])
AtA = A.T*A
vecs1 = AtA.eigenvects()

s1 = sqrt(vecs1[2][0])
s2 = sqrt(vecs1[1][0])
s3 = sqrt(vecs1[0][0])

v1 = vecs1[2][2][0].normalized()
v2 = vecs1[1][2][0].normalized()
v3 = vecs1[0][2][0].normalized()

A = Matrix([[2,0,0],[0,2,1],[0,1,2],[0,0,0]])
AAt = A*A.T
vecs2 = AAt.eigenvects()
vecs2

u1 = vecs2[3][2][0].normalized()
u2 = vecs2[2][2][0].normalized()
u3 = vecs2[1][2][0].normalized()
u4 = vecs2[0][2][0].normalized()

U = u1.row_join(u2).row_join(u3).row_join(u4)
S = diag(s1, s2, s3).col_join(zeros(1,3))
V = v1.row_join(v2).row_join(v3)
Vt = V.T

display(Math('U \Sigma V^T = {}{}{}'.format(latex(U), latex(S), latex(Vt))))
display(Latex("Test:"))
display(U*S*Vt)
display(A)
```

$$U\Sigma V^T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

Test:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

In [16]:

```
# b)

# The rank r of A is three since this is the number of nonzero singular values.

# The first three columns of U is an orthonormal basis for the column space

# The last column of U is an orthonormal basis for the left nullspace, i.e. the nullspa
ce of A transpose

# The first three columns of V is is an orthonormal basis for the row space (= Column s
pace of A transpose)

# Since V only has three columns, the null space of A must be empty as this would have
 been the remaining columns of V.


display(Math(r'Col A = ' + latex(U[:, 0]) + ' , ' + latex(U[:, 1]) + ' , ' + latex(U[:,
2])))
display(Math(r'Nul A^T = ' + latex(U[:, -1])))
display(Math(r'Col A^T = Row A = ' + latex(V[:, 0]) + ' , ' + latex(V[:, 1]) + ' , ' +
latex(V[:, 2])))
display(Math(r'Nul A = ' + latex([])))
```

$$Col A = \begin{bmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} , \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix}$$

$$Nul A^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$Col A^T = Row A = \begin{bmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} , \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

$$Nul A = []$$