Department of Mathematics and Computer Science
University of Southern Denmark, Odense

February 11, 2020
Marco Chiarandini

DM545/DM871 – Linear and integer programming

Sheet 0, Spring 2020 [pdf format]

Solution:
Included.

The assignment is meant to be a review of elements of Python Programming and Linear Algebra that will be needed in this course. You can refresh or deepen your knowledge of Python with this series of tutorials:

https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU

You need to have a working installation of Python and `numpy` and you must be able to write scripts and to execute them. You are definitely expected to be able to do this for the lab class on week 8.
You can work at the exercises below in interactive way. It is strongly recommended that you read the following tutorial on Jupyter (aka IPython) `numpy` and Linear Algebra:

http://www.imada.sdu.dk/~marco/DM871/Resources/Ipython/Tutorial.html.

Exercise 1

Write a list of the first 100 numbers in which any number divisible by three is replaced by the word "fizz" and any divisible by five by the word "buzz". Numbers divisible by both become "fizz buzz".

Solution:

```
for num in xrange(1,100):
  if num % 5 == 0 or num % 3 == 0:
    print("fizzbuzz")
  elif num % 5 == 0:
    print("buzz")
  elif num % 3 == 0:
    print("fizz")
  else:
    print(num)
```

Exercise 2 Data Types

Revise the difference between the main data types in Python: list, tuples, dictionaries and sets. Write an example for each of them in which you define and initialize a variable for each type and then print the content looping through the elements of the variable.

Solution:

```
# lists
my_list = [10,20,30,40,50,60]
for i in my_list:
  print(i)

# tuples
my_tuple = (1,2,3,4,5,6,7,8,9)
```

```
for i in my_tuple:
  print(i)

# dictionaries
my_dict = {'name': 'Esau', 'age': 2, 'occupation': 'My dog'}
for key,val in my_dict.iteritems():
  print("My {} is {}".format(key,value))

# set
my_set = {20,30,40,50,60,20,30,40,50}
for i in my_set:
  print(i)
```

Exercise 3 Python: One liner quizzes

Write a one line Python code for the following tasks:

a) Construct the set $S = \{x \in \mathbb{R} \mid x \geq 0 \wedge x \bmod 3 \equiv 1\}$

Solution:

```
S = {x for x in range(50) if x % 3 == 1}
```

b) Using list comprehension make a list for $\{(i, j) \mid i \in \{1, 2, 3, 4\}, j \in \{5, 7, 9\}\}$

Solution:

```
[(i,j) for i in (range(4)+1) for j in [5,7,9]]
```

c) Calculate the inverse of a function or the index function for an invertible function (ie, bijective = injective + surjective) given in form of a dictionary.

Solution:

```
{d[k]:k for k in d}
{v:k for k in d.keys() for v in d.values}
{v:k for k,v in d.items()}
```

d) What is the result of the following lines?

```
map(lambda x: x%3, range(5))
filter(lambda x: x%2==0, range(5))
```

(In Python 3.x, you have to enclose those lines in the list constructor `list()`.)

Exercise 4 Matrix Calculus in basic Python

The basic data structures in Pyton are lists, tuples, sets and dictionaries. Vectors and matrices can be implemented in Python as lists. How?

a) Generate a couple of numerical examples for vectors and matrices. Experiment with the operators + and ∗. Do they yield the same result as expected from linear algebra?

b) Write a function for the sum of two vectors using list comprehension.

c) Write a function for the multiplication of a vector by a scalar.

d) Write a function for the sum of two matrices using list comprehension.

e) Write a function for the multiplication of a matrix by a scalar.

f) Write a function for the multiplication of two matrices not necessarily square. (Raise a ValueError exception if the size of the matrices is not compliant.)

Solution:

It is important to note that the operators + and * are overloaded for list. They concatenate or replicate the two lists, respectively. This is definetly not what we learned to be the definition of those operations.

```
def sumVec(a,b): return [a[i]+b[i] for i in range(len(a))]
def multScalVec(alpha,a): return [alpha * a[i] for i in range(len(a))]


def sumMat(M,N)
  result = [[0 for x in range(len(N[0]))] for y in range(len(M))]
  for i in range(len(M)):
      for j in range(len(N[0])):
          result[i][j]=M[i][j]+N[i][j]
  print(result)


def mult(M,N):
    if (len(M[0])!=len(N))
        raise ValueError("The inner size of the martices does not match")

    result = [[0 for x in range(len(N[0]))] for y in range(len(M))]

    for i in range(len(M)):
        for j in range(len(N[0])):
            for k in range(len(N)):
                result[i][j] = result[i][j] + M[i][k] * N[k][j]

    return result

M=[[1,2,3],[1,2,3]]
N=[[1,2],[1,2],[1,2]]

try mult(M,N) except ValueError: print("Oops, a ValueError occurred")

def printMatrix(M):
    for row in M:
        print(["%3.0f" % a for a in row])
    print("\n")
```

Exercise 5 Matrix Calculus in `numpy` and `scipy`

The modules `numpy` and `scipy` make available another data structure in Python, the 'array' type. This exercise guides you to the discovery of how operators are overloaded for the 'array' type module. You can read more about `numpy` and `scipy` from the Tutorial linked above and from the section "Literature: Links" of the course web page.
Generate in Python two matrices A and B of size 3×2 and 2×4, respectively, made of integers numbers randomly drawn from the interval $[1, \ldots, 10]$. Calculate the following results, first by hand and then checking the correctness of your answer in Python:

a) $A + B, A - B$

b) $A \cdot B$

c) A/B

[In IPython and Jupyter it is possible from command line to ask for completion via tab. This can be used to explore which functions are available for a given module. Try for example to type

```
import numpy as np
np.
```

followed by a tab. You should see a list of available functions. Among them there are two submodules that will be useful for us: `random` and `linalg`. The first implements a function to generate random numbers and matrices. The second implements functions from linear algebra. It is possible to get a manual for each function by following the function with a question mark. For example: `np.random.randint?`.]

Solution:

```
A=np.random.randint(1,10,(3,2))
B=np.random.randint(1,10,(2,4))
A+B
A-B
A*B
```

All these produce an error because the `+,-` operators, as in linear algebra perform an element-wise addition and subtraction, which requires the two matrices to have exactly the same size.
The `*` operator performs also an element-wise multiplication, which require the two matrices to have exactly the same size. However, contrary to addition and multiplication, this operation is not defined element-wise in linear algebra.
The correct way to obtain the matrix product is:

```
np.dot(A,B)
```

The division by a matrix is not defined in linear algebra. In Python it does an element-wise operations if the matrices have the same size.

Exercise 6

Solve by Gaussian elimination the following system of linear equations $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 2 & 4 & 1 & 1 \\ 2 & 1 & 2 & 2 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$

First carry out the calculations by hand and then try using Python.

Solution:

```
A=np.array([[3,1,1,1],
  [2, 4, 1, 1],
  [2,1,2,2]])
b=np.array([ 2, 2, 1])

# np.linalg.solve(A,b)
print np.linalg.matrix_rank(A)
AA=np.column_stack([A,b])
AA=column_stack([A,b])
AA[0,:] = f(1,3) * AA[0,:] # remember indices start from 0
AA[1,:] = -2 * AA[0,:] + AA[1,:]
AA[2,:] = -2 * AA[0,:] + AA[2,:]
printm(AA)
AA[1,:] = f(3,10)* AA[1,:]
AA[2,:] += -f(1,3) * AA[1,:]
printm(AA)
AA[2,:] = f(10,13) * AA[2,:]
printm(AA)
```

```
AA[1,:] += -f(1,10) * AA[2,:]
AA[0,:] += -f(1,3) * AA[2,:]
printm(AA)
AA[0,:] += -f(1,3) * AA[1,:]
printm(AA)
```

After elementary row operations the augmented matrix looks like:

```
[1, 0, 0, 0, 9/13]
[0, 1, 0, 0, 3/13]
[0, 0, 1, 1, -4/13]
```

From here we can write directly the solution:

$$\begin{aligned} x_1 &= 9/13 \\ x_2 &= 3/13 \\ x_3 &= -4/13 - t \\ x_4 &= t \end{aligned}$$

which in vector notation is equivalent to

$$\mathbf{x} = \begin{bmatrix} 9/13 \\ 3/13 \\ -4/13 \\ 0 \end{bmatrix} + t \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

Hence the solution subspace is expressed by an affine combination.

Exercise 7

Write a one line description of the methods you know to compute the inverse of a square matrix.

Solution:

- using cofactors for the adjoint matrix and dividing by the determinant
- by row reduction of $[A|I]$

Exercise 8

Calculate by hand the inverse of

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

and check your result with the function `numpy.linalg.inv`.

Solution:

$$\begin{bmatrix} 0. & 1. & -1.5 \\ 1. & -3. & 4. \\ 0. & 0. & 0.5 \end{bmatrix}$$

Exercise 9

Use Cramer's rule to express the solution of the system $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 0 & 3 \\ 0 & 0 & 2 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix}$$

Solution:

$$x_i = \frac{\det(A_i)}{\det(A)}$$

where A_i is the matrix obtained from A by replacing the ith column with the vector b.

$$x_1 = -4.5 \qquad x_2 = 14 \qquad x_3 = 1.5$$

Exercise 10

A total of $6,300 was invested in two accounts. Part was invested in obligations at 0.045 annual interest rate and part was invested in a money market fund at 0.0375 annual interest rate. If the total simple interest for one year was $267.75, then how much was invested in each account?

Solution:

Let x represent the amount invested at 4 and 1/2% annual interest, that is, at 0.045. Let y represent the amount invested at 3 and 3/4% annual interest, that is, at 0.0375.
The total amount in both accounts can be expressed as

$$x + y = 6,300$$

To set up a second equation, use the fact that the total interest was $267.75. The interest for one year is the interest rate times the principal. Use this to add the interest in both accounts.

$$0.045x + 0.0375y = 267.75$$

These two equations together form a linear system.
Solving we obtain that $4,200 was invested at 4 and 1/2% and $2,100 was invested at 3 and 3/4%.

Exercise 11

A construction company produces five different products: p_1, p_2, p_3, p_4, p_5, using five resources: metal, concrete, plastic, water, electricity.
The amount of resources consumed for producing one unit of products is described by the following python dictionaries:

```
p_1={"metal":0, "concrete":1.3, "plastic":0.2, "water":.8, "electricity":.4}
p_2={"metal":0, "concrete":0, "plastic":1.5, "water":.4, "electricity":.3}
p_3={"metal":.25, "concrete":0, "plastic":0, "water":.2, "electricity":.7}
p_4={"metal":0, "concrete":0, "plastic":.3, "water":.7, "electricity":.5}
p_5={"metal":1.5, "concrete":0, "plastic":.5, "water":.4, "electricity":.8}
```

How much metal is consumed if the company decides to produce the following mix of products: 10, 9, 12, 13, 11? Express the calculation asked by the exercise as a linear algebra operation involving matrices and vectors. Then calculate the numerical answer using Python and numpy. (For example, you can transform the dictionary above into a numpy array as follows: `a=np.array(list(p_1.values()))`.)

Solution:

We can find the answer by scalar product between the row "material" and the vector of production. More precisely, let's indicate with $\mathbf{p}$ the production vector, with A the matrix of technological coefficients and with $\mathbf{a}_j$ a column j of A. Hence the answer is given by $\mathbf{p} \cdot \mathbf{a}_j$

```
#%%
import numpy as np
p_1={"metal":0, "concrete":1.3, "plastic":0.2, "water":.8, "electricity":.4}
p_2={"metal":0, "concrete":0, "plastic":1.5, "water":.4, "electricity":.3}
p_3={"metal":.25, "concrete":0, "plastic":0, "water":.2, "electricity":.7}
p_4={"metal":0, "concrete":0, "plastic":.3, "water":.7, "electricity":.5}
p_5={"metal":1.5, "concrete":0, "plastic":.5, "water":.4, "electricity":.8}
```

```
p = np.array([10,9,12,13,11])
#%%

A = np.array([list(p_1.values()),
              list(p_2.values()),
              list(p_3.values()),
              list(p_4.values()),
              list(p_5.values())]
              )
np.dot(A.T,p)
```

Exercise 12

Consider again our construction company that produces five different products: p_1, p_2, p_3, p_4, p_5, using five resources: metal, concrete, plastic, water, electricity.
The amount of resources consumed for producing one unit of products is described by the python dictionaries defined in the previous exercise.
Suppose now that you do not know how much of each prodcut is produced but that you know how much of each material is used. Can you determine from this information the amount of each product that is produced? Will the solution be unique?

Solution:

The given values are the b_i terms of the system of equalities $A\mathbf{x} = \mathbf{b}$. Hence we can check whether the system is solvable by considerations on the matrix A. We know that a condition for the system to be solvable is that the rank of A is the same as the rank of the augmented matrix of the system. We also know that there will be a unique solution when the matrix A has the same rank as number of variables.

```
%%
np.linalg.matrix_rank(A)
5
```

Hence since the rank of A is 5 and there are 5 materials (the rows of the matrix A) then the augmented matrix has also rank 5 and there the system is consistent. Moreover since there are 5 products then there are no free variables and hence the solution is unique.

Exercise 13

Given two points in the Cartesian plane $\mathbb{R}^2$, $A = (1, 2)$ and $B = (3, 4)$ write the vector parametric equation and the Cartesian equation of the line that passes through them. Express the vector equation as an affine combination of the two points.

Solution:

The vector equation is an affine combination of the two points:

$$\mathbf{x} = [1, 2]^T + t([3, 4]^T - [1, 2]^T), \forall t \in \mathbb{R}^2$$

To find a, b, c such that $ax + by + c = 0$ describes the line we can rewrite the equation above as

$$[x, y]^T = [1, 2]^T + t([1, 2]^T), \forall t \in \mathbb{R}^2$$

we then eliminate t by substituting in the two equations.

Exercise 14

Express the segment in $\mathbb{R}^2$ between the points $A = (1, 2)$ and $B = (3, 4)$ as a convex combination of its extremes.

Solution:

$$\{\alpha[1, 2]^T + \beta[3, 4]^T \mid \alpha, \beta \in \mathbb{R}, \alpha, \beta \geq 0, \alpha + \beta = 1\}$$

Exercise 15

Write a generic vector parametric equation and a generic Cartesian equation of a plane in $\mathbb{R}^3$.

Solution:

$$\mathbf{x} = \mathbf{p} + s\mathbf{v} + t\mathbf{w}, \quad x, t \in \mathbb{R}$$
$$ax + by + cz + d = 0$$

Exercise 16

Write a generic Cartesian equation of an hyperplane in $\mathbb{R}^n$ that does not pass through the origin.

Solution:

$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$.

Exercise 17

Prove that the following vectors in $\mathbb{R}^3$ linearly independent?
- $[6, 9, 5]^T$
- $[5, 5, 7]^T$
- $[2, 0, 7]^T$

Solution:

We need to solve homogeneous system $A\mathbf{x} = \mathbf{0}$, where the matrix A has the three vectors forming its columns. The matrix A has $\det(A) \neq 0$ and rank 3. Therefore the only solution to the homogeneous system is the trivial solution $\mathbf{0}$. The three column vectors are therefore linearly independent.

```
A=np.array([[6, 9, 5],[5, 5, 7],[2, 0, 7]]).T
np.linalg.det(A)
np.linalg.matrix_rank(A)
```