# Laboratory 3: Linear Algebra

Angelene Leow, 23162167

## List of Problems

- Problem One: Pollution Box Model
- Problem Two: Condition number for Dirichlet problem

In [103]:
```
import context
import pdb
# import the quiz script
from numlabs.lab3 import quiz3
# import image handling
from IPython.display import Image
import numpy as np
from numpy import linalg as LA
import matplotlib.pyplot as plt
from scipy.sparse import diags
```

**Problem One**

Consider a very simple three box model of the movement of a pollutant in the atmosphere, fresh-water and ocean. The mass of the atmosphere is MA (5600 x $10^{12}$ tonnes), the mass of the fresh-water is MF (360 x $10^{12}$ tonnes) and the mass of the upper layers of the ocean is MO (50,000 x $10^{12}$ tonnes). The amount of pollutant in the atmosphere is A, the amount in the fresh water is F and the amount in the ocean is O.

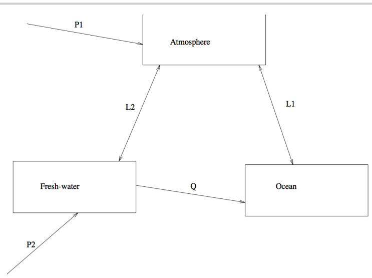
The pollutant is going directly into the atmosphere at a rate P1 = 1000 tonnes/year and into th efresh-water system at a rate P2 = 2000 tonnes/year. The pollutant diffuses between the atmosphere and ocean at a rate depending linearly on the difference in concentration with a diffusion constant L1 = 200 tonnes/year. The diffusion between the fresh-water system and the atmosphere is faster as the fresh water is shallower, L2 = 500 tonnes/year. The fresh-water system empties into the ocean at the rate of Q = 36 x $10^{12}$ tonnes/year. Lastly the pollutant decays (like radioactivity) at a rate L3 = 0.05 /year.

See the graphical presentation of the cycle described above in Figure Box Model Schematic for Problem 1.

- a) Consider the steady state. There is no change in A, O, or F. Write down the three linear governing equations. Write the equations as an augmented matrix. Use Octave to find the solution.
- b) Show mathematically that there is no solution to this problem with L3 = 0. Why, physically, is there no solution?
- c) Show mathematically that there is an infinite number of solutions if L3 = 0 and P1 = P2 = 0. Why, physically?
- d) For part c) above, what needs to be specified in order to determine a single physical solution. How would you put this in the matrix equation.

In [104]:
```
Image(filename='images/C_cycle_problem.png', width='30%')
```

Out[104]:



**Answer to 1(a):**

Equation to three isolated systems:

(1)$\frac{dA}{dt} = P_1 + L_1(\frac{O}{M_o} - \frac{A}{M_A}) + L_2(\frac{F}{M_F} - \frac{A}{M_A}) - L_3A$

(2)$\frac{dO}{dt} = Q \cdot \frac{F}{MF} + L_1(\frac{A}{MA} - \frac{O}{M_o}) - L_3O$

(3)$\frac{dF}{dt} = P_2 - Q \cdot \frac{F}{MF} + L_2(\frac{A}{MA} - \frac{F}{MF}) - L_3F$

Since $\frac{dF}{dt}, \frac{dA}{dt}, \frac{dO}{dt} = 0$ at steady state, the equations become:

(1)$-P_1 = \frac{L_1}{M_o} \cdot O - (\frac{L_1}{M_A} + \frac{L_2}{M_A} + L_3) \cdot A + \frac{L_2}{M_F} \cdot F$

(2) $0 = -(\frac{L_1}{M_o} + L_3) \cdot O + \frac{L_1}{M_A} \cdot A + \frac{Q}{M_F} \cdot F$

(3)$-P_2 = 0 + \frac{L_2}{M_A} \cdot A - (\frac{Q}{M_F} + \frac{L_2}{M_F} + L_3) \cdot F$ Rearraging to an augmented matrix:

$$\begin{array}{cccc} O & A & F & \\ \end{array}$$

$$\left[\begin{array}{ccc|c} \frac{L_1}{MO} & \frac{-(L_1+L2)}{MA} - L_3 & \frac{L2}{MF} & -P_1 \\ -(\frac{L1}{MO} + L_3) & \frac{L_1}{MA} & \frac{Q}{MF} & 0 \\ 0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} - L_3 & -P_2 \end{array}\right]$$

In [117]:
```
ma = 5600e12 #mass of atm
mf = 360e12  #mass of fresh water
mo = 50000e12 #mass of upper ocean layer
P1 = 1000
P2 = 2000
L1 = 200
L2 = 500
L3 = 0.05
q= 36e12 #water per year, s^(-1)

mat1 = np.array (([L1/mo,-(L1+L2)/ma-L3,L2/mf],
                [-L1/mo-L3,(L1/ma),q/mf],
                [0,L2/ma,-(L2+q)/mf-L3]))
mat2 = ([-P1],[0],[-P2])
mat_solve = np.linalg.solve(mat1,mat2)
mat_solve
print('Solution to steady state matrix: 0 =' +str(mat_solve[0]))
print('Solution to steady state matrix: A =' +str(mat_solve[1]))
print('Solution to steady state matrix: F =' +str(mat_solve[2]))
```

```
Solution to steady state matrix: 0 =[26666.66666646]
Solution to steady state matrix: A =[20000.00000032]
Solution to steady state matrix: F =[13333.33333322]
```

***Answer to1(b):***

In [116]:
```
ma = 5600e12 #mass of atm
mf = 360e12  #mass of fresh water
mo = 50000e12 #mass of upper ocean layer
P1 = 1000
P2 = 2000
L1 = 200
L2 = 500
L3 = 0
q= 36e12 #water per year, s^(-1)

mat1 = np.array (([L1/mo,-(L1+L2)/ma-L3,L2/mf],
                [-L1/mo-L3,(L1/ma),q/mf],
                [0,L2/ma,-(L2+q)/mf-L3]))
mat2 = ([-P1],[0],[-P2])
mat_solve = np.linalg.solve(mat1,mat2)
mat_solve
print('Solution to steady state matrix =' +str(mat_solve))
```

```
Solution to steady state matrix =[[7.56604737e+33]
 [2.42113516e+32]
 [2.16172782e+20]]
```

for $L_3 = 0$

$$
\begin{array}{cccc}
O & A & F & \\
\end{array}
$$

$$
\left[\begin{array}{ccc|c}
\frac{L_1}{MO} & \frac{-(L_1+L_2)}{MA} & \frac{L_2}{MF} & -P_1 \\
-\frac{L1}{MO} & \frac{L_1}{MA} & \frac{Q}{MF} & 0 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & -P_2
\end{array}\right]
$$

$$
E_{i2} + E_{i3} \rightarrow \left[\begin{array}{ccc|c}
\frac{L_1}{MO} & \frac{-(L_1+L_2)}{MA} & \frac{L_2}{MF} & -P_1 \\
-\frac{L_1}{MO} & \frac{L_1+L_2}{MA} & \frac{-L_2}{MF} & -P_2 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & -P_2
\end{array}\right]
$$

$$
E_{i2} - E_{i1} \rightarrow \left[\begin{array}{ccc|c}
0 & 0 & 0 & -P_2 - P_1 \\
-\frac{L_1}{MO} & \frac{L_1}{MA} & \frac{Q}{MF} & -P_2 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & -P_2
\end{array}\right]
$$

Based on $E_{i1}$, we can see that $0 = -(P_2 + P_1)$, which is impossible, therefore there is no possible solution. Determinant of tha matrix is 0 which shows no possible solution as well. Physically we know that a balanced steady state equation must have inputs and outputs to balance out. If L3 is removed from all three equations, there is a possiblity that $\frac{dO}{dt}, \frac{dA}{dt}, \frac{dF}{dt} \geq 0$ , which gives $\frac{dS}{dt} \geq 0$ if $\frac{dS}{dt} = \frac{dO}{dt} + \frac{dA}{dt} + \frac{dF}{dt}$.

Mathematically the solution blows up.

***Answer to 1(c):*** $P_1 = 0, P_2 = 0, L3 = 0$

$$
\begin{array}{ccc}
O & A & F
\end{array}
$$

$$
\left[\begin{array}{ccc|c}
\frac{L_1}{MO} & \frac{-(L_1+L2)}{MA} & \frac{L2}{MF} & 0 \\
-\frac{L_1}{MO} & \frac{L_1}{MA} & \frac{Q}{MF} & 0 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & 0
\end{array}\right]
$$

$$
E_{i2}+E_{i3} \rightarrow \left[\begin{array}{ccc|c}
\frac{L_1}{MO} & \frac{-(L_1+L_2)}{MA} & \frac{L2}{MF} & 0 \\
-\frac{L1}{MO} & \frac{L_1+L_2}{MA} & \frac{-L_2}{MF} & 0 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & 0
\end{array}\right]
$$

$$
E_{i2}-E_{i1} \rightarrow \left[\begin{array}{ccc|c}
0 & 0 & 0 & 0 \\
-\frac{L_1}{MO} & \frac{L_1}{MA} & \frac{Q}{MF} & 0 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & 0
\end{array}\right]
$$

$$
\begin{array}{c}
E_{i1} \leftrightarrow E_{i2} \\
E_{i2} \leftrightarrow E_{i3}
\end{array}
\left[\begin{array}{ccc|c}
\frac{L_1}{MO} & \frac{-L_1}{MA} & \frac{Q}{MF} & 0 \\
0 & \frac{L_2}{MA} & -\frac{(L_2+Q)}{MF} & 0 \\
0 & 0 & 0 & 0
\end{array}\right]
$$

Mathematically, We know that A depends on F, O depends on A and F. Hence F can be any arbituary value to satisfy A and O. There are many infinite solutions. Physically, if our main source of pollutant input (P1,P2) and output (L3) are zero. $\frac{dS}{dt} = 0$. There are many ways (alternating directions of flow) for L1,L2 to retain total equilibrium for A,O and F in a steady state.

***Answer to 1(d):***

> We would need to specify a value for one of the variables A/O/F to get a unique solution. I set A = 5000 To determine a particular solution, the number of initial conditions must match the number of constants in the general solution. If we can reduce the matrix into echelon form, we can find a unique set of solutions for O,A and F.

$$
\begin{array}{cccc}
O & A & F & \\
\left[\begin{array}{c} \frac{L_1}{MO} \\ -\frac{L_1}{MO} \\ 0 \\ 0 \end{array}\right. &
\begin{array}{c} \frac{-(L_1+L2)}{MA} \\ \frac{L_1}{MA} \\ \frac{L_2}{MA} \\ 1 \end{array} &
\begin{array}{c} \frac{L2}{MF} \\ \frac{Q}{MF} \\ -\frac{(L_2+Q)}{MF} \\ 0 \end{array} &
\left|\begin{array}{c} 0 \\ 0 \\ 0 \\ 5000 \end{array}\right]
\end{array}
$$

**Problem Two**

> a) Using Python, compute the condition number for the matrix $A_1$ from Equation Differential System Matrix for several values of $N$ between 5 and 50. ( **Hint:** This will be much easier if you write a small Python function that outputs the matrix $A$ for a given value of $N$.)
>
> c) Another way to write the system of equations is to substitute the boundary conditions into the equations, and thereby reduce size of the problem to one of $N-1$ equations in $N-1$ unknowns. The corresponding matrix is simply the $N-1$ by $N-1$ submatrix of $A_1$ from Equation Differential System Matrix
>
> $$A_2 = \begin{bmatrix} -2 & 1 & 0 & \dots & & & 0 \\ 1 & -2 & 1 & 0 & \dots & & \\ 0 & 1 & -2 & 1 & 0 & \dots & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & & & 0 \\ & & \dots & 0 & 1 & -2 & 1 \\ 0 & & & \dots & 0 & 1 & -2 \end{bmatrix}$$
>
> Does this change in the matrix make a significant difference in the condition number?

## Answer 2

a)

$$A_1(N=5) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

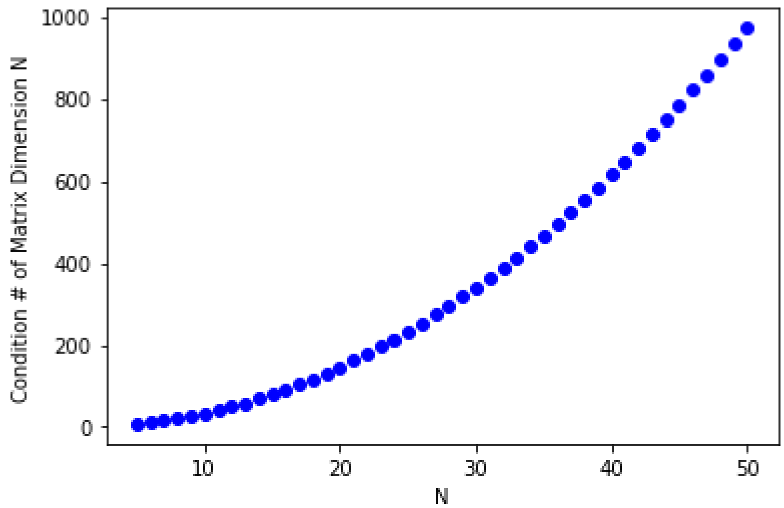
In [5]:

```
a1 = np.array([[1,0,0,0,0,],[1,-2,1,0,0],[0,1,-2,1,0],[0,0,1,-2,1],[0,0,
0,0,1]])
cond_mat = LA.cond(a1)
print('Condition number for matrix A1 of dim = 5 is ' +str(cond_mat))
```

```
Condition number for matrix A1 of dim = 5 is 7.643905814828506
```

In [322]:
```
def condMat(num):
    # initialize matrix of NxN dimension
    y = np.zeros((num,num))

    #Set first and last rows of A
    y[0][0] =1
    y[num-1][num-1] = 1

    #variable 'set' represents numerator of the discrete differential eq
uation
    set = [1,-2,1]

    for ii in range (1,num-1):
        y[ii][ii-1:ii+2] = set
    cond_Num = LA.cond(y)
    return cond_Num

for k in range(5,51):
    plt.plot(k,condMat(k),'bo')
    plt.xlabel('N')
    plt.ylabel('Condition # of Matrix Dimension N')
    print(k,condMat(k))
```

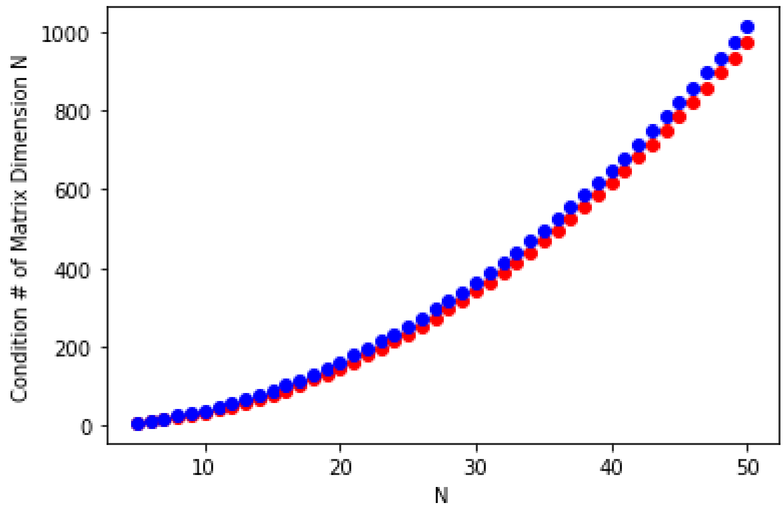
```
5 7.643905814828506
6 10.99460031976541
7 15.23311898662832
8 20.336526151680207
9 26.287086421268775
10 33.07381141107933
11 40.689754983885095
12 49.13030203691636
13 58.3922524542119
14 68.4733105887401
15 79.37178441258116
16 91.0863987713163
17 103.61617450397355
18 116.96034750138841
19 131.11831298322528
20 146.08958623336363
21 161.8737743750406
22 178.47055572507102
23 195.87966445685493
24 214.1008790481355
25 233.13401346970113
26 252.9789103874517
27 273.6354358625691
28 295.1034751794937
29 317.382929532326
30 340.473713371065
31 364.37575225987496
32 389.08898113610235
33 414.613342885362
34 440.9487871681461
35 468.09526944756664
36 496.0527501794451
37 524.8211941338741
38 554.4005698241633
39 584.7908490240067
40 615.9920063572671
41 648.0040189479296
42 680.8268661203746
43 714.4605291417475
44 748.904990999362
45 784.1602362082386
46 820.2262506436945
47 857.1030213954183
48 894.7905366401139
49 933.2887855294589
50 972.5977580919995
```



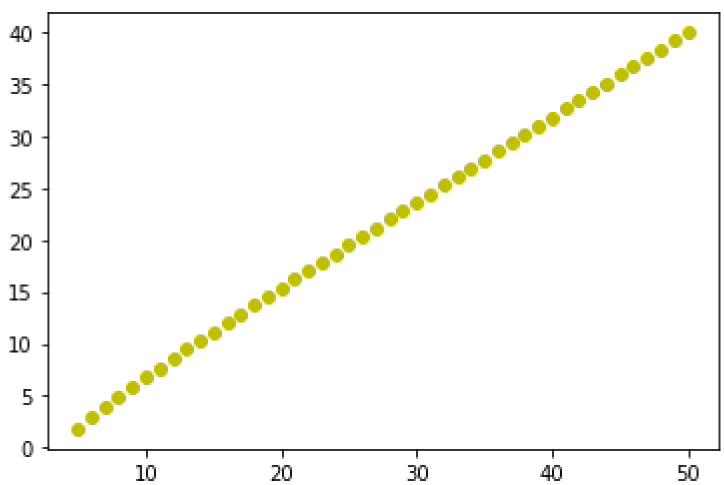

In [305]:
```
def condMat_2c(num):
    # initialize matrix of NxN dimension
    y = np.zeros((num-1,num-1))

    #Set first and last rows of A
    y[0][0] =-2
    y[0][1] =1
    y[num-2][num-3] = 1
    y[num-2][num-2] = -2

    #variable 'set' represents numerator of the discrete differential eq
uation
    set = [1,-2,1]

    for ii in range (1,num-2):
        y[ii][ii-1:ii+2] = set
    cond_Num = LA.cond(y)


    return cond_Num
```

$$A_2(N=5) = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

In [318]:
```
for k in range(5,51):
    plt.plot(k,condMat(k),'ro')
    plt.plot(k,condMat_2c(k),'bo')
    plt.xlabel('N')
    plt.ylabel('Condition # of Matrix Dimension N')
```




In [321]:
```
for k in range (5,51):
    diff = condMat_2c(k)-condMat(k)
    plt.plot(k,diff,'yo')
```




**Answer 2(c),**

> We observe that by reducing the dimension to $N - 1$, the condition number is increased but not by much. There is no relation between size of matrix and condition number.