Aproximación de un Polinomio Característico

Luis Eduardo Robles Jiménez

0224969

Input

In [50]:

```
#matrix = np.array([[3, 1, 5], [3, 3, 1], [4, 6, 4]]) #1, -10, 4, -40
matrix = np.array([[3, 2, 4], [2, 0, 2], [4, 2, 3]]) #1, -6, -15, -8
```

Method

In [53]:

```
Leverrier_Faddeev(matrix)
```

```
Matrix:

 [[3 2 4]
 [2 0 2]
 [4 2 3]]
```

Out[53]:

```
array([  1.,  -6., -15.,  -8.])
```

Leverrier Faddeev

In [52]:

```
def Leverrier_Faddeev(A):
    print("Matrix:\n\n", A, "\n\n")
    n = A.shape[0]
    b, B, i = np.empty(n+1), np.empty((n+1, n, n)), np.identity(n)
    b[n], B[0] = 1, np.zeros((n, n))
    for k in range(1, n+1):
        B[k] = (A @ B[k-1]) + (b[n-k+1] * i)
        b[n-k] = -np.trace(A @ B[k])/k
    return np.flip(b)
```

Run first

In [7]:

```
import numpy as np
from sympy import *
x, lmbd = symbols("x"), symbols("lambda")
```