Universidad de las Fuerzas Armadas

Métodos Numéricos

Actividad 1

Apellidos: Guzmán Cajas

Nombres: Diego Alejandro

NRC: 3657

1) Cree un repositorio en GitHub para los proyectos que se desarrollarán en el semestre. El repositorio debe estar asociado a su correo institucional. Los ejercicios de la actividad deben estar en el repositorio.

https://github.com/GuzmanDiegoEspe/MetodosNumericos-GuzmanDiego</center>

2) Desarrolle en Python un programa para calcular la inversa de matrices de dimensión 2 × 2. No olvide colocar comentarios en su programa.

Ingresamos los valores de la matriz a calcular:

In [4]:
```
print ("Por favor ingresar los 4 elementos de la matriz")
a = float(input ("Primer valor: "))
b = float(input ("Segundo Valor: "))
c = float(input ("Tercer Valor: "))
d = float(input ("Cuarto valor: "))
```

```
Por favor ingresar los 4 elementos de la matriz
Primer valor: 8
Segundo Valor: 5
Tercer Valor: 4
Cuarto valor: 2
```

Calculamos el determinante:

In [5]:
```
detA = (a*d)-(b*c)
```

El determinante que obtuvimos es el siguiente:

In [6]:
```
print(detA)
```

```
-4.0
```

Tenemos que realizar una condición ya que en caso de que el determinante sea 0, no podríamos encontrar la inversa. Si la condición nos indica que el determinante es diferente de cero entonces realizamos el calculo para hallar la inversa.

In [7]:
```
if(a*b-d*c ==0):
    print("No es posible encontrar la inversa en nuestra matriz")
```

```
else:
    a1 = (1/detA)*d
    b1 = (1/detA)*(-b)
    c1 = (1/detA)*(-c)
    d1 = (1/detA)*a
```

Imprimimos la matriz inversa:

In [8]:

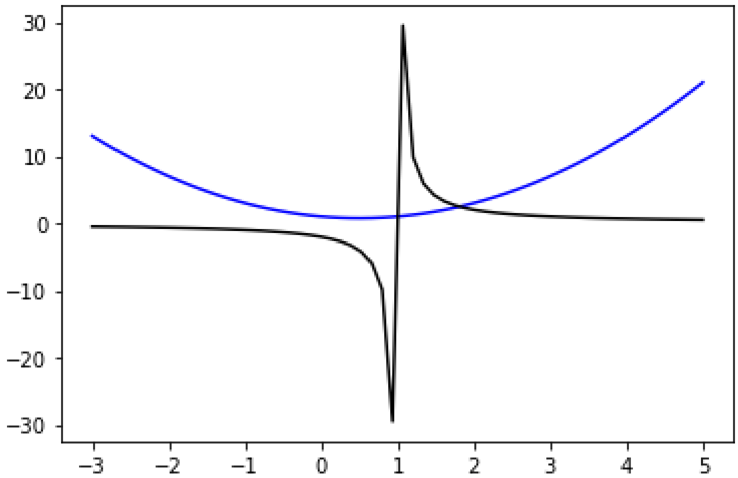
```
print("Matriz Inversa: ")
print (a1 ,b1)
print (c1 ,d1)
```

```
Matriz Inversa:
-0.5 1.25
1.0 -2.0
```

3) Grafique en Python las siguientes funciones, $f(x) = x^2 - x + 1$ **,** $g(x)$ **=** $\frac{2}{x-1}$ **. Grafique ambas funciones en el mismo gráfico.**

In [3]:

```
import numpy as np
import matplotlib.pyplot as plt

fx = lambda x : x**2-x+1
gx = lambda x : 2/(x-1)

x = np.linspace(-3,5,60)

aa = fx(x)
bb = gx(x)

plt.plot(x,aa,color="blue")
plt.plot(x,bb,color="black")

plt.show()
```



4) Sea la función $f(x) = \sqrt{x^2 + 2x + 1}$ **calcule** $f(2.045)$ **(utilice una serie de Taylor).**

$$f(x) = \sqrt{x^2 + 2x + 1}$$

(Serie de Taylor alrededor de x = 1)

$$f(x) = \sqrt{x^2 + 2x + 1}$$

$$f(1) = \sqrt{4} = 2$$

$$f'(x) = \frac{1}{2\sqrt{x^2+2x+1}} \; 2x + 2$$

$f'(x)$ = $\frac{2(x+1)}{2\sqrt{(x+1)^2}}$ = 1

$f'(1) = 1$

$f''(x)$ = 0

$f''(1) = 0$

$f'''(x) = 0$

$f'''(1) = 0$

$f^{n1}(x) = 0$

$f^{n1}(1) = 0$

$f(x) =$ f(a) + $\frac{f'(a)}{1!}$ $(x-a)$ + $\frac{f''(a)}{2!}$ $(x-a)^2$ + $\frac{f'''(a)}{3!}$ $(x-a)^3$ + ...

$\sqrt{x^2+2x+1} =$ 2 + $\frac{1}{1!}$ $(x-1)$ + $\frac{0}{2!}$ $(x-1)^2$ + $\frac{0}{3!}$ $(x-a)^3$ + ...

$\sqrt{x^2+2x+1} = 2+x-1$

$f(2,045) = 2+2,045-1 = 3,045$

$\sqrt{2,045^2+2(2,045)+1} = 3,045$

Error absoluto $|\tilde{x}-x|$ = |3,045 - 3,045| = 0

5) Calcule el error relativo, error absoluto, el error porcentual y las cifras significativas para los siguientes casos:

$$x = 0,005429, \tilde{x} = 0,00543$$
$$x = 189,3478, \tilde{x} = 18,93478$$
$$x_0 = 4,367, x_1 = 4,3689$$

A)

Error Relativo = $\frac{|x-\tilde{x}|}{|x|}$ = $\frac{|0{,}005429-0{,}00543|}{0{,}005429}$ = $0,0001841$ (4 cifras significativas)

Error Absoluto = $|x-\tilde{x}| = |0,005429-0,00543|$ = 0.000001 (1 cifra significativa)

Error Porcentual = Error Relativo * 100% = $0,01841$ % (4 cifras significativas)

B)

Error Relativo = $\frac{|x-\tilde{x}|}{|x|}$ = $\frac{|189{,}3478-18{,}93478|}{189{,}3478}$ = $0,9$ (1 cifra significativa)

Error Absoluto = $|x-\tilde{x}| = |189,3478-18,93478|$ = $170,41302$ (8 cifras significativas)

Error Porcentual = Error Relativo * 100% = 90 % (1 cifra significativa)

C)

Error Relativo = $\frac{|x_0-x_1|}{|x_0|}$ = $\frac{|4{,}367-4{,}3689|}{4{,}367}$ = $0,0004350$ (4 cifra significativas)

Error Absoluto = $|x_0-x_1| = |4,367-4,3689|$ = $0,0019$ (2 cifras significativas)

Error Porcentual = Error Relativo * 100% = $0,0435$ % (3 cifras significativas)

6. Diseñe un código que encuentre el sen $\frac{\Pi}{3}$ a través del desarrollo de Taylor, truncar cuando $n = 50$

In [35]:

```
import numpy as pn

x = pn.pi/3
n = 51
polinomio = 0

for k in range (n):
    polinomio = polinomio + (-1)**k*x**(2*k+1) / pn.math.factorial(2*k+1)
    print(k,polinomio)
```

```
0 1.0471975511965976
1 0.8558007815651173
2 0.8662952837868347
3 0.8660212716563725
4 0.8660254450997811
5 0.8660254034934827
6 0.8660254037859597
7 0.8660254037844324
8 0.8660254037844385
9 0.8660254037844385
10 0.8660254037844385
11 0.8660254037844385
12 0.8660254037844385
13 0.8660254037844385
14 0.8660254037844385
15 0.8660254037844385
16 0.8660254037844385
17 0.8660254037844385
18 0.8660254037844385
19 0.8660254037844385
20 0.8660254037844385
21 0.8660254037844385
22 0.8660254037844385
23 0.8660254037844385
24 0.8660254037844385
25 0.8660254037844385
26 0.8660254037844385
27 0.8660254037844385
28 0.8660254037844385
29 0.8660254037844385
30 0.8660254037844385
31 0.8660254037844385
32 0.8660254037844385
33 0.8660254037844385
34 0.8660254037844385
35 0.8660254037844385
36 0.8660254037844385
37 0.8660254037844385
38 0.8660254037844385
39 0.8660254037844385
40 0.8660254037844385
41 0.8660254037844385
42 0.8660254037844385
43 0.8660254037844385
44 0.8660254037844385
45 0.8660254037844385
46 0.8660254037844385
47 0.8660254037844385
48 0.8660254037844385
49 0.8660254037844385
50 0.8660254037844385
```