Universidad de las Fuerzas Armadas ESPE
Métodos numéricos NRC 3657

Lizeth Abigail Iza Moreno

4 de junio de 2021

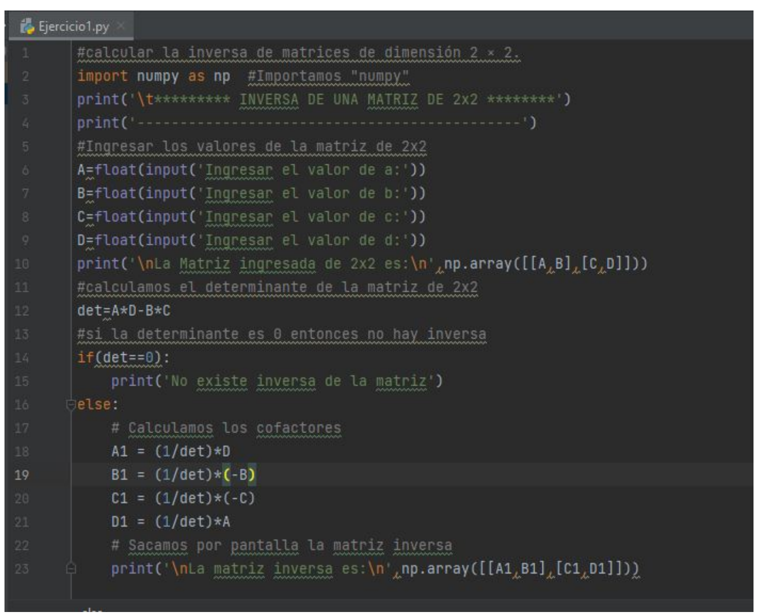
Descripción de la actividad.

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/Lizeth-Abigail-Iza-Moreno/MetodosNumericos3657.git

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

Código:



```python
#calcular la inversa de matrices de dimensión 2 × 2.
import numpy as np  #Importamos "numpy"
print('\t********* INVERSA DE UNA MATRIZ DE 2x2 ********')
print('---------------------------------------------')
#Ingresar los valores de la matriz de 2x2
A=float(input('Ingresar el valor de a:'))
B=float(input('Ingresar el valor de b:'))
C=float(input('Ingresar el valor de c:'))
D=float(input('Ingresar el valor de d:'))
print('\nLa Matriz ingresada de 2x2 es:\n',np.array([[A,B],[C,D]]))
#calculamos el determinante de la matriz de 2x2
det=A*D-B*C
#si la determinante es 0 entonces no hay inversa
if(det==0):
    print('No existe inversa de la matriz')
else:
    # Calculamos los cofactores
    A1 = (1/det)*D
    B1 = (1/det)*(-B)
    C1 = (1/det)*(-C)
    D1 = (1/det)*A
    # Sacamos por pantalla la matriz inversa
    print('\nLa matriz inversa es:\n',np.array([[A1,B1],[C1,D1]]))
```

Figura 1: Código de la Matriz Inversa

Corrida de Escritorio:

```
    ********* INVERSA DE UNA MATRIZ DE 2x2 ********
---------------------------------------------
Ingresar el valor de a:4
Ingresar el valor de b:7
Ingresar el valor de c:2
Ingresar el valor de d:6

La Matriz ingresada de 2x2 es:
 [[4. 7.]
 [2. 6.]]

La matriz inversa es:
 [[ 0.6 -0.7]
 [-0.2  0.4]]

Process finished with exit code 0
```

Figura 2: Ejecución de la Matriz Inversa 2x2

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.

Código:

```
Ejercicio2.py
#Grafique de ambas funciones en el mismo gráfico
#Importamos las librerias matplolib y numpy y las renombra para invocarla como np y plt
import matplotlib.pyplot as plt
import numpy as np
#definimos las funcion cuadratica y racional
def f(x):
    return x**2-x+1
def g(x):
    return 2/(x-1)
#coordenadas del eje x
x=np.linspace(-30,30,1000)
fig=plt.figure(figsize=(8,4))
#Graficamos y colocamos el color a cada una de las funciones
plt.plot(x,f(x),'r--',label='f(x)=$x^2-x+1$')
plt.plot(x,g(x),'b',label='$g(x)=2/x-1$')
#Damos nombres al eje 'X' y 'Y'
plt.xlabel('x')
plt.ylabel('y')
#Limita los valores de los ejes
plt.xlim(-20,20)
plt.ylim(-20,20)
plt.grid(True) #coloca cuadriculas
plt.title('Gráfica de funciones') #titulo de la gráfica
plt.legend(loc = 1)
plt.show() #Mostramos las graficas
```

Figura 3: Código de las gráficas de Funciones

Corrida de Escritorio:

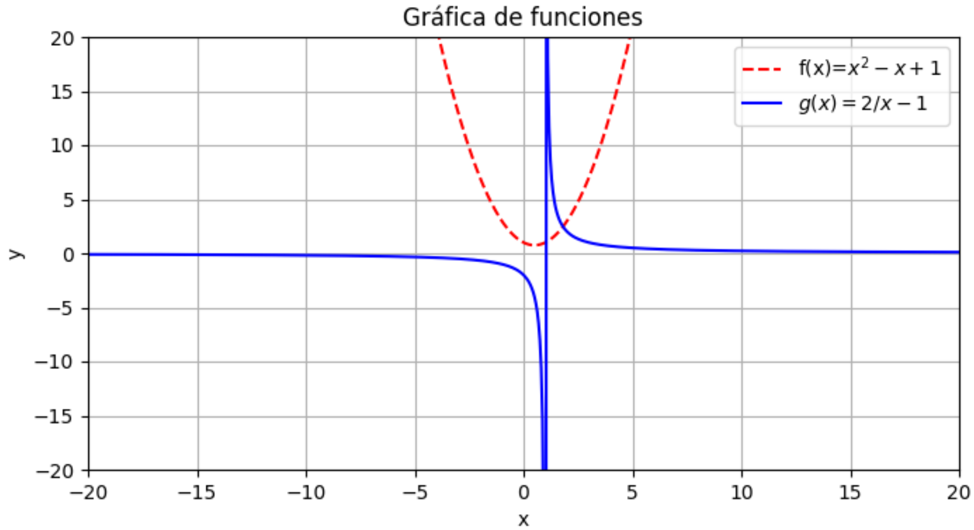


Figura 4: Ejecución del programa de las 2 funciones

4. Sea la función $f(x) = \sqrt{x^2 + 2x + 1}$ calcule f(2.045) con un error de 0.00005 (utilice una serie de Taylor).

La fórmula de la serie Taylor

$$f_{(x)} = f_{(a)} + f'_{(a)}(x-a) + f''_{(a)}\frac{(x-a)^2}{2!} + \ldots + f^{(n)}_{(a)}\frac{(x-a)^n}{n!}$$

Primero calculamos las derivadas de la función y evaluamos a=0

Derivada $------>$ Evaluó en a=0

$$f(x) = \sqrt{x^2+2x+1} ----> f(0) = \sqrt{(0)^2+2(0)+1} ----> f(o) = 1$$

$$f'(x) = \frac{x+1}{\sqrt{x^2+2x+1}} ----> f'(0) = \frac{(0)+1}{\sqrt{(0)^2+2(0)+1}} ---> f'(o) = 1$$

$$f''(x) = \frac{1}{\sqrt{x^2+2x+1}} - \frac{(x+1)(2x+2)}{2(x^2+2x+1)^{\frac{3}{2}}} ---> f''(0) = \frac{1}{\sqrt{0^2+2(0)+1}} - \frac{(0+1)(2(0)+2)}{2((0)^2+2(0)+1)^{\frac{3}{2}}} ---> f''(0) = 0$$

$$f'''(x) = 0 ---> f'''(0) = 0$$

A continuación reemplazamos los valores en la formula de la serie de Taylor y f(2.045)

$$f_{(x)} = f_{(a)} + f'_{(a)}(x-a) + f''_{(a)}\frac{(x-a)^2}{2!} + f'''_{(a)}\frac{(x-a)^3}{3!} \ldots + f^{(n)}_{(a)}\frac{(x-a)^n}{n!}$$

$$\sqrt{x^2+2x+1} = 1 + 1(2{,}045-0) + 0\frac{(2{,}045-0)^2}{2!} + 0\frac{(2{,}045-0)^3}{3!}$$

$$\sqrt{2{,}045)^2 + 2(2{,}045) + 1} = 1 + 1(2{,}045 - 0) + 0\frac{(2{,}045-0)^2}{2!} + 0\frac{(2{,}045-0)^3}{3!}$$

$$3{,}045 = 1 + 2{,}045 + 0 + 0$$

$$3{,}045 = 3{,}045$$

Por lo tanto, en esta función al calcular la serie de Taylor nos da como resultado que no hay error, porque al derivar la función la ultima deriva es cero.

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

$$x = 0{,}005429, \tilde{x} = 0{,}00543$$

Calculamos el Error Relativo:

$$e_{rel} = \frac{|x - \tilde{x}|}{|x|}$$
$$e_{rel} = \frac{|0{,}005429 - 0{,}00543|}{|0{,}005429|}$$
$$e_{rel} = 0,0001841$$

Tiene 4 cifras significativas

Calculamos el error Absoluto:

$$e_{abs} = |x - \tilde{x}|$$
$$e_{abs} = |0{,}005429 - 0{,}00543|$$
$$e_{abs} = 0,000001$$

Tiene 1 cifra significativa

Calculamos el error porcentual:

$$e_{\%} = e_{rel} * 100$$
$$e_{\%} = 0,0001841 * 100$$
$$e_{\%} = 0,018\,\%$$

Tiene 2 cifras significativas

$$x = 189{,}3478, \tilde{x} = 18{,}93478$$

Calculamos el Error Relativo:

$$e_{rel} = \frac{|x - \tilde{x}|}{|x|}$$
$$e_{rel} = \frac{|189{,}3478 - 18{,}93478|}{|189{,}3478|}$$
$$e_{rel} = 0,9$$

Tiene 1 cifra significativa

Calculamos el error Absoluto:

$$e_{abs} = |x - \tilde{x}|$$
$$e_{abs} = |189{,}3478 - 18{,}93478|$$
$$e_{abs} = 170{,}4130$$

Tiene 7 cifras significativas

Calculamos el error porcentual:

$$e_{\%} = e_{rel} * 100$$
$$e_{\%} = 0,9 * 100$$
$$e_{\%} = 90\,\%$$

Tiene 2 cifra significativa

$$x_0 = 4{,}367, x_1 = 4{,}3689$$

Calculamos el Error Relativo:

$$e_{rel} = \frac{|x - \tilde{x}|}{|x|}$$
$$e_{rel} = \frac{|4{,}367 - 4{,}3689|}{|4{,}367|}$$
$$e_{rel} = 0{,}000435$$

Tiene 3 cifra significativa

Calculamos el error Absoluto:

$$e_{abs} = |x - \tilde{x}|$$
$$e_{abs} = |4{,}367 - 4{,}3689|$$
$$e_{abs} = 0{,}0019$$

Tiene 2 cifra significativa

Calculamos el error porcentual:

$$e_{\%} = e_{rel} * 100$$
$$e_{\%} = 0{,}000435 * 100$$
$$e_{\%} = 0{,}0435\,\%$$

Tiene 3 cifra significativa

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

Código:

```python
# Código para calcular el sen(pi/3) con la serie de taylor, truncar cuando n=50
#Importamos la libreria numpy y renombramos para invocarla como pn
import numpy as pn
x = pn.pi/3
n = 51 #truncamos  para  que nos salga los  primeros 50 valores
sen_x = 0  #inicializamos el acumulador en cero
print("\n\t*****CACULAMOS EL SEN(pi/3)****")
print("\nn    pn(x)")
#ciclo para acumular /sumar los terminos
for k in range(n):
    sen_x=sen_x+(-1)**k*x**(2*k+1)/pn.math.factorial(2*k+1)
    print(k,sen_x)
```

Figura 5: Código del sen(pi/3)

Corrida de Escritorio:

```
	*****CACULAMOS EL SEN(pi/3)****

n    pn(x)
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 0.866025403784385
9 0.866025403784385
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
```

```
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
```

Figura 6: Ejecución del programa del $\sin(\frac{\pi}{3})$