

FACTORIZACIÓN DE MATRICES



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Estudiantes:

Andrés Felipe Ávila Rosas-20221578060

Luis Mario Arévalo Bastidas-20221978027

Sara Viviana Herrera Bejarano-20221578129

Docente:

Gloria Andrea Cavanzo Nisso

Universidad Distrital Francisco José de Caldas

Facultad Tecnológica

Tecnología en Sistematización de Datos

Análisis y métodos numéricos

Bogotá D.C

2023

- Dada la matriz

$$\begin{pmatrix} -5 & 8 & -11 & 2 \\ 4 & 7 & 6 & 23 \\ 0 & 7 & -23 & 4 \\ 19 & 1 & 3 & 8 \end{pmatrix}$$

Halle su inversa usando el método LU

CÓDIGO EN PYTHON

Se realizó un código en python con el fin de encontrar la inversa de la matriz dada usando el método de LU

```
def Descom_LU(m):
    n = len(m)
    L = [[0.0] * n for _ in range(n)]
    U = [[0.0] * n for _ in range(n)]

    for i in range(n):
        L[i][i] = 1.0

        for j in range(i, n):
            s = sum(L[i][k] * U[k][j] for k in range(i))
            U[i][j] = m[i][j] - s

        for j in range(i + 1, n):
            s = sum(L[j][k] * U[k][i] for k in range(i))
            L[j][i] = (m[j][i] - s) / U[i][i]

    return L, U

def stc_adelante(L, b):
    n = len(L)
    y = [0.0] * n

    for i in range(n):
        y[i] = b[i] - sum(L[i][j] * y[j] for j in range(i))

    return y

def stc_atras(U, y):
    n = len(U)
    x = [0.0] * n

    for i in reversed(range(n)):
        x[i] = (y[i] - sum(U[i][j] * x[j] for j in range(i + 1, n))) / U[i][i]

    return x
```

```
def m_inversa(m):
    n = len(m)
    m_identidad = [[1.0 if i == j else 0.0 for j in range(n)] for i in range(n)]

    L, U = Descom_LU(m)

    inversa = []
    for i in range(n):
        b = m_identidad[i]
        y = stc_adelante(L, b)
        x = stc_atras(U, y)
        inversa.append(x)

    return list(map(list, zip(*inversa)))

m = [
    [-5, 8, -11, 2],
    [4, 7, 6, 23],
    [0, 7, -23, 4],
    [19, 1, 3, 8]
]

inversa = m_inversa(m)

# Imprimir la matriz inversa
for inv in inversa:
    print(inv)
```

El resultado de la ejecución es el siguiente:

```
[0.01617211753663996, -0.021851611195340896, -0.005414771496642848, 0.0614877385507665]
[0.2339903256082593, -0.024498832815921842, -0.1095949750920511, 0.066734050489736]
[0.05580824489206554, 0.0019733833898876118, -0.06779293913796837, 0.01427093110004091]
[-0.08858566168507687, 0.05421991191971699, 0.05198180636777127, -0.03472673453180276]

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Se hace la comparación de resultados obtenidos con el programa en python y en Excel usando la función MINVERSA ()

MATRIZ				INVERSA			
-5	8	-11	2	0,01617212	-0,0218516	-0,0054148	0,06148774
4	7	6	23	0,23399033	-0,0244988	-0,109595	0,06673405
0	7	-23	4	0,05580824	0,00197338	-0,0677929	0,01427093
19	1	3	8	-0,0885857	0,05421991	0,05198181	-0,0347267

Podemos observar que los resultados son los mismos en ambos programas, solamente que en Excel no están todos los números decimales y hay una aproximación de estos.