

## Disciplina Métodos Computacionais

Professor Dr. Wanderlei Malaquias Pereira Junior

### Exemplo 010 - Eliminação de Gauss

#### Declaração do sistema linear

```
import numpy as np
A = np.array([[3, -0.10, -0.20], [0.1, 7.0, -0.30], [0.3, -0.20, 10]])
B = np.array([[7.85], [-19.3], [71.4]])
```

#### Verificando aplicabilidade do método

```
TESTE = np.linalg.det(A)
if TESTE > 0:
    print('S.P.D', 'Valor det(A): ', TESTE)

else:
    print('S.I.', 'Valor det(A): ', TESTE)
```

☞ S.P.D Valor det(A): 210.35299999999999

#### Matriz **A** e vetor **b**

```
AB = np.concatenate((A, B), axis = 1)
AB
```

#### Transformações das linhas pelo método de Gauss

```
N = 3
N_MAX = N - 1
for K_CONT in range(0, N_MAX):
    print('k: ', K_CONT)
    for I_CONT in range(K_CONT + 1, N):
        print('m: ', I_CONT)
        PIVO = AB[I_CONT, K_CONT] / AB[K_CONT, K_CONT]
        print('Pivô: ', PIVO)
        print('Linha I: ', AB[I_CONT, K_CONT:(N_MAX + 2)])
        print('Linha K: ', AB[K_CONT, K_CONT:(N_MAX + 2)])
        AB[I_CONT, K_CONT:(N_MAX + 2)] = AB[I_CONT, K_CONT:(N_MAX + 2)] - PIVO * AB[K_CONT, K_CONT:(N_MAX + 2)]

print(AB)
```

#### Montagem do algoritmo *Triangular Superior* com ajuda de uma função

```
def SISTRIANGULAR(A, B, N):
```

```
X = [0] * N
for I_CONT in range(N - 1, -1, -1):
    print('Iteração I: ', I_CONT)
    SOMA = 0
    for J_CONT in range((I_CONT + 1), N, 1):
        SOMA = SOMA + AB[I_CONT, J_CONT] * X[J_CONT]
    print('B: ', B[I_CONT, 0])
    print('SOMA: ', SOMA)
    print('A: ', A[I_CONT, I_CONT])
    VALOR = (B[I_CONT, 0] - SOMA) / A[I_CONT, I_CONT]
    print('X: ', VALOR)
    X[I_CONT] = VALOR
return X
```

```
A = AB[:, 0 : N_MAX + 1]
print(A)
B = AB[:, N_MAX + 1]
print(B)
print(B.shape)
B_NEW = np.expand_dims(B, axis=1)
print(B_NEW.shape, B_NEW)
```

```
X = SISTRIANGULAR(A, B_NEW, 3)
print('Vetor solução: ', X)
```