Nome: Gabriel de Souza Vieira

Relatório da parte 3 do Projeto computacional

Introdução

Neste relatório será mostrado a prova de corretude do algoritmo para calcular determinante de matrizes, utilizando eliminação gaussiana para transformar a matriz triangular superior e calcular o determinante.

Funcionamento do código

A idéia é receber uma matriz A e escalonar para transformar a matriz em triângular superior, em que Aij = 0 para i > j, aproveitando da propriedade que o determinante de uma matriz triangular é igual ao seu traço, depois de fazer a eliminação é só multiplicar os elementos da diagonal principal.

Eliminação Gaussiana

O código utilizará dois laços for para passar por todos os elementos da matriz, fazendo operações para tirar os números 0 da diagonal principal e também deixar os números abaixo da diagonal principal iguais a 0.

Caso em que o elemento da diagonal é igual a 0

```
if u[i][c] == 0 and i==c:

    #1
    for k in range(len(u)):
        if u[k][c] != 0:
            break
        if k == len(u) - 1:
            return 0

    for k in range(len(u)):
        if k == 0:
            continue
        #2
        if u[c+k][i] != 0:
            u[c], u[c+k] = u[c+k], u[c]
            det *= -1
            break

    break
```

Caso o elemento em que a repetição esteja passando for da diagonal principal e igual a 0 irá cair nesse condicional que irá para outra repetição, que irá passar pelos elementos da coluna e verificar onde o elemento é diferente de 0 para realizar a troca de linha.

#1

Caso passe por todos os elementos e nao encontre um número diferente de 0, significa que a coluna é igual a 0, então retorna o determinante igual a 0

#2

Encontrando uma linha em que o elemento não seja 0, irá realizar a troca de linhas, além da troca, o determinante (que não foi calculado ainda) muda de sinal, já que pelo item (i) da questão 27 da teo 6, quando há uma troca de linha, a matriz é multiplicada por uma elementar E, em que o det(E) = -1.

Caso em que o elemento está abaixo da diagonal principal

```
if c < i and u[c][i] != 0:
    divisor = u[i][c]/u[c][c]
    for w in range(len(u[i])):
        u[i][w] = u[i][w] - divisor * u[c][w]
```

Caso o elemento em que a repetição esteja passando está abaixo da diagonal principal, ele irá fazer esse elemento ser igual a 0, caso já seja igual a 0 apenas passa para a próxima iteração. Sendo diferente de 0, a linha inteira recebe a propria linha menos “divisor” vezes outra linha, transformando o elemento em 0

Cálculo do determinante

```
    for i in range(len(u)):
        det *= u[i][i]
    return det
```

Com a matriz já escalonada o determinante é igual ao traço, então resta apenas multiplicar os elementos da diagonal principal, já tendo a troca de sinal pela permutação feita na parte da eliminação gaussiana, e no final retorna o resultado do determinante

Conclusão

Conclui-se que o algoritmo funciona para quaisquer matriz, calculando seu determinante, mantendo na memória apenas uma matriz e uma variável auxiliar assim como especificado, também funcionando para casos especiais como linhas ou colunas de 0, seguindo as propriedades fundamentais do determinante.