

In [0]:

GA - Trabalho 2

Nome: Cintia da Silva Bulcao

Ra: 145107

1. A rotina número 01 deverá ser tal que: (a) O usuário deverá inserir as coordenadas inteiras da equação geral $ax + by + cz + d = 0$ de um plano qualquer que não passe pela origem, que não seja paralelo a algum plano coordenado nem paralelo a algum eixo coordenado. Essa inserção deverá ser feita de uma só vez, na ordem em que aparecem os coeficientes da equação geral, com os valores separados por vírgulas. (b) A rotina deverá, após o registro do plano, receber as coordenadas de um ponto $P = (a,b,c)$ inseridas pelo usuário. (c) Em seguida, a rotina deverá verificar se o ponto pertence ou não ao plano fornecido e deverá imprimir essa informação na tela. (d) Por final, a rotina deverá perguntar se o usuário quer testar a pertinência de outro ponto ou não. (e) Se o usuário escolher que não quer testar mais pontos, a rotina deve perguntar se ele quer inserir os coeficientes de outro plano ou não.

```
In [1]: seguir = 'y'

while seguir == 'y':
    plano_correto = False
    while not plano_correto:
        entrada_texto = input('Digite os coeficientes a,b,c,d: ')
        partes = entrada_texto.split(',')

        if len(partes) != 4:
            print('Erro: insira 4 valores separados por vírgulas.')
            continue

        try:
            a, b, c, d = map(int, [p.strip() for p in partes])
        except ValueError:
            print('Erro: insira apenas números inteiros.')
            continue

        if d == 0 or a == 0 or b == 0 or c == 0:
            print('Erro: o plano não pode passar pela origem nem ser paralelo a algum eixo coordenado.')
            continue

        plano_correto = True

    checar_ponto = 'y'
    while checar_ponto == 'y':
```

```

entrada_ponto = input('Digite as coordenadas x,y,z: ')
partes = entrada_ponto.split(',')

if len(partes) != 3:
    print('Erro: insira 3 valores separados por vírgulas.')
    continue

try:
    x, y, z = map(float, [p.strip() for p in partes])
except ValueError:
    print('Erro: insira valores numéricos válidos.')
    continue

resultado = a * x + b * y + c * z + d

if abs(resultado) < 1e-6:
    print('O ponto pertence ao plano.')
else:
    print('O ponto NÃO está no plano.')

checar_ponto = input('Outro ponto? (y/n): ').strip().lower()

seguir = input('Novo plano? (y/n): ').strip().lower()

print('Encerrado.')

```

Digite os coeficientes a,b,c,d: 0,0,0,0

Digite os coeficientes a,b,c,d:

Erro: o plano não pode passar pela origem nem ser paralelo a eixos/planos coordenados.

Digite as coordenadas x,y,z: 2,2,2

O ponto NÃO está no plano.

Outro ponto? (y/n): y

Digite as coordenadas x,y,z: 4,2,1

O ponto NÃO está no plano.

Outro ponto? (y/n): n

Novo plano? (y/n): n

Encerrado.

2. A rotina número 02 deverá ser tal que: (a) O usuário deverá inserir as coordenadas inteiras da equação geral $ax + by + cz + d = 0$ de um plano qualquer que não passe pela origem, que não seja paralelo a algum plano coordenado nem paralelo a algum eixo coordenado. Essa inserção deverá ser feita de uma só vez, na ordem em que aparecem os coeficientes da equação geral, com os valores separados por vírgulas. (b) A rotina deverá, após a captura dos coeficientes, imprimir a equação geral cujos coeficientes foram inseridos. Isso serve para conferência do usuário. (c) A rotina deverá determinar os pontos A, B e C, respectivamente pontos de interseção do plano com os eixos coordenados Ox, Oy e Oz. Deverá, também, imprimir as coordenadas desses pontos na tela. (d) A rotina deverá calcular a área do triângulo ABC e imprimir essa informação na tela. (e) A rotina deverá calcular a altura do triângulo ABC relativa ao lado xOz que pertence ao

plano, ou seja, altura relativa ao lado AC. Deverá, também, imprimir essa informação na tela. (f) A rotina deverá calcular o volume do tetraedro limitado pelo plano fornecido e pelos planos coordenados, ou seja, o volume do tetraedro OABC. Deverá imprimir adequadamente essa informação na tela. (g) A rotina deverá mostrar parte do plano fornecido, mostrar os pontos A, B e C, e mostrar o triângulo em cor destacada. (h) Ao final, deverá perguntar ao usuário se ele quer inserir os coeficientes de outra equação geral para novos cálculos ou se quer encerrar o programa.

```
In [2]: x, y, z = var('x y z')
TOLERANCIA = 1e-3
continuar = 'y'

while continuar == 'y':
    entrada = input('Digite a,b,c,d: ').replace(' ', '').split(',')

    if len(entrada) != 4:
        print('Erro: insira 4 valores separados por vírgula.')
        continue

    valido = True
    for parte in entrada:
        if parte == '' or (parte[0] not in '-0123456789'):
            valido = False
            break
        j = 0
        if parte[0] == '-':
            j = 1
        while j < len(parte):
            if parte[j] < '0' or parte[j] > '9':
                valido = False
                break
            j += 1
        if not valido:
            break

    if not valido:
        print('Erro: insira apenas números inteiros.')
        continue

    valores = []
    for parte in entrada:
        negativo = False
        if parte[0] == '-':
            negativo = True
            parte = parte[1:]
        num = 0
        for ch in parte:
            num = num * 10 + (ord(ch) - ord('0'))
        if negativo:
            num = -num
        valores.append(num)
```

```

a, b, c, d = valores

if a == 0 or b == 0 or c == 0 or d == 0:
    print('Erro: plano não pode passar pela origem nem ser paralelo a eixos')
    continue

print(f'\nPlano: ({a})x + ({b})y + ({c})z + ({d}) = 0')

A = (-d/a, 0, 0)
B = (0, -d/b, 0)
C = (0, 0, -d/c)

print('\nInterseções:')
print('A (eixo x):', A)
print('B (eixo y):', B)
print('C (eixo z):', C)

AC = [C[i] - A[i] for i in range(3)]
BC = [C[i] - B[i] for i in range(3)]

prod = [
    AC[1]*BC[2] - AC[2]*BC[1],
    AC[2]*BC[0] - AC[0]*BC[2],
    AC[0]*BC[1] - AC[1]*BC[0]
]

modulo = (prod[0]**2 + prod[1]**2 + prod[2]**2) ** 0.5
area = modulo / 2
print(f'\nÁrea do triângulo ABC: {round(area, 4)} u²')

base = (AC[0]**2 + AC[1]**2 + AC[2]**2) ** 0.5
altura = (2 * area) / base if base != 0 else 0
print(f'Altura relativa ao lado AC: {round(altura, 4)} u')

OC = [-C[0], -C[1], -C[2]]
prod_misto = OC[0]*prod[0] + OC[1]*prod[1] + OC[2]*prod[2]
volume = abs(prod_misto) / 6
print(f'Volume do tetraedro OABC: {round(volume, 4)} u³')

plano_eq = a*x + b*y + c*z + d == 0
intervalo_x = [x, -1, max(3, abs(A[0])) + 1]
intervalo_y = [y, -1, max(3, abs(B[1])) + 1]
intervalo_z = [z, -1, max(3, abs(C[2])) + 1]

fig1 = implicit_plot3d(plano_eq, intervalo_x, intervalo_y, intervalo_z,
fig2 = polygon([A, B, C], color='red', axes=True)
fig3 = points([A, B, C], size=25, color='black')
show(fig1 + fig2 + fig3)

```

```

continuar = input('\nNovo plano? (y/n): ').strip().lower()

print('\nEncerrado.')

```

Digite a,b,c,d: 0,0,0,0

Erro: plano não pode passar pela origem nem ser paralelo a eixos ou planos coordenados.

Digite a,b,c,d: 2,7,1,5

Plano: $(2)x + (7)y + (1)z + (5) = 0$

Interseções:

A (eixo x): $(-5/2, 0, 0)$

B (eixo y): $(0, -5/7, 0)$

C (eixo z): $(0, 0, -5)$

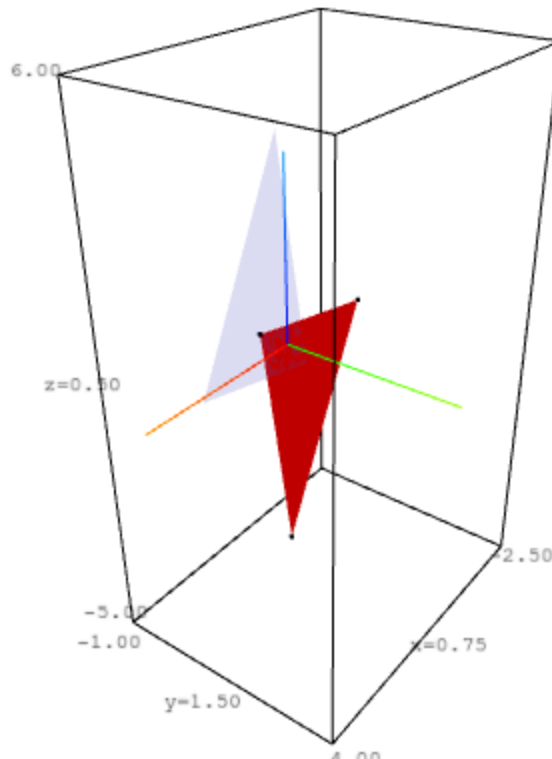
Área do triângulo ABC: 6.5611 u^2

Altura relativa ao lado AC: 2.3474 u

Volume do tetraedro OABC: 1.4881 u^3

Novo plano? (y/n):

Out[2]:



ⓘ

Digite a,b,c,d: 2,15,6,1

Plano: $(2)x + (15)y + (6)z + (1) = 0$

Interseções:

A (eixo x): $(-1/2, 0, 0)$

B (eixo y): $(0, -1/15, 0)$

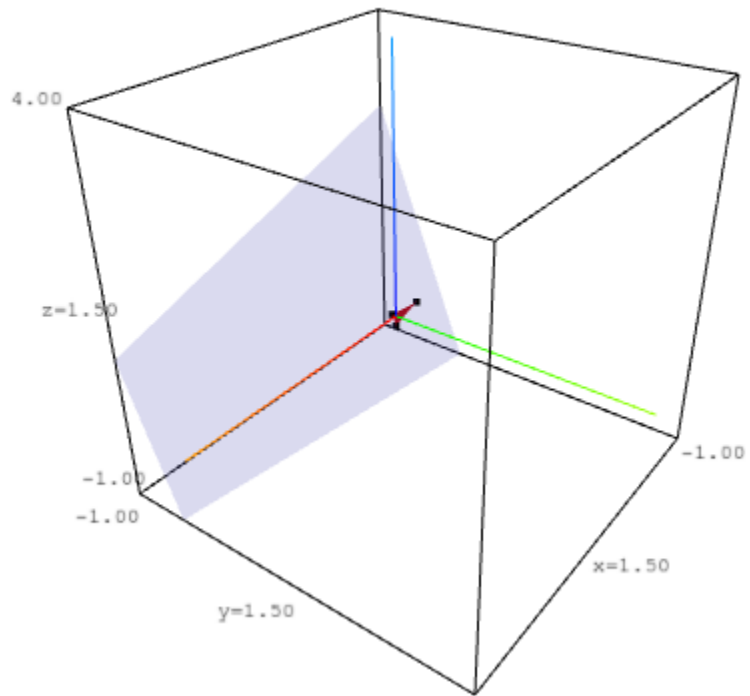
C (eixo z): $(0, 0, -1/6)$

Área do triângulo ABC: 0.0452 u^2

Altura relativa ao lado AC: 0.1716 u

Volume do tetraedro OABC: 0.0009 u^3

Out[2]:



i

Novo plano? (y/n): n
Encerrado.