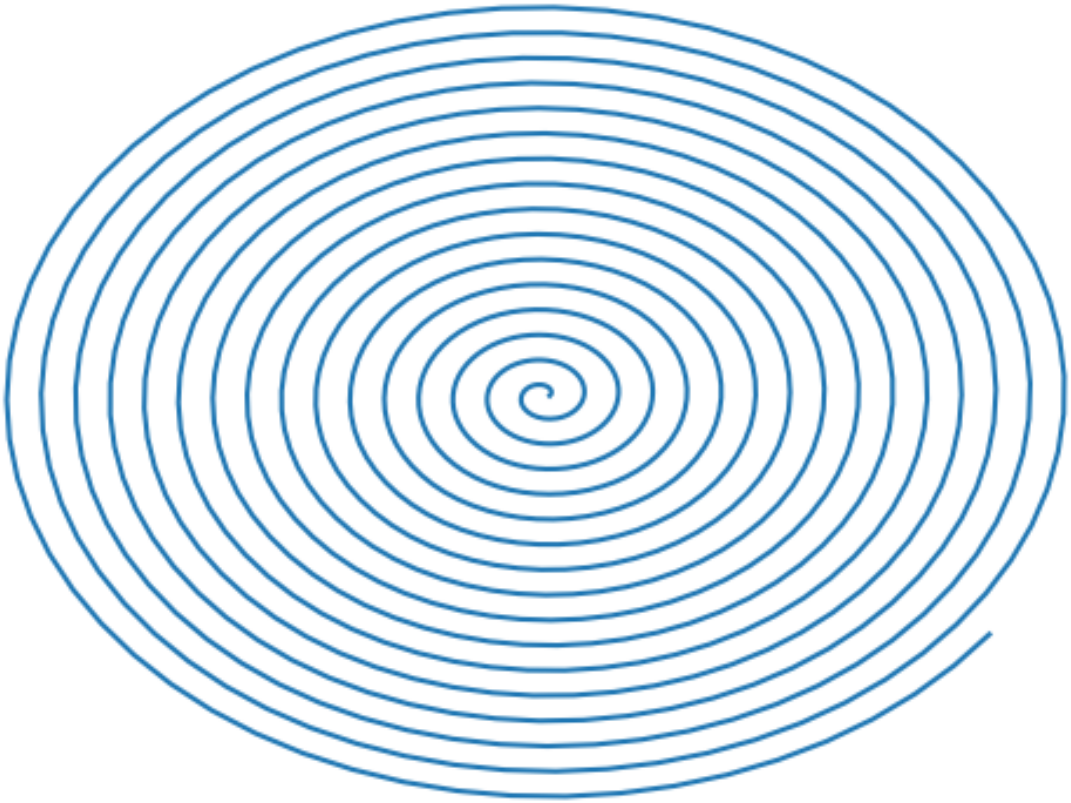
Desafios, gráficos e espirais

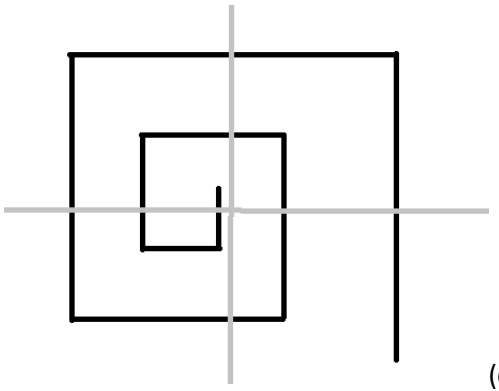


Henrique Nardy Luz 2022

Este repositório inclui alguns projetos e desafios pessoais com a intenção de treinar o uso da linguagem Python, a biblioteca gráfica Matplotlib e as minhas habilidades de pesquisa e resolução de problemas

Desafio #01: espiral quadrada

Imprimir um gráfico com uma espiral quadrada



(como a imagem ao lado)

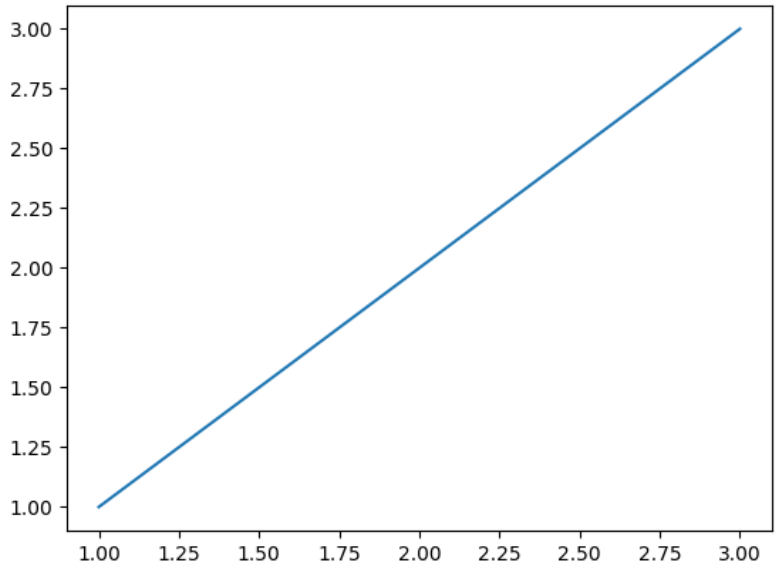
RESOLUÇÃO:

Este foi o tipo de espiral mais fácil que eu pude pensar, então era ideal para começar

A biblioteca Mat Plot Lib possui o método plot() onde passando coordenadas X e Y é criada uma linha reta

Consultar linha.py

```
import matplotlib.pyplot as plt
x = [1 , 2 , 3] #valores de x e y respectivos há uma linha de 45 graus no plano cartesiano
y = [1 , 2 , 3]
plt.plot(x,y) #comando de criação do gráfico
plt.show() #comando para exibir o gráfico
```



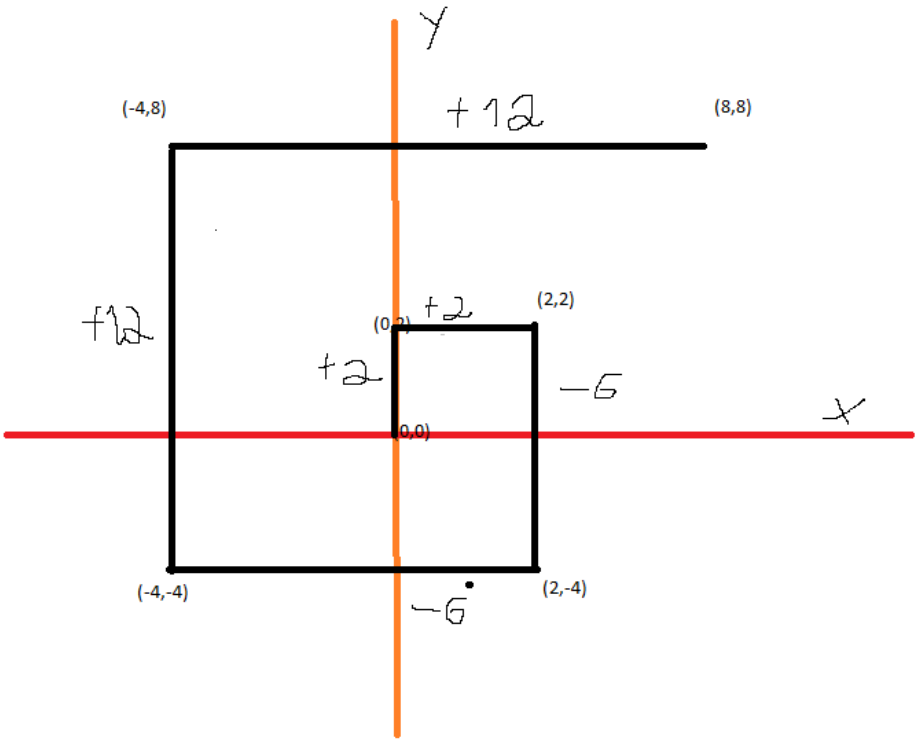
Seguindo essa linha de raciocínio, como seria uma espiral quadrada?

- Linhas retas
- Ângulos retos (múltiplos de 90 graus)
- Linhas crescentes que nunca se cruzam

Para assegurar ângulos retos no plano cartesiano podemos trabalhar apenas invertendo o sinal da operação durante a progressão do crescimento das linhas

Para que as linhas não se cruzem jamais o crescimento será feito com progressão geométrica (razão x2)

Mesmo assim a melhor tática de início é analisar o comportamento de uma espiral quadrada



O crescimento do valor das coordenas começa em 0, se adiciona 2 e logo após o valor é dobrado

(0, 2, 4, 8, 16, 32, 64...)

Para que haja a curva em 90 graus a cada vez que o valor é dobrado o sinal deve ser invertido

(n* (- 1))

(0, 2, -4, 8, -16, 32, -64)

Analisando as combinações de valores do gráfico em uma tabela

X	Y
0	0
0	2
2	2
2	-4
-4	-4
-4	8
8	8
8	-16

Podemos notar que:

- Y repete cada valor da progressão duas vezes
- X assume sempre o valor anterior de Y

Enfim, temos informação o suficiente, vamos organizar as etapas

- Criar duas listas (x e y)
- Criar uma progressão geométrica com a inversão de sinal
- Adicionar os valores da progressão a lista com a ordem e quantidade correta

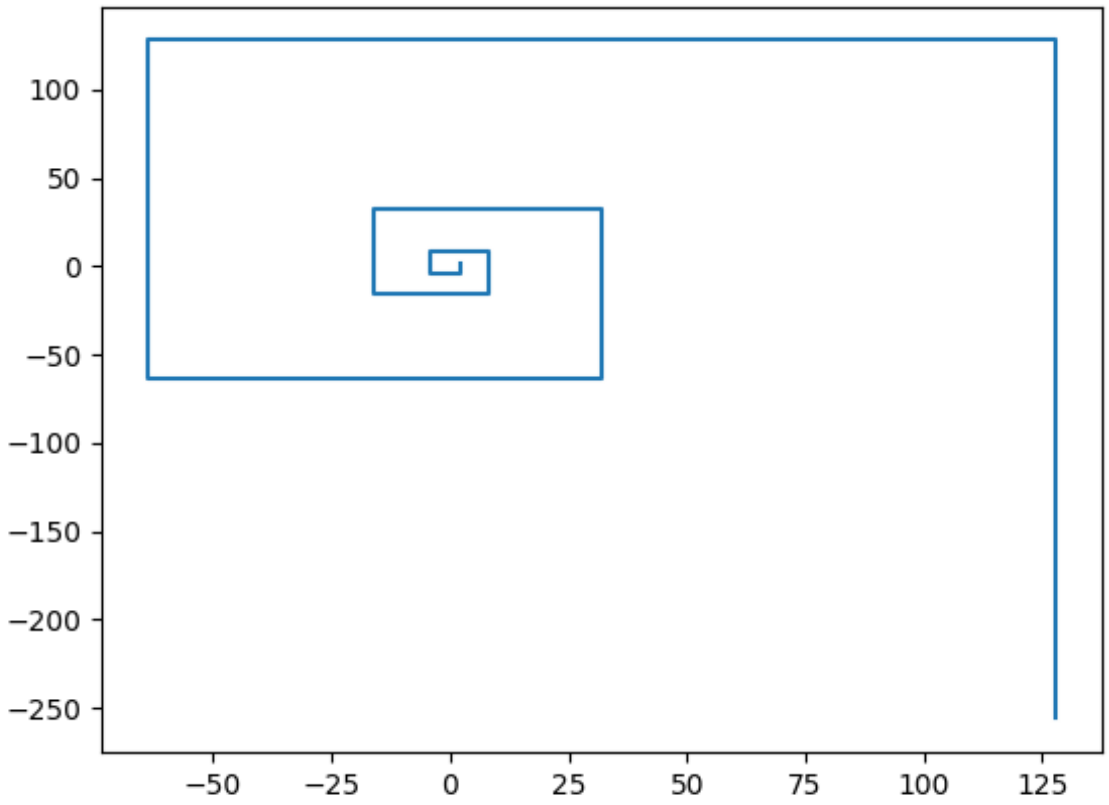
RESOLUÇÃO:

Consultar espiral quadrada.py na pasta codes

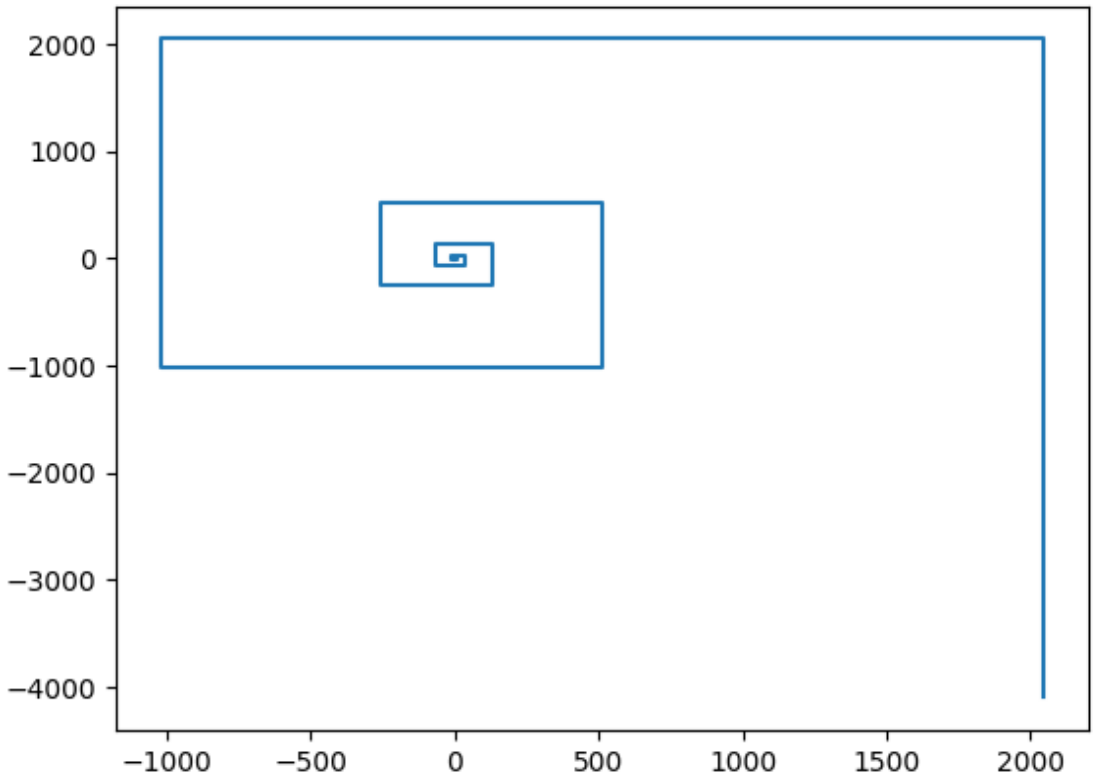
```
import matplotlib.pyplot as plt #importação da biblioteca

progressao = 2 # variável de controle da progressão geométrica ()
razao = 2  # razão da progressão geométrica
voltas = 5 # contador de "meias voltas" define o número de linhas da
espiral

x = []
y = []

#ambas as listas iniciam com o valor inicial da progressão

# colagem dos valores iniciais independente do ponto de partida da
progressão
x.append(progressao)
x.append(progressao) # o eixo x deve ter o primeiro valor duplicado
y.append(progressao)

if (progressao == 0): # caso o valor inicial da progressão for 0 devemos
somar 2
    progressao += razao
else:
    progressao = (progressao * razao)*-1 # com o valor inicial diferente
de zero podemos iniciar a inversão
y.append(progressao)


#após os valores iniciais colados podemos iniciar a repetição

for c in range(0, voltas + 1):
    x.append(progressao)
    x.append(progressao)
    y.append(progressao)
    progressao = (progressao * razao)*-1
    y.append(progressao)

plt.plot(x,y)
plt.show()

#este programa provavelmente ficaria mais  simples usando orientação a
objeto, vou melhora-lo depois
```

Enfim, o tamanho da espiral muda de acordo com o número de voltas e a razão da progressão (*2)

Este é o resultado esperado:

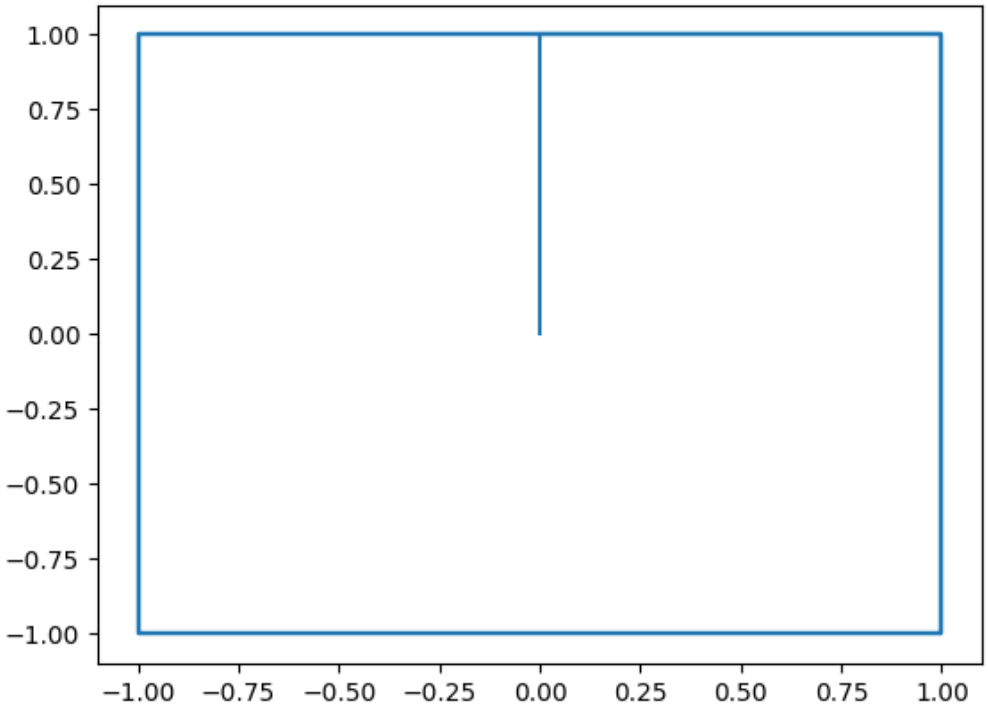


Aumentando o número de voltas para 10, a progressão alcança valores muito altos



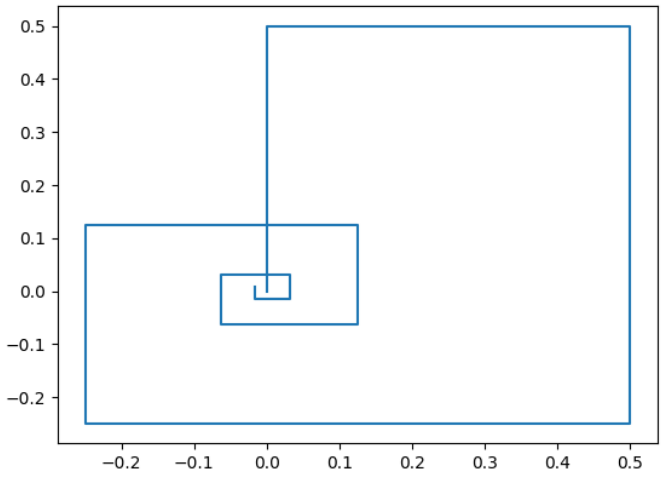
Reduzindo a razão para 1, a espiral não se forma

A progressão sempre repetira os valores (0,1,-1,1,-1,1,-1, 1, ...)



Reduzindo a razão para números menores que 1 conseguimos desenhos interessantes

Voltas = 5 | razão = 0.5



Voltas = 5 | razão = 0.1