A Estrutura do Universo pelos Primos

Autor: **Navacchia, Roberto C. M.**
olibano.incensos@gmail.com

Abstract

The distribution of prime numbers has long intrigued mathematicians, revealing deep structural patterns within mathematics. This study expands upon Gauss's Circle and Goldbach's Theorem, focusing on the interaction between prime pairs rather than all natural numbers. Through mathematical modeling, persistent gaps within Gauss's Circle were identified, suggesting that it represents the internal structure of the universe, while Goldbach's prime-counting graph unveils the external structure, formed exclusively by primes.

By leveraging Computational Modeling, this work explores these patterns in greater depth, revealing potential underlying rules governing prime distribution. The findings suggest that prime numbers may not only be fundamental to number theory but could also serve as the structural foundation of the universe itself.

Palavras Chaves: Teoria dos Números, Cosmologia, Modelos Computacionais

1. Introdução

A estrutura do universo é objeto de estudo de diversas áreas do conhecimento, desde a Física até a Matemática. Entre os padrões fundamentais que emergem na natureza, os números primos se destacam como os elementos essenciais na construção da realidade matemática. Desde a Antiguidade, matemáticos tentam compreender sua distribuição e possíveis conexões com outras áreas do saber.

No século XIX, Gauss já apontava, por meio da função de contagem de primos, uma regularidade assintótica em sua distribuição. Séculos antes, Goldbach formulou sua famosa conjectura sobre a relação entre primos e números pares. No entanto, ao invés de analisar todos os números naturais, este estudo se aprofunda na interação entre os pares de primos e sua organização dentro do Círculo de Gauss, modelos matemáticos gerados a partir dessa abordagem

revelam lacunas persistentes no círculo, sugerindo que ele representa a estrutura interna do universo. Ao mesmo tempo, a análise do gráfico de contagem de Goldbach revela uma estrutura externa, formada exclusivamente pelos primos. Essa dualidade entre as lacunas do Círculo de Gauss e a organização dos primos na conjectura de Goldbach pode ser a chave para compreender uma possível fórmula matemática que descreva a estrutura fundamental do universo.

Com os avanços da Modelagem Computacional, essa investigação ganha novos contornos, permitindo explorar padrões antes inacessíveis. Este artigo apresenta uma análise gráfica detalhada dessa questão, buscando evidências matemáticas que sustentem a hipótese de que os primos não são apenas entidades abstratas, mas sim os verdadeiros estruturadores do universo.

2. Metodologia

A metodologia deste trabalho baseia-se na construção de modelos gráficos e computacionais que possibilitem uma análise profunda da distribuição dos números primos em dois contextos principais: o Círculo de Gauss e o Teorema de Goldbach.

Inicialmente, foi realizada a adaptação do problema do Círculo de Gauss, onde:

$$\{\ \ n=(x,y)\in Z^2 \vee x^2+y^2 \leq N^2\ \ \}$$

restringindo o estudo aos pares de números primos dentro do Círculo, descartando o uso de todos os números naturais, como tradicionalmente proposto. Esta é a adaptação da questão que

$$\{\ \ n_p=(p_{1,}p_2)\in P^2 \vee p_1^2+p_2^2 \leq N^2\ \ \}$$

foi fundamental para isolar e destacar a estrutura real oculta formada pelos primos, revelando as lacunas (L) persistentes na configuração do círculo de raio (N), onde:

$$L(N)=(p_{1,}p_2)\in P^2 \vee p_1^2+p_2^2 \leq N^2$$

as lacunas ao longo de toda a contagem, não se dissipam. Esta característica foi interpretada como a possível representação da '*estrutura interna do universo*', sugerindo uma geometria

profunda regida pela distribuição dos primos dentro do Círculo de raio 'N' conforme gráficos abaixo obtidos através do código I que se encontra disponivel no apêndice :

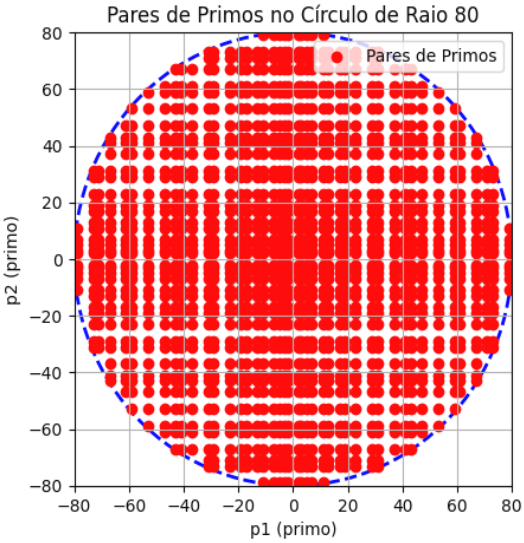

figura 1: Pares de números Primos dentro do círculo de raio 80

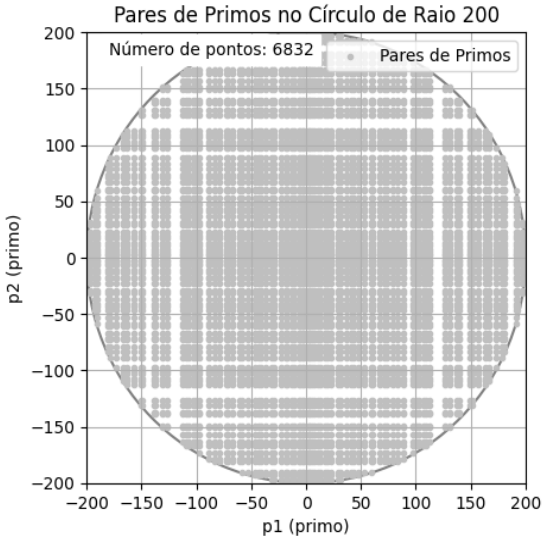

figura 2: Pares de números primos dentro do círculo de raio N= 200

O resultado obtido no gráfico dos primos no círculo de raio N é um conjunto discreto de pontos em pares$(p1,p2)$ de primos, onde cada ponto foi posicionado no plano $\mathbb{R}^2$ resultando em um espaço topológico **X** de variedade tridimensional de Hausdorff T_2 e $T_{2,1/2}$ onde cada ponto em **X** tem uma zona homeomórfica para as três dimensões Euclidianas. No espaço de Hausdorff os pontos distintos podem ser separados por abertos disjuntos onde o conjunto discreto formado pelos pares de primos herdam esta separabilidade e cada ponto pode ser isolado em uma área (bola) aberta ao redor dele que não contém nenhum outro ponto. Esta organização dos primos, como pontos isolados no plano são como as estrelas no espaço - cada uma com um 'espaço suficiente' ao redor para poder se distinguir das outras.

Outro resultado obtido pela analise usando o código foi que a medida que o raio N aumenta a população de número de primos dentro do círculo e a imagem gerada começa a parecer uma massa - mas topologicamente a separação entre os pontos e a variedade T_2 e $T_{2.1/2}$ é mantida pois cada ponto ainda é isolado e não há pontos de acumulação real; o que se altera é a nossa percepção gráfica, que vai de pontos isolados para uma "quase superfície sólida e compacta" condição tipica dos espaços que crescem em uma densidade contínua e não perdem sua estrutura de separação assim como a persistência das lacunas conforme N aumenta.

Podemos formalizar o pensamento acima cujo oconjunto de interesse é:

$$P(N) = \{ (p_1, p_2) \in \mathbb{N}^2 \ , \ p_1 \ e \ p_2 \ , \ \textbf{p1 e p2} \text{ são primos}, \ p^2_1 + p^2_2 \leq N^2 \}$$

onde:

$p^2_1 + p^2_2 \leq N^2$ esta limitado ao círculo de raio ***N.***

E o espaço topólogico definido como:

$$(P(N), \tau_N)$$

onde:

τ_N é a topologia discreta induzida pela seleção dos pares, e cada par é um ponto isolados sendo:

$$\tau_N = \mathcal{P}\,(P(N))$$

Portanto, qualquer subconjunto de pares primos é aberto, e a ***fórmula geral finalizada*** é:

$$(P(N), \tau_N) \ \ com \ \ P(N) = \{ (p_1, p_2) \in \mathbb{N}^2 \ , \ p_1 \ e \ p_2 \ \text{primos}, \ p^2_1 + p^2_2 \leq N^2 \}$$

Imagine o gráfico dos primos crescendo conforme aumentamos o raio N; aparecerão mais e mais pontos, ficando visualmente cheio e formando uma aparente superfície contínua - mas na realidade cada ponto continua isolado conforme provado acima onde: $N \rightarrow \infty$.

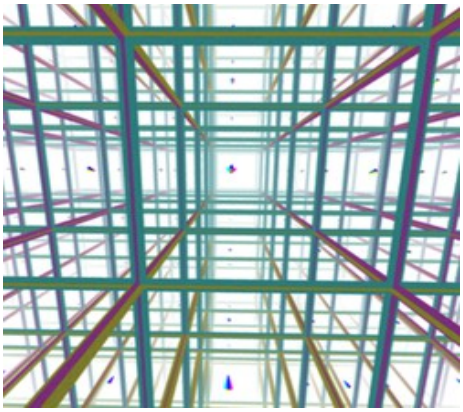

Figura 3: Uma imagem do interior de uma variedade tridimensional gerado pelo programa CurvedSpaces de Jeff Weeks. Esse espaço tem volume finito e não tem fronteira. ex. de *Compactificação.*

Como todo o comportamento 'para o infinito' pode ser trazido de volta para perto, bastando adicionar um 'ponto de Alexandroff no infinito'(ver figura 3), onde os pares de primos ficam compactados à um intervalo de ***N*** em um espaço discreto e o infinito(∞) vira à ser apenas um ponto na borda, e é definido por:

$$P'(N) = P(N) \cup \{ \infty \} \ .$$

A questão seguinte, da pesquisa versa sobre a observação das distribuição de frequências das lacunas (**L**) na contagem dos pares de números primos obtidos para qualquer raio (**N**) do círculo. O conjuntos de resultados gráficos e explicitos obtidos pelas amostras dos modelos computacionais, estão em linguagem python (**.ipynb**), e estão elencados no Apêndice para uso livre de testes destes resultados.

O cálculo seguinte envolve o uso da função zeta (ζ) de Riemann nas análises de distribuição dos números primos que é dado por:

Raio = ***(N)***

f:(N) = ζ (N +i.t)

Encontrando os primos associados ao valor 'N' (vide apêndice código [2.a] e [2.b]), se calcula as lacunas de distribuição entre eles representada gráficamente(fig.4.), o que possibilita a visualização da distribuição dos primos no interior do círculo, onde após a distribuição determinada podemos mapear a 'frequência' **(*ζ*)** de distribuição desta no círculo de raio N conforme os resultados dos gráficos abaixo obtidos ao rodarmos os códigos python ([2.c] e [2.d]).

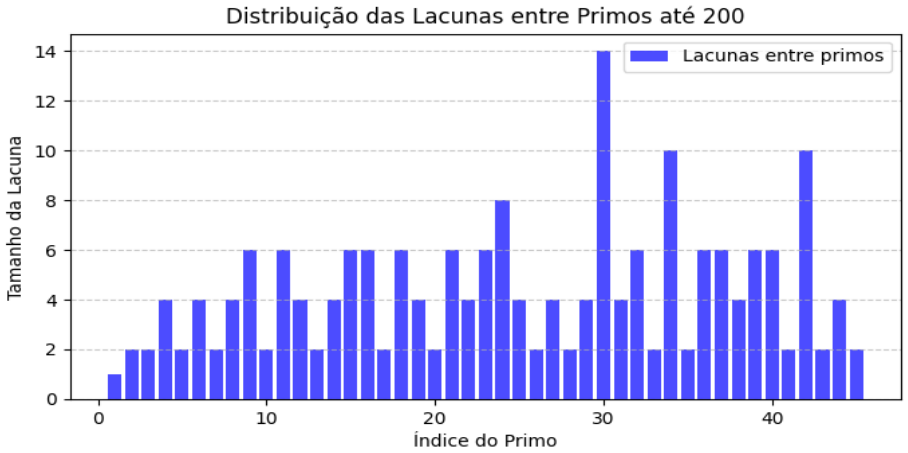

Figura 4: Distribuição das Lacunas (L) obtidas pelo uso da função (ζ)

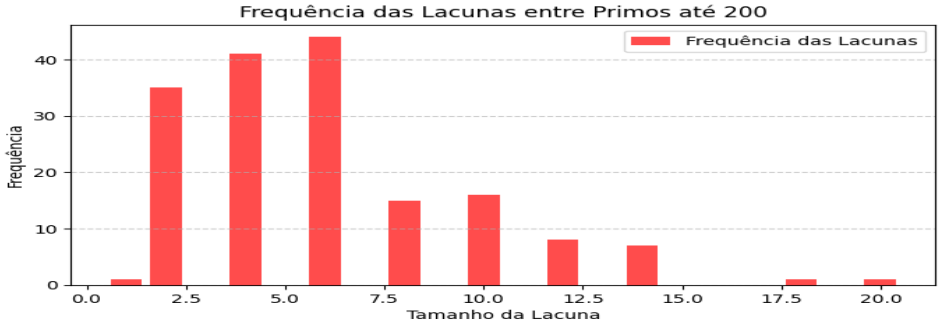

Figura 5: Densidade dos primos nas lacunas (L) =200

A função zeta de Rienmann **(*ζ*)** esta profundamente ligada à esta distribuição dos números primos, onde sua forma:

$$\zeta(s) = \sum_{k=1}^{\infty} \frac{1}{n^s}$$

possui zeros não triviais que influenciam diretamente a distribuição dos primos.

Se a Hipótese de Reimann for verdadeira, os zeros não triviais estariam alinhados em: $\mathrm{Re}(s) = \frac{1}{2}$, o que implica em um padrão de distribuição dos primos; especialmente sua versão explicita (fórmula de Riemann - von Mangoldt) que consegue prever a densidade esperada dos primos resultando como consequência , as lacunas médias entre eles.

No entanto, a distribuição *exata* dos primos é influenciada por oscilações que não são totalmente capturadas apenas pela função zeta, onde podemos concluir que o gráfico de lacunas baseado na contagem direta dos primos tem detalhes específicos que não estão visíveis na função zeta sozinha. Poderiamos aplicar a 'transformada inversa' da função zeta (ζ) para estimar a contagem de primos $\pi(x)$, aproximada das lacunas. No entanto para pequenas escalas (como até N=200) os efeitos individuais dos primos são pronunciados e a contagem direta é mais precisa. Já para grandes números a função zeta (ζ) se torna mais útil, pois aproxima a tendência geral da lacunas entre os primos (p.ex. $N \geq 10^4$).

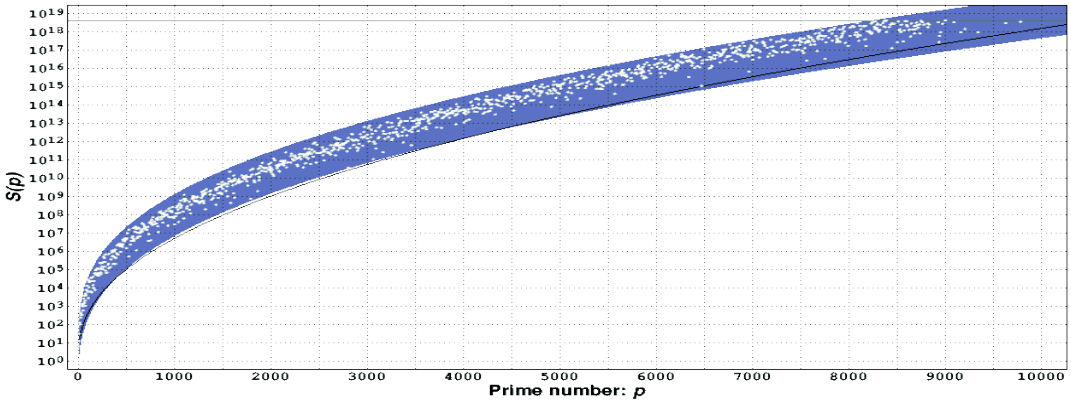

Figura 6: 'Roseta de Goldbach'

Paralelamente, foi construído o gráfico de contagem de Goldbach(ver figura 6), focalizando unicamente os números primos que diferentemente do círculo de Gauss, este gráfico revela uma distribuição densa e estruturada, caracterizando o que neste trabalho denomino de *'Estrutura Externa do Universo'*, e é composta exclusivamente pela relação entre primos e pares de primos.

O gráfico mostra a soma **S*(p)*** dos primos em função do número ***p***, com uma distribuição que segue um padrão curvado e uma faixa de densidade. O padrão lembra uma anel segmentado inclinado ou até mesmo um cinturão orbital, como os anéis de Saturno, além disso há zonas de baixas densidade dentro da faixa que indicam valores onde a soma dos primos não ocorre com frequência, isso sugere lacunas estruturais na conjectura de Goldbach.

Projetando isso em três dimensões usando pouco esforço computacional, temos algo próximo

a uma estrutura helicoidal o que permitiu uma melhor visualização desta estrutura,

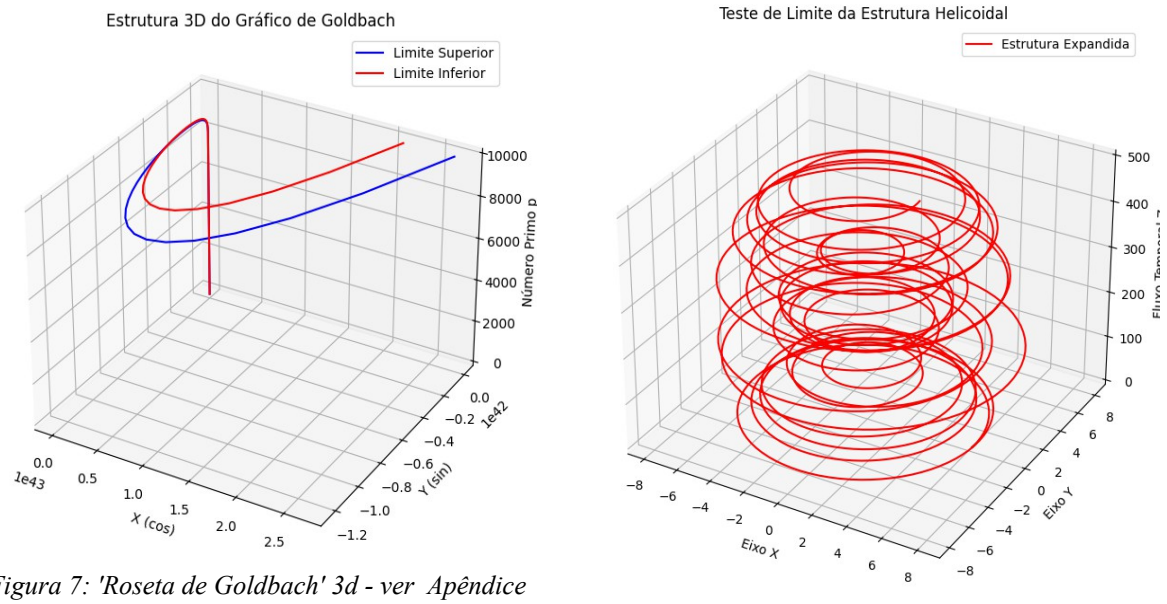

Figura 7: 'Roseta de Goldbach' 3d - ver Apêndice - código [ ]

foi necessário extrair uma fórmula para converter a relação de ***S(p)*** e ***p*** em um modelo funcional que identificasse se a curva segue alguma função conhecida, mostrando uma possível relação consistente, permitindo prever comportamentos da função para valores ainda maiores daqueles que constam no gráfico de Goldbach(até 10^{17} e além). A equação para a faixa do gráfico foi ajustada como:

$$S(\boldsymbol{p}) \approx \quad e^{\sqrt{P}}$$

e tem conexão com as funções exponenciais, a teoria dos números, e esta ligada às identidades de Euler; as expressões associadas ao crescimento assintótico de funções relacionadas aos primos é a ***função zeta de Riemann*** e a relação de Euler com os primos nos diz que a série infinita dos primos diverge pois:

$$\sum \frac{1}{p} \rightarrow \infty$$

isso indica que os primos tem uma relação com o crescimento exponencial onde a identidade de Euler para a função zeta é dada por:

$$\boldsymbol{\zeta(s) = \Pi_p} \quad \frac{1}{1-p^{-s}}$$

e tem comportamento próximo de funções exponenciais apenas em algumas regiões, sendo a equação correta e a que captura o comportamento da distribuição da curva de Goldbach :

$$S(\boldsymbol{p}) \approx \quad e^{\sqrt{P}} \quad .$$

Os códigos usados para a analise da distribuição e modelos que capturam a idéia de uma estrutura composta pela distribuição dos primos, foi revelada pela fórmula de Euler e Reimann que apontam um padrão geométrico profundo tal que a soma de dois primos formando um número par, não é um acaso mas uma consequência da propria estrutura numérica do universo onde, a equação de Euler, além de descrever padrões fundamentais na matemática também modela a 'Roseta de Goldbach', sugerindo que tudo isso faz parte de uma estrutura unificada e cósmica que segue uma ordem matemática profunda.

3. Conclusão

A presente pesquisa ainda esta em construção, análise e validação dos modelos, onde são utilizados algoritmos desenvolvidos em ambientes de Modelagem Computacional capazes de otimizar a geração gráfica e identificar automaticamente as lacunas, simetrias e padrões geométricos. Os algoritmos foram validados por meio da replicação dos modelos para diferentes faixas numéricas, verificando a manutenção das estruturas e lacunas observadas onde abordagem metodológica permitiu uma análise visual e matemática simultânea, fundamentando a hipótese de que os números primos desempenham um papel estruturador tanto no plano interno quanto externo da configuração do universo. Um resultado promissor desta pesquisa, sugere que a expansão do universo pode ser interpretada como a dilatação da estrutura helicoidal ao longo do tempo, de forma analoga a um material que se expande quando aquecido. Se esta analogia estiver correta então, o raio da helice pode inflenciar diretamente a temperatura do universo onde a taxa de expansão pode vir a ser um coeficiente de dilatação térmica.

4. Referências

- Zagier, Don : "Zetafunktionem und quatratishe Körper"(1981)
- Everest, Graham: "Heights of Polynomials and Entropy in Algebric Dynamics"(1999)
- Granville, Andrew: "Prime Number Patterns"(....)
- Dirac, Paul: "Hipótese dos grandes Números" (1937)

5. Apêndice – **Modelos Computacionais**

1- Código Python: Pares de Primos no Círculo de Raio (N).

```
import matplotlib.pyplot as plt
import numpy as np
from sympy import primerange

# Função para gerar números primos até um valor máximo
def gerar_primos(limite):
  return list(primerange(1, limite + 1))

# Função para verificar se os pares de primos estão dentro do círculo
def encontrar_pares_primos(dmax):
  primos = gerar_primos(dmax)
  pares = []

  for p1 in primos:
    for p2 in primos:
      if p1**2 + p2**2 <= dmax**2:
        pares.append((p1, p2))

  return pares

# Função para plotar os pares no gráfico
def plotar_pares_no_circulo(pares, dmax):
  fig, ax = plt.subplots()

  # Plotando o círculo
    circle = plt.Circle((0, 0), dmax, color='0.5', fill=False, linestyle='-',
linewidth=1.5)
  ax.add_artist(circle)

  # Gerando os espelhamentos nos quatro quadrantes
  x_vals = []
  y_vals = []

  for p1, p2 in pares:
    x_vals.extend([p1, -p1, p1, -p1])
    y_vals.extend([p2, p2, -p2, -p2])

  # Plotando os pares de primos dentro do círculo
  ax.scatter(x_vals, y_vals, color='0.75', marker='.', label="Pares de Primos")

  # Ajustes no gráfico
  ax.set_xlim(-dmax, dmax)
  ax.set_ylim(-dmax, dmax)
  ax.set_aspect('equal', 'box')
  ax.set_title(f"Pares de Primos no Círculo de Raio {dmax}")
  ax.set_xlabel("p1 (primo)")
  ax.set_ylabel("p2 (primo)")
  ax.legend()
```

```
    # Calculando e exibindo a quantidade de pontos
    num_pontos = len(x_vals)
    texto_num_pontos = f"Número de pontos: {num_pontos}"
    ax.text(0.05, 0.95, texto_num_pontos, transform=ax.transAxes, fontsize=10,
            color='black', backgroundcolor='white')


    plt.grid(True)
    plt.show()

# Definindo o limite (raio do círculo)
R = 200  # Você pode mudar esse valor para outro número

# Encontrando os pares de primos dentro do círculo
pares_primos = encontrar_pares_primos(R)

# Plotando o gráfico
plotar_pares_no_circulo(pares_primos, R)
```

2.a - Código Python: "O Crivo de Eratostenes" para encontrar a lista dos números primos até N.

```
import numpy as np

def crivo_eratostenes(N):
    # Criar uma matriz booleana para marcar os números primos (True = primo)
    primos = np.ones(N + 1, dtype=bool)
    primos[:2] = False  # 0 e 1 não são primos

    # Percorrer os números até a raiz quadrada de N
    for p in range(2, int(N**0.5) + 1):
        if primos[p]:
            # Eliminar múltiplos de p (exceto o próprio p)
            primos[p * p: N + 1: p] = False

    return np.nonzero(primos)[0]  # Retorna apenas os índices onde primos[i] é True

# Exemplo: encontrar primos até 200
N = 200
primos = crivo_eratostenes(N)
print(f"Primos até {N}: {primos[:500]} ... {primos[-10:]}")
```

2.b - Código Python: Encontrando a quantidade de Números Primos e de Lacunas no Círculo.

```
from sympy import primerange
# A circunferência do círculo de raio (N)=200
# Encontrar os números primos até o primo anterior a 200
primos_ate_200 = list(primerange(1, 200))

# Calcular as lacunas entre os primos
lacunas = [primos_ate_200[i] - primos_ate_200[i-1] for i in range(1,
len(primos_ate_200))]
```

```
# Número de primos e número de lacunas
num_primos = len(primos_ate_200)
num_lacunas = len(lacunas)

num_primos, num_lacunas
```

2.c - Código Python: Obtenção do gráfico de Distribuição das Lacunas no Círculo de raio N.

```
import matplotlib.pyplot as plt
import numpy as np

# Criar os índices para as barras (posições dos primos)
indices = np.arange(1, len(lacunas) + 1)

# Criar o gráfico de barras
plt.figure(figsize=(8, 4))
plt.bar(indices, lacunas, color='blue', alpha=0.7, label="Lacunas entre primos")

# Rótulos e título
plt.xlabel("Índice do Primo")
plt.ylabel("Tamanho da Lacuna")
plt.title("Distribuição das Lacunas entre Primos até 200")
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Exibir o gráfico
plt.show()
```

2.d- Código python para determinar a frequência entre primos nas lacunas(até N=200)

```
from collections import Counter

# Contar a frequência de cada lacuna
frequencia_lacunas = Counter(lacunas)

# Ordenar os dados para o gráfico
lacunas_unicas = sorted(frequencia_lacunas.keys())
frequencias = [frequencia_lacunas[l] for l in lacunas_unicas]

# Criar o gráfico de frequência das lacunas
plt.figure(figsize=(8, 4))
plt.bar(lacunas_unicas, frequencias, color='red', alpha=0.7, label="Frequência das Lacunas")

# Rótulos e título
plt.xlabel("Tamanho da Lacuna")
plt.ylabel("Frequência")
plt.title("Frequência das Lacunas entre Primos até 200")
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Exibir o gráfico
plt.show()
```

3.a-) A Estrutura de Goldbach (Prime Comet)

```
#A ESTRUTURA DE GOLDBACH

from IPython.display import display
from PIL import Image

# Caminho do arquivo enviado
file_path = "/content/i0.gif"

# Abrir e exibir a imagem
img = Image.open(file_path)
display(img)
```

3.b - Código Python: 'Revelando a Estrutura 3D do Gráfico de Goldbach baseada no código anterior(3.a)'

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Criar uma grade de pontos baseada no comportamento da função S(p)
p_values = np.linspace(1, 10000, 300)  # Simular primos até 10^4 (poderíamos expandir depois)
S_p_upper = np.exp(np.sqrt(p_values))  # Estimativa para limite superior da faixa
S_p_lower = np.exp(np.sqrt(p_values)) * 0.8  # Estimativa para limite inferior (80% do superior)

# Criar coordenadas para o gráfico 3D
theta = np.linspace(0, 2 * np.pi, len(p_values))  # Ângulos para simular anel
X_upper = S_p_upper * np.cos(theta)
Y_upper = S_p_upper * np.sin(theta)
Z_upper = p_values  # Eixo dos primos

X_lower = S_p_lower * np.cos(theta)
Y_lower = S_p_lower * np.sin(theta)
Z_lower = p_values

# Criar figura 3D
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Plotar as faixas superior e inferior
ax.plot(X_upper, Y_upper, Z_upper, color='blue', label="Limite Superior")
ax.plot(X_lower, Y_lower, Z_lower, color='red', label="Limite Inferior")

# Configurações do gráfico
ax.set_xlabel("X (cos)")
ax.set_ylabel("Y (sin)")
ax.set_zlabel("Número Primo p")
ax.set_title("Estrutura 3D do Gráfico de Goldbach")
ax.legend()

# Exibir o gráfico
plt.show()
```

4a - Código Python: 'Revelando a Hélice'

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Definir parâmetros do tubo cilíndrico
theta = np.linspace(0, 10 * np.pi, 500)  # Ângulo ao longo do tubo
z = np.linspace(0, 50, 500)  # Altura do tubo
r_base = 5  # Raio base do tubo

# Criar perturbações para simular variações na densidade dos primos
r_variation = 1.5 * np.sin(0.2 * theta) + 1.0 * np.cos(0.3 * theta)

# Definir coordenadas cilíndricas
x = (r_base + r_variation) * np.cos(theta)
y = (r_base + r_variation) * np.sin(theta)

# Criar gráfico 3D
fig = plt.figure(figsize=(10, 5))
ax = fig.add_subplot(111, projection='3d')

# Plotar o "cano flexível"
ax.plot(x, y, z, color="blue", linewidth=2, label="Estrutura Primordial (Faixa de Goldbach)")

# Configurar rótulos
ax.set_xlabel("Eixo X")
ax.set_ylabel("Eixo Y")
ax.set_zlabel("Fluxo Temporal Z")
ax.set_title("Modelo 3D da Estrutura Primordial de Goldbach")

# Exibir gráfico
ax.legend()
plt.show()
```

4.b - Código Python: 'Testando os Limites da estrutura Helicoidal de Goldbach'

```
# Expandindo a simulação para testar limites da estrutura helicoidal
import matplotlib.pyplot as plt # Import matplotlib.pyplot
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

# Novos parâmetros para simular um tamanho muito maior
theta_large = np.linspace(0, 50 * np.pi, 2000)  # Muito mais voltas na hélice
z_large = np.linspace(0, 500, 2000)  # Altura maior
r_variation_large = 2.0 * np.sin(0.15 * theta_large) + 1.2 * np.cos(0.25 * theta_large)

# Redefinindo r_base para que esteja no escopo atual
r_base = 5  # Raio base do tubo

# Definir coordenadas cilíndricas expandidas
x_large = (r_base + r_variation_large) * np.cos(theta_large)
```

```
y_large = (r_base + r_variation_large) * np.sin(theta_large)
# Criar novo gráfico 3D expandido
fig_large = plt.figure(figsize=(10, 10)) # Now plt is defined and this line should work
ax_large = fig_large.add_subplot(111, projection='3d')

# Plotar a estrutura estendida
ax_large.plot(x_large, y_large, z_large, color="red", linewidth=1.0, label="Estrutura Expandida")

# Configurações do gráfico
ax_large.set_xlabel("Eixo X")
ax_large.set_ylabel("Eixo Y")
ax_large.set_zlabel("Fluxo Temporal Z")
ax_large.set_title("Teste de Limite da Estrutura Helicoidal")

# Exibir gráfico
ax_large.legend()
plt.show()
```