Circunferência Poligonal

*Mário Leite*

...

Muitos dizem que a Matemática surgiu na Grécia com os geômetras: estudiosos e pesquisadores das figuras planas, com régua e compasso; e entre eles, Arquimedes. E sendo o círculo considerada a figura plana mais perfeita, Arquimedes deu importância especial à ela, descobrindo que o contorno do círculo (a circunferência) tinha importância fundamental nos seus estudos. A partir daí descobriu uma razão fundamental dada entre o comprimento da circunferência e o diâmetro do círculo: **PI = C/D**, onde **PI** é uma das constantes mais importantes do Universo. Assim, para qualquer superfície “perfeitamente” circular, medindo o comprimento do seu contorno e dividindo-o pelo respectivo diâmetro interno, o valor encontrado sempre será aproximadamente **3,14**. Na verdade, este valor **PI** (representado pela letra grega ***π***) com apenas *duas* decimais, é o mais conhecido e o mais usualmente empregado; entretanto, já se conseguiu valores impressionantemente grandes; atualmente já se conhece um valor inimaginável: **100 trilhões** (100.000.000.000.000) de decimais

O valor de ***π*** representa um número irracional; isto quer dizer que é impossível obter um círculo perfeito com os valores de sua circunferência e diâmetro representados por números inteiros; por exemplo, não se consegue criar círculo de diâmetro **1** e comprimento da circunferência **3**, pois a razão **3/1 = 3** e **PI** não é **3**! Ou seja: OU o tamanho da circunferência OU o diâmetro têm que ser um número irracional; mas, ambos não podem ter, simultaneamente, valores inteiros.

Teoricamente, partindo de um *triângulo* e aumentando os lados: *quadrilátero*, *pentágono*, *hexágono*, *heptágono*, etc, tende-se a um círculo quando o número de lados tender ao infinito...

Arquimedes usou polígonos inscritos e circunscritos para estimar ***π*** e assim obter um círculo, quando os polígonos tiverem lados extremamente grandes.

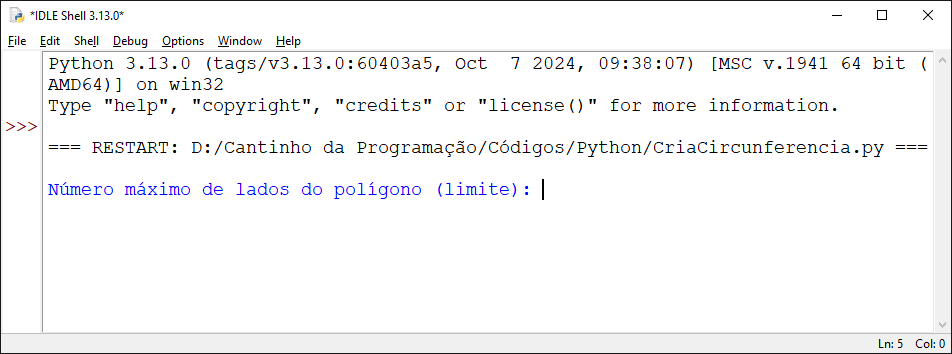
* **Polígono Inscrito**: Um polígono cujos vértices estão na circunferência. O perímetro desse polígono é menor que o comprimento da circunferência.
* **Polígono Circunscrito**: Um polígono cujos lados tangenciam a circunferência. O perímetro desse polígono é maior que o comprimento da circunferência.

Arquimedes começou com um *hexágono* regular (6 lados) e dobrou sucessivamente o número de lados, calculando os perímetros dos polígonos inscritos e circunscritos. Com isto, ele conseguiu limitar o valor de ***π*** entre dois números reais: 3,1408 < ***π*** < 3,1429.

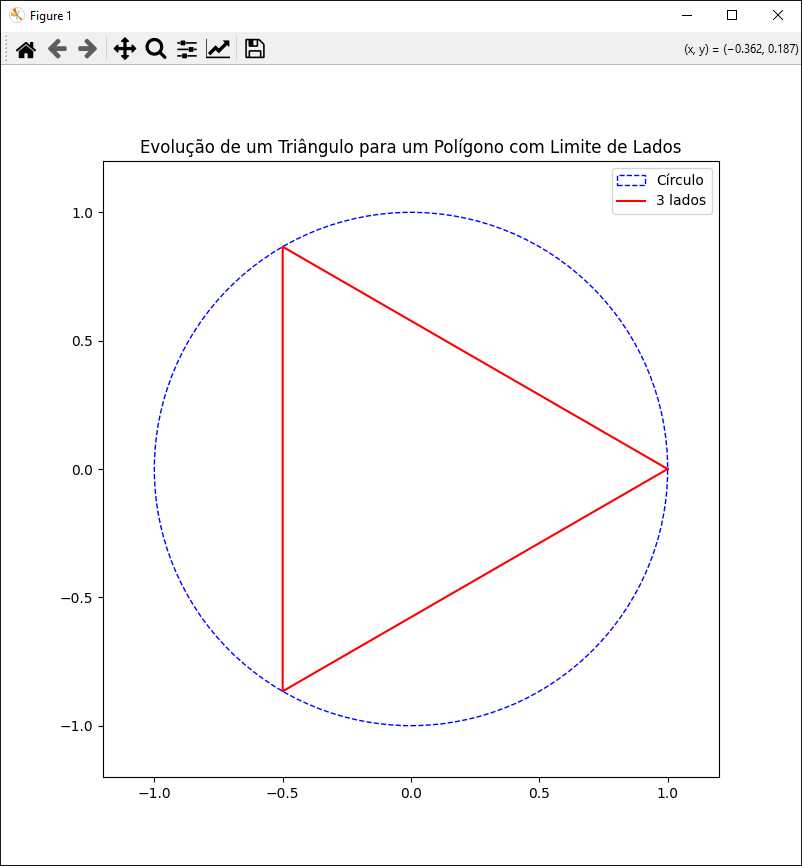
Esse método foi uma das primeiras aproximações rigorosas de ***π***; mas é possível mostrar esse experimento através de computação eletrônica. É isto que o programa **“CircunferenciaPoligonal”**, codificado em Python mostra, de maneira animada, partindo de um *triângulo* equilátero até chegar a um *hectágono*(polígono regular de 100 lados). As **figuras** **2**, **3**, **4**, **5**, e **6** mostram a evolução do formato da circunferência à medida que o número de lados do polígono aumenta. Note que já num polígono de **71** lados (**figura 5**), para os olhos humanos o resultado é quase uma circunferência; não diferindo muito do polígono final de 100 lados (**figura 6**).

Conclusão: A ideia de aproximar uma circunferência por meio de polígonos regulares é um exemplo clássico baseado na genialidade de Arquimedes. Ela nos ensina que, mesmo algo tão simples quanto um círculo, pode esconder profundidades matemáticas surpreendentes. Além disso, o estudo do número ***π*** continua sendo uma área ativa de pesquisa, com aplicações em física, engenharia, ciência da computação e muito mais.

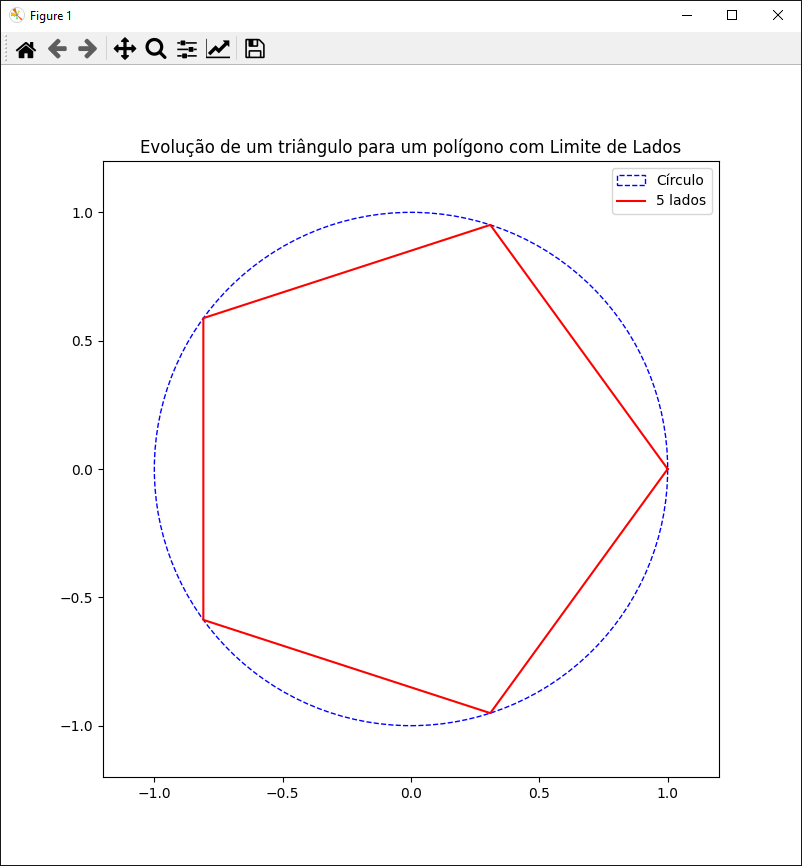
------------------------------------------------------------------------------------------------------------------------



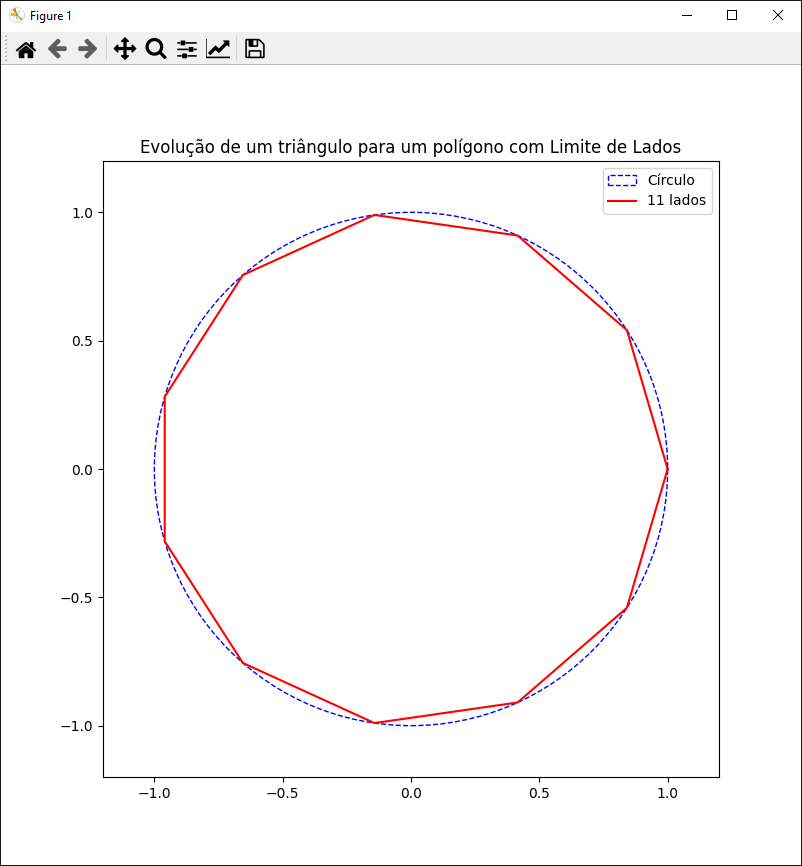
**Figura 1 - Entrada do número de lados do polígono final**



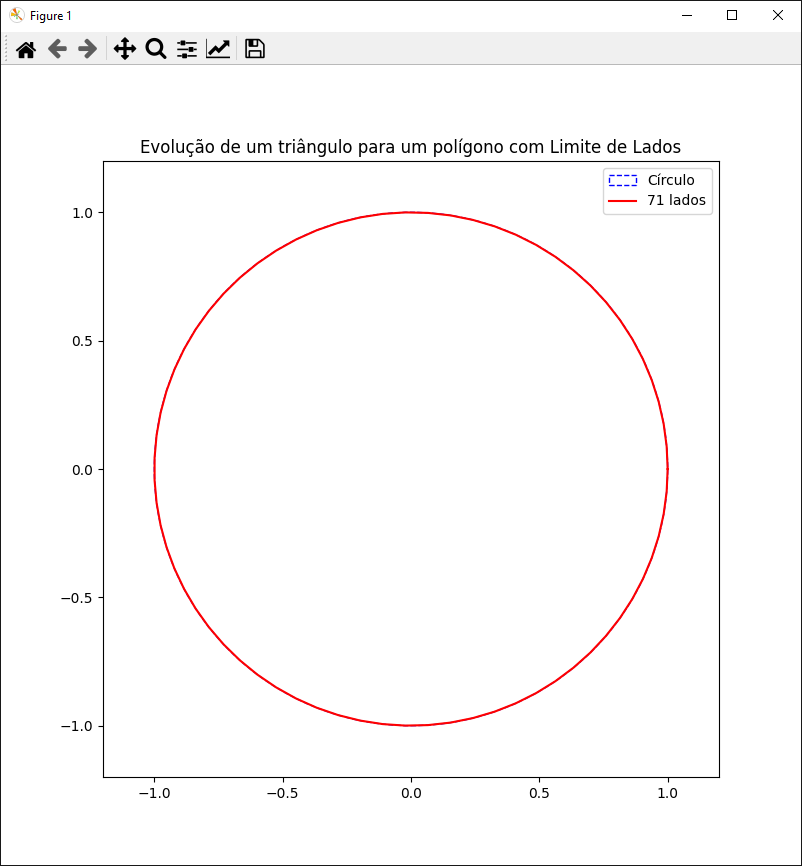
**Figura 2 - Início da simulação: polígono de 3 ladosl**



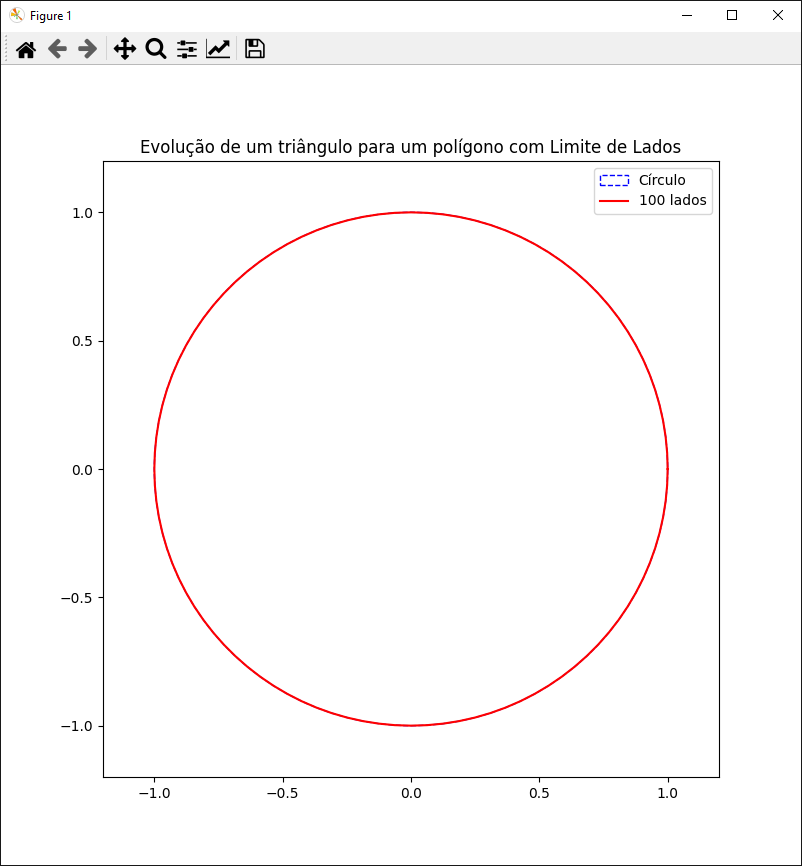
**Figura 3 - Simulação parcial: polígono de 5 ladosl**



**Figura 4 - Simulação parcial: polígono de 11 ladosl**



**Figura 5 - Simulação parcial: polígono de 71 ladosl**



**Figura 6 - Simulação final: polígono de 100 ladosl**

'''

**CircunferenciaPoligonal.py**

---------------------------------------------------------------------------

Simula o experimento de Arquimedes, criando uma circunferência a partir da

incrição de n poligonos.

'''

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** matplotlib.animation **import** FuncAnimation

**def** **PlotarPoligono**(numLados, raio=1):

"""

Calcula os vértices de um polígono regular com 'numLados' lados inscrito em

um círculo de raio 'raio'. """

*#Calcula os ângulos dos vértices do polígono*

angles = np.linspace(0, 2\*np.pi, numLados + 1)

*#Calcula as coordenadas dos vértices*

x = raio \* np.cos(angles)

y = raio \* np.sin(angles)

**return** x, y

*#-------------------------------------------------------------------------------*

*#Função principal*

**def** **main**():

*#Solicita ao usuário o número máximo de lados do polígono (limite)*

**print**("")

**try**:

numLadosFim = **int**(**input**("Número máximo de lados do polígono (limite): "))

**if**(numLadosFim < 3):

**print**("O número mínimo de lados deve ser 3 (triângulo) ")

numLadosFim = 3

**except** **ValueError**:

**print**("Entrada inválida. Usando 100 como padrão.")

numLadosFim = 100

*#Configurações iniciais*

numLadosInicio = 3 *#começa com um triângulo equilátero*

raio = 1 *#raio do círculo circunscrito*

*#Cria uma figura para a plotagem*

fig, ax = plt.subplots(figsize=(8, 8))

ax.set\_aspect('equal')

ax.set\_xlim(-1.2, 1.2)

ax.set\_ylim(-1.2, 1.2)

ax.set\_title("Evolução de um triângulo para um polígono com Limite de Lados")

*#Plota o círculo de referência*

circle = plt.Circle((0, 0), raio, color='blue', fill=False, linestyle='--',

label='Círculo')

ax.add\_artist(circle)

*#Inicia o polígono como um triângulo equilátero*

x, y = **PlotarPoligono**(numLadosInicio, raio)

LinhaPoligonal, = ax.plot(x, y, label=f'**{numLadosInicio}** lados', color='red')

*#Função de atualização para animação*

**def** **Atualizar**(frame):

numLados = numLadosInicio + frame *#aumenta o número de lados*

x, y = PlotarPoligono(numLados, raio)

LinhaPoligonal.set\_data(x, y) *#atualiza os dados do polígono*

LinhaPoligonal.set\_label(f'**{numLados}** lados') *#atualiza a legenda*

*#Remove a legenda antiga e adiciona uma nova atualizada*

ax.clear()

ax.set\_aspect('equal')

ax.set\_xlim(-1.2, 1.2)

ax.set\_ylim(-1.2, 1.2)

ax.set\_title("Evolução de um triângulo para um polígono com Limite de Lados")

ax.add\_artist(plt.Circle((0, 0), raio, color='blue', fill=**False**,

linestyle='--', label='Círculo'))

ax.plot(x, y, label=f'**{numLados}** lados', color='red')

ax.legend(loc='upper right')

**return** LinhaPoligonal,  *#retorna uma tupla com um único elemento*

*#Cria a animação com delay de 0,5 segundos (500 milissegundos)*

frames = numLadosFim - numLadosInicio + 1

ani = FuncAnimation(fig, **Atualizar**, frames=frames, interval=500, blit=**False**)

*#Mostra a animação*

plt.show()

*#===============================================================================*

*#Programa principal*

**if**(\_\_name\_\_ == "\_\_main\_\_"):

**main**()

**#Fim do programa "CircunferenciaPoligonal"**