Elisa Rwagasana Ishimwe

Scientific computing: Activity 7

September 12,2022

**PART 1**

1.

def sumOfTheFirst50():

sum=0

n=1

for iter in range(50):

sum=sum+n

n=n+1

return sum

print (sumOfTheFirst50())=1275

**PART 2**

**SECTION A**

1.

import turtle

t=turtle.Turtle()

t.speed('fastest')

# This function draws a polygon with a parameter specified number of sides

def drawPolygon(numSides):

t.setheading(90)

t.penup()

t.goto(-300,0)

t.pendown()

for sides in range(numSides):

t.forward(50)

t.right(360/numSides)

for p in range(3,31):

drawPolygon(p)

**SECTION B**

2.

#The function below approximates the number pi by using the Archimedes method using a decagon

def archimedesDecagon():

side = 2 \* math.sin(math.radians(36/2.0))

polygonCircumference = 10 \* side

return (polygonCircumference / 2)

print(archimedesDecagon())

#Pi approximation by Decagon is greater than the one with Octagon meaning that to get closer to the value of pi we have to keep increasing the sides of the polygon.

3.

#This program has a function that calculates pi using Archimedes’ approximation

#The function mathematically creates a polygon of a parameterized number of sides and then uses that value to algebraically approximate pi

def archimedesPiApprox(numsides):

circApprox = (numsides \* 2 \* math.sin(math.radians(360/(2\*numsides))))

piApprox = circApprox / 2

return piApprox

print("Pi with 250 sides: ", archimedesPiApprox(250))

def archimedesPiApprox(numsides):

circApprox = (numsides \* 2 \* math.sin(math.radians(360/(2\*numsides))))

piApprox = circApprox / 2

return piApprox

for p in range(3,251):

print("Pi with ",p," sides: ", archimedesPiApprox(p))