ArchiPi

March 3, 2020

0.1 Método de arquimedes para Pi

Considere uma circunferência de diâmetro 1. O número π é exatamente o diâmetro desta circunferência. O método de Arquimedes consiste em considerar, para cada número natural n, os polígonos regulares inscritos e circunscritos, com 2^n lados. O perímetro do polígono inscrito chamaremos de p_n e o do circunscrito P_n. As relações $p_n < \pi < P_n$ são claras e além disso $P_n, p_n \to \pi$ quando $n \to \infty$.

Usando relações trigonométricas podemos obter as equações:

$$P_{n+1} = \frac{2P_n p_n}{(P_n + p_n)}$$

$$p_{n+1} = \sqrt{p_n P_{n+1}}$$

[13]:
```
import math as m
# usaremos a biblioteca math para a raiz quadrada
def pidearquimedes(n):
    ''' Acha a aproximação inferior e superior até o n-esimo termo '''
    x= 4
    y=2*m.sqrt(2)
    k=2
    while k <= n :
        print( k ," - ", y , " < ", x, "\n")
        xnew = 2*x*y/(x+y)
        y = m.sqrt(xnew*y)
        x = xnew
        k += 1
    return (x+y)/2
```

[14]:
```
pidearquimedes(20)
```

```
2  -  2.8284271247461903  <  4

3  -  3.0614674589207187  <  3.3137084989847607

4  -  3.121445152258053  <  3.1825978780745285

5  -  3.1365484905459398  <  3.151724907429257
```

```
6  -  3.1403311569547534  <  3.144118385245905

7  -  3.1412772509327733  <  3.1422236299424577

8  -  3.141513801144302  <  3.1417503691689674

9  -  3.141572940367092  <  3.1416320807031823

10  -  3.14158772527716  <  3.1416025102568095

11  -  3.1415914215112  <  3.1415951177495893

12  -  3.141592345570118  <  3.1415932696293076

13  -  3.141592576584873  <  3.141592807599645

14  -  3.141592634338563  <  3.1415926920922543

15  -  3.1415926487769856  <  3.141592663215408

16  -  3.141592652386591  <  3.1415926559961966

17  -  3.141592653288993  <  3.141592654191394

18  -  3.141592653514593  <  3.1415926537401933

19  -  3.1415926535709926  <  3.141592653627393

20  -  3.1415926535850924  <  3.1415926535991923
```

[14]:
```
3.1415926535903793
```

O cálculo acima foi feito na aritmética de uma máquina, em ponto flutuante, aqui erros de aproximação são importantes. O programa abaixo usa uma implementação de representação decimal para qualquer precisão.

[15]:
```
import decimal

def ArchPi(precision=99):
    # x: circumference of the circumscribed (outside) regular polygon
    # y: circumference of the inscribed (inside) regular polygon

    decimal.getcontext().prec = precision+1
    D=decimal.Decimal

    # max error allowed
    eps = D(1)/D(10**precision)
```

```
    # initialize w/ square
    x = D(4)
    y = D(2)*D(2).sqrt()

    ctr = D(0)
    while x-y > eps:
        xnew = 2*x*y/(x+y)
        y = D(xnew*y).sqrt()
        x = xnew
        ctr += 1

    return str((x+y)/D(2))
```

[16]:
```
print(ArchPi(201))
```

```
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344612847564823378678316527120190914564856692346034861045432664821339360726024914127372458700660631558817488152092096282925409171536436789259036001133053054882046652138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277053921717629317675238467481846766940513200056812714526356082778577134275778960917363717872146844090122495343014654958537105079227968925892354201995611212902196086403441815981362977477130996051870721134999999837297804995105973173281609631859502445945534690830264252230825334468503526193118817101000313783875288658753320838142061717766914730359825349042875546873115956286388235378759375195778185778053217122680661300192787661119590921642019890 2
```

[17]:
```
print(ArchPi(1002))
```

```
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344612847564823378678316527120190914564856692346034861045432664821339360726024914127372458700660631558817488152092096282925409171536436789259036001133053054882046652138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277053921717629317675238467481846766940513200056812714526356082778577134275778960917363717872146844090122495343014654958537105079227968925892354201995611212902196086403441815981362977477130996051870721134999999837297804995105973173281609631859502445945534690830264252230825334468503526193118817101000313783875288658753320838142061717766914730359825349042875546873115956286388235378759375195778185778053217122680661300192787661119590921642019890 2
```