A new method for computing number $\pi$

Fernando Alonso Zotes, 03/2021

Abstract
A family of original formulae for computing number $\pi$ and its proof are presented in this paper. An algorithm is proposed to test the results. The new method for computing $\pi$ is interesting from a purely academic point of view, but it is presented with no intention of competing with other efficient formulae already in use for decades, such as the Chudnovsky algorithm [Chudnovsky].

1 Introduction
The intangible exploration of higher dimensions has been the purpose of many mathematicians for decades. Several equations are well known for computing the hypervolume of a $i$-dimensional sphere. In this paper, some of these equations are combined together with the purpose of finding a new method for computing number $\pi$. This new method is interesting from an academic point of view, however it is not as efficient as some of the other techniques already well known for decades [Beckmann] and worldwide used for computing number $\pi$.

2 Development of a new method for computing number $\pi$
The hypervolume $V_i(R)$ of an hypersphere of radius $R$ in an $i$-dimensional space is [NIST]:

$$V_i(R) = k_i R^i$$

Equation 1

with $i \in N$ and:

$$k_i = \begin{cases} i \text{ odd}, & \frac{\pi^{\frac{i-1}{2}}}{\prod_{j=1/2}^{j=i/2} j} \\ i \text{ even}, & \frac{\pi^{i/2}}{(i/2)!} \end{cases}$$

Equation 2

Note the following recursive relationship:

$$k_i = \frac{2\pi}{i} k_{i-2}$$

$$k_1 = 2, \ k_2 = \pi$$

Equation 3

The volume of the $i$-dimensional sphere can also be computed by splitting into slices and adding up their volume [Math]:

$$V_i(R) = 2 \int_{x=0}^{x=R} V_{i-1}(r(x)) \, dx$$

Equation 4

with $r(x) = \sqrt{R^2 - x^2}$. Equation 1 and Equation 4 yield:

$$V_i(R) = 2 \int_{x=0}^{x=R} k_{i-1} r^{i-1}(x) \, dx$$

Equation 5
Equation 5 yields:

\[
V_{i-1}(r(x)) = 2 \int_{y=0}^{y=r(x)} k_{i-2} r^{l-2}(y) \, dy
\]

with \( r(y) = \sqrt{r^2(x) - y^2} \), thus:

\[
V_{i-1}(r(x)) = 2 \int_{y=0}^{y=r(x)} k_{i-2} [r(x)^2 - y^2]^{\frac{l-2}{2}} \, dy
\]

Equation 6

Equation 4 and Equation 7 yield:

\[
V_i(R) = 4 k_{i-2} \int_{x=0}^{x=R} \int_{y=0}^{y=r(x)} [r(x)^2 - y^2]^{\frac{l-2}{2}} \, dy \, dx
\]

Equation 8

According to the generalized binomial theorem of Newton [Coolidge]:

\[
(a - b)^k = \sum_{n=0}^{n=\infty} \left[ (-1)^n \prod_{j=0}^{j=n} \frac{k-j+1}{n!} \right] a^{k-n} b^n
\]

Equation 9

which converges if \( k \in R \), \( |b/a| < 1 \). Let it be \( a = r^2(x) \) and \( b = y^2 \), so \( |y^2/r^2(x)| < 1 \) for \( x \in (0, R) \) and \( y \in (0, r(x)) \), thus Equation 9 yields:

\[
(r^2(x) - y^2)^{\frac{l-2}{2}}
= \sum_{n=0}^{n=\infty} \left[ (-1)^n \prod_{j=0}^{j=n} \frac{2 - j + 1}{n!} \right] r^{(l-2-n)}(x) y^{2n}
= r^{l-2}(x) \sum_{n=0}^{n=\infty} \left[ (-1)^n \prod_{j=0}^{j=n} \frac{2 - j + 1}{n!} \right] r^{-2n}(x) y^{2n}
\]

Equation 10

Integrating Equation 10 yields:

\[
\int_{x=0}^{x=R} \int_{y=0}^{y=r(x)} [r(x)^2 - y^2]^{\frac{l-2}{2}} \, dy \, dx
= \int_{x=0}^{x=R} r(x)^{l-1} P_{i-1} \, dx
= P_{i-1} \int_{x=0}^{x=R} (R^2 - x^2)^{\frac{l-1}{2}} \, dx
\]

Equation 11

with \( P_{i-1} \) independent from \( x \) and equal to:

\[
P_{i-1} = 2 \sum_{i=0}^{n=\infty} \left[ (-1)^n \prod_{j=0}^{j=n} \frac{i-j}{n! (2n+1)} \right]
\]

Equation 12

Integrating Equation 11 yields:

\[
\int_{x=0}^{x=R} \int_{y=0}^{y=r(x)} [r(x)^2 - y^2]^{\frac{l-2}{2}} \, dy \, dx
= \int_{x=0}^{x=R} r(x)^{l-1} P_{i-1} \, dx
= P_{i-1} \int_{x=0}^{x=R} (R^2 - x^2)^{\frac{l-1}{2}} \, dx
\]

Equation 13

According to Equation 11:

\[
\int_{x=0}^{x=R} (R^2 - x^2)^{\frac{l-1}{2}} \, dx = R^l P_l
\]

Equation 14

with:

\[
P_l = 2 \sum_{i=0}^{n=\infty} \left[ (-1)^n \prod_{j=0}^{j=n} \frac{i+1-j}{n! (2n+1)} \right]
\]

Equation 15

Note Equation 12 and Equation 15 are equivalent.
After replacing Equation 13 and Equation 14 in Equation 8 we get:

\[ V_i(R) = 4k_{i-2}R_i^2P_iP_{i-1} \]

Equation 16

Equation 1 and Equation 16 yield:

\[ k_i = 4k_{i-2}P_iP_{i-1} \]

Equation 17

Equation 3 and Equation 17 yield:

\[ \frac{2\pi}{i}k_{i-2} = 4k_{i-2}P_iP_{i-1} \]

Equation 18

Solving Equation 18 for \( \pi \):

\[ \pi = 2iP_iP_{i-1} \]

Equation 19

Equation 19 represents a family of formulae to compute number \( \pi \), with \( i \in \mathbb{N} \) taking any natural value.

3 Algorithm

For implementing Equation 19 in a computer and avoiding factorials of high numbers, the expression of \( P_i \) and \( P_{i-1} \) given by Equation 15 can be written as follows:

\[ P_i = 1 + \sum_{n=1}^{\infty} \frac{(-1)^nQ_{i,n}}{2n + 1} \]

Equation 20

with:

\[ Q_{i,n} = \prod_{j=1}^{n} \left[ \frac{i+1}{2j} - 1 \right] \]

Equation 21

Note the following recursive relationship:

\[ Q_{i,n} = \left( \frac{i+1}{2n} - 1 \right)Q_{i,n-1} \]

\[ Q_{i,1} = \frac{i-1}{2} \]

Equation 22

The algorithm takes into account the fact that \( Q_{i,n} = 0 \) if \( n \geq (i + 1)/2 \) and \( i \) is odd. Equation 20 can be written as follows:

\[ P_i = \begin{cases} 
1 \text{ odd}, & 1 + \sum_{n=1}^{\infty} \frac{(-1)^nQ_{i,n}}{2n + 1} \\
1 \text{ even}, & 1 + \sum_{n=1}^{\infty} \frac{(-1)^nQ_{i,n}}{2n + 1} 
\end{cases} \]

Equation 23

A python script is located at [Alonso], implementing Equation 19, Equation 22 and Equation 23. The code is as follows:

```python
class coef_qq:
    def __init__(self,i):
        self.i=i
        self.v=[(i-1)/2.0]
    def compute(self,n):
        if n>len(self.v):
            # qq(n) is saved in position n-1
            self.v.append(self.compute(n-1)*((self.i+1)/2.0/n-1))
        return self.v[n-1]

class coef_pi:
    def __init__(self,i,Ninf):
        self.qq=coef_qq(i)
        # some terms are cancelled for i odd and n>=(i+1)/2, thus limit
        set to (i-1)/2 if i odd and n>=(i+1)/2, thus limit
        if (i % 2) ==0:
            self.limit=Ninf
        else:
            self.limit=min(Ninf,int((i-1)/2))
        self.v=[0]
    def __computeSumTerm(self,n):
        if (n % 2) ==0:
            out=1.0
        else:
            out=-1.0
        out*=self.qq.compute(n)/(2*n+1)
        return out
    def compute(self):
        sum=1
        for n in range(self.limit):
            # sumatory shall run from 1 to Ninf
            sum+=out
        return sum
```

```python
class coef_pi:
    def __init__(self,i,Ninf):
        self.qq=coef_qq(i)
        # some terms are cancelled for i odd and n>=(i+1)/2, thus limit
        set to (i-1)/2 if i odd and n>=(i+1)/2, thus limit
        if (i % 2) ==0:
            self.limit=Ninf
        else:
            self.limit=min(Ninf,int((i-1)/2))
        self.v=[0]
    def __computeSumTerm(self,n):
        if (n % 2) ==0:
            out=1.0
        else:
            out=-1.0
        out*=self.qq.compute(n)/(2*n+1)
        return out
    def compute(self):
        sum=1
        for n in range(self.limit):
            # sumatory shall run from 1 to Ninf
            sum+=out
        return sum
```
In the algorithm, an upper limit $N$ for the infinity sum in Equation 23 is defined. Obviously, better results are expected for higher values of $N$.

The impact of $i$ and $N$ in the algorithm is quite important. For example, for $i = 5$ and $N = 3000000$ the resulting estimation of number $\pi$ is accurate for up to 11 decimal digits, whereas for $i = 17$ and $N = 130$ the accuracy increases to 15, surprisingly. Therefore, with a good selection of $i$ a high accuracy can be achieved even at lower values of $N$.

4 Conclusions
A family of formulae has been obtained for computing number $\pi$. Proof has been presented. An algorithm has been proposed for computing number $\pi$ based on the presented family of formulae.

5 Future work
As shown above, a good choice of the value of $i$ may yield accurate results without the need of using high values for $N$. Further investigation of values for $i$ and $N$ may lead to interesting conclusions regarding the computational efficiency of the proposed algorithm.

6 References
[Alonso] Fernando Alonso Zotes. Python script for computing number $\pi$ (2021):
https://colab.research.google.com/drive/1bI14mfFHOcJwAOxaKwm7gspt08bDng?usp=sharing

[Beckmann] Petr Beckmann. A History of $\pi$. New York: St. Martin's Press (1971). ISBN 978-0-88029-418-8. MR 0449960.

[Chudnovsky] David V. Chudnovsky and Gregory V. Chudnovsky. Approximation and Complex Multiplication according to Ramanujan. Ramanujan Revisited: Proceedings of the Centenary Conference (held at the University of Illinois at Urbana-Champaign, June 1-5, 1987), 1988.

[Coolidge] J. L. Coolidge. The Story of the Binomial Theorem. The American Mathematical Monthly. Vol. 56, No. 3 (Mar., 1949), pp. 147-157 (11 pages). Published By: Taylor & Francis, Ltd.

[Math] Multiple integral, Encyclopedia of Mathematics, EMS Press, 2001 [1994]

[NIST] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/5.19#E4, Release 1.1.1 of 2021-03-15.