Re: A Formula for Pi

Source: http://sci.tech−archive.net/Archive/sci.math/2008−06/msg02244.html

---

- *From*: Mensanator <mensanator@xxxxxxx>
- *Date*: Sun, 22 Jun 2008 18:23:53 −0700 (PDT)

On Jun 22, 7:29ýpm, r...@xxxxxxxxxxxxxx (Rob Johnson) wrote:

> In article <2e23fae6−a30d−49d8−9520−62d16cecc...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>
> Mensanator <mensana...@xxxxxxx> wrote:
>
>> On Jun 22, 8:48 am, r...@xxxxxxxxxxxxxx (Rob Johnson) wrote:
>>
>>> In article
>>> <rubrum−29BD9A.21370621062...@xxxxxxxxxxxxxxxxxxxxx>,
>>> Michael Press <rub...@xxxxxxxxxxx> wrote:
>>>
>>>> In article
>>>> <41c3a2f3−d399−42f6−b21d−ce897b6ba...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>>>> Mensanator <mensana...@xxxxxxx> wrote:
>>>>
>>>>> On Jun 20, 2:35 pm,
>>>>> r...@xxxxxxxxxxxxxx (Rob
>>>>> Johnson) wrote:
>>>>>
>>>>>> In article
>>>>>> <4f348778−0c41−45e4−bb51−9fcc3dca9...@xxxxxxxxxxxxxxxxxxx
>>>>>> Mensanator
>>>>>> <mensana...@xxxxxxx>
>>>>>> wrote:
>>>>>>
>>>>>>> On
>>>>>>> Jun
>>>>>>> 20,
>>>>>>> 10:57
>>>>>>> am,
>>>>>>> Maury
>>>>>>> Barbato
>>>>>>> <mauriziobarb...@xxxxxxxx>
>>>>>>> wrote:

Jose Carlos Santos wrote:

On 20−06−2008 7:16, Maury Barbato wrote:

I found in the book "The Penguin Dictionary of Curious and Interesting Numbers" by Wells the following formula involving pi

(pi − 3)/4

= sum_{k=1 to infty}[(−1)^(k+1)]/[2k*(2k+1)*(2k+

Is there anybody who knows a proof of this

wonderful

series?

You already got a reply. I only want to remark that the formula is equivalent to

pi − 3 = sum_{k = 1 to oo}(−1)^{k + 1}/(k(2k + 2k + 1)(k +

1)

= ýsum_{k = 1 to oo}(−1)^{k + 1}/(1^2 + + 1}/(1^2 + 2^2 + ... + k^2).

Best regards,

Jose Carlos Santos

A little slip. We have

pi − 3 = sum_{k = 1 to

oo}(−1)^{k + 1}/[k(2k+ 1)(k + 1)] = (1/6)* sum_{k = 1 to oo}(−1)^{k + 1}/(1^2 + 2^2 + ... + k^2)

It's a very very beautiful formula!

Be that as it may, how fast does it converge? How many terms do I have to sum

to
get
100
decimal
place
accuracy?

He never
claimed that
it was an
efficient
way to
compute pi,

I didn't say he claimed it was efficient. I'm _asking_ whether or not it is efficient.

simply
that it was a
beautiful
formula. ý

Fine. It's beautiful. No disagreemrnt there.

It is a
personal
opinion,

Which everyone is entitled to.

but I agree.

ý ý oo
ý ý −−− ý ý
k+1 ý ý ý 1
ý ý > ý (−1)
ý
ý−−−−−−−−−−−−−−
ý ý −−− ý ý
ý ý
2k(2k+1)(2k+2)
ý ý k=1

ý ý ý ý oo
ý ý ý 1 −−−
ý ý k−1 ý 1
ý ý 2 ý ý ý1
ý ý = − > ý
(−1) ý ý(−−
− −−−− +
−−−−) ý ý
[partial
fractions]
ý ý ý 2 −−−
ý ý ý ý ý 2k
ý 2k+1 ý
2k+2
ý ý ý ý k=1

ý ý ý ý ý oo
ý ý ý 1 ý
−−− ý ý k
ý1
ý ý = − + >
ý (−1)
ý−−−− ý ý
ý ý ý
ý[collapse
telescoping
terms]
ý ý ý 4 ý
−−− ý ý ý
2k+1
ý ý ý ý ý
k=1

ý ý ý 1 ý ý
pi
ý ý = − + (
−− − 1) ý ý
ý ý ý ý ý ý ý
ý ý ý ý
ý[Gregory
series]
ý ý ý 4 ý ý 4

ý ý ý pi − 3
ý ý =
−−−−−−
ý ý ý ý 4

Being an alternating series with monotonically decreasing terms, the error is less than 1/(8n^3) after n terms.

Some of us don't know how to do this. But given the series, I can write a program to apply it.

But I would prefer a series that converges in ~300 terms to the one that converges in ~10**34 terms.

How about Ramanujan's series that gets eight decimal places for each term?

```
1 ý ý2.sqrt{2} ý ý ý ý (4n)! ý[1103 +
26390n]
−− = −−−−−−−−− ýsum_n −−−−−−−
−−−−−−−−−−−−−−−
pi ý ý ý9801 ý ý ý ý ý(n!)^4 ý (4*99)^4n
```

Also look up Euler's (who else?)
transformation for accelerating
alternating series. In essence a difference
table.
Abramowitz and Stegun, 3.6.27.

<http://mathworld.wolfram.com/EulersSeriesTransformation.html>

At <http://www.whim.org/nebula/math/eulerxform.html>,
there is a
rigorous proof of the Euler Series Transform.

However, if we apply the acceleration using the identity

```
ý ý −−− ý ý j ýC(m,j) ý ý k ý ý ý 1
ý ý > ý (−1) ý−−−−−−−− = −−− −−−−−−−−−−
ý ý −−− ý ý ý C(n+j,k) ý k+m C(n+m,k+m)
ý ý ýj
```

proven at <http://www.whim.org/nebula/math/binom.html>,
we get

```
ý ý ým
ý ý −−− ý ý j C(m,j)
ý ý > ý (−1) ý−−−−−−
ý ý −−− ý ý ý 1/2+j
ý ý j=0
```

ý ý ý ý1 ý ý ý ý1
ý ý = −−− −−−−−−−−−−−−
ý ý ý m+1 C(1/2+m,m+1)

ý ý ý ý1 ý ý ý (m+1)!
ý ý = −−− −−−−−−−−−−−−−−−
ý ý ý m+1 (1/2+m)...(1/2)

ý ý ý ý ý ý2^{m+1} m!
ý ý = −−−−−−−−−−−−−−−−−−−−−
ý ý ý (2m+1)(2m−1)...(3)(1)

ý ý ý 2^{2m+1} m! m!
ý ý = −−−−−−−−−−−−−−
ý ý ý ý ý ý (2m+1)!

Thus, the Euler Series Transform says

ý ý pi

ý ý ý ý oo
ý ý ý ý −−− ý ý j ý 1
ý ý = 4 > ý (−1) ý−−−−
ý ý ý ý −−− ý ý ý 2j+1
ý ý ý ý j=0

ý ý ý ý ým
ý ý ý ý −−− ý ý j ý 1
ý ý = 2 > ý (−1) ý−−−−−
ý ý ý ý −−− ý ý ý 1/2+j
ý ý ý ý j=0

ý ý ý ý oo ý ý ý ý m
ý ý ý ý −−− ý−m−1 −−− ý ý j C(m,j)
ý ý = 2 > ý 2 ý ý > ý (−1) ý−−−−−−

ý ý ý ý −−− ý ý ý −−− ý ý ý 1/2+j
ý ý ý ý m=0 ý ý ý j=0

ý ý ý ý oo
ý ý ý ý −−− ý−m−1 2^{2m+1} m! m!
ý ý = 2 > ý 2 ý ý −−−−−−−−−−−−−
ý ý ý ý −−− ý ý ý ý ý(2m+1)!
ý ý ý ý m=0

ý ý ý oo
ý ý ý −−− 2^{m+1} m! m!
ý ý = > ý −−−−−−−−−−−−
ý ý ý −−− ý ý(2m+1)!
ý ý ý m=0

Which converges much faster than the Gregory series, better than a
ratio of 2 per term.

This was even easier to implement.

Did you take advantage of the fact that the ratio of one term to the next is m/(2m+1). ýThat makes it a snap to implement:

ý ý ý ý ý ý ý ý ý1 ý 1 2 ý 1 2 3 ý 1 2 3 4
ý ý pi = 2 (1 + − + − − + − − − + − − − − + ...)
ý ý ý ý ý ý ý ý ý3 ý 3 5 ý 3 5 7 ý 3 5 7 9

Uh...that's the original formula I used. I thought
I mentioned that somewhere. No wonder I got virtually
the same number of terms.

However...

pi ý ý time: 0.0 seconds
31415926535897932384626433832795028841971693993751058209749445923078164062ý8620899

pi_also terms: 1328 ý ý time: 0.344000101089 seconds

31415926535897932384626433832795028841971693993751058209749445923078164062ý8620899

pi == pi_also: True

arctan(1/5) terms: 287 ý ýarctan(1/239) terms: 85 ý ý time: 0.0780000686646 seconds

31415926535897932384626433832795028841971693993751058209749445923078164062ý8620899

pi == pi_Machin: True

Euler terms: 1329 ý ý ýtime: 3.0 seconds

31415926535897932384626433832795028841971693993751058209749445923078164062ý8620899

pi == pi_euler: True

It took just about as many terms as my original formula and is much slower to calculate. And my original can't hold a candle to the Machin formula.

I hope you did not use the factorial function in your implementation,

Well, a bit. But I certainly didn't re−calculate m! every term.
Obviously, not quite as efficient as the original, so there's
some needless calculation in my Euler algorithm.

> but used the term to term ratio I pointed out above. ýIf so, it should
> be a little over

"Starnge, the line went dead just as Professor Press
was about to reveal the secret formula."

> I wasn't trying to get the quickest converging series, I was trying
> to show that Euler Series Accleration can take a terribly slowly
> converging series, like the Gregory Series, and make it into a
> fairly quickly converging series.

Well, I'm not trying to find the quickest either, just an
easy to implement that does a reasonable job. I'm not trying
to do a billion digits or anything (this algorithm isn't
up to it).

> The series that I accelerated from the Gregory series has only a
> term to term ratio of about 2, which yields about .3 digits per term
> (about 1335 terms for 402 digits).
>
> The arccot(5) part of the Machin formula has a ratio of about 25, and
> yields about 1.4 digits per term; the arccot(239) part has a ratio of
> about 57121, and yields more than 4.75 digits per term. ýThis comes
> out to about 1.08 digit per term altogether (about 372 terms for 402
> digits).
>
> The quickest converging Machin−like series I have seen is Gauss's
>
> ý ý pi = 48 arccot(18) + 32 arccot(57) − 20 arccot(239)
>
> which yields about 1.12 digits per term altogether (about 359 terms
> for 402 digits). ýHowever, I just looked up on MathWorld, and there
> it has
>
> ý ý pi = 732 arccot(239) + 128 arccot(1023) − 272 arccot(5832)
>
> ý ý ý ý+ 48 arccot(110443) −48 arccot(4841182) − 400 arccot(6826318)
>
> which comes in at over 1.51 digits per term.

Thanks for the info, I may try one of these just for laughs.

I'm certainly less concerned about the difference
between 372 and 359 than between 372 and 10^34.

Rob Johnson <r...@xxxxxxxxxxxxxx>
take out the trash before replying
to view any ASCII art, display article in a monospaced font

.