Re: A Formula for Pi

Source: http://sci.tech−archive.net/Archive/sci.math/2008−06/msg02241.html

---

- *From*: Michael Press <rubrum@xxxxxxxxxxx>
- *Date*: Sun, 22 Jun 2008 17:50:15 −0700

In article
<4a6c3352−a021−4f21−9009−bcdfff53043d@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
Mensanator <mensanator@xxxxxxx> wrote:

> On Jun 21, 11:37?pm, Michael Press <rub...@xxxxxxxxxxx> wrote:
>
>> In article
>> <41c3a2f3−d399−42f6−b21d−ce897b6ba...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>>
>> ?Mensanator <mensana...@xxxxxxx> wrote:
>>
>>> On Jun 20, 2:35?pm, r...@xxxxxxxxxxxxxx (Rob Johnson)
>>> wrote:
>>>
>>>> In article
>>>> <4f348778−0c41−45e4−bb51−9fcc3dca9...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>>>>
>>>> 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+2)]

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)

= ?sum_{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. ?

Fine. It's beautiful. No disagreemrnt there.

It is a personal opinion,

Which everyone is entitled to.

but I agree.

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

? ? ? ? oo
? ? ? 1 −−− ? ? k−1 ? 1 ? ? 2 ? ? ?1

? ? = − > ? (−1) ? ?(−− − −−−− + −−−−) ? ?
[partial fractions]
? ? ? 2 −−− ? ? ? ? ? 2k ? 2k+1 ? 2k+2
? ? ? ? k=1

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

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

? ? ? pi − 3
? ? = −−−−−−
? ? ? ? 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 ? ?2.sqrt{2} ? ? ? ? (4n)! ?[1103 + 26390n]
−− = −−−−−−−−− ?sum_n −−−−−−− −−−−−−−−−−−−−−−
pi ? ? ?9801 ? ? ? ? ?(n!)^4 ? (4*99)^4n

> Thanks for the suggestion, but I'm trying
> to do this solely with unlimited precision
> integers and rationals. No floats take part
> in the calculations, only the creation of the
> digit string used to test for convergence.

There is a simple recurrence for successive rational
approximations to sqrt{2} − 1:

{a_n} = 0, 1, 2, 5, 12, 29, 70, 169, ...

```
a_{n−1}
−−−−−−− −−> sqrt{2} − 1
a_n
```

a_{n+1} = 2.a_n + a_{n−1}

(a_n + a_{n−1})^2 − 2.(a_n)^2 = +/− 1

so

```
|a_n + a_{n−1} | 1
|−−−−−−−−−−−−− − sqrt{2} | < −−−
| a_n | a_n
```

Or you could run a Newton iteration for successive rational approximations.

> I tried Dave Seaman's suggestion of the Machin
> formula and reduced the number of terms from 335
> to 95 for 100 digit precision. That's probably
> good enough for my purposes.
>
> I actually don't care about pi as there's a
> GMP function that's faster than anything I
> could come up with. It was simply an exercise
> in getting an arbitrary precicion float from
> a series of unlimited precision rationals.
> I thought the algorithm might come in handy
> sometime in case I need this for something
> other than pi.

At least one of the industrial strength programs
to calculate a high approximation of pi used Ramujan's series.

−−
Michael Press
.