HP Forums / HP Calculators (and very old HP Computers) / General Forum / [VA] SRC#001 - Spiky Integral

Pages (3): 1 2 3 Next »

NEW REPLY

[VA] SRC#001 - Spiky Integral

Threaded Mode | Linear Mode

07-11-2018, 12:10 AM **Post: #1**

Valentin Albillo
Senior Member

Posts: 347
Joined: Feb 2015
Warning Level: 0%

[VA] SRC#001 - Spiky Integral

.

Hi all, welcome to my ***SRC#001 - Spiky Integral***:

Here I'll deal with a real-world math problem not of my own making but which did appear at a certain math competition. The problem introduces this *"spiky"* integral:

$$I(N) = \int_0^{2\pi} \cos(x)\cos(2x)\cos(3x)\ldots\cos(Nx)\,dx$$

and asks ***for which values of N in the interval [1,10] does I(N) have a non-zero value***.

It then specifies that the result must include proof of correctness but in my extensive experience with problems of this sort I've found that it's best to *first obtain the correct result* using whatever means (praying included, bribing not excluded), as this will usually provide a most helpful "hint" to afterwards try and get proof of its validity.

With this strategy in mind (and if praying and bribing have already proved ineffective), we can use our beloved **HP calcs** (yes, **HP calcs**, not *Excel*, not *Mathematica*, not *Wolfram Alpha*, not *Maple*, not *Matlab*, not *laptops*, you get the drift) to get some numerical evidence first, then use it to make an educated conjecture on what the correct result might probably be.

As for numerical evidence, this is how I'd obtain it using an **HP-15C** and a very simple, straightforward little program I wrote which goes as follows:

The program segment which computes *Cos(x)*Cos(2x)*Cos(3x)*...*Cos(Nx)*, where **N** is assumed to be in **R0** and **x** in stack register **X** (the display), is placed under ***LBL B*** and the main program which computes the integral **I(N)** for **N** from 1 to 10 is placed after it under ***LBL A***:

```
001  LBL B       015  LBL A
002  STO 1       016  1.01
003  RCL 0       020  STO 0
004  INT         021  LBL 0
005  STO I       022  0
006  1           023  PI
007  LBL 1       024  2
008  RCL I       025  x
009  RCLx 1      026  INTEG B
010  COS         027  RCL 0
011  x           028  R/S
012  DSE I       029  X<>Y
013  GTO 1       030  R/S
014  RTN         031  ISG 0
                 032  GTO 0
```

To run it, simply press:

```
[USER] [RAD] [FIX 3]

[A]   ->  1.010    [R/S] -1.083 -04
[R/S] ->  2.010    [R/S] -5.974 -05
[R/S] ->  3.010    [R/S]  1.571
[R/S] ->  4.010    [R/S]  0.785
[R/S] ->  5.010    [R/S]  3.778 -08
[R/S] ->  6.010    [R/S]  2.733 -09
[R/S] ->  7.010    [R/S]  0.393
[R/S] ->  8.010    [R/S]  0.344
[R/S] ->  9.010    [R/S] -3.600 -04
[R/S] -> 10.010    [R/S]  1.664 -09
```

Notes:

*- It's best to run this on much faster, modern **HP-15C**-based hardware or a fast software emulator as otherwise it will take several hours for a physical vintage **HP-15C** to get the results above, mostly because the function being integrated gets more an more "spiky" as **N** grows and so it takes more and more time to integrate it accurately.*

*- FIX 3 is recommended for speed as FIX 4 or higher takes significantly longer and the extra accuracy isn't needed here. On the other hand, FIX 2 can't be used in the **HP-15C** or the **HP-41C/Advantage** versions because for **N**=4 we get:*

I(4) = 0.86 +- 0.03

*which is **wrong**, the uncertainty is lying. This is similar to what Mr. Kahan said about one of his sample integrals in some issue of the **HP Journal**: the error is too small to notice but too big to ignore. The solution is of course to use FIX 3 as above or SCI 2, which also gives the correct result, namely:*

I(4) = 7.85E-01 +- 1.48E-03

With this evidence in mind and regarding anything smaller than 1E-3 as **0** (as we are computing the integrals in FIX 3) we can see that **I(N) is non-zero for N=3, 4, 7 and 8**.

Using this numerical evidence I have my own conjecture on what the result will be for **general N**, which I'll post in a few days together with my verbatim conversion of this **HP-15C** program for the **HP-41C/Advantage ROM** (which runs very fast in modern hardware/emulators) as well as a *completely different 6-line enhanced version* for the **HP-71B/Math ROM** which profitably uses a technique discussed in my latest ***S&SMC#23*** and further includes a couple of *very nice extras* not present in the other versions.

First though, **I'd like to see what your own conjectures are** (computing for N>10 might help to check them out) and of course **your very own programs** for your preferred hardware (*RPL* comes to mind, *Prime's PPL*, even *SHARP* or *Casio* models ...) or, if you're not up to the task (you know who you are), at least your very own personal comments or ideas ! (Googling for the solution is totally *laaaaaame,* boo !).

Regards.
V.
.

PM FIND EDIT QUOTE REPORT

07-11-2018, 01:10 PM **Post: #2**

pier4r
Senior Member

Posts: 1,963
Joined: Nov 2014

RE: [VA] SRC#001 - Spiky Integral

Thanks for the post. What does SRC means in this case? Source problem?

Wikis are great, Contribute :)

EMAIL PM FIND QUOTE REPORT

07-12-2018, 06:23 AM **Post: #3**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Valentin Albillo Wrote:** (07-11-2018 12:10 AM)
> Using this numerical evidence I have my own conjecture on what the result will be for **general N**, which I'll post in a few days...

Hello, Valentin,

Here is my conjecture for the exact result when N = 39:

$$\frac{756388295}{68719476736}\pi$$

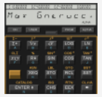
No programs. I’ve evaluated the integrals for N up to 12 on my CASIO fx-991 LA X, which does it fast enough and gives exact results for N = 1, 2, 3, 4, 7 and 8. No googling, except for a OEIS sequence.

Best regards,

Gerson.

EMAIL PM FIND QUOTE REPORT

07-12-2018, 08:10 AM **Post: #4**



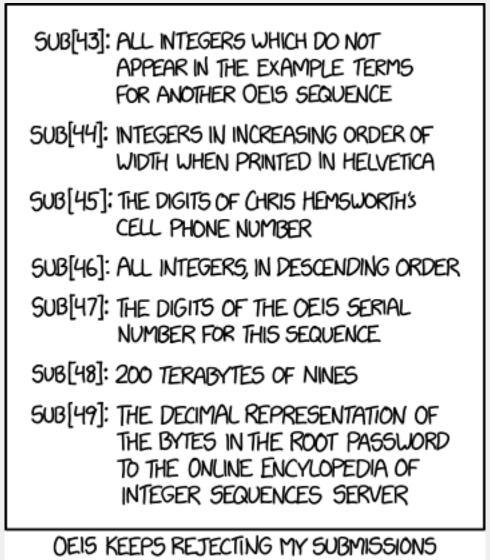
Massimo Gnerucci
Senior Member

Posts: 1,718
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Gerson W. Barbosa Wrote:** (07-12-2018 06:23 AM)
> No googling, except for a OEIS sequence.

I bet is one of these!



Greetings,
Massimo

-+×÷ ↔ *left is right and right is wrong*

PM WWW FIND QUOTE REPORT

07-12-2018, 12:32 PM **Post: #5**

Pjwum
Junior Member

Posts: 10
Joined: Jan 2018

RE: [VA] SRC#001 - Spiky Integral

With HP Prime

Code:

```
EXPORT SPIKES()
BEGIN
LOCAL N;
FOR N FROM 1 TO 20 DO
PRINT(int(product(COS(M*X),M,1,N,1),X,0,2*π));
END;
END;
```

we can go beyond 10

0
0
1/2 pi
1/4 pi
0
0
1/8 pi
7/64 pi
0
0
35/512 pi
31/512 pi
0
0
361/8192 pi
657/16384 pi
0
0
2055/65536 pi
1909/65536 pi

suggesting that I(N) will be non-zero if and only if
N = 4k+3 and N = 4k+4, for k = 0, 1, 2..

PM FIND QUOTE REPORT

07-12-2018, 10:10 PM **Post: #6**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Pjwum Wrote:** (07-12-2018 12:32 PM)
>
> With HP Prime
>
> we can go beyond 10
>
> 0
> 0
> 1/2 pi
> 1/4 pi
> 0
> 0
> 1/8 pi
> 7/64 pi
> 0
> 0
> 35/512 pi
> 31/512 pi
> 0
> 0
> 361/8192 pi
> 657/16384 pi
> 0
> 0
> 2055/65536 pi
> 1909/65536 pi

Sure we can:

```
%%HP: T(3)A(R)F(.);
\<< DUPDUP 'X' 1 ROT OVER SWAP
  FOR i OVER i ^ DUP INV + *
  NEXT NIP EXPAND FXND DROP \->STR "*X^" ROT DUPDUP * + 2 / DUP 2 + \->STR "+" + UNROT \->STR + "+" + "X^" ROT + PICK3 SWAP POS PICK3 ROT POS 1 - SUB DUP SIZE 1 -
  IF NOT
  THEN DROP2 0
  ELSE DUP SIZE OVER "+" POS 1 + SWAP SUB OBJ\-> 2 ROT 1 - ^ / \pi *
  END
\>>

'VASRC1' STO

40

%%HP: T(3)A(R)F(.);
\<< { } 1 ROT
  FOR i i VASRC1 +
  NEXT
\>>

EVAL

-->
```

{ 0 0 '1/2*π' '1/4*π' 0 0 '1/8*π' '7/64*π' 0 0 '35/512*π' '31/512*π' 0 0 '361/8192*π' '657/16384*π' 0 0 '2055/65536*π' '1909/65536*π' 0 0 '24955/1048576*π' '46923/2097152*π' 0 0 '316301/16777216*π' '299973/16777216*π' 0 0 '4136805/268435456*π' '15796439/1073741824*π' 0 0 '13853361/1073741824*π' '26585247/2147483648*π' 0 0 '756388295/68719476736*π' '182188585/17179869184*π' }

Not the best method, I fear. Expand Product{ k=1..n, x^N + 1/x^N } and take the coefficient of the power of x corresponding to the Nth triangular number in the numerator (if there is no correspondence, then the result will be zero). That's your numerator. Your denominator is 2^(N - 1). Multiply the resulting fraction by π. The cases where the results are zero should be handled more cleverly, as you have suggested, but this is only a test. The RPL program might not be fast enough on the real HP 50g as N get larger. For N = 12, it takes 16.75 seconds; for N = 20 it takes 118.41 seconds. The evaluation of the integrals would take much, much longer, I guess.

Gerson.

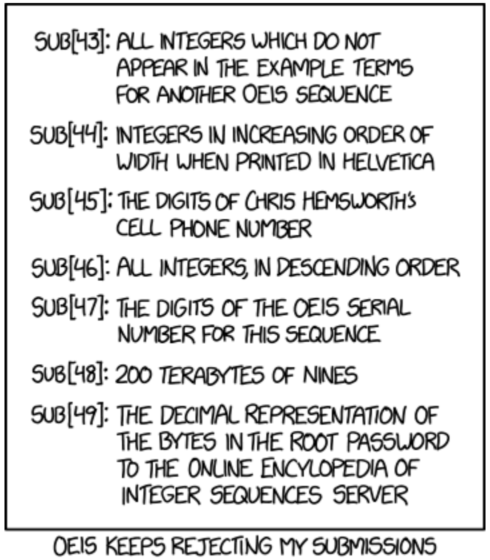
EMAIL PM FIND QUOTE REPORT

07-12-2018, 10:29 PM (This post was last modified: 07-13-2018 12:19 AM by Gerson W. Barbosa.) **Post: #7**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Massimo Gnerucci Wrote:** (07-12-2018 08:10 AM)
>
> > **Gerson W. Barbosa Wrote:** (07-12-2018 06:23 AM)
> >
> > No googling, except for a OEIS sequence.
>
> I bet is one of these!
>
> 
>

I thought this one was not at OEIS yet, but it is:

2, 10, 12, 17, 18, 19, 200, 201, 202...

BTW, at first I searched for 140, 248, but these didn't match anything in the table. 70, 124 was successful.

Edited to fix a typo

Gerson.

EMAIL PM FIND QUOTE REPORT

07-13-2018, 12:43 AM **Post: #8**

Posts: 347

Valentin Albillo
Senior Member

Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC#001 - Spiky Integral

.
Hi, all:

Thanks for your interest in my ***SRC#001***, some excellent contributions so far, much appreciated. Next Sunday night (GMT+1) I'll post my **HP-41C** and **HP-71B** solutions plus extras, but meanwhile I'll comment on some of your recent posts, read on:

> **pier4r Wrote:**
> Thanks for the post. What does SRC means in this case? Source problem?

Nope, **SRC** = **S**emi-**R**egular **C**olumn.

> **Gerson W. Barbosa Wrote:**
> Here is my conjecture for the exact result when **N = 39**: 756388295 / 68719476736*Pi

Fully correct. You might want to check the exact result for **N=71**, namely:

```
I(71) = 335205518724079925 / 73786976294838206464 * Pi
```

which I got by actually *computing* the integral, then *identifying* the resulting constant, using dedicated programs I wrote myself (no *OEIS* involved). It's the hard way but hey, it's fun !

> **Pjwum Wrote:**
> we can go beyond 10 [...] **suggesting that I(N) will be non-zero if and only if N = 4k+3 and N = 4k+4**, for k = 0, 1, 2..

Fully correct as well, we have a winner ! Well done, congratulations. 8-)

> **Gerson (again) Wrote:**
> Sure we can: [...]
>
> *{ 0 0 '1/2*п' '1/4*п' 0 0 '1/8*п' '7/64*п' 0 0 '35/512*п' '31/512*п' 0 0 '361/8192*п' '657/16384*п' 0 0 '2055/65536*п' '1909/65536*п' 0 0 '24955/1048576*п' '46923/2097152*п' 0 0 '316301/16777216*п' '299973/16777216*п' 0 0 '4136805/268435456*п' '15796439/1073741824*п' 0 0 '13853361/1073741824*п' '26585247/2147483648*п' 0 0 '756388295/68719476736*п' '182188585/17179869184*п' }*
>
> Not the best method, I fear. [...] **The evaluation of the integrals would take much, much longer**, I guess.

It would take much, much longer or not, depending on how you go about computing them. Your results are correct, congratulations, but without explaining *why* would you compute the numerators and denominators that way it all seems a "magic trick" unrelated to the problem at hand as the relation to the integral seems shrouded in mystery ... 8-D

Regards.
V.
.

PM FIND EDIT QUOTE REPORT

07-13-2018, 05:54 AM **Post: #9**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

Valentin Albillo Wrote: (07-13-2018 12:43 AM)

Pjwum Wrote:
we can go beyond 10 [...] **suggesting that I(N) will be non-zero if and only if N = 4k+3 and N = 4k+4**, for k = 0, 1, 2..

Fully correct as well, we have a winner ! Well done, congratulations. 8-)

Or, alternatively, if N(N + 1)/2 mod 2 = 0.

EMAIL PM FIND QUOTE REPORT

07-13-2018, 08:52 PM (This post was last modified: 07-14-2018 03:47 AM by ijabbott.)

ijabbott
Senior Member

Posts: 533
Joined: Jul 2015

RE: [VA] SRC#001 - Spiky Integral

You don't really need to do any integration to solve this problem. You just need to use the trig identities to convert the product of cosines into a sum of cosines. When there is a non-zero constant term in the sum, the integra to 2π will be non-zero. The integrals of all the $\cos(kx)$ terms from 0 to 2π will be zero.

The general form for converting the product of cosines to a sum of cosines is:

$$\cos(s)\cos(t) = \frac{1}{2}\cos(s-t) + \frac{1}{2}\cos(s+t)$$

So:

$$\cos(1x) = \cos(1x)$$

$$\cos(1x)\cos(2x) = \frac{1}{2}\cos(1x) + \frac{1}{2}\cos(3x)$$

$$\begin{aligned}\cos(1x)\cos(2x)\cos(3x) &= \frac{1}{2}\cos(1x)\cos(3x) + \frac{1}{2}\cos(3x)\cos(3x) \\ &= \frac{1}{2}\left(\frac{1}{2}\cos(2x) + \frac{1}{2}\cos(4x)\right) + \frac{1}{2}\left(\frac{1}{2}\cos(0x) + \frac{1}{2}\cos(6x)\right) \\ &= \frac{1}{4}\cos(2x) + \frac{1}{4}\cos(4x) + \frac{1}{4}\cos(0x) + \frac{1}{4}\cos(6x) \\ &= \frac{1}{4}\cos(0x) + \frac{1}{4}\cos(2x) + \frac{1}{4}\cos(4x) + \frac{1}{4}\cos(6x)\end{aligned}$$

$$\begin{aligned}\cos(1x)\cos(2x)\cos(3x)\cos(4x) &= \frac{1}{4}\cos(0x)\cos(4x) + \frac{1}{4}\cos(2x)\cos(4x) + \frac{1}{4}\cos(4x)\cos(4x) + \frac{1}{4}\cos(6x)\cos(4x)\\ &= \frac{1}{4}\left(\frac{1}{2}\cos(4x) + \frac{1}{2}\cos(4x)\right) + \frac{1}{4}\left(\frac{1}{2}\cos(2x) + \frac{1}{2}\cos(6x)\right) + \frac{1}{4}\left(\frac{1}{2}\cos(0x) + \frac{1}{2}\cos(8x)\right) + \frac{1}{4}\left(\frac{1}{2}\cos(2x) + \frac{1}{2}\cos(10x)\right)\\ &= \frac{1}{8}\cos(4x) + \frac{1}{8}\cos(4x) + \frac{1}{8}\cos(2x) + \frac{1}{8}\cos(6x) + \frac{1}{8}\cos(0x) + \frac{1}{8}\cos(8x) + \frac{1}{8}\cos(2x) + \frac{1}{8}\cos(10x)\\ &= \frac{1}{8}\cos(0x) + \frac{1}{4}\cos(2x) + \frac{1}{4}\cos(4x) + \frac{1}{8}\cos(6x) + \frac{1}{8}\cos(8x) + \frac{1}{8}\cos(10x)\end{aligned}$$

$$\begin{aligned}\cos(1x)\cos(2x)\cos(3x)\cos(4x)\cos(5x) &= \frac{1}{8}\cos(0x)\cos(5x) + \frac{1}{4}\cos(2x)\cos(5x) + \frac{1}{4}\cos(4x)\cos(5x) + \frac{1}{8}\cos(6x)\cos(5x) + \frac{1}{8}\cos(8x)\cos(5x) + \frac{1}{8}\cos(10x)\cos(5x)\\ &= \frac{1}{8}\left(\frac{1}{2}\cos(5x) + \frac{1}{2}\cos(5x)\right) + \frac{1}{4}\left(\frac{1}{2}\cos(3x) + \frac{1}{2}\cos(7x)\right) + \frac{1}{4}\left(\frac{1}{2}\cos(1x) + \frac{1}{2}\cos(9x)\right) + \frac{1}{8}\left(\frac{1}{2}\cos(1x) + \frac{1}{2}\cos(11x)\right) + \frac{1}{8}\left(\frac{1}{2}\cos(3x) + \frac{1}{2}\cos(13x)\right) + \frac{1}{8}\left(\frac{1}{2}\cos(5x) + \frac{1}{2}\cos(\right.\\ &= \frac{1}{16}\cos(5x) + \frac{1}{16}\cos(5x) + \frac{1}{8}\cos(3x) + \frac{1}{8}\cos(7x) + \frac{1}{8}\cos(1x) + \frac{1}{8}\cos(9x) + \frac{1}{16}\cos(1x) + \frac{1}{16}\cos(11x) + \frac{1}{16}\cos(3x) + \frac{1}{16}\cos(13x) + \frac{1}{16}\cos(5x) + \frac{1}{16}\cos(15x)\\ &= \frac{3}{16}\cos(1x) + \frac{3}{16}\cos(3x) + \frac{3}{16}\cos(5x) + \frac{1}{8}\cos(7x) + \frac{1}{8}\cos(9x) + \frac{1}{16}\cos(11x) + \frac{1}{16}\cos(13x) + \frac{1}{16}\cos(15x)\end{aligned}$$

It's a bit tedious to continue, but you can see a constant term $\frac{1}{4}\cos(0x) = \frac{1}{4}$ in the expansion of $\prod_{i=1}^{3}\cos(ix)$, a constant term of $\frac{1}{8}\cos(0x) = \frac{1}{8}$ in the expansion of $\prod_{i=1}^{4}\cos(ix)$, and the constant term goes away in the ex of $\prod_{i=1}^{5}\cos(ix)$. The constant terms are the coefficients of $\cos(0x)$ in those expansions whose terms consist of cosines of *even* multiples of x.

Your challenge, should you choose to accept it, is to come up with a general formula for each term (or a summation formula for all the terms).

Of course, once you've found the coefficient for the constant $\cos(0x)$ term, calculating the integral from 0 to 2π is a piece of cake! (Just multiply the coefficient by 2π.)

— Ian Abbott

EMAIL PM FIND QUOTE

07-14-2018, 03:30 AM **Post: #11**

rprosperi
Senior Member

Posts: 3,279
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

I'm truly impressed at your diligence writing all those equations before essentially saying "... and so on". I'd probably be at least as impressed at the math, if I followed it, but I just wanted to give you kudos for all the work to explain your analysis.

--Bob Prosperi

EMAIL PM FIND QUOTE REPORT

07-14-2018, 03:39 AM **Post: #12**

ijabbott
Senior Member

Posts: 533
Joined: Jul 2015

RE: [VA] SRC#001 - Spiky Integral

> **rprosperi Wrote:** ⇒ (07-14-2018 03:30 AM)
> I'm truly impressed at your diligence writing all those equations before essentially saying "... and so on". I'd probably be at least as impressed at the math, if I followed it, but I just wanted to give you kudos for all the work to explain your analysis.

Thanks! I'm slowly getting the hang of this MathJax lark!

— Ian Abbott

EMAIL PM FIND QUOTE REPORT

07-14-2018, 04:08 PM (This post was last modified: 07-14-2018 04:09 PM by ijabbott.) **Post: #13**

ijabbott
Senior Member

Posts: 533
Joined: Jul 2015

RE: [VA] SRC#001 - Spiky Integral

Converting the product of cosines $\prod_{i=1}^{n}\cos(ix)$ into a sum of cosines results in the angle multipliers being in the sum of cosines being all odd (when $(n \bmod 4) \in \{1,2\}$) or all even (when $(n \bmod 4) \in \{0,3\}$).

Identity:

$$\cos(s)\cos(t) = \frac{1}{2}\cos(s-t) + \frac{1}{2}\cos(s+t)$$

Let $s = \mathrm{a}x, t = \mathrm{b}x$. Then:

$$\cos(\mathrm{a}x)\cos(\mathrm{b}x) = \frac{1}{2}\cos((\mathrm{a}-\mathrm{b})x) + \frac{1}{2}\cos((\mathrm{a}+\mathrm{b})x)$$

When a and b are both odd or both even, then $(\mathrm{a}-\mathrm{b})$ and $(\mathrm{a}+\mathrm{b})$ are both even, otherwise $(\mathrm{a}-\mathrm{b})$ and $(\mathrm{a}+\mathrm{b})$ are both odd. This results in the factors k of x in the $\cos(\mathrm{k}x)$ terms of the summation switching between all odd and all even after every two $\cos(ix)$ factors are appended to the product of cosines.

As discussed in my earlier post, the summations with all odd k, $\cos(\mathrm{k}x)$ terms all integrate to 0 over the interval $[0, 2\pi]$, but the summations with all even k, $\cos(\mathrm{k}x)$ terms all include a constant term $\mathrm{q}\cos(0x)$ for some positive rational factor q which integrates to $2\mathrm{q}\pi$ over the interval $[0, 2\pi]$.

— Ian Abbott

EMAIL PM FIND QUOTE REPORT

07-14-2018, 10:38 PM **Post: #14**

Thomas Klemm
Senior Member

Posts: 1,449
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **ijabbott Wrote:** ⇒ (07-13-2018 08:52 PM)
> It's a bit tedious to continue

We can use:

$$cos(nx) = \frac{e^{inx} + e^{-inx}}{2}$$

Thus:

$$\begin{aligned} cos(mx)cos(nx) &= (\frac{e^{imx}+e^{-imx}}{2})(\frac{e^{inx}+e^{-inx}}{2}) \\ &= \frac{e^{i(m+n)x}+e^{-i(m+n)x}}{4} + \frac{e^{i(m-n)x}+e^{-i(m-n)x}}{4} \\ &= \frac{cos(m+n)}{2} + \frac{cos(m-n)}{2} \end{aligned}$$

Let's forget the factor $\frac{1}{2}$ for a moment and define:

$$a_k = a^k + a^{-k}$$

For the same reason as above we have:

$$\begin{aligned} a_m a_n &= (a^m + a^{-m})(a^n + a^{-n}) \\ &= a^{m+n} + a^{-(m+n)} + a^{m-n} + a^{-(m-n)} \\ &= a_{m+n} + a_{m-n} \end{aligned}$$

But of course $a_k = a_{-k}$.

This allows us to rewrite it as:

$$a_m a_n = a_{m+n} + a_{|m-n|}$$

We want to calculate the product:

$$p_N = \Pi_{k=1}^{N} a_k$$

Let's assume we already have p_{N-1} represented as a sum of a_k with coefficients b_k:

$$p_{N-1} = \Sigma_{k=1}^{M} b_k a_k$$

Then $p_N = a_N p_{N-1}$ and thus:

$$\begin{aligned} p_N &= a_N \Sigma_{k=1}^{M} b_k a_k \\ &= \Sigma_{k=1}^{M} b_k a_N a_k \\ &= \Sigma_{k=1}^{M} b_k (a_{N+k} + a_{N-k}) \end{aligned}$$

This *Python* program allows us to calculate the coefficients b_k:

Code:

```
def spiky(N):
    b = [0] * N
    b[0] = [0]
    b[1] = [0, 1]
    for k in range(2, N):
        M = len(b[k-1])
```

```
        b[k] = [0] * (M + k)
        for i in range(M):
            b[k][k + i] += b[k - 1][i]
            b[k][abs(k - i)] += b[k - 1][i]
```

Here's the result for the first couple of values:

Code:

```
>>> for b in spiky(11):
...   print b
...
[0]
[0, 1]
[0, 1, 0, 1]
[1, 0, 1, 0, 1, 0, 1]
[1, 0, 2, 0, 2, 0, 1, 0, 1, 0, 1]
[0, 3, 0, 3, 0, 3, 0, 2, 0, 2, 0, 1, 0, 1, 0, 1]
[0, 5, 0, 5, 0, 4, 0, 4, 0, 4, 0, 3, 0, 2, 0, 2, 0, 1, 0, 1, 0, 1]
[4, 0, 8, 0, 8, 0, 7, 0, 7, 0, 6, 0, 5, 0, 5, 0, 4, 0, 3, 0, 2, 0, 2, 0, 1, 0, 1, 0, 1]
```

But we're only interested in the coefficient with the index **0**:

Code:

```
>>> for b in spiky(40):
...   print b[0]
...
0
0
0
1
1
0
0
4
```

Now it's time to deal with the factor $\frac{1}{2}$ that we neglected.
But that's trivial. We just have to add it to each factor a_k.
This leads to: $\frac{1}{2^{N-1}}$

And since we integrate the constant over 2π we loose another 2.

These are the examples for $N = 39$ and $N = 71$:

Code:

```
>>> b = spiky(100)
>>> print b[39][0], '/', 2**37
1512776590 / 137438953472
>>> print b[71][0], '/', 2**69
2681644149792639400 / 590295810358705651712
```

However the ratios haven't been reduced.

> **Gerson W. Barbosa Wrote:** ⇒ (07-12-2018 10:10 PM)
>
> Expand Product{ k=1..n, x^N + 1/x^N } and take the coefficient of the power of x corresponding to the Nth triangular number in the numerator (if there is no correspondence, then the result will be zero). That's your numerator. Your denominator is 2^(N - 1). Multiply the resulting fraction by π.

Not sure if I understood that correctly but it might explain the expression:

$$x^N + \frac{1}{x^N}$$

Thanks both for the challenge and the contributions.

Cheers
Thomas

PM FIND QUOTE REPORT

07-15-2018, 12:54 AM **Post: #15**

Valentin Albillo
Senior Member

Posts: 347
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC#001 - Spiky Integral

.

Hi all,

First of all, thanks for your interest and outstanding contributions, I've really enjoyed them all, much appreciated. As promised, these are my versions for other HP models (**HP-71B** and **HP-41C**) plus relevant comments:

As already mentioned, an educated conjecture might be that ***I(N)*** *is non-zero if the reminder of dividing N by 4 (that is,* ***N mod 4****) is either* ***0*** *or* ***3*** *(i.e.: N=3, 4, 7, 8, 11, 12, 15, 16,...)* which actually is the correct answer.

1) My version for the HP-71B is the following 220-byte 6-liner:

```
1  DESTROY ALL @ DIM S$[200] @ T=1/10^8 @ S$="COS(IVAR)" @ FOR N=1 TO 16
2  IF N>1 THEN S$=S$&"*COS("&STR$(N)&"*IVAR)"
3  S=VAL("INTEGRAL(0,2*PI,"&STR$(T)&","&S$&")") @ IF ABS(S)<T THEN 6
4  CALL IDENTIFY(S/PI,T$) @ T$=T$&"*Pi"
5  DISP USING "3D,':',2DZ.10D,' = ',20A";N,S,T$
6  NEXT N
```

which makes good use of a new technique I discovered long ago and then explained in my recent ***S&SMC#23***, which in this case consist of incrementally creating an arbitrarily large string representing the function to be integrated, and then prepend to it the *INTEGRAL* keyword, the limits of integration and the tolerance before passing it to *VAL* to perform the actual computation. For instance, for N=16 this is the generated string passed to *VAL*:

```
INTEGRAL(0,2*PI,.00000001,COS(IVAR)*COS(2*IVAR)*COS(3*IVAR)*COS(4*IVAR)*COS(5*IVAR)*COS(6*IVAR)*COS(7*IVAR)*COS(8*IVAR)*COS(9*IVAR)
                          *COS(10*IVAR)*COS(11*IVAR)*COS(12*IVAR)*COS(13*IVAR)*COS(14*IVAR)*COS(15*IVAR)*COS(16*IVAR))
```

This means that, for each N, the string is parsed just the one time when passed to *VAL* but the function itself doesn't need any additional parsing no matter how many times the *INTEGRAL* keyword evaluates it, nor are there any loops whatsoever while evaluating it. This all results in extra simplicity (no loops) and much faster execution (neither loops nor calls to a multi-line user-defined function or extra parsing).

As a nice extra, line 4 (which is optional, can be omitted) does *CALL* my *IDENTIFY* subprogram (see my ***"Boldly Going - Identifying Constants"*** article) to *identify* every numeric result after integration and display the *symbolic* value. Finally, taking advantage of ***J-F Garnier****'s* ***Emu71****'s* much greater speed, the results are provided not just up to N=10 but up to **N=16**, and results below the tolerance are considered to be **0** and aren't output. Let's see:

```
>RUN
   3: 1.5707963268 = 1/2*Pi
   4: 0.7853981634 = 1/4*Pi
   7: 0.3926990817 = 1/8*Pi
   8: 0.3436116965 = 7/64*Pi
  11: 0.2147573103 = 35/512*Pi
  12: 0.1902136177 = 31/512*Pi
  15: 0.1384417661 = 361/8192*Pi
```

```
16: 0.1259781722 = 657/16384*Pi
```

Without the identification calls this runs in 19 sec. in my *POPS* system.

2) My version for the HP-41C/Advantage ROM is a *verbatim* port of the one I gave for the *HP-15C*, namely:

```
LBL "FX"
STO 01
RCL 00
INT
STO 02
 1
LBL 01
RCL 02
RCL 01
 x
COS
 x
DSE 02
GTO 01
END
```

```
LBL "CINT"
1.01
STO 00
"FX"
LBL 00
 0
PI
ST+ X
INTEG
RCL 00
STOP
X<>Y
STOP
ISG 00
GTO 00
END
```

To run it (the results are exactly the same as the *HP-15C*'s but obtained *much* faster):

RAD, **FIX 3**
XEQ "CINT" -> 1.010 [R/S] -> -1.083 -04
[R/S] -> 2.010 [R/S] -> -5.974 -05
[R/S] -> 3.010 [R/S] -> **1.571**
[R/S] -> 4.010 [R/S] -> **0.785**
[R/S] -> 5.010 [R/S] -> 3.778 -08
[R/S] -> 6.010 [R/S] -> 2.733 -09
[R/S] -> 7.010 [R/S] -> **0.393**
[R/S] -> 8.010 [R/S] -> **0.344**
[R/S] -> 9.010 [R/S] -> -3.600 -04
[R/S] -> 10.010 [R/S] -> 1.664 -09

With *FIX 2*, for **N=4** we'd get *I(4) = 0.86 +- 0.03*, which is *10% wrong*. Details:

N=4, f(x) = cos(x)*cos(2*x)*cos(3*x)*cos(4*x)

FIX 3: **0.78539818**71 +- 0.003141592705, correct
FIX 2: **0.8**645215735 +- 0.03141592659 , 10% wrong
SCI 2: **0.78539818**71 +- 0.001484303379 . correct

The exact value is *Pi/4 =* ***0.78539816****3397* so we get almost 8 correct digits with *FIX 3* or *SCI 2* but hardly *one* with *FIX 2*.

3) Extra comments: A little theory now.

After some algebraic massaging, any ***product*** of *cosines* of *real* arguments can be converted into a ***sum*** of *exponentials* of *complex* arguments using the following identity:

Cos(x) = (e^(i*x) + e^(-i*x))/2

This can be checked with this little **HP-71B** routine:

```
10 DESTROY ALL @ COMPLEX A,B @ INPUT X
15 !
20 DEF FNC1(X)=COS(X)
```

```
25 DEF FNC2(X)=COS(X)*COS(2*X)
30 DEF FNC3(X)=COS(X)*COS(2*X)*COS(3*X)
35 DEF FNC4(X)=COS(X)*COS(2*X)*COS(3*X)*COS(4*X)
40 !
45 DEF FNE(X)=REPT(EXP((0,X))+EXP((0,-X)))
50 !
55 DEF FNE1(X)=1/2*FNE(X)
60 DEF FNE2(X)=1/4*(FNE(X)+FNE(3*X))
65 DEF FNE3(X)=1/8*(FNE(0)+FNE(2*X)+FNE(4*X)+FNE(6*X))
70 DEF FNE4(X)=1/16*(FNE(0)+2*FNE(2*X)+2*FNE(4*X)+FNE(6*X)+FNE(8*X)+FNE(10*X))
75 !
80 DISP 1;FNC1(X),FNE1(X) @ DISP 2;FNC2(X),FNE2(X) @ DISP 3;FNC3(X),FNE3(X) @ DISP 4;FNC4(X),FNE4(X)
```

where the *FNC1*, *FNC2*, *FNC3*, *FNC4* are user-defined functions (*UDFs*) implementing the **products** of 1, 2, 3 and 4 *cosines*, while the *FNE1*, *FNE2*, *FNE3*, *FNE4* are the equivalent **sums** of 1, 2, 4 and 8 complex *exponentials* (not all necessarily distinct). Let's run it:

```
>RUN

? .2018

1 .979707385531 .97970738553
2 .900990956269 .90099095627
3 .740861912405 .740861912405
4 .512323594236 .512323594236
```

As you may see, every *FNE1..4* gives the same result as the corresponding *FNC1..4* for an arbitrary argument (*.2018*).

What is to be gained by converting a *product* of *cosines* into a *sum* of *exponentials* ? Two main things:

- the *integral of a sum* of functions is the *sum of the integrals* of each function separately
- the integral of a simple *exponential* function is *another* simple *exponential* function so no need for numerical integration

In this way, the integral of the *product* of cosines becomes a *sum* of exponential functions and no numerical approximations are required. There's of course the drudgery of determining the components of the sum but that's another story ! XD

Again, thanks for your interest and regards.
V.
.

PM FIND EDIT QUOTE REPORT

07-15-2018, 04:53 PM (This post was last modified: 07-15-2018 04:55 PM by Gerson W. Barbosa.) **Post: #16**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

Thomas Klemm Wrote: ⇒ (07-14-2018 10:38 PM)

These are the examples for $N = 39$ and $N = 71$:

Code:

```
>>> b = spiky(100)
>>> print b[39][0], '/', 2**37
1512776590 / 137438953472
>>> print b[71][0], '/', 2**69
2681644149792639400 / 590295810358705651712
```

However the ratios haven't been reduced.

In exact mode, the HP 50g does it automatically.

> **Thomas Klemm Wrote:** ⇒ (07-14-2018 10:38 PM)
>
> > **Gerson W. Barbosa Wrote:** ⇒ (07-12-2018 10:10 PM)
> > Expand Product{ k=1..n, x^N + 1/x^N } and take the coefficient of the power of x corresponding to the Nth triangular number in the numerator (if there is no correspondence, then the result will be zero). That's your numerator. Your denominator is 2^(N - 1). Multiply the resulting fraction by π.
>
> Not sure if I understood that correctly but it might explain the expression:
>
> $$x^N + \frac{1}{x^N}$$

I haven't trodden any of the hard (and beautiful) paths you all have done. Instead, I took an unallowed (according to Valentin's rules) and dull shortcut. It was not difficult to recognize pi/2 and pi/4 in his his three-digit results for n = 3 and n = 4. Then I evaluated the integrals for n up to 12 on my CASIO fx-991 LA X, which doesn't take too long to return exact results in terms of pi for small values of n (1/8*pi after 3m 30s, for n = 7; 0.3436116965 after 4m 31s, for n = 8). The pattern soon became apparent: a fraction of pi, the denominator being a power of 2. For denominators = 2^(n - 1), the first numerators, starting with n = 1, are 0, 0, 2, 2, 0, 0, 8, 14, 0, 0, 70, 124, 0..., that is, OEIS sequence A063865.

Quoting from the formula section:

"a(n) = constant term in expansion of Product_{ k = 1..n } (x^k + 1/x^k). - N. J. A. Sloane, Jul 07 2008"

Thus, since

(X + 1/X)(X^2 + 1/X^2)(X^3 + 1/X^3)(X^4 + 1/X^4)(X^5 + 1/X^5)(X^6 + 1/X^6)(X^7 + 1/X^7)(X^8 + 1/X^8)

=

X^36 + 1/X^36 + X^34 + 1/X^34 + X^32 + 1/X^32 + 2 X^30 + 2/X^30 + 2 X^28 + 2/X^28 + 3 X^26 + 3/X^26 + 4 X^24 + 4/X^24 + 5 X^22 + 5/X^22 + 6 X^20 + 6/X^20 + 7 X^18 + 7/X^18 + 8 X^16 + 8/X^16 + 9 X^14 + 9/X^14 + 10 X^12 + 10/X^12 + 11 X^10 + 11/X^10 + 12 X^8 + 12/X^8 + 13 X^6 + 13/X^6 + 13 X^4 + 13/X^4 + 13 X^2 + 13/X^2 + **14**

a(8) = 14

So, the numerator can be found by means of a polynomial expansion. However, when expanding this polynomial on the HP 50g, using the EXPAND command, I got

'(X^72+X^70+X^68+2*X^66+2*X^64+3*X^62+4*X^60+5*X^58+6*X^56+7*X^54+8*X^52+9*X^50+
10*X^48+11*X^46+12*X^44+13*X^42+13*X^40+13*X^38+**14***X^36+13*X^34+13*X^32+13*X^30+12*X^28+11*X^26+10*X^24+9*X^22+8*X^20+7*X^18+6*X^1
6+5*X^14+4*X^12+3*X^10+2*X^8+2*X^6+X^4+X^2+1)/X^36',

which is equivalent to the previous polynomial, except that the constant term in the numerator is 1. But notice 14 is the coefficient in the numerator that corresponds to the power of the denominator (...+14*X^36+.../X^36). Also, 36 = 8*(8 + 1)/2, that is, the 8th triangular number. This has worked for some othe values of n I tried, so I assumed it is a valid property.

The degree of the numerator polynomial is n*(n + 1), which means its size grows quadratically as n increases, which is both memory and time-consuming. Hopefully your method of generating the coefficients is faster (since I don't know python, I can't properly decode your algorithm).

I am disappointed there isn't a formula to directly compute the terms of the sequence. There is an asymptotic formula at OEIS, but it is not good enough for practical purposes. So I made some adjustments, which are by no means exact, but might give 5 or 6 correct significant digits for relatively low n, when computing the integral:

a(n) ~ sqrt(6/pi)*2^n*(1-6/(5*n)+21/(20*n^2)-1/(8*n^3)+3/n^4)/(n*sqrt(n))

or

a(n) ~ (sqrt(3/p) 2^(n - 5/2) (n (2 n (4 n (5 n - 6) + 21) - 5) + 120))/(5 n^(11/2))

The following HP-42S program computes the integral for n >= 1 and n up to about 20000 (on Free42) and returns exact results for n < 15.

Code:

```
00 { 90-Byte Prgm }

01▸LBL "INCS"
02 ENTER
03 ENTER
04 ENTER
05 5
06 ×
07 6
08 -
```

Gerson.

EMAIL PM FIND QUOTE REPORT

07-15-2018, 08:26 PM **Post: #17**

Thomas Klemm
Senior Member

Posts: 1,449
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

This *RPL* program calculates the coefficients:

Code:

```
«
  « → m
    « { 1 m } 0 CON OBJ→ DROP m →LIST
    »
  » → n ZEROS
  « { 0 1 } 2 n
    FOR k DUP SIZE → a s
      « k ZEROS EVAL a +
        a 1 k SUB 0 + REVLIST s 1 - ZEROS EVAL +
        ADD
        a k 1 + s SUB k 2 * ZEROS EVAL +
```

Example:

The value for **4** is:
`{ 1 0 2 0 2 0 1 0 1 0 1 }`

To get the value for **5** we create the following lists:
`{ 0 0 0 0 0 1 0 2 0 2 0 1 0 1 0 1 }`
`{ 0 2 0 2 0 1 0 0 0 0 0 0 0 0 0 0 }`
`{ 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 }`

And then we just `ADD` them up:
`{ 0 3 0 3 0 3 0 2 0 2 0 1 0 1 0 1 }`

The 2nd and the 3rd list is just the 1st list reversed and then again mirrored at the left border.
That's a consequence of $a_k = a^k + a^{-k}$ being symmetric, that is $a_k = a_{-k}$. We don't want negative indices.

Cheers
Thomas

PS: Is there a better way to create a list of *m* zeros?

PM FIND QUOTE REPORT

07-15-2018, 09:10 PM (This post was last modified: 07-15-2018 09:16 PM by Gerson W. Barbosa.) **Post: #18**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

The RPL program has been rewritten so as not to needlessly waste time expanding a polynomial when results are supposed to be zero.

```
%%HP: T(3)A(R)F(.);
\<< DUPDUP DUPDUP * + 2 / 2 MOD
  IF NOT
  THEN DUP 'X' 1 ROT OVER SWAP
    FOR i OVER i ^ DUP INV + *
    NEXT NIP EXPAND FXND DROP \->STR "*X^" ROT DUPDUP * + 2 / DUP 2 + \->STR "+" + UNROT \-
>STR + "+" + "X^" ROT + PICK3 SWAP POS PICK3 ROT POS 1 - SUB DUP SIZE OVER "+" POS 1 + SWAP SUB OBJ\-> 2 ROT 1 - ^ / \pi *
  ELSE DROP2 0
  END
\>>
```

```
71 -> '3352055187240799 25/73786976294838206464*π'

-> NUM -> 1.42718843886E-2
```

267.5 bytes, but it takes too long for this example (28m 34s... on the emulator!)

As a comparison, the RPN program on Free42 returns `1.4271(9054304)E-2` instantly (maybe a couple of seconds on a real 42S).

EMAIL PM FIND QUOTE REPORT

07-15-2018, 09:35 PM (This post was last modified: 07-15-2018 09:36 PM by Gerson W. Barbosa.) **Post: #19**

Gerson W. Barbosa
Senior Member

Posts: 1,135
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Thomas Klemm Wrote:** ⇒ (07-15-2018 08:26 PM)
> PS: Is there a better way to create a list of *m* zeros?

Better, but probably not the best way, at least on the 49/50g:

« { m } 0 CON AXL »

EMAIL PM FIND QUOTE REPORT

07-15-2018, 09:53 PM **Post: #20**

DavidM
Senior Member

Posts: 722
Joined: Dec 2013

RE: [VA] SRC#001 - Spiky Integral

> **Gerson W. Barbosa Wrote:** (07-15-2018 09:35 PM)
>
> > **Thomas Klemm Wrote:** (07-15-2018 08:26 PM)
> >
> > PS: Is there a better way to create a list of *m* zeros?
>
> Better, but probably not the best way, at least on the 49/50g:
>
> « { m } 0 CON AXL »

Gerson's method above is also how I would do it if the ListExt library isn't available.

Using the ListExt library:

« 0 m LMRPT »

Creating a list of 1000 0s with the CON method: 0.77s
Creating the same list with the ListExt command: 0.07s

Either should give reasonable performance in this scenario.