07-22-2018, 02:33 AM **Post: #41**

Valentin Albillo
Senior Member

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

RE: [VA] SRC#001 - Spiky Integral

.

Hi, Albert:

> **Albert Chan Wrote:** (07-21-2018 06:00 PM)
> From Valentin Albillo last post, my estimate match correct values, to 15 digits. Comparing values in binary form, accuracy is even better, mine is just 6 ULP over !
>
> **Thanks Valentin**. I am amazed at how your integration function work.

You're welcome. Thanks to you for your fresh and effective insight, your contribution to this subject has been truly outstanding and I for one appreciate it. Also, I'm glad if my numeric quadrature results were useful for you to check your remarkably accurate approximations.

> **Quote:**
> You mentioned the problems of it computing *I(20000)*.
> **Shrinking** the integral range 800,000 times may help: (0 to Pi/400000) instead of (0, 2 Pi)

Sure it does. These are the results my quadrature program outputs for the various ranges you mention:

I(20000, 0..Pi/40000)

```
0    0.0000007674489756250120597286778127897367391796009 94
1    0.00000076744898327601408455534276629150023521856930 3
2    0.00000076744898327603472020851084672424898967032756 3
3    0.00000076744898327603472020851855979357618271359961 0
4    0.00000076744898327603472020851855979357618271359852 9  (all 51 decimals digits shown are correct)
```

I(20000, 0..Pi/400000)

```
0    0.00000076744898327603472020851855979357618264054999631
       35064117112150402698116508077139817678384714549561511

1    0.00000076744898327603472020851855979357618264054999631
       35038511379081941548835069200608164307257330429915401

2    0.00000076744898327603472020851855979357618264054999631
       35038511379081941548835069200608164289806491475047963  (all 106 digits shown are correct)
```

However, I concur with you that the fact that these results are accurate to *51* and *106* decimal digits, respectively, for both shrunk ranges does not necessarily mean that after multiplying them by the correct factor the resulting *approximate* values for the *full-range (0..2*Pi)* integral will be accurate to that many digits, in fact it's quite conceivable that they are *not* because both results above match only up to ***42*** digits, namely:

```
0.000000767448983276034720208518559793576182
```

Last but not least, using your remarkable *shrinking* technique, the value on the full integral for **N=20,000** indeed **can** be computed using an HP calc using just its native 12-digit precision (!!), as these *3 lines* of code I wrote for the **HP-71B** convincingly demonstrates:

```
1   DESTROY ALL @ SFLAG -1 @ DIM P,I @ DISP 4*INTEGRAL(0,PI/40000,1E-8,FNF(IVAR))
2   DEF FNF(X) @ P=1 @ FOR I=1 TO 20000 @ P=P*COS(I*X) @ IF NOT P THEN END
3   NEXT I @ FNF=P

>RUN

      3.0697959335 6E-6      (75 min. in Emu71) wich is accurate to 10 digits.
```

Note: That simplest of tricks in the function's definition *("IF NOT P THEN END")* halves the computing time.

Regards.
V.
.

PM FIND EDIT QUOTE REPORT

07-22-2018, 06:50 AM (This post was last modified: 07-22-2018 12:37 PM by Albert Chan.) **Post: #42**

Albert Chan
Senior Member

Posts: 623
Joined: Jul 2018

RE: [VA] SRC#001 - Spiky Integral

> **Valentin Albillo Wrote:** ➡ (07-22-2018 02:33 AM)
> both results above match only up to ***42*** digits, namely:
>
> 0.000000767448983276034720208518559793576182

Hi Valentin, Thanks for the check.

The fact that two integral only match 42 digits does not mean anything.
The spike formula asked for **full spike area**.

I chopped the spike only for speedier computation, trading accuracy for time.

Plotting N vs spike formula digits accuracy, and it followed a straight line,
already reaching 15 digits accuracy for N = 60

You might like to try the spike formula again, for N = 1000.
For *I(1000)*, you have a correct value to match against ...

EMAIL PM FIND QUOTE REPORT

07-22-2018, 04:01 PM **Post: #43**

Albert Chan
Senior Member

Posts: 623
Joined: Jul 2018

RE: [VA] SRC#001 - Spiky Integral

Using Python + numpy, I managed to calculate *I(1000)*

Code:

```
import numpy

def spike(n):
    terms = 1 + n*(n+1) // 2    # terms count of Product[z^i + 1/z^i, {i, n}]
    if terms % 2 == 0: return 0 # no constant term
    b = numpy.array([0] * (terms//2 + 1), dtype=object)
    b[-2:] = 1                  # 1 + z
    for i in range(2, n+1):     # multiply by 1 + z^i
        b[:-i] += b[i:]         # build half terms
    return b[0]                 # constant term
```

>>> spike.spike(1000)
467708586991053780130476928496471502490480200263913521598374850751872554493810444457551220080062611698129340084492921358002542871554381676735263500063986630523649271451391806356678763700553308953903873511563867155215734010997332808966175771715884247295296277348179194597363883854664431808932677416L

Above calculation take **49 seconds** on my laptop

I(1000) = b[0] / 2^1000 * (2 Pi) =

0.000274258153608378926807734432669808007979394749673091726358234027755841714506724234556964545380120825381783157659756758893238403974033229771909645027440110048173961155202604290335688141770901611096863576444159483197361960300243717555848542910006760212326308258399120935101281203608176357009606892564575924775067719 ...

Rounded to 45 digits, it match Valentin value **exactly**

Is it possible to explain how the integration code work ?
How does it handle so many spikes ?

EMAIL PM FIND QUOTE REPORT

07-22-2018, 11:06 PM (This post was last modified: 07-22-2018 11:07 PM by Valentin Albillo.) **Post: #44**

Valentin Albillo
Senior Member

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

RE: [VA] SRC#001 - Spiky Integral

,
Hi, **Albert**:

> **Albert Chan Wrote:**
> Hi Valentin, Thanks for the check.

You're welcome.

> **Quote:**
> The fact that two integral only match 42 digits does not mean anything. The spike formula asked for full spike area. [...]You might like to try the spike formula again, for N = 1000. For I(1000), you have a correct value to match against ...

I'll do it soon, when I get home in a few hours.

> **Quote:**
> Rounded to 45 digits, it match Valentin value **exactly**

Of course. That's why I stated *"(all 48 decimal digits shown are correct)"* ... :-) ... (the 48 decimal digits mentioned include the three initial zeros as well).

> **Quote:**
> Is it possible to explain how the integration code work ? How does it handle so many spikes ?

I developed the code myself as a program for the **HP-71B**, good for 12 digits and easily surpassing the speed of the assembly-language *Math ROM*'s *INTEGRAL* keyword, then ported it nearly *verbatim* to a multiprecision environment, which is the version I've used to produce the results I've posted in this thread.

The code was intended to be included in one of my PDF *Datafile* articles some 10 years ago, as explaining it and providing relevant examples would take at least some 10-12 pages. Alas, for reasons which I won't repeat here the article was never submitted for publication and

remains unpublished to this day.

If (and when) I can find some adequate place where to publish it, I'll post an announcement here for interested people (like yourself) to download it for free.

Thanks for your interest and regards.
V.
.

PM FIND EDIT QUOTE REPORT

07-24-2018, 02:13 AM (This post was last modified: 03-27-2019 08:02 PM by Albert Chan.) **Post: #45**

Albert Chan
Senior Member

Posts: 623
Joined: Jul 2018

RE: [VA] SRC#001 - Spiky Integral

Using Gaussian Quadrature for **spike formula = 4*I(1000, 0 to Pi/2000)**, against exact **I(1000)**:

- 200 digits accuracy
- 204 matched digits after decimal point

EMAIL PM FIND QUOTE REPORT

« Next Oldest | Next Newest »

Enter Keywords Search Thread

Pages (3): « Previous 1 2 **3**

NEW REPLY

View a Printable Version
Send this Thread to a Friend
Subscribe to this thread

User(s) browsing this thread: Valentin Albillo*