1st April, 2022, 21:25

Post: #1

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

[VA] SRC #011 - April 1st, 2022 Bizarro Special

Hi, all,

Well, my ***HP*** calc site reached the ***100,000*** *downloads* mark a few days ago and on top of that today it's ***April, 1st***, so let's celebrate the occasion ... *(drum roll)* ...

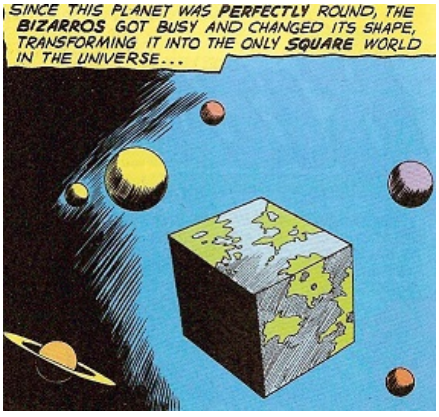
Welcome to my *SRC #11 - April 1st, 2022 Bizarro Special*

intended to once again put your brains and your ***HP*** *calculators* to work, this time featuring a ***bizarre*** *challenge* which nevertheless has important, useful *real-life* applications. But first, some exposition *(**Note**: What follows are mostly my own ramblings © me, **DC Comics** are not to blame !)*:

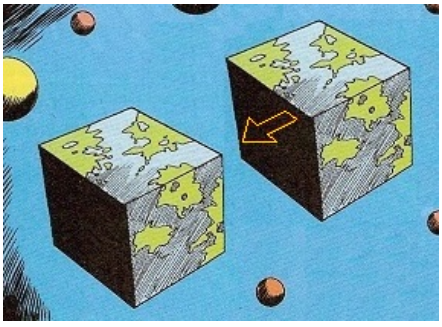
The seasoned veterans among you might fondly remember the classic ***Superman*** comics of the *'60s*, and that within the *Superman* universe there was the ***Bizarro World*** (also known as ***Htrae***), a fictional planet home to the eponymous people, whose society is ruled by the ***Bizarro Code***, which states:

> ***"Us do opposite of all Earthly things ! Us hate beauty! Us love ugliness ! Is big crime to make anything perfect on Bizarro World !"***

Originally a normal planet, the *Bizarro World* is now ***cube-shaped*** because the *Bizarros* couldn't stand the *perfection* of its spherical shape and *Bizarroformed* it like this:

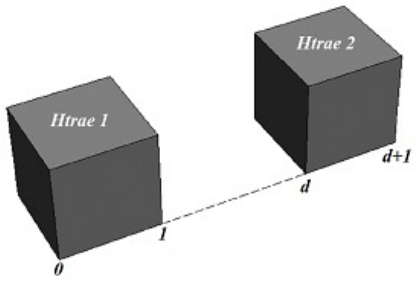


In time, the *Bizarro* society thrived to the point where their cubic-shaped planet was quickly becoming overpopulated, so they set upon themselves the huge task of attaching face to face a *second* almost-identical planet to their original one, thus nearly duplicating the habitable surface and best of all, *augmenting* the imperfection of their homeworld's shape relative to perfect *sphericity* by making it *prismatic*, like this:



However, they first needed to make sure that the project was feasible, in particular that the *gravitational force* ***F*** between the planets when they were in *contact* would be manageable. Therefore, they commissioned *Lebon Prize* laureate scientist ***Rd. Nitnelav Albizarro #1*** to carry out the computation, who immediately set to the task of finding out ***F*** when their respective centers were initially separated by an arbitrary distance ***d***, which would then be shortened until the planets were in contact face to face.

To simplify matters and as the results could be easily rescaled afterwards, it was assumed that the *gravitational constant* ***G*** was ***1*** (in some units) and the planets were homogeneous cubes of side ***1*** (ditto) and mass ***1*** (ditto), initially placed like this:



Now, ***Rd. Albizarro*** considered a pair of sample points, (x_1, y_1, z_1) in ***Htrae 1*** and (x_2, y_2, z_2) in ***Htrae 2***, knowing that their contribution to the overall force would be:

$$\frac{1}{r^2} = \frac{1}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

and duly taking into account the symmetry (which nullifies the force exerted in the ***y*** and ***z*** directions) and integrating over all possible values for the respective point coordinates, quickly got this *sextuple integral* for the value of the force ***F*** between the planets:

$$F = \int_0^1 \int_0^1 \int_d^{d+1} \int_0^1 \int_0^1 \int_0^1 \frac{x_2 - x_1}{\left[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2\right]^{3/2}} \cdot dx_1\, dy_1\, dz_1\, dx_2\, dy_2\, dz_2$$

to be numerically evaluated for the particular case when both planets are in contact, i.e. the distance ***d*** between their centers is ***1***.

As it happened, ***Rd. Albizarro*** was a *Code*-abiding citizen of the *Bizarro* society and wouldn't go for an *unnecessarily accurate* (ugh!) result but would instead be satisfied with obtaining about ***three*** *correct digits*. To that effect, and aware of the need to use some computing device in order to meet the assigned deadline, ***Rd. Albizarro*** promptly proceeded to check out the ones available in ugly human *Earth* at the time (ca. *1982*) and saw that computers were so *perfectly* fit for the job that using one would be *disgusting* and could result in being severely reprimanded or worse, so the alternative was to get instead a *programmable calculator* (*progcalc* for short) as non-programmable ones certainly wouldn't do.

Restricting thus the search to *progcalcs*, it was soon apparent that there were essentially two main contenders, the ones branded ***TI*** (probably standing for *"**T**otally **I**deal"*, ugh !), and the ones branded ***HP*** (possibly standing for *"**H**ardly **P**erfect"*), the latter being just what was needed !

Now there was the question of selecting *which* particular ***HP*** model to use but that was easy-peasy, just a matter of looking at the specs and chosing the *least capable* one, which happened to be the **HP-10C**, the proverbial *runt of the litter*, a severely limited model having only ***79*** bytes of *RAM* available for storing programs *and* data, *no* subroutines, *no* flags, *no* indirection, *no* loop intructions, just *two* conditional tests, a meager function set, *no* I/O, and *very* slow to boot ... indeed, the *least perfect* model for the task at hand.

But ***Rd. Albizarro*** was unfazed, thinking that *"If ugly Earth's Newton could do his gravitational calculations using this thing, it'll do for me as well"*, and against all odds actually ***succeeded*** by quickly writing a clumsy ***RPN*** *(**R**eally **P**erfect **N**ot)* program for the **HP-10C**, keying in the pertinent inputs, pressing the `[R/S]` *(**R**un **S**lowly)* key, going out to have a quick dinner and lo and behold, upon returning the computed value of ***F*** was already waiting in the display, which indeed was correct to ***three*** *decimal places*, as required:

Well, once the story's been told, **it's your move**:

> Try to emulate ***Rd. Albizarro***'s achievement and ***write a program*** for the **HP-10C** which computes in a reasonable time a numerical value for the above sextuple integral correct to at least ***~ three*** *decimal places*.

That failing (shame on you !), see if you can do it using other more perfect (ugh!) models, preferably ***HP*** and preferably ***vintage***.

> **Note:** ***No cheating*** whatsoever allowed. Also, doing symbolic manipulations (transformations, dimensionality reduction, changes of variables), either ***by hand*** or using any ***CAS*** and/or giving **math lectures** is a ***big crime***, you'll get arrested or worse, so stick to purely numerical computations. This is my last ***SRC*** for a looong while so don't spoil it for me, Ok ? Thanks.

I'll post my *Original Solution* for the **HP-10C** with results and extensive comments within a few days ... ***Or not***, after all this might be an elaborate *April Fools' Day* practical **joke** !!

V.

PM WWW FIND EDIT QUOTE REPORT

1st April, 2022, 23:46 **Post: #2**

Massimo Gnerucci
Senior Member

Posts: 2,587
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Just to say that I love this one: Bizarro... cubic worlds... all the silliness of my youth's Superman!

PM WWW FIND QUOTE REPORT

2nd April, 2022, 05:01 **Post: #3**

Ren
Member

Posts: 180
Joined: Mar 2016

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

I remember some of the Bizarro storyline, but I couldn't afford to buy comic books back then, so could only read what came my way by those who loaned comics to my older brothers.

I think a parallel can be seen in Star Trek: Lower Decks interactions with the Paklid.

PM FIND QUOTE REPORT

2nd April, 2022, 11:32 (This post was last modified: 2nd April, 2022 12:33 by J-F Garnier.) **Post: #4**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

A sextuple integral, that even a HP-71B Math ROM couldn't handle, and on a HP-10C, no less. That's a challenge!

As a very crude and *ugly* approximation, we can put all the mass at the center of each cube, and calculate the force $F = Mm/D^2$
M=m=D=1 so F=1.

To improve this first approximation, we can then divide each cube in 8 smaller cubes of side 1/2, and calculate the force between each pair of small cubes (putting again the mass of each at their center), and sum them up.
Then we can divide the cubes further in 27, 64, etc parts and hope that this all converges to some value.

This would be with no problem, in principle, with enough computing resources, but here we have only a 10C (or another machine but using only 10C features and resources) so this seems impractical.

Another approach would be to use a statistical method by taking random points in each cube and calculating their contribution to the force.
With enough randomly distributed points, we can have a good approximation of the force, and the implementation may be easier.
We need to take 6 random values for the x1, x2, y1, y2, z1, z2 values, calculate the integrand and sum it.
... *and here we realize that the 10C has no random function.*
We could use the well-known user-code random generator, but *we also realize that the 10C has no subroutine*, duplicating the random generator code 6 times is just not possible.

However, we don't need a perfect random generator,
So it appears that the challenge is to find a poor-man random generator, compact enough to fit 6 times in the 10C program memory, yet not too bad.
Any tentative?

J-F

EMAIL PM WWW FIND QUOTE REPORT

2nd April, 2022, 22:31 **Post: #5**

ijabbott
Senior Member

Posts: 1,242
Joined: Jul 2015

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (2nd April, 2022 11:32)
> A sextuple integral, that even a HP-71B Math ROM couldn't handle, and on a HP-10C, no less. That's a challenge!
>
> As a very crude and *ugly* approximation, we can put all the mass at the center of each cube, and calculate the force F = Mm/D²
> M=m=D=1 so F=1.

TBH, I don't get why that is not the correct answer!

EMAIL PM FIND QUOTE REPORT

3rd April, 2022, 03:25 **Post: #6**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **ijabbott Wrote:** (2nd April, 2022 22:31)
>
> > **J-F Garnier Wrote:** (2nd April, 2022 11:32)
> > As a very crude and *ugly* **approximation**, we can put all the mass at the center of each cube, and calculate the force F = Mm/D², M=m=D=1 so **F=1**.
>
> **TBH, I don't get why that is not the correct answer!**

Because a cube doesn't have *spherical* symmetry. cf. Wikipedia:

> *"[...] an object with a **spherically symmetric distribution of mass** exerts the same gravitational attraction on external bodies as if all the object's mass were concentrated at a point at its center. (This is **not** generally true for **non-spherically-symmetrical** bodies.)"*

V.

PM WWW FIND EDIT QUOTE REPORT

3rd April, 2022, 08:21 **Post: #7**

rawi
Member

Posts: 138
Joined: Nov 2019

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Garnier wrote:

> **Quote:**
> However, we don't need a perfect random generator,
> So it appears that the challenge is to find a poor-man random generator, compact enough to fit 6 times in the 10C program memory, yet not too bad.
> Any tentative?

I do not think that you need the random number generator 6 times. Why not generate one random number, store it in a register and use it six times?

EMAIL PM FIND QUOTE REPORT

3rd April, 2022, 17:38 **Post: #8**

vaklaff
Member

Posts: 118
Joined: Dec 2019

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **rawi Wrote:** (3rd April, 2022 08:21)

I do not think that you need the random number generator 6 times. Why not generate one random number, store it in a register and use it six times?

Or just use 17 which is the best random anyway :-)

EMAIL PM FIND QUOTE REPORT

3rd April, 2022, 17:41 (This post was last modified: 3rd April, 2022 23:30 by ijabbott.) **Post: #9**

ijabbott
Senior Member

Posts: 1,242
Joined: Jul 2015

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Valentin Albillo Wrote:** (3rd April, 2022 03:25)
>
> > **ijabbott Wrote:** (2nd April, 2022 22:31)
> >
> > **TBH, I don't get why that is not the correct answer!**
>
> Because a cube doesn't have *spherical* symmetry. cf. Wikipedia:
>
> > *"[...] an object with a **spherically symmetric distribution of mass** exerts the same gravitational attraction on external bodies as if all the object's mass were concentrated at a point at its center. (This is **not** generally true for **non-spherically-symmetrical** bodies.)"*
>
> **V.**

Indeed, splitting each cube into 8 cubelets and summing the gravitation forces between each pair of cubelets (one from Htrae 1 and one from Htrae 2) using the point mass assumption produces an answer that differs from 1 (actually about 1.09217), which means that cubes of uniform density cannot be replaced with point masses.

EDIT: As pointed out to me by Albert Chan, I forgot to resolve the vectors, so my sum is wrong. It should be less than 1.

EMAIL PM FIND QUOTE REPORT

3rd April, 2022, 18:35 **Post: #10**

rprosperi
Super Moderator

Posts: 5,742
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Though I don't have the time, and probably am not up to the analysis involved here, I wanted to add some comments about this entertaining and thought-provoking post:

- Thanks to Valentin for the huge effort and true creativity to provide this great post!
- This is a typical post from Valentin in the sense that, even if I am not able to contribute a possible solution, I am entertained but I also learn a lot, simply by reading the ensuing discussions, and I want to thank Valentin for that. I hope this is *not* the last SRC for some time...
- I think Bizarro was found in Heavy Metal, wasn't it? A true treasure for many years, it inspired so many other authors
- Has anyone else noted the important word in low-contrast, barely-readable white font near the bottom...

EMAIL PM FIND QUOTE REPORT

3rd April, 2022, 21:16 **Post: #11**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Below is my attempt to implement the sextuple integral computation on the 10C.

Unfortunately, I didn't succeed to fit into the 10C memory, so this is a 41C version but using only the instruction set of the 10C (except the LBLs needed by the 41C).
The programs expects the number of random samples in X as an input.

I used a simple random number generator of mine (unless I knew it from past readings, can't say). The code is documented and easy to understand.

Now, the results. I run it with different numbers of random samples, between 20 and 999:
#samples integral
20 0.628
50 0.823
100 0.898
200 0.910
500 0.981

999 0.967

So I hardy get 1 correct digit if I round all results in FIX 0.
I'm even not sure if the integral is smaller or larger than 1.
Is it really a surprise? Even with 999 samples, this means an average of about 3 samples per dimension of each cube.

But probably *Rd. Albizarro* preferred a wrong result with 3 figures rather than one reliable estimation that would be *big crime* !

At the end, did it make any difference for *Bizarro World* ?

J-F

```
01*LBL "SRC11"
STO 02 STO 01 ; #samples
PI STO 00 ; rnd seed
0 STO 03 ; init sum

08*LBL 00
RCL 00 PI * FRC STO 00 ; z2
RCL 00 PI * FRC STO 00 ; z1
- X^2 ; (z2-z1)²
RCL 00 PI * FRC STO 00 ; y2
RCL 00 PI * FRC STO 00 ; y1
- X^2 + ; (y2-y1)²+(z2-z1)²
RCL 00 PI * FRC STO 00
1 + ; x2
RCL 00 PI * FRC STO 00 ; x1
- X^2 ; (x2-x1)²
+ ; d²
LASTX SQRT X<>Y ; (x2-x1) d²
ENTER^ SQRT * ; d^3
/ ; (x2-x1)/d^3
ST+ 03 ; add to sum
1 ST- 02 ; decr counter
RCL 02 X<>Y X<=Y? GTO 00 ; loop

63*LBL 01
RCL 03 RCL 01 / ; result
END
```

EMAIL PM WWW FIND QUOTE REPORT

5th April, 2022, 00:34 **Post: #12**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

.
Hi, all,

Thanks for your interest in my ***SRC#11*** and most definitely for your posts. I'll give here some comments. Let's begin ...

> **Massimo Gnerucci Wrote:**
> Just to say that **I love this one: Bizarro... cubic worlds...** all the silliness of my youth's Superman!

I'm very glad for your appreciation of the *Bizarro*-themed story and thanks for telling me, I've always appreciated the concept myself so this was a nice opportunity to include it in this bookend ***SRC***.

> **Ren Wrote:**
> I think a parallel can be seen in **Star Trek: Lower Decks** interactions with the **Paklid**.

Thanks for the tip. I know nothing about the plethora of new *ST* series, I only watched the *Original Series*, *Deep Space 9*, *Voyager* and *Enterprise* (as well as all the movies), so no idea what *Lower Decks* is about or the aliens (I suppose they're aliens) you mention, sorry.

> **rawi Wrote:**
> > **J-F Garnier Wrote:**
> > *We need to take 6 random values for the x1, x2, y1, y2, z1, z2 values,* ***calculate the integrand*** *and sum it.*

I do not think that you need the random number generator 6 times. **Why not generate one random number, store it in a register and use it six times?**

> Because if you do as **J-F** says, i.e. *"calculate the integrand*", you'll find that if all the variables use the *same* random number you'll get a nice but useless ***0 divided by 0*** value for the integrand and there's not much you or anyone else can do with that.

rprosperi Wrote:

Thanks to Valentin for the huge effort and true creativity to provide this great post!

> **Thank you very much** for your continued appreciation of my productions, **Bob**, you're always too kind but the thing I appreciate *the most* is that you ***do*** take the trouble and time to post it and *let me know*. Thanks again.

J-F Garnier Wrote:

Now, the results. I run it with different numbers of random samples, between 20 and **999** [...]

> I'm curious ... why *999* instead of *1,000* ? I've inspected your code and see no reason for stopping short of the more natural *1,000.* You're not using `ISG` or something like that.

J-F Garnier Wrote:

But probably Rd. Albizarro preferred a wrong result with 3 figures rather than one reliable estimation that would be big crime !

> Nope, ***Rd. Albizarro*** sticks to *Bizarro* society *Code* but ***Mathematics*** is neither *perfect* nor *imperfect*, it just ***is***.

J-F Garnier Wrote:

At the end, did it make any difference for Bizarro World ?

> Of course it did, they got 10 habitable faces where previously they only had six

I'll post the rest of ***Rd. Albizarro***'s story (which includes my *Original Solution* and extensive comments) next ***Saturday 9*** so you've got ***5*** *additional days* to try and come up with your own *Solution* ... or at least die trying !

Best regards.
V.

PM WWW FIND EDIT QUOTE REPORT

5th April, 2022, 09:23 (This post was last modified: 5th April, 2022 09:37 by J-F Garnier.) **Post: #13**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Valentin Albillo Wrote: (5th April, 2022 00:34)

> **rawi Wrote:**
> I do not think that you need the random number generator 6 times. **Why not generate one random number, store it in a register and use it six times?**
>
> > Because if you do as **J-F** says, i.e. *"calculate the integrand*", you'll find that if all the variables use the *same* random number you'll get a nice but useless ***0 divided by 0*** value for the integrand and there's not much you or anyone else can do with that.

No, it's not as severe. If you use the same random value, and since `x2=1+rnd` you get exactly 1 for the integrand, and 1 as the result !

Quote:

> **J-F Garnier Wrote:**
> Now, the results. I run it with different numbers of random samples, between 20 and **999** [...]
>
> > I'm curious ... why *999* instead of *1,000* ?

I was expecting someone would ask... and you did!
It's just that, as you may know, I'm not very good in post formatting, so I used 999 instead of 1000 to keep the figures well aligned :-)

J-F

EMAIL PM WWW FIND QUOTE REPORT

5th April, 2022, 19:20 **Post: #14**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (5th April, 2022 09:23)
>
> > **Valentin Albillo Wrote:** (5th April, 2022 00:34)
> >
> > Because if you do as **J-F** says, i.e. *"calculate the integrand*", you'll find that if all the variables use the *same* random number you'll get a nice but useless ***0 divided by 0*** value for the integrand and there's not much you or anyone else can do with that.
>
> No, it's not as severe. If you use the same random value, and **since** `x2=1+rnd` **you get exactly 1 for the integrand, and 1 as the result** !

Sorry but it doesn't fly. That ***x2=1+rnd*** doesn't appear *explicitly* in the integrand expression as such, only ***x2-x1*** in both numerator and denominator, and further it only appears when using your stochastic method, not if using other possible methods.

Thus, I don't think that ***rawi***, when trying to compute the value of the *integrand* as it appears in my OP, would think or know about the need to add *1* to *x2*, which is only *implicit* in the *[d, d+1]* limits of integration and only when substituting *d=1*, thus he'd surely get *0 divided by 0* as I said he would.

> **J-F Garnier Wrote:**
>
> > **Valentin Albillo Wrote:**
> >
> > I'm curious ... why *999* instead of *1,000* ?
>
> I was expecting someone would ask... and you did!
> It's just that, as you may know, I'm not very good in post formatting, so **I used 999 instead of 1000 to keep the figures well aligned** :-)

It doesn't fly either, you could have used **1E3**, which aligns properly with 100, 200 and 500, and furthermore you include **20** and **50** which don't align with them either and you didn't seem to mind.

Best regards.
V.

PM WWW FIND EDIT QUOTE REPORT

9th April, 2022, 23:41 **Post: #15**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Hi, all,

Thanks for your interest in my ***SRC #11*** and your valuable posts, much appreciated. Now let's conclude ***Rd. Albizarro***'s story, featuring my *Original Solution*, results and comments:

Last time on Bizarro World:

> ***Rd. Albizarro*** came back from a quick dinner and found the computed value of the gravitational force ***F*** already waiting in the **HP-10C**'s display, which was indeed correct to the required *three* decimal digits. How was the feat accomplished ?

And now, the conclusion:

Rd. Albizarro was aware that the **79** bytes of *RAM* available in the **HP-10C** wouldn't allow for any *deterministic* cubature methods, as even just storing each of the six integration variables in their own register would require *42* bytes, leaving only *37* program steps for the method's implementation and no memory at all for anything else. Also, most deterministic cubature methods suffer from the so-called *curse of dimensionality*, requiring a number of sample points (and thus running time)

exponentially growing with the dimension, to achieve even modest accuracy.

What cubature method would do, then ? Only a *non-deterministic* one would do, such as an ***OEM*** method (*OLRAC ET NOM*), which are free from this *"curse"* (though *slowly* converging) but still quite challenging to implement on the **HP-10C** because evaluating the integrand would require generating *six* uniform *(pseudo-)*random numbers per evaluation and the **HP-10C**'s instruction set *doesn't* include a random number generator *(RNG)*.

Thus, some suitable *RNG* would have to be implemented in *RPN* user code as a subroutine to be called 6 times per integrand's evaluation, a brilliant plan except for the fact that the **HP-10C** *doesn't have subroutine capability* either, which meant that the *RNG* code would have to appear in-lined six times.

To know how many program steps would be available for this, ***Rd. Albizarro*** first made a quick estimation of the number of storage registers needed: one for storing the *seed*; another to count down the number of times the program would loop through to compute the integral; another to store the number of *samples*, needed for obtaining the final value; and a fourth to keep the running *summation* of the integrand's evaluations, so 4 storage registers in all which would require *28* bytes.

That left *51* steps for the program, with the integrand's evaluation requiring at least *22* of them (assuming six *1*-step *(nonexistent!)* **RAN#** instructions) and leaving only *29* available for everything else. However, implementing the nonexistent *1-step* **RAN#** instruction in *RPN* meant that the *ersatz RNG* code would had to be *(29+6)/6 ~ 5.8*, i.e. ***5*** *steps long at most*.

Could a decent *RNG* be implemented in only *5 program steps* or less ? ***Yes !***. ***Rd. Albizarro*** had recently read about some *RNG* for ***HP*** *RPN* models which had been advocated by one ugly-*Earth* native ***Mr. Albillo*** because of its extreme simplicity and fairly reasonable behavior, who then went to publish it in some magazine called *"PPC Calculator Journal"*. This *RNG* could be implemented in as little as ***4*** program steps, namely:

```
RCL (seed), R-D, FRAC, STO (seed)
```

where **R-D** was a *radians-to-degrees* conversion, which most fortunately *(thanks, **Yhprum** !) did* exist in the **HP-10C** under the name **->DEG**, so it could be used to generate the six random numbers, requiring *24* program steps in all. However, that would still leave only *11* steps for everything else, which included some initialization, updating the ongoing summation, checking for loop termination, computing the final average, ...

Fortunately, after a modicum of further reflection, ***Rd. Albizarro*** discovered that *each pair* of random numbers could be generated using just ***7*** steps instead of ***8***, resulting in *14 steps* being available for the remaining operations, which was more than enough so the **HP-10C** program was indeed feasible and could be implemented in just *49* program steps, like this:

```
01 STO 1     11 STO 3     21 ENTER     31 ENTER     41 /
02 STO 2     12 -         22 ->DEG     32 ->DEG     42 STO+ 0
03 CLX       13 1         23 FRAC      33 FRAC      43 RCL 1
04 STO 0     14 STO- 1    24 STO 3     34 STO 3     44 X=0?
05 RCL 3     15 +         25 -         35 -         45 GTO 47
06 ->DEG     16 ENTER     26 X^2       36 X^2       46 GTO 05
07 FRAC      17 X^2       27 +         37 +         47 RCL 0
08 ENTER     18 RCL 3     28 RCL 3     38 ENTER     48 RCL 2
09 ->DEG     19 ->DEG     29 ->DEG     39 SQRT      49 /
10 FRAC      20 FRAC      30 FRAC      40 x            (GTO 00, default end of program)
```

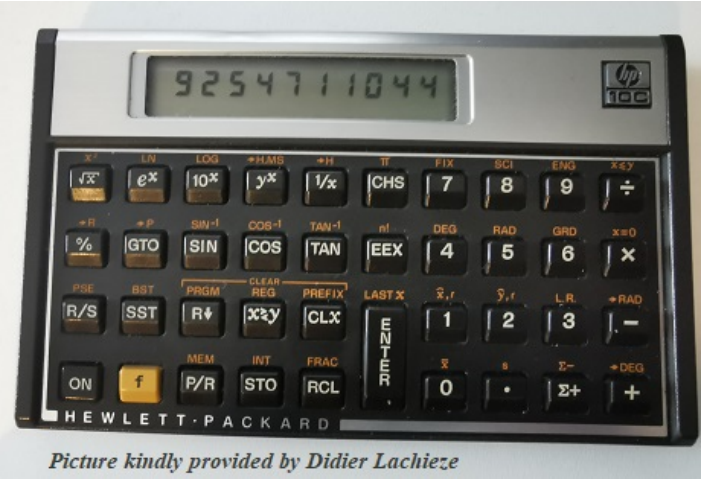
Before running it, the initial *seed* had to be stored in **R3** and the number of *samples* (pairs of points to generate and use in the integrand's evaluation) had to be specified, which ***Rd. Albizarro*** heuristically estimated as follows:

> A good way to test a program is to run it against problems whose solutions are *known*. In this case, the gravitational force ***F*** between two contacting *spheres* (instead of *cubes*) is:
>
> $G = 1,\ m_1 = 1,\ m_2 = 1,\ d = 1,\ \mathbf{F} = G*m_1*m_2/d^2 = \mathbf{1.000}$,
>
> so a version of this program particularized for *spheres* instead of *cubes* was executed, watching for the number of samples needed to achieve 3-digit accuracy, which was ***687*** *samples*, returning ***F ~ 0.999***.
>
> However, the volume of a *sphere ("ball" would be more correct but whatever)* of diameter ***1*** is $4/3*Pi*(1/2)^3 = Pi/6 =$ ***0.524***+, significantly smaller than the volume of a *cube* of side ***1***, which is exactly ***1***, thus to maintain the same *sample density* (in order to achieve a comparable accuracy) the number of samples must be multiplied times the volumes' ratio, *6/Pi*, so the estimated number of samples for the *cubes* case would be *687*6/Pi ~* ***1,312*** *samples*.

So, ***Rd. Albizarro*** stored some suitable *seed* (say, ***1***) in register **R3**, entered the number of samples to use (***1312***) in the display, set **[FIX 3]** and executed the program, like this:

```
1 [STO 3] 1312 [R/S] -> 0.925  (0.925 4711044)
```

which returned the sought-after gravitational force ***F = 0.925*** after ~ *72'*, nicely spent having a tasty quick dinner:



Picture kindly provided by Didier Lachieze

The theoretically correct 3-digit result is ***0.926*** *(**0.925** 9812605, to 10 correct digits)* so indeed *3 correct digits* (save *1 ulp*) were obtained, as required.

Having produced the desired result and met the strict deadline, the mission was fully accomplished but after some months had elapsed a much more relaxed ***Rd. Albizarro*** leisurely pondered whether the *sextuple* integral could perhaps be tackled *symbolically*, and after a few days finally succeeded in reducing it to a *triple* integral first, then to a *double* integral, and finally to a *single*-dimensional definite integral, which once evaluated resulted in this nice, ***exact*** symbolic value:

$$F = \frac{1}{3}\left(\frac{26\pi}{3} - 14 + 2\sqrt{2} - 4\sqrt{3} + 10\sqrt{5} - 2\sqrt{6} + 26\log(2) - \log(25) + 10\log(1+\sqrt{2}) \right.$$
$$\left. + 20\log(1+\sqrt{3}) - 35\log(1+\sqrt{5}) + 6\log(1+\sqrt{6}) - 2\log(4+\sqrt{6}) - 22\tan^{-1}(2\sqrt{6})\right)$$

and after some rearranging, this longish expression would also *exactly* fit as a ***79**-step* **HP-10C** program with *no* inputs required and *no* registers used (though alas, none were available as the program uses up *all 79* bytes of *RAM*), like this:

01	**PI**	*17*	**x**	*33*	√	*49*	**0**	*65*	**LN**
	3		**-**		**1**		**x**		**6**
	/		**6**		**+**		**+**		**x**
	2		√		**LN**		**5**		**+**
	LN		**-**		**2**		√		**6**
	+		**LASTX**		**x**		**1**		√
	2		**4**		**2**		**+**		**2**
	6		**+**		√		**LN**		**x**
	x		**5**		**1**		**3**		**TAN^{-1}**
	2		**x**		**+**		**5**		**2**
	√		**LN**		**LN**		**x**		**2**
	7		**-**		**+**		**-**		**x**
	-		**2**		**5**		**6**		**-**
	3		**x**		√		√		**3**
	√		**+**		**+**		**1**	*79*	**/**
	2		**3**		**1**		**+**		

Assuming **RAD**ians mode: **[R/S]** -> ***0.9259812667*** *(exact: **0.9259812605**...)* in a few seconds, correct to ***8** decimal digits*, the last two being lost to rounding errors throughout.

Nothing else to do here, so as ***Superman***'s imperfect duplicate ***Bizarro #1*** would say ...

Well, this concludes ***Rd. Albizarro***'s story, and I still have a number of hopefully interesting ***Comments*** ready to post as an *Epilogue* of sorts, including some *real-life* applications, but first let's hear from you.

Regrettably, nobody posted a working **HP-10C** program and/or the sextuple integral's value correct to three digits, as required (although ***J-F Garnier*** came pretty close on both counts, many thanks for your efforts and great posts, ***J-F***), but if any of you

can produce working code for *any other* ***HP*** *model* (say, the **HP-71B** + Math *ROM* or some *RPL* model), with or without using *symbolic* manipulations (say, *dimensionality reduction*) and/or have some comments of your own, now that a correct solution has been provided and thus there's no spoiling the challenge for anyone, ***this*** is the time to post them and yes, *I'm looking at* ***you*** *!*

Regards.
V.

PM WWW FIND EDIT QUOTE REPORT

10th April, 2022, 01:53 **Post: #16**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

It would be nice to have back of envelope calculation for F

We can keep planet intact, and split the other planet, to 2x2x2 = 8 cubes.
With symmetry, only 1 corner to consider

(3/4, 1/4, 1/4) → D1^2 = 11/16, cos(θ1) = 3/√11
(5/4, 1/4, 1/4) → D2^2 = 27/16, cos(θ2) = 5/√27

Force is proportional to 1/D^2
Force is an vector, we need cos(θ) correction, to sum only force between planets.

F = ((16/11)*(3/√11) + (16/27)*(5/√27)) / 2 ≈ 0.94295

Since 1 planet still intact, we would expect F to be even less, i.e. F < 0.94295

EMAIL PM FIND QUOTE REPORT

10th April, 2022, 02:04 **Post: #17**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Difference of two uniform distribution form triangular distribution.
This reduced sextuple to triple integral.

We just integrate ∫∫∫(f(x,y,z) dz dy dx), with all variables from 0 to 1

Let x = |x2-x1|, y = |y2-y1|, z = |z2-z1|, all with distribution ◣
Squaring is going to remove abs() anyway.

For x, we do need ◢◣ distribution, center at d, thus (d+x), (d-x) terms.
Because of the discontinuity at x=0, we fold the 2 integrals as 1.
This moved the discontinuity to the edge.

Here is **HP71B** equivlent code (smaller P actually make things worse)

```
10 P=.01 @ D=1 @ T=TIME
20 DEF FNZ(X,Y)=INTEGRAL(0,1,P,X/(X*X+Y*Y+(1-SQR(IVAR))^2)^1.5)
30 DEF FNY(X)=INTEGRAL(0,1,P,FNZ(X,1-SQR(IVAR)))
40 DEF FNX(X)=FNY(D+X)+FNY(D-X)
50 DISP INTEGRAL(0,1,P,FNX(1-SQR(IVAR)))/2, IBOUND, TIME-T

>RUN
 .92596918938     1.84480545818E-2     21.66
```

EMAIL PM FIND QUOTE REPORT

10th April, 2022, 02:11 **Post: #18**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

This version improve accuracy by removing square roots. Let x=u^2, v=1-u

∫(f(1-√x), x=0..1) = ∫(f(1-u)*(2u), u=0..1) = 2*∫(f(v)*(1-v), v=0..1)

Transformed integral (with discontinuity at x=0, evaluate in 2 pieces)

$$I = 4\int_{-1}^{1}\int_{0}^{1}\int_{0}^{1}\frac{(d+x)\,(1-|x|)\,(1-y)\,(1-z)}{[(d+x)^2+y^2+z^2]^{3/2}}\,dz\,dy\,dx$$

```
10 P=.0001 @ D=1 @ T=TIME
20 DEF FNZ(X,Y)=INTEGRAL(0,1,P,X/(X*X+Y*Y+IVAR*IVAR)^1.5*(1-IVAR))
30 DEF FNY(X)=INTEGRAL(0,1,P,FNZ(X,IVAR)*(1-IVAR))
40 I1=INTEGRAL(0,1,P,FNY(D+IVAR)*(1-IVAR))*4 @ DISP I1,IBOUND*4,TIME-T
50 I2=INTEGRAL(0,1,P,FNY(D-IVAR)*(1-IVAR))*4 @ DISP I2,IBOUND*4,TIME-T
60 DISP I1+I2

>RUN
 .236198653356     2.35758565639E-5     128.35
 .689787720788     6.90121464808E-5     763.87
 .925986374144
```

1st integral is sum of force for mass separated by D or more.
2nd integral is sum of force for all mass separated by D or less.

2nd integral (for D=1) contributed about 3/4 of full force.
Its singularity (where planets are touching) make it harder to calculate.

EMAIL PM FIND QUOTE REPORT

10th April, 2022, 02:37 **Post: #19**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Valentin Albillo Wrote:** (9th April, 2022 23:41)
>
> $$F = \frac{1}{3}\left(\frac{26\pi}{3} - 14 + 2\sqrt{2} - 4\sqrt{3} + 10\sqrt{5} - 2\sqrt{6} + 26\log(2) - \log(25) + 10\log(1+\sqrt{2})\right.$$
> $$\left. + 20\log(1+\sqrt{3}) - 35\log(1+\sqrt{5}) + 6\log(1+\sqrt{6}) - 2\log(4+\sqrt{6}) - 22\tan^{-1}(2\sqrt{6})\right)$$

Trivia, from my thread, SOHCAHTOA, for arc-trig

```
atan(√(24/1))          // TOA, O=24, A=1
= acos(√(1/25))        // CAH, H = O+A = 25
= acos(1/5)
```

EMAIL PM FIND QUOTE REPORT

10th April, 2022, 03:21 **Post: #20**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Albert Chan Wrote:** (10th April, 2022 02:37)
>
> > **Valentin Albillo Wrote:** (9th April, 2022 23:41)
> >
> > $$F = \frac{1}{3}\left(\frac{26\pi}{3} - 14 + 2\sqrt{2} - 4\sqrt{3} + 10\sqrt{5} - 2\sqrt{6} + 26\log(2) - \log(25) + 10\log(1+\sqrt{2})\right.$$
> > $$\left. + 20\log(1+\sqrt{3}) - 35\log(1+\sqrt{5}) + 6\log(1+\sqrt{6}) - 2\log(4+\sqrt{6}) - 22\tan^{-1}(2\sqrt{6})\right)$$
>
> Trivia, from my thread, SOHCAHTOA, for arc-trig
>
> ```
> atan(√(24/1))          // TOA, O=24, A=1
> = acos(√(1/25))        // CAH, H = O+A = 25
> = acos(1/5)
> ```

That shaves off 2 steps from the program which evaluates the exact ***F***'s expression, leaving it at *77 steps* (for now ...). Well done !

Regards.
V.

PM WWW FIND EDIT QUOTE REPORT

10th April, 2022, 11:20 (This post was last modified: 11th April, 2022 10:10 by EdS2.) **Post: #21**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

A marvellous and remarkable feat, Valentin - well done, and thank you for sharing your approach, and your other approach.

The double pseudo-random number generator is very nice indeed. I would never have guessed there'd be room to generate (and use) six random numbers.

Thanks also Albert for your subsequent workings. It's particularly nice to see your arc-SOHCAHTOA method in use, and so soon after posting.

It feels to me that uniform sampling would be just as accurate as random sampling, although as it turns out it would more expensive in terms of machinery. Because in this presentation we're allowed first to decide how many steps to run, a uniform approach is natural. A random sampling approach has the great advantage that it can keep running and continue to make progress, without needing to complete some particular number of steps.

But, it seems to me that there might be an interesting way to use uniform sampling with an unbounded count, using some sort of space-filling reordering of the numbers in the interval. Perhaps a simple bitwise operation, if implementing on a 16C or a conventional machine.

In the past I've used(*) int(x+phi) as an iterator, as a way to 'fill space' in a deterministic way. It's not uniform. And it's very much not random. But it is perhaps approximately as pseudo-random as the R-D approach seen here... and simpler conventionally, but less simple in a world where we have an R-D function!

(*) oops, I meant frac(x+phi) of course!

EMAIL PM FIND QUOTE REPORT

10th April, 2022, 18:29 **Post: #22**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **EdS2 Wrote:** (10th April, 2022 11:20)
> It feels to me that uniform sampling would be just as accurate as random sampling, although as it turns out it would more expensive in terms of machinery.

I am not so sure random sampling is accurate at all.
This is like throwing dart to estimate pi. Try running it twice

Uniform sampling might not need as many points, if we extrapolate results.
Below, FNA(A,B,C) is Aitken's delta-squared process, extraploate from 3 known points.

No tricks. Just sum all forces pressing the 2 planets, divided up N^3, M^3 tiny cubes
(If either N, M is even, it take advantage of quadrant symmetry, and do just 1 corner)

Code:

```
10 DEF FNA(A,B,C)=C-(C-B)^2/(C-B-(B-A))
20 INPUT "N,M ? ";N,M
30 N3=1/N @ N1=N3/2 @ N2=.5 @ IF MOD(N,2) THEN N2=1
40 M3=1/M @ M1=M3/2 @ M2=1.5-N2 @ IF MOD(M*N,2) THEN M2=1
50 T=TIME @ S=0
60 FOR X1=N1 TO 1 STEP N3 @ FOR X2=M1 TO 1 STEP M3 @ X=1+X2-X1
70 FOR Y1=N1 TO N2 STEP N3 @ FOR Y2=M1 TO M2 STEP M3 @ Y=Y2-Y1
80 FOR Z1=N1 TO N2 STEP N3 @ FOR Z2=M1 TO M2 STEP M3 @ Z=Z2-Z1
90 S=S+X/(X*X+Y*Y+Z*Z)^1.5
100 NEXT Z2 @ NEXT Z1 @ NEXT Y2 @ NEXT Y1 @ NEXT X2 @ NEXT X1
110 DISP S/(M*N)^3/(N2*M2)^2,TIME-T @ GOTO 20
```

```
>RUN
N,M ? 2,2
 .942585572032      .11
N,M ? 4,4
 .929717192068      5.66

>FNA(1, .942585572032, .929717192068)
 .925999798263
```

We get F required 3 digits accuracy, wth 2^6/4 + 4^6/4 = 16 + 1024 = 1040 points.

With N=M, F is over-estimated (unrealistically many points with cos(θ) = 1)

Extrapolated result removed (most of) this built-in bias.

Because we are doing F(1,1), F(2,2), F(4,4), we can also do Richardson Extrapolation.
Again, we get F required 3 digits accuracy (third column)

1
.942585572032 .923447429376
.929717192068 .925427732080 .925559752260

EMAIL PM FIND QUOTE REPORT

11th April, 2022, 12:12 (This post was last modified: 11th April, 2022 12:18 by J-F Garnier.) **Post: #23**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Thanks Valentin, for the conclusion.

A nice "challenge" for the April 1st Fools' Day
I enjoyed the reading, really.

It reminded me a quote from "les Shadoks" (French nonsense humour animated cartoon of the late sixties that became a cult):
"If you have 999 chances in 1000 that the thing will fail, so hurry to do the 999 first tests, because the 1000th will probably be the right one."

J-F

EMAIL PM WWW FIND QUOTE REPORT

12th April, 2022, 20:37 (This post was last modified: 12th April, 2022 20:48 by J-F Garnier.) **Post: #24**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

I had to come back to Valentin's solution, because I was puzzled: how did Valentin succeed to get (almost) 3 correct figures whereas I was unable to get more than one?
Our solutions are so close, we even used the same sequence ENTER SQRT * to compute the power 3/2.
Do 1312 samples make a so big difference with 999?
Is Valentin's random generator so much better than mine?

Then I understood. **I made a big, stupid mistake**. I set the initial seed of my random generator to PI.
It was not good because a seed must be between 0 and 1 (both excluded).
I wanted to keep PI as part of the seed because PI is a transcendental number (and what is more random than a transcendental number?) so I changed my sequence
PI STO 00 ; rnd seed
to
PI **1/X** STO 00 ; rnd seed
And my program immediately worked much better, even better than Valentin's one.
My program worked so much better that I was puzzled again.

And again, the light came.
Valentin's generator suffers from a big flaw: the sequence [previous seed] ->DEG FRAC, with inputs between about 0.175 and 1 (so about 82.5% of the time) gives a pseudo-random number with only 8 decimal places.
On the contrary, my sequence [previous seed] PI * FRAC *always* gives 9 decimal places.
So my generator provides numbers that are *10 times more dense* (in the mathematical sense of the word) 82.5% of the time and so is *more efficient by a factor of 8.25.*

The consequence is that my (corrected) program needs much less samples.
A more detailed analysis revealed that the amount of needed samples is reduced by a factor of 7.25 only, not 8.25 because numbers ending with one 0 at the last place don't count.
So I was just needing 1312 (the number Valentin estimated for his generator) divided by 7.25, i.e. **just 181 samples. What a difference!**

Running my very slightly modified program (1/X inserted at step 5) with 181 samples quickly provides me this answer, actually **more accurate than Valentin's result**:
--> **0.92616** [19182] at less than 0.0002 from the exact value 0.92598 [12605] !

Of course, my program using my version of the random number generator can't fit in a 10C.
And it's fortunate for *Rd. Albizarro*, otherwise he wouldn't had time to enjoy a good diner !

J-F

P.S. If you are tempted to take all this too seriously, just experiment with either program version (mine or Valentin's one) on a fast 15C or 41C emulator for instance, and you will get it soon.

EMAIL PM WWW FIND QUOTE REPORT

12th April, 2022, 23:46 (This post was last modified: 12th April, 2022 23:48 by Albert Chan.) **Post: #25**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (12th April, 2022 20:37)
> Valentin's generator suffers from a big flaw: the sequence [previous seed] ->DEG FRAC, with inputs between about 0.175 and 1 (so about 82.5% of the time) gives a pseudo-random number with only 8 decimal places.
> On the contrary, my sequence [previous seed] PI * FRAC *always* gives 9 decimal places.

I think Valentin "losing" 2 digits (when integer part get removed) is more random, not less.
Any patch of small seed will quickly get randomized. (it get multiply by 57.29..., not 3.14...)

Also, gain of **least significant random digits** mean very little when we sum forces.

Say, we sum 1000 point mass forces, and expected to get around 926.

If we solve FNF(X,X,X) = 926, we get X ≈ 0.0144
Hitting even 1 case within this sphere, we already passed sum of 926.

The singularity make Monte Carlo integration unsuitable.

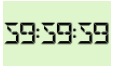
```
10 DEF FNF(X,Y,Z)=X/(X*X+Y*Y+Z*Z)^1.5
20 INPUT "N ? ";N @ S=0
30 FOR I=1 TO N @ S=S+FNF(1+RND-RND,RND-RND,RND-RND) @ NEXT I
40 DISP S/N @ GOTO 20

>RUN
N ? 1000
 .924145126462
N ? 1000
 .899430246256
N ? 1000
 .919886932633
N ? 1000
 1.11304106775
```

If I stopped at first 1000 samples, I get F = 0.924.
But, that's just lucky. Result cannot be repeated.

With RND giving 12 random digits, and 4000 samples, we can barely get 1 digit.

EMAIL PM FIND QUOTE REPORT

13th April, 2022, 13:24 (This post was last modified: 13th April, 2022 16:06 by Werner.) **Post: #26**



Werner
Senior Member

Posts: 777
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Albert Chan Wrote:** (10th April, 2022 18:29)
> I am not so sure random sampling is accurate at all.
> [..]
> If I stopped at first 1000 samples, I get F = 0.924.
> But, that's just lucky. Result cannot be repeated.
>
> With RND giving 12 random digits, and 4000 samples, we can barely get 1 digit.

Valentin pulled an April fool's joke on us all after all.
Just try the exact same program with seed 0.5 instead of 1.
Then your result is 0.848...
If you run it with seed 0.5 and 10000 points, I get 1.02899..
100'000 points (equivalent program on Free42) -> 1.1189...
Seed 1 just happened to get very close to the correct result ;-)
ie. I got it, J-F ;-)

Cheers, Werner

EMAIL PM FIND QUOTE REPORT

13th April, 2022, 17:05 (This post was last modified: 13th April, 2022 17:13 by J-F Garnier.) **Post: #27**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Werner Wrote:** (13th April, 2022 13:24)
> Seed 1 just happened to get very close to the correct result ;-)
> ie. I got it, J-F ;-)

Let's introduce the *Valentin-Bizarro* conjecture: for any seed, there is at least one value for the number of samples that makes the sum be as close as desired to the exact value. :-)

J-F

EMAIL PM WWW FIND QUOTE REPORT

14th April, 2022, 00:10 **Post: #28**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (13th April, 2022 17:05)
> > **Werner Wrote:** (13th April, 2022 13:24)
> > Seed 1 just happened to get very close to the correct result ;-)
> > ie. **I got it, J-F** ;-)
>
> Let's introduce the *Valentin-Bizarro* **conjecture**: for any seed, there is at least one value for the number of samples that makes the sum be as close as desired to the exact value. :-)

You *both* got it ! And that's not a *conjecture*, that's a *theorem* ... 😀

Next, ***My Comments***.

Best regards.
V.

PM WWW FIND EDIT QUOTE REPORT

14th April, 2022, 00:42 **Post: #29**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Hi, all,

Thanks a lot to those of you who posted some comments, namely **Albert Chan**, **EdS2**, **ijabbott**, **Massimo Gnerucci**, **rawi**, **Ren**, **rprosperi**, **vaklaff**, **Werner**, and last but certainly not least, **J-F Garnier**, much appreciated. Now, for my own final *Comments*:

My Comments

1) On my solution's algorithm and poor-man's random number generator

As I've said, there's no way to implement a deterministic cubature algorithm for a *non-trivial definite* ***6-D*** *integral* like the one featured here in just ***79*** *bytes* of *RAM* (including both program and required data storage) and with *no* subroutines, so I had to use the *poor-man*'s *RNG* which I advocated (and sent to ***Richard Nelson*** for publication in ***PPC CJ***, where it was indeed eventually published *40+* years ago) as the *fastest & smallest* one which was still capable of producing decent results.

As you can see in the above linked vintage letter *(page 7)*, it can be used in the **HP-41**, **HP-67/97** and any calculators featuring a built *radians-to-degrees* conversion, and is particularly useful for games and simulations, due to its speed and simplicity.

This *RNG* generates uniformly distributed pseudo-random numbers in the interval *0-1* and the seed can be any integer or real number *except* ***0***, ***Pi*** or its multiples. It essentially uses a multiplier equal to *180/Pi*, and as you can see in the vintage letter, a trial test generating and analyzing 1,000 values produced a decent uniform distribution with *mean* = ***4.4*** and *standard deviation* = ***2.9*** where the theoretical values are ***4.5*** and ***3.0***, respectively, close enough. I also couldn't find its period back then after generating 3,000 values.

J-F Garnier had the correct idea and almost succeeded in duplicating my program, as seen in his post, but he couldn't fit his program in the **HP-10C** and more important, he didn't discover the possibility of using the ***->DEG*** instruction available in the *HP-10C*, which essentially uses *180/Pi = 57.2957795+* as the multiplier, and so he used *Pi = 3.14159265+* instead. The problem with such a low-valued multiplier is that it's very prone to *ascendent runs*. For instance, if the seed ever becomes a low value such as *0.01*, you'll get a long ascendent run:

```
0.0100 -> 0.0314 -> 0.0987 -> 0.3101 -> 0.9741
```

that is, *5 consecutive values in increasing order*, which means a linear *dependency* from the previous value and damages the overall randomness, which is probably why ***J-F***'s program couldn't achieve a sufficiently accurate answer. If the seed eventually gets *smaller* than *0.01*, you'll get even *longer* ascending runs (7 values, 10 values, ...).

My method has this problem too, but to a far lesser extent because the **->DEG** multiplier *(57.30+)* is much bigger than ***J-F***'s *3.14+* and thus any ascending runs are far shorter. For instance, with the same seed:

```
0.0100 -> 0.5730
```

and the next number generated may or may not be in ascending order. Matter of fact, the worst that can happen to a method which uses just a *multiplier* (unlike *linear congruential generators*, which use *multiplication*, *addition* and *modulus* operations) is that if ever the seed becomes *exactly* ***0***, then the method will be stuck on an indefinite loop, always producing ***0***, and as I've explained above, whenever the seed becomes *very small*, then you'll get a *long* ascendent run if the multiplier is also small, like ***J-F***'s *Pi*.

2) On non-deterministic cubature methods

When numerically computing definite integrals in a *single* variable, it's quite common and efficient to use *deterministic* methods that evaluate the function being integrated at a number of well-chosen arguments (such as *16-point Gaussian* quadrature, say), which for reasonably well-behaved integrands are both fast and accurate.

However, computing a *double* integral in two variables to the same level of accuracy would roughly need 16^2 integrand evaluations and in general computing a multiple integral in *D* variables will require *grosso modo* about 16^N evaluations, which for the *6-D* integral featured here would be $16^6 \sim 17$ *million* evaluations.

When the number of evaluations needed to compute the integral grows *exponentially* with the dimension ***D***, then the integration method suffers from the so-called ***curse of dimensionality***, which means that deterministic methods are utterly *inefficient* for high-dimensional integrals (such as the ones appearing in *mathematical/computational finance*, where integrals having hundreds *(D > 100)* and even thousands *(D > 1000)* of variables aren't uncommon) and in practice it is mandatory to resort to *non-deterministic* ***Monte Carlo (MC)*** methods, which *do not* suffer the curse of dimensionality but converge *very slowly*.

Matter of fact, simple *MC* methods converge as slowly as $\mathbf{1/\sqrt{D}}$, which means that to get one additional correct digit (*10x* accuracy) we must use *100x* the number of evaluations, but we can resort instead to ***Quasi-Monte Carlo (QMC)*** methods, which attempt to speed up the convergence to ***1/D*** (i.e. to increase *10x* the accuracy you have to increase *10x* the number of evaluations, not *100x*) by using *low-discrepancy* sequences *(aka quasi-random sequences)* instead of sequences of *(pseudo-)*random numbers, as ***MC*** uses.

The gains in speed and accuracy that ***QMC*** methods afford over simple ***MC*** (let alone *deterministic* methods) for multidimensional integration is extremely noticeable, e.g. requiring as little as *500* integrand evaluations to compute a ***25-D*** test integral within *0.01*, as compared to *220,000* evaluations using ***MC***.

Mind you, none of this would fit in ***79*** *bytes* at all, so I did the best I could given the circumstances !! 😀

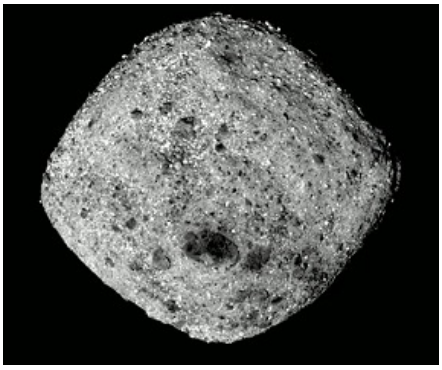
3) On the gravitational force F between two cubical planets

In the case of spherical planets in contact *(m_1=1, m_2=1, d=1, G=1)*, the gravitational force ***F*** is ***1***, but if the planets are *cubical* and in contact, we have ***F<1*** because they have part of their mass in the *corners*, which are farther away.

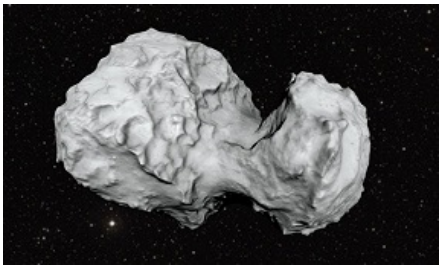
If instead of being in contact the cubes were at a distance *d > 1* or even *d >> 1*, then ***F*** would quickly approach $\mathbf{1/d^2}$ and the cubes would act more and more like *spheres* in that their mass could be considered as a point mass at their centers, like in the spherical case. Indeed, by the time the centers of the cubic planets are ***4*** or more units apart *(d >= 4)*, they can be practically considered spherical as far as gravity is concerned.

4) On real-life applications of computing the gravitational field of a cubical object

In the past few years, a number of spacecraft have been sent to visit diverse astronomical objects, from ***1 Ceres*** (dwarf planet, 939 km mean diameter, visited by ***Dawn***) to ***101955 Bennu*** (asteroid, 490 m mean diameter, visited by ***OSIRIS-REx***, which first orbited, then successfully touched down on its surface and later departed for Earth). *Ceres* is big enough to have a reasonably spherical shape, but *Bennu* is markedly *"squarish"*:



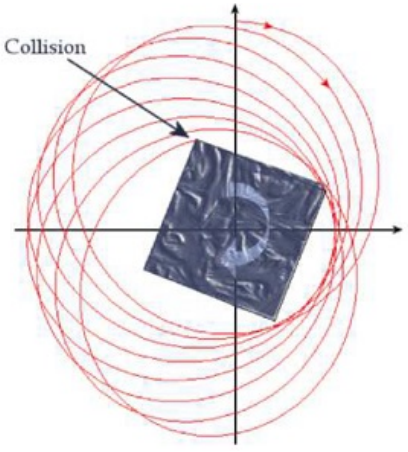
and other irregular objects also visited by spacecraft include two-lobed comet ***67P/Churyumov–Gerasimenko***, which is much further away from sphericity:



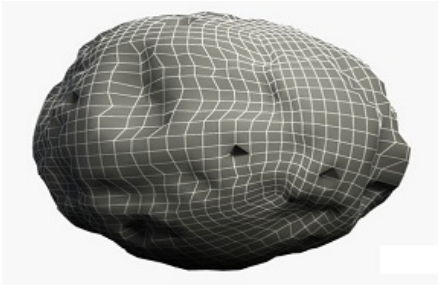
As another such instance, the asteroid ***433 Eros*** (~17 km mean diameter) also has a highly irregular shape and was visited by the ***NEAR Shoemaker*** spacecraft, which was initially put on a relatively distant *~320-360 km* elliptical orbit. At that distance, *Eros*' gravitational field could be considered as if the mass of the asteroid were concentrated in its center but later, when *NEAR* was moved to a much closer orbit and eventually *landed* on the asteroid, it was necessary to compute a more accurate gravitational field, least the spacecraft would impact the asteroid at a potentially dangerous speed.

The problem is compounded if, as it's usually the case, the object not only has an *irregular* shape but it's also *rotating*. In the case of a cubic planet with a side equal to Earth's diameter, can an artificial satellite orbit it ? For starters, it will feel a *stronger* gravitational attraction near the cube's *corners* and there will be additional *resonances* as the planet rotates. Moreover, again due to the corners, the satellite won't follow a *closed* elliptical orbit but will instead be subject to rapid *precession* and in general the orbit won't close at all.

If both the planet and the satellite rotate in the same direction, with the satellite orbiting a few planet's radii high, the differential rotation will perturb the orbit so much when the satellite is near the cube's corners that eventually it will *collide* with the planet, like this:



Thus, launching satellites in low orbits around non-spherical bodies requires careful calculation to overcome the perturbations, and there's a number of academic publications on the gravity field of a cube, with the resulting formulae being used in real life to compute the gravitational field of a body of *irregular* shape by superimposing on the object a *3D* grid of cubic blocks, iteratively reducing their size until the desired accuracy is attained, like this *(approximately, you get the idea)* :



Such methods can be tested as I did here, by applying them to a case whose solution is *known*, i.e. a *spherical* planet, whose shape is approximated by iteratively filling up its volume with cubic blocks of diminishing size as per the algorithm, then integrating the gravitational force over all the cubic blocks. This can also be applied to *non-homogeneous* bodies by using cubic blocks small enough for them to be considered individually *homogeneous*, then having their individual densities vary as needed.

And just in case you'd think that *cubical planets* wouldn't be taken seriously by anyone ...

The New York Times

THE CUBICAL PLANET.; THE SURPRISING THEORY BASED ON ITS ALLEGED DISCOVERY.

November 16, 1884, Page 10

The alleged discovery by Arndt, of Munich, of a planet whose orbit lies beyond that of Neptune, and which differs from all other reputable celestial bodies in the fact that it is neither a sphere nor a spheroid, but a cube, has naturally caused some excitement among scientific men, who, however, nearly all discredit the report.

Well, that's more than enough !

If you want to comment something about my *OP*, my *Original Solution* and/or my *ersatz RNG* you're welcome to post it to ***this*** very thread. But for comments on general ***Monte Carlo*** or ***quasi-Monte Carlo*** methods or general space exploration please create ***another*** thread so that this one may remain on-topic and focused on my *OP*. ***Thanks !***

This will be my last ***SRC*** for a long while, hope you enjoyed it. Thanks for your interest and

Best regards.
V.

*P.S.: A final question: knowing that the Bizarro given name "Nitnelav" is unisex (like the English given names Morgan, Cameron or Hayden, say), what do you think ? Is **Rd. Albizarro** a Bizarro-man or a Bizarro-woman ?*

PM WWW FIND EDIT QUOTE REPORT

14th April, 2022, 07:51 **Post: #30**

Massimo Gnerucci
Senior Member

Posts: 2,587
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Valentin Albillo Wrote:** (14th April, 2022 00:42)
> *P.S.: A final question: knowing that the Bizarro given name "Nitnelav" is unisex (like the English given names Morgan, Cameron or Hayden, say), what do you think ? Is **Rd. Albizarro** a Bizarro-man or a Bizarro-woman ?*

Gender **X**

PM WWW FIND QUOTE REPORT

14th April, 2022, 14:49 (This post was last modified: 15th April, 2022 20:34 by Albert Chan.) **Post: #31**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (12th April, 2022 20:37)
> my sequence [previous seed] PI * FRAC *always* gives 9 decimal places.

Amazingly, above random generator is so bad, it is more likely get the right answer

s = FRAC(deg(s)), for (1+RND-RND), we get expected triangular distribution.

```
>>> def r(): global s; s*=m; s-=int(s); return s
...
>>> m, s, t = 57.295779513082323, 1, [0]*20
>>> for i in range(100000): t[int(10*(1+r()-r()))] += 1
...
>>> for i in range(20): print (i+.5)/10, '*' * int(t[i]/200+0.5)
...
0.05 **
0.15 ********
0.25 ************
0.35 ****************
0.45 ************************
0.55 ****************************
0.65 ********************************
0.75 ************************************
0.85 ********************************************
0.95 ************************************************
1.05 ********************************************
1.15 ************************************
1.25 ********************************
1.35 ****************************
1.45 ************************
1.55 ********************
1.65 ****************
1.75 ************
1.85 ********
1.95 ***
```

s = FRAC(PI*s), and same seed of 1, we get this instead

```
>>> m, s, t = 3.1415926535897931, 1, [0]*20
>>> for i in range(100000): t[int(10*(1+r()-r()))] += 1
...
>>> for i in range(20): print (i+.5)/10, '*' * int(t[i]/200+0.5)
...
0.05
0.15
0.25
0.35 ******************
0.45 ************************
0.55 ************************
0.65 ************************************
0.75 ****************************************************
0.85 ****************************************************
0.95 ************************************************************
1.05 ******************************************
1.15 ******************************************
1.25 ********************************************
1.35 **************************
1.45 ********************
1.55 ********************
1.65 ********
1.75
1.85 *********
1.95 ************

>>> t
[0, 0, 0, 3857, 4859, 4853, 7512, 10460, 10454, 12339, 8511, 8838, 9051, 5188, 4293, 4155, 1554, 0, 1715, 2361]
>>> sum(t[0:10]), sum(t[10:])
(54334, 45666)
```

This has distribution slightly closer to singularity, but with zero chance of getting too close !

Update: I was wrong.

JFG's rand generator, s = FRAC(PI*s), F converge to 0.941
VA's rand generator, s = FRAC(DEG(s)), F converge to 0.925, almost dead-on (true F = 0.926)

The only problem is that it take a lot of random numbers for F convergence.

EMAIL PM FIND QUOTE REPORT

14th April, 2022, 16:48 **Post: #32**

J-F Garnier
Senior Member

Posts: 819
Joined: Dec 2013

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Albert Chan Wrote:** (14th April, 2022 14:49)
> Amazingly, above random generator is so bad, it is more likely get the right answer
>
> >>> t
> [0, 0, 0, 3857, 4859, 4853, 7512, 10460, 10454, 12339, 8511, 8838, 9051, 5188, 4293, 4155, 1554, 0, 1715, 2361]
> >>> sum(t[0:10]), sum(t[10:])
> (54334, 45666)

Ok, ok, my generator is not so good...

Comparing your results with an actual 10-digit BCD machine (HP-15C, actually the ultra fast 15C emulator from HP - just a few seconds for the 100,000 iterations):
>>> t
[0, 0, 0, 3857, 4859, 4853, 7512, 10460, 10454, 12339, 8511, 8838, 9051, 5188, 4293, 4155, 1554, 0, 1715, 2361]
(0, 0, 0, 4064, 4910, 4930, 7568, 10477, 10290, 12258, 8228, 8821, 9377, 5204, 4274, 4051, 1543, 0, 1775, 2230 - real 15C)
>>> sum(t[0:10]), sum(t[10:])
(54334, 45666)
(54497, 45503 - real 15C)
so no big difference due to the platform, and the same empty classes.

J-F

EMAIL PM WWW FIND QUOTE REPORT

14th April, 2022, 17:39 (This post was last modified: 14th April, 2022 17:49 by Albert Chan.) **Post: #33**

Albert Chan
Senior Member

Posts: 2,148
Joined: Jul 2018

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **J-F Garnier Wrote:** (14th April, 2022 16:48)
> so no big difference due to the platform, and the same empty classes.

Thanks for checking on a decimal machine.

We can explain the empty classes by getting min(1+RND-RND)
Here RND = (s = FP(PI*s)), and assumed calculations from left to right.
In other words, result of 1st RND is seed of 2nd RND

To minimize the expression, 2nd RND must be close to 1 (but, not yet reached it)
2nd RND cannot be close to 2 or 3, because first RND must be small.

min(1+RND-RND) = 1 + 1/PI - 1 = 1/PI ≈ 0.31831

3 classes, range of [0,0.1), [0.1,0.2), [0.2,0.3) must be empty

---

By same logic, with RND = (s = FP(DEG(s))), min(1+RND-RND) = PI/180 ≈ 0.01745

EMAIL PM FIND QUOTE REPORT

15th April, 2022, 08:49 **Post: #34**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

Very nice twist in the tail with the carefully sculpted RNG, Valentin! And your comments on that topic, and the expected number of trials for various dimensions of problem, led to a very interesting read on The Unreasonable Effectiveness of Quasirandom Sequences. Thanks!

EMAIL PM FIND QUOTE REPORT

25th April, 2022, 00:02 **Post: #35**

ijabbott
Senior Member

Posts: 1,242
Joined: Jul 2015

RE: [VA] SRC #011 - April 1st, 2022 Bizarro Special

> **Valentin Albillo Wrote:** (5th April, 2022 00:34)
>
> .
>
> > **Ren Wrote:**
> > I think a parallel can be seen in **Star Trek: Lower Decks** interactions with the **Paklid**.
>
> Thanks for the tip. I know nothing about the plethora of new *ST* series, I only watched the *Original Series*, *Deep Space 9*, *Voyager* and *Enterprise* (as well as all the movies), so no idea what *Lower Decks* is about or the aliens (I suppose they're aliens) you mention, sorry.

Just FYI, the Pakleds also appeared in one episode of TNG and 18 episodes of DS9. But Lower Decks is worth a watch (IMHO) if you get around to it.

EMAIL PM FIND QUOTE REPORT

« Next Oldest | Next Newest »

Enter Keywords Search Thread

NEW REPLY

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

User(s) browsing this thread: Valentin Albillo*