Normal computer floating point compute limitations

And why quantum computers give more accurate results.

Let us use this simple example:

A = 12345678912333653845631

B = 1

Type of A = <class 'int'>

Type of B = <class 'int'>

1. Normally if we say

Ok let us see it in computer tool like Python, which uses float data type as default datatype in case of divide two integer values. It is not the case if your numbers range exceeds the machine default precision where ; result represented in scientific notation and if you cast it to int the value is different.

(A/2) \* 2 = 1.2345678912333655e+22

int value for A != (A/2) \* 2 = 12345678912333654523904

1. Normally if we say.

But with float point data types, it is not the case if your numbers range exceeds the machine default precision where

A = (A \* 1) = 12345678912333653845631

int value for A != A \* 1.0 = 12345678912333654523904

so, the issue is not the operations used {multiplication \*} the issue how the hardware is handling the floating point.

because it is clear it is not depending on the value we multiply with. Even 1.0 representation as float will give us different value than what will have if used the multiplication operation but on 1.

The default hardware representation for {divide operation is float} this is why even if both nominator and denominator are integers the result will be float and

Type of A/B = <class 'float'>

 A /1 != A/B = 1.2345678912333655e+22

int value for A/B = 12345678912333654523904

float Value for A/B = 1.2345678912333655e+22

str Value for A/B = 1.2345678912333655e+22

Decimal value for A/B = 12345678912333654523904

Decimal value for A/B = 12345678912333654523904

My suggestion to workaround this limitation in normal computers, is to stick to parsing the scientific notation and multiply by power of 10 is the most accurate or to be more specific the best way to get steady results between different machines.

The issue is already caried from the hardware calculations to the results, so no casting change in the data type on the final calculation will give you the exact or steady results between different machines and tools. Here are some other examples for these limitations.

A \* 2 = 24691357824667307691264

(A/1 \* 2) != (A/B) \* 2 = 2.469135782466731e+22

int value for (A/B) \* 2 = 24691357824667309047808

round(round(A,0)/B,0) = 12345678912333654523904

round(round(A,0)/round(B,0),0) = 12345678912333654523904

int value for (A \* B\*\*-1) =12345678912333654523904

B\*\*-1 =1.0

This is how much your accuracy can be affected for this simple example.

A = 12345678912333653845632

B = 1

int((A/B) \* 1.0) = 12345678912333654523904

A-int((A/B) \* 1.0) = -678272

This residue will be different based on the number you choose for A

For example, if A have different value as to have 9 as last digit instead of 2 as last digit the residue will be different.

A = 12345678912333653845639

B = 1

int((A/B) \* 1.0) = 12345678912333654523904

A-int((A/B) \* 1.0) = -678265

Even if we try to readd this residue back will never get the same A again.

(A + 678265) != A = 12345678912333654523904

(A - 678265) != A = 12345678912333653167374

If A ends in odd number (3,5,7,9) adding 1 or subtract 1 will give same scientific representation number or its int casing will be the same number. 1.2345678912333652e+16 if number eds in 1 or 2 or 3.

Also the numbers for int((A/B) \* 1.0) = (A + 1) or (A -1)

For example

if A ends in 1 the scientific number will be A+1

if A ends in 2 the scientific number will be A-1

if A ends in 3 the scientific number will be A+1

If A ends in 5 the scientific number will be A-1

If A ends in 7 the scientific number will be A+1

If A ends in 9 the scientific number will be A-1

A = 12345678912333651

B = 1

int((A/B) \* 1.0) = 12345678912333652

A-int((A/B) \* 1.0) = -1

int((A/B) \* 1.0 + 1) =12345678912333652

int((A/B) \* 1.0 - 1) = 12345678912333652

int((A/B) \* 1.0)+ 1.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)- 1.0 =1.2345678912333652e+16

int((A/B) \* 1.0)+ 5.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 4.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 3.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333658e+16

int((A/B) \* 1.0)- 6.0 = 1.2345678912333646e+16

A = 12345678912333652

B = 1

int((A/B) \* 1.0) = 12345678912333652

A-int((A/B) \* 1.0) = 0

int((A/B) \* 1.0 + 1) =12345678912333652

int((A/B) \* 1.0 - 1) = 12345678912333652

int((A/B) \* 1.0)+ 1.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)- 1.0 =1.2345678912333652e+16

int((A/B) \* 1.0)+ 5.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 4.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 3.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333658e+16

int((A/B) \* 1.0)- 6.0 = 1.2345678912333646e+16

A = 12345678912333653

B = 1

int((A/B) \* 1.0) = 12345678912333652

A-int((A/B) \* 1.0) = 1

int((A/B) \* 1.0 + 1) =12345678912333652

int((A/B) \* 1.0 - 1) = 12345678912333652

int((A/B) \* 1.0)+ 1.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)- 1.0 =1.2345678912333652e+16

int((A/B) \* 1.0)+ 5.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 4.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 3.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333648e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333658e+16

int((A/B) \* 1.0)- 6.0 = 1.2345678912333646e+16

if A = 12345678912333655

+ 1 or -1 = 1.2345678912333656e+16

A = 12345678912333655

B = 1

int((A/B) \* 1.0) = 12345678912333656

A-int((A/B) \* 1.0) = -1

int((A/B) \* 1.0 + 1) =12345678912333656

int((A/B) \* 1.0 - 1) = 12345678912333656

int((A/B) \* 1.0)+ 1.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 1.0 =1.2345678912333656e+16

int((A/B) \* 1.0)+ 5.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 4.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 3.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333662e+16

int((A/B) \* 1.0)- 6.0 = 1.234567891233365e+16

A = 12345678912333657

+1 or -1 = 1.2345678912333656e+16

A = 12345678912333657

B = 1

int((A/B) \* 1.0) = 12345678912333656

A-int((A/B) \* 1.0) = 1

int((A/B) \* 1.0 + 1) =12345678912333656

int((A/B) \* 1.0 - 1) = 12345678912333656

int((A/B) \* 1.0)+ 1.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)- 1.0 =1.2345678912333656e+16

int((A/B) \* 1.0)+ 5.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 4.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 3.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333652e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333662e+16

int((A/B) \* 1.0)- 6.0 = 1.234567891233365e+16

A= 12345678912333659

+1 or -1 = 1.234567891233366e+16

A = 12345678912333659

B = 1

int((A/B) \* 1.0) = 12345678912333660

A-int((A/B) \* 1.0) = -1

int((A/B) \* 1.0 + 1) =12345678912333660

int((A/B) \* 1.0 - 1) = 12345678912333660

int((A/B) \* 1.0)+ 1.0 = 1.234567891233366e+16

int((A/B) \* 1.0)- 1.0 =1.234567891233366e+16

int((A/B) \* 1.0)+ 5.0 = 1.2345678912333664e+16

int((A/B) \* 1.0)- 5.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)+ 4.0 = 1.2345678912333664e+16

int((A/B) \* 1.0)- 4.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)+ 3.0 = 1.2345678912333664e+16

int((A/B) \* 1.0)- 3.0 = 1.2345678912333656e+16

int((A/B) \* 1.0)+ 6.0 = 1.2345678912333666e+16

int((A/B) \* 1.0)- 6.0 = 1.2345678912333654e+16

in a published paper we showed how we can get a workaround over this limitation when using normal computer machines “Journal of Mathematics Research; Vol. 15, No. 2; April 2023 ISSN 1916-9795 E-ISSN 1916-9809 Published by Canadian Center of Science and Education 31

“Relation Between the Golden Ratio Phi and Zeta Function SUM”

And

A = 12345678912333653845632384283483572489237528348737

B = 1

((sqrt(A)+1)/2) =1.756820915199786e+24

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 268435456.0

A != int(sqrtA \* A / sqrtA) = 12345678912333654175131969130212141977322787962880

N = math.sqrt(A)+1.0 = 3.513641830399572e+24

D = math.ceil(math.sqrt(A)/5.0) = 702728366079914394255360

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000009

A = 1234567891233365384563238428348357

B = 1

((sqrt(A)+1)/2) =1.756820915199786e+16

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 2.0

A != int(sqrtA \* A / sqrtA) = 1234567891233365349661899040489472

N = math.sqrt(A)+1.0 = 3.513641830399572e+16

D = math.ceil(math.sqrt(A)/5.0) = 7027283660799144

R = (D - 4) /(2 \* D) = 0.4999999999999997

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.4999999999999997

A = 12345678912333653845632

B = 1

((sqrt(A)+1)/2) =55555555331.0285

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 1.0000076293945312

A != int(sqrtA \* A / sqrtA) = 12345678912333654523904

N = math.sqrt(A)+1.0 = 111111110662.057

D = math.ceil(math.sqrt(A)/5.0) = 22222222133

R = (D - 4) /(2 \* D) = 0.49999999991

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000009

A = 123456789123336538456339

B = 1

((sqrt(A)+1)/2) =175682091520.47858

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 0.999969482421875

A != int(sqrtA \* A / sqrtA) = 123456789123336532656128

N = math.sqrt(A)+1.0 = 351364183040.95715

D = math.ceil(math.sqrt(A)/5.0) = 70272836608

R = (D - 4) /(2 \* D) = 0.4999999999715395

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.4999999999999994

A = 1234567891233365384563547

B = 1

((sqrt(A)+1)/2) =555555553305.785

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 1.0

A != int(sqrtA \* A / sqrtA) = 1234567891233365494333440

N = math.sqrt(A)+1.0 = 1111111106611.57

D = math.ceil(math.sqrt(A)/5.0) = 222222221323

R = (D - 4) /(2 \* D) = 0.499999999991

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000017

A = 12345678912333653845635475469859609286509264096209843652093864592865092836459246852

B = 1

((sqrt(A)+1)/2) =5.55555553305285e+40

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 0.0

A != int(sqrtA \* A / sqrtA) = 12345678912333653265479606169162259410924923957057170371538869050594188708829724672

N = math.sqrt(A)+1.0 = 1.11111110661057e+41

D = math.ceil(math.sqrt(A)/5.0) = 22222222132211396854822768981882198556672

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.4999999999999991

A = 1234567891233365384563547546985960928650926409620984365209386459286509283645983793994577540459246852

B = 1

((sqrt(A)+1)/2) =1.756820915199786e+49

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 0.0

A != int(sqrtA \* A / sqrtA) = 1234567891233365378769919853050218767635870041462831497102556543628100338678719450656609359427010560

N = math.sqrt(A)+1.0 = 3.513641830399572e+49

D = math.ceil(math.sqrt(A)/5.0) = 7027283660799143178462418801421762269550487797760

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5

A = 123456789123336538456354754698596092865092640962098089966340027523401978643563454365209386459286509283645983793994998698645963450957754045924685285233983745

B = 1

((sqrt(A)+1)/2) =1.7568209151997861e+77

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 2.5711008708143844e+61

A != int(sqrtA \* A / sqrtA) = 123456789123336549730735696513746947941234572899950011950324907921272512644492626950757053915350643657387371807511337744015612236541076557841011447275978752

N = math.sqrt(A)+1.0 = 3.5136418303995723e+77

D = math.ceil(math.sqrt(A)/5.0) = 70272836607991442910535919177408840471148018655887122613737328641726425858048

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000009

A = 13031

B = 1

((sqrt(A)+1)/2) =57.57670277792858

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 1.000000000000007

A != int(sqrtA \* A / sqrtA) = 13031

N = math.sqrt(A)+1.0 = 115.15340555585716

D = math.ceil(math.sqrt(A)/5.0) = 23

R = (D - 4) /(2 \* D) = 0.41304347826086957

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000004

A = 13353647586767576879876078765436476879411234567886523431234456778645679123456431234587536142235567896765436587876522345847362

B = 1

((sqrt(A)+1)/2) =5.777899182827521e+61

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= -1.141798154164768e+46

A != int(sqrtA \* A / sqrtA) = 13353647586767576971496915938502538007267211498391505023495887742352403234310427738752913303164385985487119148744348071362560

N = math.sqrt(A)+1.0 = 1.1555798365655042e+62

D = math.ceil(math.sqrt(A)/5.0)  = 23111596731310085019379151552953331562909021373344362012868608

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.4999999999999991

Equation (2) = round(((N/D) - (S /N)+R)) , 2) = 0.5

A = 1335364758676757687987607876543647687941123456788

B = 1

((sqrt(A)+1)/2) =5.7778991828275217e+23

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 0.0

A != int(sqrtA \* A / sqrtA) = 1335364758676757657044308162856531714698876289024

N = math.sqrt(A)+1.0 = 1.1555798365655043e+24

D = math.ceil(math.sqrt(A)/5.0)  = 231115967313100866060288

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5

Equation (2) = round(((N/D) - (S /N)+R)) , 2) = 0.5

A = 133536475867675768798760787654364768794112345673

B = 1

((sqrt(A)+1)/2) =1.8271321508560608e+23

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= 33554432.0

A != int(sqrtA \* A / sqrtA) = 133536475867675777873876578476655425838238400512

N = math.sqrt(A)+1.0 = 3.6542643017121216e+23

D = math.ceil(math.sqrt(A)/5.0)  = 73085286034242430566400

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.5000000000000009

Equation (2) = round(((N/D) - (S /N)+R)) , 2) = 0.5

A = 1335364758676757687987607876543647687941123450796857465342335678909776123428790876432147897324167867398675313422354786987563313658796753646374656858973631264357846735241326345847356214352345487356245132364

B = 1

((sqrt(A)+1)/2) =5.777899182827521e+101

Equation (1) D = 2  & ((sqrtA+1)/D) - (((A-1)/D) / (sqrtA+1))= -1.2433080910244666e+86

A != int(sqrtA \* A / sqrtA) = 1335364758676757637856482091226075157529655383084377805969683274345063734007691136223067140587682810567832712364730627102079412295857484339816091689712244508822714261556019691973568392566469958531008167936

N = math.sqrt(A)+1.0 = 1.1555798365655042e+102

D = math.ceil(math.sqrt(A)/5.0)  = 231115967313100839182046493667808402894907054094800400161993457082661659579288674500829274339973005312

R = (D - 4) /(2 \* D) = 0.5

S = (A-1) /D

Equation (2) = ((N/D) - (S /N)+R)) = 0.4999999999999982

Equation (2) = round(((N/D) - (S /N)+R)) , 2) = 0.5