

Вычисление машинного эпсилон в типах библиотек:

bigfloat

NumPy

mpmath

```
In [1]: In [1]: import sys
!{sys.executable} --version
```

Python 3.6.10 :: Anaconda, Inc.

```
In [2]: import math
import matplotlib.pyplot as plt
import numpy as np
import unittest
import math
from bigfloat import *
```

```
In [ ]: #1.4142135623730950488016887242096980785696718753769480731766796
        #1.4142135623730950488016887242096980785696718753769480731766797379907324784621070388
        503875348
        #1.4142135623730950488016887242096980785696718753769480731766797379907324784621070388
        503875343276415727350138462309122970252
```

```
In [17]: #precision(50)
Context(precision=1000)
with precision(400):
    print(sqrt(2))
```

1.4142135623730950488016887242096980785696718753769480731766797379907324784621070388
503875343276415727350138462309122970252

```
In [24]: with quadruple_precision:
          #const_pi()
          print(const_pi())
          print(BigFloat.exact('3.14159265358979323846264338327950280', precision=113))
```

3.14159265358979323846264338327950280
3.14159265358979323846264338327950280

[illegible][illegible]

In [26]: `getcontext()`

Out[26]: `Context(precision=53, emax=1073741823, emin=-1073741823, subnormalize=False, rounding=ROUND_TIES_TO_EVEN)`

In [34]: `precision(200)`
`#getcontext()`

Out[34]: `Context(precision=200)`

In [30]: `with RoundTowardNegative + precision(200): # RoundTowardPositive`
`print(const_pi())`

3.1415926535897932384626433832795028841971693993751058209749445

In [31]: `quadruple_precision`

Out[31]: `Context(precision=113, emax=16384, emin=-16493, subnormalize=True)`

In [35]: `half_precision`

Out[35]: `Context(precision=11, emax=16, emin=-23, subnormalize=True)`

In [36]: `with half_precision:`
`print(log(2))`

0.69336

```
In [38]: with precision(100):  
         i = 0  
         epsFloat = 1  
         while 1 + epsFloat != 1:  
             epsFloat = div(epsFloat, 2)  
             i += 1  
         print (i, epsFloat)
```

1 0.50000000000000000000000000000000
2 0.25000000000000000000000000000000
3 0.12500000000000000000000000000000
4 0.06250000000000000000000000000000
5 0.03125000000000000000000000000000
6 0.01562500000000000000000000000000
7 0.00781250000000000000000000000000
8 0.00390625000000000000000000000000
9 0.00195312500000000000000000000000
10 0.00097656250000000000000000000000
11 0.00048828125000000000000000000000
12 0.00024414062500000000000000000000
13 0.00012207031250000000000000000000
14 6.103515625000000000000000000000e-5
15 3.051757812500000000000000000000e-5
16 1.525878906250000000000000000000e-5
17 7.629394531250000000000000000000e-6
18 3.814697265625000000000000000000e-6
19 1.907348632812500000000000000000e-6
20 9.536743164062500000000000000000e-7
21 4.768371582031250000000000000000e-7
22 2.384185791015625000000000000000e-7
23 1.192092895507812500000000000000e-7
24 5.960464477539062500000000000000e-8
25 2.980232238769531250000000000000e-8
26 1.490116119384765625000000000000e-8
27 7.450580596923828125000000000000e-9
28 3.725290298461914062500000000000e-9
29 1.862645149230957031250000000000e-9
30 9.313225746154785156250000000000e-10
31 4.656612873077392578125000000000e-10
32 2.328306436538696289062500000000e-10
33 1.164153218269348144531250000000e-10
34 5.820766091346740722656250000000e-11
35 2.910383045673370361328125000000e-11
36 1.455191522836685180664062500000e-11
37 7.275957614183425903320312500000e-12
38 3.637978807091712951660156250000e-12
39 1.8189894035458564758300781250000e-12
40 9.0949470177292823791503906250000e-13
41 4.5474735088646411895751953125000e-13
42 2.2737367544323205947875976562500e-13
43 1.1368683772161602973937988281250e-13
44 5.6843418860808014869689941406250e-14
45 2.8421709430404007434844970703125e-14
46 1.4210854715202003717422485351562e-14
47 7.1054273576010018587112426757812e-15
48 3.5527136788005009293556213378906e-15
49 1.7763568394002504646778106689453e-15
50 8.8817841970012523233890533447266e-16
51 4.4408920985006261616945266723633e-16
52 2.2204460492503130808472633361816e-16
53 1.1102230246251565404236316680908e-16
54 5.5511151231257827021181583404541e-17
55 2.7755575615628913510590791702271e-17
56 1.3877787807814456755295395851135e-17
57 6.9388939039072283776476979255676e-18
58 3.4694469519536141888238489627838e-18
59 1.7347234759768070944119244813919e-18
60 8.6736173798840354720596224069595e-19
61 4.3368086899420177360298112034798e-19
62 2.1684043449710088680149056017399e-19
63 1.0842021724855044340074528008699e-19
64 5.4210108624275221700372640043497e-20
65 2.7105054312137610850186320021749e-20

```
66 1.3552527156068805425093160010874e-20
67 6.7762635780344027125465800054371e-21
68 3.3881317890172013562732900027186e-21
69 1.6940658945086006781366450013593e-21
70 8.4703294725430033906832250067964e-22
71 4.2351647362715016953416125033982e-22
72 2.1175823681357508476708062516991e-22
73 1.0587911840678754238354031258496e-22
74 5.2939559203393771191770156292478e-23
75 2.6469779601696885595885078146239e-23
76 1.3234889800848442797942539073119e-23
77 6.6174449004242213989712695365597e-24
78 3.3087224502121106994856347682799e-24
79 1.6543612251060553497428173841399e-24
80 8.2718061255302767487140869206996e-25
81 4.1359030627651383743570434603498e-25
82 2.0679515313825691871785217301749e-25
83 1.0339757656912845935892608650875e-25
84 5.1698788284564229679463043254373e-26
85 2.5849394142282114839731521627186e-26
86 1.2924697071141057419865760813593e-26
87 6.4623485355705287099328804067966e-27
88 3.2311742677852643549664402033983e-27
89 1.6155871338926321774832201016991e-27
90 8.0779356694631608874161005084957e-28
91 4.0389678347315804437080502542479e-28
92 2.0194839173657902218540251271239e-28
93 1.0097419586828951109270125635620e-28
94 5.0487097934144755546350628178098e-29
95 2.5243548967072377773175314089049e-29
96 1.2621774483536188886587657044525e-29
97 6.3108872417680944432938285222623e-30
98 3.1554436208840472216469142611311e-30
99 1.5777218104420236108234571305656e-30
100 7.8886090522101180541172856528279e-31
```

```
In [39]: print(pow(2, -24, precision(100)))
```

```
5.9604644775390625000000000000000e-8
```

```
In [2]: ### Расчет машинного эpsilon в типе данных float*32  
i = 0  
epsFloat = np.float32(1.)  
while np.float32(1 + epsFloat) != 1:  
    epsFloat /= 2; i += 1  
    print (i, epsFloat)  
print ( "Вычисленное значение машинного эpsilon = ", epsFloat )  
print ( "Теоретическое значение эpsilon, для мантиссы длиной 23 бита (float) = ", 2**(-24))
```

```
1 0.5  
2 0.25  
3 0.125  
4 0.0625  
5 0.03125  
6 0.015625  
7 0.0078125  
8 0.00390625  
9 0.001953125  
10 0.0009765625  
11 0.00048828125  
12 0.000244140625  
13 0.0001220703125  
14 6.103515625e-05  
15 3.0517578125e-05  
16 1.52587890625e-05  
17 7.62939453125e-06  
18 3.814697265625e-06  
19 1.9073486328125e-06  
20 9.5367431640625e-07  
21 4.76837158203125e-07  
22 2.384185791015625e-07  
23 1.1920928955078125e-07  
24 5.960464477539063e-08  
Вычисленное значение машинного эpsilon = 5.960464477539063e-08  
Теоретическое значение эpsilon, для мантиссы длиной 23 бита (float) = 5.960464477539063e-08
```

```
In [3]: ### Расчет машинного эpsilon в типе данных float*64
i = 0
epsFloat64 = np.float64(1.0)
while np.float64(1 + epsFloat64) != 1:
    epsFloat64 /= 2; i += 1
    print (i, epsFloat64)
print ( "Вычисленное значение машинного эpsilon = ", epsFloat64 )
print ( "Теоретическое значение эpsilon, для мантиссы длиной 52 бита (float) = ", 2**(-53))
```

1 0.5
2 0.25
3 0.125
4 0.0625
5 0.03125
6 0.015625
7 0.0078125
8 0.00390625
9 0.001953125
10 0.0009765625
11 0.00048828125
12 0.000244140625
13 0.0001220703125
14 6.103515625e-05
15 3.0517578125e-05
16 1.52587890625e-05
17 7.62939453125e-06
18 3.814697265625e-06
19 1.9073486328125e-06
20 9.5367431640625e-07
21 4.76837158203125e-07
22 2.384185791015625e-07
23 1.1920928955078125e-07
24 5.960464477539063e-08
25 2.9802322387695312e-08
26 1.4901161193847656e-08
27 7.450580596923828e-09
28 3.725290298461914e-09
29 1.862645149230957e-09
30 9.313225746154785e-10
31 4.656612873077393e-10
32 2.3283064365386963e-10
33 1.1641532182693481e-10
34 5.820766091346741e-11
35 2.9103830456733704e-11
36 1.4551915228366852e-11
37 7.275957614183426e-12
38 3.637978807091713e-12
39 1.8189894035458565e-12
40 9.094947017729282e-13
41 4.547473508864641e-13
42 2.2737367544323206e-13
43 1.1368683772161603e-13
44 5.684341886080802e-14
45 2.842170943040401e-14
46 1.4210854715202004e-14
47 7.105427357601002e-15
48 3.552713678800501e-15
49 1.7763568394002505e-15
50 8.881784197001252e-16
51 4.440892098500626e-16
52 2.220446049250313e-16
53 1.1102230246251565e-16

Вычисitanное значение машинного эпсилон = 1.1102230246251565e-16

Теоретическое значение эпсилон, для мантиссы длиной 52 бита (float) = 1.1102230246251565e-16


```
In [4]: ### Расчет машинного эпсилон в типе данных float*128 (не работает!!?)
i = 0
epsFloat128 = np.longdouble(1.0)
while np.longdouble(1 + epsFloat128) != 1:
    epsFloat128 /= 2; i += 1
    print (i, epsFloat128)
print ( "Вычисленное значение машинного эпсилон = ", epsFloat128 )
print ( "Теоретическое значение эпсилон, для мантиссы длиной 64 бита (float) = ", 2**(-64))
```

1 0.5
2 0.25
3 0.125
4 0.0625
5 0.03125
6 0.015625
7 0.0078125
8 0.00390625
9 0.001953125
10 0.0009765625
11 0.00048828125
12 0.000244140625
13 0.0001220703125
14 6.103515625e-05
15 3.0517578125e-05
16 1.52587890625e-05
17 7.62939453125e-06
18 3.814697265625e-06
19 1.9073486328125e-06
20 9.5367431640625e-07
21 4.76837158203125e-07
22 2.384185791015625e-07
23 1.1920928955078125e-07
24 5.960464477539063e-08
25 2.9802322387695312e-08
26 1.4901161193847656e-08
27 7.450580596923828e-09
28 3.725290298461914e-09
29 1.862645149230957e-09
30 9.313225746154785e-10
31 4.656612873077393e-10
32 2.3283064365386963e-10
33 1.1641532182693481e-10
34 5.820766091346741e-11
35 2.9103830456733704e-11
36 1.4551915228366852e-11
37 7.275957614183426e-12
38 3.637978807091713e-12
39 1.8189894035458565e-12
40 9.094947017729282e-13
41 4.547473508864641e-13
42 2.2737367544323206e-13
43 1.1368683772161603e-13
44 5.684341886080802e-14
45 2.842170943040401e-14
46 1.4210854715202004e-14
47 7.105427357601002e-15
48 3.552713678800501e-15
49 1.7763568394002505e-15
50 8.881784197001252e-16
51 4.440892098500626e-16
52 2.220446049250313e-16
53 1.1102230246251565e-16

Вычисitanное значение машинного эпсилон = 1.1102230246251565e-16

Теоретическое значение эпсилон, для мантиссы длиной 64 бита (float) = 5.421010862427522e-20

```
In [12]: ## Расчет машинного эpsilon в типе данных с назначенной длиной слова  
from mpmath import *  
mp.dps = 40 #  
i = 0  
epsFloat = mpf (1.0)  
while 1. + epsFloat != 1.:  
    epsFloat /= 2; i += 1  
    print (i, epsFloat)  
print ( "Вычисленное значение машинного эpsilon = ", epsFloat )  
print ("Теоретическое значение эpsilon = ", 10**(-40))
```

1 0.5
2 0.25
3 0.125
4 0.0625
5 0.03125
6 0.015625
7 0.0078125
8 0.00390625
9 0.001953125
10 0.0009765625
11 0.00048828125
12 0.000244140625
13 0.0001220703125
14 0.00006103515625
15 0.000030517578125
16 0.0000152587890625
17 0.00000762939453125
18 0.000003814697265625
19 0.0000019073486328125
20 0.00000095367431640625
21 0.000000476837158203125
22 0.0000002384185791015625
23 0.00000011920928955078125
24 0.000000059604644775390625
25 0.0000000298023223876953125
26 0.00000001490116119384765625
27 0.000000007450580596923828125
28 0.0000000037252902984619140625
29 0.00000000186264514923095703125
30 0.000000000931322574615478515625
31 0.0000000004656612873077392578125
32 0.00000000023283064365386962890625
33 0.000000000116415321826934814453125
34 0.0000000000582076609134674072265625
35 0.00000000002910383045673370361328125
36 0.000000000014551915228366851806640625
37 0.0000000000072759576141834259033203125
38 0.00000000000363797880709171295166015625
39 0.000000000001818989403545856475830078125
40 9.094947017729282379150390625e-13
41 4.5474735088646411895751953125e-13
42 2.27373675443232059478759765625e-13
43 1.136868377216160297393798828125e-13
44 5.684341886080801486968994140625e-14
45 2.8421709430404007434844970703125e-14
46 1.42108547152020037174224853515625e-14
47 7.10542735760100185871124267578125e-15
48 3.552713678800500929355621337890625e-15
49 1.7763568394002504646778106689453125e-15
50 8.8817841970012523233890533447265625e-16
51 4.44089209850062616169452667236328125e-16
52 2.220446049250313080847263336181640625e-16
53 1.1102230246251565404236316680908203125e-16
54 5.5511151231257827021181583404541015625e-17
55 2.77555756156289135105907917022705078125e-17
56 1.387778780781445675529539585113525390625e-17
57 6.938893903907228377647697925567626953125e-18
58 3.469446951953614188823848962783813476563e-18
59 1.734723475976807094411924481391906738281e-18
60 8.673617379884035472059622406959533691406e-19
61 4.336808689942017736029811203479766845703e-19
62 2.168404344971008868014905601739883422852e-19
63 1.084202172485504434007452800869941711426e-19
64 5.421010862427522170037264004349708557129e-20
65 2.710505431213761085018632002174854278564e-20

66 1.355252715606880542509316001087427139282e-20
67 6.776263578034402712546580005437135696411e-21
68 3.388131789017201356273290002718567848206e-21
69 1.694065894508600678136645001359283924103e-21
70 8.470329472543003390683225006796419620514e-22
71 4.235164736271501695341612503398209810257e-22
72 2.117582368135750847670806251699104905128e-22
73 1.058791184067875423835403125849552452564e-22
74 5.293955920339377119177015629247762262821e-23
75 2.646977960169688559588507814623881131411e-23
76 1.323488980084844279794253907311940565705e-23
77 6.617444900424221398971269536559702828526e-24
78 3.308722450212110699485634768279851414263e-24
79 1.654361225106055349742817384139925707132e-24
80 8.271806125530276748714086920699628535658e-25
81 4.135903062765138374357043460349814267829e-25
82 2.067951531382569187178521730174907133915e-25
83 1.033975765691284593589260865087453566957e-25
84 5.169878828456422967946304325437267834786e-26
85 2.584939414228211483973152162718633917393e-26
86 1.292469707114105741986576081359316958697e-26
87 6.462348535570528709932880406796584793483e-27
88 3.231174267785264354966440203398292396741e-27
89 1.615587133892632177483220101699146198371e-27
90 8.077935669463160887416100508495730991854e-28
91 4.038967834731580443708050254247865495927e-28
92 2.019483917365790221854025127123932747963e-28
93 1.009741958682895110927012563561966373982e-28
94 5.048709793414475554635062817809831869909e-29
95 2.524354896707237777317531408904915934954e-29
96 1.262177448353618888658765704452457967477e-29
97 6.31088724176809444329382852262289837386e-30
98 3.155443620884047221646914261131144918693e-30
99 1.577721810442023610823457130565572459346e-30
100 7.888609052210118054117285652827862296732e-31
101 3.944304526105059027058642826413931148366e-31
102 1.972152263052529513529321413206965574183e-31
103 9.860761315262647567646607066034827870915e-32
104 4.930380657631323783823303533017413935458e-32
105 2.465190328815661891911651766508706967729e-32
106 1.232595164407830945955825883254353483864e-32
107 6.162975822039154729779129416271767419322e-33
108 3.081487911019577364889564708135883709661e-33
109 1.54074395550978868244478235406794185483e-33
110 7.703719777548943412223911770339709274152e-34
111 3.851859888774471706111955885169854637076e-34
112 1.925929944387235853055977942584927318538e-34
113 9.629649721936179265279889712924636592691e-35
114 4.814824860968089632639944856462318296345e-35
115 2.407412430484044816319972428231159148173e-35
116 1.203706215242022408159986214115579574086e-35
117 6.018531076210112040799931070577897870432e-36
118 3.009265538105056020399965535288948935216e-36
119 1.504632769052528010199982767644474467608e-36
120 7.523163845262640050999913838222372338039e-37
121 3.7615819226313200254999569191118616902e-37
122 1.88079096131566001274997845955559308451e-37
123 9.403954806578300063749892297777965422549e-38
124 4.701977403289150031874946148888982711275e-38
125 2.350988701644575015937473074444491355637e-38
126 1.175494350822287507968736537222245677819e-38
127 5.877471754111437539843682686111228389093e-39
128 2.938735877055718769921841343055614194547e-39
129 1.469367938527859384960920671527807097273e-39
130 7.346839692639296924804603357639035486367e-40

```
131 3.673419846319648462402301678819517743183e-40
132 1.836709923159824231201150839409758871592e-40
133 9.183549615799121156005754197048794357958e-41
134 4.591774807899560578002877098524397178979e-41
135 2.29588740394978028900143854926219858949e-41
136 1.147943701974890144500719274631099294745e-41
Вычисitanное значение машинного эпсилон = 1.147943701974890144500719274631099294745e-41
Теоретическое значение эпсилон = 1e-40
```

In []: