# Численные методы. Лабораторная работа №1.

ПИН-21 Чендемеров Алексей

March 27, 2021

[1]:
```
import numpy as np
```

### 0.1 Задание 1

Зададим коэффициенты для полинома

$$(x-1)(x-2)....(x-20)=0$$

[2]:
```
coefs = np.array(range(1, 21))
print(coefs)
p = np.poly(coefs)
print(p)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
[ 1.00000000e+00 -2.10000000e+02  2.06150000e+04 -1.25685000e+06
  5.33279460e+07 -1.67228082e+09  4.01717716e+10 -7.56111184e+11
  1.13102770e+13 -1.35585183e+14  1.30753501e+15 -1.01422999e+16
  6.30308121e+16 -3.11333643e+17  1.20664780e+18 -3.59997952e+18
  8.03781182e+18 -1.28709312e+19  1.38037598e+19 -8.75294804e+18
  2.43290201e+18]
```

Найдём корни (корни действительные)

[3]:
```
np.roots(p)
```

[3]:
```
array([20.00003842, 18.99944526, 18.00332672, 16.98821521, 16.02622951,
       14.95636365, 14.05068644, 12.9559752 , 12.0299548 , 10.98552896,
       10.00542601,  8.9985865 ,  8.0002459 ,  6.9999773 ,  5.99999991,
        5.00000023,  3.99999998,  3.        ,  2.        ,  1.        ])
```

Изменим один из коэффициентов на $10^{-7}$ и заново найдём корни (стали комплексными).

[4]:
```
p[1] += 1e-7
np.roots(p)
```

[4]:
```
array([20.42191428+0.99919021j, 20.42191428-0.99919021j,
       18.15709998+2.47016309j, 18.15709998-2.47016309j,
       15.31451742+2.69846576j, 15.31451742-2.69846576j,
       12.84588825+2.06157319j, 12.84588825-2.06157319j,
       10.92152839+1.10102893j, 10.92152839-1.10102893j,
```

```
9.57172741+0.j        ,  9.11219077+0.j        ,
7.99387715+0.j        ,  7.00032223+0.j        ,
5.99998503+0.j        ,  5.00000068+0.j        ,
3.99999999+0.j        ,  3.        +0.j        ,
2.        +0.j        ,  1.        +0.j        ])
```

Попробуем закастовать $2^{1023}$ в `float` и потом вывести его в научном формате (получилось).

[5]:
```
print('{:e}'.format(2**1023))
```

```
8.988466e+307
```

Теперь попробуем закастовать $2^{1024}$ (не получилось, т.к. размер порядка ограничен 1024)

[6]:
```
print('{:e}'.format(2**1024))
```

```
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
<ipython-input-6-f24da7173b90> in <module>
----> 1 print('{:e}'.format(2**1024))

OverflowError: int too large to convert to float
```

Максимальное целое число

[7]:
```
np.finfo('d').max
```

[7]:
```
1.7976931348623157e+308
```

Минимальное целое число

[8]:
```
np.finfo('d').min
```

[8]:
```
-1.7976931348623157e+308
```

16 символов после запятой

[9]:
```
np.sqrt(2)
```

[9]:
```
1.4142135623730951
```

Ошибки не происходит, скорее всего, из-за того, что Python поддерживает длинную арифметику или из-за реализации функций в numpy

[10]:
```
np.float_power(10, 8) + np.float_power(10,-7)
```

[10]:
```
100000000.0000001
```

[11]:
```
np.float_power(10, 8) + np.float_power(10,-8)
```

[11]:
```
100000000.00000001
```

Считал для $10^{11}$ и $10^{10}$ и ошибки не происходило (скорее всего из-за длинной арифметики или внутреннего представления `float` в Python) [ Хотя CPython явно представляет float как 64 битное число.]

[12]:
```
summ = 1
for i in range(int(1e11)):
    summ += 1e-10
print(summ)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-12-712195d6f226> in <module>
      1 summ = 1
      2 for i in range(int(1e11)):
----> 3     summ += 1e-10
      4 print(summ)

KeyboardInterrupt:
```

При $n = 53$ происходит ошибка

[16]:
```
print('n\t\teps\t\t\t\t\t\tresult')
for i in range(0, 60):
    eps = 2 ** -i
    print(f'{i:<5}\t\t{eps:<25}\t\t\t{(1+eps-1)/eps:<25}')
```

```
n               eps                                             result
0               1                                               1.0
1               0.5                                             1.0
2               0.25                                            1.0
3               0.125                                           1.0
4               0.0625                                          1.0
5               0.03125                                         1.0
6               0.015625                                        1.0
7               0.0078125                                       1.0
8               0.00390625                                      1.0
9               0.001953125                                     1.0
10              0.0009765625                                    1.0
11              0.00048828125                                   1.0
12              0.000244140625                                  1.0
13              0.0001220703125                                 1.0
14              6.103515625e-05                                 1.0
15              3.0517578125e-05                                1.0
16              1.52587890625e-05                               1.0
17              7.62939453125e-06                               1.0
18              3.814697265625e-06                              1.0
```

```
19              1.9073486328125e-06             1.0
20              9.5367431640625e-07             1.0
21              4.7683715820312e-07             1.0
22              2.384185791015625e-07           1.0
23              1.1920928955078125e-07          1.0
24              5.960464477539063e-08           1.0
25              2.9802322387695312e-08          1.0
26              1.4901161193847656e-08          1.0
27              7.450580596923828e-09           1.0
28              3.725290298461914e-09           1.0
29              1.862645149230957e-09           1.0
30              9.313225746154785e-10           1.0
31              4.656612873077393e-10           1.0
32              2.3283064365386963e-10          1.0
33              1.1641532182693481e-10          1.0
34              5.820766091346741e-11           1.0
35              2.9103830456733704e-11          1.0
36              1.4551915228366852e-11          1.0
37              7.275957614183426e-12           1.0
38              3.637978807091713e-12           1.0
39              1.8189894035458565e-12          1.0
40              9.094947017729282e-13           1.0
41              4.547473508864641e-13           1.0
42              2.2737367544323206e-13          1.0
43              1.1368683772161603e-13          1.0
44              5.684341886080802e-14           1.0
45              2.842170943040401e-14           1.0
46              1.4210854715202004e-14          1.0
47              7.105427357601002e-15           1.0
48              3.552713678800501e-15           1.0
49              1.7763568394002505e-15          1.0
50              8.881784197001252e-16           1.0
51              4.440892098500626e-16           1.0
52              2.220446049250313e-16           1.0
53              1.1102230246251565e-16          0.0
54              5.551115123125783e-17           0.0
55              2.7755575615628914e-17          0.0
56              1.3877787807814457e-17          0.0
57              6.938893903907228e-18           0.0
58              3.469446951953614e-18           0.0
59              1.734723475976807e-18           0.0
```

Из-за переполнения ответ становится неправильным – обычная практика в олимпиадном программировании, например.

[15]:
```
I = 1 / np.exp(1)
print('n\t\tI')
print(f'{1:<5}\t\t{I:<25}')
```

```
for n in range(2, 31):
    I = 1 - n * I
    print(f'{n:<5}\t\t{I:<25}')
```

```
n               I
1               0.36787944117144233
2               0.26424111765711533
3               0.207276647028654
4               0.17089341188538398
5               0.14553294057308008
6               0.1268023565615195
7               0.11238350406936348
8               0.10093196744509214
9               0.09161229299417073
10              0.0838770700582927
11              0.07735222935878028
12              0.07177324769463667
13              0.06694777996972334
14              0.06273108042387321
15              0.059033793641901866
16              0.05545930172957014
17              0.05719187059730757
18              -0.029453670751536265
19              1.559619744279189
20              -30.19239488558378
21              635.0402925972594
22              -13969.886437139707
23              321308.38805421325
24              -7711400.313301118
25              192785008.83252797
26              -5012410228.645727
27              135335076174.43463
28              -3789382132883.17
29              109892081853612.92
30              -3296762455608386.5
```

Находит правильно, если учитывать погрешностии округлять результат (например с $\epsilon = 10^{-16}$)

[14]:
```
def mysin(x):
    u = x
    i = 1
    flag = False
    summ = u
    while True:
        u = u * (- (x ** 2) / ((2*(i-1)+2)*(2*(i-1)+3)))
        i += 1
```

```
        summ += u
        if flag:
            break
        if np.abs(u) < 1e-17:
            flag = True
    return summ

points = [0, np.pi / 3, np.pi / 2, np.pi, np.pi*2]
sins = [mysin(x) for x in points]
print(points)
print(sins)
```

```
[0, 1.0471975511965976, 1.5707963267948966, 3.141592653589793,
6.283185307179586]
[0.0, 0.8660254037844385, 1.0000000000000002, 2.4790609271195177e-16,
4.3878932039122245e-16]
```

Опять происходит переполнение для больших входных $x$, поэтому в реальности все подобные алгоритмы приводят входные данные к какому-то ограниченному диапазону (очевидно, что для тригонометрических функций достаточно отрезка от $\frac{-\pi}{2}$ до $\frac{\pi}{2}$или от $0$ до $\pi$)

[13]:
```
def mysin_print(x):
    u = x
    print('n\t\tu')
    print(f'{0:<5}\t\t{u:<25}')
    i = 1
    flag = False
    summ = u
    while True:
        u = u * (- (x ** 2) / ((2*(i-1)+2)*(2*(i-1)+3)))
        print(f'{i:<5}\t\t{u:<25}')
        i += 1
        summ += u
        if flag:
            break
        if np.abs(u) < 1e-17:
            flag = True
    return summ

x = []
x.append(mysin_print(np.pi*12))
print()
x.append(mysin_print(np.pi*13))
print()
x.append(mysin_print(np.pi*14))
print()
print(x)
```

```
n               u
```

| | |
|---|---|
| 0 | 37.69911184307752 |
| 1 | -8929.807683926347 |
| 2 | 634562.4183707595 |
| 3 | -21472731.555832986 |
| 4 | 423854731.73371917 |
| 5 | -5476291888.243632 |
| 6 | 49891231866.331665 |
| 7 | -337650323385.4057 |
| 8 | 1764251532907.3298 |
| 9 | -7331564081604.561 |
| 10 | 24809018443430.754 |
| 11 | -69682111574511.336 |
| 12 | 165056370017519.22 |
| 13 | -334162271990305.9 |
| 14 | 584875761040825.1 |
| 15 | -893805272556234.8 |
| 16 | 1202932425236984.8 |
| 17 | -1436668294789759.5 |
| 18 | 1532902456774330.0 |
| 19 | -1470037975755849.5 |
| 20 | 1273934043683861.5 |
| 21 | -1002516282597193.5 |
| 22 | 719595571991636.6 |
| 23 | -473036911148918.7 |
| 24 | 285837990621617.4 |
| 25 | -159309622036949.84 |
| 26 | 82153303460817.77 |
| 27 | -39312514201256.56 |
| 28 | 17503712624598.932 |
| 29 | -7269631665207.737 |
| 30 | 2822887423366.538 |
| 31 | -1027125608753.4813 |
| 32 | 350907349452.402 |
| 33 | -112781005824.58708 |
| 34 | 34161756873.01164 |
| 35 | -9768908599.904654 |
| 36 | 2641514063.327634 |
| 37 | -676428942.4853929 |
| 38 | 164278262.77512643 |
| 39 | -37889654.49554902 |
| 40 | 8310131.128111686 |
| 41 | -1735314.3950649824 |
| 42 | 345415.79678938625 |
| 43 | -65612.52160125064 |
| 44 | 11906.285368049981 |
| 45 | -2066.1156302264953 |
| 46 | 343.1990561106805 |
| 47 | -54.62064991132304 |

| | |
|---|---|
| 48 | 8.336353712708581 |
| 49 | -1.2211727390276248 |
| 50 | 0.17183750741604203 |
| 51 | -0.023245709461550114 |
| 52 | 0.003025397227360346 |
| 53 | -0.0003791010602881931 |
| 54 | 4.576853202541888e-05 |
| 55 | -5.327378536917623e-06 |
| 56 | 5.982453449912522e-07 |
| 57 | -6.485431457966371e-08 |
| 58 | 6.791367942759498e-09 |
| 59 | -6.873699295661437e-10 |
| 60 | 6.728002593741318e-11 |
| 61 | -6.3721126598701246e-12 |
| 62 | 5.842705345742685e-13 |
| 63 | -5.1892184832045295e-14 |
| 64 | 4.466470951748517e-15 |
| 65 | -3.727452375943013e-16 |
| 66 | 3.01751035197196e-17 |
| 67 | -2.370677289564589e-18 |
| 68 | 1.808319648740604e-19 |

| n | u |
|---|---|
| 0 | 40.840704496667314 |
| 1 | -11353.464977769787 |
| 2 | 946858.0568581794 |
| 3 | -37602960.50558273 |
| 4 | 871116003.0622387 |
| 5 | -13208994427.893934 |
| 6 | 141231512001.13235 |
| 7 | -1121756936946.623 |
| 8 | 6878857452614.389 |
| 9 | -33548773983349.1 |
| 10 | 133233615531831.55 |
| 11 | -439187273162442.8 |
| 12 | 1220913641423325.2 |
| 13 | -2900910193215591.0 |
| 14 | 5958880893733033.0 |
| 15 | -1.068730506338276e+16 |
| 16 | 1.6880711128882252e+16 |
| 17 | -2.3660843700707324e+16 |
| 18 | 2.9628690122834868e+16 |
| 19 | -3.334653382151728e+16 |
| 20 | 3.39151154800249e+16 |
| 21 | -3.132290290025647e+16 |
| 22 | 2.638658969391714e+16 |
| 23 | -2.0357011609439044e+16 |
| 24 | 1.4436541276415564e+16 |

| | |
|---|---|
| 25 | -9442987754031000.0 |
| 26 | 5715005638943512.0 |
| 27 | -3209568610193969.0 |
| 28 | 1677143530466761.5 |
| 29 | -817479133738897.4 |
| 30 | 372547832210119.0 |
| 31 | -159087571280886.38 |
| 32 | 63786587867943.414 |
| 33 | -24060080875503.332 |
| 34 | 8553138989088.722 |
| 35 | -2870487041743.0176 |
| 36 | 910933521753.6548 |
| 37 | -273766403733.79266 |
| 38 | 78030121571.13542 |
| 39 | -21121610984.363304 |
| 40 | 5436738990.627072 |
| 41 | -1332394983.6526499 |
| 42 | 311258504.99936146 |
| 43 | -69388895.28578915 |
| 44 | 14777594.474539159 |
| 45 | -3009582.7762294146 |
| 46 | 586707.941668782 |
| 47 | -109586.4751253036 |
| 48 | 19629.102401870667 |
| 49 | -3374.6257835378537 |
| 50 | 557.302121881652 |
| 51 | -88.47890721980497 |
| 52 | 13.514611377740636 |
| 53 | -1.9874690249194729 |
| 54 | 0.28160253847929045 |
| 55 | -0.038468685944267056 |
| 56 | 0.005069875975414267 |
| 57 | -0.000645031752139395 |
| 58 | 7.927270771728507e-05 |
| 59 | -9.416319240877441e-06 |
| 60 | 1.0816854988904039e-06 |
| 61 | -1.20232676617014e-07 |
| 62 | 1.2938301501657544e-08 |
| 63 | -1.3486195505518861e-09 |
| 64 | 1.3623108680397445e-10 |
| 65 | -1.3342832168331302e-11 |
| 66 | 1.26767784748653e-12 |
| 67 | -1.16884462564903e-13 |
| 68 | 1.0463663355478251e-14 |
| 69 | -9.098636652018564e-16 |
| 70 | 7.688039814716382e-17 |
| 71 | -6.3150630645581e-18 |
| 72 | 5.044680288484288e-19 |

| n | u |
|---|---|
| 0 | 43.982297150257104 |
| 1 | -14180.203868457116 |
| 2 | 1371539.4245829931 |
| 3 | -63170573.85861541 |
| 4 | 1697220006.385679 |
| 5 | -29847040443.17852 |
| 6 | 370111425747.61163 |
| 7 | -3409329799355.2793 |
| 8 | 24246883576935.53 |
| 9 | -137146787068041.94 |
| 10 | 631672782179605.5 |
| 11 | -2414890617243964.5 |
| 12 | 7785778254272807.0 |
| 13 | -2.1454615469456176e+16 |
| 14 | 5.111172313197143e+16 |
| 15 | -1.063147178105709e+17 |
| 16 | 1.9475350808100528e+17 |
| 17 | -3.165877779620511e+17 |
| 18 | 4.597754060317196e+17 |
| 19 | -6.00141071992664e+17 |
| 20 | 7.078892519640264e+17 |
| 21 | -7.582342346771482e+17 |
| 22 | 7.407881313973412e+17 |
| 23 | -6.628177692761962e+17 |
| 24 | 5.4514576439738144e+17 |
| 25 | -4.135502411624303e+17 |
| 26 | 2.902718239944355e+17 |
| 27 | -1.8906200068522304e+17 |
| 28 | 1.1457693051132858e+17 |
| 29 | -6.476986546377322e+16 |
| 30 | 3.4233218046690564e+16 |
| 31 | -1.6953965852899512e+16 |
| 32 | 7883767176814510.0 |
| 33 | -3448822702892051.5 |
| 34 | 1421898781436465.5 |
| 35 | -553436897454549.5 |
| 36 | 203689466288653.28 |
| 37 | -70995595094748.62 |
| 38 | 23468368730313.99 |
| 39 | -7367447095129.922 |
| 40 | 2199367670042.8867 |
| 41 | -625117574468.1617 |
| 42 | 169363302545.8232 |
| 43 | -43788233634.46938 |
| 44 | 10815349655.944696 |
| 45 | -2554538659.6424766 |

```
46              577560548.8078797
47              -125112838.78435919
48              25990505.58017433
49              -5182141.581022749
50              992530.1704559936
51              -182751.99953884477
52              32373.921981269843
53              -5521.556106674785
54              907.3337234501068
55              -143.74978561860064
56              21.97184649906847
57              -3.242049797850811
58              0.4620954019208789
59              -0.06365880695443715
60              0.008481012347932486
61              -0.0010932980415693815
62              0.0001364465907034843
63              -1.6494693096842047e-05
64              1.9324148942774502e-06
65              -2.1950354826054636e-07
66              2.418643110330005e-08
67              -2.5863603839303394e-09
68              2.685254052327308e-10
69              -2.7079915867620456e-11
70              2.6537253869466307e-12
71              -2.5280602150226737e-13
72              2.3421393812182525e-14
73              -2.111049236971358e-15
74              1.851851661828974e-16
75              -1.5815896199131873e-17
76              1.315571946675627e-18
77              -1.0661492401644133e-19

[-0.26048840103934273, -2.9021751309903294, -83.11717551225941]
```