Computational Physics
Homework 1

Yi-Hsuan Hsu

09/11/2014

1 Problem 1

Calculating integration using recursion relationship.

$$I_n = \frac{1}{n} - 5I_{n-1} \tag{1}$$

1.1 Algorithms

Single-precise(32 bit) floating point number can be assigned by np.float32(). Double-precise floating point number are default in Python. Put initial single and double precise number into simple loop recursion function. Stop at given number of N (here N=30) and return final result into array.

Exact solution could be solved by Python integration function integrate.quad(). Integrate.quad would return integration and estimating error.

1.2 Description of program

It's a simple practice for new python user as I am. Well organized structure and comment keep it clear to review. Variable are well named as it represent.

1.3 Sample output

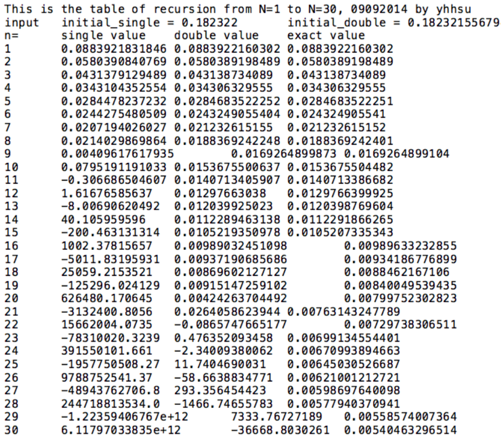
Error analysis as below

$$I_{exact} = I_{approx.} + \epsilon \tag{2}$$

$$I_N = \tfrac{1}{N} - 5 * I_{N-1} \pm 5^N * \epsilon \tag{3}$$

Where we can estimate the error of numerical approximation.

Single-precise input return good approach till N=7, error become bigger and bigger after. Interesting things is that N=11 return the first negative number, which is the same order as professor's Example 2. Afterward, the number jump positive and negative term by term, which is expected from the recursion equation. Finally, N=30 returns error up to tenth of twelve, roughly consists with (6). On the last two columns shows the error compare to exact number, which also confirmed that error multiple 5 at each recursion.



```
This is the table of recursion from N=1 to N=30, 09092014 by yhhsu
input   initial_single = 0.182322       initial_double = 0.18232155679
n=      single value    double value    exact value
1       0.0883921831846 0.0883922160302 0.0883922160302
2       0.0580390840769 0.0580389198489 0.0580389198489
3       0.0431379129489 0.043138734089  0.043138734089
4       0.0343104352554 0.034306329555  0.034306329555
5       0.0284478237232 0.0284683522252 0.0284683522251
6       0.0244275480509 0.0243249055404 0.024324905541
7       0.0207194026027 0.021232615155  0.021232615152
8       0.0214029869864 0.0188369242248 0.0188369242401
9       0.00409617617935        0.0169264899873 0.0169264899104
10      0.0795191191033 0.0153675500637 0.0153675504482
11      -0.306686504607 0.0140713405907 0.0140713386682
12      1.61676585637   0.0129766303 8  0.0129766399925
13      -8.00690620492  0.012039925023  0.0120398769604
14      40.105959596    0.0112289463138 0.0112291866265
15      -200.463131314  0.0105219350978 0.0105207335343
16      1002.37815657   0.00989032451098        0.00989633232855
17      -5011.83195931  0.00937190685686        0.00934186776899
18      25059.2153521   0.00869602127127        0.0088462167106
19      -125296.024129  0.00915147259102        0.00840049539435
20      626480.170645   0.00424263704492        0.00799752302823
21      -3132400.8056   0.0264058623944 0.00763143247789
22      15662004.0735   -0.0865747665177        0.00729738306511
23      -78310020.3239  0.476352093458  0.00699134554401
24      391550101.661   -2.34009380062  0.00670993894663
25      -1957750508.27  11.7404690031   0.00645030526687
26      9788752541.37   -58.6638834771  0.00621001212721
27      -48943762706.8  293.356454423   0.00598697640098
28      244718813534.0  -1466.74655783  0.00577940370941
29      -1.22359406767e+12      7333.76727189   0.00558574007364
30      6.11797033835e+12       -36668.8030261  0.00540463296514
```

Figure 1: Result table snapshot.

Double-precise input, in the other hand, performs well till N=18. Obviously performs much better than single precise floating point number.

The build-in integration function gives the same number as professor's example from Mathematica. Numerical computation returns error within ten powered sixteenth precision.

2 Problem 2

Find numerical approximation of the bridge height to 5 significant digits.

2.1 Theory

Method.1

Consider the upper fan-shaped area as a triangle. Therefore, the bridge height d simply equal square root of arc square minus distance square.

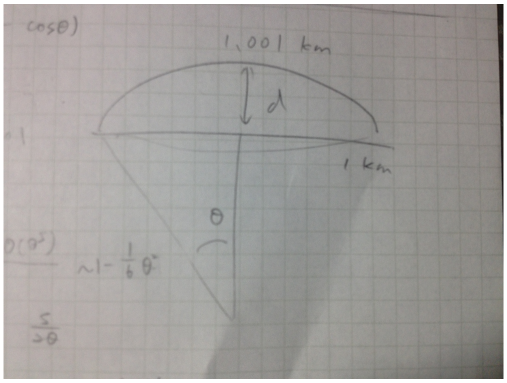


Figure 2: Picture and variables.

$$d = \sqrt{0.5005^2 - 0.5^2} = 0.022366 \tag{4}$$

Method.2

$$d = r - rcos(\theta) = r(1 - cos(\theta)) \tag{5}$$

$$L = 2rsin(\theta) \tag{6}$$

$$S = 2r\theta \tag{7}$$

From (2) and (3), we have

$$\frac{L}{S} = \frac{sin(\theta)}{\theta} \tag{8}$$

Using Taylor expansion of Sin,

$$Sin\theta = \theta - \frac{1}{6}\theta^3 + \varepsilon(\theta^5) \tag{9}$$

Ignoring the error term, (8) can be rearrange to

$$\theta = \sqrt{6(1 - \frac{L}{S})} \tag{10}$$

Finally, Combining (5)(7)(10) we have

$$d = \frac{S}{\theta}(1 - cos(\theta)) = 0.019364 \tag{11}$$

2.2 Discussion

Obviously, Method.1 do not approach to the answer very well. While Method.2 using Taylor expansion, the error term depends on theta powered five. In this case, theta is as small as 0.077421. Therefore I could expect that the result is within 5 significant digit precision.