Exercise No. 1

David Bubeck, Pascal Becht, Patrick Nisblè

April 28, 2017

2 - Numerical Integration

We are to use

$$y_n(a) = \int_0^1 \left(\frac{x^n}{x-a} \right) \text{dx} = \frac{1}{n} - a \cdot y_{n-1}(a) \tag{1}$$

for the following tasks

a)

Plotting the Integrand for $n \in \{1, 5, 10, 20, 30, 50\}$ and $x \in [0, 1]$

Listing 1: 01-2a.py

```
import numpy as np
import matplotlib.pyplot as plt

a = 5
n = np.array([1,5,10,20,30,50])
xdata = np.linspace(0,1, 1000)

y = lambda x, a, n: x**n/(x+a)


plt.figure(figsize=(10,5))

plt.xlabel(’x’)
plt.ylabel(’y’)

for it in n:
    plt.plot(xdata, y(xdata, a, it), label=”n={}”.format(it))
```

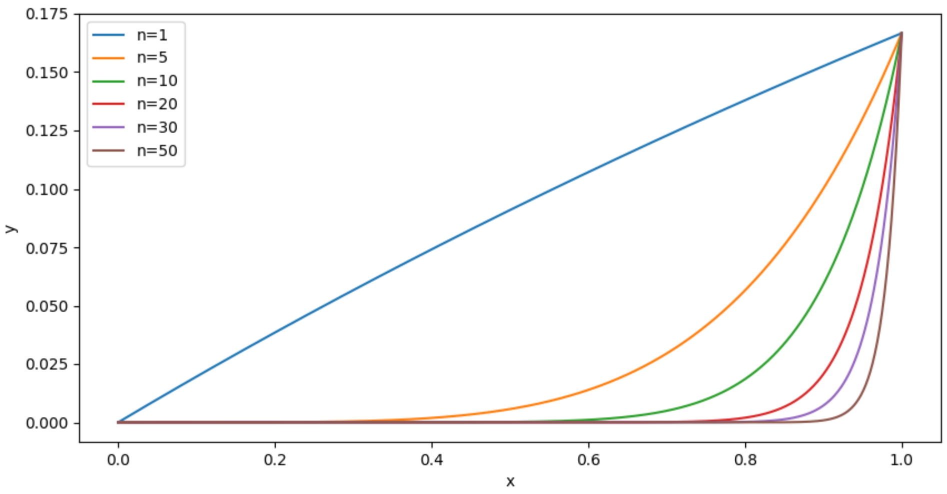


Figure 1: plotted integrand for given n and x interval

b)

Now for the Integration steps: iterate over a list between n_0 and n_1 starting at the lower, using a given a and y_0

Listing 2: 01-2b.py

```python
import numpy as np
import sys
import pandas as pd

def get_nset(n0,n1):
    if n0 > n1:
        n = np.arange(n1, n0+1)
    elif n1 > n0:
        n = np.arange(n0, n1+1)
    else:
        n = []
        print('no iteration possible')

    return n

def iteration(a, y0, n):

    y = [y0]
    for i in n:
        y1 = 1/(i+1) - a * y[-1]
        y.append(y1)

    return y


if __name__ == '__main__':

    # accepting args as: a, n0, y0, n1

    print(sys.argv)

    a = float(sys.argv[1])
    n0 = int(sys.argv[2])
    n1 = int(sys.argv[4])
    y0 = float(sys.argv[3])

    ndata = get_nset(n0,n1)
    ydata = np.array(iteration(a, y0, ndata))

    ndata = np.append([0],ndata)
    print(ydata.__class__)


    f = open("01-2b.tex", 'w')
    f.write(
        pd.DataFrame([ndata, ydata],
        index=['n',"$y_n(" +str(a)+ ")$"]
        ).transpose().to_latex(escape=False))
    f.close()
```

	n	$y_n(5.0)$
0	0.0	1.000000e+01
1	10.0	-4.990909e+01
2	11.0	2.496288e+02
3	12.0	-1.248067e+03
4	13.0	6.240407e+03
5	14.0	-3.120197e+04
6	15.0	1.560099e+05
7	16.0	-7.800494e+05
8	17.0	3.900247e+06
9	18.0	-1.950124e+07
10	19.0	9.750618e+07
11	20.0	-4.875309e+08
12	21.0	2.437654e+09
13	22.0	-1.218827e+10
14	23.0	6.094136e+10
15	24.0	-3.047068e+11
16	25.0	1.523534e+12
17	26.0	-7.617670e+12
18	27.0	3.808835e+13
19	28.0	-1.904418e+14
20	29.0	9.522088e+14
21	30.0	-4.761044e+15

Figure 2: output of 01-2b.py

c)

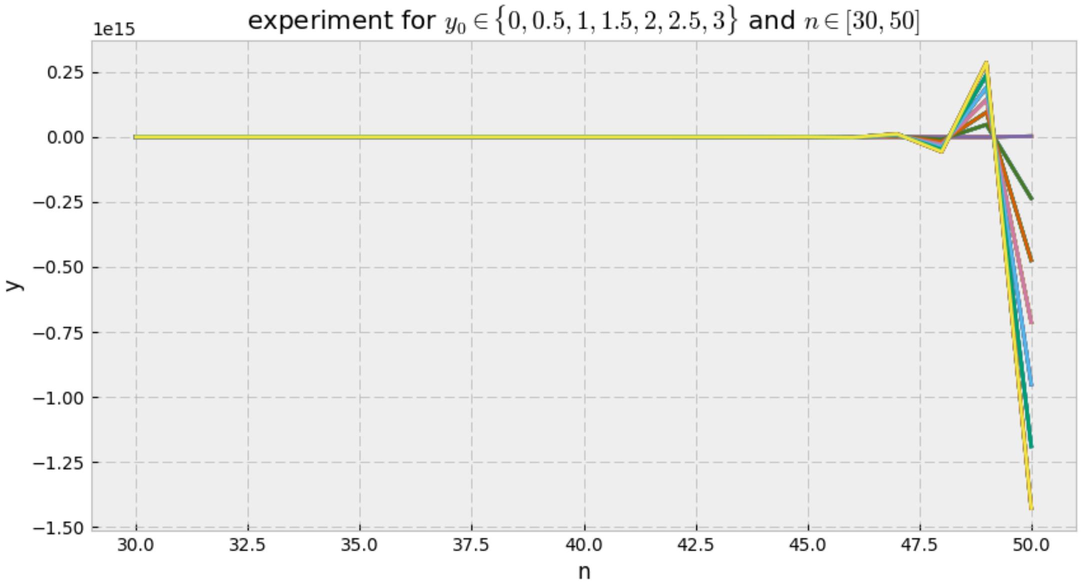
repeat b) with given values

Listing 3: 01-2c.py

```python
#! /usr/bin/python3

import sys
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
b = __import__('01-2b')

if __name__ == '__main__':

    a = 5
    n0 = 0
    n1 = 30
    y0 = np.log((1+a)/a)

    ndata = b.get_nset(n0,n1)
    ydata = np.array(b.iteration(a,y0,ndata))

    ndata = np.append([0],ndata)
    f = open("01-2c.tex", 'w')
    f.write(pd.DataFrame(
        [ndata, ydata],
        index=['n',"$y_n(" +str(a)+ ")$"]
    ).transpose().to_latex(escape=False))
    f.close()

    n0 = 50
    n1 = 30
    y0 = np.linspace(0,3,7)


    ndata = b.get_nset(n0,n1)
    print(ndata)

    plt.style.use('bmh')
    plt.figure(figsize=(10,5))
    for y in y0:
        ydata = np.array(b.iteration(a,y0,ndata))[1:]
        plt.plot(ndata,ydata,label="$y_0=${}".format(y))


    plt.title("experiment for $y_0\in \{0,0.5,1,1.5,2,2.5,3\}$ and $n \in [30,50]$")
    plt.ylabel('y')
    plt.xlabel('n')
    plt.savefig('01-2c.png')
```

	n	$y_n(5)$
0	0.0	0.182322
1	0.0	0.088392
2	1.0	0.058039
3	2.0	0.043139
4	3.0	0.034306
5	4.0	0.028468
6	5.0	0.024325
7	6.0	0.021233
8	7.0	0.018837
9	8.0	0.016926
10	9.0	0.015368
11	10.0	0.014071
12	11.0	0.012977
13	12.0	0.012040
14	13.0	0.011229
15	14.0	0.010522
16	15.0	0.009890
17	16.0	0.009372
18	17.0	0.008696
19	18.0	0.009151
20	19.0	0.004243
21	20.0	0.026406
22	21.0	-0.086575
23	22.0	0.476352
24	23.0	-2.340094
25	24.0	11.740469
26	25.0	-58.663883
27	26.0	293.356454
28	27.0	-1466.746558
29	28.0	7333.767272
30	29.0	-36668.803026
31	30.0	183344.047389

Figure 3: output of 01-2c.py


experiment for $y_0 \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$ and $n \in [30, 50]$
1e15
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
y
30.0
32.5
35.0
37.5
40.0
42.5
45.0
47.5
50.0
n