ex01

April 27, 2023

1 Numerical Integration

1.1 Paris J. Huth: Gruppe 1

1.2 Q' inich Figueroa Coc: Gruppe 5

2 Numerical Integration

Basic imports for calculations

[ ]:
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

2.0.1 a)

Defining function we're gonna use.

[ ]:
```
def inte(x,a,n):
    return x**n/(x+a)
```

Assigning variables.

[ ]:
```
a = 5
x = np.linspace(0,1,100)
n = np.array([1,5,10,20,30,50])
```

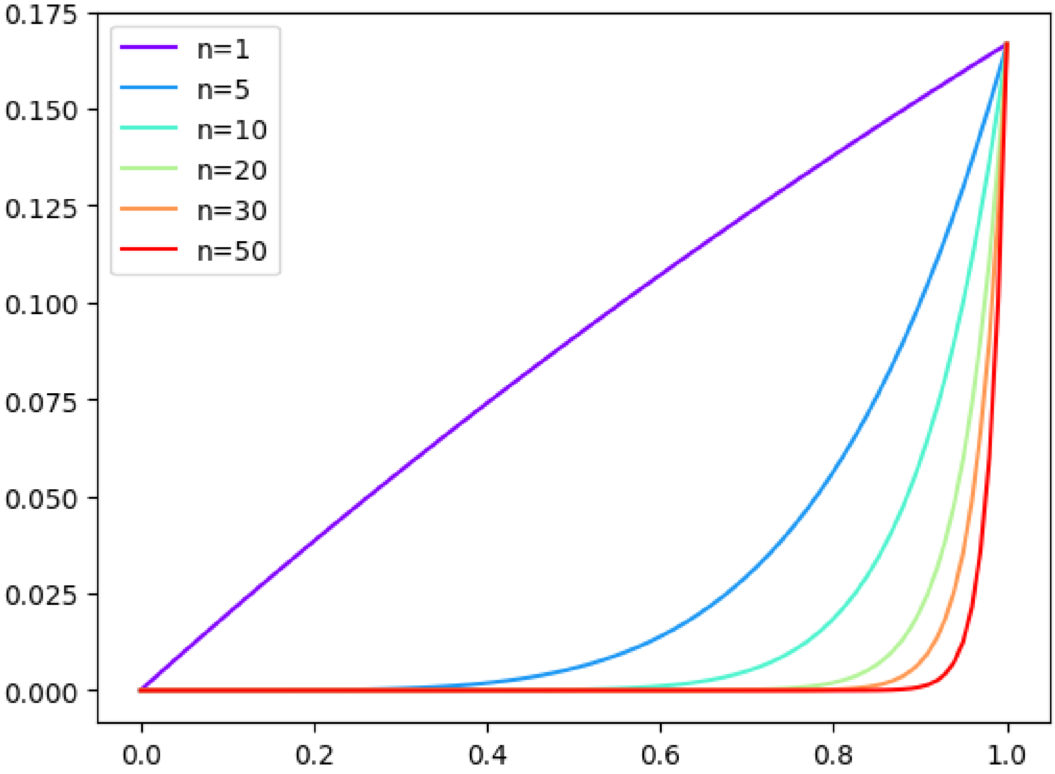
Calculating corresponding values.

[ ]:
```
y = np.array([inte(x,a,i) for i in n])
```

Plotting the different curves.

[ ]:
```
colors = plt.cm.rainbow(np.linspace(0,1,len(n)))
for i in range(len(n)):
    plt.plot(x,y[i],color=colors[i],label='n='+str(n[i]))

plt.legend(loc='upper left')
plt.show()
```



2.0.2 b)

Defining a function, while handling different cases.

[ ]:

```
def it(a,n0,n1,y0):
    y_n = None

    # Non valid cases
    if n0 <= 0:
        print("A boundy of 0 has been givin, which is not define. By defauls it will the value set to 1")
        n0 = 1
    elif n1 <= 0:
        print("A boundy of 0 has been givin, which is not define. By defauls it will the value set to 1")
        n1 = 1

    # Handling said cases
    if n0 < n1:
        un = None
        y_n = np.array([y0])
        for n in range(n0,n1):
            un = 1/n - a*y_n[-1]
            y_n = np.append(y_n, un)
    elif n0 > n1:
        un = None
        y_n = np.array([y0])
        for n in range(n1,n0):
            un = 1/a*(1/n - y_n[-1])
            y_n = np.append(y_n, un)

    # Handling a trivial case
    else:
        y_n = np.array([y0])

    return y_n
```

Introducing a parameter ‘val’ to have the option of printing out the value for each itteration.

[ ]:

```
def iteration(a,n0,n1,y0, val=True):
    a = it(a,n0,n1,y0)
    if val == True:
        print('The last value is y_{} = {}'.format(n1,a[-1]))
    elif val == False:
        if n0<n1:
            for i in range(0,len(a)):
                print('y_{} = {}'.format(n0+i,a[i]))
        elif n0>n1:
            for i in range(0,len(a)):
                print('y_{} = {}'.format(n0- i,a[i]))
```

2.0.3 c)

[ ]:

```
iteration(5,0,30,np.log((1+5)/5),0)
```

```
A boundy of 0 has been givin, which is not define. By defauls it will the value
set to 1
y_0 = 0.1823215567939546
y_1 = 0.08839221603022707
y_2 = 0.05803891984886467
y_3 = 0.04313873408900998
y_4 = 0.03430632955495011
y_5 = 0.02846835222524946
y_6 = 0.024324905540419356
y_7 = 0.02123261515504607
y_8 = 0.018836924224769652
y_9 = 0.016926489987262844
y_10 = 0.015367550063685786
y_11 = 0.01407134059066198
y_12 = 0.012976630380023432
```

```
y_13 = 0.012039925022959766
y_14 = 0.011228946313772595
y_15 = 0.010521935097803692
y_16 = 0.00989032451098154
y_17 = 0.009371906856857001
y_18 = 0.008696021271270546
y_19 = 0.009151472591015689
y_20 = 0.00424263704492156
y_21 = 0.026405862394439816
y_22 = -0.08657476651765363
y_23 = 0.47635209345783336
y_24 = -2.3400938006225003
y_25 = 11.740469003112501
y_26 = -58.66388347710097
y_27 = 293.35645442254184
y_28 = -1466.746557826995
y_29 = 7333.7672718935955
```

[ ]:

```
iteration(5,50,30,np.log((1+5)/5),False)
```

```
y_50 = 0.1823215567939546
y_49 = -0.029797644692124255
y_48 = 0.01241114184165066
y_47 = 0.0037677716316698684
y_46 = 0.005307051734272088
y_45 = 0.004820942594322053
y_44 = 0.004750097195421303
y_43 = 0.004605536116471295
y_42 = 0.004484298182111147
y_41 = 0.004366298258314613
y_40 = 0.004254945476542205
y_39 = 0.004149010904691559
y_38 = 0.0040482465995494935
y_37 = 0.0039522554419948635
y_36 = 0.003860711702298702
y_35 = 0.0037733122049948054
y_34 = 0.0036897820034454834
y_33 = 0.003609869686267425
y_32 = 0.003533345211682685
y_31 = 0.0034599976243301295
y_30 = 0.0033896331281951988
```

changing the y0 shows that independat from this, the backwards iteration from 50 to 30 converges into 0.0033