

A6 Python Code

Kang, Eui Hyeon

2021-01-23

차 례

Implementation - basic	2
Implementation - 'The first-timer would write'	3
Running estimate approach - Implementation	4
Convergence trajectory	5

Implementation - basic

```
np.random.seed(1234)

MC_N=10**3

x=np.random.uniform(0,1,size=MC_N)*2-1
y=np.random.uniform(0,1,size=MC_N)*2-1

t=np.sqrt(x**2+y**2)

pi_hat=4*sum(t<=1)/MC_N
pi_hat
```

```
## 3.06
```

Implementation - 'The first-timer would write'

```
np.random.seed(1234)

MC_N=10**6
count=0

for MC_i in range(MC_N):
    x_i=np.random.uniform(0,1)*2-1
    y_i=np.random.uniform(0,1)*2-1
    t_i=np.sqrt(x_i**2+y_i**2)

    if t_i <= 1 :
        count+=1

pi_hat=4*count/MC_N
pi_hat

## 3.140204
```

Running estimate approach - Implementation

```
np.random.seed(1234)
beg_time=time.time()

old_est=0
n=1
MC_N=10**6

while True:
    x_i=np.random.uniform(0,1)*2-1
    y_i=np.random.uniform(0,1)*2-1
    t_i=np.sqrt(x_i**2+y_i**2)

    A_n=4*(t_i<=1)

    new_est=((n-1)/n)*old_est + (1/n)*A_n

    if n > MC_N:
        break

    n+=1
    old_est=new_est

print(new_est)

## 3.1402048597951002

end_time=time.time()

print('Time difference of ',end_time-beg_time,' secs')

## Time difference of 15.237017393112183 secs
```

Convergence trajectory

```
np.random.seed(1234)
```

```
#beg_time=time.time()
```

```
old_est=0
```

```
n=1
```

```
MC_N=10**6
```

```
results=np.repeat(0,MC_N+1).astype('float')
```

```
while True:
```

```
    x_i=np.random.uniform(0,1)*2-1
```

```
    y_i=np.random.uniform(0,1)*2-1
```

```
    t_i=np.sqrt(x_i**2+y_i**2)
```

```
    A_n=4*(t_i<=1)
```

```
    new_est=((n-1)/n)*old_est + (1/n)*A_n
```

```
    results[n]=new_est # to save
```

```
    if n >= MC_N :
```

```
        break
```

```
    n+=1
```

```
    old_est=new_est
```

```
plt.plot(results[1:21],marker='o')
```

```
plt.title('Estimate '+r'$\pi$'+ ' trying 20 times',fontweight='bold')
```

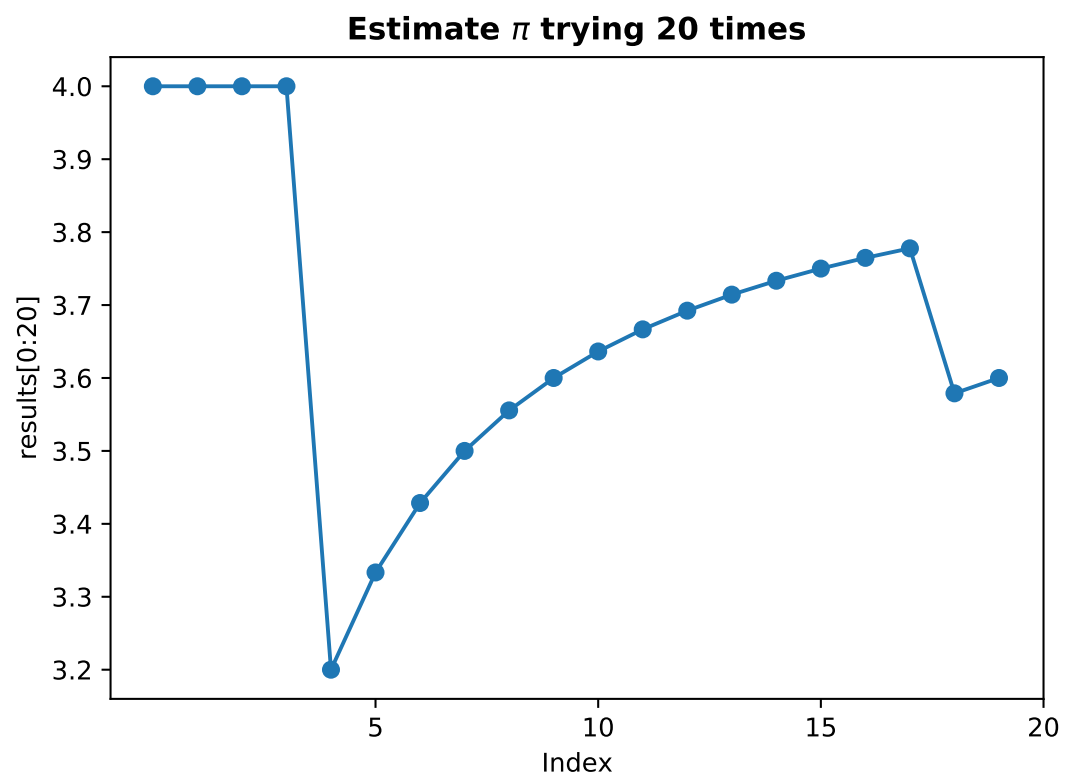
```
plt.xlabel('Index')
```

```
plt.ylabel('results[0:20]')
```

```
plt.xticks([5,10,15,20])
```

```
## ([<matplotlib.axis.XTick object at 0x000000002C90C128>, <matplotlib.axis.XTick object at 0x000000002C8FECC0>,
```

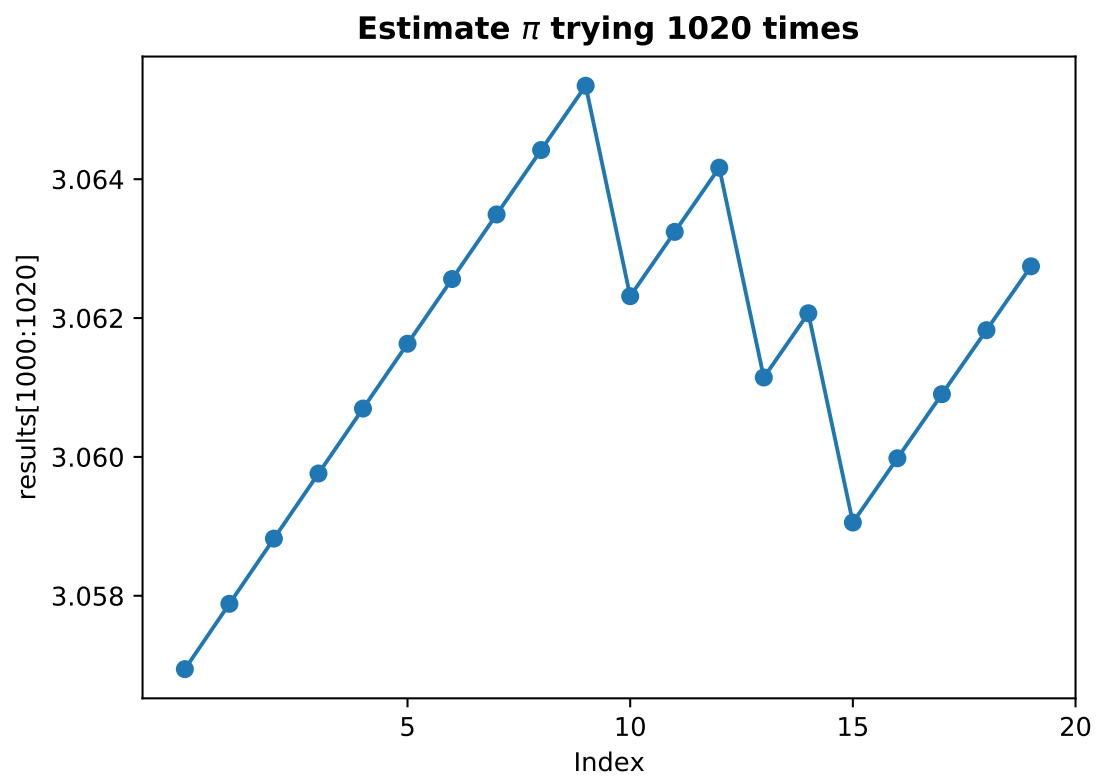
```
plt.show())
```



```
plt.plot(results[1001:1021],marker='o')
plt.title('Estimate '+r'$\pi$'+ ' trying 1020 times',fontweight='bold')
plt.xlabel('Index')
plt.ylabel('results[1000:1020]')
plt.xticks([5,10,15,20])
```

```
## ([<matplotlib.axis.XTick object at 0x000000002CBA6208>, <matplotlib.axis.XTick object at 0x000000002CBA0DA0>,
```

```
plt.show())
```



```
plt.plot(results[100000:100021],marker='o')
ax=plt.gca()
ax.get_yaxis().get_major_formatter().set_useOffset(False)
plt.title('Estimate '+r'$\pi$'+ ' trying 100020 times',fontweight='bold')
plt.xlabel('Index')
plt.ylabel('results[100000:100020]')
plt.xticks([5,10,15,20])
```

```
## ([<matplotlib.axis.XTick object at 0x000000002CA12898>, <matplotlib.axis.XTick object at 0x000000002CA122E8>]
```

```
plt.show()
```

