7.4

In [3]:

```
import numpy as np
import scipy.stats as st
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex

%matplotlib inline
```

In [4]:

```
data = []

file = open('6.csv', 'r')

mylambda = 1 / float(file.readline()[9:])
t_0 = float(file.readline()[6:])
t = float(file.readline()[4:])

for str in file:
    for c in str.split():
        data.append(float(c))
file.close()
```

In [23]:

```
def estimate_func(_alpha, _beta):
    est = np.array([0], dtype=float)
    Sum = 0
    for i in range(1, len(data)):
        Sum += data[i] - data[i-1]
        alpha = _alpha + i
        beta = _beta + Sum
        est = np.append(est, alpha / beta)
    n = np.arange(len(est))
    y = [mylambda for i in range(len(est))]
    print("alpha = ", _alpha)
    print("beta  = ", _beta)
    plt.figure(figsize=(10, 5))
    plt.plot(n, est, 'green', alpha=0.5, label='lambda estimate')
    plt.plot(n, y, 'brown', alpha=0.5, label='lambda value')
    plt.ylim([0.04, 0.08])
    plt.legend()
    plt.show()
```

Найдем байесовскую оценку параметра λ, взяв в качестве априорного распределения $\Gamma(\alpha, \beta)$ (сопряженное к экспоненциальному).

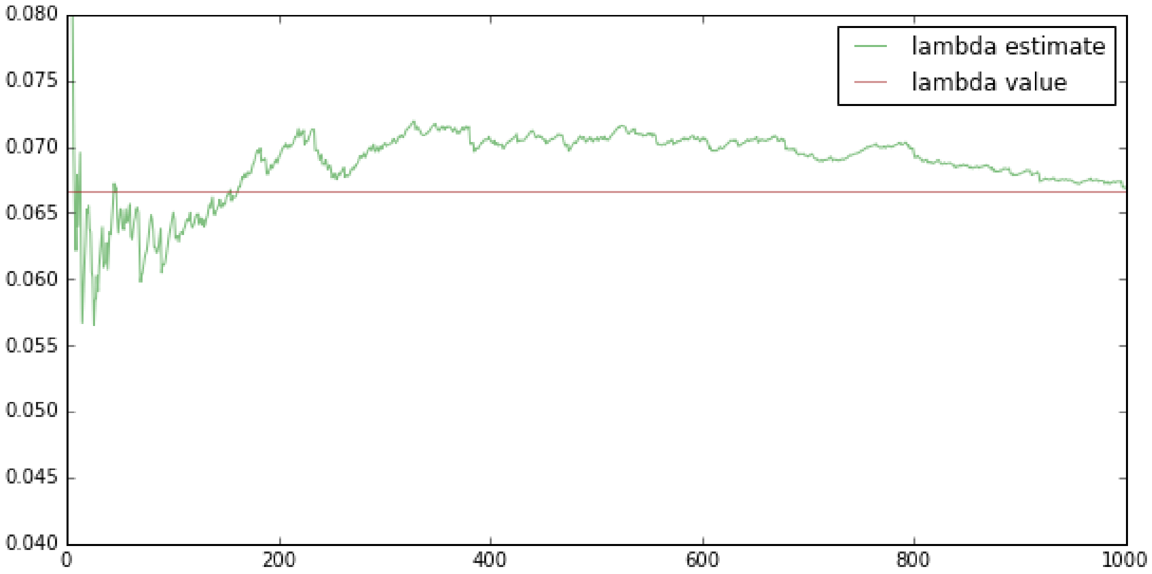
Байесовская оценка: $\lambda_1 = \frac{\alpha+n}{\beta+\sum_{i=1}^{n} x_i}$

Построим графики зависимости $f(n) = \lambda_1$ для различных параметром гамма распределения и сравним с истинным значением $\lambda = 1/15$

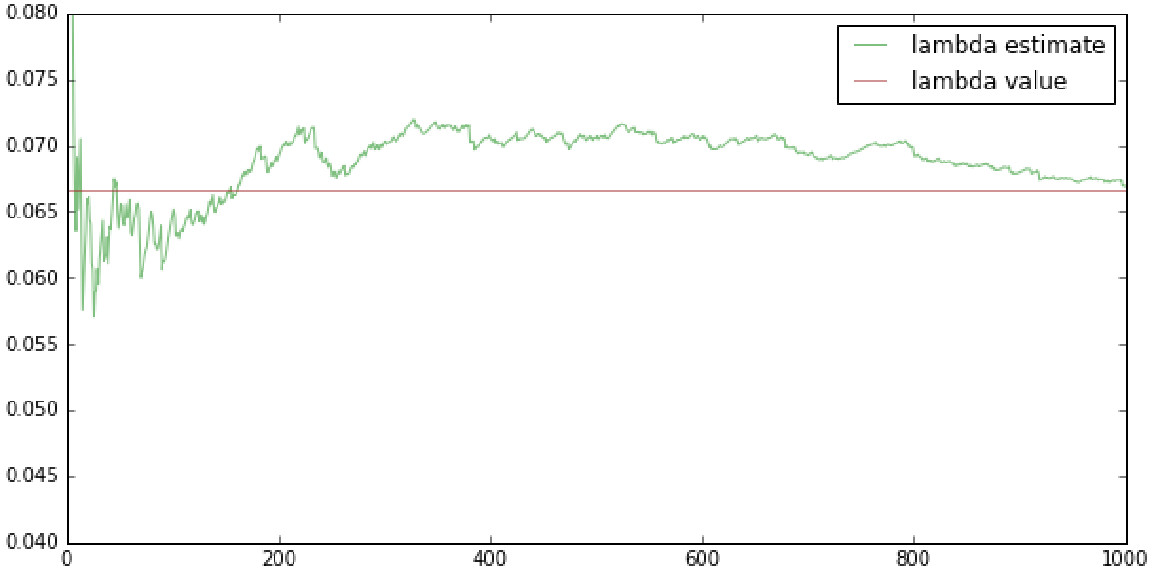
In [31]:

```
estimate_func(0.5,0.5)
estimate_func(1,5)
estimate_func(8,2)
estimate_func(20,20)
estimate_func(50,50)
```

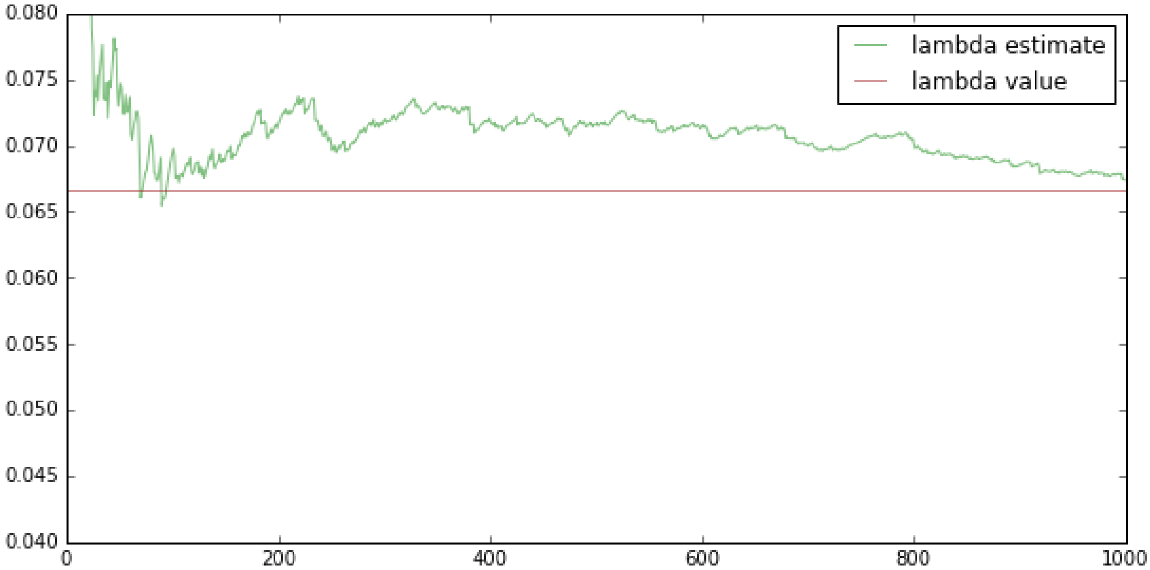
```
alpha =  0.5
beta  =  0.5
```



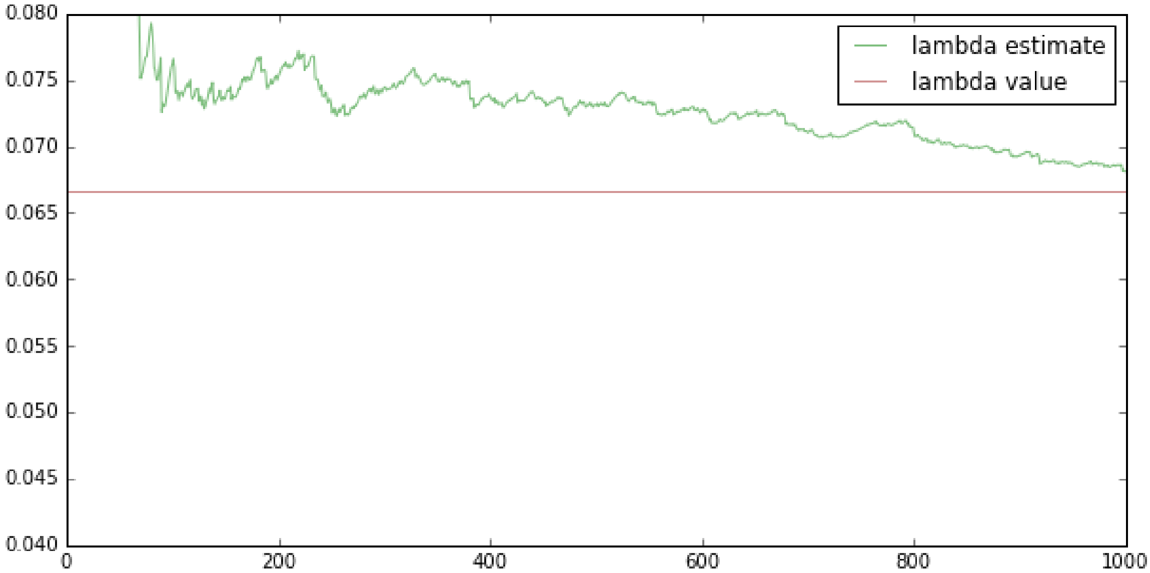

```
alpha =  1
beta  =  5
```



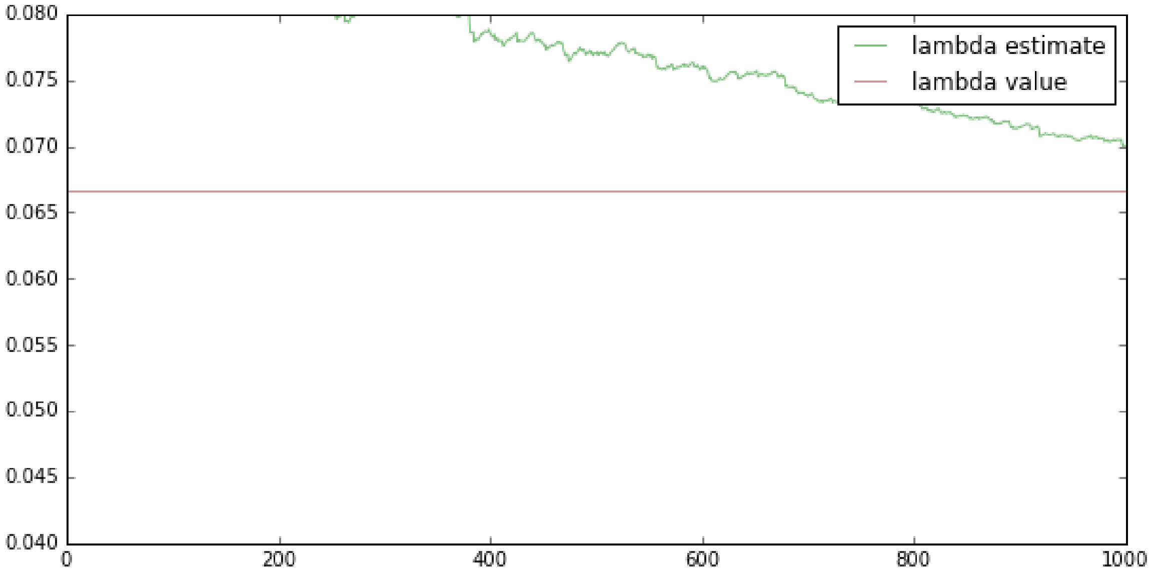

```
alpha =  8
beta  =  2
```




```
alpha =  20
beta  =  20
```




```
alpha =  50
beta  =  50
```



Из графиков видно, что с увеличением параметром гамма распределения оценка ухудшается.