In [1]:

```
import numpy as np
import scipy.stats as sps
import matplotlib.pyplot as plt
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [9]:

```
N = 1000
k = 10000
theta = 5
```

In [17]:

```
sample = sps.uniform.rvs(loc=0, scale=theta, size=k*N)
sample = sample.reshape((N, k))
```

Оценка максимального правдоподобия для $R(0, \theta)$ - $X_{(n)}$.

In [30]:

```
est_1 = np.empty((N, k))
for i in range(N):
    est_1[i, :] = sample[:(i + 1), :].max(axis=0)
```

In [33]:

```
est_2 = np.cumsum(sample, axis=0)
s = np.arange(1, N + 1, 1).reshape(N, 1)
est_2 = est_2 / s
est_2 = (est_2 * 2 + est_1) / 2
```

По выборкам $\widehat{\theta_j^n}$ и $\widetilde{\theta_j^n}$ найдем оценки дисперсий(выюорочная дисперсия $S^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})}{n}$):

In [50]:

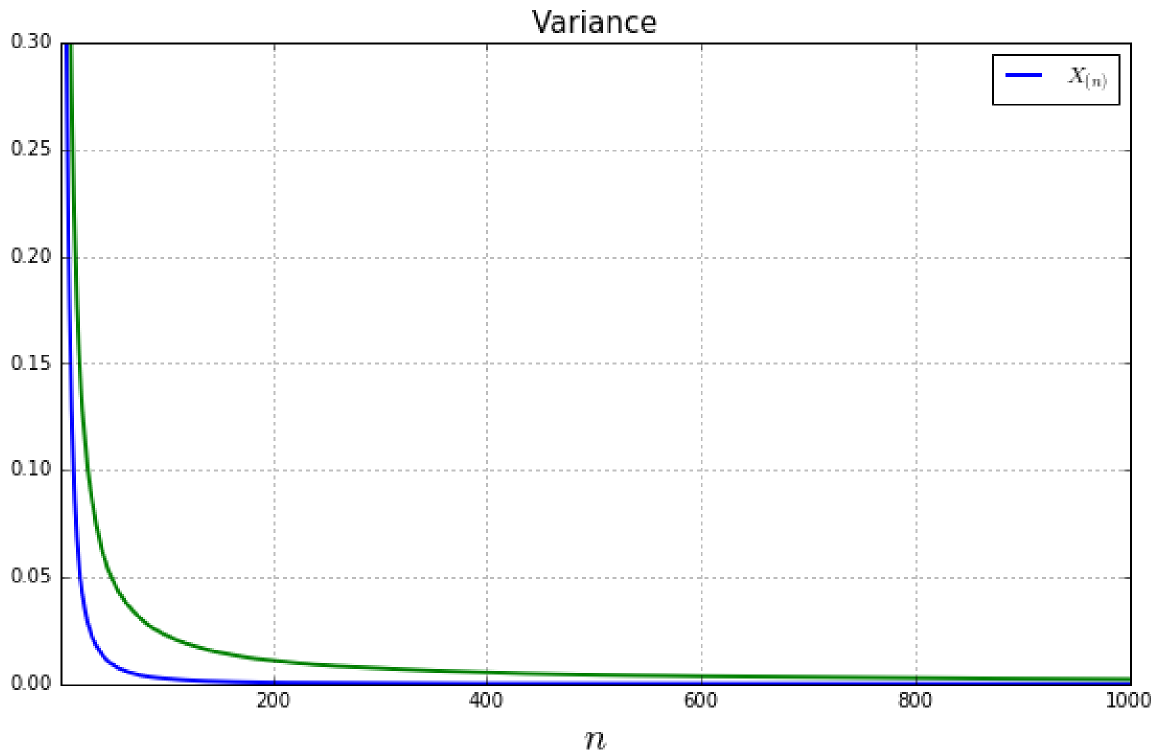
```
def f_var(est):
    theta_mean = (est.sum(1) / k).reshape(N, 1)
    res = ((est - theta_mean) ** 2).sum(1) / k
    return res
```

Построим график зависимости дисперсий от n:

In [65]:

```
c = np.arange(1, N + 1, 1)
plt.figure(figsize=(10, 6))

plt.plot(c, f_var(est_1), linewidth = 2, label=r'$X_{(n)}$')
plt.plot(c, f_var(est_2), linewidth = 2)
plt.legend()
plt.xlim(1, n)
plt.ylim(0, 0.3)
plt.xlabel('$n$', fontsize = 20)
plt.title('Variance', fontsize = 15)
plt.grid()
plt.show()
```



Из графика видно, что дисперсия для обеих оценок уменьшается при увеличении n. При этом при всех n дисперсия оценки максимального правдоподобия меньше, значит она дает более точный результат(ОМП - состоятельна).