```
import matplotlib.pyplot as plt
import numpy as np
from scipy.special import factorial
from scipy import stats
import warnings

warnings.simplefilter('ignore')

%matplotlib inline
```

```
def gen_evaluation_by_n(N, data, callback):
    return np.fromiter(
      map(callback, np.arange(0, N, 1)),
      dtype=np.float64
    )
```

Первое задание

(К теоретической задаче 1)
Сгенерируйте выборку $X_1, \ldots, X_N$ из равномерного распределения на отрезке $[0, \theta]$ для $N = 10^4$. Для всех $n \leq N$ посчитайте оценки параметра θ из теоретической задачи: $2\overline{X}$, $\overline{X} + X_{(n)}/2$, $(n+1)X_{(1)}$, $X_{(1)} + X_{(n)}$, $\frac{n+1}{n}X_{(n)}$. Постройте на одном графике разными цветами для всех оценок функции модуля разности оценки и истинного значения θ в зависимости от n. Если некоторые оценки (при фиксированном значении n) сильно отличаются от истинного значения параметра θ, то исключите их и постройте еще один график со всеми кривыми (для измененного значения θ). Для избавления от больших значений разности в начале ограничьте масштаб графика. Для наглядности точки можно соединить линиями. Какая оценка получилась лучше (в смысле упомянутого модуля разности при $n = N$)? Проведите эксперимент для разных значений θ (количество графиков равно количеству значений θ).

```
N, THETA = 10**4, 1
```

Сгенерируем выборку из равномерного распределения на отрезке $[0; \theta]$ и найдем по ней порядковые статистики:

```
observations = stats.uniform.rvs(size=N, loc=0, scale=THETA)
```

Посчитаем оценки параметра

- $2\overline{X}$

```
double_mean = gen_evaluation_by_n(
    N, observations, lambda n: 2*observations[:n+1].mean())
```

- $\overline{X} + X_{(n)}/2$

```
mean_plus_last = gen_evaluation_by_n(
    N, observations, lambda n: observations[:n+1].mean() + observations[:n+1].max()/2.0)
```

- $(n+1)X_{(1)}$

```
multi_first = gen_evaluation_by_n(
    N, observations, lambda n: (n+2) * observations[:n+1].min())
```

- $X_{(1)} + X_{(n)}$

```
first_plus_last = gen_evaluation_by_n(
    N, observations, lambda n: observations[:n+1].min() + observations[:n+1].max())
```

- $\frac{n+1}{n} X_{(n)}$

```
frac_last = gen_evaluation_by_n(
    N, observations, lambda n: (n+2.)/(n+1.) * observations[:n+1].max())
```

Посчитаем разности

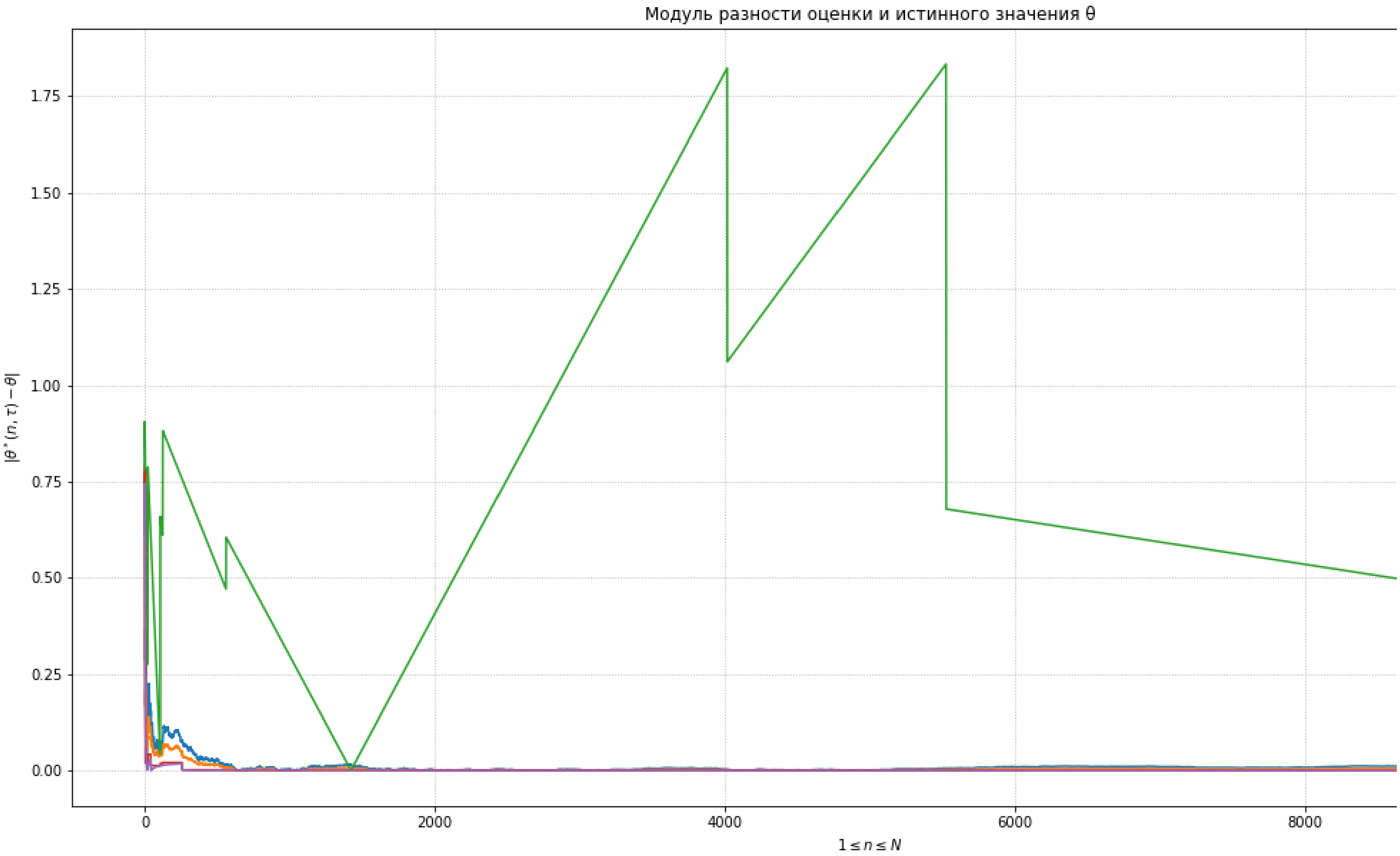
```
def get_abs_diff(param_estimation, param=THETA):
    return np.abs(param_estimation - param)
```

```
abs_diff_double_mean     = get_abs_diff(double_mean)
abs_diff_mean_plus_last  = get_abs_diff(mean_plus_last)
abs_diff_multi_first     = get_abs_diff(multi_first)
abs_diff_first_plus_last = get_abs_diff(first_plus_last)
abs_diff_frac_last       = get_abs_diff(frac_last)
```

Построим график

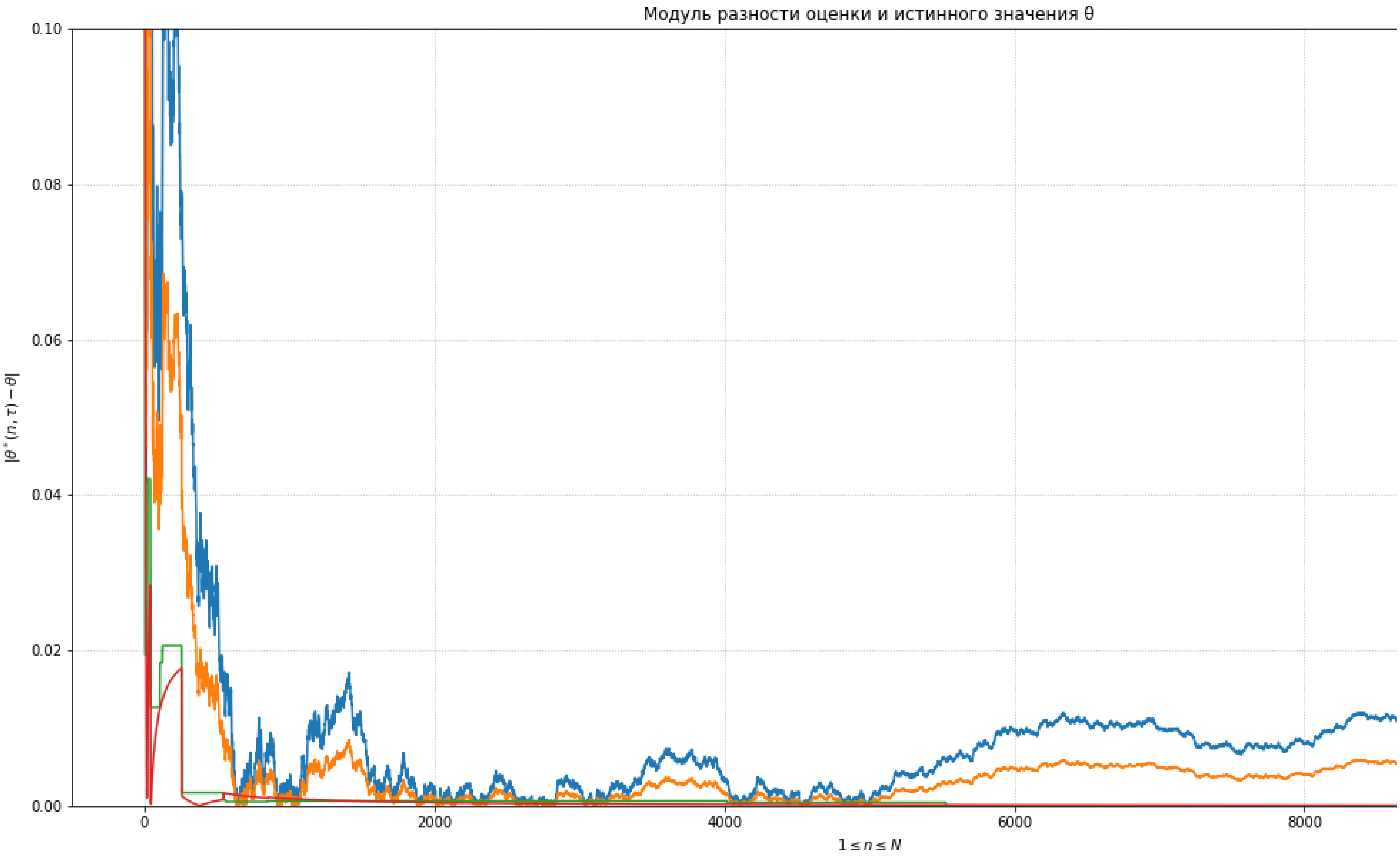
```
plt.figure(figsize=(20, 10))
plt.title("Модуль разности оценки и истинного значения θ")
plt.xlabel('$1\leq n \leq N$')
plt.ylabel('$|θ^*(n, \\tau) - θ|$')
plt.plot(
  np.linspace(0, N, N),
  abs_diff_double_mean,
  label='$2\overline{X}$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_mean_plus_last,
  label='$\overline{X} + X_{(n)}/2$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_multi_first,
  label='$(n + 1)X_{(1)}$'
)
```

```
plt.plot(
  np.linspace(0, N, N),
  abs_diff_first_plus_last,
  label='$X_{(1)} + X_{(n)}$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_frac_last,
  label='$\\frac{n + 1}{n}X_{(n)}$'
)
plt.grid(ls=':')
plt.legend()
plt.show()
```



Оценка $(n+1)X_{(1)}$ является несостоятельной, что и объясняет расходимость графика относительно оси абсцисс. Подробнее рассмотрим поведение остальных оценок.

```
plt.figure(figsize=(20, 10))
plt.title("Модуль разности оценки и истинного значения θ")
plt.xlabel('$1\leq n \leq N$')
plt.ylabel('$|θ^*(n, \\tau) - θ|$')
plt.ylim((.0, .1))
plt.plot(
  np.linspace(0, N, N),
  abs_diff_double_mean,
  label='$2\overline{X}$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_mean_plus_last,
  label='$\overline{X} + X_{(n)}/2$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_first_plus_last,
  label='$X_{(1)} + X_{(n)}$'
)
plt.plot(
  np.linspace(0, N, N),
  abs_diff_frac_last,
  label='$\\frac{n + 1}{n}X_{(n)}$'
)
plt.grid(ls=':')
plt.legend()
plt.show()
```



Проведём эксперименты для разных значений θ

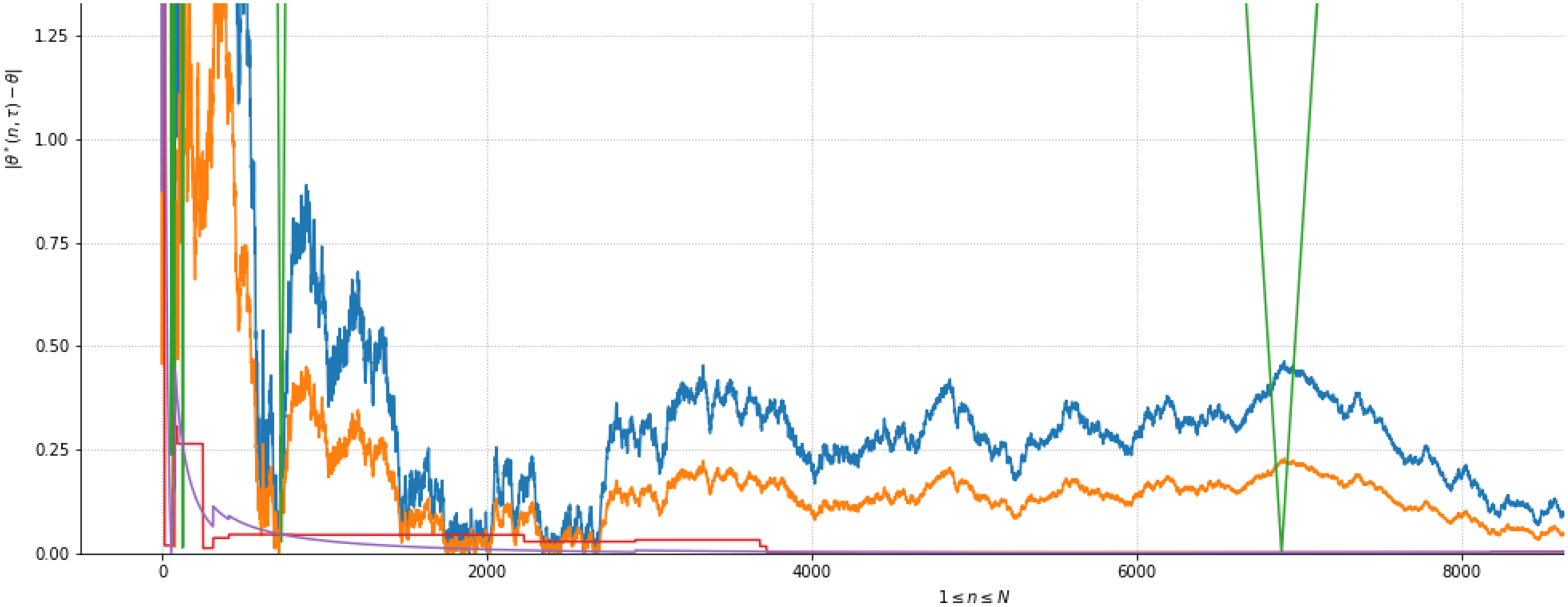
Напишем функцию для генерации выборки и построения графиков.
Избавимся от больших значений разности в начале ограничим масштаб графика $\frac{\theta}{20}$.

Disclaimer: много строк кода

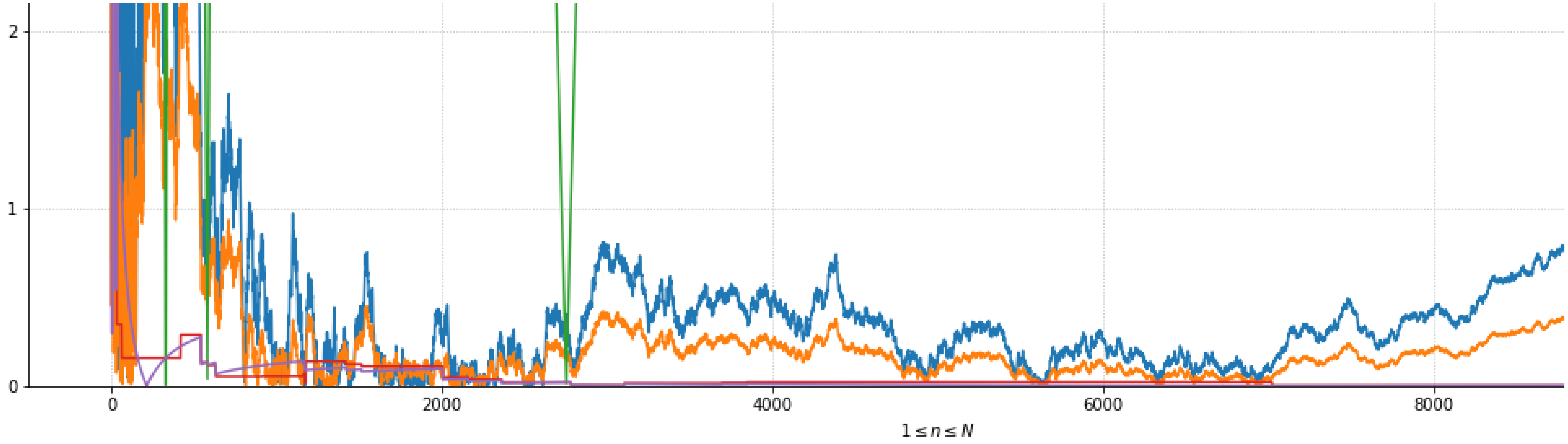
```
def investigate(theta):
    observations = stats.uniform.rvs(size=N, loc=0, scale=theta)
    double_mean = gen_evaluation_by_n(
      N, observations, lambda n: 2*observations[:n+1].mean())
    mean_plus_last = gen_evaluation_by_n(
      N, observations, lambda n: observations[:n+1].mean() + observations[:n+1].max()/2.0)
    multi_first = gen_evaluation_by_n(
      N, observations, lambda n: (n+2) * observations[:n+1].min())
    first_plus_last = gen_evaluation_by_n(
      N, observations, lambda n: observations[:n+1].min() + observations[:n+1].max())
    frac_last = gen_evaluation_by_n(
      N, observations, lambda n: (n+2.)/(n+1.) * observations[:n+1].max())
    abs_diff_double_mean     = get_abs_diff(double_mean, theta)
    abs_diff_mean_plus_last  = get_abs_diff(mean_plus_last, theta)
    abs_diff_multi_first     = get_abs_diff(multi_first, theta)
    abs_diff_first_plus_last = get_abs_diff(first_plus_last, theta)
    abs_diff_frac_last       = get_abs_diff(frac_last, theta)
    plt.figure(figsize=(20, 10))
    plt.title("Модуль разности оценки и истинного значения θ=%.2f" % theta)
    plt.xlabel('$1\leq n \leq N$')
    plt.ylabel('$|θ^*(n, \\tau) - θ|$')
    plt.ylim((.0, .05*theta))
    plt.plot(
      np.linspace(0, N, N),
      abs_diff_double_mean,
      label='$2\overline{X}$'
    )
    plt.plot(
      np.linspace(0, N, N),
      abs_diff_mean_plus_last,
      label='$\overline{X} + X_{(n)}/2$'
```

```
      label='$\overline{X} + X_{(n)}/2$'
    )
    plt.plot(
      np.linspace(0, N, N),
      abs_diff_multi_first,
      label='$(n + 1)X_{(1)}$'
    )
    plt.plot(
      np.linspace(0, N, N),
      abs_diff_first_plus_last,
      label='$X_{(1)} + X_{(n)}$'
    )
    plt.plot(
      np.linspace(0, N, N),
      abs_diff_frac_last,
      label='$\\frac{n + 1}{n}X_{(n)}$'
    )
    plt.grid(ls=':')
    plt.legend()
    plt.show()


for theta in [0.01, 0.2, 0.5, 2, 5, 10, 42, 100]:
    investigate(theta)
```


1.25
1.00
0.75
0.50
0.25
0.00
$|\theta^*(n,\tau)-\theta|$
0
2000
4000
6000
8000
$1 \leq n \leq N$


Модуль разности оценки и истинного значения θ=100.00
5
4
3
$|\theta^*(n,\tau)-\theta|$


2
1
0
0
2000
4000
6000
8000
$1 \leq n \leq N$

Вывод к первому заданию

В этом задании мы сравнили модуль разности оценок $2\overline{X}, \overline{X} + X_{(n)}/2, (n+1)X_{(1)}, X_{(1)} + X_{(n)}, \frac{n+1}{n}X_{(n)}$ с истинным значением параметра θ для $\theta \in \{0.01, 0.2, 0.5, 1, 2, 5, 10, 42, 100\}$ для всех $n \leq N$. На практике показали несостоятельность

Второе задание

(К теоретической задаче 5)
Сгенерируйте выборку $X_1, \ldots, X_N$ из экспоненциального распределения с параметром $\theta = 1$ для $N = 10^4$. Для всех $n \leq N$ посчитайте оценку $(k!/\overline{X^k})^{1/k}$ параметра θ. Проведите исследование, аналогичное предыдущей задаче, и выясните, при каком k оценка ведет себя лучше (рассмотрите не менее 10 различных значений k).

```
N, THETA = 10**4, 1
```

Сгенерируем выборку из экспоненциального распределения с параметром $\theta = 1$:

```
observations = stats.expon.rvs(size=N, loc=0, scale=THETA)
```

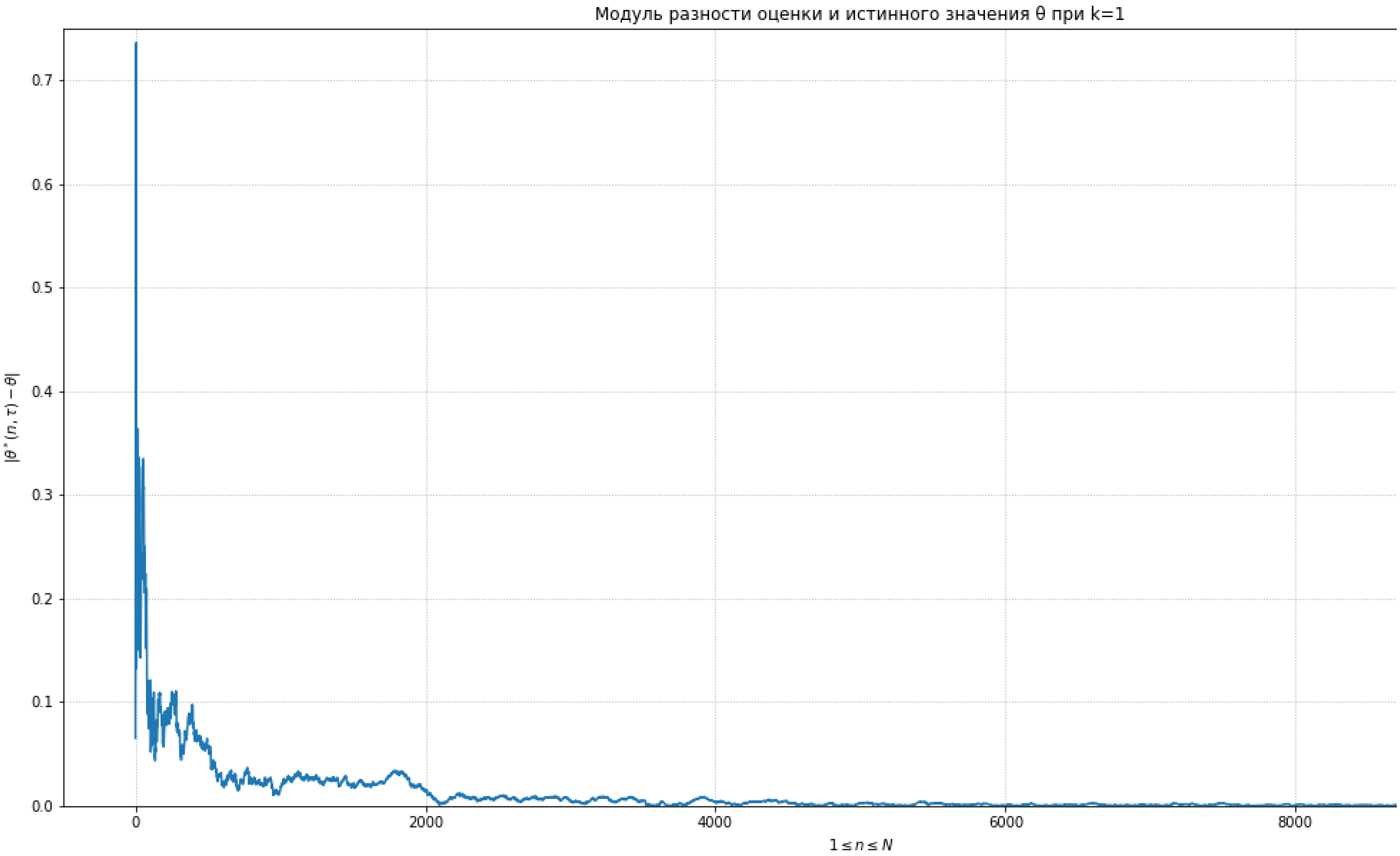
Напишем функцию построения графиков для каждого k :

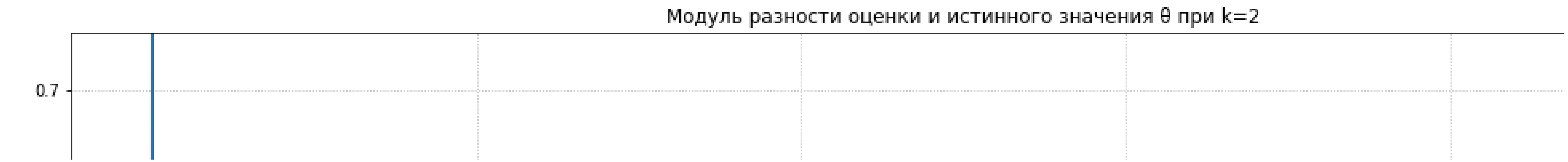
```
def investigate_kth_param(observations, k=1, param=THETA):
    param_estimation= np.fromiter(
      map(
        lambda n: (factorial(k) / np.mean(observations[:n+1]**k)) ** (1/k),
        np.arange(0, N, 1)),
      dtype=np.float64
    )
    abs_diff = np.abs(param_estimation - param)
    plt.figure(figsize=(20, 10))
    plt.ylim((0., .75))
    plt.title("Модуль разности оценки и истинного значения θ при k=%d" % k)
    plt.xlabel('$1\leq n \leq N$')
    plt.ylabel('$|θ^*(n, \\tau) - θ|$')
```

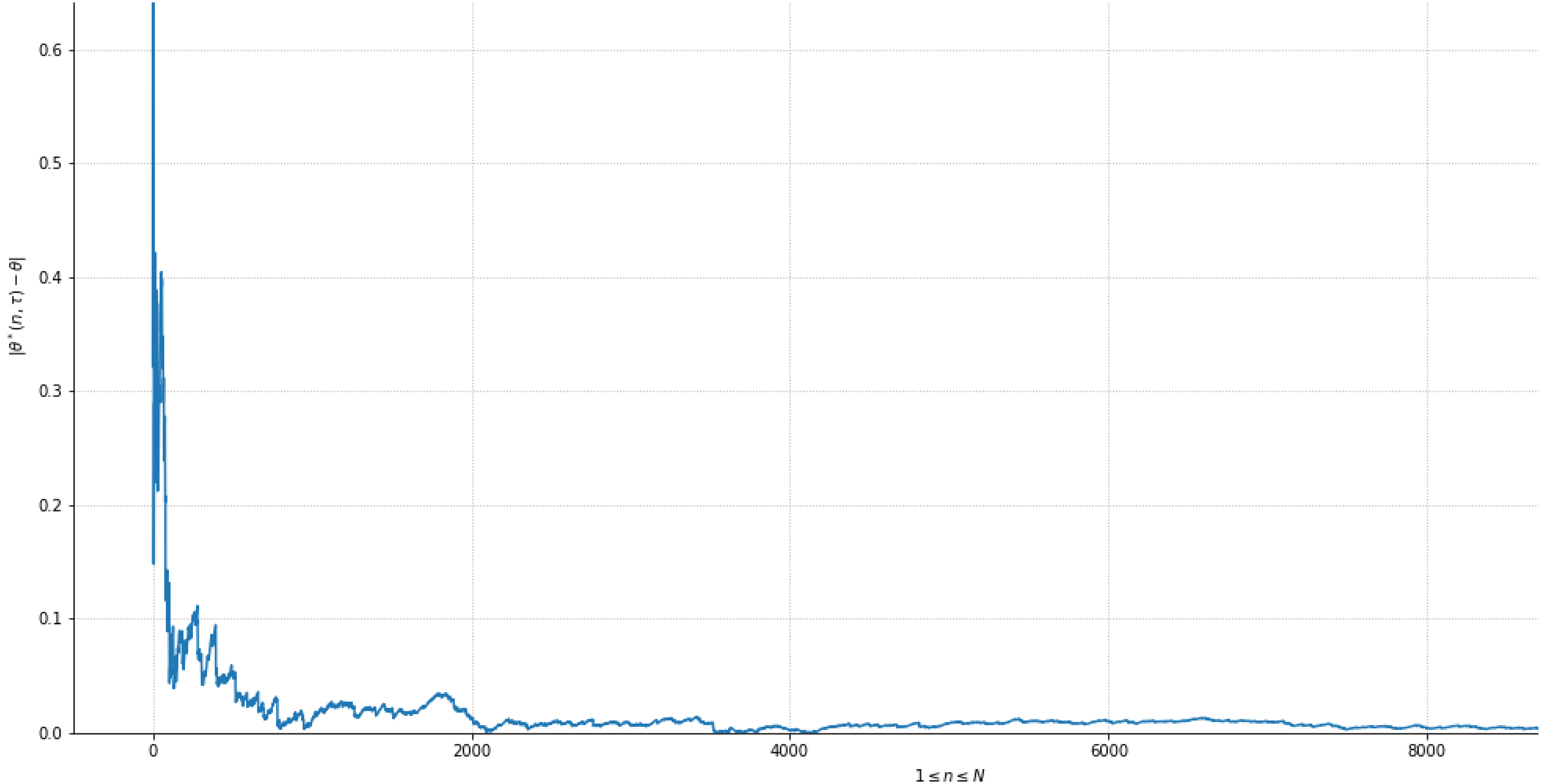
```
    plt.plot(np.linspace(0, N, N), abs_diff)
    plt.grid(ls=':')
    plt.show()
```

Исследуем при каком k оценка ведет себя лучше

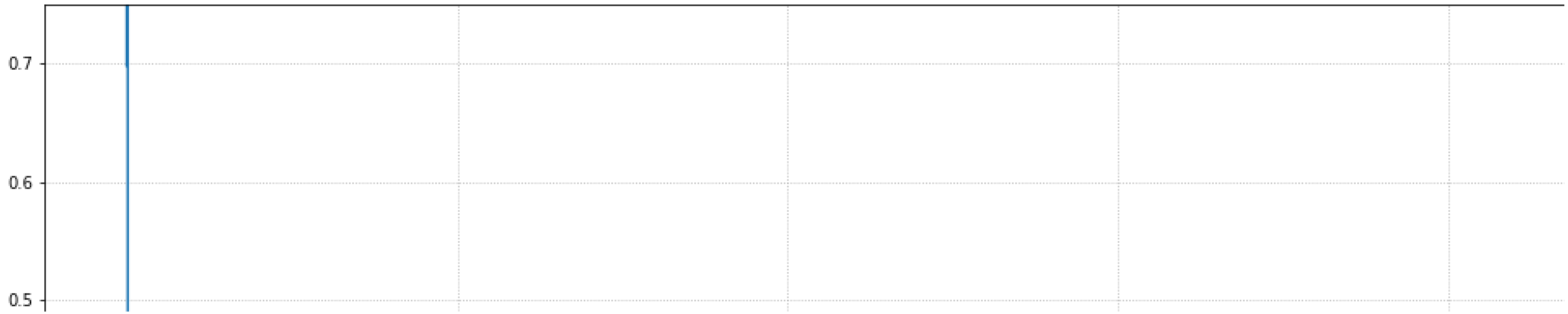
```
max_k = 20
for k in range(1, max_k+1):
    investigate_kth_param(observations, k)
```

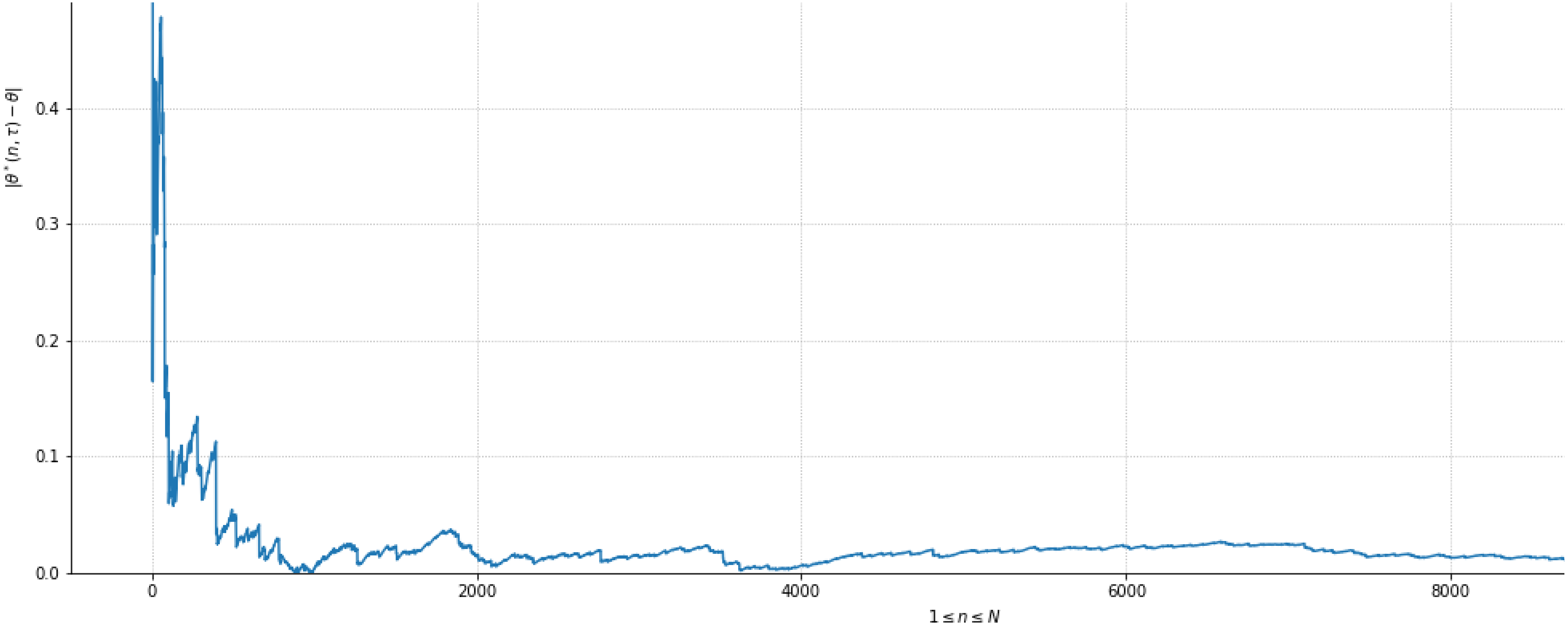

Модуль разности оценки и истинного значения θ при k=1
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
$|\theta^*(n, \tau) - \theta|$
0
2000
4000
6000
8000
$1 \leq n \leq N$


Модуль разности оценки и истинного значения θ при k=2
0.7

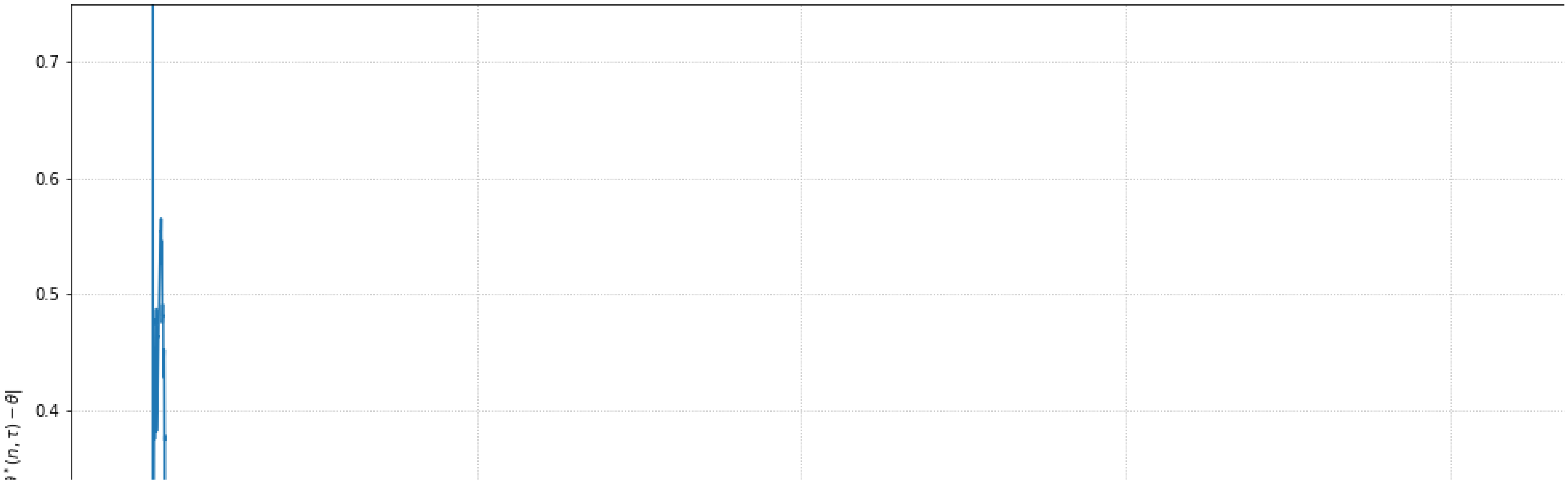


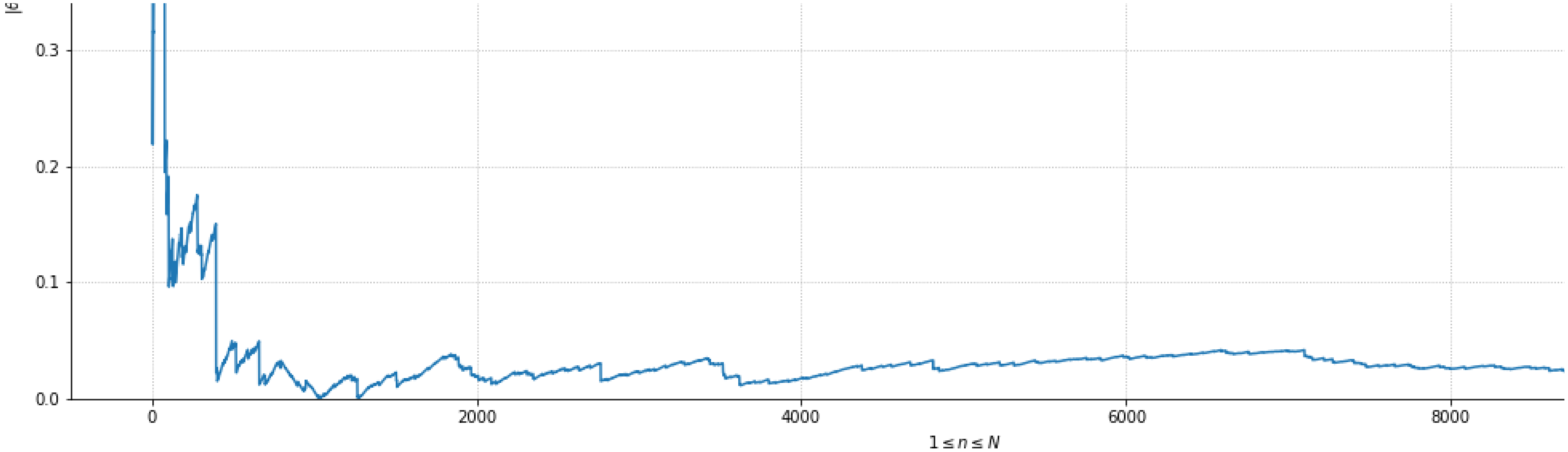
Модуль разности оценки и истинного значения θ при k=3



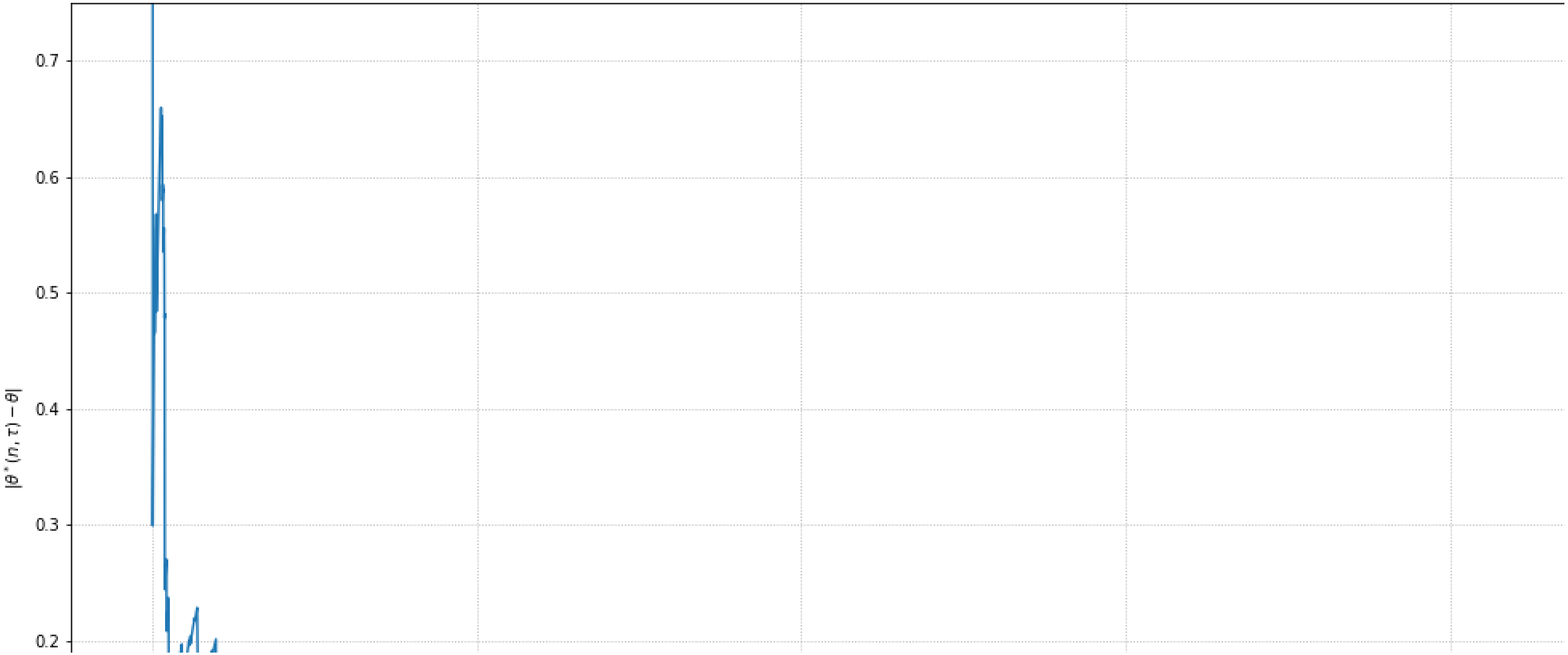


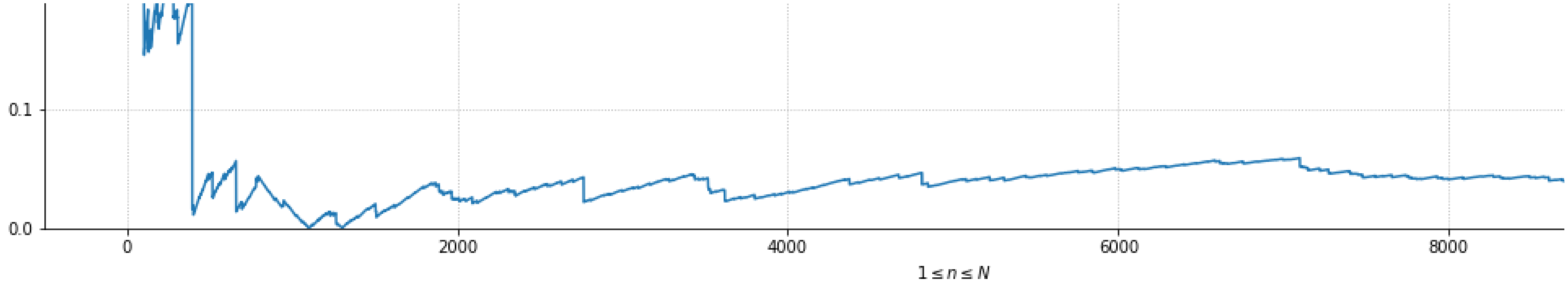
Модуль разности оценки и истинного значения θ при k=4



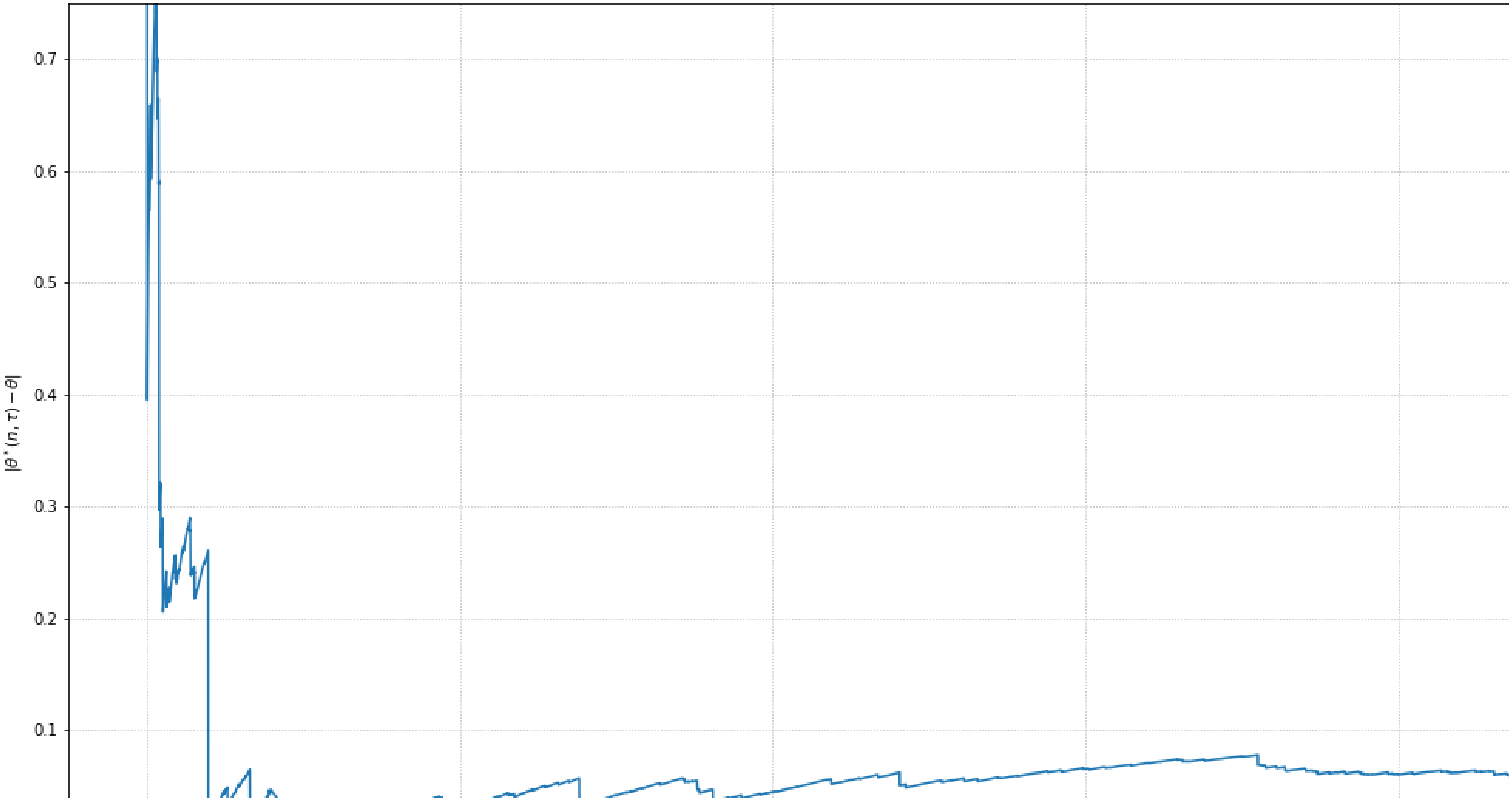


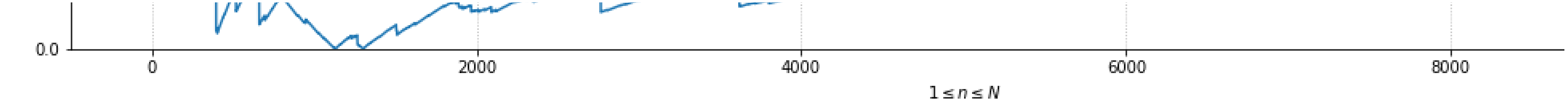
Модуль разности оценки и истинного значения θ при k=5



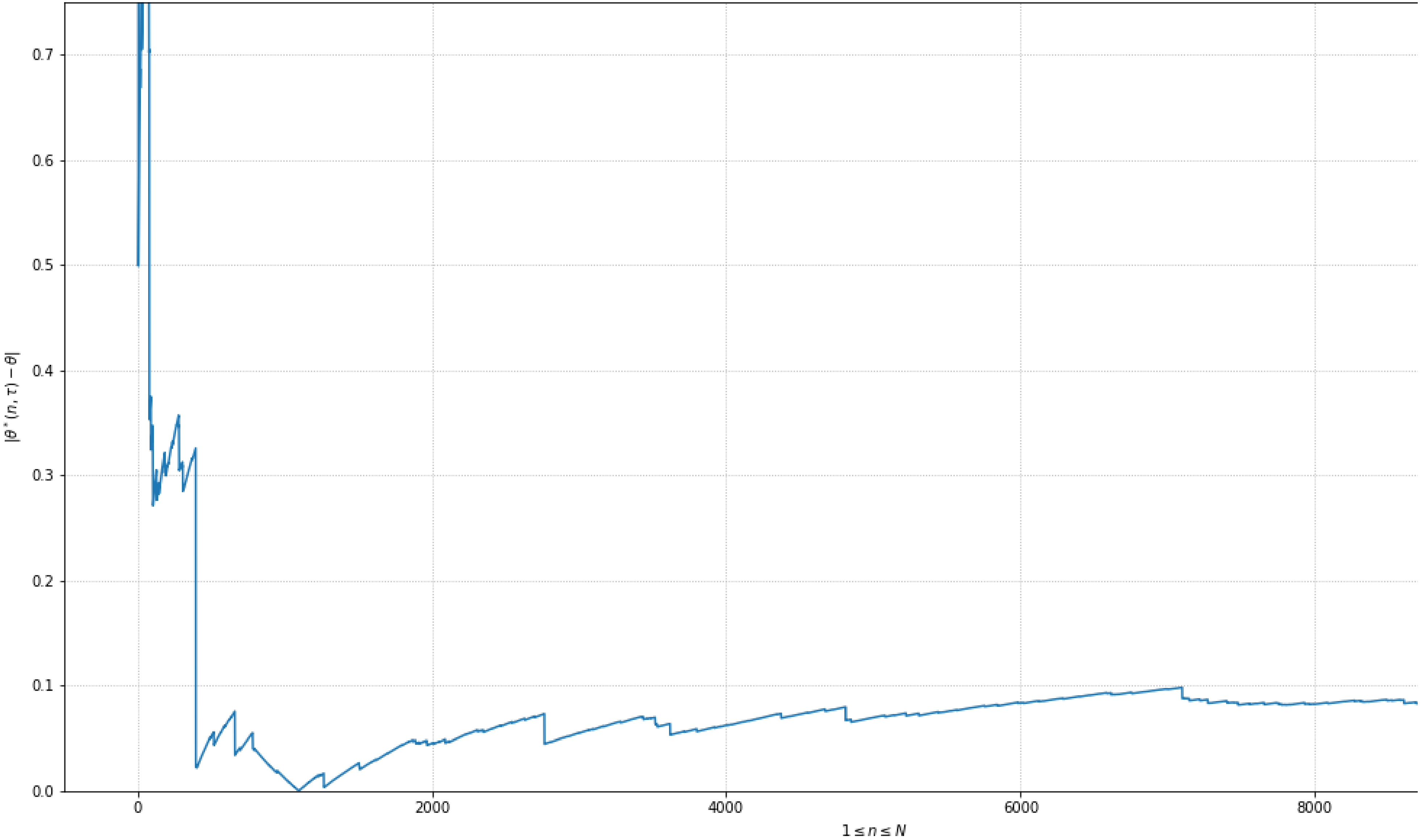


Модуль разности оценки и истинного значения θ при k=6

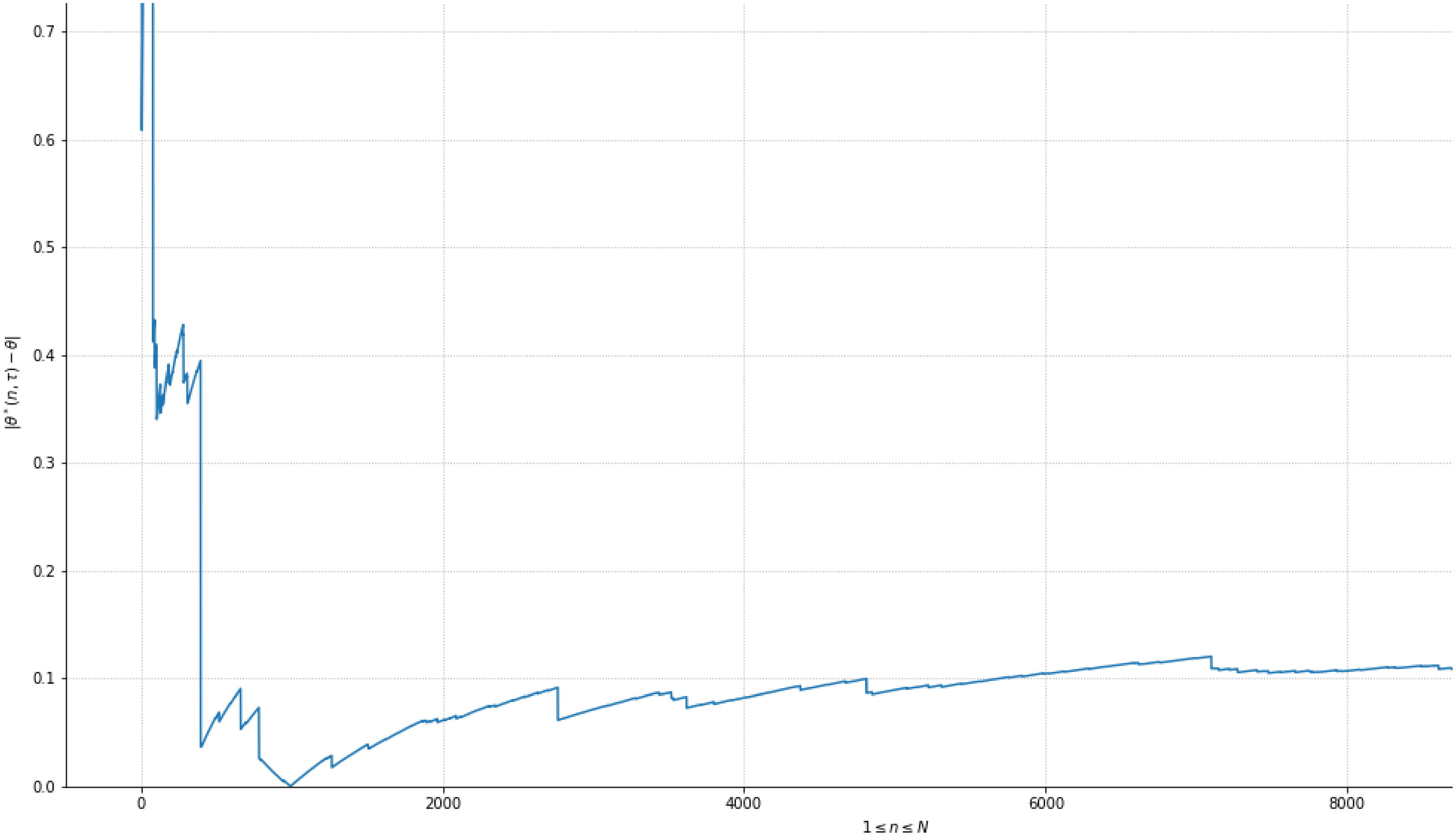




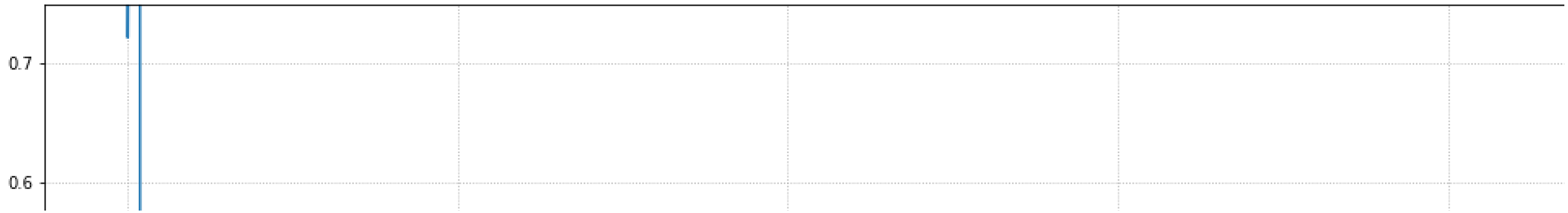
Модуль разности оценки и истинного значения θ при k=7

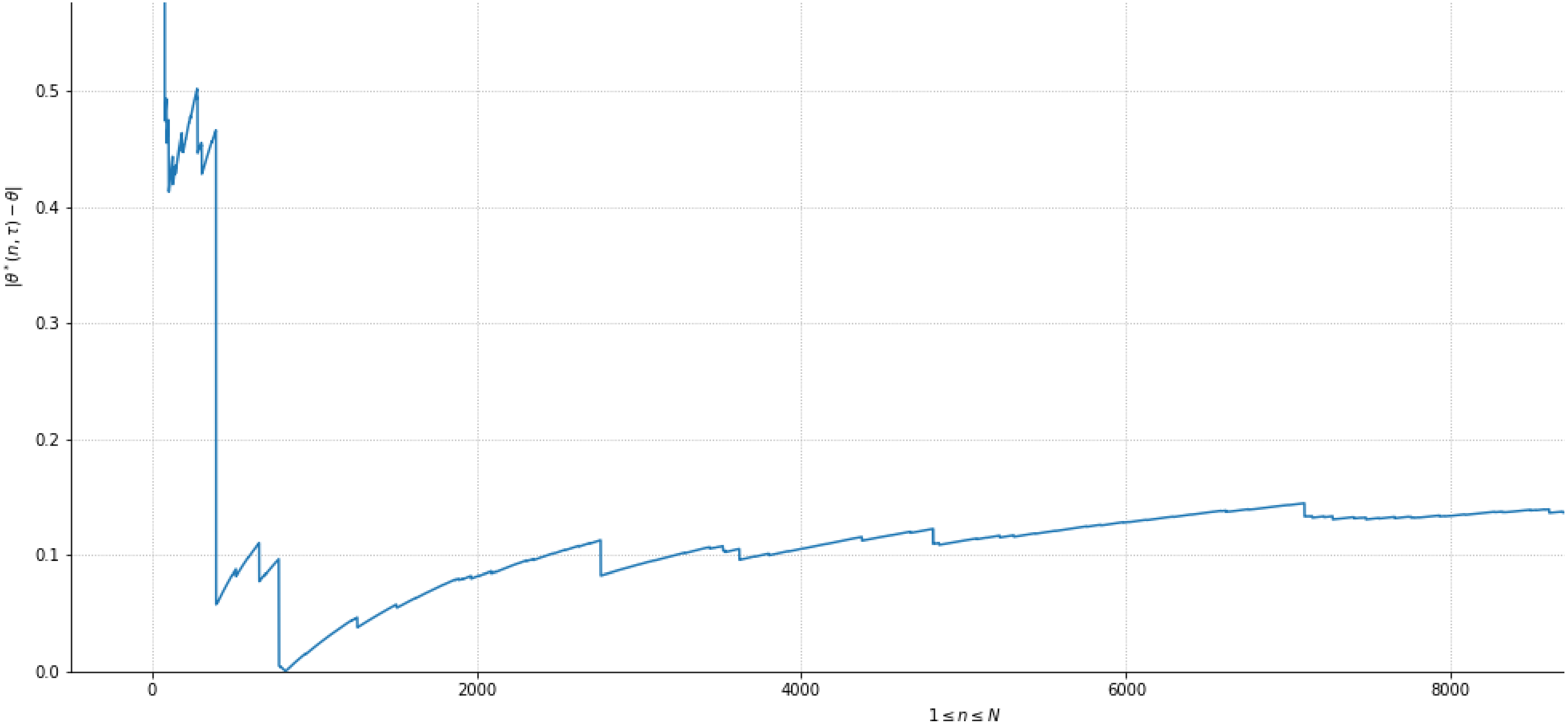


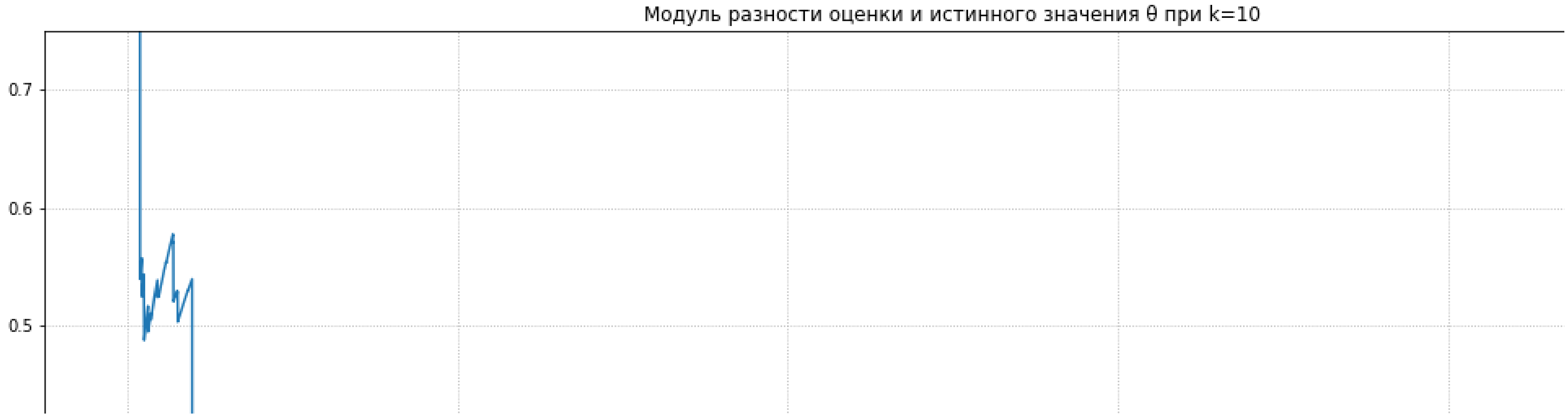
Модуль разности оценки и истинного значения θ при k=8

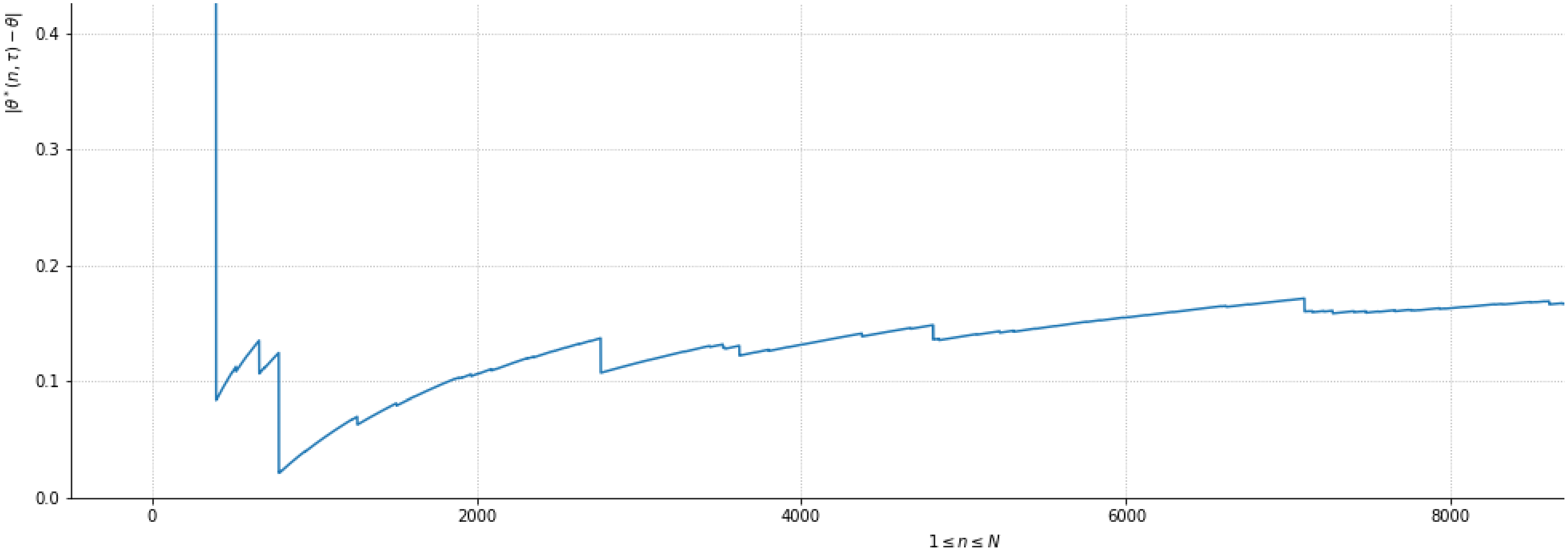


Модуль разности оценки и истинного значения θ при k=9

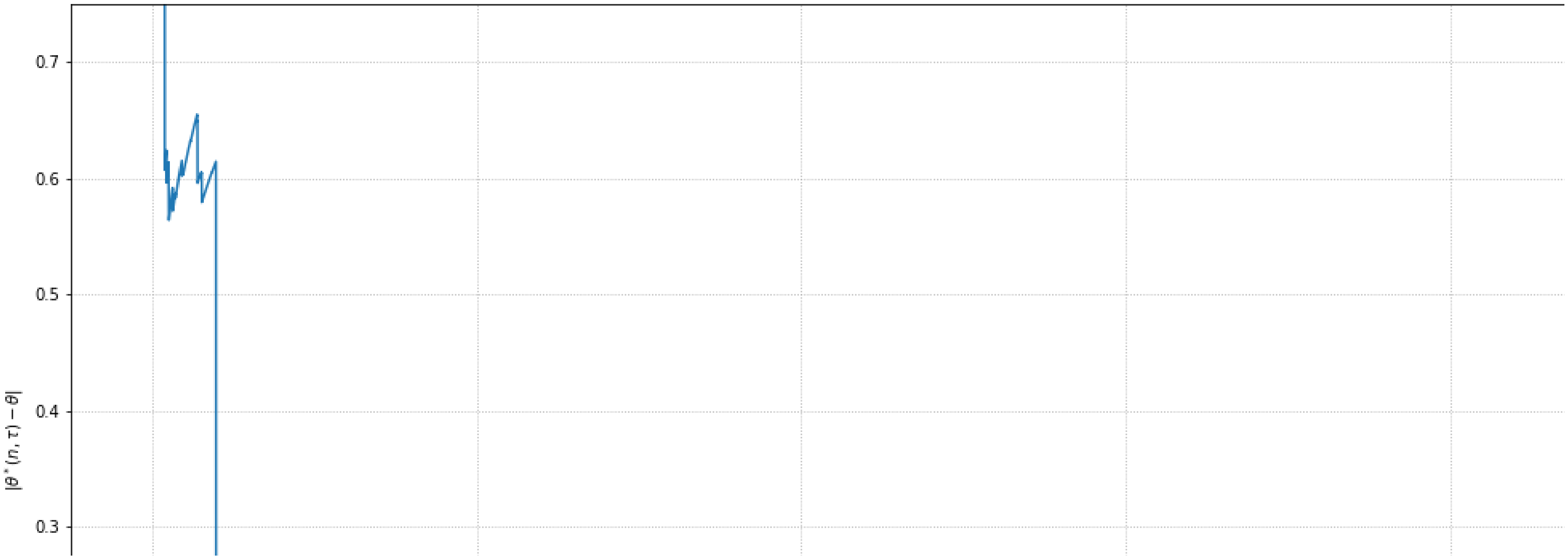


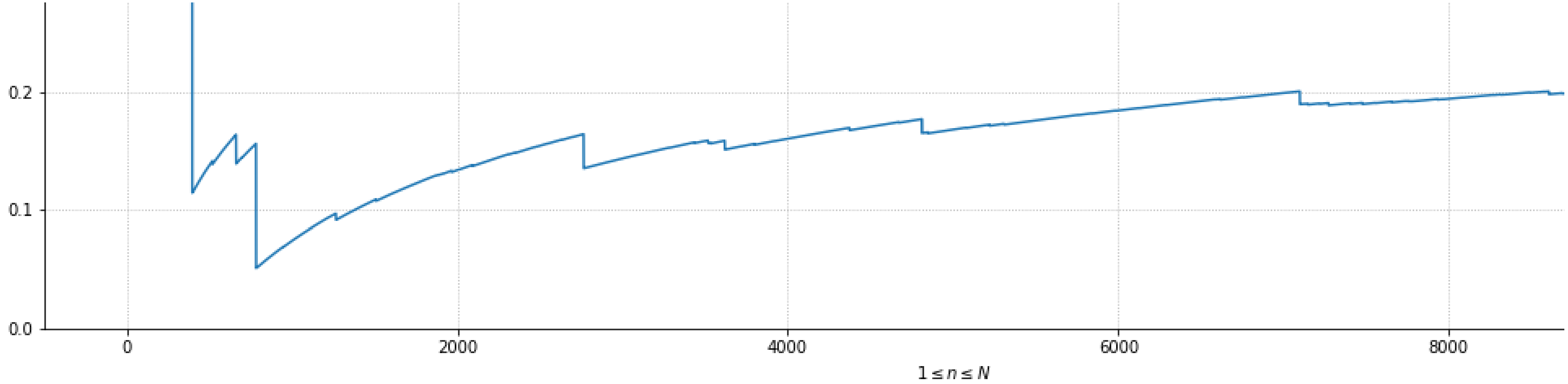

0.5
0.4
0.3
0.2
0.1
0.0
$|\theta^*(n,\tau)-\theta|$
0
2000
4000
6000
8000
$1 \leq n \leq N$

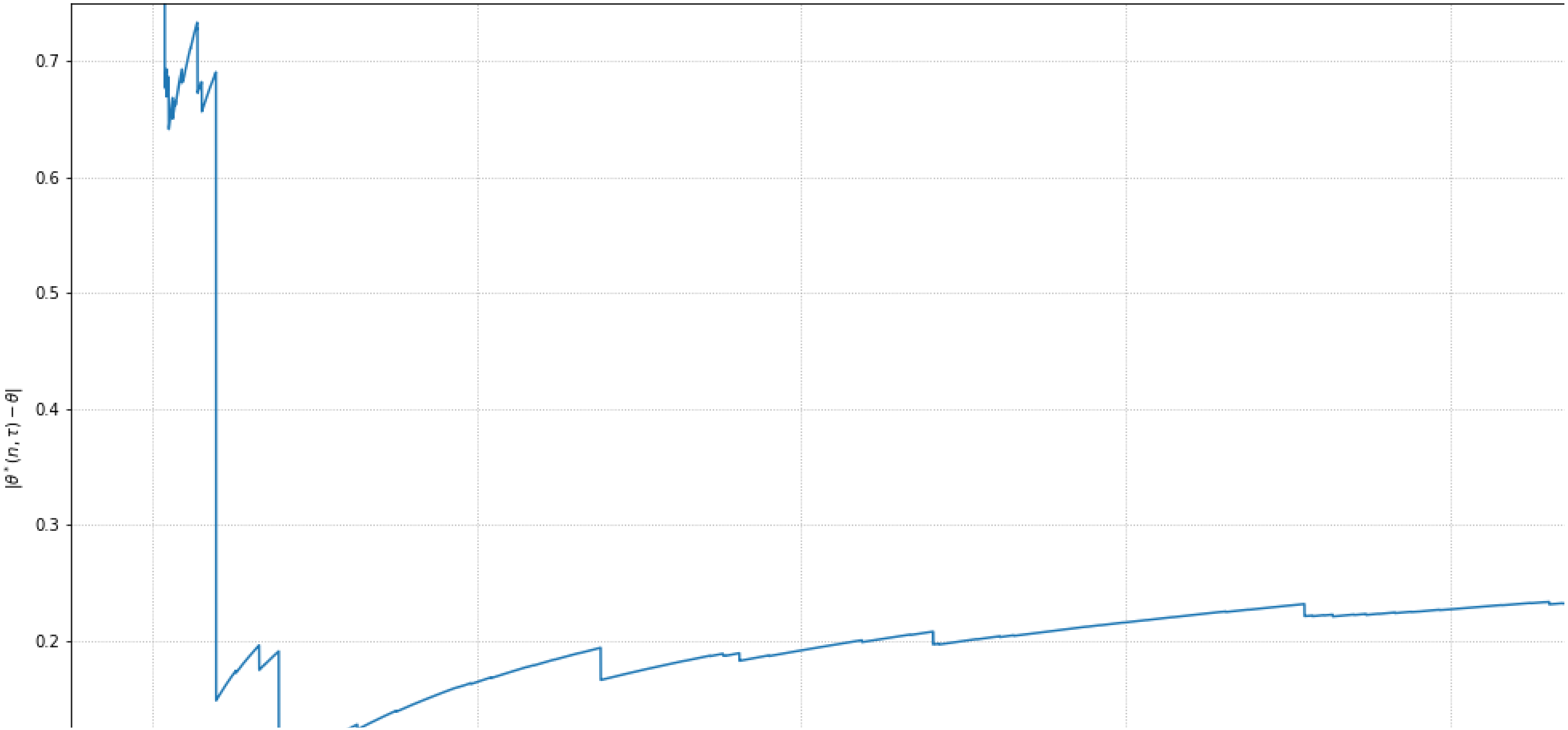

Модуль разности оценки и истинного значения θ при k=10
0.7
0.6
0.5

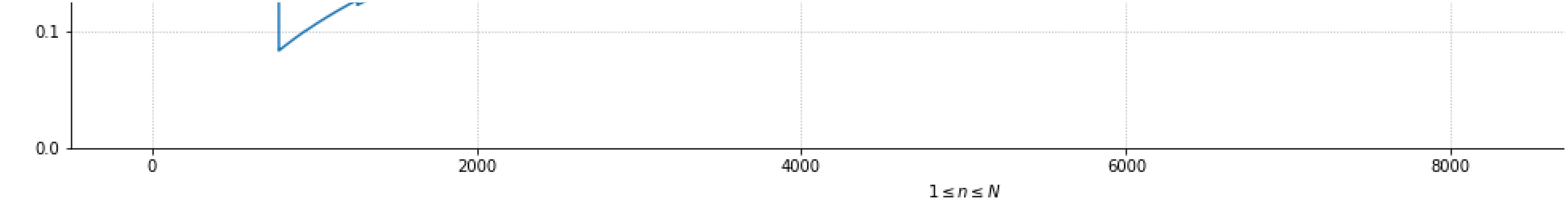


Модуль разности оценки и истинного значения θ при k=11

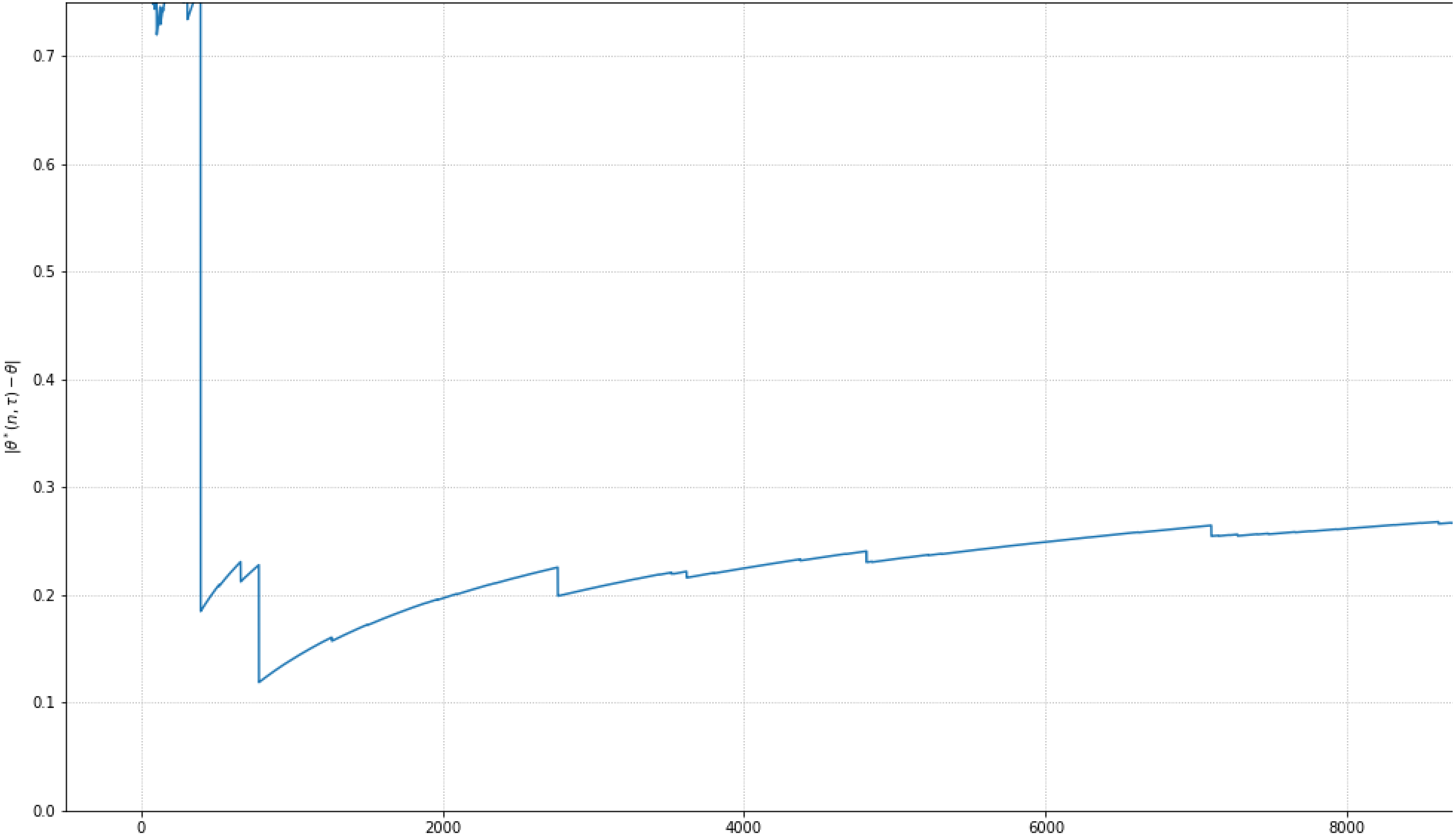



0.2
0.1
0.0
0
2000
4000
6000
8000
$1 \leq n \leq N$


Модуль разности оценки и истинного значения θ при k=12
0.7
0.6
0.5
0.4
0.3
0.2
$|\theta^*(n,\tau) - \theta|$

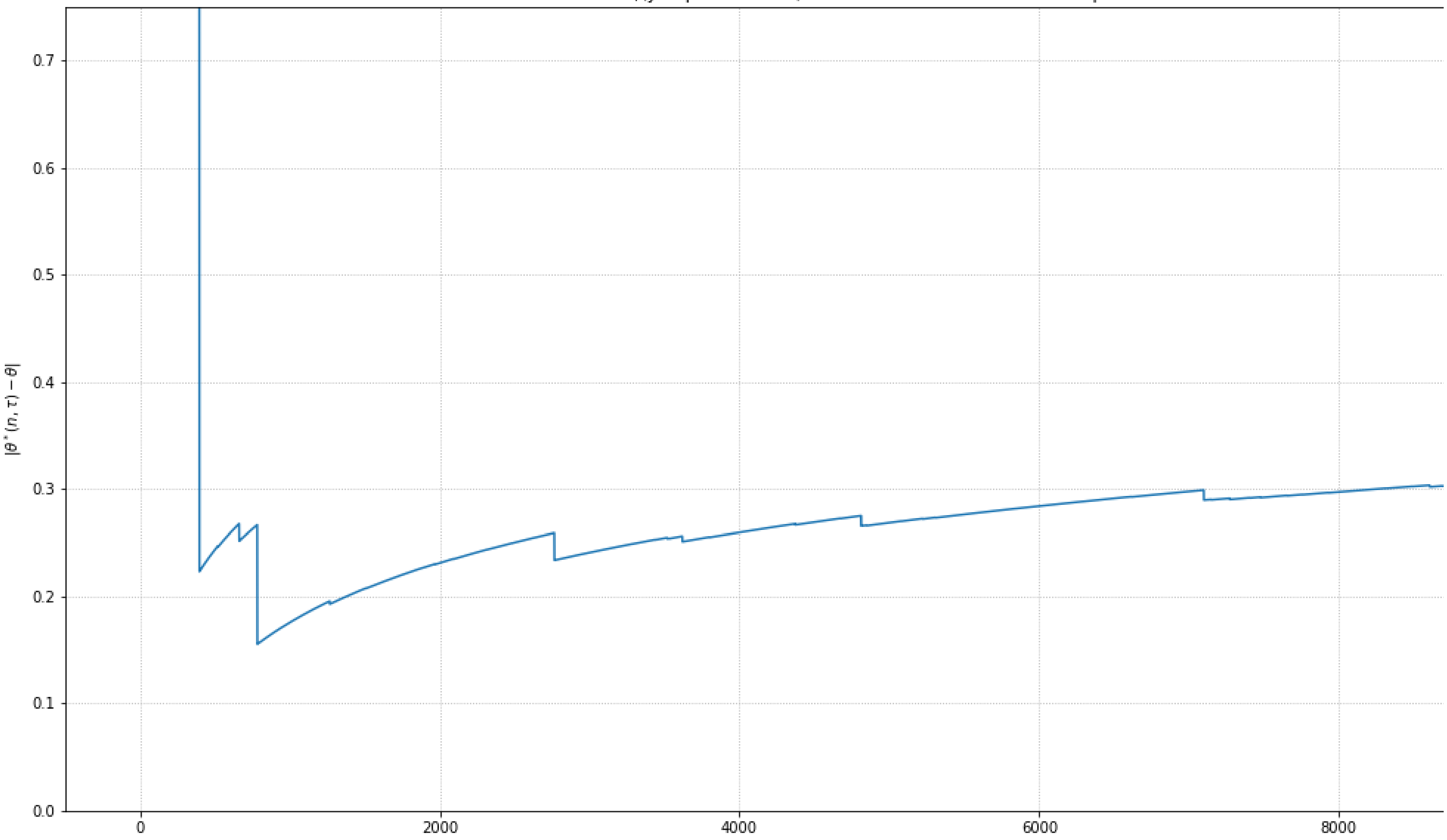


Модуль разности оценки и истинного значения θ при k=13

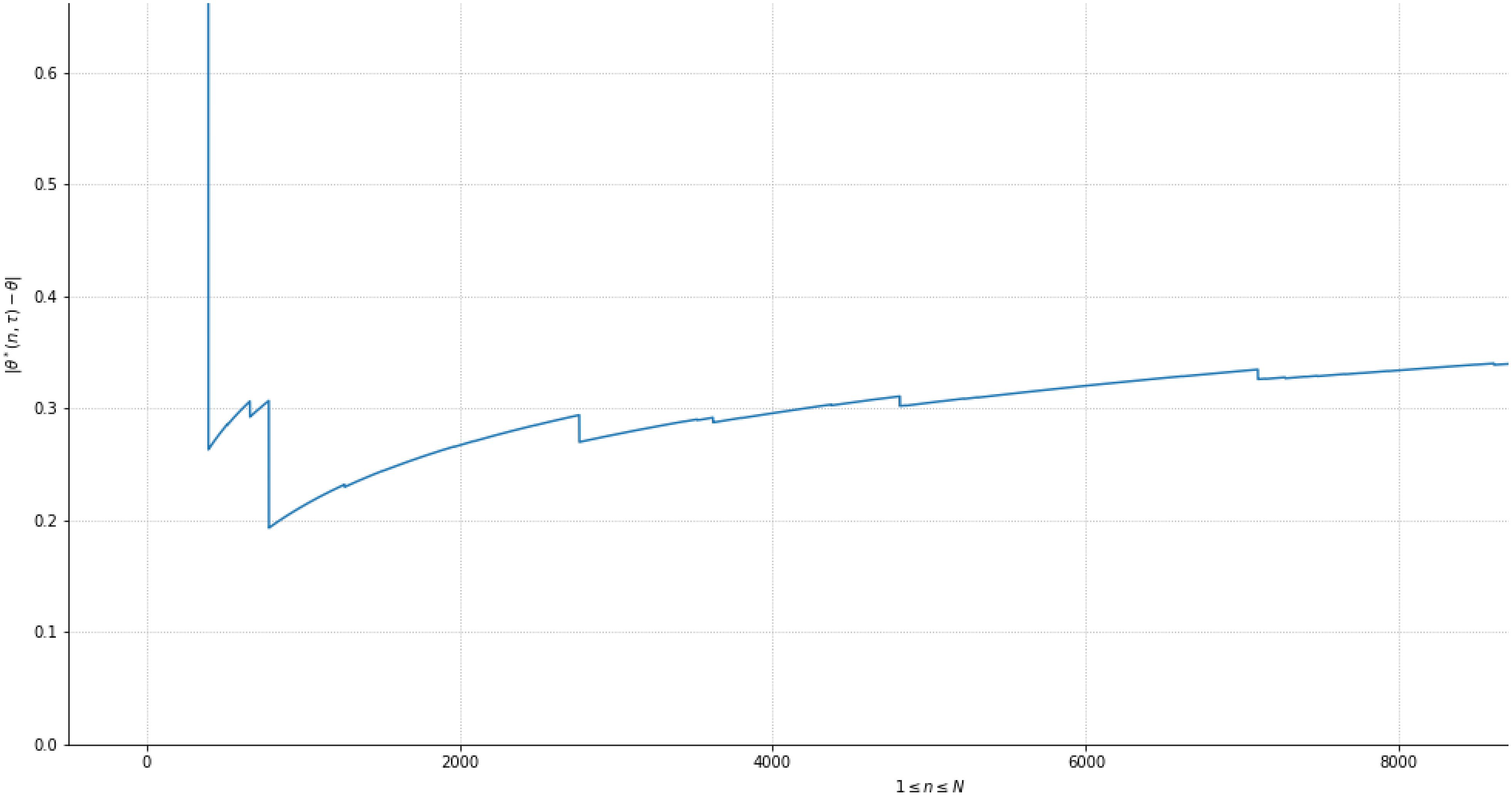


$1 \leq n \leq N$

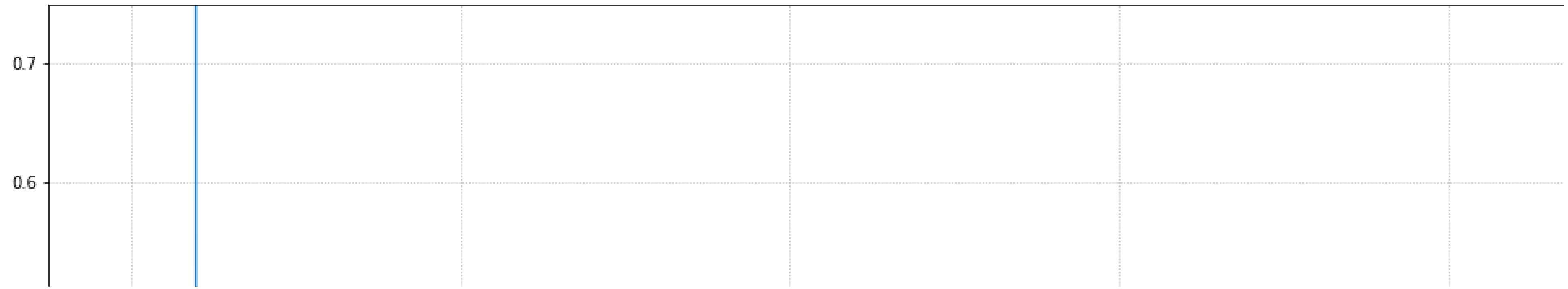
Модуль разности оценки и истинного значения θ при k=14

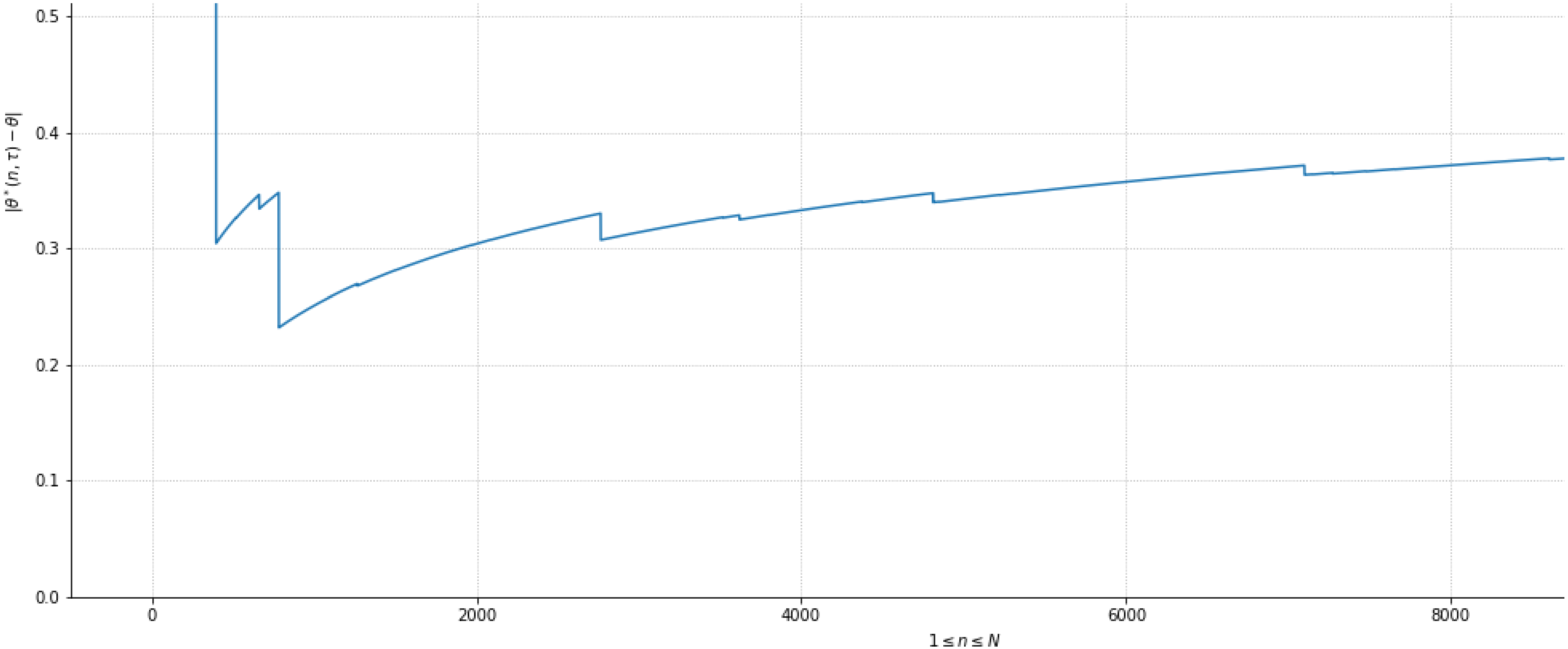


Модуль разности оценки и истинного значения θ при k=15

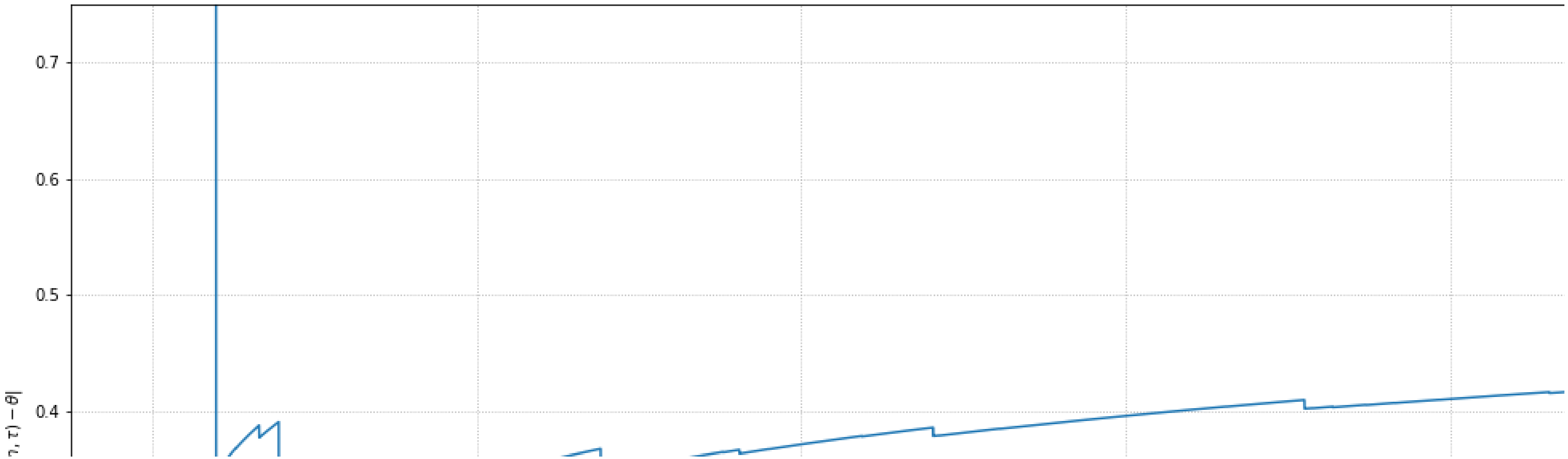


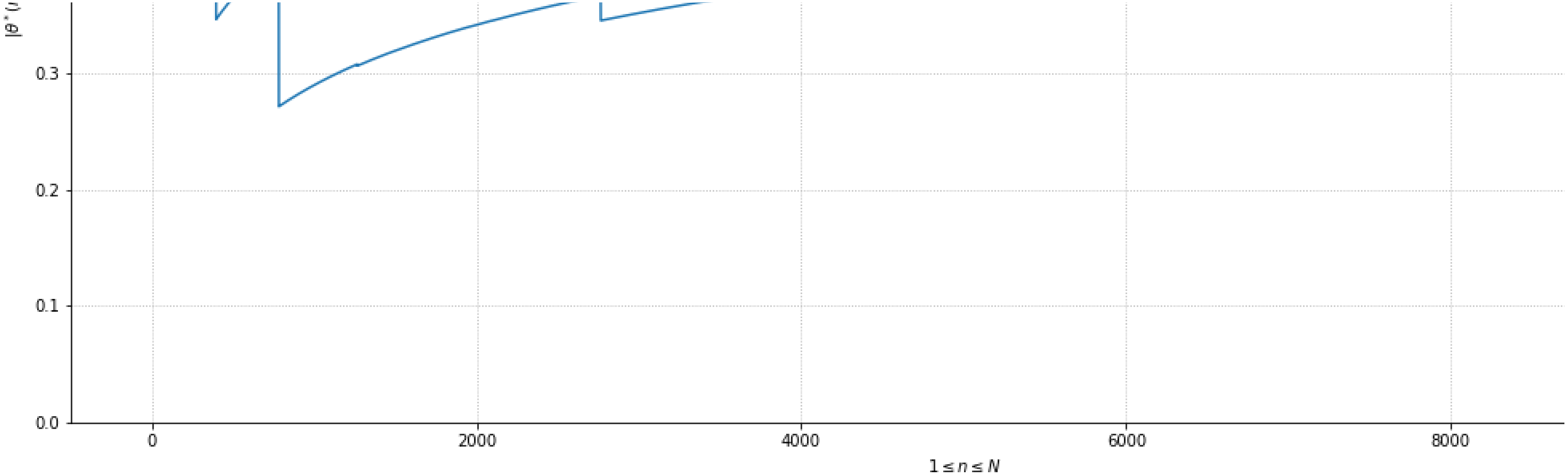
Модуль разности оценки и истинного значения θ при k=16



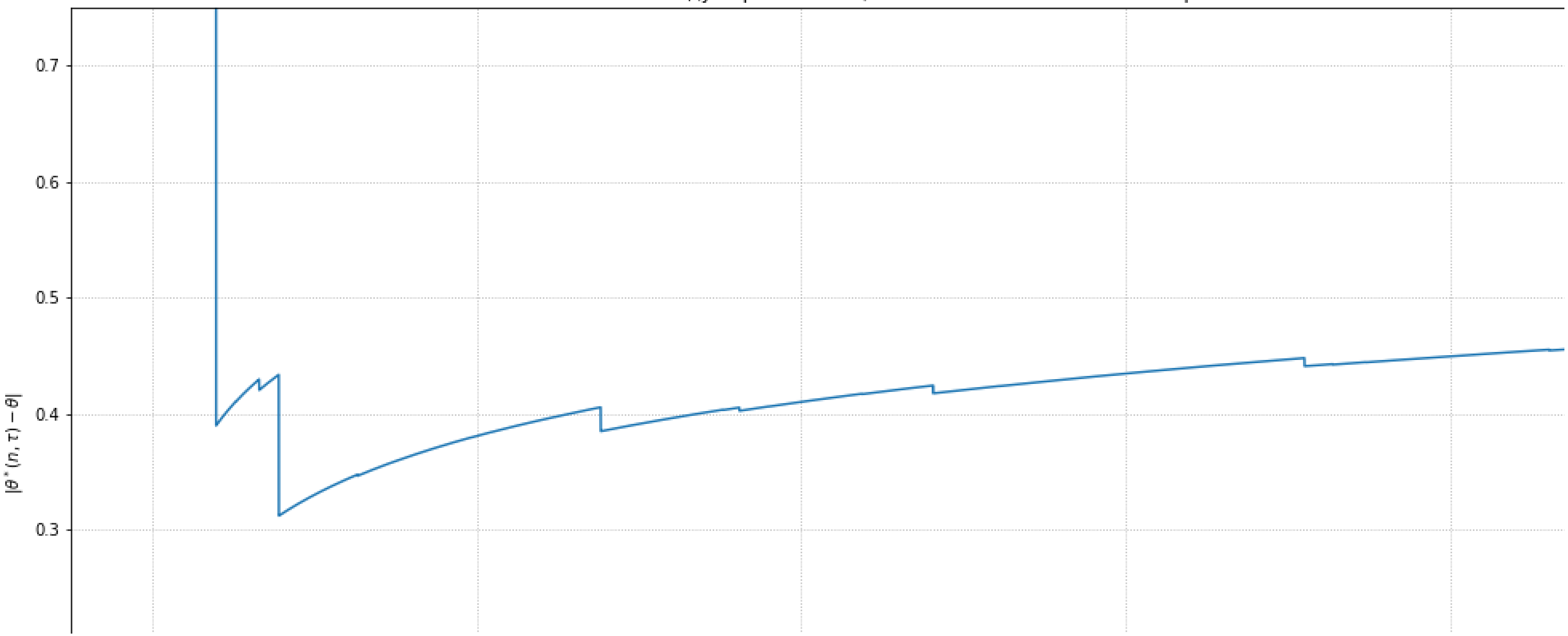


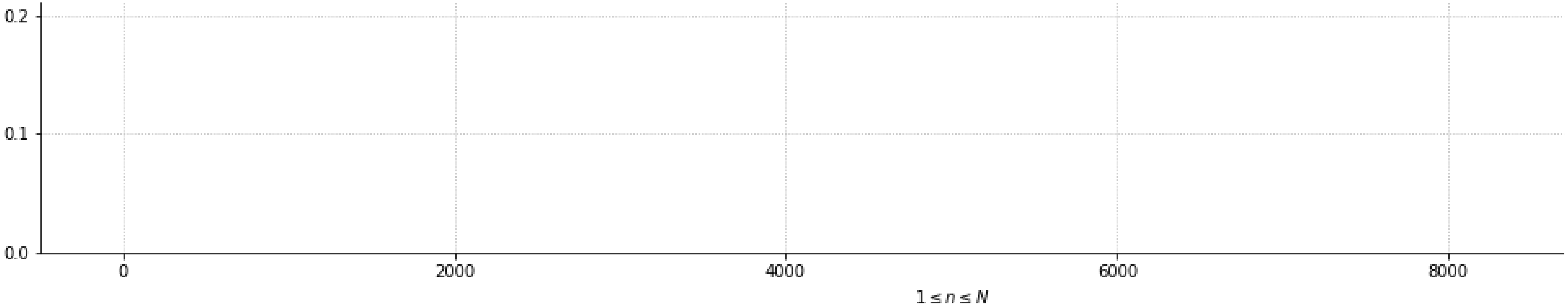
Модуль разности оценки и истинного значения θ при k=17

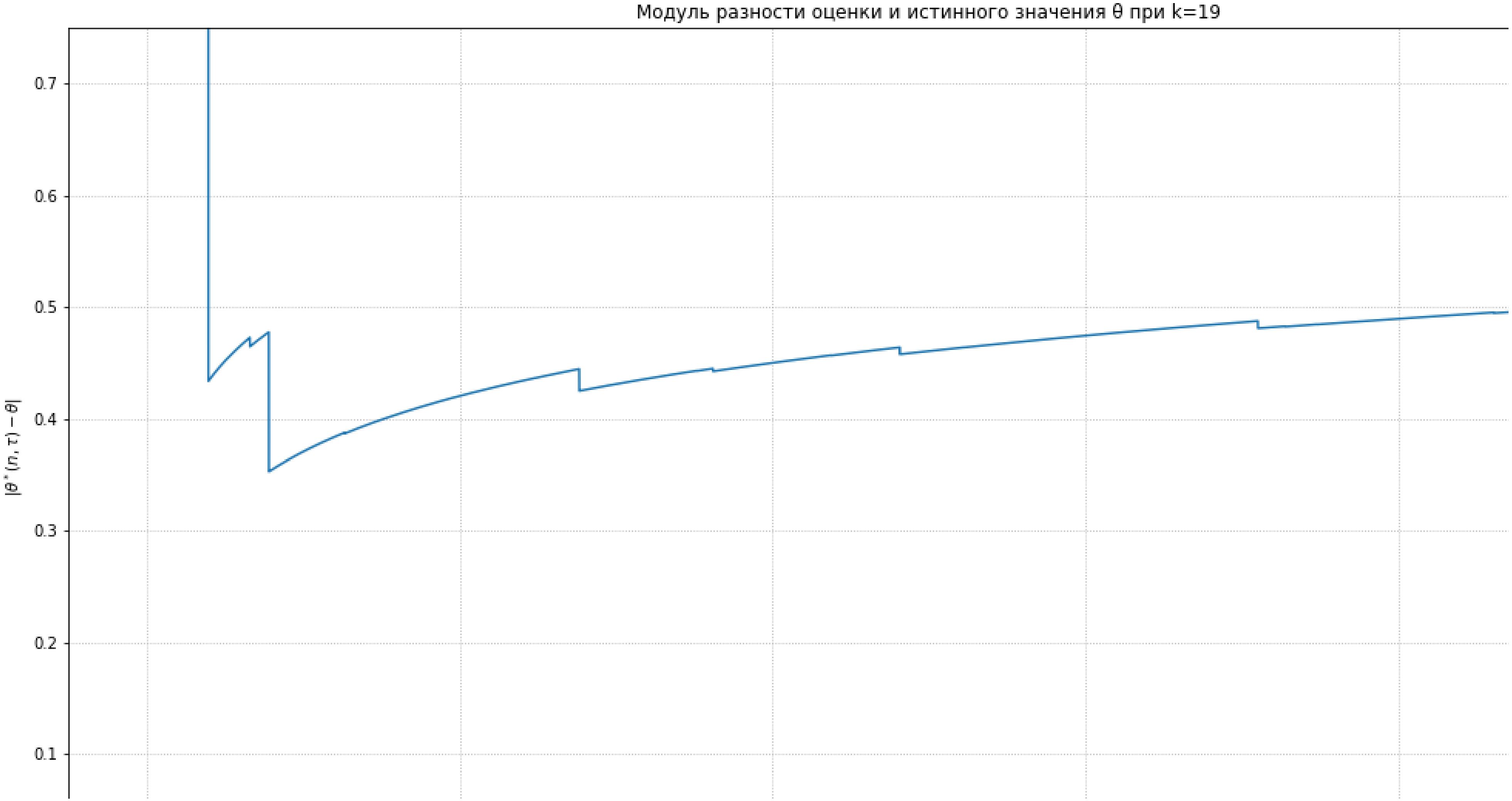


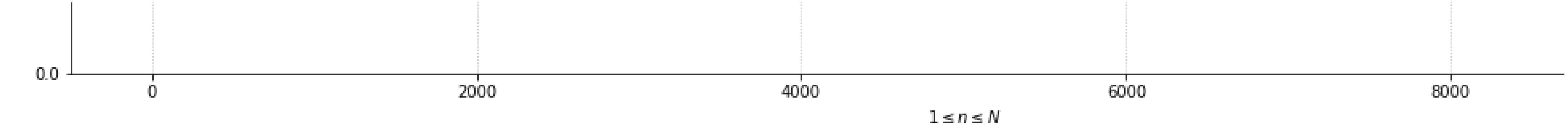


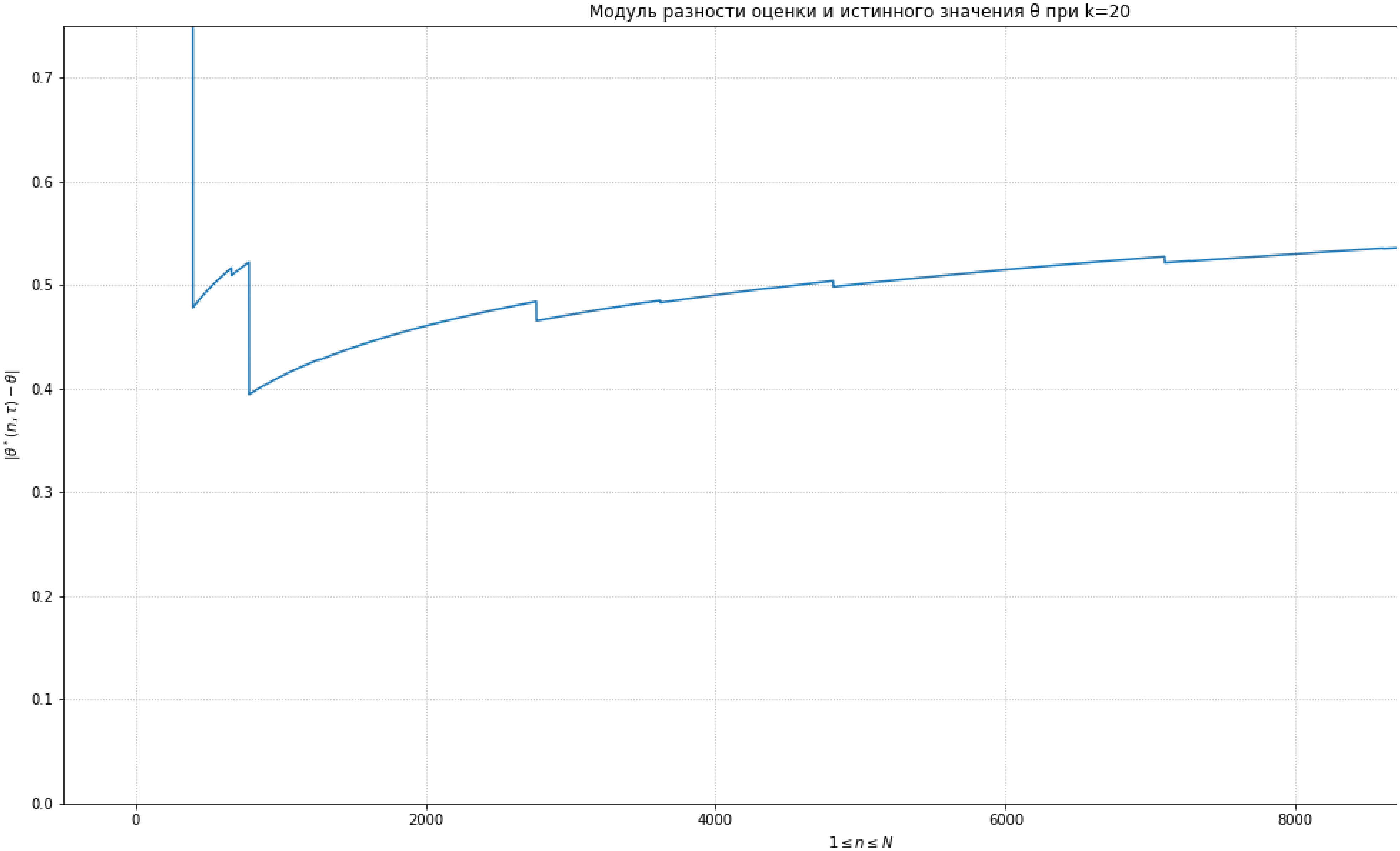
Модуль разности оценки и истинного значения θ при k=18




0.2
0.1
0.0
0
2000
4000
6000
8000
$1 \leq n \leq N$


Модуль разности оценки и истинного значения θ при k=19
$|\theta^*(n, \tau) - \theta|$
0.7
0.6
0.5
0.4
0.3
0.2
0.1


0.0
0
2000
4000
6000
8000
$1 \leq n \leq N$


Модуль разности оценки и истинного значения θ при k=20
$|\theta^*(n,\tau) - \theta|$
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2000
4000
6000
8000
$1 \leq n \leq N$

Как видим, достаточно рассмотреть только при $k \leq 5$

```
max_k = 5
param_estimation = {
  k: np.fromiter(map(
      lambda n: (factorial(k) / np.mean(observations[:n+1]**k)) ** (1/k),
      np.arange(0, N, 1)), dtype=np.float64)
    for k in range(1, max_k+1)
}
plt.figure(figsize=(20, 10))
plt.ylim((0., .75))
plt.title("Модуль разности оценки и истинного значения θ при $k \leq 5$")
plt.xlabel('$1\leq n \leq N$')
plt.ylabel('$|θ^*(n, \\tau) - θ|$')
for k in range(1, max_k+1):
    plt.plot(
      np.linspace(0, N, N),
      np.abs(param_estimation[k] - THETA),
      label=("k=%d" % k)
    )
plt.grid(ls=':')
plt.legend()
plt.show()
```