

Распараллеливание вычисления интеграла

Варивода Иван Алексеевич, Давыдов Алексей Андреевич, студенты 4 курса кафедры прикладной кибернетики СПбГУ varivoda_ivan@mail.ru, alex_davidov95@mail.ru

Аннотация

Существует огромное количество методов для приближенного вычисления интеграла. Одним из главных требований, предъявляемых к схеме, является скорость вычисления. Раньше математики разрабатывали новые модификации методов, для того, чтобы выиграть хоть немного времени. В наши дни существуют мощные средства для повышения скорости вычисления, например, распараллеливание процесса.

В данной работе рассматривается распараллеливание вычислений интеграла с помощью средств Open MP.

Ключевые слова: *OpenMP, распараллеливание, вычисление кратного интеграла*

Пусть a, b — вещественные числа, где $b > a$, n, N — натуральные числа. Возьмем $H = \frac{b-a}{N}$, $h = \frac{H}{n}$. Положим $x_k = a + kH$, $k = 0, 1, \dots, N$.

Представляя интеграл в виде

$$\int_a^b f(x)dx = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} f(x)dx \quad (1)$$

и используя формулу Ньютона-Котеса

$$\int_{x_k}^{x_{k+1}} f(x)dx \approx h \sum_{k=0}^n B_k^n f(x_k), \quad (2)$$

где

$$B_k^n = \frac{(-1)^{n-k}}{nk!(n-k)!} \int_0^n \frac{t(t-1) \cdots (t-n)}{t-k} dt \quad (3)$$

получаем формулу

$$\int_a^b f(x)dx \approx H \sum_{k=0}^n B_k^n \sum_{l=0}^N f(x_k^l). \quad (4)$$

Здесь использовано обозначение $x_k^l = a + Hl + hk$, $k = 0, 1, \dots, N$, $l = 0, 1, \dots, n$. В силу свойства $B_j^n = B_{n-j}^n$ можем переписать (3) при нечетном n в виде

$$\int_a^b f(x)dx \approx H \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} B_k^n \sum_{l=0}^N (f(x_k^l) + f(x_{n-k}^l)). \quad (5)$$

При четном n имеем

$$\int_a^b f(x)dx \approx H \sum_{k=0}^{\frac{n}{2}-1} B_k^n \sum_{l=0}^N (f(x_k^l) + f(x_{n-k}^l)) + HB_{\frac{n}{2}}^n \sum_{l=0}^N f(x_{\frac{n}{2}}^l). \quad (6)$$

Из формулы (1) получаем значения коэффициентов B_k^n

$$\begin{array}{lll} \text{Для } n=4: & B_0^4 = \frac{7}{90} & B_1^4 = \frac{32}{90} \quad B_2^4 = \frac{12}{90} \\ \text{Для } n=5: & B_0^5 = \frac{19}{288} & B_1^5 = \frac{75}{288} \quad B_2^5 = \frac{50}{288} \end{array}$$

Поэтому для $n = 4$ имеем следующую составную формулу

$$\int_a^b f(x)dx \approx H \left[\frac{7}{90} \sum_{l=0}^N (f(x_0^l) + f(x_4^l)) + \frac{32}{90} \sum_{l=0}^N (f(x_1^l) + f(x_3^l)) + \frac{12}{90} \sum_{l=0}^N (f(x_2^l) + f(x_3^l)) \right], \quad (7)$$

а для $n = 5$ получаем

$$\int_a^b f(x)dx \approx H \left[\frac{19}{288} \sum_{l=0}^N (f(x_0^l) + f(x_5^l)) + \frac{75}{288} \sum_{l=0}^N (f(x_1^l) + f(x_4^l)) + \frac{50}{288} \sum_{l=0}^N (f(x_2^l) + f(x_3^l)) \right]. \quad (8)$$

Ниже приведен пример кода для параллельного вычисления интеграла с $n = 4$. Распараллеливание происходит за счет синхронного вычисления трех сумм из (7).

```
#pragma omp parallel reduction(+: sum_1,sum_2,sum_3)
                               num_threads(THREAD_AMOUNT)
{
//Calculating the sum_1
#pragma omp for
for (int l = 0; l < N; l++){
sum_1 += f(a + l*H) + f(a + H + l*H);
}

//Calculating the sum_2
#pragma omp for
for (int l = 0; l < N; l++){
sum_2 += f(a + h + l*H) + f(a + (n - 1)*h + l*H);
}

//Calculating the sum_3
#pragma omp for
for (int l = 0; l < N; l++){
sum_3 += f(a + 2 * h + l*H);
}
}
```

В качестве подынтегральной функции была выбрана функция $f(x) = \sin(x) + x^5$
При вычислении были получены следующие результаты

График времени вычисления при n=4

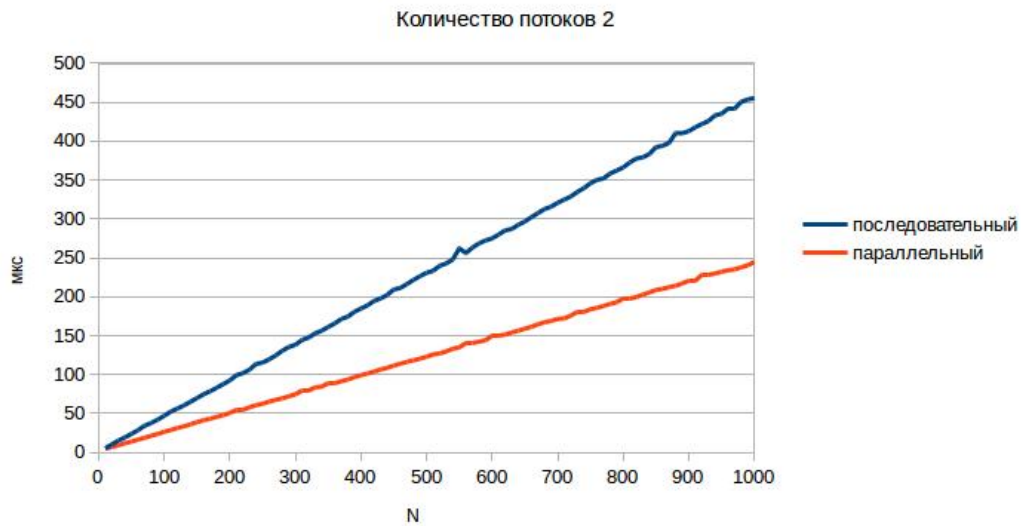


График времени вычисления при n=4

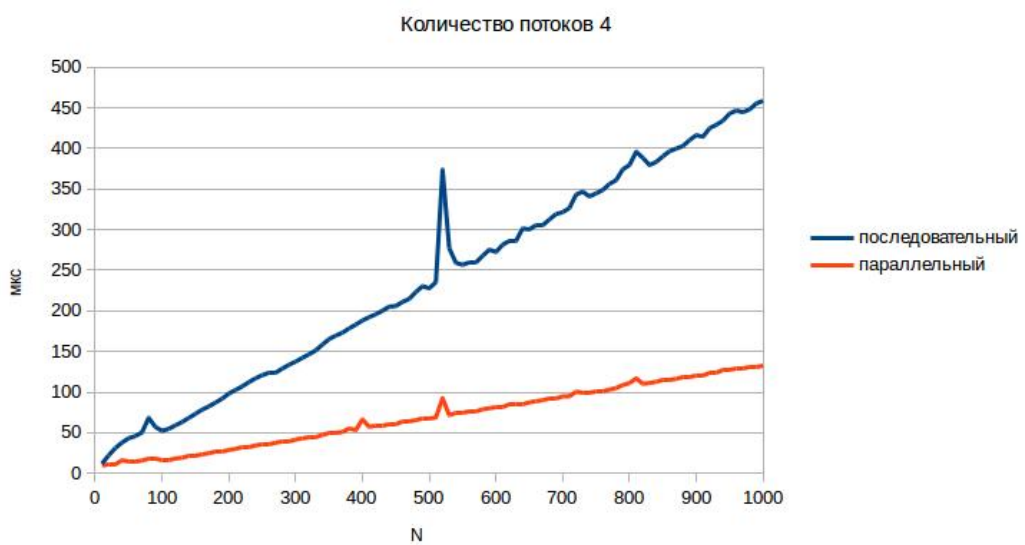


График для коэффициента при $n = 4$

Количество потоков 2

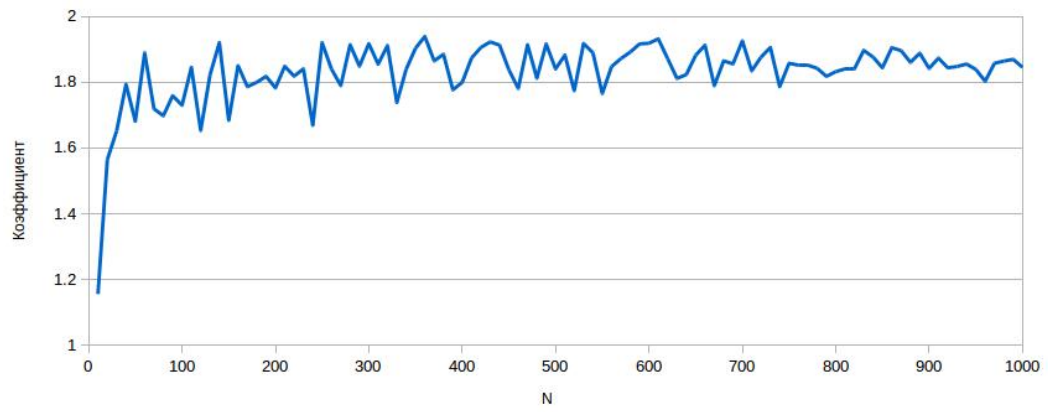


График для коэффициента при $n=4$

Количество потоков 4

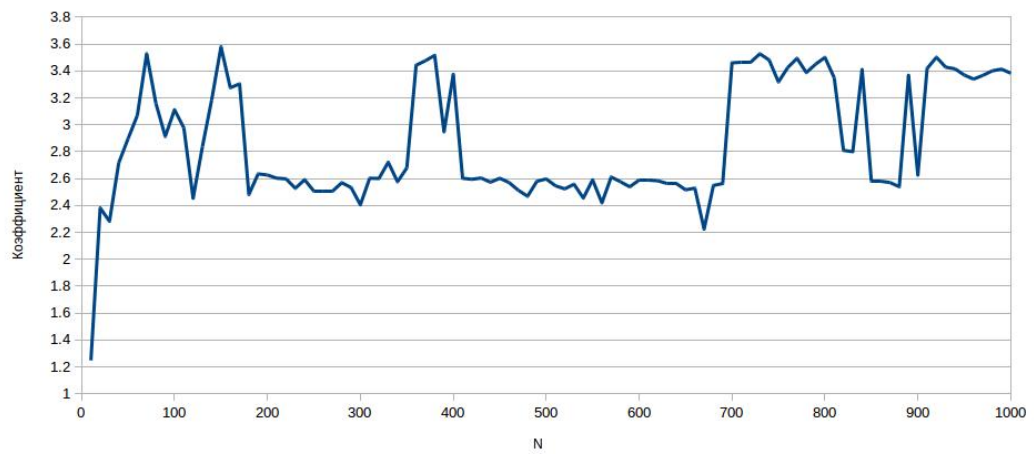


График времени вычисления при $n=5$

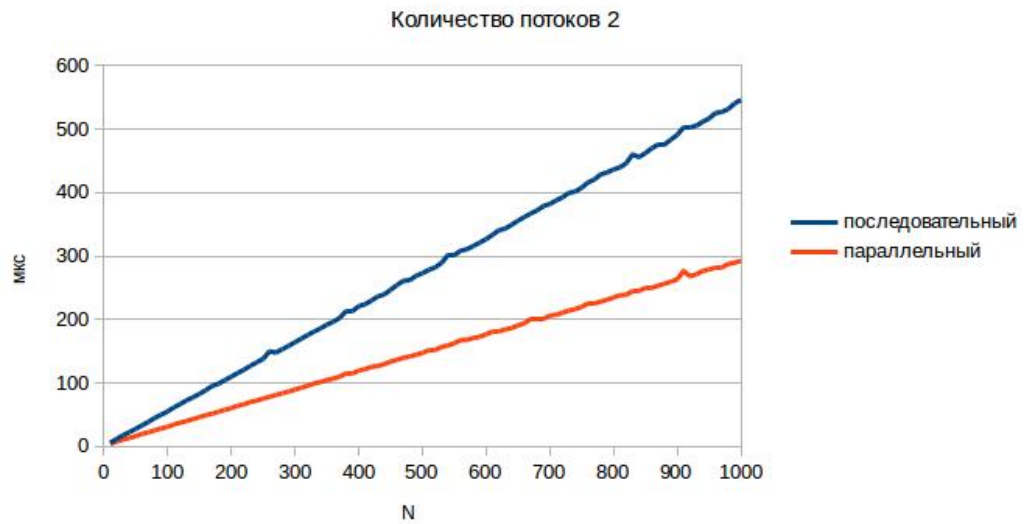
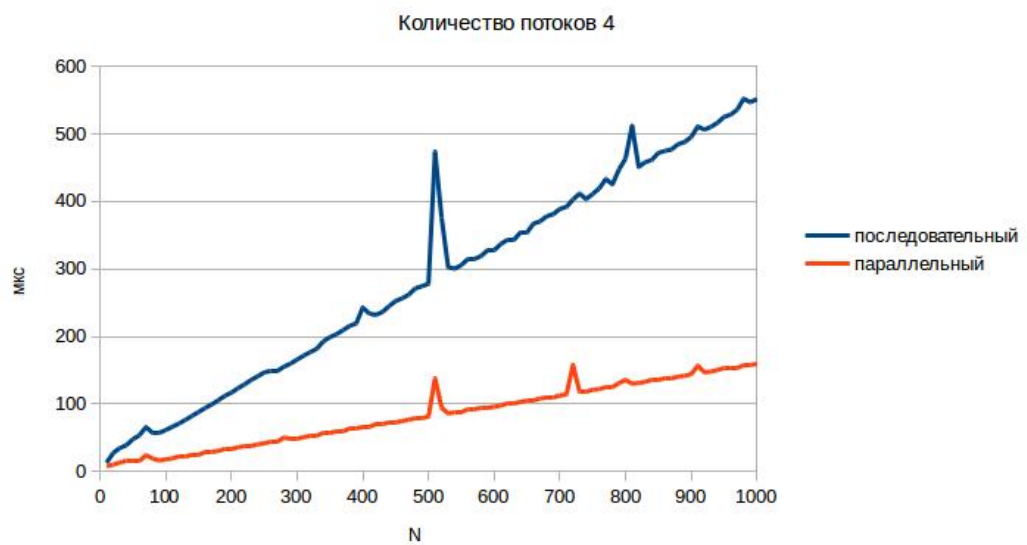
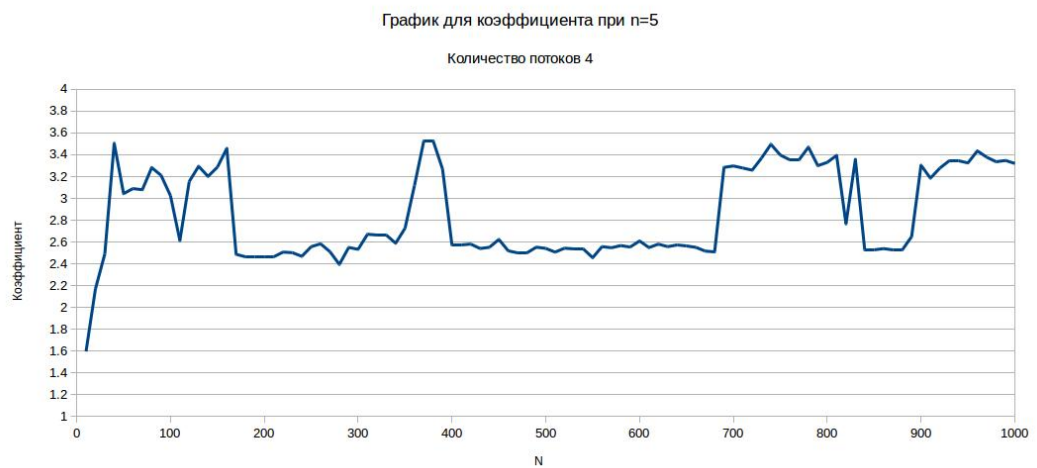
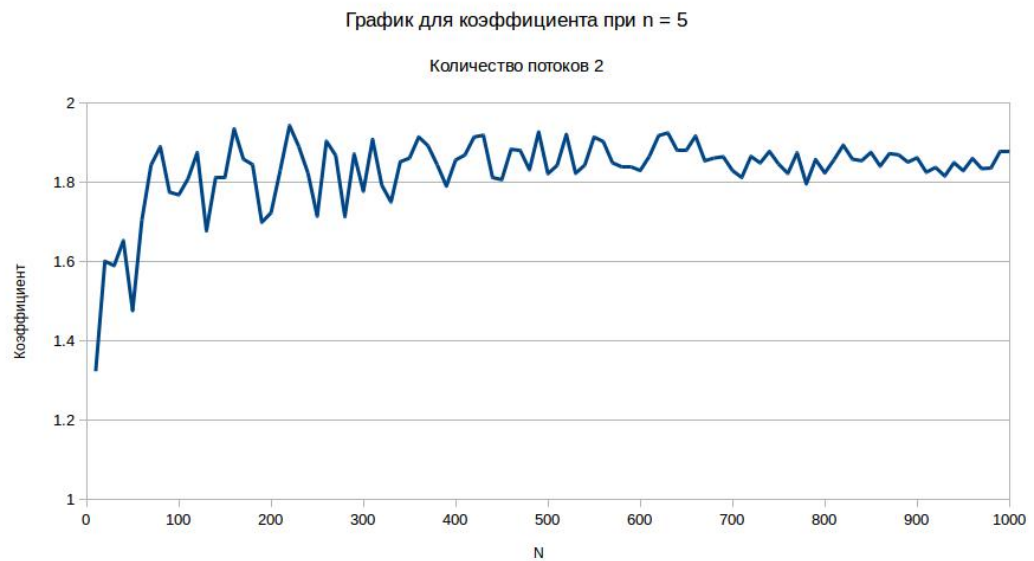


График времени вычисления при $n=5$





Список литературы

1. Крылов В.И. Приближенное вычисление интегралов. - Москва.: Изд-во Наука, 1967. -500 с.
2. Антонов А.С. Параллельное программирование с использованием технологии OpenMP: Учебное пособие. - Москва. Изд-во МГУ, 2009. -77 с.