Квадратурная формула средних прямоугольников:

2 > rectangle > C rectangle.c

```c
#include <stdio.h>
#include <mpi.h>

Codeium: Refactor | Explain | Generate Function Comment | ×
double f(double x) {
    return 1.0 / (1.0 + x * x);
}

Codeium: Refactor | Explain | Generate Function Comment | ×
int main(int argc, char *argv[]) {
    int n, myrank, nprocs, i;
    double h, local_sum = 0.0, total_sum = 0.0, x;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);

    if (myrank == 0) {
        printf("Enter the number of intervals (n): ");
        scanf("%d", &n);
    }

    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    h = 1.0 / n;
    for (i = myrank + 1; i <= n; i += nprocs) {
        x = h * (i - 0.5);
        local_sum += f(x);
    }

    local_sum *= 4.0 * h;
    MPI_Reduce(&local_sum, &total_sum, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

    if (myrank == 0) {
        printf("Approximated value of pi: %.16f\n", total_sum);
    }

    MPI_Finalize();
    return 0;
}
```

```
compilation terminated.
neo@stepan:~/Omsu/super_comp/2/rectangle$ mpicc -o rectangle rectangle.c
neo@stepan:~/Omsu/super_comp/2/rectangle$ ./rectangle
Enter the number of intervals (n): 4
Approximated value of pi: 3.1468005183939427
neo@stepan:~/Omsu/super_comp/2/rectangle$
```

Квадратурная формула трапеции:

2 > trapec > C trapec.c

```c
#include <stdio.h>
#include <mpi.h>

Codeium: Refactor | Explain | Generate Function Comment | ×
double f(double x) {
    return 1.0 / (1.0 + x * x);
}

Codeium: Refactor | Explain | Generate Function Comment | ×
int main(int argc, char *argv[]) {
    int n, myrank, nprocs, i;
    double h, local_sum = 0.0, total_sum = 0.0, x;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);    Ctrl+I Codeium Command

    // Процесс с рангом 0 запрашивает количество интервалов
    if (myrank == 0) {
        printf("Enter the number of intervals (n): ");
        scanf("%d", &n);
    }

    // Передача значения n всем процессам
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    h = 1.0 / n;  // Ширина интервала

    // Вычисление частичных сумм на каждом процессе
    for (i = myrank; i < n; i += nprocs) {
        x = i * h;
        if (i == 0 || i == n - 1) {
            local_sum += f(x);  // Границы добавляются без умножения
        } else {
            local_sum += 2 * f(x);  // Внутренние узлы умножаются на 2
        }
    }

    local_sum *= h / 2.0;  // Умножение на h/2 в формуле трапеции

    // Суммирование всех локальных результатов в процесс 0
    MPI_Reduce(&local_sum, &total_sum, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

    // Процесс 0 выводит результат
    if (myrank == 0) {
        printf("Approximated value of pi: %.16f\n", total_sum);
    }

    MPI_Finalize();
    return 0;
}
```

```
neo@stepan:~/Omsu/super_comp/2/trapec$ ./trpec
Enter the number of intervals (n): 4
Approximated value of pi: 0.6402941176470588
neo@stepan:~/Omsu/super_comp/2/trapec$
```

Квадратурная формула Симпсона:

2 > simp > C simp.c

```c
#include <stdio.h>
#include <mpi.h>

Codeium: Refactor | Explain | Generate Function Comment | ×
double f(double x) {
    return 1.0 / (1.0 + x * x);
}

Codeium: Refactor | Explain | Generate Function Comment | ×
int main(int argc, char *argv[]) {
    int n, myrank, nprocs, i;
    double h, local_sum = 0.0, total_sum = 0.0, x;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);

    // Процесс с рангом 0 запрашивает количество интервалов
    if (myrank == 0) {
        printf("Enter the number of intervals (n, even only): ");
        scanf("%d", &n);
        if (n % 2 != 0) {
            printf("Error: n must be an even number.\n");
            MPI_Abort(MPI_COMM_WORLD, 1);
        }
    }

    // Передача значения n всем процессам
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    h = 1.0 / n;  // Ширина интервала

    // Вычисление частичных сумм на каждом процессе
    for (i = myrank; i <= n; i += nprocs) {
        x = i * h;
        if (i == 0 || i == n) {
            local_sum += f(x);  // Границы добавляются без умножения
        } else if (i % 2 == 0) {
            local_sum += 2 * f(x);  // Узлы с чётным индексом умножаются на 2
        } else {
            local_sum += 4 * f(x);  // Узлы с нечётным индексом умножаются на 4
        }
    }

    local_sum *= h / 3.0;  // Умножение на h/3 в формуле Симпсона

    // Суммирование всех локальных результатов в процесс 0
    MPI_Reduce(&local_sum, &total_sum, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

    // Процесс 0 выводит результат
    if (myrank == 0) {
        printf("Approximated value of pi: %.16f\n", total_sum);
    }

    MPI_Finalize();
    return 0;
}
```

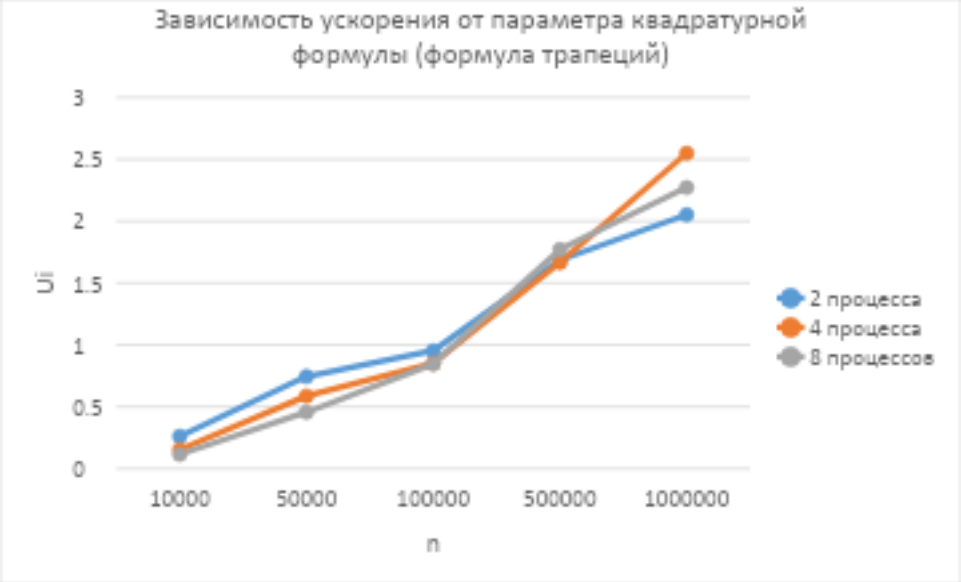
Квадратурная формула средних прямоугольников			
Значение n	Время вычислений (в сек.)	Точность вычислений	Относительное ускорение U_i
Количество параллельных процессов = 2			
10000	0,0006104	1,667 * 10 ^ -9	0,2785
50000	0,0006906	6,667 * 10 ^ -11	0,9430
100000	0,0011883	1,667 * 10 ^- 11	0,6324
500000	0,0046835	7,491 * 10 ^ -13	0,7656
1000000	0,0073919	1,643 * 10 ^ -13	1,0993
Количество параллельных процессов = 4			
10000	0,0011248	1,37721* 10 ^- 11	0,1511
50000	0,0011979	6,66613* 10 ^- 11	0,5437
100000	0,0014603	1,66747* 10 ^-11	0,5146
500000	0,0028966	6,62581* 10 ^-13	1,2380
1000000	0,0027612	2,08722* 10 ^-13	2,9431
Количество параллельных процессов = 8			
10000	0,0014133	1,66667* 10 ^-9	0,1202
50000	0,0015270	6,66667* 10 ^-11	0,4265
100000	0,0016197	1,66613* 10 ^-11	0,4639
500000	0,0021473	6,59917* 10 ^-13	1,6701
1000000	0,0029030	1,71418* 10 ^-13	2,7993

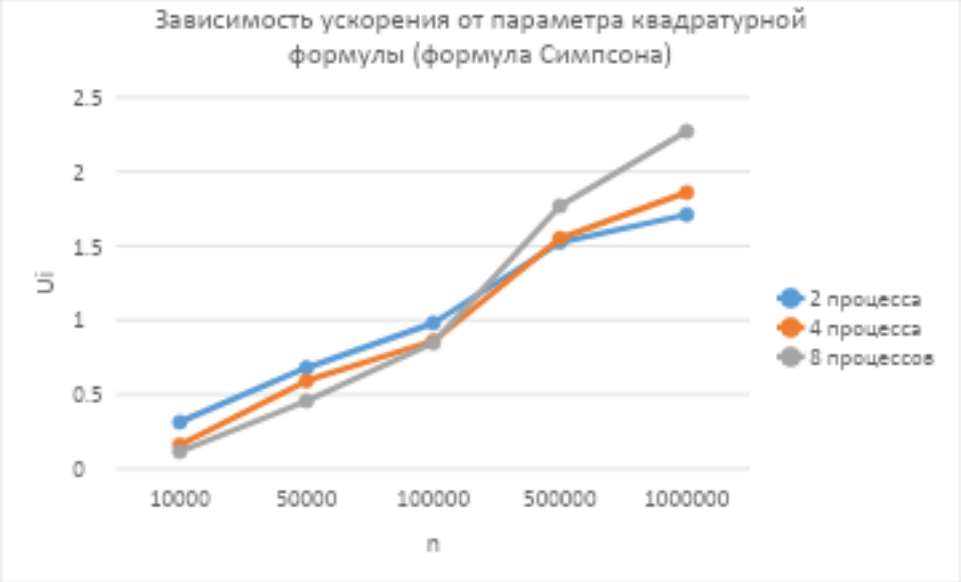
Квадратурная формула трапеций			
Значение n	Время вычислений (в сек.)	Точность вычислений	Относительное ускорение U_i
Количество параллельных процессов = 2			
10000	0,0006422	1,667 * 10 ^ -9	0,2612
50000	0,0008811	6,667 * 10 ^ -11	0,7444
100000	0,0013055	1,667 * 10 ^- 11	0,9523
500000	0,0023229	7,491 * 10 ^ -13	1,6849
1000000	0,0047626	1,643 * 10 ^ -13	2,0521
Количество параллельных процессов = 4			
10000	0,0010885	1,66667* 10 ^- 9	0,1541
50000	0,001117	6,66613* 10 ^- 11	0,5871
100000	0,0014603	1,66747* 10 ^-11	0,8514
500000	0,0023505	6,62581* 10 ^-13	1,6651
1000000	0,0038342	2,08722* 10 ^-13	2,5490
Количество параллельных процессов = 8			

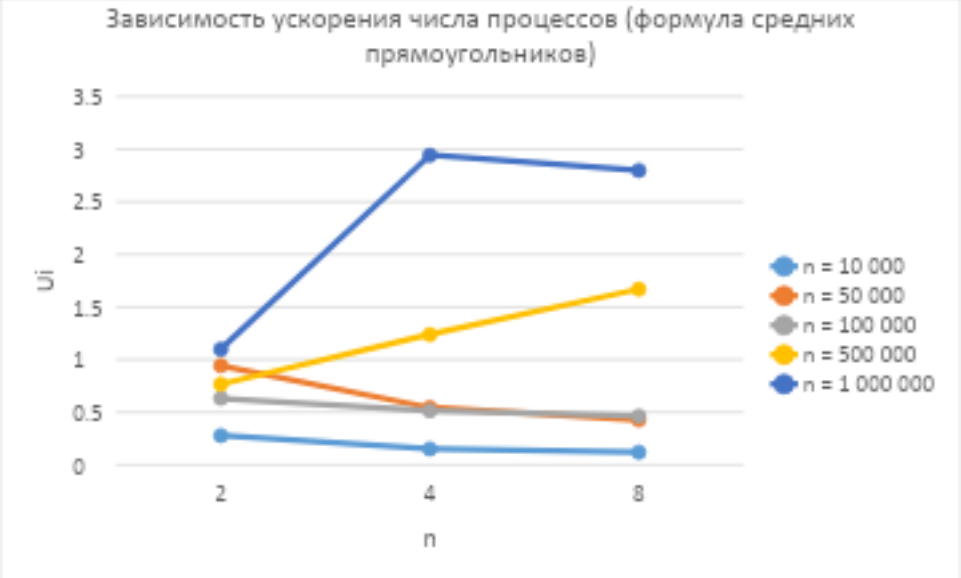
Квадратурная формула трапеций			
Значение n	Время вычислений (в сек.)	Точность вычислений	Относительное ускорение U_i
10000	0,0012515	1,66667* 10 ^-9	0,1340
50000	0,0013748	6,66667* 10 ^-11	0,6795
100000	0,0015333	1,66613* 10 ^-11	0,8108
500000	0,0021650	6,59917* 10 ^-13	1,8078
1000000	0,0030954	1,71418* 10 ^-13	3,1574

Квадратурная формула Симпсона			
Значение n	Время вычислений (в сек,)	Точность вычислений	Относительное ускорение U_i
Количество параллельных процессов = 2			
10000	0,0005093	1,667 * 10 ^ -9	0,3139
50000	0,0009652	6,667 * 10 ^ -11	0,6794
100000	0,0011785	1,667 * 10 ^- 11	0,9806
500000	0,0030027	7,491 * 10 ^ -13	1,5245
1000000	0,0047616	1,643 * 10 ^ -13	1,7105
Количество параллельных процессов = 4			
10,000	0,0009968	1,66667* 10 ^- 9	0,1604
50,000	0,0011069	6,66613* 10 ^- 11	0,5924
100000	0,0013437	1,66747* 10 ^-11	0,8600
500000	0,0029500	6,62581* 10 ^-13	1,5517
1000000	0,0043784	2,08722* 10 ^-13	1,8602
Количество параллельных процессов = 8			
10000	0,0013913	1,66667* 10 ^-9	0,1149
50000	0,0014368	6,66667* 10 ^-11	0,4564
100000	0,0013663	1,66613* 10 ^-11	0,8458
500000	0,0025869	6,59917* 10 ^-13	1,76960
1000000	0,0035856	1,71418* 10 ^-13	2,2715

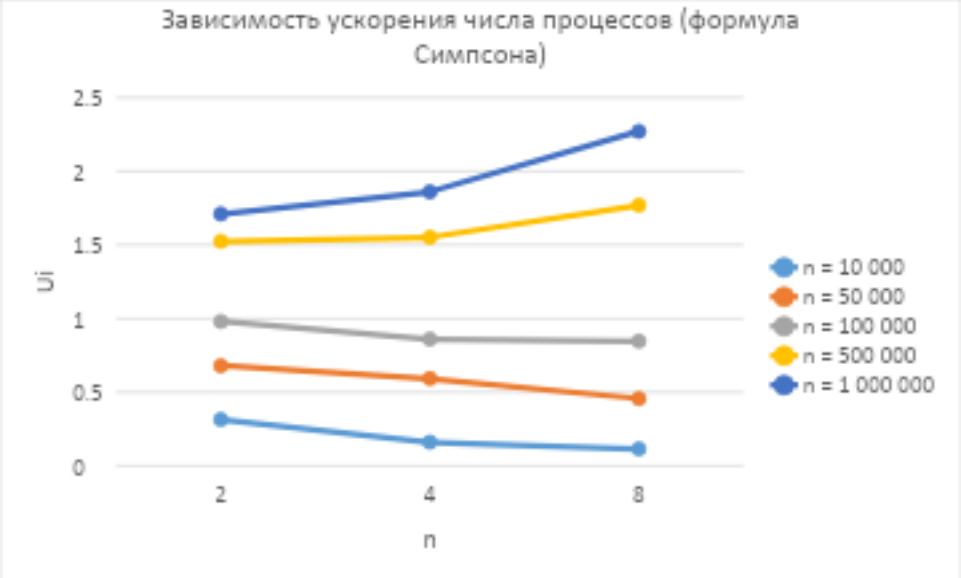

Зависимость ускорения от параметра квадратурной формулы (формула средних прямоугольников)
3.5
3
2.5
2
1.5
1
0.5
0
Ui
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов


Зависимость ускорения от параметра квадратурной формулы (формула трапеций)
3
2.5
2
1.5
1
0.5
0
Ui
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов

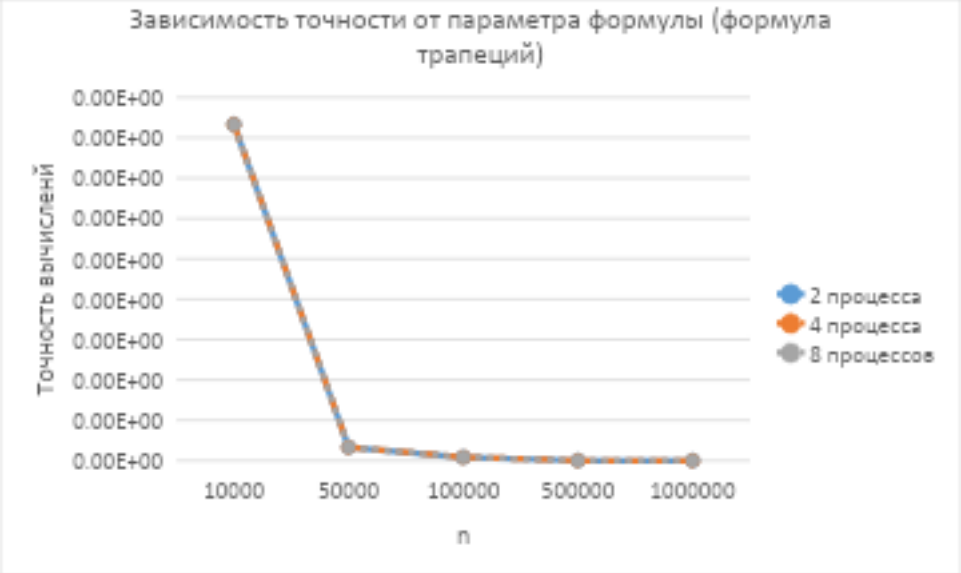

Зависимость ускорения от параметра квадратурной формулы (формула Симпсона)
2.5
2
1.5
1
0.5
0
Ui
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов

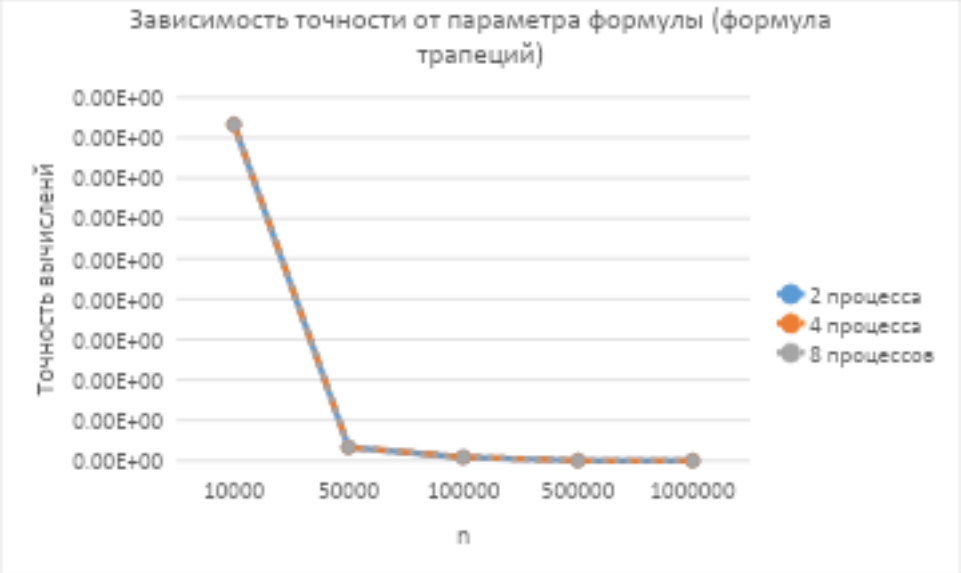

Зависимость ускорения числа процессов (формула средних прямоугольников)
3.5
3
2.5
2
1.5
1
0.5
0
Ui
2
4
8
n
n = 10 000
n = 50 000
n = 100 000
n = 500 000
n = 1 000 000

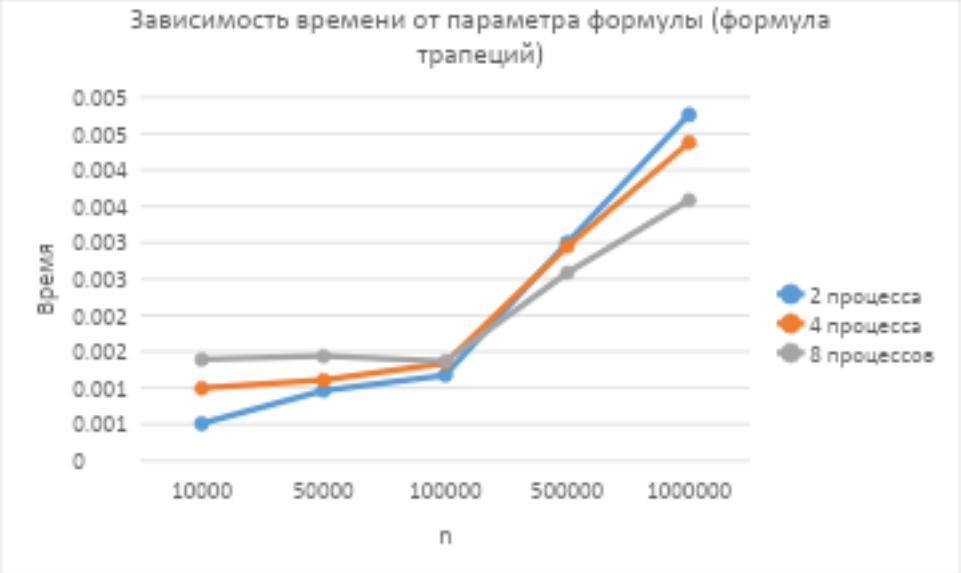

Зависимость ускорения числа процессов (формула трапеций)
3.5
3
2.5
2
1.5
1
0.5
0
Ui
2
4
8
n
n = 10 000
n = 50 000
n = 100 000
n = 500 000
n = 1 000 000


Зависимость ускорения числа процессов (формула Симпсона)
2.5
2
1.5
1
0.5
0
Ui
2
4
8
n
n = 10 000
n = 50 000
n = 100 000
n = 500 000
n = 1 000 000

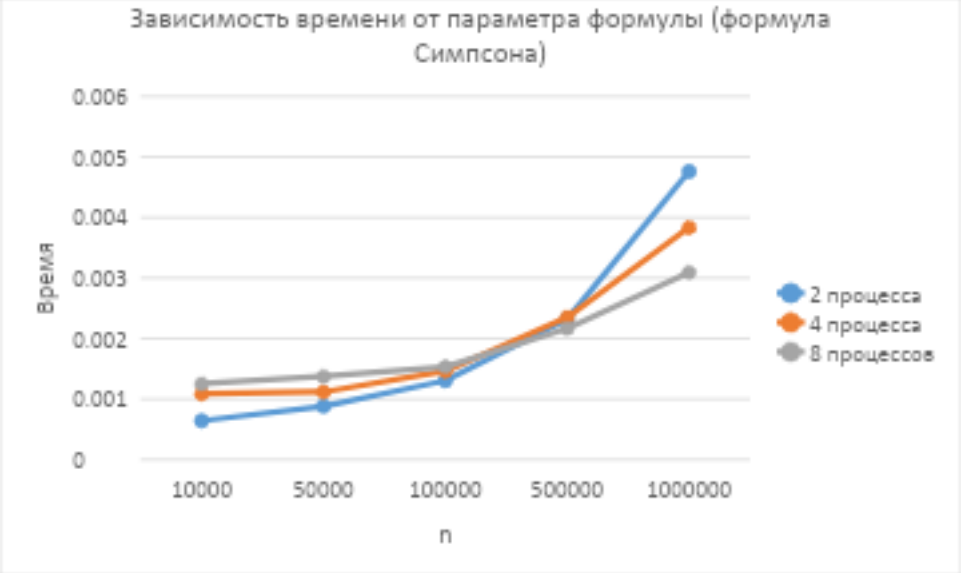

Зависимость точности от параметра формулы (формула средних прямоугольников)
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
Точность вычислений
10000
50000
100000
500000
1000000
n


Зависимость точности от параметра формулы (формула трапеций)
Точность вычислений
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов


Зависимость точности от параметра формулы (формула трапеций)
Точность вычислений
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
0.00E+00
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов


Зависимость времени от параметра формулы (формула трапеций)
0.005
0.005
0.004
0.004
0.003
0.003
0.002
0.002
0.001
0.001
0
Время
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов


Зависимость времени от параметра формулы (формула средних прямоугольников)
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
Время
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов


Зависимость времени от параметра формулы (формула Симпсона)
0.006
0.005
0.004
0.003
0.002
0.001
0
Время
10000
50000
100000
500000
1000000
n
2 процесса
4 процесса
8 процессов