



**Отчет по 3 заданию курса  
«Суперкомпьютерное моделирование и технологии»**

**Выполнил:**  
Студент 627 группы  
Федяшкин Максим Алексеевич  
Вариант 3

## Оглавление

Математическая постановка.....	3
Аналитическое решение.....	3
Числовой метод решения.....	4
Краткое описание работ по созданию гибридного решения.....	5
Создание MPI версии.....	5
Разбиение на кубы и создание топологии.....	5
Обмен гало.....	5
Генерация $u_0, u_1, \dots, u_n$ .....	5
Вычисление точного значения на шаге.....	5
Подсчет ошибки.....	5
Итерационный алгоритм.....	5
Гибридная версия.....	6
График решения разностной схемы.....	9
График погрешности.....	13
Результаты расчетов.....	17
Последовательная программа.....	17
$L_x = L_y = L_z = 1.0$ .....	17
$L_x = L_y = L_z = \text{Pi}$ .....	17
Графики.....	17
MPI версия.....	18
$L_x = L_y = L_z = 1.0$ .....	18
Графики для $L_x = L_y = L_z = 1.0$ .....	18
$L_x = L_y = L_z = \text{Pi}$ .....	19
Графики для $L_x = L_y = L_z = \text{Pi}$ .....	19
MPI + OpenMP.....	20
$L_x = L_y = L_z = 1.0$ .....	20
Графики для $L_x = L_y = L_z = 1.0$ .....	20
$L_x = L_y = L_z = \text{Pi}$ .....	21
Графики для $L_x = L_y = L_z = \text{Pi}$ .....	21

## Математическая постановка

В трехмерной заданной области

$$\Omega = [0 \leq x \leq L_x] \times [0 \leq y \leq L_y] \times [0 \leq z \leq L_z]$$

Необходимо найти решение  $u(x, y, z, t)$  для  $t \in (0, T]$  уравнения в частных производных:

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

С начальными условиями 1-го рода:

$$\begin{aligned} u(0, y, z, t) &= 0 & u(L_x, y, z, t) \\ u(x, y, 0, t) &= 0 & u(x, y, L_z, t) \end{aligned}$$

И начальными периодическими условиями:

$$u(x, 0, z, t) = u(x, L_y, z, t) \quad u_y(x, y, 0, t) = u_y(x, L_y, z, t)$$

## Аналитическое решение

$$u(x, y, z, t) = \phi(x, y, z) \cdot \pi \cdot \cos\left(t \cdot \sqrt{\frac{1}{L_x^2} + \frac{4}{L_y^2} + \frac{9}{L_z^2}}\right)$$

$$\phi(x, y, z) = \sin\left(\frac{1 \cdot \pi}{L_x} \cdot x\right) \cdot \sin\left(\frac{2 \cdot \pi}{L_y} \cdot y\right) \cdot \sin\left(\frac{3 \cdot \pi}{L_z} \cdot z\right)$$

Устойчивость схемы можно высчитать по формуле

$$\frac{c \cdot \tau}{h} \leq \frac{1}{\sqrt{(3)}} \Leftrightarrow K \geq \sqrt{(3)} \cdot N, \text{ где } \tau \cdot K = T$$

## Числовой метод решения

Введем на  $\Omega$  сетку  $w_{h\tau} = \bar{w}_h \times w_\tau$  пусть

$$T = T_0,$$

$$L_x = L_{x_0}, L_y = L_{y_0}, L_z = L_{z_0},$$

$$\bar{w}_h = \{(x_i = i \cdot h_x, y_j = j \cdot h_y, z_k = k \cdot h_z) \mid i, j, k = 0, 1, \dots, N \quad h_x \cdot N = L_x, h_y \cdot N = L_y, h_z \cdot N = L_z\},$$

$$w_\tau = \{t_n = n \cdot \tau, n = 0, 1, \dots, K \quad \tau \cdot K = T\}$$

Аппроксимация исходного уравнения

$$\frac{u_{ijk}^{n+1} - 2 \cdot u_{ijk}^n + u_{ijk}^{n-1}}{\tau^2} = \Delta_h u^n, \quad (x_i, y_j, z_k) \in w_h, \quad n = 1, 2, \dots, K-1$$

$$\Delta_h u^n = \frac{u_{i-1,j,k}^n - 2 \cdot u_{ijk}^n + u_{i+1,j,k}^n}{h^2} + \frac{u_{i,j-k}^n - 2 \cdot u_{ijk}^n + u_{i,j+1,k}^n}{h^2} + \frac{u_{i,j,k-1}^n - 2 \cdot u_{ijk}^n + u_{i,j,k+1}^n}{h^2}$$

$$u_{ijk}^0 = \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in w_h$$

$$u_{ijk}^1 = u_{ijk}^0 + \frac{\tau}{2} \cdot \Delta_h \cdot \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in w_h$$

Разностная аппроксимация для периодического условия по  $u$

$$u_{i0k}^{n+1} = u_{iNk}^{n+1}, \quad u_{i1k}^{n+1} = u_{i(N+1)k}^{n+1}$$

# Краткое описание работ по созданию гибридного решения

## Создание MPI версии

### Разбиение на кубы и создание топологии

С помощью метода `MPI_Dims_create` создается карта разбиения куба. По карте происходит создание виртуальной топологии, циклической только по  $u$ . Каждый процесс обрабатывает свою часть куба и хранит только ее и гало, под гало понимаются смежные стороны соседних кубов глубиной в 1. Соответственно в каждом измерении размерность подкуба увеличена на 2. Генерируется вспомогательная структура, которая хранит границы, смещения, список соседей, и другую информацию упрощающее отображение исходного куба на его часть с гало.

### Обмен гало

По каждому измерению четные процессы обмениваются гало с нечетными (если сосед есть). Затем по тому же измерению нечетные обмениваются с четными (если есть сосед). В итоге за 6 итераций происходит полный обмен гало.

### Генерация $u_0, u_1, \dots, u_n$

Определены методы которые позволяют вычислить  $u_0, u_1, u_n$  – решение разностной схемы на шаге  $n$ . Методы реализуют вычисление значения элементов подкуба (гало не меняются). Логика функций повторяет программную реализацию.

### Вычисление точного значения на шаге

Метод который возвращает точное значение решения на шаге в заданной точке. Используется формула точного аналитического решения.

### Подсчет ошибки

Вычисляется максимальная локальная ошибка на подкубе путем прохода по всем элементам подкуба и сравнением по модулю с точным аналитическим решением, Затем с помощью `MPI_reduce` на 0 процессе вычисляется максимальная ошибка и печатается в консоль.

### Итерационный алгоритм

Итерации происходят по временным шагам.

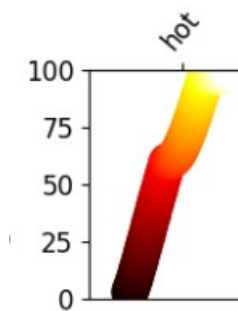
1. Вычисление  $u_n$ ,  $n = 0, \dots, K$
2. Подсчет и вывод ошибки
3. Обмен гало

## Гибридная версия

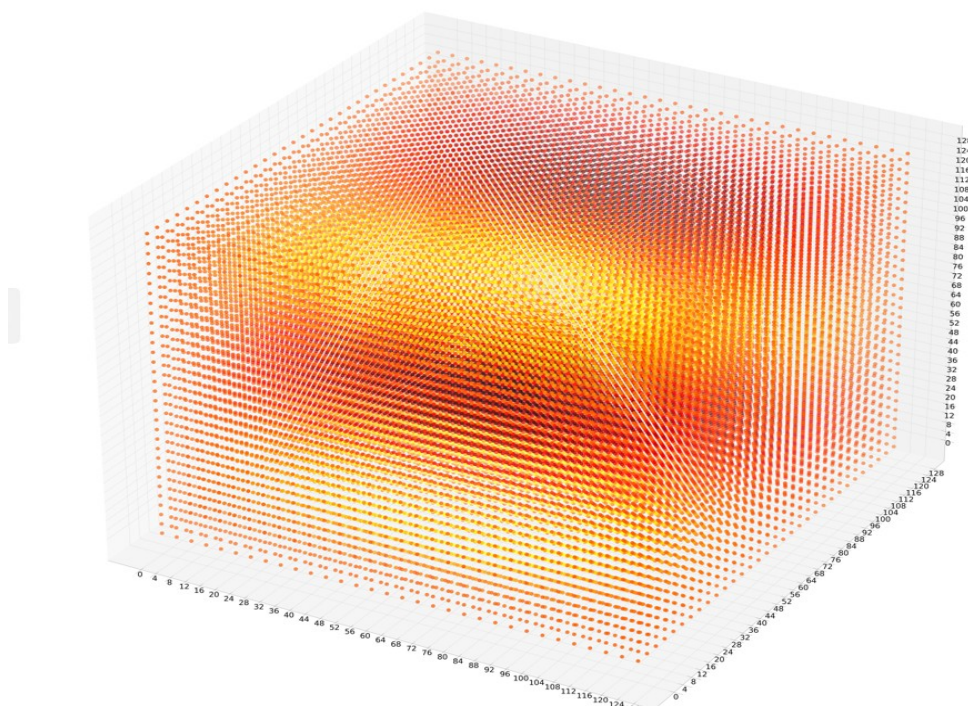
Все операции над матрицами в циклах являются не зависимыми по данным. С помощью соответствующей директивы `#pragma parallel for` происходит распараллеливание внешнего цикла, каждой нити своя часть матрицы. Исключение составляет поиск ошибки в подкубе, где необходимо создать временный массив в котором хранится максимальное значение части подкуба обрабатываемой нитью. После в массиве ищется максимальное значение и возвращается из метода. График аналитического решения

При построении графиков обнаружено, что визуальной разницы между шагами  $[t == 0, t == 20]$  нет. Поэтому представлены графики для  $t == 20$ . Для удобства отображаются не все точки а их часть (координаты которой кратны шагу), шаг указан в легенде графика (s).

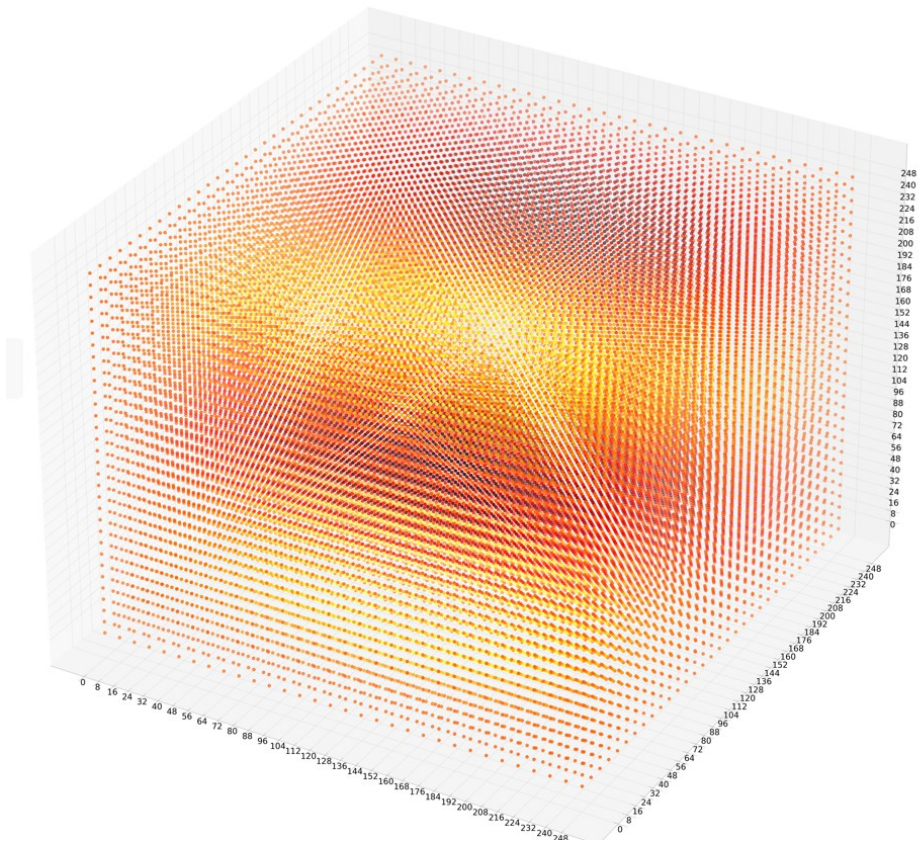
Диапазон:



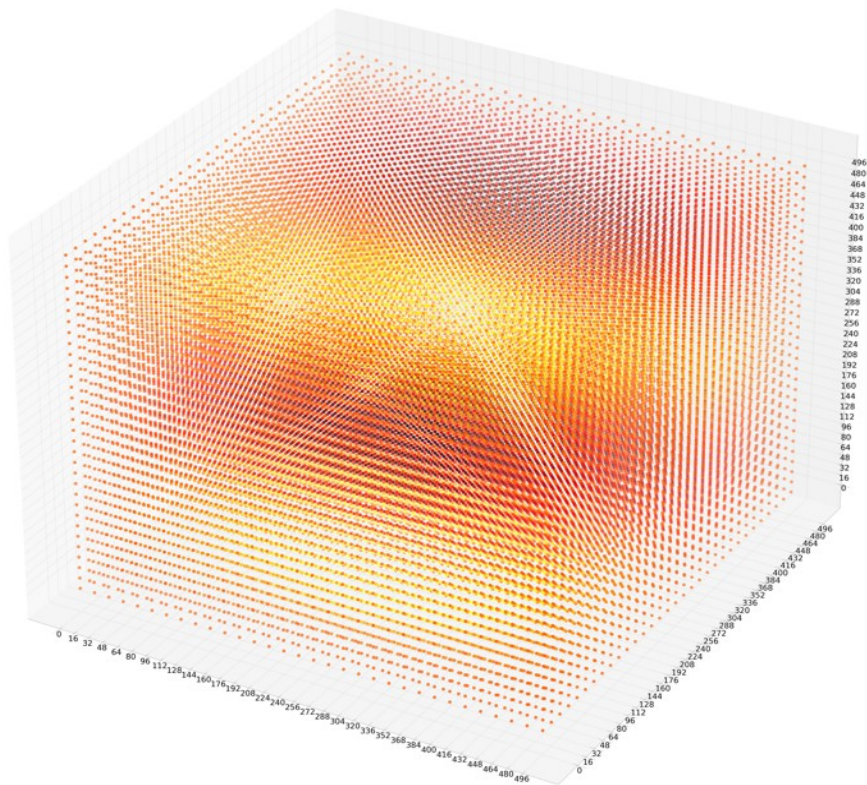
Аналитические значения принадлежат диапазону  $[-1,1]$  и отображаются в  $[0, 100]$



Фигура 1:  $L=1, N=128, K=220, T=1, t=20, s=4$

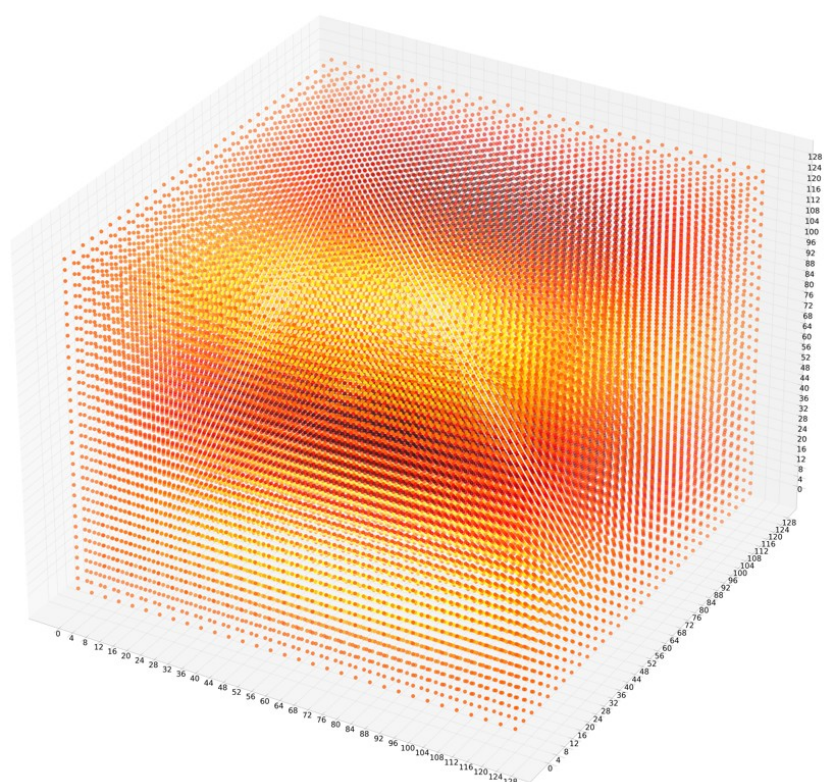


Фигура 2:  $L=1$ ,  $N=256$ ,  $K=422$ ,  $T=1$ ,  $t=20$ ,  $s=16$

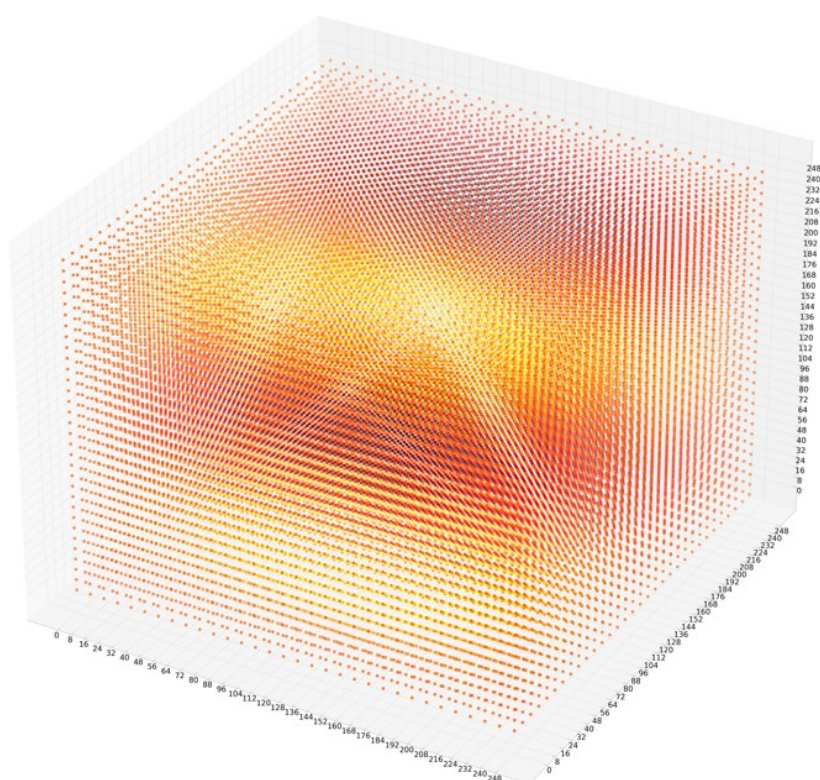


Фигура 3:  $L=1$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  $s=16$



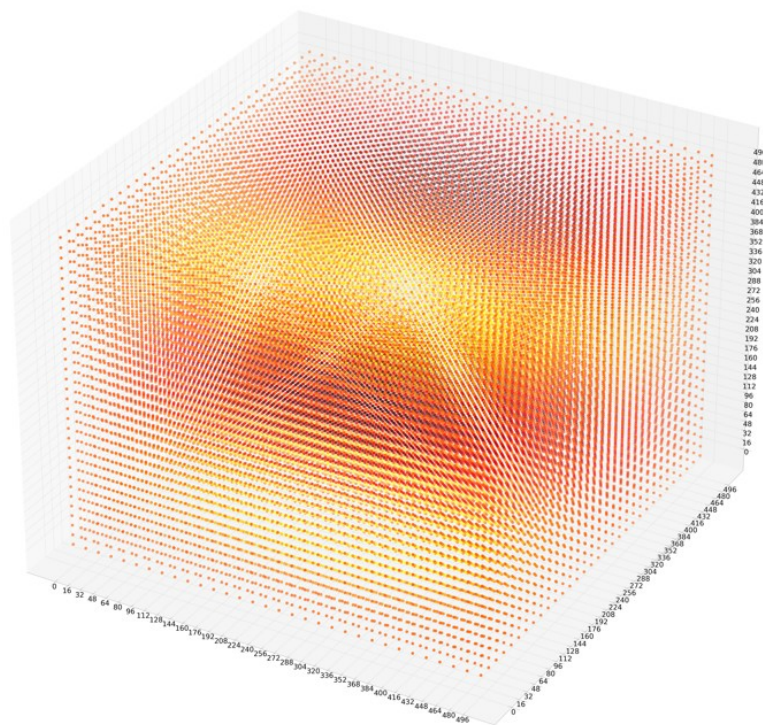


Фигура 4:  $L=3.14$ ,  $N=128$ ,  $K=220$ ,  $T=1$ ,  $t=20$ ,  $s=4$



Фигура 5:  $L=3.14$ ,  $N=256$ ,  $K=442$ ,  $T=1$ ,  $t=20$ ,  $s=8$



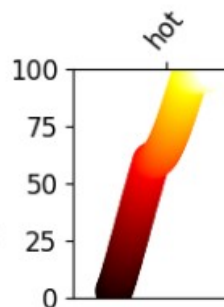


Фигура 6:  $L=3.14$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  $s=16$

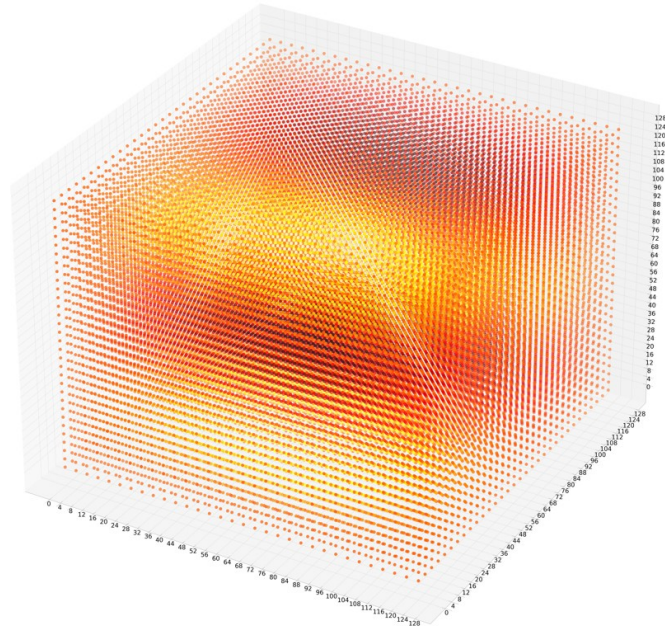
## График решения разностной схемы

При построении графиков обнаружено, что визуальной разницы между шагами  $[t=0, t=20]$  нет. Поэтому представлены графики для  $t=20$ . Для удобства отображаются не все точки а их часть (координаты которой кратны шагу), шаг указан в легенде графика ( $s$ ).

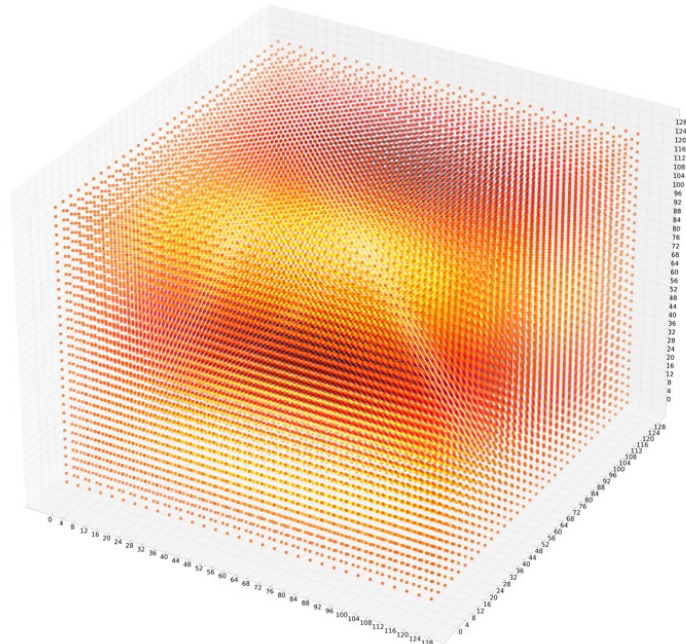
Диапазон:



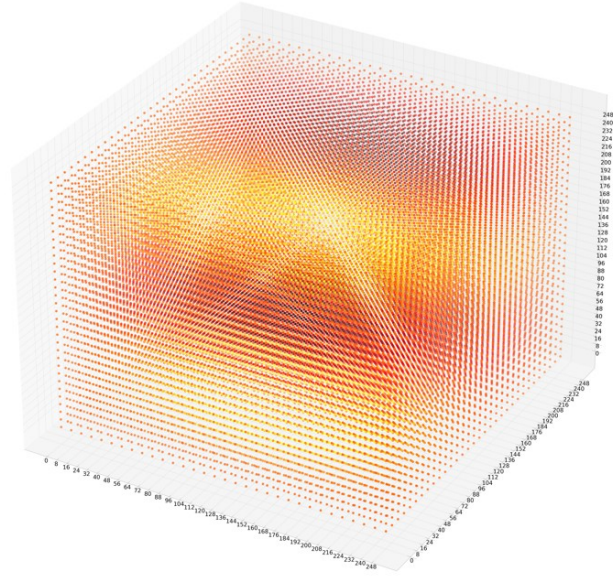
Решение разностной схемы принадлежит диапазону  $[-1,1]$  и отображаются в  $[0, 100]$



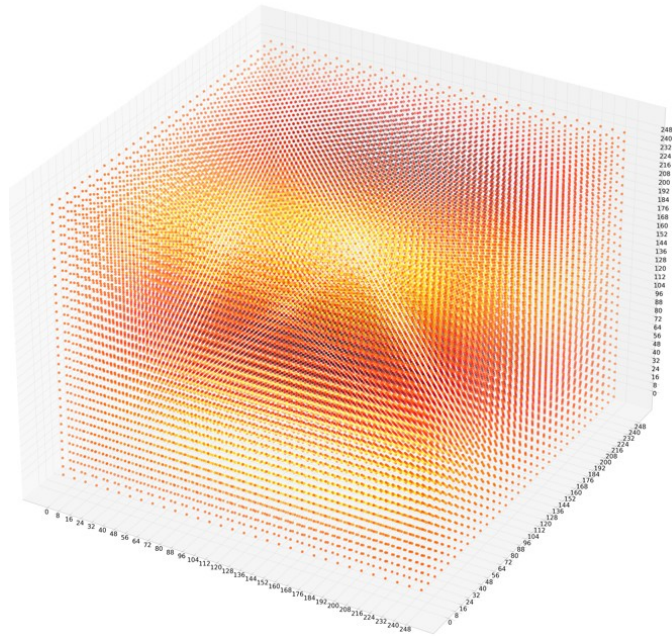
Фигура 7:  $L=1$ ,  $N=128$ ,  $K=220$ ,  $T=1$ ,  $t=20$ ,  $s=4$



Фигура 8:  $L=3.14$ ,  $N=128$ ,  $K=220$ ,  $T=1$ ,  $t=20$ ,  $s=4$

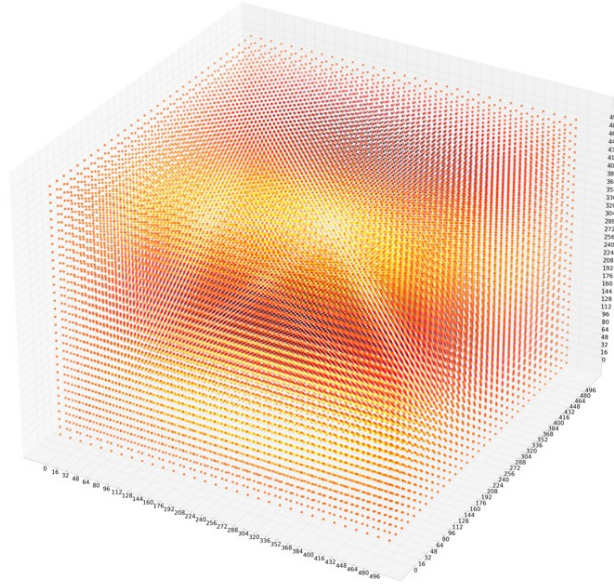


Фигура 9:  $L=1$ ,  $N=256$ ,  $K=442$ ,  $T=1$ ,  $t=20$ ,  $s=8$

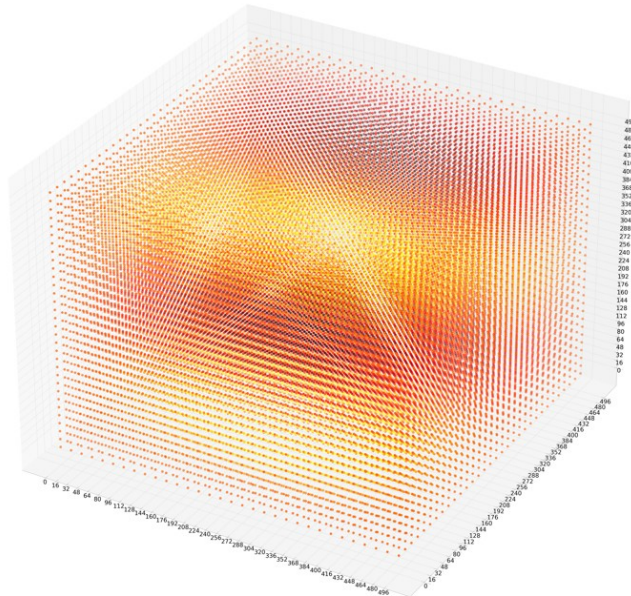


Фигура 10:  $L=3.14$ ,  $N=256$ ,  $K=442$ ,  $T=1$ ,  $t=20$ ,  $s=8$





*Фигура 11:*  $L=1$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  
 $s=16$

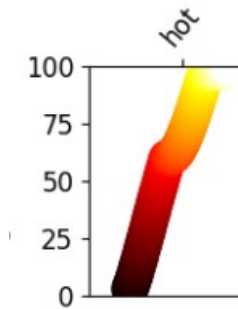


*Фигура 12:*  $L=3.14$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  
 $s=16$

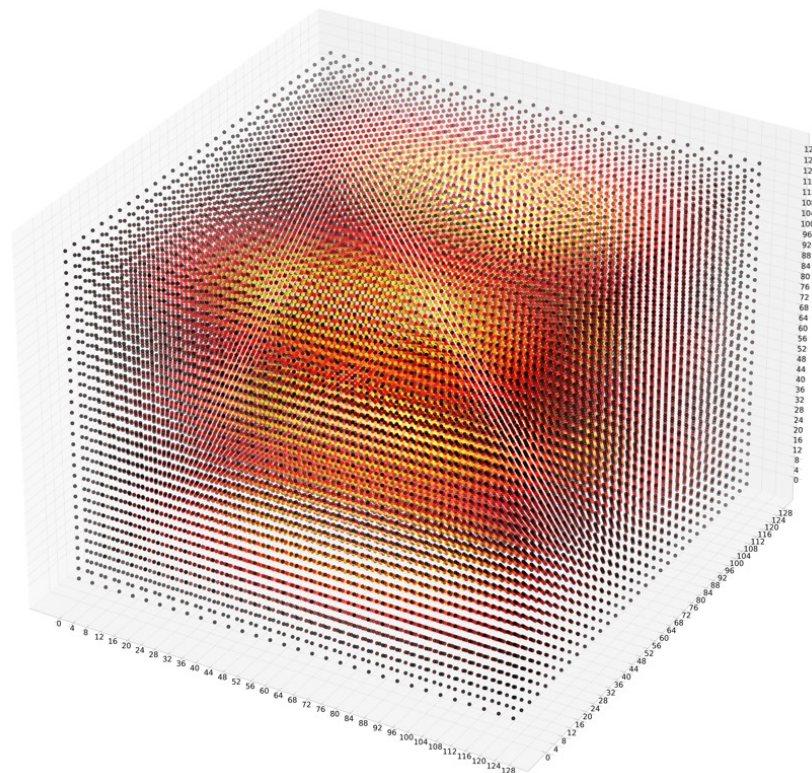
## График погрешности

Максимальная погрешность достигается на шаге  $t = 20$ . Поэтому представлены графики для  $t = 20$ . Для удобства отображаются не все точки а их часть (координаты которой кратны шагу), шаг указан в легенде графика (s).

Диапазон:

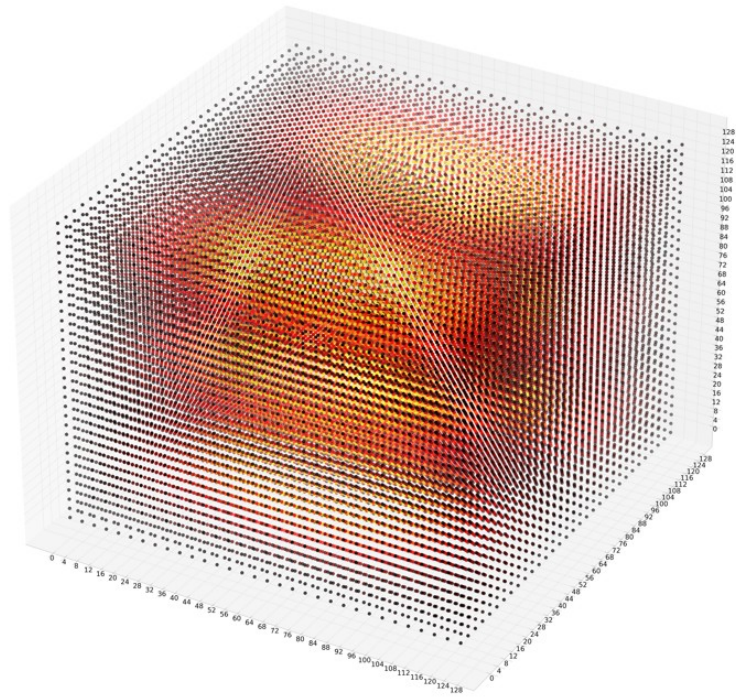


Погрешность схемы принадлежит диапазону  $[0, 0.000056]$  и отображаются в  $[0, 100]$

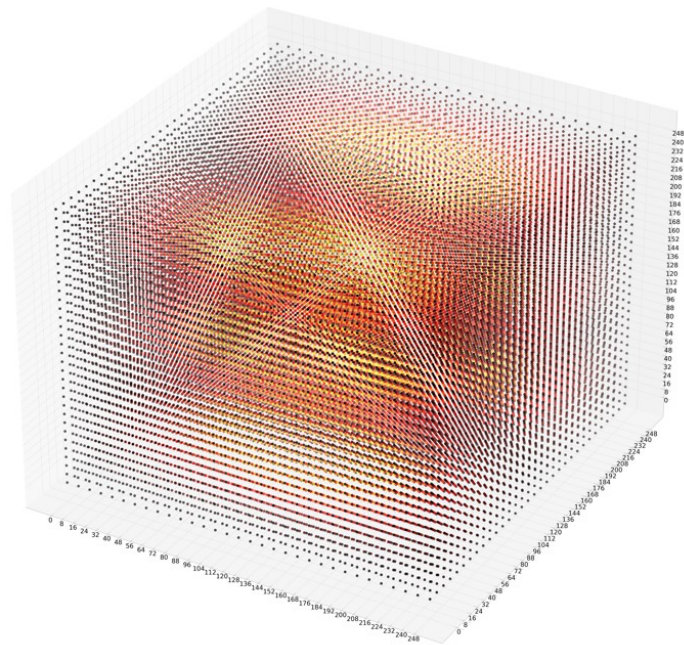


Фигура 13:  $L=1, N=128, K=220, T=1, t=20, s=4$

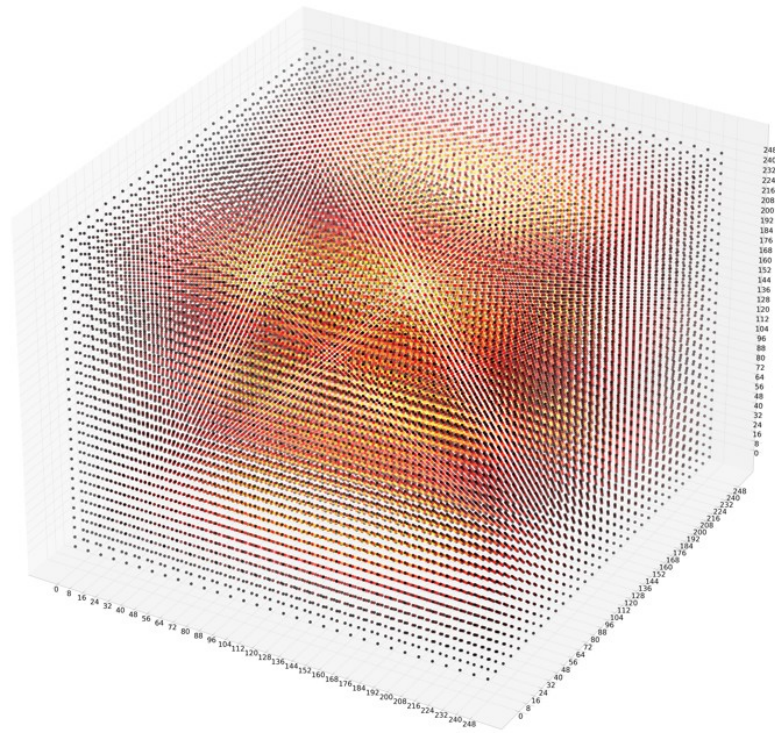




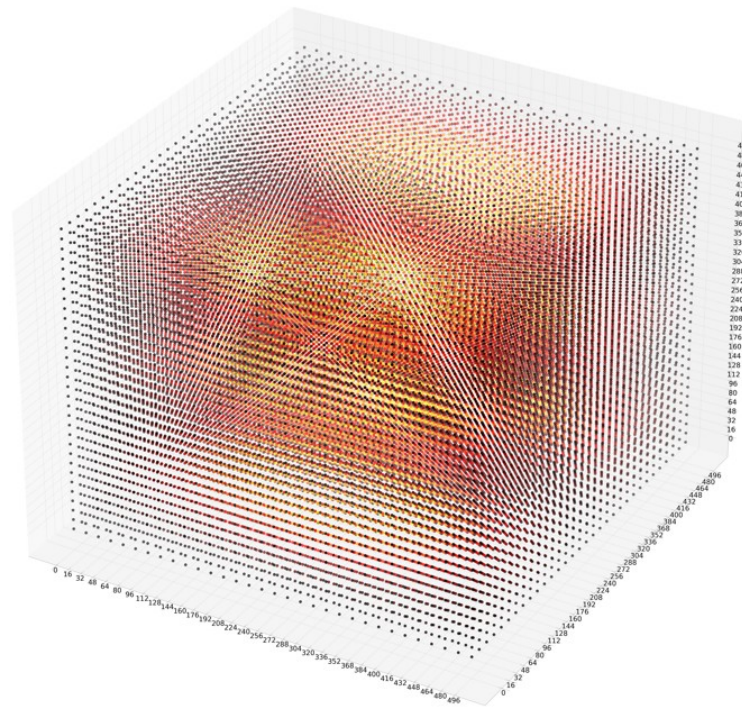
Фигура 14:  $L=3.14$ ,  $N=128$ ,  $K=220$ ,  $T=1$ ,  $t=20$ ,  $s=4$



Фигура 15:  $L=1$ ,  $N=256$ ,  $K=442$ ,  $T=1$ ,  $t=20$ ,  $s=8$

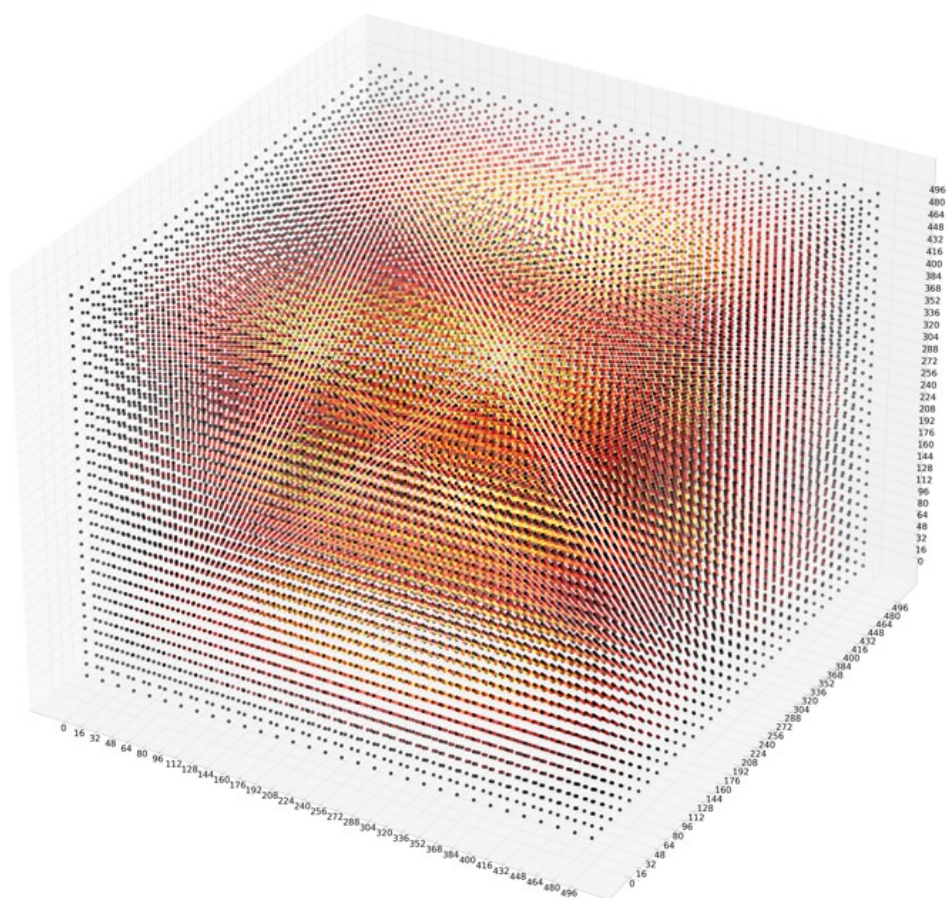


Фигура 16:  $L=3.14$ ,  $N=256$ ,  $K=442$ ,  $T=1$ ,  $t=20$ ,  $s=8$



Фигура 17:  $L=1$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  $s=16$





Фигура 18:  $L=3.14$ ,  $N=512$ ,  $K=886$ ,  $T=1$ ,  $t=20$ ,  $s=16$

# Результаты расчетов

## Последовательная программа

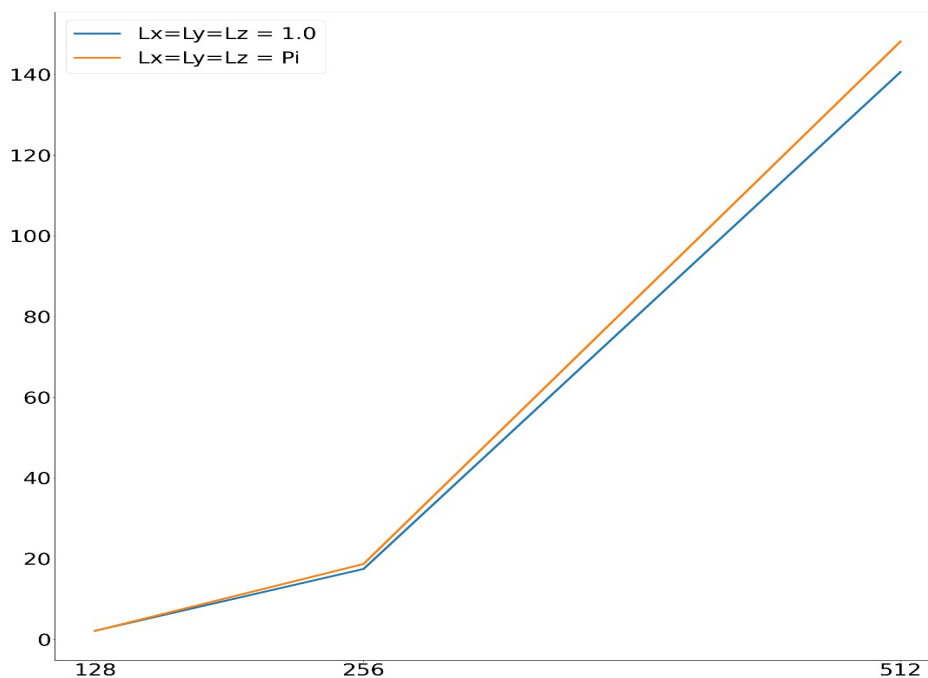
$L_x = L_y = L_z = 1.0$

Число точек сетки N	Число точек сетки по T	Максимальная погрешность	Время решения T
128	220	0.0000557521	2.13
256	442	0.0000039918	17.454
512	886	0.0000002568	140.62

$L_x = L_y = L_z = \pi$

Число точек сетки N	Число точек сетки по T	Максимальная погрешность	Время решения T
128	220	0.0000188797	2.098
256	442	0.0000011774	18.686
512	886	0.0000000733	148.195

## Графики



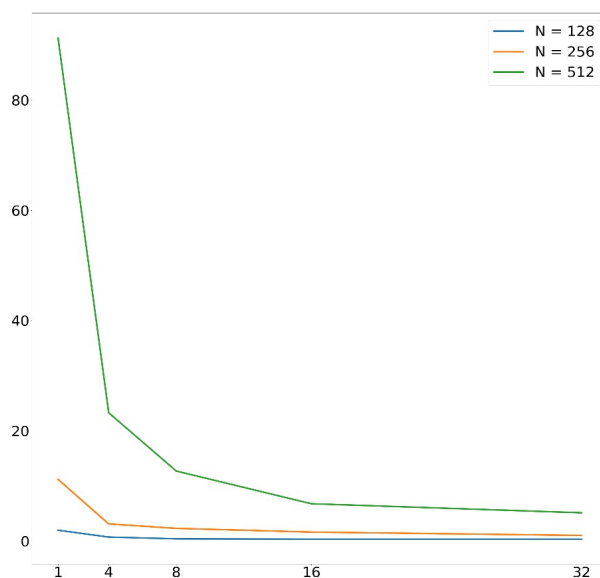
Фигура 19: График времени выполнения

## МРІ версія

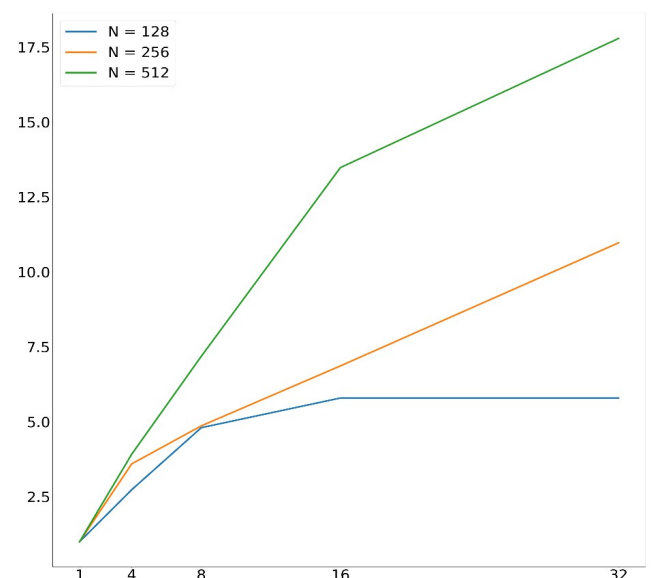
$L_x = L_y = L_z = 1.0$

Число МРІ процесов Nr	Число точек сетки N	Число точек сетки по T	Время Решения T	Ускорение S	Погрешность $\delta$
1	128	220	1.97	1	0.0000557521
4	128	220	0.72	2.736	0.0000557521
8	128	220	0.41	4.805	0.0000557521
16	128	220	0.34	5.794	0.0000557521
32	128	220	0.34	5.794	0.0000557521
1	256	442	11.19	1	0.0000039918
4	256	442	3.11	3.598	0.0000039918
8	256	442	2.30	4.865	0.0000039918
16	256	442	1.63	6.865	0.0000039918
32	256	442	1.02	10.971	0.0000039918
1	512	886	91.26	1	0.0000002568
4	512	886	23.27	3.921	0.0000002568
8	512	886	12.7	7.186	0.0000002568
16	512	886	6.77	13.48	0.0000002568
32	512	886	5.13	17.789	0.0000002568

Графики для  $L_x = L_y = L_z = 1.0$



Фигура 20: Время выполнения



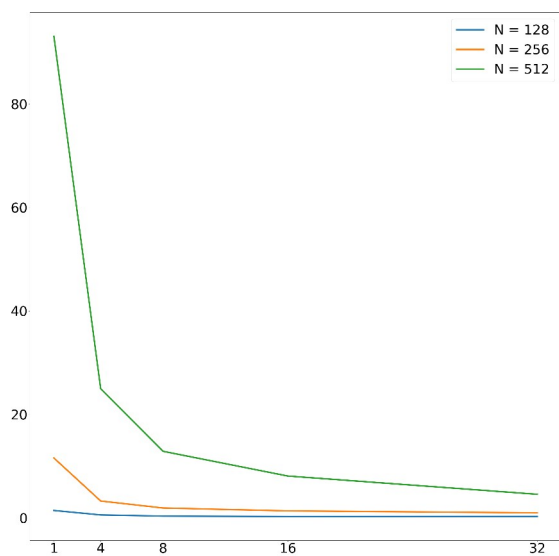
Фигура 21: Ускорение



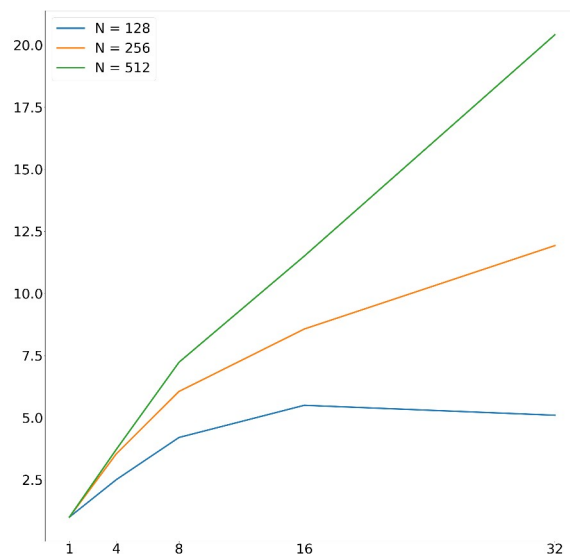
$$Lx = Ly = Lz = Pi$$

Число MPI процессов Nr	Число точек сетки N	Число точек сетки по T	Время Решения T	Ускорение S	Погрешность $\delta$
1	128	220	1.43	1	0.0000188797
4	128	220	0.57	2.509	0.0000188797
8	128	220	0.34	4.205	0.0000188797
16	128	220	0.26	5.5	0.0000188797
32	128	220	0.28	5.10	0.0000188797
1	256	442	11.57	1	0.0000011774
4	256	442	3.26	3.549	0.0000011774
8	256	442	1.91	6.057	0.0000011774
16	256	442	1.35	8.570	0.0000011774
32	256	442	0.97	11.927	0.0000011774
1	512	886	93.08	1	0.0000000733
4	512	886	24.96	3.729	0.0000000733
8	512	886	12.87	7.232	0.0000000733
16	512	886	8.09	11.505	0.0000000733
32	512	886	4.56	20.412	0.0000000733

Графики для  $Lx = Ly = Lz = Pi$



Фигура 22: Время выполнения



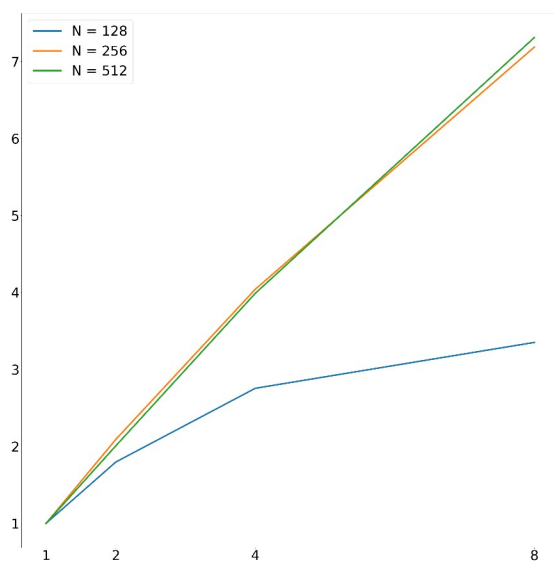
Фигура 23: Ускорение

## MPI + OpenMP

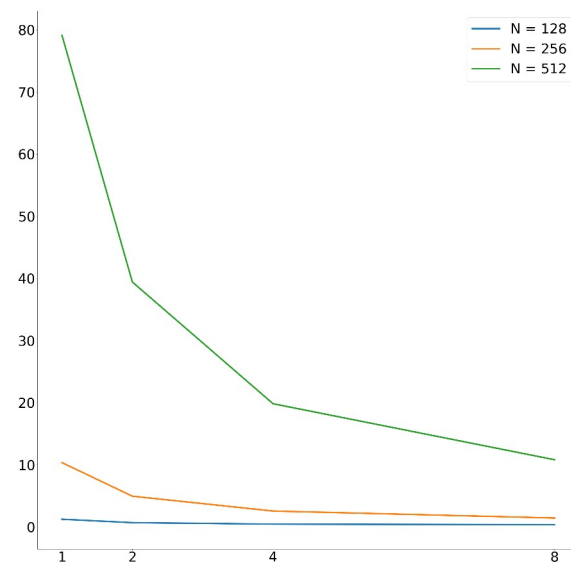
$L_x = L_y = L_z = 1.0$

Число MPI процессов в $N_p$	Число OpenMP нитей в процессе	Число точек сетки $N$	Число точек сетки по $T$	Время Решения $T$	Ускорение $S$	Погрешность $\delta$
1	4	128	220	1.24	1	0.0000557521
2	4	128	220	0.69	1.797	0.0000557521
4	4	128	220	0.45	2.755	0.0000557521
8	4	128	220	0.37	3.351	0.0000557521
1	4	256	442	10.35	1	0.0000039918
2	4	256	442	4.95	2.090	0.0000039918
4	4	256	442	2.56	4.042	0.0000039918
8	4	256	442	1.44	7.187	0.0000039918
1	4	512	886	79.12	1	0.0000002568
2	4	512	886	39.42	2.007	0.0000002568
4	4	512	886	19.83	3.989	0.0000002568
8	4	512	886	10.82	7.312	0.0000002568

Графики для  $L_x = L_y = L_z = 1.0$



Фигура 24: Ускорение

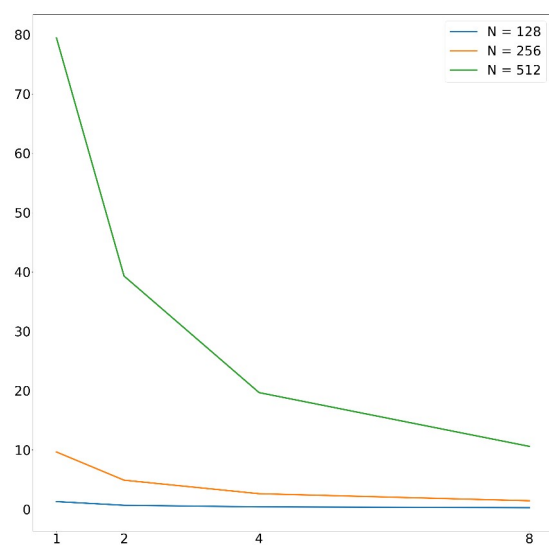


Фигура 25: Время выполнения

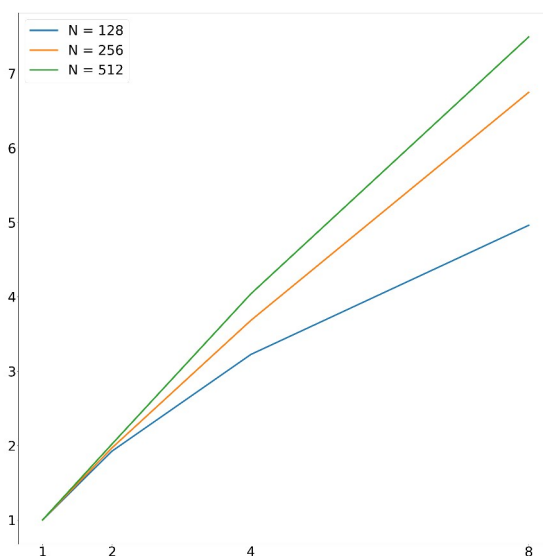
$$Lx = Ly = Lz = Pi$$

Число MPI процессов в Nr	Число OpenMP нитей в процессе	Число точек сетки N	Число точек сетки по T	Время Решения T	Ускорение S	Погрешность $\delta$
1	4	128	220	1.29	1	0.0000188797
2	4	128	220	0.67	1.925	0.0000188797
4	4	128	220	0.40	3.225	0.0000188797
8	4	128	220	0.26	4.961	0.0000188797
1	4	256	442	9.65	1	0.0000011774
2	4	256	442	4.90	1.969	0.0000011774
4	4	256	442	2.62	3.683	0.0000011774
8	4	256	442	1.43	6.748	0.0000011774
1	4	512	886	79.43	1	0.0000000733
2	4	512	886	39.31	2.020	0.0000000733
4	4	512	886	19.66	4.040	0.0000000733
8	4	512	886	10.60	7.493	0.0000000733

Графики для  $Lx = Ly = Lz = Pi$



Фигура 26: Время выполнения



Фигура 27: Ускорение