Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Отчет

По курсу: «Суперкомпьютерное моделирование и технологии»

«MPI + OpenMP»

Цирунов Леонид Александрович

группа 628

30 ноября 2024 г.

Математическая постановка задачи

В трехмерной замкнутой области

$$\Omega = \left[0 \le x \le L_x\right] \times \left[0 \le y \le L_y\right] \times \left[0 \le z \le L_z\right]$$

Для $0 < t \le T$ требуется найти решение $u(x,\ y,\ z,\ t)$ уравнения в частных производных

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

С начальными условиями

$$u\,|_{t=0} = \varphi(x,\ y,\ z)$$

$$\frac{\partial u}{\partial t}\,|_{t=0} = 0$$

При условии, что на границах области заданы однородные граничные условия первого рода

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

Либо периодические граничные условия

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

Численный метод решения задачи

Введем на Ω сетку: $\omega_{h\tau} = \bar{\omega}_h \times \omega_\tau$

$T = T_0$

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

$\bar{\omega}_h = \{\left(x_i = ih_x,\ y_j = jh_y,\ z_k = kh_z\right),\ i, j, k = 0, 1, \ldots, N,\ h_x N = L_x, h_y N = L_y, h_z N = L_z\}$

$\omega_\tau = \left\{t_n = n\tau,\ n = 0, 1, \ldots, K,\ \tau K = T\right\}$

Через ω_h обозначим множество внутренних, а через γ_h - множество гранитных узлов сетки $\bar{\omega}_h$.

Для аппроксимации исходного уравнения с начальными условиями воспользуемся следующей системой уравнений:

$$\frac{u_{ijk}^{n+1} - 2u_{ijk}^{n} + u_{ijk}^{n-1}}{\tau^2} = \triangle_h u^n,\ \left(x_i, y_j,\ z_k\right) \in \omega_h,\ n = 1, 2, \ldots,\ K - 1$$

Где $\triangle_h$ - семиточечный разностный аналог оператора Лапласа:

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

Для начала счета должны быть заданы значения $u_{i,j,k}^{0}$ и $u_{i,j,k}^{1}$, $(x_i, y_j, z_k) \in \omega_h$.

$$u_{i,j,k}^{0} = \varphi(x_i, y_j,\ z_k), \qquad (x_i, y_j, z_k) \in \omega_h.$$

$$u_{i,j,k}^{1} = u_{i,j,k}^{0} + \frac{\tau^2}{2} \triangle_h \varphi(x_i, y_j,\ z_k)$$

Для вычисления значений u^0 , $u^1 \in \gamma_h$ допускается использование аналитического значения , которое задается в программе еще для вычисления погрешности решения задачи.

Из варианта №8 следуют следующие формулы:

$$u_{analytical} = \sin(\frac{2\pi}{L_x}x)*\sin\left(\frac{4\pi}{L_y}y\right)*\sin\left(\frac{6\pi}{L_z}z\right)*\cos(a_t*t),$$

$$a_t = \pi\sqrt{\left(\frac{4}{L_x{}^2} + \frac{16}{L_y{}^2} + \frac{36}{L_z{}^2}\right)}$$

Граничные условия:

$$\text{П:}\quad u\left(0, y, z, t\right) = u\left(L_x, y, z, t\right), \quad u_x\left(0, y, z, t\right) = u_x\left(L_x, y, z, t\right),$$
$$\text{П:}\quad u(x, 0, z, t) = u\left(x, L_y, z, t\right), \quad u_y(x, 0, z, t) = u_y\left(x, L_y, z, t\right),$$
$$\text{П:}\quad u\left(x, y, 0, t\right) = u\left(x, y, L_z, t\right), \quad u_z\left(x, y, 0, t\right) = u_z\left(x, y, L_z, t\right)$$

Алгоритм численного решения:

1. Вычисление граничных значений u^0 и u^1.
2. Вычисление u^0 внутри области: $u^0_{i,j,k} = \varphi\left(x_i, y_j, z_k\right)$
3. Вычисление u^1 внутри области: $u^1_{i,j,k} = u^0_{i,j,k} + \frac{\tau^2}{2} \triangle_h \varphi(x_i, y_j, z_k)$
4. Вычисление K – 1 раз u^{n+1}:
   $u^{n+1}_{ijk} = 2u^n_{ijk} - u^{n-1}_{ijk} + \tau^2 \triangle_h u^n$

Программный метод решения задачи

Программная реализация состоит из 4 файлов: ***main.cpp***, ***equation_solution.cpp***, ***equation.h*** и ***Makefile***.

Файл ***main.cpp*** содержит основную функцию, получающую входные значения, задающую количество потоков, отключающую динамическое управление количеством потоков, производящую инициализацию MPI с созданием декартовой топологии процессов, и запускающую решение задачи, а также две функции для сохранения результатов:

- ***dump_block_to_CSV*** - сохранение погрешности или сетки (для каждого блока), полученной численным или аналитическим способом, в файл в формате CSV (для дальнейшей визуализации);
- ***save_statistics*** - вывод в файл информации о времени работы решения и максимальной погрешности вычисления.

На вход программа получает 3 или 6 значений в зависимости от переданного ***L_type***. Описание параметров в порядке передачи:

- ***N*** - размер сетки по одной координате (конечный размер N^3);
- ***L_type*** - тип значений L по разным координатам, принимает значения: *1*, *pi*, *custom.*

В случае *1* и *pi*, значения L по всем координатам равны числу соответственно. Если ***L_type*** передано значение *custom* , в этом случае необходимо передать значения L по каждой координате: L_x, L_y, L_z.

Файл ***equation.h*** содержит объявления типов: класса сетки с функцией получения индекса элемента в линейном массиве описывающем сетку, класса блока с функцией получения индекса элемента в линейном массиве описывающем блок, а также функцию $u_{analytical}$ для вычисления аналитических значений точек сетки.

Файл ***equation_solution.cpp*** содержит основной алгоритм решения задачи. Содержит функции:

- ***laplace_operator*** - вычисление значения разностного аналога оператора Лапласа;
- ***fill_send_buffers*** - функция, используемая для заполнения буферов для отправки данных соседним процессам.
- ***exchange_ghost_layers*** - функция реализующую пересылку данных между процессами, а также последующее заполнение дополнительного слоя, содержащего данные других процессов(гало).
- ***init*** - вычисление внутренних u^0 , u^1 и граничных начальных точек, необходимых для запуска алгоритма;
- ***run_algo*** - итерация по K – 1 оставшимся временным шагам алгоритма с вычислением значений граничных и внутренних точек на новом шаге алгоритма. На каждом шаге производится подсчет максимальной погрешности численного решения от аналитического с выводом информации в консоль. Также в этом цикле производится заполнение буферов для отправки данных соседним процессам.
- ***solve_equation*** - инициализируются переменные, выполняется запуск алгоритма и ведется подсчет времени алгоритма.

Распараллеливание производится с использованием технологии MPI и OpenMP.

Файл ***Makefile*** содержит цели для компиляции и запуска задач. Для запуска на Polus используются команды «***make compile_polus***» и «***make run_all_polus***».

Результаты расчетов

1. Количество OpenMP нитей на процесс = 1:

Таблица 1.1: Результаты при L = 1

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,77943	0,264147	1,000	1,000	1.4004e-06
1	128	8,78846	0,694904	0,9990	0,3801	1.4004e-06
2	128	5,84676	0,366543	1,5016	0,721	1.4004e-06
4	128	2,49951	0,210749	3,5125	1,253	1.4004e-06
8	128	1,74339	0,145915	5,0358	1,810	1.4004e-06
16	128	1,16079	0,145204	7,5633	1,819	1.4004e-06
32	128	0,590567	0,082852	14,8661	3,188	1.4004e-06
0-Sequential	256	62,994	2,175710	1,000	1,000	3.49927e-07
1	256	64,9288	5,531750	0,9702	0,3933	3.49927e-07
2	256	34,2605	2,854500	1,8387	0,7622	3.49927e-07
4	256	21,9	1,489550	2,8764	1,4606	3.49927e-07
8	256	8,97345	0,777758	7,0200	2,7974	3.49927e-07
16	256	6,44694	0,750220	9,7711	2,9001	3.49927e-07
32	256	3,13201	0,625402	20,1130	3,4789	3.49927e-07
0-Sequential	512	484,384	16,5496	1,000	1,000	8.71479e-08
1	512	499,143	43,5845	0,9704	0,3797	8.71479e-08
2	512	252,266	22,7211	1,9201	0,7284	8.71479e-08
4	512	134,291	11,6664	3,6070	1,4186	8.71479e-08
8	512	79,4925	6,1500	6,0935	2,6910	8.71479e-08
16	512	54,0828	4,4813	8,9563	3,6930	8.71479e-08
32	512	22,3816	4,0910	21,6421	4,0454	8.71479e-08

Таблица 1.2: Результаты при L = π

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,47777	0,264720	1,000	1,000	1.41974e-07
1	128	8,73889	0,715269	0,9701	0,3701	1.41974e-07
2	128	4,72145	0,372932	1,7956	0,7098	1.41974e-07
4	128	2,4073	0,210389	3,5217	1,2582	1.41974e-07
8	128	1,51146	0,115169	5,6090	2,2985	1.41974e-07
16	128	0,817303	0,145567	10,3729	1,8185	1.41974e-07
32	128	0,55731	0,080940	15,2119	3,2706	1.41974e-07
0-Sequential	256	62,5438	2,1346	1,000	1,000	3.55074e-08
1	256	64,8378	5,5436	0,9646	0,3851	3.55074e-08
2	256	33,4614	2,8521	1,8691	0,7484	3.55074e-08
4	256	18,3309	1,4855	3,4119	1,4369	3.55074e-08
8	256	9,3261	0,7838	6,7063	2,7235	3.55074e-08
16	256	6,05766	0,7432	10,3247	2,8720	3.55074e-08
32	256	3,8675	0,6035	16,1716	3,5368	3.55074e-08
0-Sequential	512	480,591	16,8837	1,000	1,000	8.8744e-09
1	512	501,871	43,7070	0,9576	0,3863	8.8744e-09
2	512	264,851	22,8260	1,8146	0,7397	8.8744e-09
4	512	143,562	11,6729	3,3476	1,4464	8.8744e-09
8	512	87,9549	6,0956	5,4641	2,7698	8.8744e-09
16	512	39,0933	4,2927	12,2934	3,9331	8.8744e-09
32	512	20,6163	4,0634	23,3112	4,1551	8.8744e-09

2) Количество OpenMP нитей на процесс = 2:

Таблица 2.1: Результаты при L = 1

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,77943	0,264147	1,000	1,000	1.4004e-06
1	128	4,75797	0,358247	1,8452	0,7373	1.4004e-06
2	128	3,26669	0,198587	2,6876	1,330	1.4004e-06
4	128	1,76244	0,122219	4,9814	2,161	1.4004e-06
8	128	1,16557	0,094433	7,5323	2,797	1.4004e-06
16	128	0,820489	0,158068	10,7002	1,671	1.4004e-06
32	128	0,513303	0,074237	17,1038	3,558	1.4004e-06
0-Sequential	256	62,994	2,175710	1,000	1,000	3.49927e-07
1	256	33,9802	2,837690	1,8538	0,7667	3.49927e-07
2	256	21,4607	1,475250	2,9353	1,4748	3.49927e-07
4	256	11,073	0,767067	5,6890	2,8364	3.49927e-07
8	256	5,19309	0,596515	12,1304	3,6474	3.49927e-07
16	256	4,33959	0,920512	14,5161	2,3636	3.49927e-07
32	256	2,78488	0,593498	22,6200	3,6659	3.49927e-07
0-Sequential	512	484,384	16,5496	1,000	1,000	8.71479e-08
1	512	264,1	22,3383	1,8341	0,7409	8.71479e-08
2	512	142,708	11,7204	3,3942	1,4120	8.71479e-08
4	512	68,9858	5,9162	7,0215	2,7973	8.71479e-08
8	512	45,2255	4,2471	10,7104	3,8967	8.71479e-08
16	512	26,8418	4,4700	18,0459	3,7024	8.71479e-08
32	512	17,6212	4,0284	27,4887	4,1083	8.71479e-08

Таблица 2.2: Результаты при L = π

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность

0-Sequential	128	8,47777	0,264720	1,000	1,000	1.41974e-07
1	128	4,56087	0,362507	1,8588	0,7302	1.41974e-07
2	128	3,26108	0,196841	2,5997	1,3448	1.41974e-07
4	128	1,30556	0,123675	6,4936	2,1404	1.41974e-07
8	128	0,999156	0,101602	8,4849	2,6055	1.41974e-07
16	128	0,663293	0,120158	12,7813	2,2031	1.41974e-07
32	128	0,466975	0,074413	18,1547	3,5574	1.41974e-07
0-Sequential	256	62,5438	2,1346	1,000	1,000	3.55074e-08
1	256	33,6383	2,8476	1,8593	0,7496	3.55074e-08
2	256	17,9906	1,4698	3,4765	1,4523	3.55074e-08
4	256	9,85455	0,7762	6,3467	2,7502	3.55074e-08
8	256	7,98199	0,5990	7,8356	3,5636	3.55074e-08
16	256	3,15786	0,9011	19,8058	2,3688	3.55074e-08
32	256	2,73497	0,6152	22,8682	3,4695	3.55074e-08
0-Sequential	512	480,591	16,8837	1,000	1,000	8.8744e-09
1	512	254,271	22,4315	1,8901	0,7527	8.8744e-09
2	512	139,785	11,7556	3,4381	1,4362	8.8744e-09
4	512	85,3684	5,9184	5,6296	2,8528	8.8744e-09
8	512	48,5356	4,1949	9,9018	4,0249	8.8744e-09
16	512	32,6227	4,4031	14,7318	3,8345	8.8744e-09
32	512	14,7935	3,9967	32,4866	4,2245	8.8744e-09

3) Количество OpenMP нитей на процесс = 4 :

Таблица 3.1: Результаты при L = 1

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,77943	0,264147	1,000	1,000	1.4004e-06
1	128	2,89351	0,190026	3,0342	1,3901	1.4004e-06
2	128	1,96257	0,106635	4,4734	2,477	1.4004e-06
4	128	1,09617	0,091188	8,0092	2,897	1.4004e-06
8	128	0,945275	0,095627	9,2877	2,762	1.4004e-06

16	128	0,732593	0,125249	11,9840	2,109	1.4004e-06
32	128	0,325534	0,092155	26,9693	2,866	1.4004e-06
0-Sequential	256	62,994	2,175710	1,000	1,000	3.49927e-07
1	256	20,3292	1,455320	3,0987	1,4950	3.49927e-07
2	256	13,4621	0,753066	4,6794	2,8891	3.49927e-07
4	256	7,30345	0,600027	8,6252	3,6260	3.49927e-07
8	256	3,43487	0,590471	18,3396	3,6847	3.49927e-07
16	256	2,66733	0,898799	23,6169	2,4207	3.49927e-07
32	256	1,81473	0,608717	34,7126	3,5743	3.49927e-07
0-Sequential	512	484,384	16,5496	1,000	1,000	8.71479e-08
1	512	153,466	11,4420	3,1563	1,4464	8.71479e-08
2	512	82,864	5,9579	5,8455	2,7778	8.71479e-08
4	512	48,7568	4,0019	9,9347	4,1354	8.71479e-08
8	512	29,0539	4,3950	16,6719	3,7655	8.71479e-08
16	512	21,6885	4,5113	22,3337	3,6685	8.71479e-08
32	512	11,6228	4,0345	41,6753	4,1021	8.71479e-08

Таблица 3.2: Результаты при L = π

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,47777	0,264720	1,000	1,000	1.41974e-07
1	128	2,3831	0,190110	3,5575	1,3925	1.41974e-07
2	128	2,02599	0,106365	4,1845	2,4888	1.41974e-07
4	128	1,13927	0,100148	7,4414	2,6433	1.41974e-07
8	128	0,860866	0,095837	9,8480	2,7622	1.41974e-07
16	128	0,38323	0,126452	22,1219	2,0934	1.41974e-07
32	128	0,337238	0,089952	25,1388	2,9429	1.41974e-07
0-Sequential	256	62,5438	2,1346	1,000	1,000	3.55074e-08
1	256	17,0477	1,4550	3,6688	1,4671	3.55074e-08
2	256	12,0244	0,7449	5,2014	2,8657	3.55074e-08
4	256	6,94968	0,5805	8,9995	3,6774	3.55074e-08

8	256	5,36409	0,5862	11,6597	3,6417	3.55074e-08
16	256	2,68311	0,9556	23,3102	2,2337	3.55074e-08
32	256	1,93354	0,6273	32,3468	3,4027	3.55074e-08
0-Sequential	512	480,591	16,8837	1,000	1,000	8.8744e-09
1	512	128,187	11,5147	3,7491	1,4663	8.8744e-09
2	512	91,6757	5,9252	5,2423	2,8495	8.8744e-09
4	512	54,3328	3,9632	8,8453	4,2602	8.8744e-09
8	512	31,1446	4,1771	15,4310	4,0420	8.8744e-09
16	512	18,3425	4,4811	26,2010	3,7678	8.8744e-09
32	512	11,8672	4,0070	40,4974	4,2135	8.8744e-09

4) Количество OpenMP нитей на процесс = 8:

Таблица 4.1: Результаты при L = 1

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,77943	0,264147	1,000	1,000	1.4004e-06
1	128	1,88086	0,098315	4,6678	2,6867	1.4004e-06
2	128	1,26382	0,088709	6,9467	2,978	1.4004e-06
4	128	0,807667	0,097474	10,8701	2,710	1.4004e-06
8	128	0,558438	0,106883	15,7214	2,471	1.4004e-06
16	128	0,326319	0,143369	26,9044	1,842	1.4004e-06
32	128	0,239759	0,160234	36,6177	1,649	1.4004e-06
0-Sequential	256	62,994	2,175710	1,000	1,000	3.49927e-07
1	256	12,8527	0,735769	4,9012	2,9571	3.49927e-07
2	256	8,85137	0,579825	7,1169	3,7524	3.49927e-07
4	256	4,19413	0,521574	15,0196	4,1714	3.49927e-07
8	256	3,44527	0,566811	18,2842	3,8385	3.49927e-07
16	256	1,9664	0,991064	32,0352	2,1953	3.49927e-07
32	256	1,3886	0,663963	45,3651	3,2769	3.49927e-07
0-Sequential	512	484,384	16,5496	1,000	1,000	8.71479e-08

1	512	93,8672	5,8083	5,1603	2,8493	8.71479e-08
2	512	46,0996	3,8064	10,5073	4,3478	8.71479e-08
4	512	32,7387	4,0546	14,7955	4,0817	8.71479e-08
8	512	24,4265	4,1814	19,8303	3,9579	8.71479e-08
16	512	16,0972	4,4803	30,0912	3,6939	8.71479e-08
32	512	9,77882	4,0429	49,5340	4,0935	8.71479e-08

Таблица 4.2: Результаты при L = π

Число MPI процессов	Число точек сетки N по одной оси	Время решения Polus	Время решения локально M3 arm	Ускорение Polus	Ускорение локально	Погреш-ность
0-Sequential	128	8,47777	0,264720	1,000	1,000	1.41974e-07
1	128	1,3541	0,097535	6,2608	2,7141	1.41974e-07
2	128	1,2967	0,087771	6,5380	3,0160	1.41974e-07
4	128	0,734378	0,101998	11,5442	2,5953	1.41974e-07
8	128	0,641668	0,100482	13,2121	2,6345	1.41974e-07
16	128	0,295744	0,144042	28,6659	1,8378	1.41974e-07
32	128	0,23282	0,166175	36,4134	1,5930	1.41974e-07
0-Sequential	256	62,5438	2,1346	1,000	1,000	3.55074e-08
1	256	10,6586	0,7449	5,8679	2,8655	3.55074e-08
2	256	6,59082	0,5820	9,4895	3,6680	3.55074e-08
4	256	4,87802	0,5153	12,8216	4,1428	3.55074e-08
8	256	3,38477	0,5672	18,4780	3,7631	3.55074e-08
16	256	2,46825	0,9196	25,3393	2,3213	3.55074e-08
32	256	1,49754	0,6655	41,7644	3,2074	3.55074e-08
0-Sequential	512	480,591	16,8837	1,000	1,000	8.8744e-09
1	512	79,9815	5,7897	6,0088	2,9161	8.8744e-09
2	512	63,1985	3,9530	7,6045	4,2711	8.8744e-09
4	512	25,1734	4,0040	19,0912	4,2168	8.8744e-09
8	512	22,1404	4,1155	21,7065	4,1025	8.8744e-09
16	512	13,8749	4,5040	34,6374	3,7486	8.8744e-09
32	512	9,62267	4,0433	49,9436	4,1757	8.8744e-09

График 1: зависимость ускорения на Polus от числа процессов для сетки 128 при L = 1:

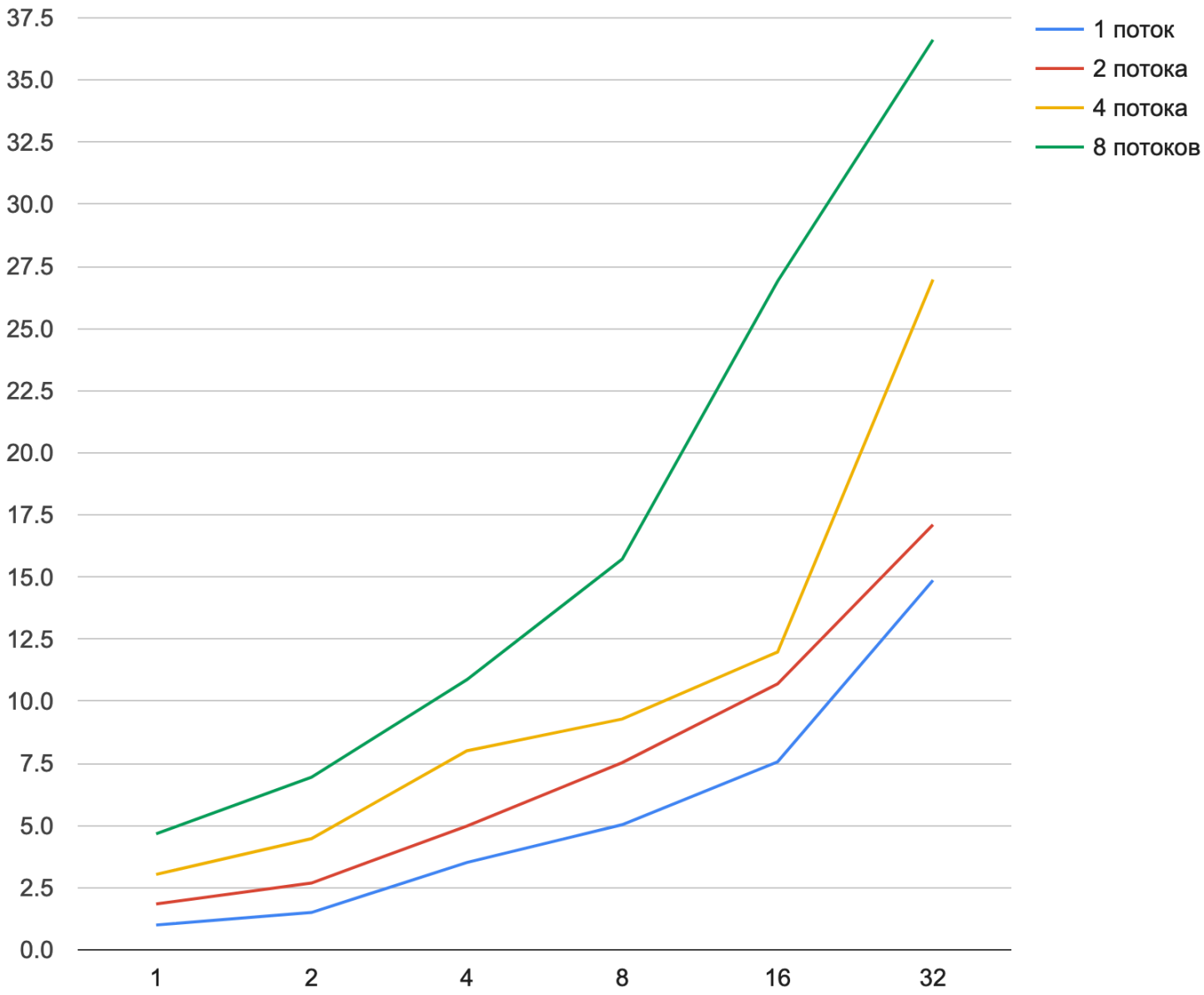


График 2: зависимость ускорения на Polus от числа процессов для сетки 128 при L = π:

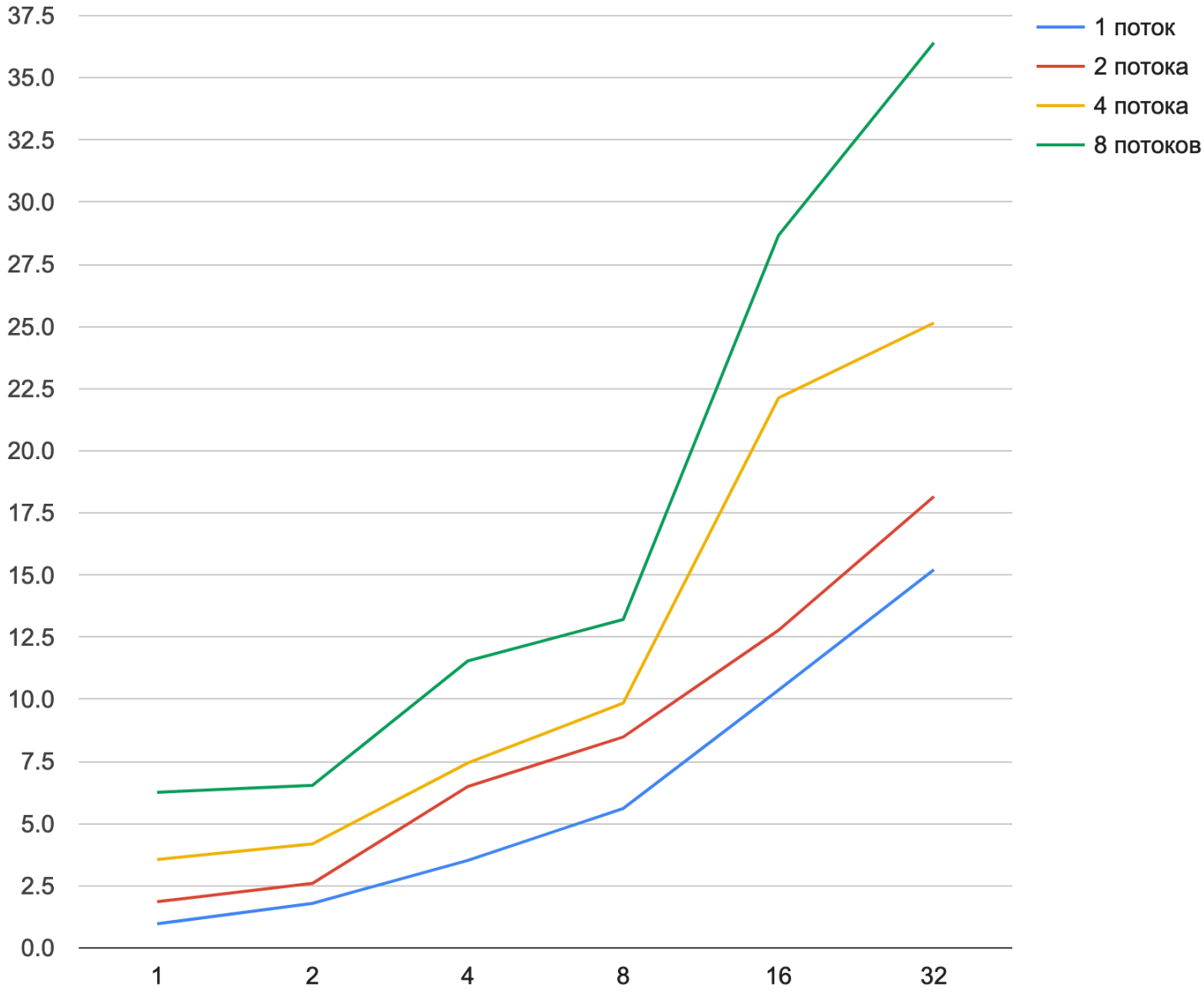


График 3: зависимость ускорения на Polus от числа процессов для сетки 256 при L = 1:

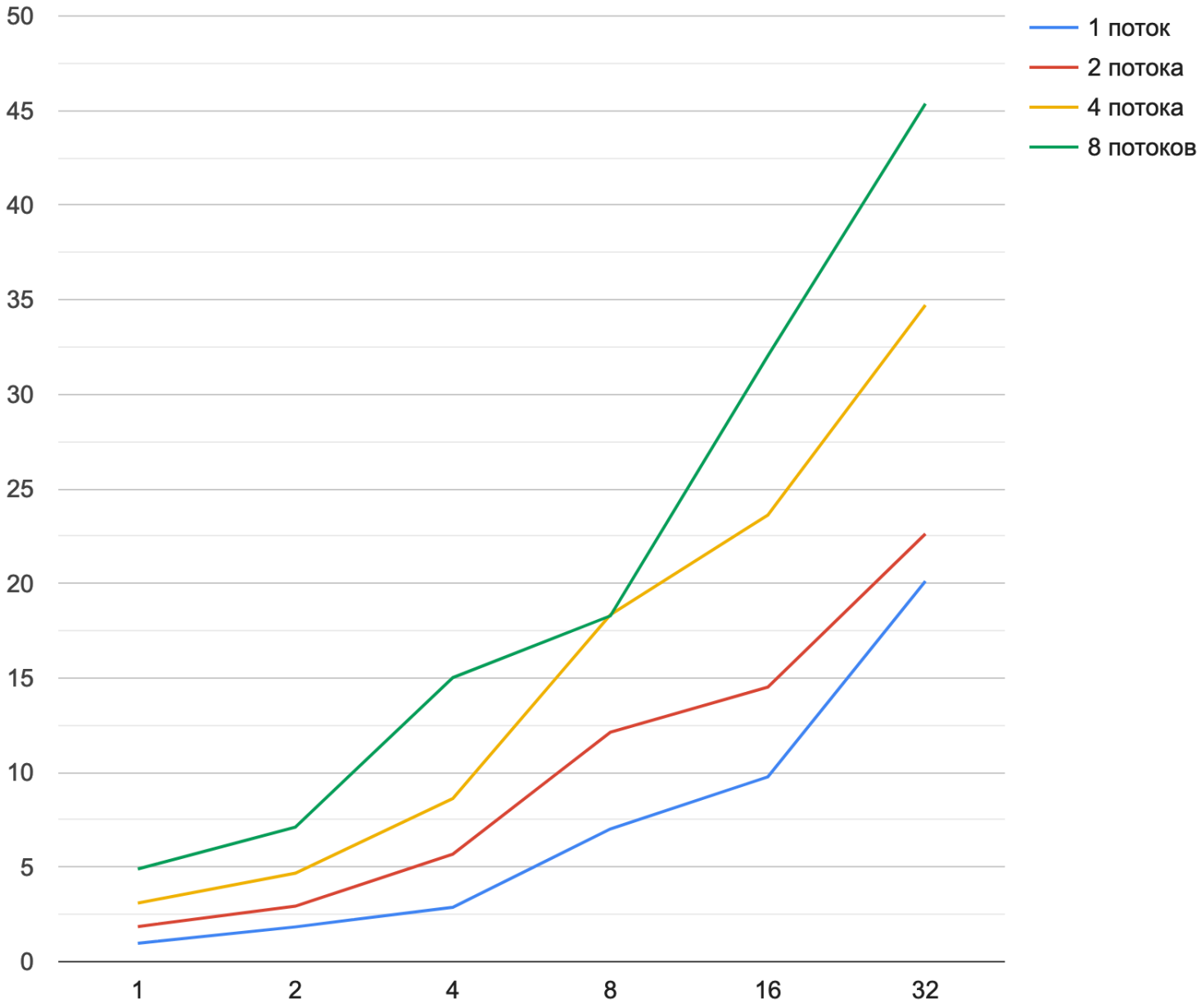


График 4: зависимость ускорения на Polus от числа процессов для сетки 256 при L = π:

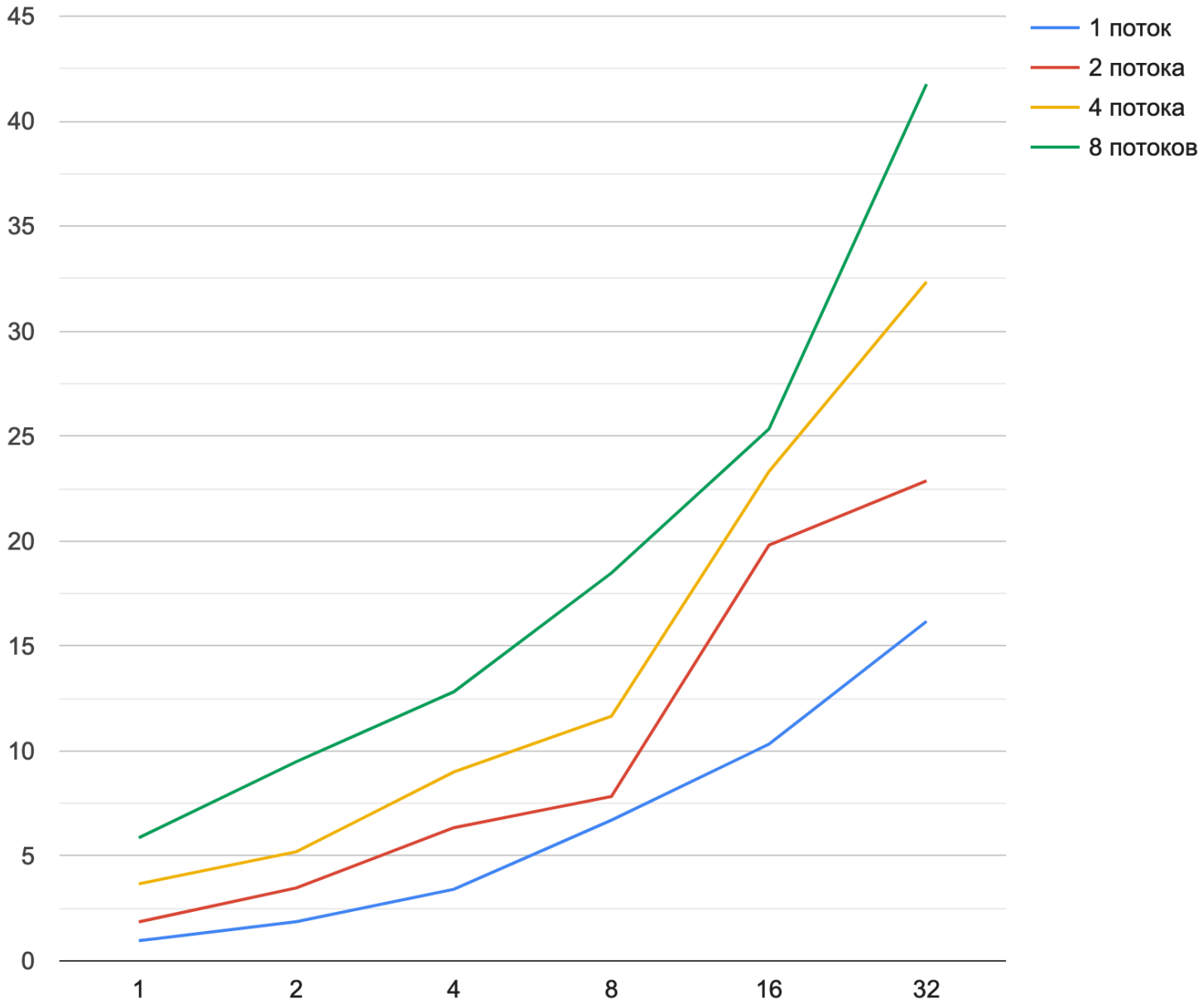


График 5: зависимость ускорения на Polus от числа процессов для сетки 512 при L = 1:

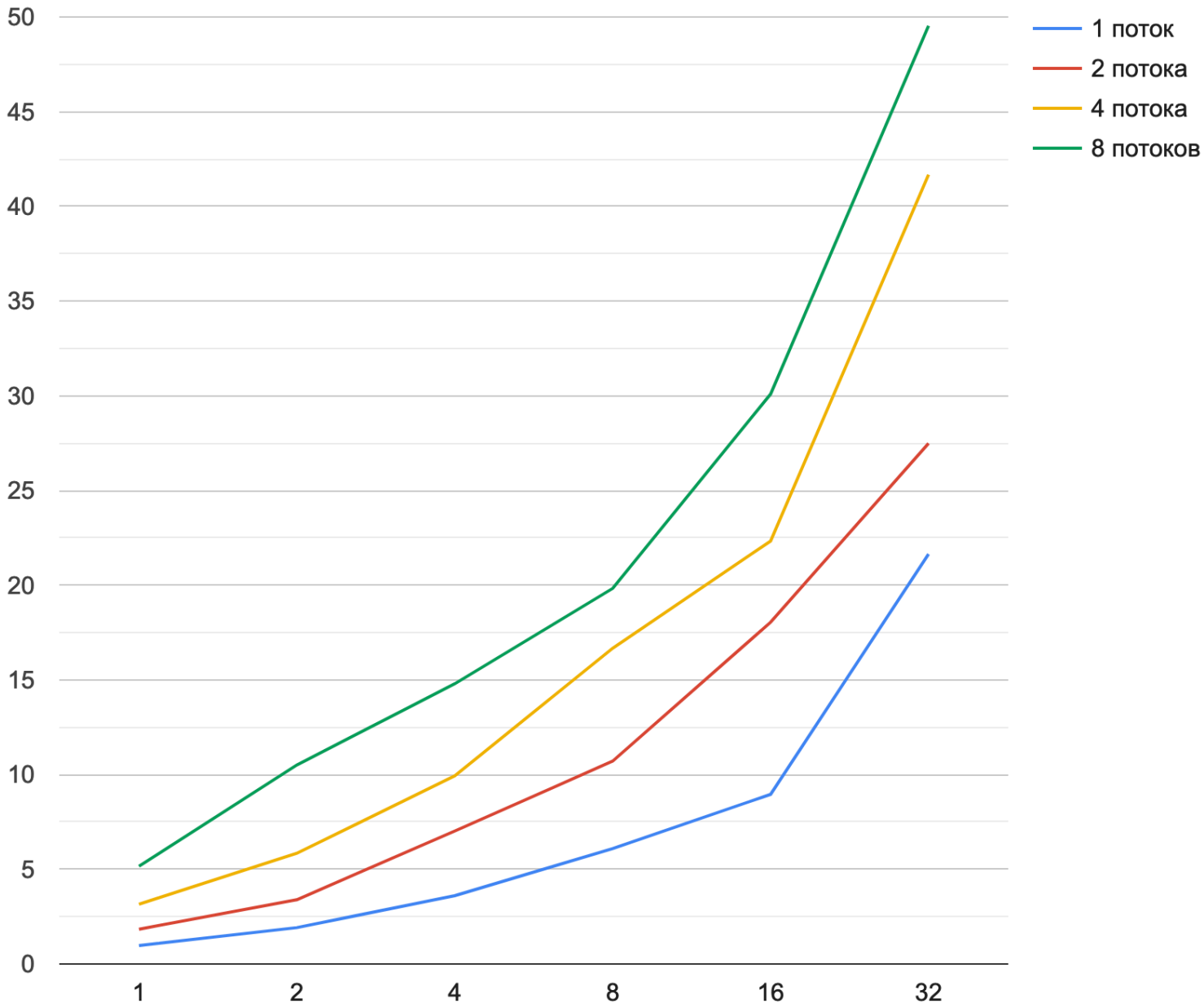
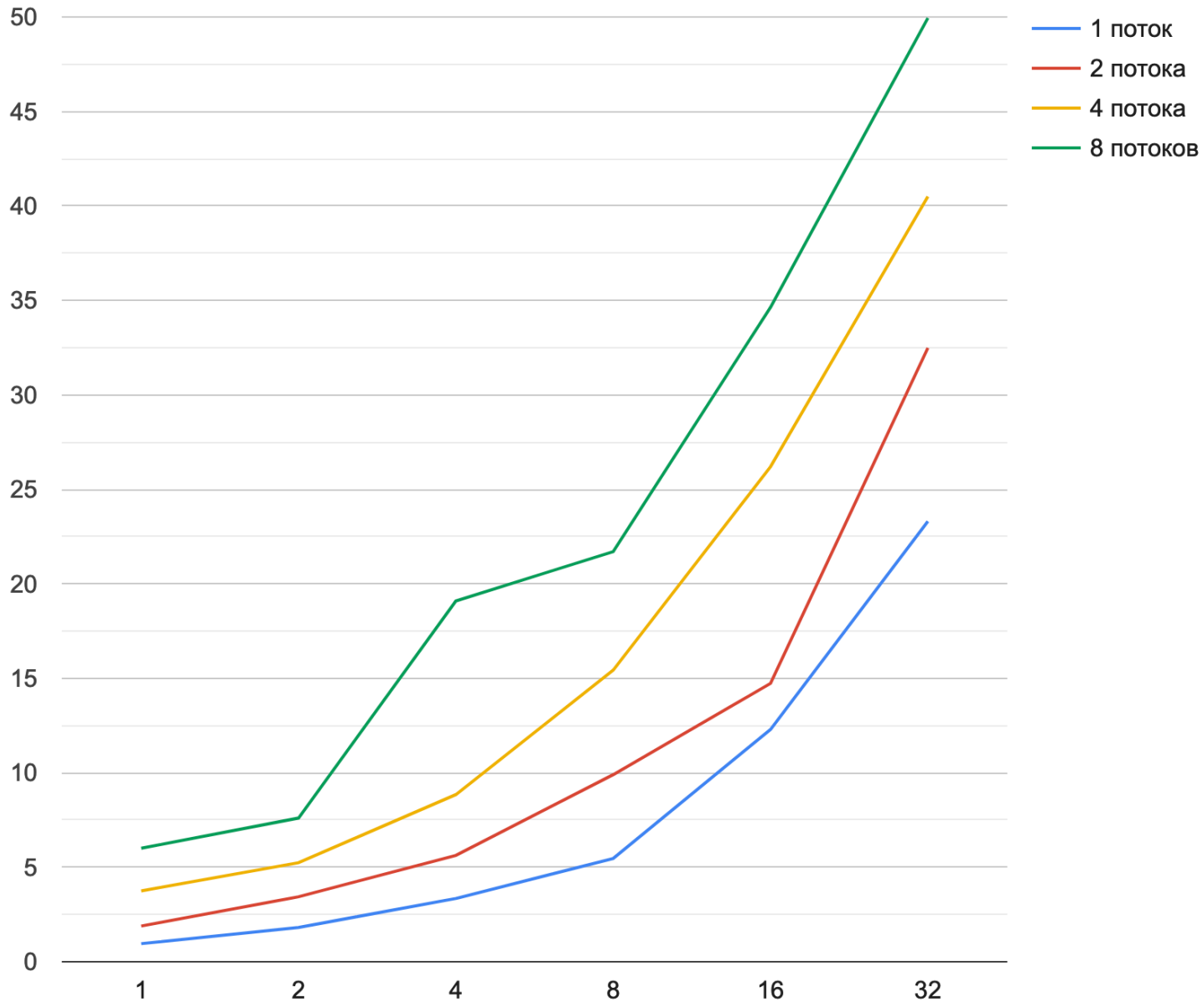
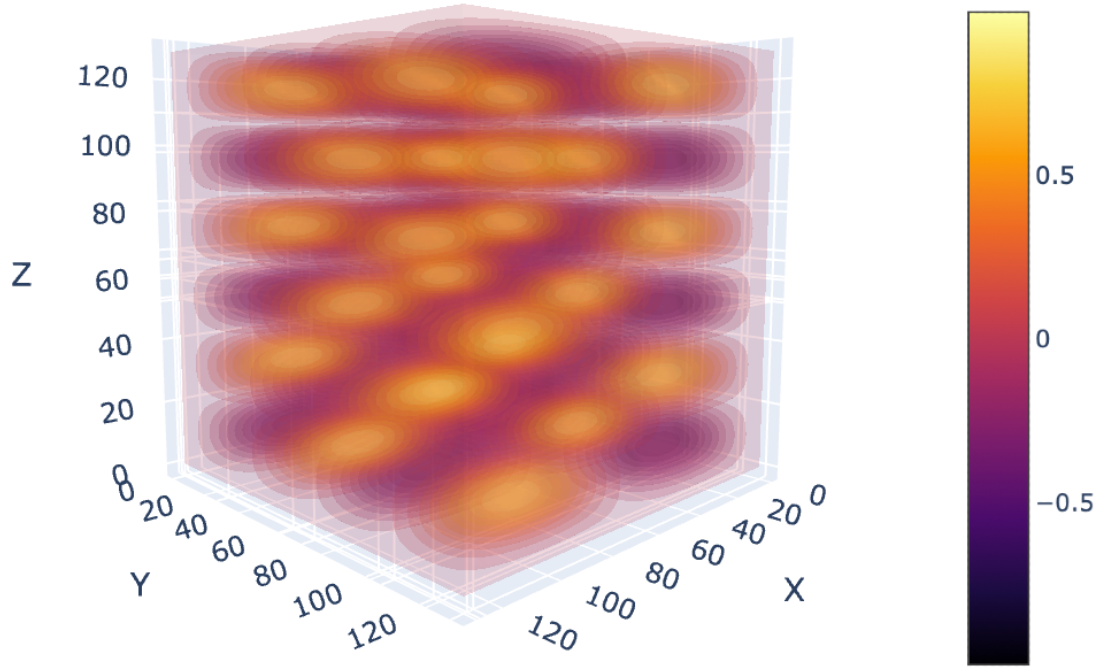
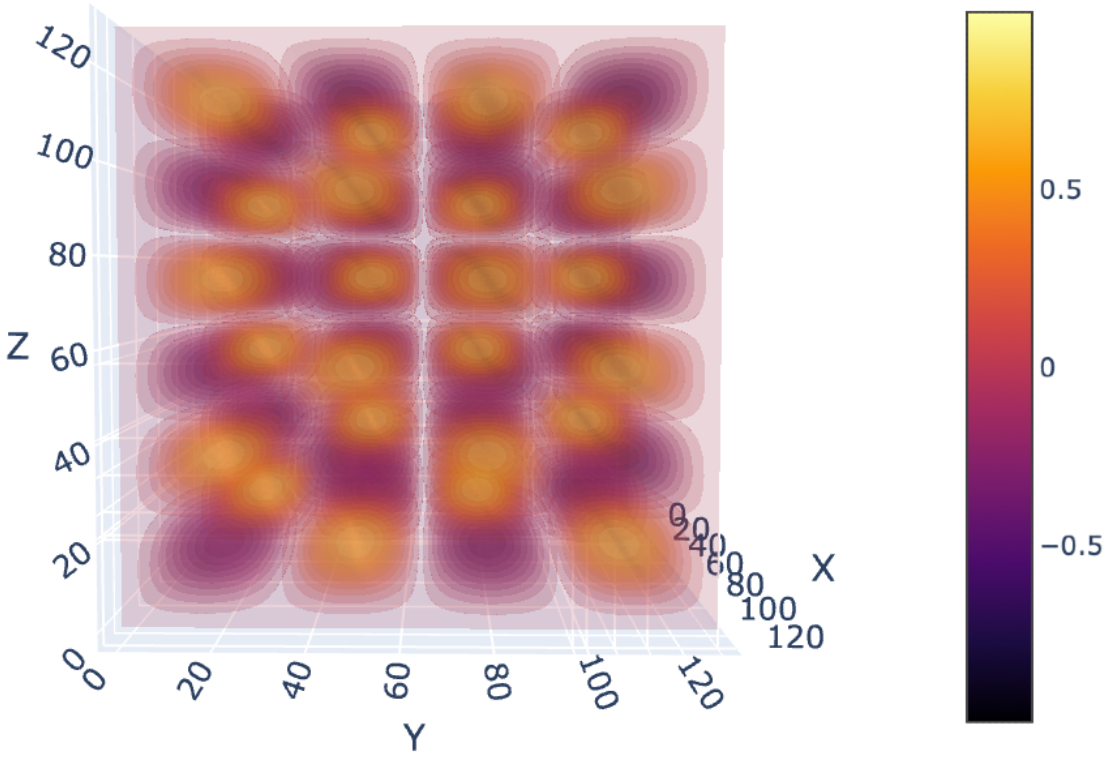


График 6: зависимость ускорения на Polus от числа процессов для сетки 512 при L = π:

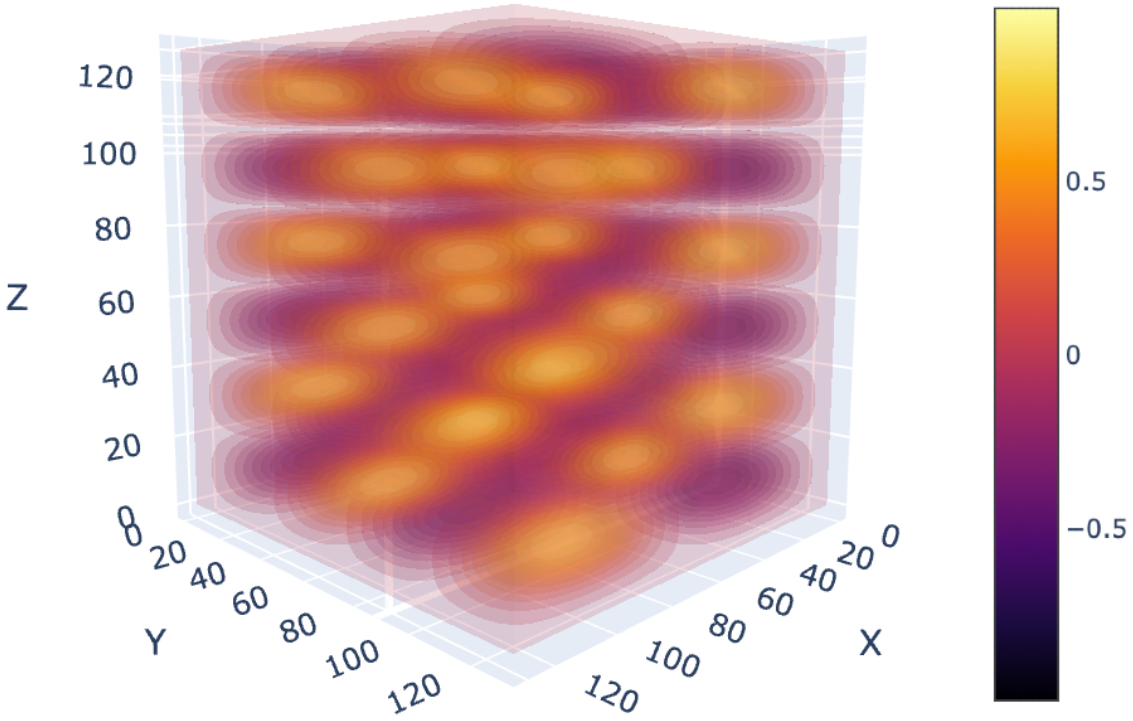


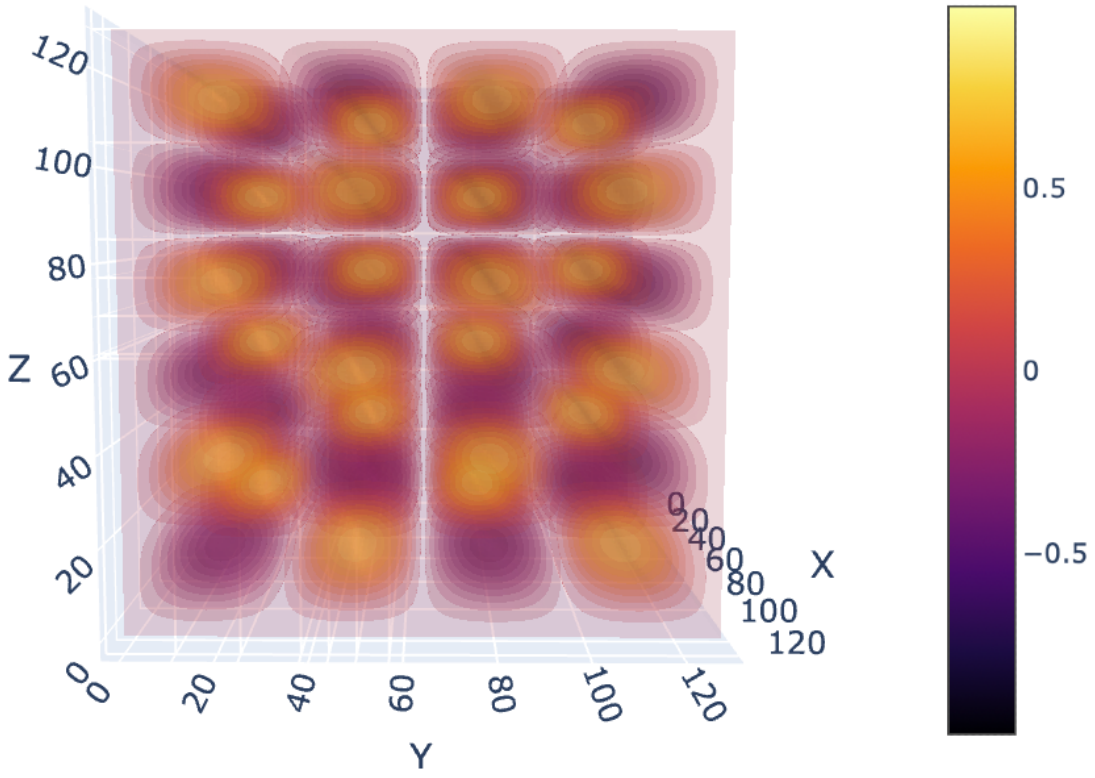
Визуализация сетки, полученной аналитическим решением:



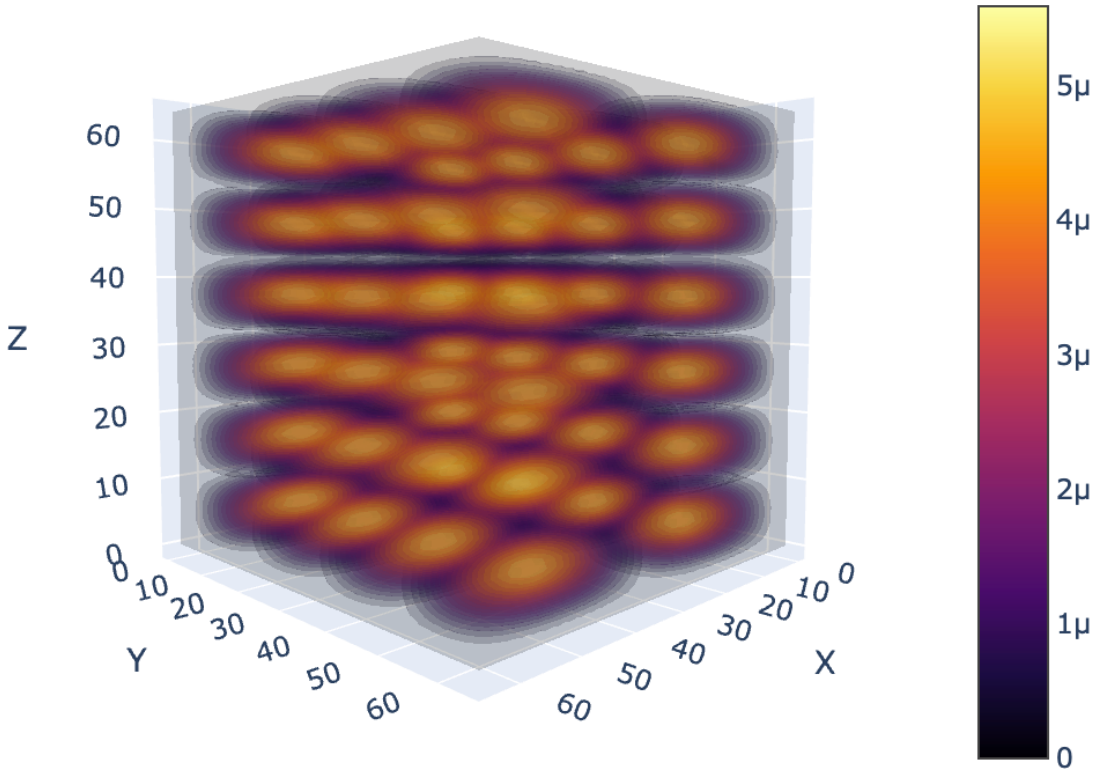


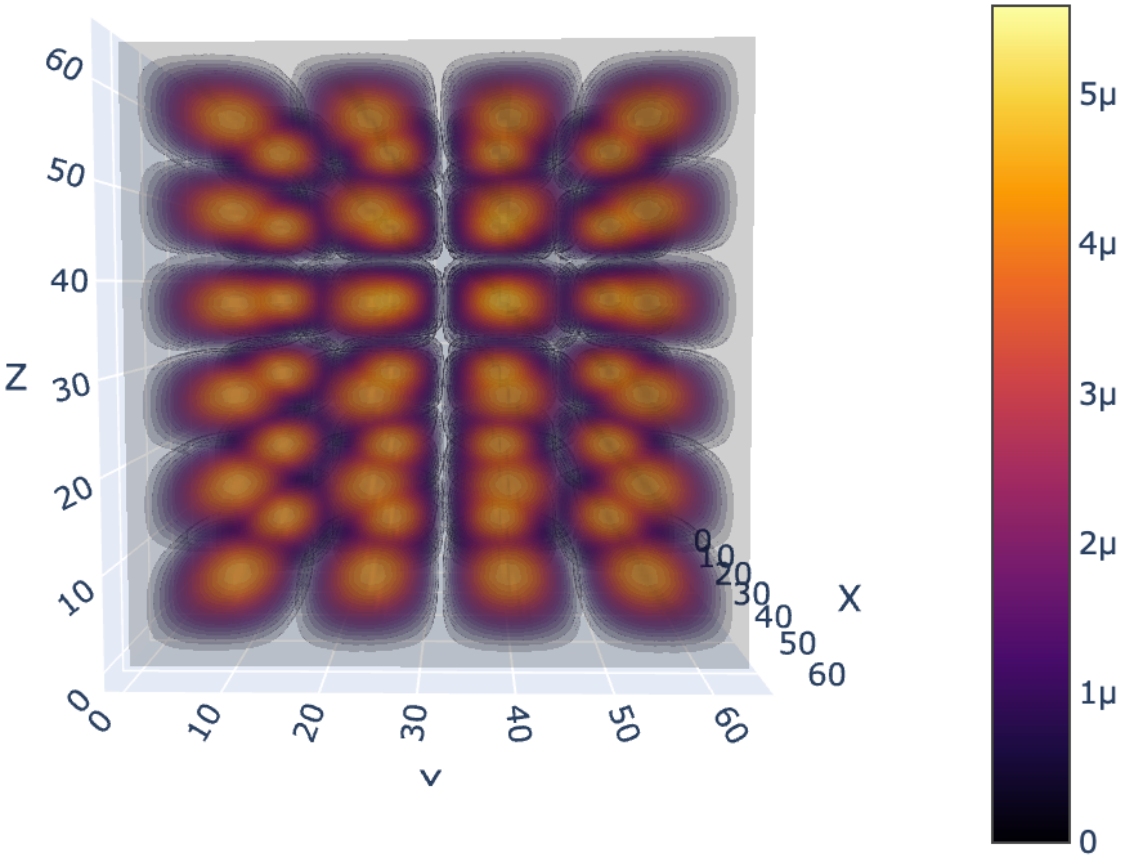
Визуализация сетки, полученной численным решением:





Визуализация сетки погрешности:





Вывод

Задача для трехмерного гиперболического уравнения в области, представляющей из себя прямоугольный параллелепипед, подходит для распараллеливания с помощью технологии MPI и OpenMP в смешанной реализации, позволяя получить ускорение вплоть до 45-49 раз. Причем при большем размере сетки ниже погрешность, а распараллеливание дает лучшие результаты по сравнению с мелкими сетками.

В сравнении с OpenMP и MPI решением мы получили такую же точность, но больший прирост в скорости работы.