Московский Государственный Университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

Численное решение краевой задачи для уравнения Пуассона с потенциалом в прямоугольной области

Выполнил: Мякшин Владислав Эдуардович, 614 группа
9 вариант

Москва, 2022 г.

Постановка задачи

В прямоугольнике $\Pi = (x, y): A_1 \leq x \leq A_2, B_1 \leq y \leq B_2$, граница Γ которого состоит из отрезков

$$\lambda_R = \{(A\_2, y), B\_1 \leq y \leq B\_2\}, \lambda_\mathrm{L} = \{(A\_1, y), B\_1 \leq y \leq B\_2\},$$

$$\lambda_\mathrm{T} = \{(x, B\_2), A\_1 \leq y \leq A\_2\}, \lambda_\mathrm{B} = \{(x, B\_1), A\_1 \leq y \leq A\_2\},$$

Рассматривается дифференциальное уравнение Пуассона с потенциалом

$$-\Delta u + q(x, y)u = F(x, y), \tag{1}$$

в котором оператор Лапласа

$$\Delta \mathrm{u} = \frac{\partial}{\partial x}\left(k(x, y)\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(k(x, y)\frac{\partial}{\partial y}\right)$$

Для выделения единственного решения уравнение (1) дополняется граничными условиями. На каждом отрезке границы прямоугольника Π задаются следующие условия:

$$\lambda_R: \left(k\frac{\partial u}{\partial n}\right)(x, y) = \psi(x, y) \qquad \lambda_\mathrm{L}: \left(k\frac{\partial u}{\partial n}\right)(x, y) = \psi(x, y)$$

$$\lambda_\mathrm{T}: \left(k\frac{\partial u}{\partial n}\right)(x, y) = \psi(x, y) \qquad \lambda_\mathrm{B}: \left(k\frac{\partial u}{\partial n}\right)(x, y) = \psi(x, y)$$

Функция $F(x, y)$, $\phi(x, y)$, коэффициент $k(x, y)$, потенциал $q(x, y)$ считаются известными. Требуется найти функцию $u(x, y)$, удовлетворяющую уравнению (1) и граничным условиям.

1. Пользуясь явным видом функций $u(x, y), k(x, y), q(x, y)$ определить правую часть уравнения Пуассона $F(x, y)$ и граничные условия;
2. Собрать разностную схему для уравнения Пуассона с граничными условиями;
3. Разработать последовательный код программы, вычисляющий приближенное решение разностной схемы методом наименьших невязок, проверить точность схемы, выполнив расчеты на сгущающихся сетках
$$(M, N) = (20{,}20), (40{,}40), (80{,}80), (160{,}160);$$
4. Используя средства библиотеки MPI, разработать параллельный код программы, вычисляющий приближенное решение разностной схемы методом наименьших невязок, проверить качество работы алгоритма, выполнив расчеты на сетке $(M, N) = (160, 160)$ на одном, четырех и шестнадцати процессах;
5. Провести исследование параллельных характеристик MPI-программы, выполнив расчеты на вычислительном комплексе IBM Polus;
6. Разработать гибридный MPI/ OpenMP код программы, провести исследование параллельных характеристик гибридной программы и сравнить полученные результаты с программой, не использующей директивы OpenMP.

В соответствии с вариантом задания рассматриваю следующие данные:

- $A_1 = -2, A_2 = 3, B_1 = -1, B_2 = 4$
- $u(x, y) = \frac{2}{1 + x^2 + y^2}$
- $k(x, y) = 1 + (x + y)^2$
- $q(x, y) = 1$

Задача практикума заключается в восстановлении известной гладкой функции $u(x, y)$ по ее образу $F(x, y) = -\Delta u + q(x, y)$ и ее граничным значениям.

Определение функций $F(x,y), \psi(x,y)$

Определим функцию $F(x,y)$. Для этого вычислим оператор Лапласа, используя явный вид функций $u(x,y), k(x,y)$

$$\frac{\partial u}{\partial x} = \frac{-4x}{1+x^2+y^2}$$

$$\frac{\partial u}{\partial x} = \frac{-4y}{1+x^2+y^2}$$

$$\Delta u = \frac{16x^2}{(1+x^2+y^2)^3} - \frac{8}{(1+x^2+y^2)^2} + \frac{16y^2}{(1+x^2+y^2)^3}$$

$$F(x,y) = -\frac{16x^2+16y^2}{(1+x^2+y^2)^3} + \frac{8}{(1+x^2+y^2)^2} + \frac{2}{1+x^2+y^2}$$

Граничные условия:

$$\psi(x,y) = \begin{cases} u_x(-2,y) = \dfrac{8}{(5+y^2)^2} \\ u_x(3,y) = \dfrac{-12}{(10+y^2)^2} \\ u_y(x,-1) = \dfrac{4}{(2+x^2)^2} \\ u_y(x,4) = \dfrac{-16}{(17+x^2)^2} \end{cases}$$

Разностная схема решения задачи

Краевые задачи для уравнения Пуассона с потенциалом (1) предлагается численно решать методом конечных разностей. В расчетной области П определяется равномерная прямоугольная сетка $\overline{\omega}_h = \overline{\omega}_1 \times \overline{\omega}_2$, где

$$\overline{\omega}_1 = \{x_i = -2 + ih_1, i = \overline{0, M}\}, \qquad \overline{\omega}_2 = \{y_j = -1 + jh_2, j = \overline{0, N}\}$$

Здесь $h_1 = 5/M, h_2 = 5/N$. Через ω_h обозначим множество внутренних узлов сетки $\overline{\omega}_h$, т.е. множество узлов сетки прямоугольника, не лежащих на границе Г.

Рассмотрим линейное пространство H функций, заданных на сетке $\overline{\omega}_h$. Обозначим через w_{ij} значение сеточной функции $w \in H$ в узле сетки $(x_i, y_j) \in \overline{\omega}_h$. Будем считать, что в пространстве H задано скалярное произведение и евклидова норма

$$[\mathrm{u}, \mathrm{v}] = \sum_{i=0}^{M} h_1 \sum_{j=0}^{N} h_2 \rho_{ij} u_{ij} v_{ij} = h_1 h_2 \sum_{i=0}^{M} \sum_{j=0}^{N} \rho_{ij} u_{ij} v_{ij},$$

$$\|u\|_E = \sqrt{[u, u]}$$

Весовая функция $\rho_{i,j} = \rho^{(1)}(x_i)\rho^{(2)}(y_j)$, где

$$\rho^{(1)}(x_i) = \begin{bmatrix} 1, & 1 \le i \le M-1 \\ 1/2, & i = 0, i = M \end{matrix}, \qquad \rho^{(2)}(y_j) = \begin{bmatrix} 1, & 1 \le i \le N-1 \\ 1/2, & i = 0, i = N \end{matrix}$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вида

$$Aw = B,$$

где $A : H \to H$ – оператор, действующий в пространстве сеточных функций, B \in H – известная правая часть. Задача (5) называется разностной схемой. Решение этой задачи считается численным решением исходной дифференциальной задачи.

При построении разностной схемы следует аппроксимировать все уравнения краевой задачи их разностными аналогами – сеточными уравнениями, связывающими значения искомой сеточной функции в узлах сетки. Полученные таким образом уравнения должны быть функционально независимыми, а их общее количество – совпадать с числом неизвестных, т.е. с количеством узлов сетки.

Уравнение (1) во всех внутренних точках сетки аппроксимируется разностным уравнением

$$-\Delta_h w_{ij} + q_{ij} w_{ij} = F_{ij}, i = \overline{1, M-1}, j = \overline{1, N-1},$$

в котором $F_{ij} = F(x_i, y_j), q_{ij} = q(x_i, y_j)$, разностный оператор Лапласа

$$\Delta_h w_{ij} = \frac{1}{h_1}\left(k(x_i + 0.5h_1, y_j)\frac{w_{i+1j} - w_{ij}}{h_1}\right) - k(x_i - 0.5h_1, y_j)\frac{w_{ij} - w_{i-1j}}{h_1} +$$
$$+\frac{1}{h_2}\left(k(x_i, y_j + 0.5h_2)\frac{w_{ij+1} - w_{ij}}{h_2}\right) - k(x_i, y_j - 0.5h_2)\frac{w_{ij} - w_{ij-1}}{h_2}.$$

Введем обозначения правой и левой разностных производных по переменным x, y соответственно:

$$w_{x,ij} = \frac{w_{i+1j} - w_{ij}}{h_1}, \qquad w_{\bar{x},ij} = w_{x,i-1j} = \frac{w_{ij} - w_{i-1j}}{h_1},$$

$$w_{y,ij} = \frac{w_{ij+1} - w_{ij}}{h_2}, \qquad w_{\bar{y},ij} = w_{x,ij-1} = \frac{w_{ij} - w_{ij-1}}{h_2},$$

а также определим сеточные коэффициенты

$$a_{ij} = k(x_i - 0.5h_1, y_j), b_{ij} = k(x_i, y_i - 0.5h_2)$$

С учетом принятых обозначений (6)-(7) разностный оператор Лапласа можно представить в более компактном и удобном виде

$$\Delta_h w_{ij} = (aw_{\bar{x}})_{x,ij} + (bw_{\bar{y}})_{y,ij}.$$

В самом деле, из (5) с учетом (6)-(7) получаем

$$\Delta_h w_{ij} = \frac{1}{h_1}\left(a_{i+1j}w_{x,ij} - a_{ij}w_{\bar{x},ij}\right) + \frac{1}{h_2}\left(b_{ij+1}w_{y,ij} - b_{ij}w_{\bar{y},ij}\right) =$$

$$= \frac{1}{h_1}\left(a_{i+1j}w_{\bar{x},i+1j} - a_{ij}w_{\bar{x},ij}\right) + \frac{1}{h_2}\left(b_{ij+1}w_{\bar{y},ij+1} - b_{ij}w_{\bar{y},ij}\right) =$$

$$= (aw_{\bar{x}})_{x,ij} + (bw_{\bar{y}})_{y,ij}.$$

Аппроксимация граничных условий второго рода для задачи описанной вариантом имеет вид:

$\frac{2}{h_2}\left(bw_{\bar{y}}\right)_{iN} + q_{iN}w_{iN} - (aw_{\bar{x}})_{x,iN} = F_{iN} + \frac{2}{h_2}\psi_{iN}$ *– верх*

$-\frac{2}{h_2}\left(bw_{\bar{y}}\right)_{i1} + q_{i0}w_{i0} - (aw_{\bar{x}})_{x,i0} = F_{i0} + \frac{2}{h_2}\psi_{i0}$ *- низ*

$-\frac{2}{h_1}(aw_{\bar{x}})_{1j} + q_{0j}w_{0j} - \left(bw_{\bar{y}}\right)_{y,0j} = F_{0j} + \frac{2}{h_1}\psi_{0j}$ *– слева*

$\frac{2}{h_1}(aw_{\bar{x}})_{Mj} + q_{Mj}w_{Mj} - \left(bw_{\bar{y}}\right)_{y,Mj} = F_{Mj} + \frac{2}{h_1}\psi_{Mj}$ *- справа*

Сеточных уравнений недостаточно, чтобы определить разностную схему для задачи с граничными условиями Неймана. Требуются сеточные уравнения для угловых точек прямоугольника П. Они имеют следующий вид:

Сеточные уравнения для угловых точек прямоугольника:

$-(2/h_1)(a\omega_{\bar{x}})_{10} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{01} + q_{00}\omega_{00} = F_{00} + (2/h_1 + 2/h_2)\psi_{00}$

$-(2/h_1)(a\omega_{\bar{x}})_{M0} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{M1} + q_{M0}\omega_{M0} = F_{M0} + (2/h_1 + 2/h_2)\psi_{M0}$

$-(2/h_1)(a\omega_{\bar{x}})_{10} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{MN} + q_{MN}\omega_{MN} = F_{MN} + (2/h_1 + 2/h_2)\psi_{MN}$

$-(2/h_1)(a\omega_{\bar{x}})_{1N} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{0N} + q_{0N}\omega_{0N} = F_{0N} + (2/h_1 + 2/h_2)\psi_{0N}$

Соберем все уравнения, получим систему линейных алгебраических уравнений (СЛАУ):

$$\begin{cases} -\Delta_h\omega_{ij} + q_{ij}\omega_{ij} = F_{ij}, \\ \frac{2}{h_2}\left(bw_{\bar{y}}\right)_{iN} + q_{iN}w_{iN} - (aw_{\bar{x}})_{x,iN} = F_{iN} + \frac{2}{h_2}\psi_{iN} \\ -\frac{2}{h_2}\left(bw_{\bar{y}}\right)_{i1} + q_{i0}w_{i0} - (aw_{\bar{x}})_{x,i0} = F_{i0} + \frac{2}{h_2}\psi_{i0} \\ -\frac{2}{h_1}(aw_{\bar{x}})_{1j} + q_{0j}w_{0j} - \left(bw_{\bar{y}}\right)_{y,0j} = F_{0j} + \frac{2}{h_1}\psi_{0j} \\ \frac{2}{h_1}(aw_{\bar{x}})_{Mj} + q_{Mj}w_{Mj} - \left(bw_{\bar{y}}\right)_{y,Mj} = F_{Mj} + \frac{2}{h_1}\psi_{Mj} \\ -(2/h_1)(a\omega_{\bar{x}})_{10} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{01} + q_{00}\omega_{00} = F_{00} + (2/h_1 + 2/h_2)\psi_{00} \\ -(2/h_1)(a\omega_{\bar{x}})_{M0} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{M1} + q_{M0}\omega_{M0} = F_{M0} + (2/h_1 + 2/h_2)\psi_{M0} \\ -(2/h_1)(a\omega_{\bar{x}})_{10} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{MN} + q_{MN}\omega_{MN} = F_{MN} + (2/h_1 + 2/h_2)\psi_{MN} \\ -(2/h_1)(a\omega_{\bar{x}})_{1N} - (2/h_2)\left(b\omega_{\bar{y}}\right)_{0N} + q_{0N}\omega_{0N} = F_{0N} + (2/h_1 + 2/h_2)\psi_{0N} \end{cases}$$

Метод решения системы линейных алгебраических уравнений

Приближенное решение системы уравнений (6) для сформулированных выше краевых задач может быть получено итерационным методом наименьших невязок. Этот метод позволяет получить последовательность сеточных функций $\omega^{(k)} \in H, k = 1,2, \ldots$, сходящуюся по норме пространства H к решению разностной схемы, т.е.

$$\left\|\omega - \omega^{(k)}\right\|_E \to 0, k \to \infty$$

Начальное приближение $\omega^{(0)}$ можно выбрать любым способом, например, равным нулю во всех точках расчетной сетки.

Метод является одношаговым. Итерация $\omega^{(k+1)}$ вычисляется по итерации ω^k согласно равенствам:

$$\omega_{ij}^{(k+1)} = \omega_{(ij)}^{(k)} - \tau_{k+1} r_{ij}^{(k)},$$

где невязка $r^{(k)} = A\omega^{(k)} - B$, итерационный параметр

$$\tau_{k+1} = \frac{\left[Ar^{(k)}, r^{(k)}\right]}{\left\|Ar^{(k)}\right\|_E^2}.$$

В качестве условия остановки итерационного процесса следует использовать неравенство

$$\left\|\omega^{(k+1)} - \omega^{(k)}\right\|_E < \epsilon,$$

где ϵ- положительное число, определяющее точность итерационного метода. Оценку точности приближенного решения сеточных уравнений можно проводить в других нормах пространства сеточных функций, например, в максимум норме

$$\|\omega\|_C = \max_{x \in \overline{\omega}_h} |\omega(x)|$$

Константу ϵ для данной задачи предлагается взять равное 10^{-6}.

Краткое описание проделанной работы по созданию MPI программы и гибридной реализации MPI/OpenMP

Была разобрана последовательная программа, вычисляющая приближенное решение разностной схемы методом наименьших невязок, и проверена точность вычислений.

$$(M, N) = (20{,}20), (40{,}40), (80{,}80), (160{,}160)$$

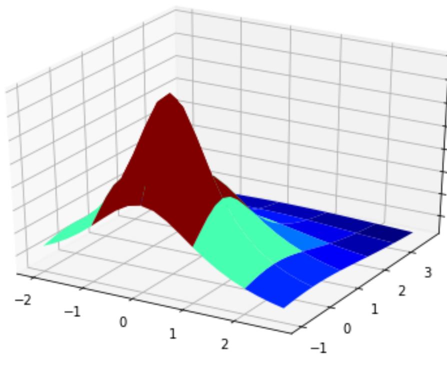

Рисунок 1: (M, N) = (20, 20). Число итераций: 980.

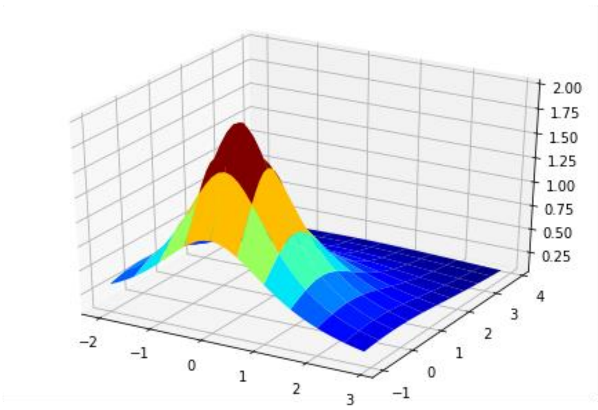

Рисунок 2: (M, N) = (40, 40). Число итераций: 3671.

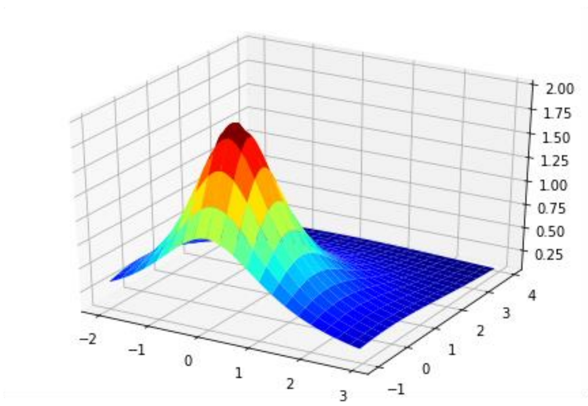

Рисунок 3: (M, N) = (80, 80). Число итераций: 12421.

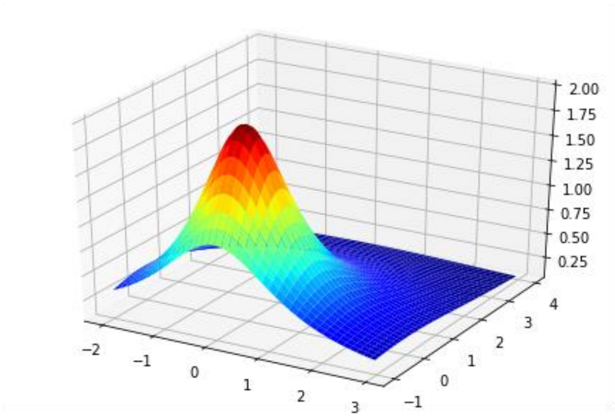

Рисунок 4: (M, N) = (160, 160). Число итераций: 43322.

Расчетная область разбита с помощью двумерного разбиения на подобласти таким образом, что отношение количества узлов по переменным x и y в каждой подобласти принадлежит диапазону [1/2, 2], и количество узлов по переменным x и y любых двух подобластей отличается не более чем, на единицу.

В каждой подобласти была создана своя часть сетки. Каждый процесс для своей части сетки считает $(B, \omega) \rightarrow A\omega \rightarrow (A\omega - B) \rightarrow r \rightarrow Ar$.

Метод минимальных невязок работает до момента, пока евклидова норма разницы решения на текущем и предыдущем шаге не будет меньше 10^{-6}.

В ходе работы метода для расчета работы оператора A на матрицы ω и r используется разностные производные. Для производных нужны значения крайних векторов от соседних процессов, для этого используется MPI_Isend, для отправки без блокирования текущего процесса, и MPI_Recv, для получения соседних векторов, прием блокирующий, так как без получения этих значений выполнение дальнейшего расчета процессов невозможен.

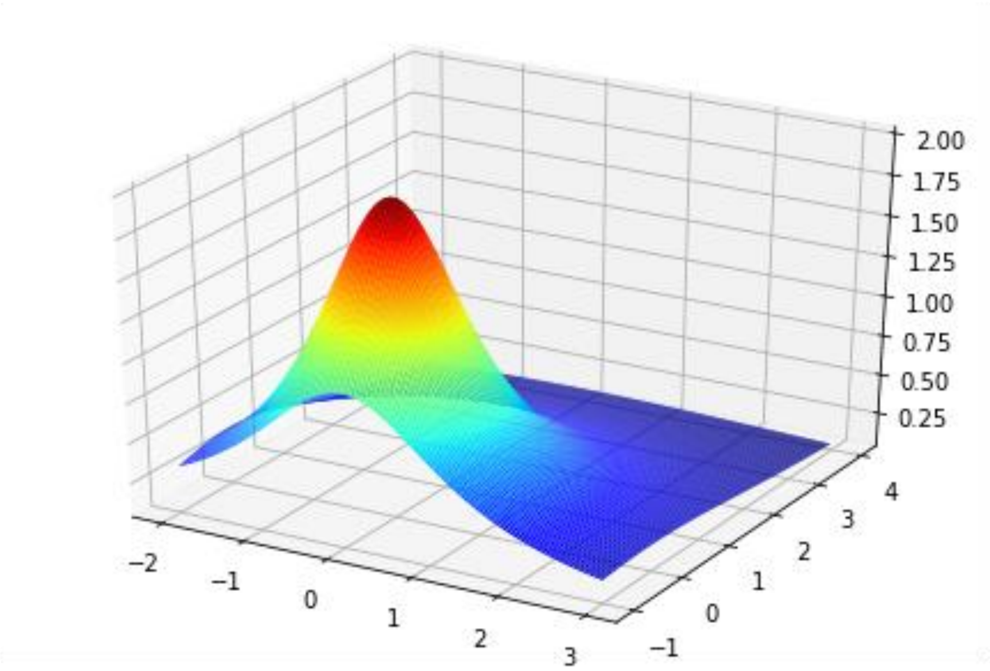

Рисунок 5: (M, N) = (500, 1000)

Число процессов MPI	Число точек сетки $M \times N$	Время решения (с)	Ускорение
4	500×500	210,256	1
8	500×500	111,251	1,88992458
16	500×500	62,198	3,38043024
32	500×500	32,934	6,38416226
4	500×1000	340,609	1
8	500×1000	197,628	1,72348554
16	500×1000	110,382	3,08572956
32	500×1000	56,653	6,01219706

Таблица 1. Таблица с результатами расчетов на ПВС IBM Polus (MPI)

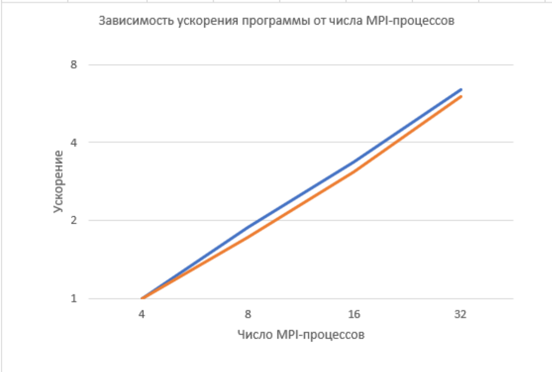

Рисунок 6: Зависимость ускорения программы от числа MPI-процессов на Polus

Множество циклов в данной задаче можно распараллелить с помощью директив OpenMP. Это возможно сделать, так как значения внутри циклов не зависят друг от друга. Операция умножения матрицы на матрицу тоже была распараллелена, используя pragma omp parallel for reduction(+:ans). Но прироста времени по сравнению с обычным MPI нет.

Число процессов MPI	Количество OMP нитей в процессе	Число точек сетки $M \times N$	Время решения (с)	Ускорение
1	4	500×500	410,565	1
2	4	500×500	230,923	1,77793031
4	4	500×500	123,84	3,31528585
8	4	500×500	65,431	6,27477801
1	4	500×1000	611,321	1
2	4	500×1000	380,456	1,6068113
4	4	500×1000	183,341	3,33433875
8	4	500×1000	99,345	6,15351553

Таблица 2. Таблица с результатами расчетов на ПВС IBM Polus (MPI-OpenMP)

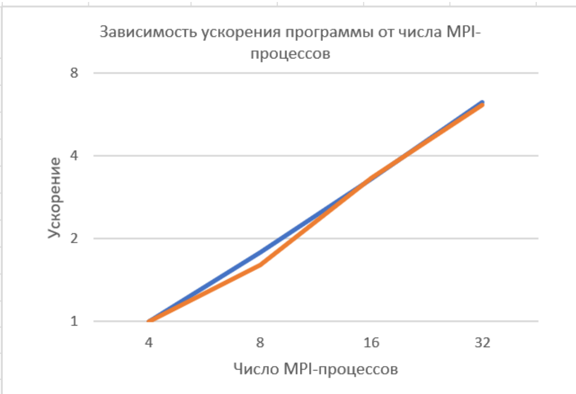

Рисунок 7: Зависимость ускорения программы с OMP от числа MPI-процессов на Polus