# Отчет по домашнему заданию №2 по курсу "Суперкомпьютерное моделирование и технологии"

Синякова Марина Алексеевна, студентка 621 группы

Ноябрь 2020

## 1 Математическая постановка задачи

В трехмерной замкнутой области

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

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

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

с начальными условиями

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

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

при условии, что на границах области заданы однородные граничные условия первого рода и периодические условия

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

Требуется найти численным методом решение это задачи.

Для проверки точности решения зададим аналитическое решение, которое будет иметь вид:

$$u_{analytical} = \sin\left(\frac{\pi}{L_x}x\right) \cdot \sin\left(\frac{2\pi}{L_y}\right) \cdot \sin\left(\frac{3\pi}{L_z}z\right) \cdot \cos\left(a_t \cdot t\right), \quad a_t = \pi\sqrt{\frac{1}{L_x^2} + \frac{4}{L_y^2} + \frac{9}{L_z^2}} \quad (1)$$

# 2 Численный метод решения

На области $\Omega$ введем разбиение $\omega_{h\tau} = \omega_h \times \omega_\tau$, где

$$\omega_h = \{(x_i = ih_x, y_j = jh_y, z_k = kh_z), i, j, k = 0, 1, \ldots, N,\ h_x N = L_x, h_y N = L_y, h_z N = L_z\} \tag{2}$$

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

Для апроксимации решаемой задачи воспользуемся явной разностной схемой:

$$\frac{u^{n+1}_{i,j,k} - 2u^n_{i,j,k} + u^{n-1}_{i,j,k}}{\tau^2} = \Delta_h u^n, \quad (x_i, y_j, z_k) \in \omega_h, \quad n = 1, 2, \ldots, K-1 \tag{3}$$

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

Функция $\phi$ будет иметь вид:

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

Для того чтобы воспользоваться разностной схемой (3), необходимо вычислить 0 и 1 временной слой. Нулевой временной слой вычисляется по формуле:

$$u^0_{i,j,k} = \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \omega_h.$$

Первый временной слой вычисляем по формуле:

$$u^1_{i,j,k} = u^0_{i,j,k} + \frac{\tau^2}{2}\delta_h \phi(x_i, y_j, z_k)$$

Так как для вычисления первого слоя используется оператор Лапласа, то граничные значения области будут получены из аналитического решения (1).

Для вычисления последующих временных слоев воспользуемся разностной схемой (3) для внутренних точек области (2), а для вычисления граничных значений воспользуемся граничными условиями исходной задачи.

Граничные условия первого рода будут нулевыми на любом временном слое:

$$u(0, y, z, t) = 0 \ \Rightarrow \ u^{n+1}_{0,j,k} = 0$$
$$u(L_x, y, z, t) = 0 \ \Rightarrow \ u^{n+1}_{N,j,k} = 0$$
$$u(x, y, 0, t) = 0 \ \Rightarrow \ u^{n+1}_{i,j,0} = 0$$
$$u(x, y, L_z, t) = 0 \ \Rightarrow \ u^{n+1}_{i,j,N} = 0$$

Периодические граничные условия будут иметь вид:

$$u(x, 0, y, z, t) = u(x, L_y, z, t), \ \Rightarrow \ u^{n+1}_{i,0,k} = u^{n+1}_{i,N,k}$$

$$u_y(x,0,y,z,t) = u_y(x,L_y,z,t) \quad \Rightarrow \quad \frac{u^{n+1}_{i,1,k} - u^{n+1}_{i,1,k}}{h} = \frac{u^{n+1}_{i,N,k} - u^{n+1}_{i,N-1,k}}{h} \qquad (4)$$

Так как область у нас ограничена, то при подсчете производной на границе при $y = 0$ вычисляем производную при стремление к 0 справа. На границе $y = L_y$ вычисляется производная при стремление к $L_y$ слева. Получим соотношения (4). Домножив обе части правого равенства (4) на $h$ и подставим соотношение (2), получим:

$$\frac{1}{2}\left(u^{n+1}_{i,1,k} + u^{n+1}_{i,N-1,k}\right) = u^{n+1}_{i,0,k} = u^{n+1}_{i,N,k}.$$

# 3 Реализация MPI и OpenMP

Разбиение (2) разделено на блоки. Каждый блок вычисляется на одном процессоре. Число блоков, на которое разбивается сетка (2), меньше или равно числу процессов, на которых могут выполняться блоки.

Изначально получаем доступное количество процессоров при помощи команды `MPI_Comm_size`. Затем заполняем размерность распределения процессоров в пространстве командой `MPI_Dims_create`. Создаем новый коммуникатор, к которому уже добавлена информация о топологии при помощи `MPI_Cart_create`. Узнаем и сохраняем свой ранг `MPI_Comm_rank` и ранг своих соседей `MPI_Cart_shift` дальнейшего обмена внутренними значениями блоков. Процессор узнает свои координаты в пространстве при помощи команды `MPI_Cart_coords`. Используя эту информацию распределяем сетку (2) между процессорами. Для каждого процессора вычисляем начало сетки и ее размер по каждой координате.

Так как для вычисления каждого значения сетки нужно получить значения $-1$ и $+1$ по каждой координате, то необходимо реализовать обмен гранями блоков между процессорами. Данный обмен реализован при помощи команды `MPI_Sendrecv`. Также стоит отметить, что граничные точки блока не всегда являются граничными точками области и будут вычисляться по формулами (3), а не описываться граничными условиями.

На каждой итерации по времени в каждом блоке вычисляется отклонение от аналитического решения в каждом узле блока. По окончанию вычисления временного слота, происходит обмен между блоками максимальным значением отклонения в блоке, с целью выбора максимального из отклонений. Данная операция осуществляется командой `MPI_Allreduce`. По окончанию работы программы аналогичным образом происходит обмен между блоками временем работы программы и выбирается максимальное значение. Время замеряется при помощи функции `MPI_Wtime()`.

В качестве основной директивы OpenMP была использована `pragma omp parallel for` с параметрами `shared` и `private`. В качестве `shared` указывались матрицы вычисляемого и предыдущих слоев. В качестве `private` указывались индексы массивов. Директива `pragma omp parallel for` использовалась для распараллеливания вычисления внутренних значений сетки для 0 и 1 временных слоев, а также для распараллеливания вычислений значений сетки каждого последующего временного слоя. По результатам экспериментов, наибольшее ускорение было получено при распараллеливание только внешнего из трех вложенных циклов по сетке, иначе больше времени тратится на распределение данных по нитям, нежели на вычисления. Также стоит отметить, что распараллелить цикл по времени нельзя, так как в нем существует зависимость от двух предыдущих временных слоев.

Вычисление граничных условий области зависит от внутренних точек сетки, которые вычисляются чуть раньше. Каждую грань можно вычислять независимо от другой, для этого была использована директива `pragma omp sections`, где в качестве каждой секции вычислялась одна грань.

## 4 Графики аналитического и посчитанного решений

Для наглядного сравнения полученного решения и аналитического, построим графики обоих решений. Так как область решения трехмерная, то возьмем срез по каждой переменной. На рисунке (1) представлены сравнения двух решений на слоях при $x = 0.5$, $y = 0.25$ и $z = 0.5$.

## 5 График погрешности

Как видно из графиков (2) и (3) для обоих значений $L$ с увеличением числа точек в сетке отклонение вычисленного решения от аналитического уменьшается. Так как шаг по сетке $\omega_h$ меньше и решение получается вычислить более точно. Поэтому чем больше узлов сетки у нас будет, тем меньше будет отклонение от аналитического решения. Но не стоит забывать, что при этом понадобится намного больше времени на вычисление задачи.

Также стоит отметить, что отклонение от аналитического решения растет на каждом шаге по времени. Это обусловлено тем, что ошибка наращивается на каждом временном слое. При переходе с одного момента времени на следующий ошибка, из-за аппроксимации производной по времени увеличивается, потому что складывается с ошибкой на предыдущем временном слое. Это широко известный эффект в теории разностных сеток, поэтому можно сказать, что данный метод расходится.

## 6 Результаты расчетов

Результаты расчетов на Blue Gene с использованием MPI приведены в таблице (1).

Погрешность решения не зависит от количества процессоров, на которых происходит вычисления. Погрешность уменьшается с увеличением числа узлов сетки. Время решения задачи уменьшается пропорционально увеличению числу процессоров, которые использовались при вычислении.

| L | Число точек сетки $N^3$ | Число процессоров | Время решения | Ускорение | Погрешность |
|---|---|---|---|---|---|
| 1 | 128 | 64 | 1.779947 | $x$ | 0.002366 |
| | | 128 | 0.912821 | $1.94 \times x$ | 0.002366 |
| | | 256 | 0.461349 | $3.86 \times x$ | 0.002366 |
| | 256 | 64 | 14.032499 | $x$ | 0.000588 |
| | | 128 | 7.184613 | $1.95 \times x$ | 0.000588 |
| | | 256 | 3.586753 | $3.91 \times x$ | 0.000588 |
| | 512 | 64 | 117.726938 | $x$ | 0.000146 |
| | | 128 | 56.656205 | $2.07 \times x$ | 0.000146 |
| | | 256 | 28.149711 | $4.18 \times x$ | 0.000146 |
| $\pi$ | 128 | 64 | 2.084217 | $x$ | 0.022982 |
| | | 128 | 1.042503 | $1.99 \times x$ | 0.022982 |
| | | 256 | 0.532034 | $3.92 \times x$ | 0.022982 |
| | 256 | 64 | 15.645009 | $x$ | 0.005780 |
| | | 128 | 7.955400 | $1.97 \times x$ | 0.005780 |
| | | 256 | 3.994738 | $3.91 \times x$ | 0.005780 |
| | 512 | 64 | 135.844675 | $x$ | 0.001446 |
| | | 128 | 63.529322 | $2.14 \times x$ | 0.001446 |
| | | 256 | 31.461323 | $4.32 \times x$ | 0.001446 |

Таблица 1: Результаты расчетов на Blue Gene с использованием MPI.

Проверим выводы из вычислений на BlueGene на Polus. Результаты расчетов на Polus приведены в таблице (2).

Вычисления гибридной версии проводились при $L = 1$ и на числе нитей 2 и 4. Результаты работы гибридной программы MPI/OpenMP приведены в таблице (3).

| L | Число точек сетки $N^3$ | Число процессоров | Время решения | Ускорение | Погрешность |
|---|---|---|---|---|---|
| 1 | 128 | 10 | 3.567562 | $x$ | 0.002366 |
| | | 20 | 1.207293 | $2.95 \times x$ | 0.002366 |
| | | 40 | 0.870292 | $4.01 \times x$ | 0.002366 |
| | 256 | 10 | 28.875149 | $x$ | 0.000588 |
| | | 20 | 16.889358 | $1.71 \times x$ | 0.000588 |
| | | 40 | 7.961963 | $3.62 \times x$ | 0.000588 |
| | 512 | 10 | 229.865662 | $x$ | 0.000146 |
| | | 20 | 123.315947 | $1.86 \times x$ | 0.000146 |
| | | 40 | 65.362031 | $3.52 \times x$ | 0.000146 |
| $\pi$ | 128 | 10 | 3.583759 | $x$ | 0.022982 |
| | | 20 | 2.047095 | $1.75 \times x$ | 0.022982 |
| | | 40 | 1.088501 | $3.29 \times x$ | 0.022982 |
| | 256 | 10 | 27.105510 | $x$ | 0.005780 |
| | | 20 | 14.887006 | $1.82 \times x$ | 0.005780 |
| | | 40 | 7.957649 | $3.40 \times x$ | 0.005780 |
| | 512 | 10 | 254.270676 | $x$ | 0.001446 |
| | | 20 | 126.328977 | $2.01 \times x$ | 0.001446 |
| | | 40 | 66.345348 | $3.83 \times x$ | 0.001446 |

Таблица 2: Результаты расчетов на Polus.

| Число нитей | Число точек сетки $N^3$ | Число процессоров | Время решения | Ускорение | Отношение MPI к MPI/OpenMP |
|---|---|---|---|---|---|
| 2 | 128 | 64 | 0.940043 | $x$ | 1.892966 |
| | | 128 | 0.507191 | $1,853 \times x$ | 1.799757 |
| | | 256 | 0.263086 | $3,573 \times x$ | 1.753605 |
| | 256 | 64 | 7.381695 | $x$ | 1.90098 |
| | | 128 | 3.804669 | $1,94 \times x$ | 1.888367 |
| | | 256 | 1.970210 | $3,746 \times x$ | 1.820492 |
| | 512 | 64 | 58.995760 | $x$ | 1.995505 |
| | | 128 | 29.378861 | $2,01 \times x$ | 1.928468 |
| | | 256 | 14.595683 | $4,04 \times x$ | 1.928632 |
| 4 | 128 | 64 | 0.499255 | $x$ | 3.565206 |
| | | 128 | 0.284122 | $1,757 \times x$ | 3.212778 |
| | | 256 | 0.158459 | $3,15 \times x$ | 2.911472 |
| | 256 | 64 | 3.821808 | $x$ | 3.672390 |
| | | 128 | 1.990212 | $1,920 \times x$ | 3.609973 |
| | | 256 | 1.010719 | $3,781 \times x$ | 3.548714 |
| | 512 | 64 | 29.577895 | $x$ | 3.980233 |
| | | 128 | 14.733953 | $2,01 \times x$ | 3.845282 |
| | | 256 | 7.367155 | $4,01 \times x$ | 3.820974 |

Таблица 3: Результаты расчетов на Blue Gene с использованием MPI/OpenMP.

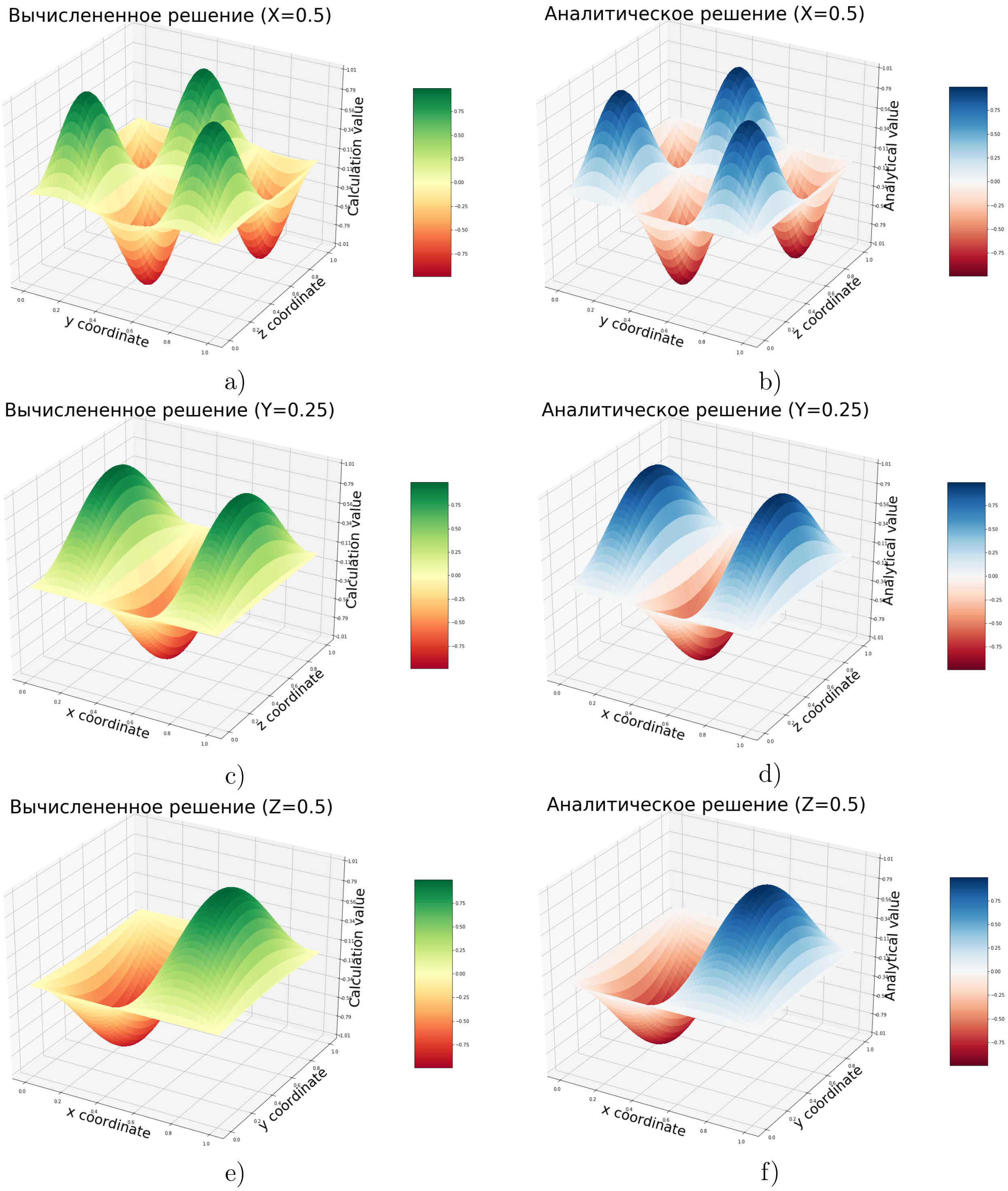


Рис. 1: Сравнение аналитического и вычисленного решений на сетке $N = 128$. По горизонтали отражены графики срезов по одной переменной, по вертикали графики вычисленного решения (a, c, e) и аналитического решения (b, d, f).

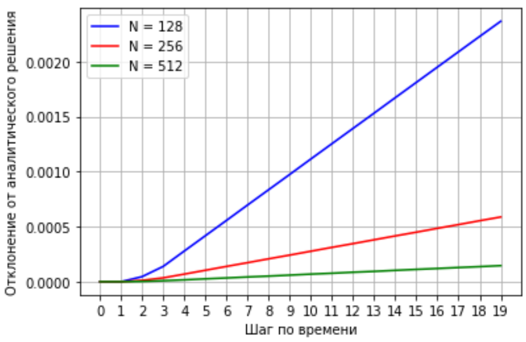


Рис. 2: Отклонение от аналитического решения при $L = 1$

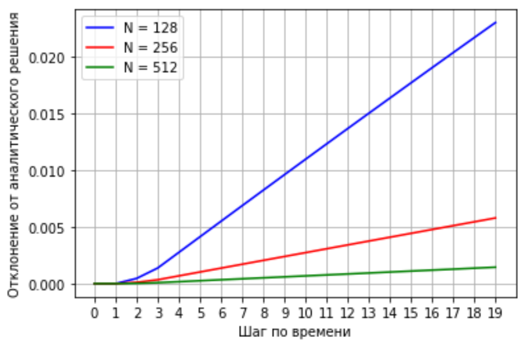


Рис. 3: Отклонение от аналитического решения при $L = \pi$