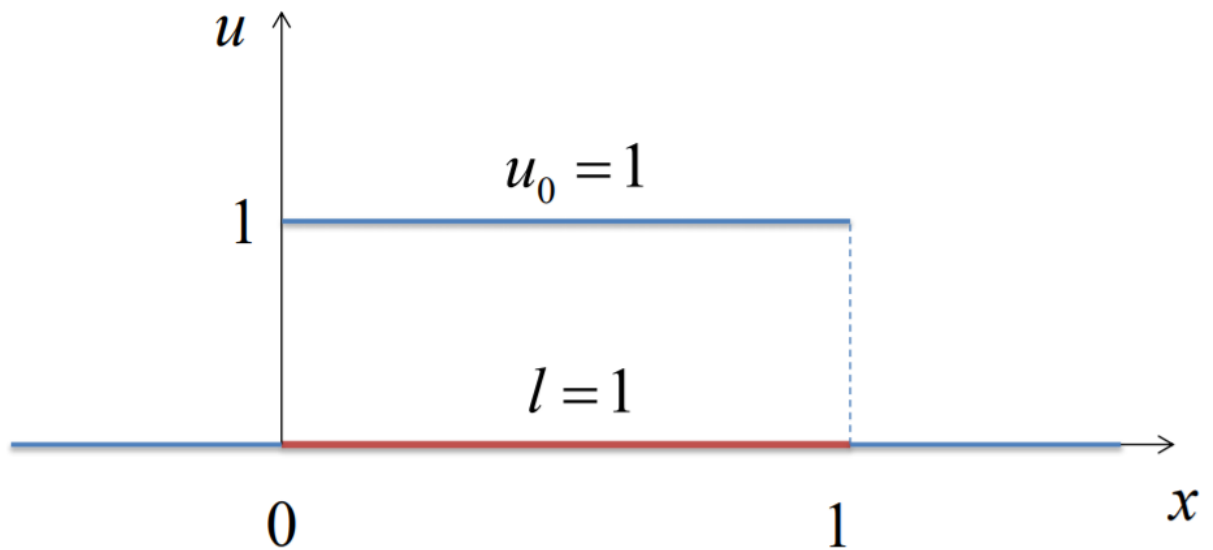


# Домашнее задание №3 (MPI)

## Распараллеливание уравнения теплопроводности

### Постановка задачи

Решить одномерное однородное уравнение теплопроводности с использованием средств распараллеливания MPI.



Стержень длиной  $l = 1$  в момент времени  $t_0 = 0$  имеет температуру

$u_0 = 1$ . Температура окружающей среды поддерживается равной 0.

Начальное условие:  $u(x \in l, 0) = u_0$ .

Граничное условие:  $u(x \notin l, t) = 0$ .

Задание начального распределения температуры и вывод итогового распределения происходит на root процессе.

Необходимо решить одномерное однородное уравнение теплопроводности, которое представлено в виде конечно-разностной схемы:

$$u_i^{n+1} = u_i^n + \frac{k\tau}{h^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$
$$\frac{k\tau}{h^2} < 1 \Rightarrow \tau < \frac{h^2}{k}$$

## Задания

### 1) (2 балла) Проверка правильной работы алгоритма:

Реализовать параллельную версию алгоритма и получить распределение температуры вдоль стержня на момент времени

$T=0.1$ ,

используя следующие параметры:

$k=1$  (коэффициент теплопроводности),

$h = 0.02$  (шаг по пространству);

$\tau = 0.0002$  (шаг по времени).

Вычислить значения температуры в 11-ти точках длиной 0.1 включая начало (0, 0.1, 0.2, ..., 1) и сравнить полученные значения с точным решением:

$$u(x,t) = \frac{4u_0}{\pi} \sum_{m=0}^{\infty} \frac{\exp\left(-k\pi^2(2m+1)^2 \frac{t}{l^2}\right)}{2m+1} \sin\left(\frac{\pi(2m+1)x}{l}\right)$$

### 2) (2 балла) Масштабируемость:

Сравнить время работы программы для

разного кол-ва процессов (1,2,4,8,16,24) и

разного кол-ва точек ( $N = 10000, 25000, 50000$ ),

конечное время можно уменьшить до  $10^{-4}$  для уменьшения времени работы, условие для величины шага ( $\tau$ ) приведены выше).

Построить 3 графика соответственно кол-ву точек  $N$ .

### 3) (2 балла) Организовать асинхронную передачу данных и повторить задание пункта 2.

### 4) (2 балла) использовать коллективные операции для рассылки начального распределения температур и сбора итоговых результатов

### 5) (1 балл)\* Реализовать гибридную задачу с двумя MPI-процессами, каждый из которых считает свой кусок на GPU.

Цель этой задачи не ускорить программу, а научиться компилировать и писать гибридные программы.

### 6) (1 балл)\* Реализовать решение двумерной задачи теплопроводности с граничными условиями Дирихле, используя встроенную в MKL библиотеку.

## Примечание:

$$u_i^{n+1} = u_i^n + \frac{k\tau}{h^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

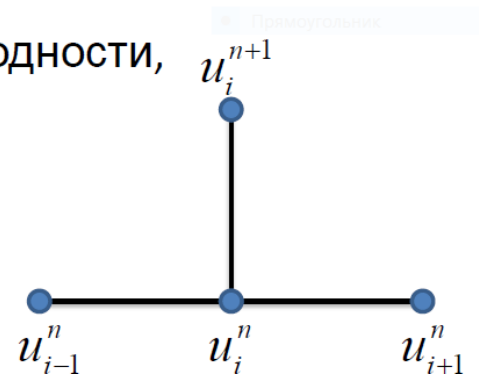
$$\frac{k\tau}{h^2} < 1 \Rightarrow \tau < \frac{h^2}{k}$$

где:

$k$  - коэффициент температуропроводности,

$\tau$  - шаг по времени,

$h$  - шаг по пространству



Задача состоит в нахождении распределения температур в момент времени  $n+1$ .

Таким образом нижний индекс – это индекс по пространству, а верхний индекс – это индекс по времени.

### Как работает формула?

Пусть у нас есть начальный момент времени  $n = 0$ , в котором у нас задано распределение температуры:

$$u_{i \in l}^0 = 1, u_{i \notin l}^0 = 0$$

Мы хотим найти распределение в следующий момент времени, то есть в момент времени  $n+1 = \tau$ :

$$u_{i \in l}^1$$

Рассмотрим самую первую точку  $i = 0$ :

$$u_0^1 = u_0^0 + \frac{k\tau}{h^2} (u_1^0 - 2u_0^0 + u_{-1}^0)$$

$$u_0^1 = 1 + 0.5(1 - 2 + 0)$$

$$u_0^1 = 0.5$$

Таким образом мы получили значение в следующий момент времени для точки  $i = 0$ .

### Как это работает параллельно?

Разумно будет поделить весь отрезок на `NumberOfProc` частей, чтобы каждый процессор считал свою часть ( $N / \text{NumberOfProc}$ ).

При нахождении на границе между процессами воспользоваться передачей граничных значений “соседнему” процессору.

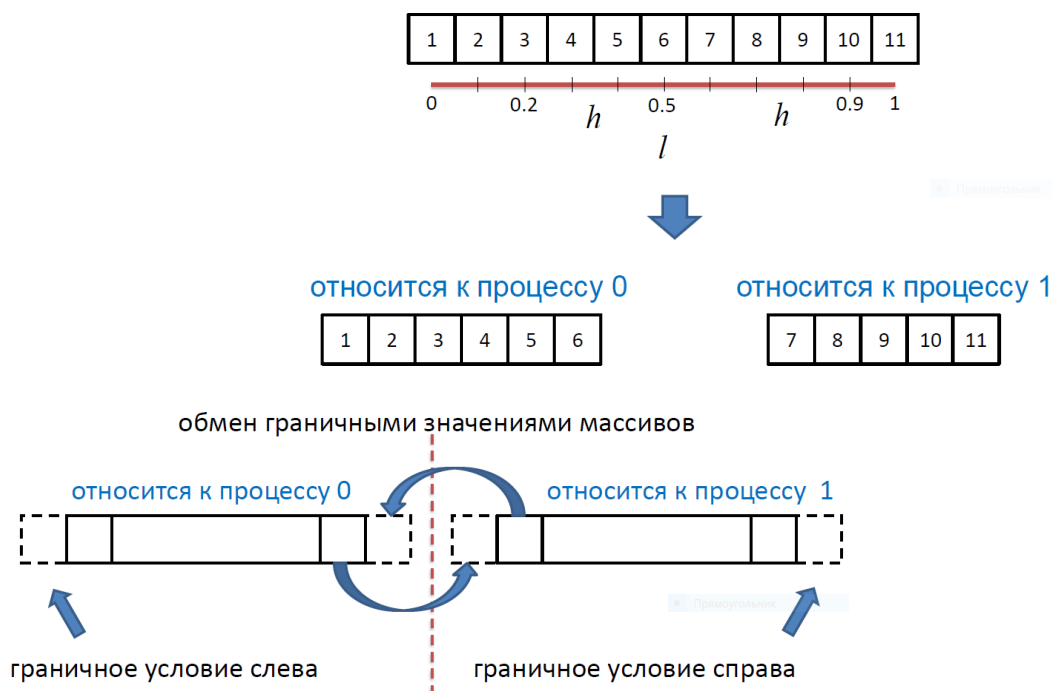
### Пример

Допустим у нас есть 2 процесса, и мы разделили весь стержень на 11 точек.

Нулевой процесс будет обрабатывать 6 точек, а первый – 5 точек.

При расчёте точек на границах нужно воспользоваться передачей с предыдущего или следующего процесса, чтобы получить значение в предыдущей или следующей точке соответственно.

В данном примере нужно было бы при расчёте нулевым процессом в следующий момент времени в точке  $i = 6$  передавать с первого процесса значение в точке  $i = 7$ .



Сдача будет происходить на семинарах.

Перед сдачей необходимо загрузить файлы в github. Среди файлов должен быть Makefile для сборки и запуска, а также README.txt с кратким описанием структуры и запусков.

### Дедлайн 08.12.2024

Защита только по тому, что было отправлено к дедлайну. Без возможности исправления. До дедлайна можно сдавать задание с возможностью доработки.

По результатам собеседования, семинарист определит ваше понимание.

**!После дедлайна ставится оценка 0.**