

Семинар 1. Работа в ОС Windows и UNIX. Измерение производительности компьютера.

1. Основные команды для работы в ОС Windows и ОС Unix.

- 1.1. Вход, выход, работа с файлами и программами.
- 1.2. Разработка и исполнение программ.

2. Функции измерения времени.

Функция clock

- Библиотечная функция определена в заголовочном файле time.h;
- Прототип clock_t clock();
- Возвращает время, прошедшее с момента запуска программы в единицах 1/CLK_TCK секунды;
- Используется в Windows и Linux;

Пример использования - ex01a.c

Достоинства: высокая платформенная независимость.

Недостатки: низкая точность, при высокой загрузке процессора – неприемлемая точность, так как измеряется интервал времени, во время которого помимо процесса исследуемой программы исполнялись и другие процессы

Команда RDTSC

- Платформенно-зависимый вариант для x86
- Возвращает число тактов с момента запуска процессора;
- Используется в Windows и UNIX для процессоров Intel;

Пример использования - ex01b.c

Достоинства: максимально возможная точность.

Недостатки: зависимость от архитектуры процессора, ухудшение точности при высокой загрузке процессора.

Функция gettimeofday

- Библиотечная функция определена в заголовочном файле sys/time.h;
- Прототип int gettimeofday(struct timeval* tv, struct timezone* tz);
- Время можно вычислить из структуры timeval
- Используется в UNIX

Пример использования -ex01c.c

Достоинства: высокая платформенная независимость.

Недостатки: низкая точность, при высокой загрузке процессора – неприемлемая точность, так как измеряется интервал времени, во время которого помимо процесса исследуемой программы исполнялись и другие процессы

Функция times

- Библиотечная функция определена в заголовочном файле sys/times.h
- Прототип clock_t times(struct tms *buf);
- Возвращает время, прошедшее с момента запуска программы в единицах 1/CLK_TCK секунды
- Используется в UNIX

Пример использования - ex01d.c

Достоинства: высокая точность (относительная независимость от других процессов системы)

Недостатки: для малых интервалов она зависит от интервала времени прерываний по таймеру.

Функция QueryPerformanceCounter

- Библиотечная функция MS Windows определена в заголовочном файле windows.h
- Прототип bool WINAPI QueryPerformanceCounter(long long int *ticks);
- Возвращает время, прошедшее с момента запуска программы в секундах
- Используется только в MS Windows

Пример использования:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <windows.h>
int main (int argc, char *argv[])
{
    long long int t1, t2;
    long long int i, n = 1024*1024*1024;
    double s=0, t=0, h = 1.0/n;
    QueryPerformanceCounter(&t1);
    for (i=0; i<n; i++) {
        s = s + sin(h*i);
    }
    QueryPerformanceCounter(&t2);
    t = 1e-6 * (t2 - t1);
    printf("s=%le t=%lf\n", s, t);
    return 0;
}
```

Трансляция и результаты выполнения:

```
>gcc -o ex01a.px -O2 -lm ex01a.c
>gcc -o ex01b.px -O2 -lm ex01b.c
>gcc -o ex01c.px -O2 -lm ex01c.c
>gcc -o ex01d.px -O2 -lm ex01d.c
>ex01a.px
Time: 1.760000 sec Pi = 3.141592653590
>ex01b.px
Time: 1.879260 sec Pi = 3.141592653590
>ex01c.px
Time: 1.758059 sec Pi = 3.141592653590
>ex01d.px
Time: 1.750000 sec Pi = 3.141592653590
Time: 1.730000 sec Pi = 3.141592653590
```

3. Измерение времени операций.

3.1. Сложение, умножение, деление.

3.2. Функции exp, sin и др.

Пример 1 (ex02a.c). Косвенное определение скорости выполнения операций.

Трансляция и результаты выполнения:

```
>gcc -o ex02a.px -O2 -lm ex02a.c msum.c
>ex02a.px
Time: 17.379798 sec sum = 3.141592651592
Time: 17.322942 sec sum = 2.418399150980 diff: -0.056856
Time: 17.323221 sec sum = 2.418399151314 diff: -0.056577
Time: 34.646773 sec sum = 3.627598725473 diff: 17.266975
```

Пример 2 (ex02b.c). Косвенное определение производительности ПК.

Трансляция и результаты выполнения:

```
>gcc -o ex02b.px -O2 -lm ex02b.c msum.c
>ex02b.px
Time: 17.381489 34.667101 sec Div. perf.: 5.785158e-02 GFlops
Time: 17.334578 17.807337 sec Mult. perf.: 8.460971e+00 Gflops
```

Задание 1 (ex02c.c): Измерить производительность компьютера на операциях других типов и функциях (сложение, вычитание, возведение в степень, exp, log, sin).

Семинар 2. Создание локальных процессов и простейший обмен данными между ними.

1. Создание локальных процессов.

- 1) Создание “тяжелых” локальных последовательных процессов:
 - а) Запуск пользователем нескольких программ (main1.px, main2.px, ...);
 - б) Запуск из головной программы нескольких процессов (fork, execl, execv, system, wait).
- 2) Создание “легких” локальных последовательных процессов – ветвление процесса (pthread_create, pthread_join).
- 3) Гибридный способ – создание процессов обоего типа (server, client, запускающие треды).

2. Простой обмен данными между локальными процессами.

- 1) обмен сигналами (функции kill – посылка сигнала процессу, signal – вызов функции пользователя при получении сигнала с заданным номером);
- 2) обмен с помощью разделяемых файлов (функция блокировки записи lockf());
- 3) обмен с помощью каналов (функции pipe, read, write; два файловых дескриптора: для чтения и для записи);
- 4) обмен с помощью очередей сообщений (функции msgget, msgsnd, msgrcv, msgctl – порции данных одинаковой длины);

3. Примеры организации локальных параллельных процессов.

Пример 1 (ex03a.c). Простейшее разветвление процесса.

Трансляция и результат выполнения:

```
>gcc -o ex03a.px -O2 -lm ex03a.c
>ex03a.px
pid=10450 -> Hello!
pid=10451 -> I am slave, mp=1
pid=10451 -> i=0 pids=10450
pid=10451 -> i=1 pids=0
pid=10451 -> i=2 pids=0
pid=10451 -> i=3 pids=0
pid=10451 -> i=4 pids=0
pid=10452 -> I am slave, mp=2
pid=10452 -> i=0 pids=10450
pid=10452 -> i=1 pids=10451
pid=10452 -> i=2 pids=0
pid=10452 -> i=3 pids=0
pid=10452 -> i=4 pids=0
pid=10453 -> I am slave, mp=3
pid=10453 -> i=0 pids=10450
pid=10453 -> i=1 pids=10451
pid=10453 -> i=2 pids=10452
pid=10453 -> i=3 pids=0
pid=10453 -> i=4 pids=0
pid=10454 -> I am slave, mp=4
pid=10454 -> i=0 pids=10450
pid=10454 -> i=1 pids=10451
pid=10454 -> i=2 pids=10452
pid=10454 -> i=3 pids=10453
pid=10454 -> i=4 pids=0
pid=10450 -> I am master, mp=0
pid=10450 -> i=0 pids=10450
pid=10450 -> i=1 pids=10451
pid=10450 -> i=2 pids=10452
pid=10450 -> i=3 pids=10453
pid=10450 -> i=4 pids=10454
```

Пример 2 (ex03b.c). Разветвление процесса с обменом данными с помощью очереди сообщений.

Трансляция и результаты выполнения:

```
>gcc -o ex03b.px -O2 -lm ex03b.c
>ex03b.px
pid=10740 -> Hello!
pid=10744 -> I am slave, mp=4
pid=10744 -> i=0 pids=10740
pid=10744 -> i=1 pids=10741
pid=10744 -> i=2 pids=10742
pid=10744 -> i=3 pids=10743
pid=10744 -> i=4 pids=10744
pid=10742 -> I am slave, mp=2
pid=10742 -> i=0 pids=10740
pid=10742 -> i=1 pids=10741
pid=10742 -> i=2 pids=10742
pid=10742 -> i=3 pids=10743
pid=10742 -> i=4 pids=10744
pid=10741 -> I am slave, mp=1
```

```

pid=10741 -> i=0 pids=10740
pid=10741 -> i=1 pids=10741
pid=10741 -> i=2 pids=10742
pid=10741 -> i=3 pids=10743
pid=10741 -> i=4 pids=10744
pid=10743 -> I am slave, mp=3
pid=10743 -> i=0 pids=10740
pid=10743 -> i=1 pids=10741
pid=10743 -> i=2 pids=10742
pid=10743 -> i=3 pids=10743
pid=10743 -> i=4 pids=10744
pid=10740 -> I am master, mp=0
pid=10740 -> i=0 pids=10740
pid=10740 -> i=1 pids=10741
pid=10740 -> i=2 pids=10742
pid=10740 -> i=3 pids=10743
pid=10740 -> i=4 pids=10744

```

Пример 3 (ex04a.c). Численное интегрирование, использование тяжелых процессов.

Пример 4 (ex04b.c). Численное интегрирование, использование легких процессов.

Формула средних прямоугольников:

$$I = \int_a^b f(x)dx = \sum_{k=0}^{p-1} \int_{a_k}^{b_k} f(x)dx \approx \sum_{k=0}^p S_k = \sum_{k=0}^p \sum_{i=1}^{N_k} f(x_{ik})h_k,$$

$$a_k = a + kh, \quad b_k = a_k + h, \quad h = \frac{b-a}{p},$$

$$x_{ik} = a_k + (i-0.5)h_k, \quad h_k = \frac{(b_k - a_k)}{N_k}, \quad N_k = \frac{N}{p}.$$

Трансляция и результаты выполнения:

```
>gcc -o ex04a.px -O2 -lm ex04a.c mycom.c
```

```
>gcc -o ex04b.px -O2 -lm -pthread ex04b.c mycom.c
```

```
>ex04a.px 1
```

```
time=18.469721 sum= 3.141592651589794e+00
```

```
>ex04a.px 2
```

```
mp=1 a1=5.000000e-01 b1=1.000000e+00 n1=500000000 s1=1.287002e+00
```

```
mp=0 a1=0.000000e+00 b1=5.000000e-01 n1=500000000 s1=1.854590e+00
```

```
time=8.688225 sum= 3.141592648389793e+00
```

```
>ex04b.px 1
```

```
time=18.295300 sum= 3.141592651589794e+00
```

```
>ex04b.px 2
```

```
mt=0 a1=0.000000e+00 b1=5.000000e-01 n1=500000000 s1=1.854590e+00
```

```
mt=1 a1=5.000000e-01 b1=1.000000e+00 n1=500000000 s1=1.287002e+00
```

```
time=8.689911 sum= 3.141592648389793e+00
```

Задание 2 (ex04c.c): На базе примеров ex04a.c, ex04b.c написать программу, решающую задачу с помощью тяжелых и легких процессов одновременно.

Семинар 3. Использование семафоров и общих сегментов памяти.

1. Другие способы обмена между локальными процессами.

- 1) разделяемые сегменты общей памяти (функции shmget - создание на сервере, shmat - присоединение, shmdt - отсоединение, shmctl - удаление);
- 2) семафоры как механизм доступа к общим переменным (функции semget, semop, semctl);
- 3) легкие семафоры – mutex.
- 4) передача большого числа параметров в треды.

Пример 1 (ex05a.c). Вычисление однократного интеграла, тяжелые процессы.

Пример 2 (ex05b.c). Вычисление однократного интеграла, легкие процессы.

Трансляция и результаты выполнения:

```
>gcc -o ex05a.px -O2 -lm ex05a.c mусom.c
>gcc -o ex05b.px -O2 -lm ex05b.c mусom.c
>ex05a.px 1
time=17.390892 sum= 3.141592651591870e+00
>ex05a.px 2
mp=1 a1=5.000000e-01 b1=1.000000e+00 n1=500000000 s1=1.287002e+00
mp=0 a1=0.000000e+00 b1=5.000000e-01 n1=500000000 s1=1.854590e+00
time=8.698782 sum= 3.141592648390306e+00
>ex05b.px 1
time=17.350918 sum= 3.141592651591870e+00
>ex05b.px 2
mt=1 a1=5.000000e-01 b1=1.000000e+00 n1=500000000 s1=1.287002e+00
mt=0 a1=0.000000e+00 b1=5.000000e-01 n1=500000000 s1=1.854590e+00
time=8.804832 sum= 3.141592648390306e+00
```

Задание 3 (ex05c.c): На базе кодов ex05a.c, ex05b.c написать программу, решающую задачу с помощью тяжелых и легких процессов одновременно, сделать одну расчетную функцию myjob. Количество тяжелых и легких процессов задаем через параметры командной строки.

Семинар 4. Удаленные процессы и обмен данными между ними. Стандарт MPI. Базовые функции.

1. Создание удаленных параллельных процессов и обмен данными между ними.

Создание удаленных процессов:

- 1) запуск программ на различных машинах (telnet, rlogin, rsh, ssh);
- 2) использование интерфейсов удаленного запуска: PVM, MPI.

Обмен данными:

- 1) сокеты TCP/IP для межмашинной передачи данных;
- 2) библиотеки межмашинных коммуникаций: PVM, MPI.

2. Стандарт MPI. Базовые функции.

- 1) Конфигурирование сервера (сервер и клиенты, доверенные хосты);
- 2) Инсталляция MPI на сервере (www.mpi-forum.org);
- 3) Разработка MPI-приложений;
- 4) Запуск MPI-приложений (mpirun и др.);

3. Базовые функции MPI.

- 1) Инициализация и параметры среды (MPI_Init, MPI_Initialized, MPI_Comm_size, MPI_Comm_rank, MPI_Get_processor_name, MPI_Finalize, MPI_Abort, MPI_Wtick, MPI_Wtime);
- 2) Синхронные обмены типа точка-точка (синхронные и асинхронные) (MPI_Send, MPI_Recv, MPI_Sendrecv);
- 3) Процедуры синхронизации (MPI_Barrier).

Пример 1 (ex06a.c). Простейшая программа на MPI.

Трансляция и результаты выполнения:

```
>mpicc -o ex06a.px -O2 -lm ex06a.c
>mpirun -np 1 -nolocal -machinefile hosts ex06a.px
Before MPI_Init ii=0
Between MPI_Init & MPI_Finalize ii=1
Netsize: 1, process: 0, system: cl73.limm, tick=1.000000e-06
mp=0, time=2.460000e-04 res=1.231108e+01
After MPI_Finalize ii=1
>mpirun -np 2 -nolocal -machinefile hosts ex06a.px
Before MPI_Init ii=0
Before MPI_Init ii=0
Between MPI_Init & MPI_Finalize ii=1
Netsize: 4, process: 0, system: cl73.limm, tick=1.000000e-06
Between MPI_Init & MPI_Finalize ii=1
Netsize: 4, process: 1, system: cl74.limm, tick=1.000000e-06
mp=0, time=2.450000e-04 res=1.231108e+01
mp=1, time=2.610000e-04 res=1.231108e+01
After MPI_Finalize ii=1
After MPI_Finalize ii=1
```

Пример 2 (ex06b.c). Двухнаправленный синхронный обмен с соседями в линейной топологии.

Трансляция и результаты выполнения:

```
>mpicc -o ex06b.px -O2 -lm ex06b.c
>mpirun -np 1 -nolocal -machinefile hosts ex06b.px
Netsize: 1, process: 0, system: cl73.limm, tick=1.000000e-06
mp=0 a1=-1.000000e+00 a2=0.000000e+00 a3=-1.000000e+00
>mpirun -np 2 -nolocal -machinefile hosts ex06b.px
Netsize: 2, process: 0, system: cl73.limm, tick=1.000000e-06
Netsize: 2, process: 1, system: cl74.limm, tick=1.000000e-06
00 <-> 01
01 <-> 00
mp=0 a1=-1.000000e+00 a2=0.000000e+00 a3=3.140000e+00
mp=1 a1=0.000000e+00 a2=3.140000e+00 a3=-1.000000e+00
>mpirun -np 3 -nolocal -machinefile hosts ex06b.px
Netsize: 3, process: 0, system: cl73.limm, tick=1.000000e-06
Netsize: 3, process: 1, system: cl74.limm, tick=1.000000e-06
Netsize: 3, process: 2, system: cl78.limm, tick=1.000000e-06
00 <-> 01
01 <-> 00
01 <-> 02
02 <-> 01
mp=0 a1=-1.000000e+00 a2=0.000000e+00 a3=3.140000e+00
mp=1 a1=0.000000e+00 a2=3.140000e+00 a3=6.280000e+00
mp=2 a1=3.140000e+00 a2=6.280000e+00 a3=-1.000000e+00
>mpirun -np 4 -nolocal -machinefile hosts ex06b.px
00 <-> 01
01 <-> 00
02 <-> 03
01 <-> 02
03 <-> 02
```

```

02 <-> 01
mp=2 a1=3.140000e+00 a2=6.280000e+00 a3=9.420000e+00
mp=0 a1=-1.000000e+00 a2=0.000000e+00 a3=3.140000e+00
mp=1 a1=0.000000e+00 a2=3.140000e+00 a3=6.280000e+00
mp=3 a1=6.280000e+00 a2=9.420000e+00 a3=-1.000000e+00

```

Пример 3 (ex06c.c). Однонаправленный синхронный обмен в топологии кольцо.

Трансляция и результаты выполнения:

```

>mpicc -o ex06c.px -O2 -lm ex06c.c
>mpirun -np 1 -nolocal -machinefile hosts ex06c.px
Too small network
>mpirun -np 2 -nolocal -machinefile hosts ex06c.px
mp=0 time=3.210000e-04 res=2.000000e+00
mp=1 time=2.170000e-04 res=2.000000e+00
>mpirun -np 3 -nolocal -machinefile hosts ex06c.px
mp=2 time=1.213900e-02 res=3.000000e+00
mp=1 time=5.149000e-03 res=2.000000e+00
mp=0 time=4.609000e-03 res=3.000000e+00
>mpirun -np 4 -nolocal -machinefile hosts ex06c.px
mp=1 time=9.090000e-04 res=2.000000e+00
mp=2 time=9.127000e-03 res=3.000000e+00
mp=0 time=7.405000e-03 res=4.000000e+00
mp=3 time=6.542000e-03 res=4.000000e+00

```

Пример 4 (ex07a.c). Вычисление интеграла, сборка суммы на нулевом процессоре.

Трансляция и результаты выполнения:

```

>mpicc -o ex07a.px -O2 -lm ex07a.c mycom.c mynet.c
>mpirun -np 1 -nolocal -machinefile hosts ex07a.px
mp=0 t1=17.624254 t2=0.000000 t3=17.624254 int= 3.141592651591870e+00
>mpirun -np 2 -nolocal -machinefile hosts ex07a.px
mp=0 t1=8.713395 t2=5.357119 t3=14.070514 int= 3.141592648390306e+00
mp=1 t1=14.068855 t2=0.000404 t3=14.069259 int= 1.287002215586667e+00
>mpirun -np 4 -nolocal -machinefile hosts ex07a.px
mp=0 t1=4.358262 t2=0.028721 t3=4.386983 int= 3.141592642065422e+00
mp=1 t1=4.366067 t2=0.020719 t3=4.386786 int= 8.746757802958527e-01
mp=2 t1=4.360264 t2=0.025230 t3=4.385494 int= 7.194139966100185e-01
mp=3 t1=4.384680 t2=0.000514 t3=4.385194 int= 5.675882164165912e-01
>mpirun -np 8 -nolocal -machinefile hosts ex07a.px
mp=0 t1=2.179436 t2=1.423302 t3=3.602738 int= 3.141592629478097e+00
mp=2 t1=2.194378 t2=1.408253 t3=3.602631 int= 4.551680250680343e-01
mp=4 t1=3.600436 t2=0.000350 t3=3.600786 int= 3.798068224947328e-01
mp=3 t1=2.192403 t2=1.410172 t3=3.602575 int= 4.195077517209749e-01
mp=6 t1=2.190496 t2=1.410177 t3=3.600673 int= 3.013155610478502e-01
mp=7 t1=2.200823 t2=1.399654 t3=3.600477 int= 2.662726531032736e-01
mp=1 t1=2.182648 t2=1.424412 t3=3.607060 int= 4.824946705556977e-01
mp=5 t1=2.187454 t2=1.418299 t3=3.605753 int= 3.396071712388637e-01

```

Задание 4 (ex07b.c): реализовать суммирование частичных сумм методом сдвигания, добавить трэды.

Семинар 5. Библиотека MPI. Асинхронные обмены. Коллективные операции.

1. Библиотека MPI. Дополнительные функции.

- 1) Асинхронные обмены (MPI_Isend, MPI_Irecv, MPI_Wait, MPI_Waitall);
- 2) Коллективные обмены и операции (MPI_Bcast, MPI_Reduce, MPI_Allreduce).

Пример 1 (ex08a.c). Вычисление двойного интеграла в прямоугольнике $[a,b] \times [c,d]$.

Формула средних прямоугольников. Результат не зависит от числа процессоров. Топология – решетка.

$$I = \int_{y_a}^{y_b} \int_{x_a}^{x_b} f(x, y) dx dy \approx \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} f(x_i, y_j) h_x h_y = \sum_{k_2=0}^{p_2-1} \sum_{k_1=0}^{p_1-1} \sum_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} f(x_i, y_j) h_x h_y,$$

$$x_i = x_a + (i - 0.5)h_x, \quad y_j = y_a + (j - 0.5)h_y, \quad h_x = \frac{x_b - x_a}{N_x}, \quad h_y = \frac{y_b - y_a}{N_y}.$$

Трансляция и результаты выполнения:

```
>mpicc -o ex08a.px -O2 -lm ex08a.c mynet.c
>mpirun -np 1 -nolocal -machinefile hosts ex08a.px 1 1 0
mp=0 grid=1x1 coord=(0,0)
mp=0 i1=0 i2=9999 j1=0 j2=9999
mp=0 t1=1.811444e+01 t2=0.000000e+00 t3=1.811444e+01 int=6.382290e-01
>mpirun -np 2 -nolocal -machinefile hosts ex08a.px 2 1 0
mp=0 grid=2x1 coord=(0,0)
mp=1 grid=2x1 coord=(1,0)
mp=0 i1=0 i2=4999 j1=0 j2=9999
mp=1 i1=5000 i2=9999 j1=0 j2=9999
mp=0 t1=9.298582e+00 t2=4.600000e-05 t3=9.298628e+00 int=6.382290e-01
mp=1 t1=8.900826e+00 t2=5.300000e-05 t3=8.900879e+00 int=3.552322e-01
>mpirun -np 4 -nolocal -machinefile hosts ex08a.px 2 2 0
mp=0 grid=2x2 coord=(0,0)
mp=1 grid=2x2 coord=(1,0)
mp=2 grid=2x2 coord=(0,1)
mp=3 grid=2x2 coord=(1,1)
mp=0 i1=0 i2=4999 j1=0 j2=4999
mp=1 i1=5000 i2=9999 j1=0 j2=4999
mp=2 i1=0 i2=4999 j1=5000 j2=9999
mp=3 i1=5000 i2=9999 j1=5000 j2=9999
mp=3 t1=4.388040e+00 t2=5.284000e-03 t3=4.393324e+00 int=2.717513e-01
mp=1 t1=4.502950e+00 t2=4.100000e-05 t3=4.502991e+00 int=8.348090e-02
mp=2 t1=4.659110e+00 t2=5.200000e-05 t3=4.659162e+00 int=2.164915e-01
mp=0 t1=4.698314e+00 t2=5.500000e-05 t3=4.698369e+00 int=6.382290e-01
```

Пример 2 (ex08b.c). Вычисление двойного интеграла, ввод из файла, вывод в файл, асинхронные пересылки.

Автоматическая генерация решетки процессоров.

Трансляция и результаты выполнения:

```
>mpicc -o ex08b.px -O2 -lm ex08b.c mycom.c mynet.c
>mpirun -np 1 -nolocal -machinefile hosts ex08b.px
Grid=1x1
t1=8.531479e+00 t2=0.000000e+00 t3=8.531479e+00 int=2.635526e+02
>mpirun -np 12 -nolocal -machinefile hosts ex08b.px
Grid=3x4
t1=7.514590e-01 t2=3.230000e-04 t3=7.517820e-01 int=2.635526e+02
```

Задание 5 (ex08c.c): посчитать трехмерный интеграл, используя пример ex08b.c. Топология – трехмерная решетка, которая вычисляется автоматически. Добавить треды.

$$I = \int_{z_a}^{z_b} \int_{y_a}^{y_b} \int_{x_a}^{x_b} f(x, y, z) dx dy dz \approx \sum_{k=1}^{N_z} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} f(x_i, y_j, z_k) h_x h_y h_z = \sum_{l_3=0}^{p_3-1} \sum_{l_2=0}^{p_2-1} \sum_{l_1=0}^{p_1-1} \sum_{k=k_1}^{k_2} \sum_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} f(x_i, y_j, z_k) h_x h_y h_z,$$

$$x_i = x_a + (i - 0.5)h_x, \quad y_j = y_a + (j - 0.5)h_y, \quad z_k = z_a + (k - 0.5)h_z,$$

$$h_x = \frac{x_b - x_a}{N_x}, \quad h_y = \frac{y_b - y_a}{N_y}, \quad h_z = \frac{z_b - z_a}{N_z}.$$

Правило построения решетки процессоров:

$$\frac{p_1}{p_2} \approx \frac{N_x}{N_y}, \quad \frac{p_1}{p_3} \approx \frac{N_x}{N_z}, \quad p_1 p_2 p_3 = N_x N_y N_z.$$

Семинар 6. Дополнительные средства MPI. Параллельная сортировка.

1. Дополнительные средства MPI.

1) Коммуникаторы, группы, топологии обменов:

MPI_COMM_WORLD – коммуникатор, объединяющий все процессы приложения;

MPI_COMM_NULL – ошибочный коммуникатор;

MPI_COMM_SELF – коммуникатор, включающий только вызывающей процесс;

MPI_GROUP_EMPTY – коммуникатор пустой группы;

MPI_GROUP_NULL – коммуникатор плохой группы;

Функции работы с коммуникаторами:

int MPI_Comm_create(MPI_Comm, MPI_Group, MPI_Comm *) – создание коммуникатора группы;

int MPI_Comm_dup(MPI_Comm, MPI_Comm *) – дублирование коммуникатора;

int MPI_Comm_split(MPI_Comm, int, int, MPI_Comm *) – разбиение коммуникатора на несколько новых;

int MPI_Comm_free(MPI_Comm *) – удаление коммуникатора;

Функции работы с группами:

int MPI_Comm_group(MPI_Comm, MPI_Group *) – создание группы;

int MPI_Group_free(MPI_Group *) – удаление группы;

int MPI_Group_size(MPI_Group group, int *) – определение размера группы;

int MPI_Group_rank(MPI_Group group, int *) – определение номера в группе;

int MPI_Group_translate_ranks (MPI_Group, int, int *, MPI_Group, int *) – трансляция номера из одной группы в другую;

int MPI_Group_incl(MPI_Group group, int, int *, MPI_Group *) – включение в группу новых членов;

int MPI_Group_excl(MPI_Group group, int, int *, MPI_Group *) – исключение членов из группы;

int MPI_Group_compare(MPI_Group, MPI_Group, int *) – сравнение групп;

int MPI_Group_union(MPI_Group, MPI_Group, MPI_Group *) – объединение двух групп в третью;

int MPI_Group_intersection(MPI_Group, MPI_Group, MPI_Group *) – пересечение двух групп в третью;

int MPI_Group_difference(MPI_Group, MPI_Group, MPI_Group *) – дополнение двух групп в третью;

int MPI_Group_range_incl(MPI_Group group, int, int [[3], MPI_Group *) – включение в группу;

int MPI_Group_range_excl(MPI_Group group, int, int [[3], MPI_Group *) – исключение из группы;

Пример 1 (ex09a.c). Создание двух групп и обмен между членами с одинаковыми номерами.

Трансляция и результаты выполнения:

```
>mpicc -o ex09a.px -O2 -lm ex09a.c mynet.c
>mpirun -np 2 -nolocal -machinefile hosts ex09a.px
00 <-> 01
mp=0 i=0 a=0 b=1
mp=0 i=1 a=0 b=-32766
mp=0 i=2 a=-32766 b=0
mp=0 i=3 a=1 b=0
01 <-> 00
mp=1 i=0 a=1 b=0
mp=1 i=1 a=-32766 b=0
mp=1 i=2 a=0 b=-32766
mp=1 i=3 a=0 b=1
>mpirun -np 3 -nolocal -machinefile hosts ex09a.px
mp=2 i=0 a=2 b=0
mp=2 i=1 a=-32766 b=-1078937432
mp=2 i=2 a=1 b=-1078938104
mp=2 i=3 a=-32766 b=134513195
00 <-> 01
mp=0 i=0 a=0 b=1
mp=0 i=1 a=0 b=-32766
mp=0 i=2 a=-32766 b=0
mp=0 i=3 a=1 b=0
01 <-> 00
mp=1 i=0 a=1 b=0
mp=1 i=1 a=-32766 b=0
mp=1 i=2 a=0 b=-32766
mp=1 i=3 a=0 b=1
>mpirun -np 4 -nolocal -machinefile hosts ex09a.px
00 <-> 02
mp=0 i=0 a=0 b=2
mp=0 i=1 a=0 b=-32766
mp=0 i=2 a=-32766 b=0
mp=0 i=3 a=2 b=0
02 <-> 00
mp=2 i=0 a=2 b=0
mp=2 i=1 a=-32766 b=0
mp=2 i=2 a=0 b=-32766
mp=2 i=3 a=0 b=2
01 <-> 03
```

```

mp=1 i=0 a=1 b=3
mp=1 i=1 a=1 b=-32766
mp=1 i=2 a=-32766 b=1
mp=1 i=3 a=3 b=1
03 <-> 01
mp=3 i=0 a=3 b=1
mp=3 i=1 a=-32766 b=1
mp=3 i=2 a=1 b=-32766
mp=3 i=3 a=1 b=3

```

Пример 2 (ex09b.c). Создание двух групп и соответствующих коммунитаторов.

Трансляция и результаты выполнения:

```

>mpicc -o ex09b.px -O2 -lm ex09b.c mynet.c
>mpirun -np 2 -nolocal -machinefile hosts ex09b.px
mp=0 np1=1 np2=1 mp1=0 mp2=-32766 cm1=135 cm2=0 cm3=135 s=5.000000e+00 p=5.000000e+00
mp=1 np1=1 np2=1 mp1=-32766 mp2=0 cm1=0 cm2=135 cm3=135 s=1.000000e+01 p=1.000000e+01
>mpirun -np 3 -nolocal -machinefile hosts ex09b.px
mp=0 np1=1 np2=2 mp1=0 mp2=-32766 cm1=135 cm2=0 cm3=135 s=5.000000e+00 p=5.000000e+00
mp=1 np1=1 np2=2 mp1=-32766 mp2=0 cm1=0 cm2=135 cm3=135 s=1.000000e+01 p=2.500000e+01
mp=2 np1=1 np2=2 mp1=-32766 mp2=1 cm1=0 cm2=135 cm3=135 s=1.500000e+01 p=0.000000e+00
>mpirun -np 4 -nolocal -machinefile hosts ex09b.px
mp=0 np1=2 np2=2 mp1=0 mp2=-32766 cm1=135 cm2=0 cm3=135 s=5.000000e+00 p=1.500000e+01
mp=1 np1=2 np2=2 mp1=1 mp2=-32766 cm1=135 cm2=0 cm3=135 s=1.000000e+01 p=0.000000e+00
mp=2 np1=2 np2=2 mp1=-32766 mp2=0 cm1=0 cm2=135 cm3=135 s=1.500000e+01 p=3.500000e+01
mp=3 np1=2 np2=2 mp1=-32766 mp2=1 cm1=0 cm2=135 cm3=135 s=2.000000e+01 p=0.000000e+00

```

2) Стандарт MPI-2:

- а) объединение локальных и удаленных процессов;
- б) параллельный ввод-вывод.

2. Параллельная сортировка.

Параллельная сортировка распределенного массива. Топология: линейка процессоров. Синхронные обмены. Условие: Каждый вычислитель имеет часть случайного массива и память для одной копии этой части.

Пример 3 (ex10a.c). Заготовка для генерации распределенного массива и проведения параллельной сортировки.

Трансляция и результаты работы:

```

>mpicc -o ex10a.px -O2 -lm ex10a.c mynet.c myrand.c
>mpirun -np 1 -nolocal -machinefile hosts ex10a.px
mp=0 ns=10000 i1=0 i2=9999 nc=10000
mp=0 amin=1.536087e+00 amax=3.276779e+04 t1=7.759000e-03
>mpirun -np 2 -nolocal -machinefile hosts ex10a.px
mp=0 ns=10000 i1=0 i2=4999 nc=5000
mp=1 ns=10000 i1=5000 i2=9999 nc=5000
mp=0 amin=1.536087e+00 amax=3.276779e+04 t1=1.320900e-02
mp=1 amin=2.633987e+00 amax=3.275803e+04 t1=9.561000e-03

```

Задание 6 (ex10b.c): на основе кода ex10a.c сделать реализацию решения задачи, написав функцию слияния и подобрав необходимое число циклов обменов.

Семинар 7. Решение линейных пространственно одномерных краевых задач.

1. Постановка задачи.

$$\frac{d}{dx} \left(k(x) \frac{du}{dx} \right) - q(x)u = -f(x), \quad a < x < b, \quad u(a) = u_a, \quad u(b) = u_b.$$

Тестовый пример:

$$k(x) = 1 + \left(\frac{x-a}{b-a} \right)^2, \quad q(x) = 1 + \left(\frac{b-x}{b-a} \right)^2, \quad f(x) = q(x)u(x) - k'(x)u'(x) - k(x)u''(x),$$

$$u(x) = u_a \cos \left(\frac{\pi(x-a)}{2(b-a)} \right) + u_b \sin \left(\frac{\pi(x-a)}{2(b-a)} \right), \quad u'(x) = \frac{\pi}{2(b-a)} \left[-u_a \sin \left(\frac{\pi(x-a)}{2(b-a)} \right) + u_b \cos \left(\frac{\pi(x-a)}{2(b-a)} \right) \right],$$

$$u''(x) = \frac{\pi^2}{4(b-a)^2} \left[-u_a \cos \left(\frac{\pi(x-a)}{2(b-a)} \right) - u_b \sin \left(\frac{\pi(x-a)}{2(b-a)} \right) \right], \quad k'(x) = 1 + \frac{2(x-a)}{(b-a)^2}.$$

2. Численный алгоритм.

$$\text{Разностная схема на равномерной сетке } \omega_x = \left\{ x = x_i = ih_x, i = 0, \dots, N_x, h_x = \frac{b-a}{N_x} \right\} :$$

$$\frac{1}{h_x} \left\{ k_{i+1/2} \frac{y_{i+1} - y_i}{h_x} - k_{i-1/2} \frac{y_i - y_{i-1}}{h_x} \right\} - q_i y_i = -f_i, \quad 0 < i < N_x, \quad y_0 = u_a, \quad y_{N_x} = u_b;$$

$$k_{i\pm 1/2} = \frac{k_i + k_{i\pm 1}}{2} \vee \frac{2k_i k_{i\pm 1}}{k_i + k_{i\pm 1}}, \quad k_i = k(x_i), \quad q_i = q(x_i), \quad f_i = f(x_i), \quad h_x = \begin{cases} 0.5h_x, & i = 0, N_x, \\ h_x, & 1 < i < N_x. \end{cases}$$

3. Параллельная реализация.

Канонический вид задачи:

$$\begin{aligned} A_i y_{i-1} - C_i y_i + B_i y_{i+1} &= -F_i, \quad 1 \leq i \leq N-1, \\ -C_0 y_0 + B_0 y_1 &= -F_0, \quad A_N y_{N-1} - C_N y_N = -F_N. \end{aligned}$$

Значения коэффициентов:

$$A_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i-1}), & 1 \leq i \leq N-1, \end{cases} \quad B_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i+1}), & 1 \leq i \leq N-1, \end{cases} \quad C_i = \begin{cases} 1, & i = 0, N, \\ A_i + B_i + h^2 q_i, & 1 \leq i \leq N-1, \end{cases}$$

$$F_0 = u_a, \quad F_N = u_b, \quad F_i = h^2 f_i, \quad 1 \leq i \leq N-1.$$

Алгоритм решения - параллельная прогонка.

Пример 1 (ex11a.c). Решение линейной первой краевой задачи методом правой прогонки.

Трансляция и результаты выполнения:

`>mpicc -o ex11a.px -O2 -lm ex11a.c mусom.c mynet.c myprog.c`

Расчеты на сходимости по сетке:

`>mpirun -np 1 -nolocal -machinefile hosts ex11a.px <10...10000000>`

```
nx=      10 t1=7.000000e-06 t2=0.000000e+00 dmax=1.260130e-03
nx=     100 t1=3.500000e-05 t2=0.000000e+00 dmax=1.258753e-05
nx=    1000 t1=3.300000e-04 t2=0.000000e+00 dmax=1.258734e-07
nx=   10000 t1=3.146000e-03 t2=0.000000e+00 dmax=1.236859e-09
nx=  100000 t1=3.346400e-02 t2=0.000000e+00 dmax=9.804504e-10
nx=1000000 t1=3.342550e-01 t2=0.000000e+00 dmax=3.189187e-08
nx=10000000 t1=3.009283e+00 t2=0.000000e+00 dmax=1.672966e-05
```

Расчеты на эффективность:

`>mpirun -np <1...16> -nolocal -machinefile hosts ex11a.px 10000000`

```
np= 1 nx=10000000 t1=3.227453e+00 t2=0.000000e+00 dmax=1.672966e-05
np= 2 nx=10000000 t1=2.164483e+00 t2=2.252420e-01 dmax=1.659745e-05
np= 3 nx=10000000 t1=1.583892e+00 t2=2.862640e-01 dmax=1.655211e-05
np= 4 nx=10000000 t1=1.214389e+00 t2=2.375560e-01 dmax=1.650617e-05
np= 5 nx=10000000 t1=9.821130e-01 t2=1.846140e-01 dmax=1.640877e-05
np= 6 nx=10000000 t1=8.235920e-01 t2=1.575800e-01 dmax=1.660942e-05
np= 7 nx=10000000 t1=7.175770e-01 t2=1.457190e-01 dmax=1.657224e-05
np= 8 nx=10000000 t1=6.279780e-01 t2=1.293170e-01 dmax=1.647258e-05
np= 9 nx=10000000 t1=5.709290e-01 t2=1.182260e-01 dmax=1.653773e-05
np=10 nx=10000000 t1=5.288050e-01 t2=1.222760e-01 dmax=1.645452e-05
np=11 nx=10000000 t1=4.882020e-01 t2=1.132150e-01 dmax=1.648861e-05
np=12 nx=10000000 t1=4.424140e-01 t2=1.010360e-01 dmax=1.657097e-05
np=13 nx=10000000 t1=4.208710e-01 t2=1.006940e-01 dmax=1.651073e-05
np=14 nx=10000000 t1=3.901090e-01 t2=9.523800e-02 dmax=1.663803e-05
np=15 nx=10000000 t1=3.699740e-01 t2=9.468500e-02 dmax=1.645407e-05
np=16 nx=10000000 t1=3.586580e-01 t2=1.007230e-01 dmax=1.648750e-05
```

Пример 2 (ex11b.c). Решение линейной второй краевой задачи методом правой прогонки.

Трансляция и результаты выполнения:

```
>mpicc -o ex11b.px -O2 -lm ex11b.c mysom.c mynet.c myprog.c
```

Расчеты на эффективность:

```
>mpirun -np <1...16> -nolocal -machinefile hosts ex11a.px 10000000
```

```
np= 1 nx=10000000 t1=3.253029e+00 t2=0.000000e+00 dmax=2.813989e-04  
np= 2 nx=10000000 t1=2.299431e+00 t2=1.907540e-01 dmax=3.204477e-04  
np= 4 nx=10000000 t1=1.282975e+00 t2=2.143770e-01 dmax=3.189007e-04  
np= 8 nx=10000000 t1=6.725080e-01 t2=1.207250e-01 dmax=3.189379e-04  
np=16 nx=10000000 t1=3.714060e-01 t2=8.903900e-02 dmax=3.184931e-04
```

Задание 7 (ex11c.c): на базе примеров ex11a.c, ex11b.c решить смешанную задачу ($u(a)=u_a$, $u'(b)=u_b$) для $k(x) = +\exp(-5*(x-a)/(b-a))$, $q(x)=1-0.5*\sin(10*(x-a)/(b-a))$, оценить эффективность распараллеливания. Добавить треды.

Семинар 8. Решение нелинейных пространственно одномерных краевых задач.

1. Постановка задачи.

$$\frac{d}{dx} \left(k(u) \frac{du}{dx} \right) - q(u)u = -f(u), \quad a < x < b, \quad u(a) = u_a, \quad u(b) = u_b.$$

Тестовый пример:

$$k(u) = 1 + u^2, \quad q(u) = 1 / (1 + u^2), \quad f(u) = q(u)u - k'(u)(u')^2 - k(u)u'',$$

$$u(x) = u_a \exp\left(\alpha \frac{x-a}{b-a}\right), \quad u_b = u_a \exp(\alpha), \quad u' = \frac{\alpha}{(b-a)} u, \quad u'' = \frac{\alpha^2}{(b-a)^2} u, \quad k'(u) = 2u.$$

2. Численный алгоритм.

$$\text{Разностная схема на равномерной сетке } \omega_x = \left\{ x = x_i = ih_x, i = 0, \dots, N_x, h_x = \frac{b-a}{N_x} \right\}:$$

$$\frac{1}{h_x} \left\{ k_{i+1/2} \frac{y_{i+1} - y_i}{h_x} - k_{i-1/2} \frac{y_i - y_{i-1}}{h_x} \right\} - q_i y_i = -f_i, \quad 0 < i < N_x, \quad y_0 = u_a, \quad y_{N_x} = u_b;$$

$$k_{i\pm 1/2} = \frac{k_i + k_{i\pm 1}}{2} \vee \frac{2k_i k_{i\pm 1}}{k_i + k_{i\pm 1}}, \quad k_i = k(y_i), \quad q_i = q(y_i), \quad f_i = f(y_i), \quad h_x = \begin{cases} 0.5h_x, & i = 0, N_x, \\ h_x, & 1 \leq i \leq N_x. \end{cases}$$

Значения коэффициентов:

$$A_i(y) = \begin{cases} 0, & i = 0, N, \\ 0.5(k(y_i) + k(y_{i-1})), & 1 \leq i \leq N-1, \end{cases} \quad B_i(y) = \begin{cases} 0, & i = 0, N, \\ 0.5(k(y_i) + k(y_{i+1})), & 1 \leq i \leq N-1, \end{cases}$$

$$C_i(y) = \begin{cases} 1, & i = 0, N, \\ 0.5k(y_{i-1}) + k(y_i) + 0.5k(y_{i+1}) + h^2 q(y_i), & 1 \leq i \leq N-1, \end{cases} \quad F_i(y) = \begin{cases} u_a, & i = 0, \\ h^2 f(y_i), & 1 \leq i \leq N-1, \\ u_b, & i = N. \end{cases}$$

3. Метод простой итерации.

$$A_i(y) y_{i-1}^{s+1} - C_i(y) y_i^{s+1} + B_i(y) y_{i+1}^{s+1} = -F_i(y), \quad 1 \leq i \leq N-1,$$

$$-C_0(y) y_0^{s+1} + B_0(y) y_1^{s+1} = -F_0(y), \quad -C_N(y) y_N^{s+1} + A_N(y) y_{N-1}^{s+1} = -F_N(y), \quad s = 0, 1, \dots$$

$$\text{Начальное приближение: } y_i^0 = u_a + (u_b - u_a) \frac{x_i - a}{b - a}, \quad i = 0, \dots, N.$$

Пример 1 (ex12a.c). Решение задачи методом простой итерации.

Трансляция и результаты выполнения:

```
>mpicc -o ex12a.px -O2 -lm ex12a.c mycom.c mynet.c myprog.c
Сходимость итераций и точность в зависимости от размерности:
>mpirun -np 1 -nolocal -machinefile hosts ex12a.px <10...1000000>
np=1 nx=10      it=30 time=5.400000e-05 dmax=2.830711e-03
np=1 nx=100     it=24 time=4.170000e-04 dmax=2.869907e-05
np=1 nx=1000    it=20 time=3.491000e-03 dmax=2.871838e-07
np=1 nx=10000   it=16 time=2.766900e-02 dmax=1.581609e-08
np=1 nx=100000  it=12 time=2.604180e-01 dmax=9.598133e-07
np=1 nx=1000000 it=7  time=1.512643e+00 dmax=2.028339e-04
Расчеты на эффективность:
>mpirun -np <1...6> -nolocal -machinefile hosts ex12a.px 1000000
np=1 nx=1000000 it=7  time=1.512643e+00 dmax=2.028339e-04
np=2 nx=1000000 it=7  time=1.063321e+00 dmax=2.028401e-04
np=3 nx=1000000 it=7  time=9.272730e-01 dmax=2.028154e-04
np=4 nx=1000000 it=7  time=6.390440e-01 dmax=2.028409e-04
np=5 nx=1000000 it=7  time=6.064660e-01 dmax=2.028184e-04
np=6 nx=1000000 it=7  time=5.269030e-01 dmax=2.028220e-04
```

4. Метод Ньютона.

$$A_i(y) y_{i-1}^{s+1} - C_i(y) y_i^{s+1} + B_i(y) y_{i+1}^{s+1} + \theta \sum_{j=0}^N \left[\frac{\partial A_i}{\partial y_j}(y) y_{i-1}^s - \frac{\partial C_i}{\partial y_j}(y) y_i^s + \frac{\partial B_i}{\partial y_j}(y) y_{i+1}^s + \frac{\partial F_i}{\partial y_j}(y) \right] y_j^{s+1} =$$

$$= -F_i(y) + \theta \sum_{j=0}^N \left[\frac{\partial A_i}{\partial y_j}(y) y_{i-1}^s - \frac{\partial C_i}{\partial y_j}(y) y_i^s + \frac{\partial B_i}{\partial y_j}(y) y_{i+1}^s + \frac{\partial F_i}{\partial y_j}(y) \right] y_j^s, \quad 1 \leq i \leq N-1,$$

$$\begin{aligned}
& -C_0(y) y_0^{s+1} + B_0(y) y_1^{s+1} + \theta \sum_{j=0}^N \left[-\frac{\partial C_0}{\partial y_j}(y) y_0^s + \frac{\partial B_0}{\partial y_j}(y) y_1^s + \frac{\partial F_0}{\partial y_j}(y) \right] y_j^{s+1} = \\
& = -F_0(y) + \theta \sum_{j=0}^N \left[-\frac{\partial C_0}{\partial y_j}(y) y_0^s + \frac{\partial B_0}{\partial y_j}(y) y_1^s + \frac{\partial F_0}{\partial y_j}(y) \right] y_j^s, \\
& A_N(y) y_{N-1}^{s+1} - C_N(y) y_N^{s+1} + \theta \sum_{j=0}^N \left[\frac{\partial A_N}{\partial y_j}(y) y_{N-1}^s - \frac{\partial C_N}{\partial y_j}(y) y_N^s + \frac{\partial F_N}{\partial y_j}(y) \right] y_j^{s+1} = \\
& = -F_N(y) + \theta \sum_{j=0}^N \left[\frac{\partial A_N}{\partial y_j}(y) y_{N-1}^s - \frac{\partial C_N}{\partial y_j}(y) y_N^s + \frac{\partial F_N}{\partial y_j}(y) \right] y_j^s,
\end{aligned}$$

$$s = 0, 1, \dots, \quad 0 \leq \theta \leq 1.$$

Если $k(u) \equiv 1$, то

$$\begin{aligned}
& \frac{\partial A_i}{\partial y_j}(y) = 0, \quad \frac{\partial B_i}{\partial y_j}(y) = 0, \quad \frac{\partial C_i}{\partial y_i}(y) = h^2 \frac{\partial q}{\partial y}(y_i), \quad \frac{\partial F_i}{\partial y_i}(y) = h^2 \frac{\partial f}{\partial y}(y_i), \\
& \tilde{C}_i(y) = C_i(y) + \theta \left(\frac{\partial C_i}{\partial y_i}(y) y_i^s - \frac{\partial F_i}{\partial y_i}(y) \right), \quad \tilde{F}_i(y) = F_i(y) + \theta \left(\frac{\partial C_i}{\partial y_i}(y) y_i^s - \frac{\partial F_i}{\partial y_i}(y) \right) y_i^s.
\end{aligned}$$

Задание 8 (ex12b.c): Реализовать метод Ньютона для $k(u)=1$. Провести расчеты на сходимость итераций при различных θ из $[0,1]$.

Семинар 9. Решение одномерного уравнения теплопроводности.

1. Постановка задачи.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(k(x) \frac{\partial u}{\partial x} \right) + f(x, t), \quad a < x < b,$$

$$u(x, 0) = g_0(x), \quad u(a) = g_1(t), \quad u(b) = g_2(t),$$

$$k(x) = \begin{cases} k_1, & a \leq x < x_k, \\ k_2, & x_k \leq x \leq b, \end{cases} \quad f(x, t) = Q_0 \exp[-(x - x_0)^2 / r_0^2] (1 - \exp[-t / \tau_0]),$$

$$g_0(x) = u_0, \quad g_1(t) = u_0 + (u_1 - u_0)(1 - \exp[-t / \tau_1]), \quad g_2(t) = u_0.$$

2. Численный алгоритм.

$$\text{Равномерная сетка } \Omega = \omega_x \times \omega_t, \quad \omega_x = \left\{ x_i = ih_x, i = 0, \dots, N_x, h_x = \frac{b-a}{N_x} \right\}, \quad \omega_t = \left\{ t_j = j\tau, j = 0, \dots, N_t, \tau = \frac{t_{\max}}{N_t} \right\}.$$

$$\begin{aligned} \frac{y_i^{j+1} - y_i^j}{\tau} = & \sigma \left[\frac{1}{h_x} \left\{ k_{i+1/2} \frac{y_{i+1}^{j+1} - y_i^{j+1}}{h_x} - k_{i-1/2} \frac{y_i^{j+1} - y_{i-1}^{j+1}}{h_x} \right\} + f_i^{j+1} \right] + \\ \text{Схема с весами:} \quad & + (1 - \sigma) \left[\frac{1}{h_x} \left\{ k_{i+1/2} \frac{y_{i+1}^j - y_i^j}{h_x} - k_{i-1/2} \frac{y_i^j - y_{i-1}^j}{h_x} \right\} + f_i^j \right], \quad 0 < i < N_x, \quad 0 \leq j < N_t, \end{aligned}$$

$$y_i^0 = g_0(x_i), \quad y_0^j = g_1(t_j), \quad y_{N_x}^j = g_2(t_j),$$

$$k_{i\pm 1/2} = \frac{2k_i k_{i\pm 1}}{k_i + k_{i\pm 1}}, \quad k_i = k(x_i), \quad f_i^j = f(x_i, t_j), \quad h_x = \begin{cases} 0.5h_x, & i = 0, N_x, \\ h_x, & 1 < i < N_x. \end{cases}$$

Устойчивость (обычная, абсолютная, как в ОДУ):

$$\tau^{(1)} = \frac{0.5h_x^2}{\max k(x)} = \frac{0.5h_x^2}{\max(k_1, k_2)}, \quad \tau^{(2)} = \frac{0.5h_x}{\sqrt{\max k(x)}} = \frac{0.5h_x}{\sqrt{\max(k_1, k_2)}}, \quad \tau^{(3)} = \frac{1}{\max |f(x, t)|} = \frac{1}{Q_0}.$$

Для $\sigma = 0$: $\tau \leq \min(\tau^{(1)}, \tau^{(2)}, \tau^{(3)}) = \min(\tau^{(1)}, \tau^{(3)})$; для $\sigma = 1$: $\tau \leq \min(\tau^{(2)}, \tau^{(3)})$;

для $\sigma = 0.5$: $\tau \leq \min(2\tau^{(1)}, 2\tau^{(2)}, 2\tau^{(3)}) = \min(2\tau^{(1)}, 2\tau^{(3)})$.

Вычисление реального шага по времени:

$$\tau^{(ycm)} \rightarrow N_t = \left\lceil \frac{t_{\max}}{\tau^{(ycm)}} \right\rceil + 1 \rightarrow \tau = \frac{t_{\max}}{N_t}.$$

Расчеты на установление, если есть стационар: $g_i = \max_i \left| \frac{y_i^{j+1} - y_i^j}{\tau y_i^j} \right| \leq \varepsilon$.

3. Параллельная реализация.

Разбиение пространственной сетки на равные интервалы. Линейная топология обменов.

$$y_i^{j+1} = y_i^j + \gamma \left\{ \frac{2k_i k_{i+1}}{k_i + k_{i+1}} (y_{i+1}^j - y_i^j) - \frac{2k_i k_{i-1}}{k_i + k_{i-1}} (y_i^j - y_{i-1}^j) \right\} + \tau f_i^j, \quad 0 < i < N_x,$$

Расчетные формулы явной схемы:

$$y_0^{j+1} = g_1(t_{j+1}), \quad y_{N_x}^{j+1} = g_2(t_{j+1}), \quad \gamma = \frac{\tau}{h_x h_x}.$$

$$\text{Уравнения неявной схемы:} \quad y_i^{j+1} - \gamma \left\{ \frac{2k_i k_{i+1}}{k_i + k_{i+1}} (y_{i+1}^{j+1} - y_i^{j+1}) - \frac{2k_i k_{i-1}}{k_i + k_{i-1}} (y_i^{j+1} - y_{i-1}^{j+1}) \right\} = y_i^j + \tau f_i^{j+1}, \quad 0 < i < N_x,$$

$$y_0^{j+1} = g_1(t_{j+1}), \quad y_{N_x}^{j+1} = g_2(t_{j+1}).$$

$$y_i^{j+1} - \frac{\gamma}{2} \left\{ \frac{2k_i k_{i+1}}{k_i + k_{i+1}} (y_{i+1}^{j+1} - y_i^{j+1}) - \frac{2k_i k_{i-1}}{k_i + k_{i-1}} (y_i^{j+1} - y_{i-1}^{j+1}) \right\} =$$

$$\text{Симметричная схема:} \quad = y_i^j + \frac{\gamma}{2} \left\{ \frac{2k_i k_{i+1}}{k_i + k_{i+1}} (y_{i+1}^j - y_i^j) - \frac{2k_i k_{i-1}}{k_i + k_{i-1}} (y_i^j - y_{i-1}^j) \right\} + \frac{\tau}{2} (f_i^{j+1} + f_i^j), \quad 0 < i < N_x,$$

$$y_0^{j+1} = g_1(t_{j+1}), \quad y_{N_x}^{j+1} = g_2(t_{j+1}).$$

Алгоритм реализации явной схемы – прямые вычисления, неявной и симметричной схем – параллельная прогонка.

Пример 1 (ex13a.c). Решение задачи по явной схеме.

Пример 2 (ex13b.c). Решение задачи по неявной схеме.

Трансляция и результаты выполнения:

```
>mpicc -o ex13a.px -O2 -lm ex13a.c mycom.c mynet.c myio.c
```

```
>mpicc -o ex13b.px -O2 -lm ex13b.c mycom.c mynet.c myio.c myprog.c
```

Расчеты на одном процессоре:

```
>mpirun -np 1 -nolocal -machinefile hosts ex13a.px
```

```
nx=200 hx=5.000000e-03 tau=1.250000e-06 ntm=1000000
ntv=50000 tv=6.250000e-02 gt=2.608438e+00 tcpu=7.621830e-01
ntv=100000 tv=1.250000e-01 gt=1.819795e-01 tcpu=1.525041e+00
ntv=150000 tv=1.875000e-01 gt=1.287988e-02 tcpu=2.287851e+00
ntv=200000 tv=2.500000e-01 gt=9.140317e-04 tcpu=2.872474e+00
ntv=250000 tv=3.125000e-01 gt=6.488034e-05 tcpu=3.444287e+00
ntv=300000 tv=3.750000e-01 gt=4.605373e-06 tcpu=4.016293e+00
ntv=350000 tv=4.375000e-01 gt=3.269213e-07 tcpu=4.588261e+00
ntv=400000 tv=5.000000e-01 gt=2.354150e-08 tcpu=5.160074e+00
```

```
>mpirun -np 1 -nolocal -machinefile hosts ex13b.px
```

```
nx=200 hx=5.000000e-03 tau=7.905694e-04 ntm=2529
ntv=100 tv=7.905694e-02 gt=1.324355e+00 tcpu=2.376000e-03
ntv=200 tv=1.581139e-01 gt=4.849999e-02 tcpu=4.717000e-03
ntv=300 tv=2.371708e-01 gt=1.802814e-03 tcpu=7.061000e-03
ntv=400 tv=3.162278e-01 gt=6.707688e-05 tcpu=9.401000e-03
ntv=500 tv=3.952847e-01 gt=2.495812e-06 tcpu=1.172200e-02
ntv=600 tv=4.743416e-01 gt=9.286492e-08 tcpu=1.404600e-02
```

Расчеты на эффективность:

```
>mpirun -np <1..12> -nolocal -machinefile hosts ex13a.px 10000 2000 10000
```

```
np=1 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=1.977835e+01
np=2 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=1.249927e+01
np=3 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=9.079243e+00
np=4 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=7.574098e+00
np=5 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=6.698560e+00
np=6 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=5.972717e+00
np=7 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=5.401055e+00
np=8 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=4.887469e+00
np=9 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=4.596798e+00
np=10 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=4.400385e+00
np=11 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=4.325092e+00
np=12 nx=10000 ntv=10000 tv=5.000000e-06 gt=9.990008e+01 tcpu=4.053824e+00
```

```
>mpirun -np <1..8> -nolocal -machinefile hosts ex13b.px 10000 2000 10000
```

```
np=1 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=2.308190e+01
np=2 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.442830e+01
np=3 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.231996e+01
np=4 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.032698e+01
np=5 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.147053e+01
np=6 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.144447e+01
np=7 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.367407e+01
np=8 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.132352e+01
np=9 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.392584e+01
np=10 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.387531e+01
np=11 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.445106e+01
np=12 nx=10000 ntv=10000 tv=1.581139e-01 gt=3.932917e-02 tcpu=1.452619e+01
```

Задание 9 (ex13c.c): Реализовать симметричную схему. Провести расчеты и сравнить решение с явной и неявной схемами в стационаре.

1. Постановка задачи.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x_1} \left(k(x_1, x_2) \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k(x_1, x_2) \frac{\partial u}{\partial x_2} \right) + f(x_1, x_2, t), \quad (x_1, x_2) \in D = (a_1, b_1) \times (a_2, b_2),$$

$$u(x_1, x_2, 0) = g_0(x_1, x_2), \quad u(a_1, x_2, t) = g_{11}(t), \quad u(b_1, x_2, t) = g_{12}(t), \quad u(x_1, a_2, t) = g_{21}(t), \quad u(x_1, b_2, t) = g_{22}(t),$$

$$k(x_1, x_2) = \begin{cases} k_1, & (x_1, x_2) \in D_0, \\ k_2, & (x_1, x_2) \notin D_0, \end{cases} \quad D_0 = [x_{11}, x_{12}] \times [x_{21}, x_{22}],$$

$$f(x_1, x_2, t) = Q_0 \exp \left[-(x_1 - x_{10})^2 / r_0^2 - (x_2 - x_{20})^2 / r_0^2 \right] (1 - \exp[-t / \tau_0]),$$

$$g_0(x_1, x_2) = u_0, \quad g_{11}(t) = u_0, \quad g_{12}(t) = u_0, \quad g_{21}(t) = u_0 + (u_1 - u_0)(1 - \exp[-t / \tau_1]), \quad g_{22}(t) = u_0.$$

2. Численные алгоритмы.

$$\text{МКР на равномерной сетке } \Omega = \omega_{x_1} \times \omega_{x_2} \times \omega_t, \quad \omega_{x_\alpha} = \left\{ x_{\alpha, i_\alpha} = i_\alpha h_\alpha, i_\alpha = 0, \dots, N_\alpha, h_\alpha = \frac{b_\alpha - a_\alpha}{N_\alpha} \right\}, \quad \alpha = 1, 2,$$

$$\omega_t = \left\{ t_j = j\tau, j = 0, \dots, N_t, \tau = \frac{t_{\max}}{N_t} \right\}.$$

Схема с весами:

$$\begin{aligned} \frac{y_{i_1 i_2}^{j+1} - y_{i_1 i_2}^j}{\tau} = & \sigma \frac{1}{h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^{j+1} - y_{i_1 i_2}^{j+1}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1 i_2}^{j+1} - y_{i_1-1, i_2}^{j+1}}{h_1} \right\} + \sigma \frac{1}{h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^{j+1} - y_{i_1 i_2}^{j+1}}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1 i_2}^{j+1} - y_{i_1, i_2-1}^{j+1}}{h_2} \right\} + \\ & + (1-\sigma) \frac{1}{h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^j - y_{i_1 i_2}^j}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1 i_2}^j - y_{i_1-1, i_2}^j}{h_1} \right\} + (1-\sigma) \frac{1}{h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^j - y_{i_1 i_2}^j}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1 i_2}^j - y_{i_1, i_2-1}^j}{h_2} \right\} + \\ & + \sigma f_{i_1 i_2}^{j+1} + (1-\sigma) f_{i_1 i_2}^j, \quad 0 < i_\alpha < N_\alpha, \quad \alpha = 1, 2, \quad 0 \leq j < N_t, \end{aligned}$$

$$y_{i_1 i_2}^0 = g_0(x_{1, i_1}, x_{2, i_2}), \quad y_{0, i_2}^j = g_{11}(t_j), \quad y_{N_1, i_2}^j = g_{12}(t_j), \quad y_{i_1, 0}^j = g_{21}(t_j), \quad y_{i_1, N_2}^j = g_{22}(t_j),$$

$$k_{i_1 \pm 1/2, i_2} = \frac{2k_{i_1 i_2} k_{i_1 \pm 1, i_2}}{k_{i_1 i_2} + k_{i_1 \pm 1, i_2}}, \quad k_{i_1, i_2 \pm 1/2} = \frac{2k_{i_1 i_2} k_{i_1, i_2 \pm 1}}{k_{i_1 i_2} + k_{i_1, i_2 \pm 1}}, \quad k_{i_1 i_2} = k(x_{1, i_1}, x_{2, i_2}), \quad f_{i_1 i_2}^j = f(x_{1, i_1}, x_{2, i_2}, t_j), \quad h_\alpha = \begin{cases} 0.5h_\alpha, & i = 0, N_\alpha, \\ h_\alpha, & 1 < i < N_\alpha. \end{cases}$$

Расчетные формулы для явной схемы:

$$y_{i_1 i_2}^{j+1} = y_{i_1 i_2}^j + B_{1i_1 i_2} (y_{i_1+1, i_2}^j - y_{i_1 i_2}^j) - A_{1i_1 i_2} (y_{i_1 i_2}^j - y_{i_1-1, i_2}^j) + B_{2i_1 i_2} (y_{i_1, i_2+1}^j - y_{i_1 i_2}^j) - A_{2i_1 i_2} (y_{i_1 i_2}^j - y_{i_1, i_2-1}^j) + \tau f_{i_1 i_2}^j,$$

$$A_{1i_1 i_2} = \gamma_1 k_{i_1-1/2, i_2}, \quad B_{1i_1 i_2} = \gamma_1 k_{i_1+1/2, i_2}, \quad A_{2i_1 i_2} = \gamma_2 k_{i_1, i_2-1/2}, \quad B_{2i_1 i_2} = \gamma_2 k_{i_1, i_2+1/2}, \quad \gamma_1 = \frac{\tau}{h_1 h_1}, \quad \gamma_2 = \frac{\tau}{h_2 h_2}.$$

Устойчивость явной схемы:

$$\tau \leq \min(\tau^{(1)}, \tau^{(3)}), \quad \tau^{(1)} = \frac{0.25 \min(h_x^2, h_y^2)}{\max k(x_1, x_2)} = \frac{0.25 \min(h_x^2, h_y^2)}{\max(k_1, k_2)}, \quad \tau^{(3)} = \frac{1}{\max |f(x_1, x_2, t)|} = \frac{1}{Q_0}.$$

Неявная локально одномерная схема:

$$\begin{aligned} \frac{y_{i_1 i_2}^{j+1/2} - y_{i_1 i_2}^j}{\tau} = & \frac{1}{2h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^{j+1/2} - y_{i_1 i_2}^{j+1/2}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1 i_2}^{j+1/2} - y_{i_1-1, i_2}^{j+1/2}}{h_1} \right\} + \\ & + \frac{1}{2h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^j - y_{i_1 i_2}^j}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1 i_2}^j - y_{i_1, i_2-1}^j}{h_2} \right\} + \frac{1}{2} f_{i_1 i_2}^j, \\ \frac{y_{i_1 i_2}^{j+1} - y_{i_1 i_2}^{j+1/2}}{\tau} = & \frac{1}{2h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^{j+1/2} - y_{i_1 i_2}^{j+1/2}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1 i_2}^{j+1/2} - y_{i_1-1, i_2}^{j+1/2}}{h_1} \right\} + \\ & + \frac{1}{2h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^{j+1} - y_{i_1 i_2}^{j+1}}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1 i_2}^{j+1} - y_{i_1, i_2-1}^{j+1}}{h_2} \right\} + \frac{1}{2} f_{i_1 i_2}^{j+1}, \end{aligned}$$

$$0 < i_\alpha < N_\alpha, \quad \alpha = 1, 2, \quad 0 \leq j < N_t.$$

Устойчивость НЛОС:

$$\tau \leq \min(\tau^{(2)}, 2\tau^{(3)}), \quad \tau^{(2)} = \frac{0.5 \min(h_x, h_y)}{\sqrt{\max k(x_1, x_2)}} = \frac{0.5 \min(h_x, h_y)}{\sqrt{\max(k_1, k_2)}}, \quad \tau^{(3)} = \frac{1}{\max |f(x, t)|} = \frac{1}{Q_0}.$$

Расчетные формулы:

$$y_{i_1 i_2}^{j+1/2} - B_{1i_1 i_2} \left(y_{i_1+1, i_2}^{j+1/2} - y_{i_1 i_2}^{j+1/2} \right) + A_{1i_1 i_2} \left(y_{i_1 i_2}^{j+1/2} - y_{i_1-1, i_2}^{j+1/2} \right) = y_{i_1 i_2}^j + B_{2i_1 i_2} \left(y_{i_1, i_2+1}^j - y_{i_1 i_2}^j \right) - A_{2i_1 i_2} \left(y_{i_1 i_2}^j - y_{i_1, i_2-1}^j \right) + \frac{\tau}{2} f_{i_1 i_2}^j,$$
$$y_{i_1 i_2}^{j+1} - B_{2i_1 i_2} \left(y_{i_1, i_2+1}^{j+1} - y_{i_1 i_2}^{j+1} \right) + A_{2i_1 i_2} \left(y_{i_1 i_2}^{j+1} - y_{i_1, i_2-1}^{j+1} \right) = y_{i_1 i_2}^{j+1/2} + B_{1i_1 i_2} \left(y_{i_1+1, i_2}^{j+1/2} - y_{i_1 i_2}^{j+1/2} \right) - A_{1i_1 i_2} \left(y_{i_1 i_2}^{j+1/2} - y_{i_1-1, i_2}^{j+1/2} \right) + \frac{\tau}{2} f_{i_1 i_2}^{j+1},$$
$$A_{1i_1 i_2} = \gamma_1 k_{i_1-1/2, i_2}, \quad B_{1i_1 i_2} = \gamma_1 k_{i_1+1/2, i_2}, \quad A_{2i_1 i_2} = \gamma_2 k_{i_1, i_2-1/2}, \quad B_{2i_1 i_2} = \gamma_2 k_{i_1, i_2+1/2}, \quad \gamma_1 = \frac{\tau}{2\hbar_1 h_1}, \quad \gamma_2 = \frac{\tau}{2\hbar_2 h_2}.$$

3. Параллельная реализация.

В случае явной схемы параллелим расчетные формулы.

В случае неявной ЛОС применяем комбинацию параллельных прогонок.

Пример 1 (ex14a.c). Реализация явной схемы на решетке процессоров.

Трансляция и результаты выполнения:

```
>mpicc -o ex14a.px -O2 -lm ex14a.c мусом.с мунет.с myio.с
```

Расчеты на эффективность:

```
>mpirun -np <1..16> -nolocal -machinefile hosts ex14a.px 100 100 5000 25000
Grid=1x1 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=2.241195e+01
Grid=1x2 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=1.596867e+01
Grid=1x3 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=1.253931e+01
Grid=2x2 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=1.030746e+01
Grid=1x5 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=9.375307e+00
Grid=2x3 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=8.089331e+00
Grid=1x7 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=7.802669e+00
Grid=2x4 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=6.910250e+00
Grid=3x3 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=6.936690e+00
Grid=2x5 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=7.533998e+00
Grid=1x11 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=7.455552e+00
Grid=3x4 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=7.639191e+00
Grid=1x13 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=7.125131e+00
Grid=2x7 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=5.894694e+00
Grid=3x5 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=6.417959e+00
Grid=4x4 n1=100 n2=100 ntv=25000 tv=6.250000e-02 gt=2.196164e+00 tcpu=5.550545e+00
```

Пример 2 (ex14b.c). Заготовка для реализации неявной локально одномерной схемы на решетке процессоров.

Трансляция:

```
>mpicc -o ex14b.px -O2 -lm ex14b.c мусом.с мунет.с myio.с
```

Задание 10 (ex14b.c). Реализовать второй недостающий шаг неявной локально одномерной схемы на решетке процессоров. Сравнить решение с решением по явной схеме. Измерить эффективность.

1. Постановка задачи.

$$\frac{\partial}{\partial x_1} \left(k_1(x_1, x_2) \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k_2(x_1, x_2) \frac{\partial u}{\partial x_2} \right) = -f(x_1, x_2), \quad (x_1, x_2) \in D = (0, 1) \times (0, 1),$$

$$u(x_1, x_2) = g(x_1, x_2), \quad (x_1, x_2) \in \partial D.$$

2. Конечно-разностная схема.

Равномерная сетка: $\Omega = \omega_{x_1} \times \omega_{x_2}$, $\omega_{x_\alpha} = \{x_\alpha = i_\alpha h_\alpha, i_\alpha = 0, \dots, N_\alpha, h_\alpha = 1/N_\alpha\}$, $\alpha = 1, 2$.

$$\frac{1}{h_1} \left\{ k_{1, i_1+1/2, i_2} \frac{y_{i_1+1, i_2} - y_{i_1, i_2}}{h_1} - k_{1, i_1-1/2, i_2} \frac{y_{i_1, i_2} - y_{i_1-1, i_2}}{h_1} \right\} +$$

$$\text{Разностная схема: } + \frac{1}{h_2} \left\{ k_{2, i_1, i_2+1/2} \frac{y_{i_1, i_2+1} - y_{i_1, i_2}}{h_2} - k_{2, i_1, i_2-1/2} \frac{y_{i_1, i_2} - y_{i_1, i_2-1}}{h_2} \right\} = -f_{i_1, i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1,$$

$$y_{i_1, i_2} = g_{i_1, i_2}, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

Эквивалентные уравнения:

$$\left(\frac{h_2}{h_1} (k_{1, i_1+1/2, i_2} + k_{1, i_1-1/2, i_2}) + \frac{h_1}{h_2} (k_{2, i_1, i_2+1/2} + k_{2, i_1, i_2-1/2}) \right) y_{i_1, i_2} - \frac{h_2}{h_1} k_{1, i_1+1/2, i_2} y_{i_1+1, i_2} - \frac{h_2}{h_1} k_{1, i_1-1/2, i_2} y_{i_1-1, i_2} -$$

$$- \frac{h_1}{h_2} k_{2, i_1, i_2+1/2} y_{i_1, i_2+1} - \frac{h_1}{h_2} k_{2, i_1, i_2-1/2} y_{i_1, i_2-1} = h_1 h_2 f_{i_1, i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1,$$

$$y_{i_1, i_2} = g_{i_1, i_2}, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

3. Решение уравнения Пуассона с помощью преобразования Фурье.

Пример 1: $k_1(x_1, x_2) \equiv 1, \quad k_2(x_1, x_2) \equiv 1, \quad f(x_1, x_2) = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2),$
 $g(x_1, x_2) \equiv 0, \quad u(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2).$

Эквивалентные уравнения:

$$2 \left(\frac{h_2}{h_1} + \frac{h_1}{h_2} \right) y_{i_1, i_2} - \frac{h_2}{h_1} y_{i_1+1, i_2} - \frac{h_2}{h_1} y_{i_1-1, i_2} - \frac{h_1}{h_2} y_{i_1, i_2+1} - \frac{h_1}{h_2} y_{i_1, i_2-1} = h_1 h_2 f_{i_1, i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1,$$

$$y_{i_1, i_2} = 0, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

Преобразование Фурье по обоим координатам:

$$y_{i_1, i_2} = \sum_{k_1=1, k_2=1}^{N_1-1, N_2-1} c_{k_1 k_2} 2 \sin(\pi k_1 x_{i_1}) \sin(\pi k_2 x_{i_2}), \quad f_{i_1, i_2} = \sum_{k_1=1, k_2=1}^{N_1-1, N_2-1} \varphi_{k_1 k_2} 2 \sin(\pi k_1 x_{i_1}) \sin(\pi k_2 x_{i_2}),$$

$$c_{k_1 k_2} = \frac{\varphi_{k_1 k_2}}{\lambda_{k_1 k_2}}, \quad \lambda_{k_1 k_2} = \frac{4}{h_1^2} \left(\sin \frac{\pi k_1 h_1}{2} \right)^2 + \frac{4}{h_2^2} \left(\sin \frac{\pi k_2 h_2}{2} \right)^2 = \frac{2}{h_1^2} (1 - \cos(\pi k_1 h_1)) + \frac{2}{h_2^2} (1 - \cos(\pi k_2 h_2)).$$

$$\varphi_{k_1 k_2} = \sum_{i_1=1, i_2=1}^{N_1-1, N_2-1} f(x_{i_1}, x_{i_2}) 2 \sin(\pi k_1 x_{i_1}) \sin(\pi k_2 x_{i_2}) h_1 h_2.$$

В итоге решение можно получить по следующим расчетным формулам:

$$y_{i_1, i_2} = \sum_{k_1=1, k_2=1}^{N_1-1, N_2-1} \gamma_{k_1 k_2} \sin \left(\frac{\pi k_1 i_1}{N_1} \right) \sin \left(\frac{\pi k_2 i_2}{N_2} \right),$$

$$\gamma_{k_1 k_2} = \frac{2 \varphi_{k_1 k_2}}{\lambda_{k_1 k_2}} = \frac{2}{N_1 N_2 \left[N_1^2 \left(1 - \cos \frac{\pi k_1}{N_1} \right) + N_2^2 \left(1 - \cos \frac{\pi k_2}{N_2} \right) \right]} \sum_{i_1=1, i_2=1}^{N_1-1, N_2-1} f(x_{i_1}, x_{i_2}) \sin \left(\frac{\pi k_1 i_1}{N_1} \right) \sin \left(\frac{\pi k_2 i_2}{N_2} \right).$$

Распараллеливание можно проводить по обоим индексам на решетке процессоров.

Если $N_\alpha = 2^{m_\alpha}$, то можно применить БПФ или оптимизированное ПФ.

$$\text{Оптимизация: } \sin\left(\frac{\pi k i}{N}\right) = \begin{cases} +\sin\left(\frac{\pi m}{N}\right), & \text{если } (2ki/N) \bmod 4 = 0, \quad m = (ki) \bmod (N/2), \\ +\sin\left(\frac{\pi m}{N}\right), & \text{если } (2ki/N) \bmod 4 = 1, \quad m = (N/2) - (ki) \bmod (N/2), \\ -\sin\left(\frac{\pi m}{N}\right), & \text{если } (2ki/N) \bmod 4 = 2, \quad m = (ki) \bmod (N/2), \\ -\sin\left(\frac{\pi m}{N}\right), & \text{если } (2ki/N) \bmod 4 = 3, \quad m = (N/2) - (ki) \bmod (N/2). \end{cases}$$

4. Решение уравнения Пуассона комбинированным методом ПФ-ЛГ.

$$k_1(x_1, x_2) \equiv k(x_1) = 1 + (x_1 - 0.5)^2, \quad k_2(x_1, x_2) \equiv 1,$$

Пример 2: $f(x_1, x_2) = -2(x_1 - 0.5)\pi \cos(\pi x_1) \sin(\pi x_2) + 2\pi^2 \sin(\pi x_1) \sin(\pi x_2),$

$$g(x_1, x_2) \equiv 0, \quad u(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2).$$

Эквивалентные уравнения:

$$\left(\frac{h_2}{h_1} (k_{i_1+1/2} + k_{i_1-1/2}) + 2 \frac{h_1}{h_2} \right) y_{i_1, i_2} - \frac{h_2}{h_1} k_{i_1+1/2} y_{i_1+1, i_2} - \frac{h_2}{h_1} k_{i_1-1/2} y_{i_1-1, i_2} -$$

$$- \frac{h_1}{h_2} y_{i_1, i_2+1} - \frac{h_1}{h_2} y_{i_1, i_2-1} = h_1 h_2 f_{i_1, i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1, \quad y_{i_1, i_2} = 0, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

Представление решения: $y_{i_1, i_2} = \sum_{k_2=1}^{N_2-1} c_{i_1 k_2} \sqrt{2} \sin(\pi k_2 x_{2, i_2}), \quad f_{i_1, i_2} = \sum_{k_2=1}^{N_2-1} \varphi_{i_1 k_2} \sqrt{2} \sin(\pi k_2 x_{2, i_2}), \quad i_\alpha = 0, \dots, N_\alpha.$

$$\frac{h_2}{h_1} (k_{i_1+1/2} + k_{i_1-1/2}) c_{i_1, k_2} - \frac{h_2}{h_1} k_{i_1+1/2} c_{i_1+1, k_2} - \frac{h_2}{h_1} k_{i_1-1/2} c_{i_1-1, k_2} + h_1 h_2 \lambda_{k_2} c_{i_1, k_2} = h_1 h_2 \varphi_{i_1, k_2},$$

Задача для коэффициентов: $i_\alpha = 1, \dots, N_\alpha - 1, \quad \alpha = 1, 2, \quad \lambda_{k_2} = \frac{4}{h_2^2} \left(\sin \frac{\pi k_2 h_2}{2} \right)^2 = \frac{2}{h_2^2} (1 - \cos(\pi k_2 h_2)),$

$$c_{0, k_2} = 0, \quad c_{N_1, k_2} = 0, \quad k_2 = 1, \dots, N_2 - 1.$$

Для любого k_2 коэффициенты c_{i_1, k_2} находим методом прогонки.

5. Реализация примеров.

Пример 1 (ex15a.c). Решение уравнения Пуассона методом Фурье на решетке процессоров.

Трансляция и результаты расчетов:

```
>mpicc -o ex15a.px -O2 -lm ex15a.c mycom.c mynet.c myio.c
>mpirun -np <1...16> -nolocal -machinefile hosts ex15a.px 128 128
np= 1 (1x1) n1=      128 n2=      128 dmax=1.499661e-32 time=7.579066e+01
np= 2 (1x2) n1=      128 n2=      128 dmax=1.499661e-32 time=3.824831e+01
np= 3 (1x3) n1=      128 n2=      128 dmax=1.499661e-32 time=2.548009e+01
np= 4 (2x2) n1=      128 n2=      128 dmax=1.499661e-32 time=1.936292e+01
np= 5 (1x5) n1=      128 n2=      128 dmax=1.499661e-32 time=1.556806e+01
np= 6 (2x3) n1=      128 n2=      128 dmax=1.499661e-32 time=1.332567e+01
np= 7 (1x7) n1=      128 n2=      128 dmax=1.499661e-32 time=1.161343e+01
np= 8 (2x4) n1=      128 n2=      128 dmax=1.499661e-32 time=9.934915e+00
np= 9 (3x3) n1=      128 n2=      128 dmax=1.499661e-32 time=9.327514e+00
np=10 (2x5) n1=      128 n2=      128 dmax=1.499661e-32 time=8.118166e+00
np=11 (1x11) n1=      128 n2=      128 dmax=1.499661e-32 time=7.635249e+00
np=12 (3x4) n1=      128 n2=      128 dmax=1.499661e-32 time=7.074516e+00
np=13 (1x13) n1=      128 n2=      128 dmax=1.499661e-32 time=6.776237e+00
np=14 (2x7) n1=      128 n2=      128 dmax=1.499661e-32 time=6.199461e+00
np=15 (3x5) n1=      128 n2=      128 dmax=1.499661e-32 time=5.503648e+00
np=16 (4x4) n1=      128 n2=      128 dmax=1.499661e-32 time=5.311533e+00
```

Задание 11 (ex15b.c). Реализовать оптимизированное ПФ.

Семинар 12. Решение уравнения Пуассона итерационными методами.

1. Постановка задачи.

$$\frac{\partial}{\partial x_1} \left(k_1(x_1, x_2) \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k_2(x_1, x_2) \frac{\partial u}{\partial x_2} \right) = -f(x_1, x_2), \quad (x_1, x_2) \in D = (0, 1) \times (0, 1),$$

$$u(x_1, x_2) = g(x_1, x_2), \quad (x_1, x_2) \in \partial D.$$

2. Конечно-разностная схема.

Равномерная сетка: $\Omega = \omega_{x_1} \times \omega_{x_2}$, $\omega_{x_\alpha} = \{x_\alpha = i_\alpha h_\alpha, i_\alpha = 0, \dots, N_\alpha, h_\alpha = 1/N_\alpha\}$, $\alpha = 1, 2$.

$$\frac{1}{h_1} \left\{ k_{1,i_1+1/2,i_2} \frac{y_{i_1+1,i_2} - y_{i_1,i_2}}{h_1} - k_{1,i_1-1/2,i_2} \frac{y_{i_1,i_2} - y_{i_1-1,i_2}}{h_1} \right\} +$$

$$\text{Разностная схема: } + \frac{1}{h_2} \left\{ k_{2,i_1,i_2+1/2} \frac{y_{i_1,i_2+1} - y_{i_1,i_2}}{h_2} - k_{2,i_1,i_2-1/2} \frac{y_{i_1,i_2} - y_{i_1,i_2-1}}{h_2} \right\} = -f_{i_1,i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1,$$

$$y_{i_1,i_2} = g_{i_1,i_2}, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

Эквивалентные уравнения:

$$\left(\frac{h_2}{h_1} (k_{1,i_1+1/2,i_2} + k_{1,i_1-1/2,i_2}) + \frac{h_1}{h_2} (k_{2,i_1,i_2+1/2} + k_{2,i_1,i_2-1/2}) \right) y_{i_1,i_2} - \frac{h_2}{h_1} k_{1,i_1+1/2,i_2} y_{i_1+1,i_2} - \frac{h_2}{h_1} k_{1,i_1-1/2,i_2} y_{i_1-1,i_2} -$$

$$- \frac{h_1}{h_2} k_{2,i_1,i_2+1/2} y_{i_1,i_2+1} - \frac{h_1}{h_2} k_{2,i_1,i_2-1/2} y_{i_1,i_2-1} = h_1 h_2 f_{i_1,i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 1,$$

$$y_{i_1,i_2} = g_{i_1,i_2}, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

3. Решение уравнения Пуассона итерационными методами.

3.1. Метод простой итерации.

$$y_{i_1,i_2}^{s+1} = y_{i_1,i_2}^s - \tau \left(\frac{h_2}{h_1} (k_{1,i_1+1/2,i_2} + k_{1,i_1-1/2,i_2}) + \frac{h_1}{h_2} (k_{2,i_1,i_2+1/2} + k_{2,i_1,i_2-1/2}) \right) y_{i_1,i_2}^s +$$

$$\tau \frac{h_2}{h_1} k_{1,i_1+1/2,i_2} y_{i_1+1,i_2}^s + \tau \frac{h_2}{h_1} k_{1,i_1-1/2,i_2} y_{i_1-1,i_2}^s + \tau \frac{h_1}{h_2} k_{2,i_1,i_2+1/2} y_{i_1,i_2+1}^s + \tau \frac{h_1}{h_2} k_{2,i_1,i_2-1/2} y_{i_1,i_2-1}^s + \tau h_1 h_2 f_{i_1,i_2}, \quad s = 0, 1, \dots$$

$$\text{Скорость сходимости } \|r^n\| \leq q^n \|r^0\|, \quad q = \max_{\lambda_{\min} \leq \lambda \leq \lambda_{\max}} |1 - \tau \lambda|.$$

$$\text{Выбор параметра: } 0 < \tau < \frac{2}{\lambda_{\max}}, \quad \tau = \tau_0 = \frac{2}{\lambda_{\max} + \lambda_{\min}}, \quad \lambda_{\min} = \min \lambda(A), \quad \lambda_{\max} = \max \lambda(A).$$

$$\text{Для достижения точности } \varepsilon \text{ потребуется } n(\varepsilon) = 1 + \left\lceil \ln \varepsilon^{-1} / \ln q^{-1} \right\rceil \text{ итераций, } q = 1 - \tau \lambda_{\min}.$$

$$\text{Для оптимального параметра получаем } q = q_0 = 1 - \tau_0 \lambda_{\min} = 1 - \frac{2}{1 + \mu} = \frac{\mu - 1}{\mu + 1}, \quad n(\varepsilon) = 1 + \left\lceil \ln \varepsilon^{-1} / \ln \frac{\mu + 1}{\mu - 1} \right\rceil.$$

Здесь $\mu = \frac{\lambda_{\max}}{\lambda_{\min}}$ – число обусловленности.

3.2. Ускорение по Чебышову.

Возьмем нестационарный метод, а именно различные параметры $\{\tau_s\}$. Тогда скорость сходимости выражается

формулами: $\|r^n\| \leq q \|r^0\|$, $q = \max_{\lambda_{\min} \leq \lambda \leq \lambda_{\max}} \left| \prod_{s=0}^{n-1} (1 - \tau_{s+1} \lambda) \right|$. Если выбрать Чебышевский набор итерационных параметров:

$$\tau = \tau_{s+1} = \left[\frac{\lambda_{\max} + \lambda_{\min}}{2} + \frac{\lambda_{\max} - \lambda_{\min}}{2} \cos \frac{\pi(2(s+1)-1)}{2n} \right]^{-1} = \frac{\tau_0}{1 + \frac{\mu-1}{\mu+1} \cos \frac{\pi(2(s+1)-1)}{2n}}, \quad s = 0, \dots, n-1, \text{ то получим}$$

максимальную скорость сходимости. Здесь n – длина итерационной серии. Для достижения точности ε необходимо сделать $n(\varepsilon) = 1 + \left\lceil 0.5 \sqrt{\mu} \ln \varepsilon^{-1} \right\rceil$. Для реализации метода важен порядок использования параметров. В случае, когда

$n = 2^m$, этот порядок вычисляется относительно просто:

$$\tau_{2s} = \tau_{i(s)}, \quad \tau_{2s+1} = \tau_{n-1-i(s)}, \quad s = 0, \dots, n/2 - 1, \quad i(s) = \dots$$

3.3. Трехслойный метод Чебышева.

$$y^1 = (E - \tau_0 A) y^0 + \tau_0 f, \quad y^{s+1} = \alpha_{s+1} (E - \tau_0 A) y^s + (1 - \alpha_{s+1}) y^{s-1} + \tau_0 \alpha_{s+1} f, \quad s = 1, 2, \dots,$$

$$\alpha_1 = 2, \quad \alpha_{s+1} = \frac{4}{4 - \rho^2 \alpha_s}, \quad \rho = \frac{\mu - 1}{\mu + 1}.$$

$$\text{Скорость сходимости } \|r^n\| \leq \frac{2q^n}{1 + q^{2n}} \|r^0\|, \quad q = \frac{\sqrt{\mu} - 1}{\sqrt{\mu} + 1}.$$

3.4. Метод Якоби.

$$\text{Каноническая двухслойная схема итераций: } B_{s+1} \frac{y - y^s}{\tau_{s+1}} + A y^s = f, \quad s = 0, 1, \dots$$

Метод Якоби: B_{s+1} – диагональные матрицы. Обычно рассматривают стационарный метод с $B = D_A$.

Для нашей задачи имеем:

$$y_{i_1, i_2}^{s+1} = (1 - \tau) y_{i_1, i_2}^s + \tau \left(\frac{\hbar_2}{h_1} (k_{1, i_1 + 1/2, i_2} + k_{1, i_1 - 1/2, i_2}) + \frac{\hbar_1}{h_2} (k_{2, i_1, i_2 + 1/2} + k_{2, i_1, i_2 - 1/2}) \right)^{-1} \times$$

$$\times \left(\frac{\hbar_2}{h_1} k_{1, i_1 + 1/2, i_2} y_{i_1 + 1, i_2}^s + \frac{\hbar_2}{h_1} k_{1, i_1 - 1/2, i_2} y_{i_1 - 1, i_2}^s + \frac{\hbar_1}{h_2} k_{2, i_1, i_2 + 1/2} y_{i_1, i_2 + 1}^s + \frac{\hbar_1}{h_2} k_{2, i_1, i_2 - 1/2} y_{i_1, i_2 - 1}^s + \hbar_1 \hbar_2 f_{i_1, i_2} \right), \quad s = 0, 1, \dots$$

Выбор параметров: либо один оптимальный $\tau = \tau_0 = \frac{2}{\min \lambda(D^{-1}A) + \max \lambda(D^{-1}A)}$, либо чебышевский набор.

3.5. Метод Зейделя (Некрасова).

В методе Зейделя $B = D_A + A_-$:

$$y_{i_1, i_2}^{s+1} = (1 - \tau) y_{i_1, i_2}^s + \tau \left(\frac{\hbar_2}{h_1} (k_{1, i_1 + 1/2, i_2} + k_{1, i_1 - 1/2, i_2}) + \frac{\hbar_1}{h_2} (k_{2, i_1, i_2 + 1/2} + k_{2, i_1, i_2 - 1/2}) \right)^{-1} \times$$

$$\times \left(\frac{\hbar_2}{h_1} k_{1, i_1 + 1/2, i_2} y_{i_1 + 1, i_2}^s + \frac{\hbar_2}{h_1} k_{1, i_1 - 1/2, i_2} y_{i_1 - 1, i_2}^{s+1} + \frac{\hbar_1}{h_2} k_{2, i_1, i_2 + 1/2} y_{i_1, i_2 + 1}^s + \frac{\hbar_1}{h_2} k_{2, i_1, i_2 - 1/2} y_{i_1, i_2 - 1}^{s+1} + \hbar_1 \hbar_2 f_{i_1, i_2} \right), \quad s = 0, 1, \dots$$

Выбор параметров τ : можно выбрать также как и выше.

4. Параллельная реализация на примерах.

Тестовая задача: $k_\alpha \equiv 1$, $f(x_1, x_2) = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2)$, $g(x_1, x_2) \equiv 0$, $u(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2)$.

4.1. Метод Якоби.

$$y_{i_1, i_2}^{s+1} = (1 - \tau) y_{i_1, i_2}^s + \frac{\tau}{2} \left(\frac{h_2}{h_1} + \frac{h_1}{h_2} \right)^{-1} \left(\frac{h_2}{h_1} \left(y_{i_1 + 1, i_2}^s + y_{i_1 - 1, i_2}^s \right) + \frac{h_1}{h_2} \left(y_{i_1, i_2 + 1}^s + y_{i_1, i_2 - 1}^s \right) + h_1 h_2 f_{i_1, i_2} \right), \quad s = 0, 1, \dots,$$

$$y_{i_1, i_2}^{s+1} = 0, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

$$\lambda_{\min} = \lambda_{\min}(D^{-1}A) = \frac{1}{2} \left(\frac{h_2}{h_1} + \frac{h_1}{h_2} \right)^{-1} h_1 h_2 \left(\frac{4}{h_1^2} \sin^2 \frac{\pi h_1}{2} + \frac{4}{h_2^2} \sin^2 \frac{\pi h_2}{2} \right) \approx \frac{\pi^2 h_1^2 h_2^2}{h_1^2 + h_2^2},$$

$$\text{Выбор оптимального параметра: } \lambda_{\max} = \lambda_{\max}(D^{-1}A) = \frac{1}{2} \left(\frac{h_2}{h_1} + \frac{h_1}{h_2} \right)^{-1} h_1 h_2 \left(\frac{4}{h_1^2} \cos^2 \frac{\pi h_1}{2} + \frac{4}{h_2^2} \cos^2 \frac{\pi h_2}{2} \right) \approx 2,$$

$$\tau = \tau_0 = \frac{2}{\lambda_{\min} + \lambda_{\max}} \approx \frac{2}{\frac{\pi^2 h_1^2 h_2^2}{h_1^2 + h_2^2} + 2}.$$

Если шаги одинаковы, то $\tau_0 \approx \frac{1}{1 + 0.25\pi^2 h^2}$.

Параллельная реализация выполняется на решетке процессоров.

4.2. Метод Зейделя.

$$y_{i_1, i_2}^{s+1} = (1 - \tau) y_{i_1, i_2}^s + \frac{\tau}{2} \left(\frac{h_2}{h_1} + \frac{h_1}{h_2} \right)^{-1} \left(\frac{h_2}{h_1} \left(y_{i_1 + 1, i_2}^s + y_{i_1 - 1, i_2}^{s+1} \right) + \frac{h_1}{h_2} \left(y_{i_1, i_2 + 1}^s + y_{i_1, i_2 - 1}^{s+1} \right) + h_1 h_2 f_{i_1, i_2} \right), \quad s = 0, 1, \dots,$$

$$y_{i_1, i_2}^{s+1} = 0, \quad i_\alpha = 0, N_\alpha, \quad \alpha = 1, 2.$$

4.3. Реализация примеров.

Пример 1 (ex16a.c). Метод Якоби с ЧНП.

Трансляция и результаты расчетов:

```
>mpicc -o ex16a.px -O2 -lm ex16a.c mycom.c mynet.c myio.c myrand.c
```

```
>mpirun -np <1...24> -nolocal -machinefile hosts ex16a.px a <2...1024> <2...1024>
```

np= 1	(1x1)	n1=	2	n2=	2	it=	0	rka=	0.000000e+00	dka=	2.337006e-01	time=	5.000000e-06
np= 1	(1x1)	n1=	4	n2=	4	it=	4	rka=	6.664829e-03	dka=	3.027414e-02	time=	1.300000e-05
np= 1	(1x1)	n1=	8	n2=	8	it=	8	rka=	5.405428e-04	dka=	5.849598e-03	time=	5.000000e-05
np= 1	(1x1)	n1=	16	n2=	16	it=	16	rka=	3.580093e-05	dka=	1.355761e-03	time=	2.450000e-04
np= 1	(1x1)	n1=	32	n2=	32	it=	32	rka=	2.269387e-06	dka=	3.322884e-04	time=	1.463000e-03
np= 1	(1x1)	n1=	64	n2=	64	it=	64	rka=	1.423352e-07	dka=	8.265655e-05	time=	1.002400e-02
np= 1	(1x1)	n1=	128	n2=	128	it=	128	rka=	8.903733e-09	dka=	2.063817e-05	time=	7.370100e-02
np= 1	(1x1)	n1=	256	n2=	256	it=	256	rka=	5.570938e-10	dka=	5.157920e-06	time=	4.412720e-01

np= 1	(1x1)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	4.359555e+00
np= 2	(1x2)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	2.509570e+00
np= 3	(1x3)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.853683e+00
np= 4	(2x2)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.575268e+00
np= 5	(1x5)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.697655e+00
np= 6	(2x3)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.410437e+00
np= 7	(1x7)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.550687e+00
np= 8	(2x4)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.259869e+00
np= 9	(3x3)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.630450e+00
np=10	(2x5)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.092398e+00
np=11	(1x11)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.283869e+00
np=12	(3x4)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.198221e+00
np=15	(3x5)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.473445e+00
np=16	(4x4)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.519445e+00
np=20	(4x5)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.520552e+00
np=24	(4x6)	n1=	512	n2=	512	it=	512	rka=	3.844114e-11	dka=	1.289379e-06	time=	1.524282e+00

np= 1	(1x1)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	3.397578e+01
np= 2	(1x2)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	1.825585e+01
np= 3	(1x3)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	1.316699e+01
np= 4	(2x2)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	1.053883e+01
np= 5	(1x5)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	9.632603e+00
np= 6	(2x3)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	7.909441e+00
np= 7	(1x7)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	7.578117e+00
np= 8	(2x4)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.534893e+00
np= 9	(3x3)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	5.893631e+00
np=10	(2x5)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.094514e+00
np=11	(1x11)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.221716e+00
np=12	(3x4)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	5.829750e+00
np=13	(1x13)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	7.089344e+00
np=14	(2x7)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.674810e+00
np=15	(3x5)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.097073e+00
np=16	(4x4)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.121966e+00
np=20	(4x5)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	6.039219e+00
np=24	(4x6)	n1=	1024	n2=	1024	it=	1024	rka=	2.391924e-11	dka=	3.223407e-07	time=	5.609448e+00

Задание 12 (ex16b.c): Реализовать метод Зейделя с оптимальным параметром.

Семинар 13. Решение волнового уравнения в двумерном случае.

1. Постановка задачи.

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x_1} \left(k(x_1, x_2) \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k(x_1, x_2) \frac{\partial u}{\partial x_2} \right) + f(x_1, x_2, t), \quad (x_1, x_2) \in D = (a_1, b_1) \times (a_2, b_2),$$

$$u(x_1, x_2, 0) = \varphi(x_1, x_2), \quad u_t(x_1, x_2, 0) = \psi(x_1, x_2),$$

$$u(a_1, x_2, t) = g_{11}(t), \quad u(b_1, x_2, t) = g_{12}(t), \quad u(x_1, a_2, t) = g_{21}(t), \quad u(x_1, b_2, t) = g_{22}(t),$$

$$k(x_1, x_2) = \begin{cases} k_1, & (x_1, x_2) \in D_0, \\ k_2, & (x_1, x_2) \notin D_0, \end{cases} \quad D_0 = [x_{11}, x_{12}] \times [x_{21}, x_{22}],$$

$$f(x_1, x_2, t) = \sum_{m=1}^2 Q_m \exp \left[-(x_1 - x_{1,m0})^2 / r_m^2 - (x_2 - x_{2,m0})^2 / r_m^2 \right] \sin(\omega_m t),$$

$$\varphi(x_1, x_2) = 0, \quad \psi(x_1, x_2) = 0, \quad g_{11}(t) = 0, \quad g_{12}(t) = 0, \quad g_{21}(t) = u_0, \quad g_{22}(t) = 0.$$

2. Численные алгоритмы.

$$\text{МКР на равномерной сетке } \Omega = \omega_{x_1} \times \omega_{x_2} \times \omega_t, \quad \omega_{x_\alpha} = \left\{ x_{\alpha, i_\alpha} = i_\alpha h_\alpha, i_\alpha = 0, \dots, N_\alpha, h_\alpha = \frac{b_\alpha - a_\alpha}{N_\alpha} \right\}, \quad \alpha = 1, 2,$$

$$\omega_t = \left\{ t_j = j\tau, j = 0, \dots, N_t, \tau = \frac{t_{\max}}{N_t} \right\}.$$

Схема с весами:

$$\begin{aligned} & \frac{y_{i_1 i_2}^{j+1} - 2y_{i_1 i_2}^j + y_{i_1 i_2}^{j-1}}{\tau^2} = \\ & = \sigma \frac{1}{h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^{j+1} - y_{i_1+1, i_2}^{j-1}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1-1, i_2}^{j+1} - y_{i_1-1, i_2}^{j-1}}{h_1} \right\} + \sigma \frac{1}{h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^{j+1} - y_{i_1, i_2+1}^{j-1}}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1, i_2-1}^{j+1} - y_{i_1, i_2-1}^{j-1}}{h_2} \right\} + \\ & + (1-2\sigma) \frac{1}{h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^j - y_{i_1+1, i_2}^{j-2}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1-1, i_2}^j - y_{i_1-1, i_2}^{j-2}}{h_1} \right\} + (1-2\sigma) \frac{1}{h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^j - y_{i_1, i_2+1}^{j-2}}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1, i_2-1}^j - y_{i_1, i_2-1}^{j-2}}{h_2} \right\} + \\ & + \sigma \frac{1}{h_1} \left\{ k_{i_1+1/2, i_2} \frac{y_{i_1+1, i_2}^{j-1} - y_{i_1+1, i_2}^{j-3}}{h_1} - k_{i_1-1/2, i_2} \frac{y_{i_1-1, i_2}^{j-1} - y_{i_1-1, i_2}^{j-3}}{h_1} \right\} + (1-\sigma) \frac{1}{h_2} \left\{ k_{i_1, i_2+1/2} \frac{y_{i_1, i_2+1}^{j-1} - y_{i_1, i_2+1}^{j-3}}{h_2} - k_{i_1, i_2-1/2} \frac{y_{i_1, i_2-1}^{j-1} - y_{i_1, i_2-1}^{j-3}}{h_2} \right\} + \\ & + \sigma f_{i_1 i_2}^{j+1} + (1-2\sigma) f_{i_1 i_2}^j + \sigma f_{i_1 i_2}^{j-1}, \quad 0 < i_\alpha < N_\alpha, \quad \alpha = 1, 2, \quad 0 \leq j < N_t, \\ & y_{i_1 i_2}^0 = \varphi_{i_1 i_2}, \quad y_{i_1 i_2}^1 = y_{i_1 i_2}^0 + \tau \psi_{i_1 i_2}, \quad y_{0, i_2}^j = g_{11}(t_j), \quad y_{N_1, i_2}^j = g_{12}(t_j), \quad y_{i_1, 0}^j = g_{21}(t_j), \quad y_{i_1, N_2}^j = g_{22}(t_j), \\ & k_{i_1 \pm 1/2, i_2} = \frac{2k_{i_1 i_2} k_{i_1 \pm 1, i_2}}{k_{i_1 i_2} + k_{i_1 \pm 1, i_2}}, \quad k_{i_1, i_2 \pm 1/2} = \frac{2k_{i_1 i_2} k_{i_1, i_2 \pm 1}}{k_{i_1 i_2} + k_{i_1, i_2 \pm 1}}, \quad k_{i_1 i_2} = k(x_{1, i_1}, x_{2, i_2}), \quad f_{i_1 i_2}^j = f(x_{1, i_1}, x_{2, i_2}, t_j), \quad h_\alpha = \begin{cases} 0.5h_\alpha, & i = 0, N_\alpha, \\ h_\alpha, & 1 < i < N_\alpha. \end{cases} \end{aligned}$$

Расчетные формулы для схемы с весами:

$$\begin{aligned} & y_{i_1 i_2}^{j+1} - \sigma \left[B_{i_1 i_2} (y_{i_1+1, i_2}^{j+1} - y_{i_1+1, i_2}^{j-1}) - A_{i_1 i_2} (y_{i_1+1, i_2}^{j+1} - y_{i_1-1, i_2}^{j+1}) + B_{2i_1 i_2} (y_{i_1, i_2+1}^{j+1} - y_{i_1, i_2+1}^{j-1}) - A_{2i_1 i_2} (y_{i_1, i_2+1}^{j+1} - y_{i_1, i_2-1}^{j+1}) \right] = \\ & = 2y_{i_1 i_2}^j - y_{i_1 i_2}^{j-1} + (1-2\sigma) \left[B_{i_1 i_2} (y_{i_1+1, i_2}^j - y_{i_1+1, i_2}^{j-2}) - A_{i_1 i_2} (y_{i_1+1, i_2}^j - y_{i_1-1, i_2}^j) + B_{2i_1 i_2} (y_{i_1, i_2+1}^j - y_{i_1, i_2+1}^{j-2}) - A_{2i_1 i_2} (y_{i_1, i_2+1}^j - y_{i_1, i_2-1}^j) \right] + \\ & + \sigma \left[B_{i_1 i_2} (y_{i_1+1, i_2}^{j-1} - y_{i_1+1, i_2}^{j-3}) - A_{i_1 i_2} (y_{i_1+1, i_2}^{j-1} - y_{i_1-1, i_2}^{j-1}) + B_{2i_1 i_2} (y_{i_1, i_2+1}^{j-1} - y_{i_1, i_2+1}^{j-3}) - A_{2i_1 i_2} (y_{i_1, i_2+1}^{j-1} - y_{i_1, i_2-1}^{j-1}) \right] + \\ & + \sigma \tau^2 f_{i_1 i_2}^{j+1} + (1-2\sigma) \tau^2 f_{i_1 i_2}^j + \sigma \tau^2 f_{i_1 i_2}^{j-1}, \end{aligned}$$

$$A_{i_1 i_2} = \gamma_1 k_{i_1-1/2, i_2}, \quad B_{i_1 i_2} = \gamma_1 k_{i_1+1/2, i_2}, \quad A_{2i_1 i_2} = \gamma_2 k_{i_1, i_2-1/2}, \quad B_{2i_1 i_2} = \gamma_2 k_{i_1, i_2+1/2}, \quad \gamma_1 = \frac{\tau^2}{h_1 h_1}, \quad \gamma_2 = \frac{\tau^2}{h_2 h_2}.$$

В этом случае можно ограничиться явной схемой:

$$\begin{aligned} & y_{i_1 i_2}^{j+1} = 2y_{i_1 i_2}^j - y_{i_1 i_2}^{j-1} + \left[B_{i_1 i_2} (y_{i_1+1, i_2}^j - y_{i_1+1, i_2}^{j-2}) - A_{i_1 i_2} (y_{i_1+1, i_2}^j - y_{i_1-1, i_2}^j) + B_{2i_1 i_2} (y_{i_1, i_2+1}^j - y_{i_1, i_2+1}^{j-2}) - A_{2i_1 i_2} (y_{i_1, i_2+1}^j - y_{i_1, i_2-1}^j) \right] + \tau^2 f_{i_1 i_2}^j, \\ & y_{i_1 i_2}^0 = 0, \quad y_{i_1 i_2}^1 = 0, \quad y_{0, i_2}^j = 0, \quad y_{N_1, i_2}^j = 0, \quad y_{i_1, 0}^j = 0, \quad y_{i_1, N_2}^j = 0. \end{aligned}$$

3. Параллельная реализация.

См. уравнение теплопроводности.

4. Реализация примеров.

Пример 1 (ex17a.c). Реализация явной схемы на решетке процессоров.

Трансляция и результаты расчетов:

```
>mpicc -o ex17a.px -O2 -lm ex17a.c mycom.c mynet.c myio.c
```

```
>mpirun -np <1...16> -nolocal -machinefile hosts ex17a.px
```

```
1x1 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=8.317579e+01
2x2 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=2.967879e+01
3x3 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.745440e+01
4x4 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.407800e+01
5x5 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.373613e+01
```

Задание 13. Реализовать пример ex17a.c с помощью MPI-процессов и трэдов.

