

Распараллеливание решения линейной обратной задачи гравиметрии на МВС-1000 и графических процессорах *

Е.Н. Акимова, Д.В. Белоусов

Для решения линейной обратной задачи гравиметрии о восстановлении переменной плотности в слое численно реализованы на многопроцессорном вычислительном комплексе МВС-1000 и графических процессорах на видеокартах NVIDIA регулярные параллельные итерационные алгоритмы. Для решения модельной задачи гравиметрии и задачи с реальными данными проведено сравнение времени счета параллельных алгоритмов на видеоускорителях GeForce и МВС-1000 с анализом эффективности и ускорения.

1. Введение

Рассматривается линейная обратная задача гравиметрии о восстановлении переменной плотности в горизонтальном или криволинейном слое. После предварительной обработки гравитационных данных по методике, предложенной П.С. Мартышко и И.Л. Пруткиным [1], задача нахождения неизвестной плотности сводится к решению линейного двумерного интегрального уравнения Фредгольма первого рода [2].

Задача гравиметрии являются существенно некорректной задачей, решение которой обладает сильной чувствительностью к погрешности правой части, полученной в результате измерений и предварительной обработки геофизических данных. При разработке методов решения задач используются идеи итеративной регуляризации [3].

Для решения линейной обратной задачи гравиметрии о восстановлении переменной плотности в слое численно реализованы на многопроцессорном вычислительном комплексе МВС-1000 и графических процессорах на видеокартах NVIDIA регулярные параллельные итерационные алгоритмы.

Для решения модельной задачи гравиметрии и задачи с реальными данными проведено сравнение времени счета параллельных алгоритмов на видеоускорителях GeForce GTX 285, GeForce GTX 260 и МВС-1000/64 с анализом эффективности и ускорения.

2. Параллельные алгоритмы решения линейной обратной задачи гравиметрии о восстановлении плотности в слое

Одной из важнейших моделей строения земной коры является модель горизонтальной слоистой среды.

Рассматривается задача о нахождении переменной плотности $\sigma = \sigma(x, y)$ в горизонтальном слое $\Pi = \{(x, y, z) \in R^3 : (x, y) \in D, H_1 \leq z \leq H_2\}$, где H_1, H_2 — константы, либо криволинейном слое $\Pi_1 = \{(x, y, z) \in R^3 : (x, y) \in D, H_1(x, y) \leq z \leq H_2(x, y)\}$ по гравитационным данным, измеренным на площади $D = \{(x, y) \in R^2 : a \leq x \leq b, c \leq y \leq d\}$ земной поверхности. Используется априорная информация об отсутствии аномалий плотности вне слоя с криволинейными границами $H_1 = H_1(x, y)$ и $H_2 = H_2(x, y)$ такими, что $H_1 < H_2 \forall (x, y)$, и выполняется условие $H_i(x, y) \xrightarrow[x \rightarrow \pm\infty]{y \rightarrow \pm\infty} h_i = \text{const}$.

При этом предполагается, что распределение плотности $\sigma = \sigma(x, y)$ внутри слоя не зависит от z (ось z направлена вниз) (рис. 1).

*Работа поддержана грантом РФФИ № 09-01-00053, Междисциплинарным проектом УрО РАН, Программой Президиума РАН.

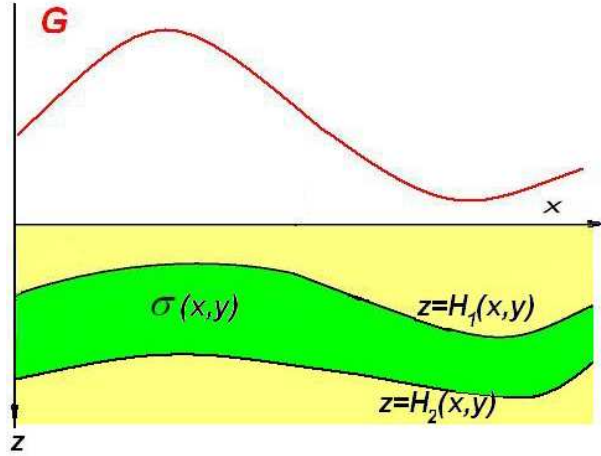


Рис. 1. Задача о нахождении плотности в слое

Задача нахождения неизвестной плотности $\sigma(x, y)$ сводится к решению линейного двумерного интегрального уравнения Фредгольма первого рода

$$A\sigma \equiv f \iint_{ac}^{bd} \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + H_1^2(x', y')]^{1/2}} - \frac{1}{[(x-x')^2 + (y-y')^2 + H_2^2(x', y')]^{1/2}} \right\} \sigma(x', y') dx' dy' = \Delta g(x, y), \quad (1)$$

где f — гравитационная постоянная, $\Delta g(x, y)$ — гравитационный эффект, порождаемый источниками в горизонтальном или криволинейном слое.

После дискретизации уравнения на сетке, где задана $\Delta g(x, y)$, и аппроксимации интегрального оператора по квадратурным формулам задача (1) сводится к решению системы линейных алгебраических уравнений (СЛАУ) либо с симметричной положительно определенной матрицей (горизонтальный слой), либо с несимметричной матрицей (криволинейный слой). Так как уравнение (1) относится к классу некорректно поставленных задач, то СЛАУ, возникающее в результате дискретизации уравнения, является плохо обусловленной и преобразуется к виду

$$(A + \alpha E)z = b, \quad (2)$$

где α — параметр регуляризации.

В случае криволинейного слоя исходная матрица СЛАУ несимметрична, поэтому эта система предварительно преобразуется к виду

$$(A^T A + \alpha' E)z = A^T b, \quad (3)$$

где A^T — транспонированная матрица, α' — параметр регуляризации.

В.В. Васин предложил методику использования нескольких регулярных методов решения некорректных задач для сравнения результатов.

Для решения уравнений (2) и (3) используются регулярные итерационные методы градиентного типа [4]: метод минимальных невязок, метод наискорейшего спуска, метод минимальной ошибки и метод простой итерации (МПИ) в виде

$$z^{k+1} = z^k - \frac{1}{\lambda_{max}} [(A + \alpha E)z^k - b], \quad (4)$$

где λ_{max} — максимальное собственное значение матрицы $A + \alpha E$ (симметричный случай).

Условием останова итерационных процессов является следующее: $\frac{\|Az^k - b\|}{\|b\|} < \varepsilon$.

В работах [5]– [6] предложены и численно реализованы регулярные параллельные итерационные алгоритмы решения линейной обратной задачи гравиметрии (1) с помощью библиотеки MPI [7] на языке Фортран на многопроцессорном вычислительном комплексе МВС–1000 — российском массивно-параллельном суперкомпьютере кластерного типа с распределенной памятью, установленном в Институте математики и механики УрО РАН.

Алгоритмы были реализованы на следующих вычислителях:

1. МВС–1000/17ЕК (UM32), состоящем из 16 2-х процессорных модулей Xeon 2.4 ГГц, интерфейса GigabitEthernet и 68 Гбайт оперативной памяти;
2. МВС–1000/64 (UM64), состоящем из 14 2-х процессорных 2-х ядерных модулей AMD Opteron 64 bit (2.6 ГГц), интерфейса GbitEthernet и 112 Гб оперативной памяти.

Распараллеливание итерационных методов градиентного типа основано на разбиении матрицы A горизонтальными полосами на m блоков, а вектора решения z и вектора правой части b СЛАУ на m частей так, что $n = m \times L$, где n — размерность системы уравнений, m — число процессоров. На каждой итерации каждый из m процессоров вычисляет свою часть вектора решения. В случае умножения матрицы A на вектор z каждый из m процессоров умножает свою часть строк матрицы A на вектор z . В случае матричного умножения $A^T A$ каждый из m процессоров умножает свою часть строк транспонированной матрицы A^T на всю матрицу A . Host-процессор отвечает за пересылки данных и также вычисляет свою часть вектора решения (рис. 2). Для метода простой итерации (4) максимальное собственное значение λ_{max} матрицы $A + \alpha E$ находится с помощью степенного метода с использованием параллельного алгоритма умножения матрицы на вектор.

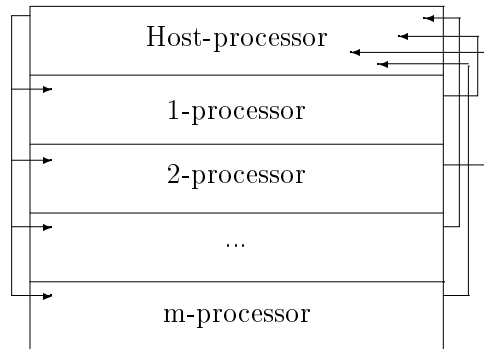


Рис. 2. Схема распределения данных по процессорам для итерационных методов

В предыдущих работах для решения линейной задачи гравиметрии проведен анализ эффективности и ускорения параллельных алгоритмов

$$S_m = T_1/T_m, \quad E_m = S_m/m,$$

где T_m — время выполнения параллельного алгоритма на МВС–1000 с числом процессоров m ($m > 1$), T_1 — время выполнения последовательного алгоритма на одном процессоре. T_m представляет собой совокупность чистого времени счета и накладных расходов на межпроцессорные обмены $T_m = T_c + T_o$. Число процессоров m соответствует упомянутому разбиению векторов на m частей и разбиению исходной области на m подобластей.

В общем случае эффективность распараллеливания меняется в пределах $0 < E_m < 1$. В идеальном случае при равномерной и сбалансированной загрузке процессоров и минимальном времени обменов между ними E_m близко к единице, но при решении практических задач она уменьшается за счет накладных расходов.

При решении задачи о восстановлении плотности в слое на МВС–1000 с помощью параллельных алгоритмов матрица СЛАУ большой размерности формируется и хранится в памяти каждого процессора по частям, что дает эффективность распараллеливания $E_m > 1$.

3. Результаты численных экспериментов на МВС—1000

Задача 1. На многопроцессорном вычислительном комплексе МВС—1000/64 решена модельная задача о восстановлении плотности в горизонтальном слое между глубинами $H_1 = 1$ км и $H_2 = 1.5$ км для области S_1 , имеющей размеры 10×10 км². Шаги сетки: $\Delta x \approx \Delta y \approx 0.05$ км. Гравитационная постоянная $f = 6.67 \cdot 10^{-8}$ см³/г·с².

После дискретизации исходного уравнения на сетке, где задана функция $\Delta g(x, y)$, и аппроксимации интегрального оператора по квадратурным формулам задача (1) сводится к системе линейных алгебраических уравнений с симметричной матрицей 40000×40000 .

Для решения задачи использовался параллельный итеративно регуляризованный метод простой итерации с параметром регуляризации $\alpha = 0.0005$.

На рис. 3 изображено исходное аномальное гравитационное поле для области S_1 .

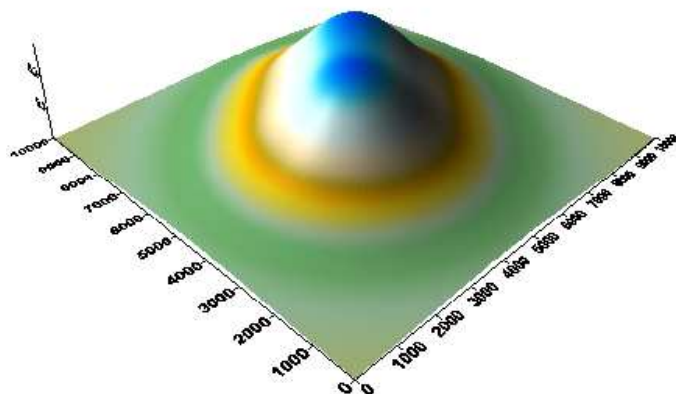


Рис. 3. Аномальное гравитационное поле $\Delta g(x, y)$ для области S_1

На рис. 4 изображены линии уровня и распределение плотности в слое, восстановленной по выделенному аномальному полю для области S_1 .

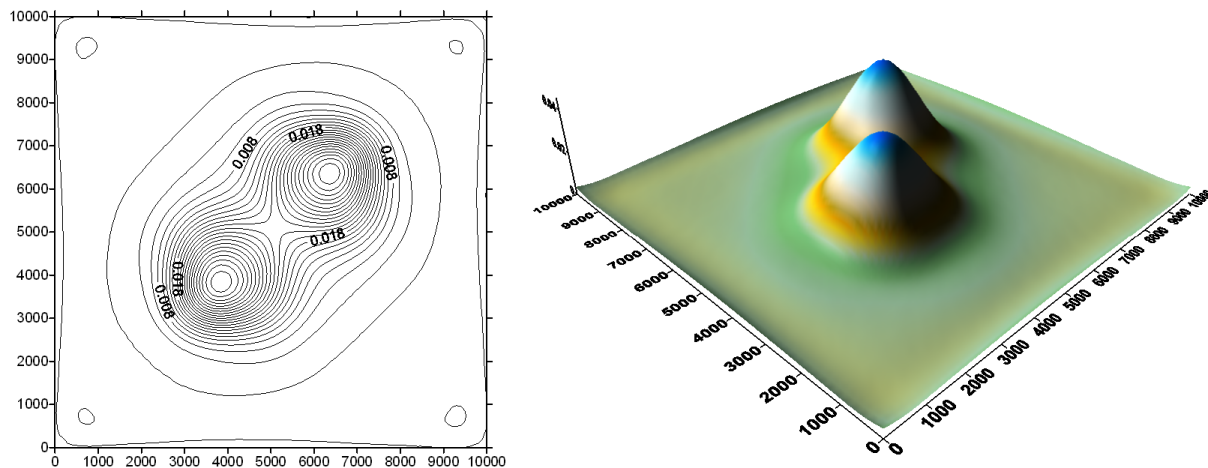


Рис. 4. Линии уровня и распределение восстановленной плотности в слое для области S_1

Задача 2. На многопроцессорном вычислительном комплексе МВС—1000/64 решена задача с реальными данными о восстановлении плотности в горизонтальном слое между глубинами $H_1 = 10$ км и $H_2 = 20$ км для области S_2 , имеющей размеры 120×220 км². Шаги сетки: $\Delta x \approx 0.6$ км, $\Delta y \approx 1.1$ км. Гравитационная постоянная $f = 6.67 \cdot 10^{-8}$ см³/г·с².

После дискретизации исходного уравнения на сетке, где задана функция $\Delta g(x, y)$, и аппроксимации интегрального оператора по квадратурным формулам задача (1) сводится к системе линейных алгебраических уравнений с симметричной матрицей 40000×40000 .

Для решения задачи использовался параллельный итеративно регуляризованный метод простой итерации с параметром регуляризации $\alpha = 0.001$.

На рис. 5 изображено исходное аномальное гравитационное поле для области S_2 .

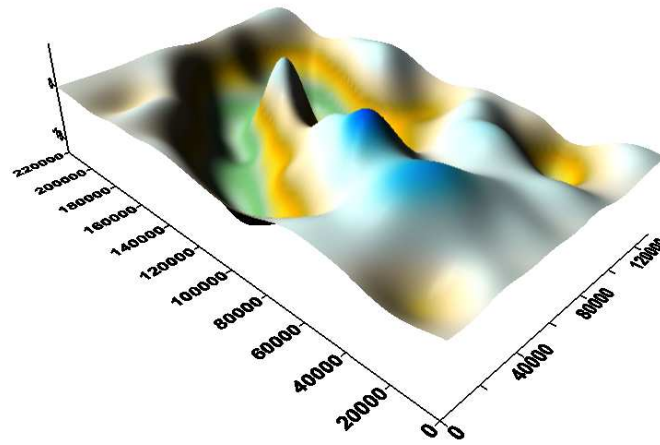


Рис. 5. Аномальное гравитационное поле $\Delta g(x, y)$ для области S_2

На рис. 6 изображены линии уровня и распределение плотности в слое, восстановленной по выделенному аномальному полю для области S_2 .

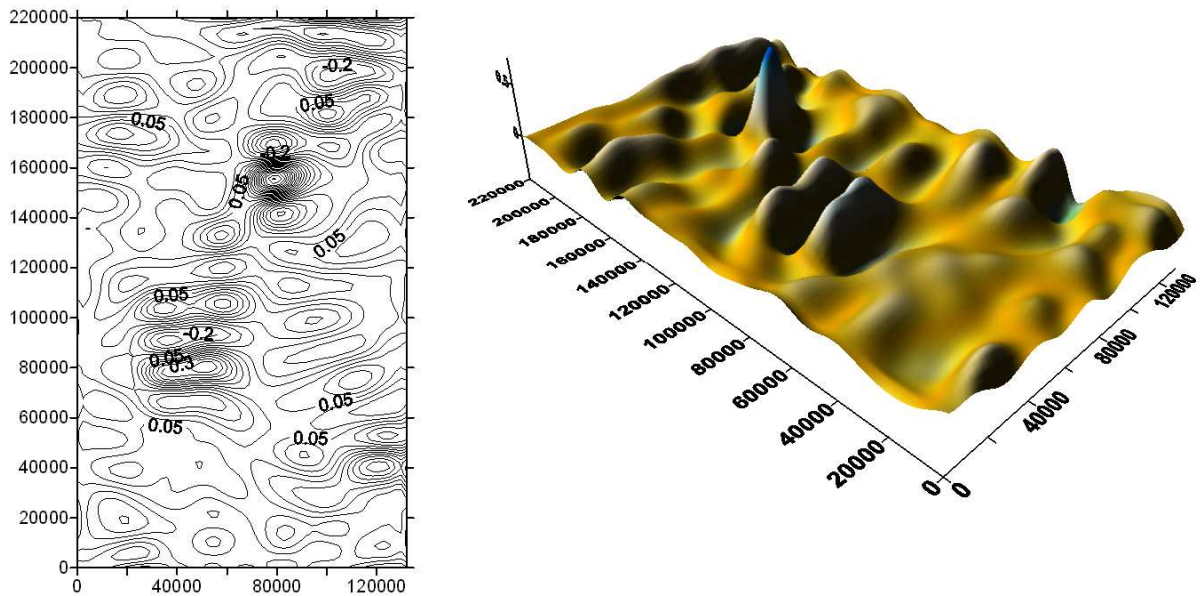


Рис. 6. Линии уровня и распределение восстановленной плотности в слое для области S_2

Результаты решения задачи 2 переданы специалистам в Институт геофизики УрО РАН для геофизической интерпретации.

В табл. 1 приведены времена счета и коэффициенты ускорения и эффективности решения модельной задачи о восстановлении плотности в слое с использованием параллельного алгоритма МПИ (число итераций — 72) на МВС-1000/64 для 200×200 точек сетки.

В табл. 2 приведены времена счета и коэффициенты ускорения и эффективности решения задачи гравиметрии с реальными данными с использованием параллельного алгоритма МПИ (число итераций — 430) на МВС-1000/64 для 200×200 точек сетки.

Отметим, что матрица СЛАУ формируется и хранится в памяти каждого процессора по частям.

Результаты вычислений показывают, что решение задач гравиметрии о нахождении плотности в слое на МВС–1000 с использованием параллельных алгоритмов существенно уменьшает время счета.

Таблица 1. Решение модельной задачи 1 о восстановлении плотности в слое

m	$T_m, \text{min.}$	S_m	E_m
1	9.15	—	—
2	5.39	1.70	0.85
3	3.27	2.80	0.93
4	2.76	3.32	0.83
6	1.48	6.18	1.03
8	1.43	6.40	0.80
10	1.15	7.96	0.80
15	0.72	12.7	0.85

Таблица 2. Решение задачи гравиметрии 2 о восстановлении плотности в слое

m	$T_m, \text{min.}$	S_m	E_m
1	55.82	—	—
2	32.96	1.69	0.85
3	20.83	2.80	0.93
4	15.13	3.69	0.92
8	7.72	7.23	0.90
10	6.26	8.92	0.89
15	4.16	13.4	0.89
20	3.28	17.0	0.85
30	2.12	26.3	0.88
40	1.80	30.0	0.76
60	1.19	46.9	0.78
80	0.88	63.4	0.79

4. Распараллеливание на видеоускорителях с помощью технологии CUDA и результаты численных экспериментов

Для организации параллельных вычислений актуальным в настоящее время является использование видеоускорителей (GPU) компании NVIDIA (рис. 7) [8]. Основой распараллеливания служит архитектура графических процессоров. В видеочипах NVIDIA базовым блоком является мультипроцессор, содержащий восемь—десять ядер, несколько сотен арифметико-логических устройств (ALU), несколько тысяч регистров и небольшое количество разделяемой общей памяти. Видеоускорители NVIDIA содержат быструю глобальную память с возможностью доступа к ней всех мультипроцессоров, локальную память в каждом мультипроцессоре, а также специальную память для констант. Работа нескольких ядер

мультипроцессора основана на архитектуре типа SIMD, т.е. каждый из процессоров выполняет одну и ту же команду над разными элементами данных.

Видеочипы GPU состоят из массивов исполнительных блоков, управляющих потоками блоков, разделяемой памяти небольшого объема и контроллеров памяти на несколько каналов. Такая архитектура позволяет чипу обрабатывать нескольких тысяч потоков данных, требующих высокой пропускной способности памяти.



Рис. 7. Видеоускоритель GeForce GTX 285

Для поддержки параллельных вычислений компания NVIDIA разработала технологию CUDA [9] — среду разработки программ на языке Си, позволяющую создавать программное обеспечение для решения сложных вычислительных задач.

Модель программирования в CUDA основывается на группировании потоков. Потоки объединяются в блоки потоков (thread block) — одномерные или двумерные сетки потоков, взаимодействующих между собой при помощи разделяемой памяти и точек синхронизации. Программа (ядро, kernel) выполняется над сеткой (grid) блоков потоков (thread blocks) (рис. 8). Каждый блок может быть одно-, двух- или трехмерным по форме и состоять из 512 потоков на текущем аппаратном обеспечении.

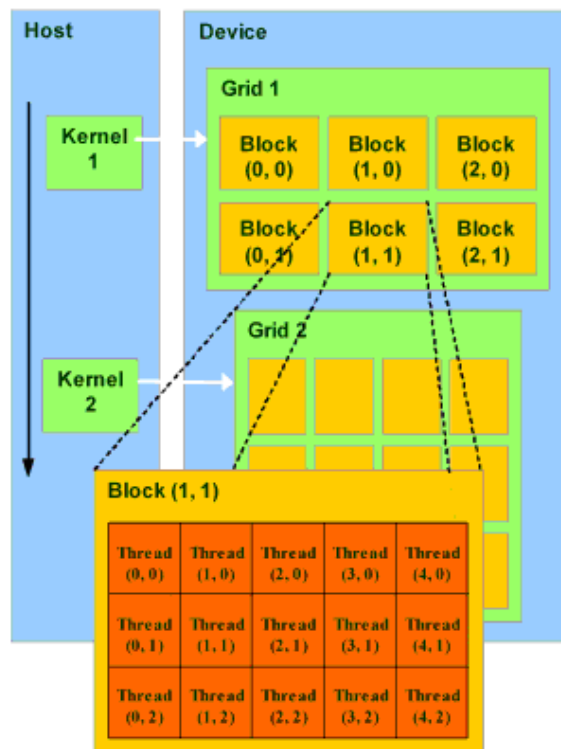


Рис. 8. Модель параллельной обработки данных

Основной процесс приложения CUDA работает на универсальном компьютере (host). CPU-код инициализирует GPU, распределяет память на видеокарте, копирует константы в память видеокарты, запускает несколько копий процессов kernel на видеокарте, копирует полученный результат из видеопамати, освобождает память и завершает работу.

Параллельная часть кода программы, написанной на языке CUDA, выполняется как множество нитей (потоков). Для оптимизации работы с памятью нити группируют в блоки фиксированного размера. Блоки объединяются в группы блоков. Параллельная процедура выполняется над группой блоков. Типовая процедура (kernel) параллельной обработки массива Data выглядит следующим образом: `kernel (Data) <<<blocks, threads>>>`, где `blocks` — количество блоков в сетке, `threads` — количество потоков в блоке.

Рассмотрим распараллеливание решения обратной задачи гравиметрии (1) о восстановлении переменной плотности в слое методом простой итерации с помощью CUDA. Базовыми операциями для распараллеливания итерационных процессов (в частности, МПИ) является реализация параллельных функций над матрицами и векторами: параллельное умножение матрицы на вектор, параллельное умножение матриц и т.д.

Для оптимизации работы с памятью весь объем данных разбивается на блоки для вычисления в процедуре kernel. Для оптимизации счета матрица A порядка N и вектор Z размерности N расширяются до размерности M и дополняются нулями таким образом, чтобы M было кратно числу блоков. Размер блока `BLOCK_SIZE` (threads) выбирается равным 16, поскольку в одном блоке группируются до 512 потоков. Тогда количество блоков вычисляется по формуле: **blocks = $M/BLOCK_SIZE$** .

Синтаксис процедуры выглядит следующим образом:

`matrixMulVector <<< M/BLOCK_SIZE, BLOCK_SIZE >>> (X, A, Z)`, где X — адрес массива из M ячеек для сохранения результатов. Тогда блок из `BLOCK_SIZE` потоков будет выполняться на одном мультипроцессоре. Потоки будут иметь общую разделяемую память и обрабатывать `threads` строк данных для нашей задачи (рис. 9).

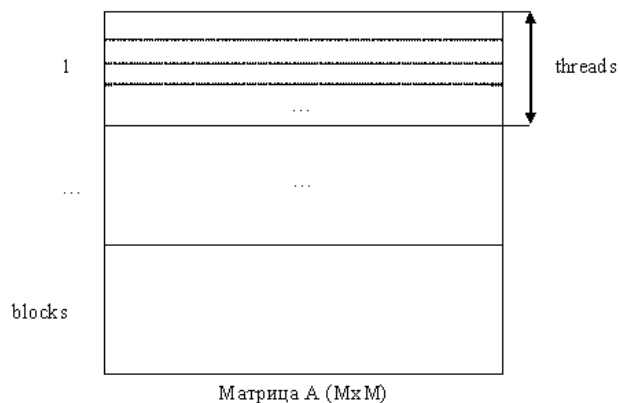


Рис. 9. Разбиение данных для процедуры CUDA

Описанные выше линейные задачи гравиметрии 1 и 2 были решены на видеоускорителях GeForce GTX 285 (GPU-1) и GeForce GTX 260 (GPU-2).

Результаты решения задач представлены на рис. 4 и 6.

В табл. 3 приводятся технические характеристики систем.

Отметим, что для эффективных расчетов объем памяти на host-компьютере должен превышать суммарный объем используемых в задаче массивов данных. Размер требуемой оперативной памяти для матрицы размерности $M \times N$ вещественных чисел одинарной точности вычисляется по формуле: **size_ts = $M \times N \times \text{sizeof}(\text{float})$** .

В табл. 4 приводятся результаты решения модельной задачи гравиметрии 1 и задачи гравиметрии с реальными данными 2 на Host-процессоре Intel Core I7 без использования видеоускорителей и на Host-процессоре с использованием видеоускорителей GeForce на более крупной сетке 150×150 (матрица СЛАУ 22500×22500).

Таблица 3. Технические характеристики систем

Характеристики подсистемы GPU-1:	GeForce GTX 285
Количество процессорных ядер	240
Частота ядра (МГц)	648
Частота процессора (МГц)	1476
Частота памяти (МГц)	2484
Количество видеопамяти (Мб)	1024
Полоса пропускания памяти (Гб/сек)	159
Характеристики подсистемы GPU-2:	GeForce GTX 260
Количество процессорных ядер	192
Частота ядра (МГц)	576
Частота процессора (МГц)	1242
Частота памяти (МГц)	1998
Количество видеопамяти (Мб)	896
Полоса пропускания памяти (Гб/сек)	111.9
Характеристики CPU:	Host
Процессор (ГГц)	Intel Core I7 2.4
Оперативная память (Гб)	3
Разрядность ОС (Бит)	64

Таблица 4. Решение задач гравиметрии

Система		Время, мин.
Host Intel Core I7	Задача 1	3.33
Host+GeForce GTX 285 (240 ядер)	Задача 1	1.25
Host+GeForce GTX 260 (192 ядра)	Задача 1	1.68
Система		Время, мин.
Host Intel Core I7	Задача 2	26.45
Host+GeForce GTX 285 (240 ядер)	Задача 2	7.42
Host+GeForce GTX 260 (192 ядра)	Задача 2	10.3

Для решения задач гравиметрии на более мелкой сетке 200×200 (матрица СЛАУ 40000×40000) требуется увеличение оперативной памяти на host-компьютере до 8 Гбайт.

Таким образом, время решения задач гравиметрии довольно большой размерности на видеоускорителях GeForce сокращает время счета и сравнимо с временем решения задач на МВС-1000/64 (см. табл. 1 и табл. 2).

5. Заключение

Для решения линейной обратной задачи гравиметрии о восстановлении переменной плотности в слое численно реализованы на многопроцессорном вычислительном комплексе МВС–1000 и графических процессорах на видеокартах NVIDIA регулярные параллельные итерационные алгоритмы.

Для решения модельной задачи гравиметрии и задачи с реальными данными проведено сравнение времени счета параллельных алгоритмов на видеоускорителях GeForce GTX 285, GeForce GTX 260 и МВС–1000/64 с анализом эффективности и ускорения.

Литература

1. Мартышко П.С., Пруткин И.Л. Технология разделения источников гравитационного поля по глубине // Геофизический журнал. 2003. Т. 25. № 3. С. 159–168.
2. Мартышко П.С., Кокшаров Д.Е. Об определении плотности в слоистой среде по гравитационным данным // Геофизический журнал. 2005. Т. 27. № 4. С. 678–684.
3. Васин В.В., Агеев А.Л. Некорректные задачи с априорной информацией. Екатеринбург: Наука, 1993. 262 с.
4. Васин В.В., Еремин И.И. Операторы и итерационные процессы Фейеровского типа. Теория и приложения. Екатеринбург: УрО РАН. 2005. 210 с.
5. Акимова Е.Н., Гемайдинов Д.В. Параллельные алгоритмы решения задачи гравиметрии о восстановлении плотности в слое // Труды института математики и механики УрО РАН. 2007. Т. 13. № 3. С. 3–21.
6. Акимова Е.Н. Параллельные алгоритмы решения обратных задач гравиметрии и магнитометрии на МВС-1000 // Вестник ННГУ. 2009. № 4. С. 181–189.
7. Baranov A.V., Latsis A.O., Sazhin C.V., Khramtsov M.Yu. The MVS-1000 System User's Guide. URL: <http://parallel.ru/mvs/user.html>.
8. URL: <http://www.nvidia.ru/>
9. URL: <http://www.ixbt.com/video3/cuda-1.shtml>