Требования к оформлению программ

1. Каждая версия (последовательная, OpenMP и др.) является отдельной программой.
2. Все программы должны являться консольными приложениями на языке C++ (с расширениями Cilk Plus, CUDA C для соответствующих версий).
3. Программы должны допускать сборку без модификации исходных файлов под ОС Windows и Linux. Других ограничений на среду разработки нет. Сборка при проверке будет осуществляться напрямую из исходных файлов (т.е. не через файлы Visual Studio или других сред). При разработке с помощью Visual Studio настоятельно рекомендуется создавать проекты типа Console Application и устанавливать флаг Empty Project.
4. Не разрешается использование сторонних библиотек кроме необходимых для рассматриваемых технологий параллельного программирования[1].
5. Ввод входных данных, вывод результата и времени вычислений осуществляются через файлы. Формат и содержание файлов описаны в задании.
6. Имена входного и выходного файлов и файла для записи времени передаются через параметры командной строки. В некоторых задачах требуется более одного входного и/или выходного файла, в их описании это указано явным образом.
7. Необходимо записать в файл для времени одно вещественное число – время работы вычислительной части программы (непосредственно реализации алгоритма) без учета чтения из файла, записи результата и других вспомогательных действий. В качестве единицы измерения времени использовать секунды.
8. В командной строке сначала указываются входные файлы, затем выходные файлы, затем файл для записи времени.

   Запуск программ с одним входным и одним выходным файлом выглядит следующим образом:
   ```
   program inputfile outputfile timefile
   ```
   Для программ с несколькими входными и выходными файлами:
   ```
   program inputfile1 … inputfileN outputfile1 … outputfileN timefile
   ```
9. Примеры входа и выхода в описании задания даны с целью иллюстрации формата, если он не является стандартным. Примеры не обязательно являются «разумным» входом и корректным выходом.

[1] Данное требование может вызвать диссонанс с принципами написания качественного кода. Действительно, для ряда задач выделение общей для всех версий функциональности в библиотеку являлось бы хорошим решением, однако предыдущий опыт показал, что иногда это усложняет даже ручную проверку, и тем более нежелательно в случае автоматической. Так как по объему кода все задачи являются маленькими (а большинство – очень маленькими) и курс посвящен высокопроизводительным вычислениям, а не написанию качественного кода, данное требование представляется меньшим из зол.

Сборка и проверка программ

1. Сборка и проверка программ будет производиться автоматизировано на одном из университетских серверов.
2. В ограниченный период времени перед крайним сроком будет предоставляться доступ для проверки успешности сборки, корректности и производительности разработанных программ на данном сервере. Подробности будут объявлены по мере приближения крайних сроков.
3. Термин «проверка» в предыдущем пункте означает проверку, что готовая работающая (разработанная и отлаженная заранее) программа также собирается и работает на тестирующем сервере. Тестирующий сервер не будет предоставляться под разработку и отладку.
4. Характеристики сервера: Fedora 16 64 bit, 2x Intel Xeon E5520, Intel C++ Compiler for Linux 12.1.5, NVIDIA Tesla C1050, CUDA 5.0.
5. Для финальных версий всех программ будет выполняться просмотр кода. При обнаружении явных случаев подгонки под ответ или манипуляций со временем работы задание будет считаться несданным.
6. В случае возникновения спорных ситуаций / обнаружения неустранимых ошибок в тестирующей системе для соответствующих задач может / будет применяться ручная проверка. Все требования к оформлению программ и способ проверки корректности сохраняются.
7. Методы должны быть реализованы в общем виде, как описано в задании, даже если проверка осуществляется на указанных тестовых задачах. Проверка корректности с помощью тестовых задач состоит в вызове общего метода для указанных данных, не допускается жесткое внесение этих данных непосредственно в реализацию метода.
8. Способ проверки корректности «сравнение с эталоном» означает запуск тестируемой и эталонной программы на одинаковых входных данных и проверку, что результаты отличаются в рамках допустимой погрешности.
9. Входные данные для проверки корректности и производительности будут по возможности подобраны так, чтобы время вычислений было в пределах от 0.1 до 10 секунд.
10. Указанные в заданиях граничные значения размера задач и численные границы для проверки корректности стоит рассматривать как ориентировочные. Они могут быть индивидуально изменены в случае, если разумная реализация будет работать слишком быстро или долго, или не будет удовлетворять слишком жестким требованиям по корректности.
11. Для параллельных версий будет проверяться эффективность по отношению к последовательной версии. Данная проверка будет проводиться на входных данных, для которых время работы последовательной версии превышает 0,1 секунды. Для OpenMP, Cilk и TBB версий эффективность масштабирования при выполнении на N ядрах определяется как $E(N) = T(1) / (N * T(N))$, где $T(1)$ – время последовательной версии, $T(N)$ – время параллельной. В постановках задач указана минимальная допустимая величина $E(N)$. При проверке будет использоваться время работы вычислительной части программы (из файла). Для CUDA-версии требуется, чтобы ее время работы не превышало время работы OpenMP-версии на 4 ядрах.

Стандартные форматы

1. Стандартный формат векторов (в качестве разделителя служит пробел или окончание строки):

   ```
   КоличествоЭлементов
   элемент1
   элемент2
   …
   элементN
   ```

   Пример: вектор $\begin{pmatrix} 0 \\ 1 \\ -3 \end{pmatrix}$ задается в данном формате следующим образом:

   ```
   3
   0
   1
   -3
   ```

2. Стандартный формат плотных матриц (в качестве разделителя служит пробел или окончание строки):

   ```
   КоличествоСтрок КоличествоСтолбцов
   элемент11 элемент12 … элемент1N
   элемент21 элемент22 … элемент2N
   …
   элементM1 элементM2 … элементMN
   ```

   Пример: матрица $\begin{pmatrix} -1 & 3 & 2 \\ 1 & 0 & 1 \end{pmatrix}$ задается в данном формате следующим образом:

   ```
   2 3
   -1 3 2
   1 0 1
   ```

3. Стандартный формат разреженных матриц mtx (широко применяемый формат, описание можно найти, например, в http://math.nist.gov/MatrixMarket/formats.html, пункт Matrix Market Exchange Formats, Coordinate Format).

   Уточнение: считать, что во входном файле элементы не упорядочены по номеру строки или столбца. Таким образом, наиболее простой способ считать такую матрицу в формат CRS или CCS – предварительно считать ее из файла в координатный формат, а затем из координатного – в требуемый в постановке задачи. При этом время конвертаций не включать во время работы вычислительной части.

Список задач для студентов, изучающих только предмет «Модели и методы высокопроизводительных вычислений»

Генераторы псевдослучайных чисел

Задача 1

Реализовать генератор одномерного равномерного распределения на отрезке [a, b] (без использования функции rand). Оценить корректность генератора с помощью критериев согласия Пирсона (Хи-квадрат) и Колмогорова-Смирнова (гипотеза: распределение является равномерным на [a, b]). Для вычисления статистики критерия Пирсона использовать 11 интервалов равного размера. Последовательность, порождаемая параллельной версией, должна быть идентична последовательности, порождаемой последовательной версией.

Формат входа: количество чисел, a, b.
Формат выхода: значение статистики критерия согласия Пирсона (Хи-квадрат), значение статистики критерия Колмогорова-Смирнова, список всех сгенерированных чисел.
Пример входа:

    3 -0.1 0.7

Пример выхода:

    0.5
    0.712
    0.11
    0.31
    0.26

Способ проверки корректности: проверка правильности вычисления статистик, стабильное прохождение тестов для уровня значимости 5% при количестве чисел более 1000.
Ограничения на размер задач: не более 100 000 000 чисел.
Требования к масштабируемости: эффективность не менее 50%.

Задача 2

Реализовать генератор одномерного нормального распределения с заданными параметрами μ, σ^2 (без использования функции rand). Оценить корректность генератора с помощью критериев согласия Пирсона (Хи-квадрат) и Колмогорова-Смирнова (гипотеза: распределение является нормальным с параметрами μ, σ^2). Вычисление статистики критерия Пирсона производить аналогично задаче 1. Последовательность, порождаемая параллельной версией, должна быть идентична последовательности, порождаемой последовательной версией.

Формат входа: количество чисел, μ, σ^2.
Формат выхода: аналогично задаче 1.
Пример входа:
3 0.1 2.0
Пример выхода: аналогично задаче 1.
Способ проверки корректности: аналогично задаче 1.
Ограничения на размер задач: аналогично задаче 1.
Требования к масштабируемости: аналогично задаче 1.

Разреженная матричная арифметика

Задача 11

Реализовать алгоритм умножения разреженных матриц: $AB = C$, A, B, C – разреженные матрицы. В процессе вычислений представлять матрицы в координатном формате.

Формат входа: в первом файле матрица A в формате mtx, во втором файле матрица B в формате mtx.
Формат выхода: матрица C в формате mtx.
Способ проверки корректности: сравнение с эталоном. Норма разности между полученным и эталонным решениями не должна превышать 1% от нормы эталонного решения. Норма эталонного решения будет не меньше 1.
Ограничения на размер задач: количество ненулевых элементов в матрицах A и B не более 100 000 000.
Требования к масштабируемости: эффективность не менее 50%.

Задача 12

Реализовать алгоритм умножения разреженных матриц. В процессе вычислений представлять матрицы в строковом формате (CRS).

Формат входа: аналогично задаче 11.
Формат выхода: аналогично задаче 11.
Способ проверки корректности: аналогично задаче 11.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: аналогично задаче 11.

Задача 13

Реализовать алгоритм умножения разреженных матриц. В процессе вычислений представлять матрицы в столбцовом формате (CCS).

Формат входа: аналогично задаче 11.
Формат выхода: аналогично задаче 11.
Способ проверки корректности: аналогично задаче 11.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: аналогично задаче 11.

Задача 14

Реализовать алгоритм Густавсона для умножения разреженных матриц.

Формат входа: аналогично задаче 11.
Формат выхода: аналогично задаче 11.
Способ проверки корректности: аналогично задаче 11.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: аналогично задаче 11.

Задача 15

Реализовать алгоритм умножения разреженной матрицы в строковом формате (CRS) на плотную: $AB = C$, A, C – разреженные матрицы, B – плотная матрицы.

Формат входа: в первом файле матрица A в формате mtx, во втором файле матрица B в стандартном плотном формате.
Формат выхода: матрица C в формате mtx.
Способ проверки корректности: аналогично задаче 11.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: аналогично задаче 11.

Задача 16

Реализовать приведение квадратной, невырожденной, разреженной матрицы в формате CRS к верхнетреугольному виду (итерациями метода Гаусса).

Формат входа: матрица в формате mtx.
Формат выхода: приведенная матрица в формате mtx.
Способ проверки корректности: аналогично задаче 11.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: эффективность не менее 25%.

Задача 17

Реализовать решение треугольной системы $Ux = b$, где U – верхнетреугольная, квадратная, невырожденная, разреженная матрица, x, b – плотные векторы.

Формат входа: матрица U в формате mtx; в качестве b взять плотный столбец из единиц соответствующего размера.
Формат выхода: x в стандартном векторном формате.
Способ проверки корректности: проверка, что невязка системы на найденном решении не превышает 1% от нормы матрицы системы.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: эффективность не менее 25%.

Задача 18

Реализовать решение треугольной системы $Lx = b$, где L – нижнетреугольная, квадратная, невырожденная, разреженная матрица, x, b – плотные векторы.

Формат входа: матрица L в формате mtx; в качестве b взять плотный столбец из единиц соответствующего размера.
Формат выхода: аналогично задаче 17.
Способ проверки корректности: аналогично задаче 17.
Ограничения на размер задач: аналогично задаче 11.
Требования к масштабируемости: аналогично задаче 17.

Сортировки

Задача 21

Реализовать побайтовую восходящую сортировку для типа double в общем случае (числа могут быть разных знаков).

Формат входа: стандартный формат векторов.
Формат выхода: стандартный формат векторов.
Способ проверки корректности: проверка упорядоченности результата.
Ограничения на размер задач: количество элементов массива не более 100 000 000.
Требования к масштабируемости: эффективность не менее 50%.

Задача 22

Реализовать побайтовую восходящую сортировку для типов int, unsigned int, float.

Формат входа: в начале файла символ, определяющий тип (i – int, u – unsigned int, f – float), далее согласно стандартному формату векторов.
Формат выхода: стандартный формат векторов.
Пример входа:
i
3
1
2
-3
Способ проверки корректности: проверка упорядоченности результата.
Ограничения на размер задач: количество элементов массива не более 100 000 000.
Требования к масштабируемости: эффективность не менее 50%.

Задача 23

Реализовать побитовую восходящую сортировку для типов int, unsigned int, float.

Формат входа: аналогично задаче 22.
Формат выхода: стандартный формат векторов.
Способ проверки корректности: проверка упорядоченности результата.
Ограничения на размер задач: количество элементов массива не более 100 000 000.
Требования к масштабируемости: аналогично задаче 22.

Список задач для студентов, изучающих предметы «Модели и методы высокопроизводительных вычислений» и «Параллельные численные методы»

Быстрое преобразование Фурье

Задача 111

Реализовать «in-place» быстрое преобразование Фурье без использования перестановки элементов массива для размера входного сигнала, являющегося степенью двойки. Реализовать прямое преобразование в форме без нормирующего множителя:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn}$$

Формат входа: первое число (всегда четное) задает количество вещественных чисел в файле, каждый элемент сигнала является комплексным и задается двумя подряд идущими вещественными числами – вещественной и мнимой частью. Таким образом, формально формат выглядит так же, как стандартный векторный формат (сначала N, затем N вещественных чисел), но интерпретируется по-другому.

Пример входа:

6
1.0
2.0
-1.0
-3.0
1.2
3.5

Данная запись задает сигнал из 3 компонент: (1 + 2i, –1 – 3i, 1.2 + 3.5i).

Формат выхода: такой же, как и формат входа.

Способ проверки корректности: сравнение с эталоном. Норма разности между полученным и эталонным решениями не должна превышать 1% от нормы эталонного решения. Норма эталонного решения будет не меньше 1.

Ограничения на размер задач: количество элементов вектора не более 10 000 000.

Требования к масштабируемости: эффективность не менее 75%.

Задача 112

Реализовать алгоритм быстрого преобразования Фурье на общий случай размера входного сигнала [Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток: Пер. с англ. - М.: Радио и связь, 1985. - 248 с.]. Реализовать прямое преобразование в форме без множителя (аналогично задаче 111).

Формат входа: аналогично задаче 111.

Формат выхода: аналогично задаче 111.

Способ проверки корректности: аналогично задаче 111.

Ограничения на размер задач: аналогично задаче 111.

Требования к масштабируемости: эффективность не менее 25%.

Решение трехдиагональных СЛАУ

Задача 121

Реализовать метод блочной прогонки для решения трехдиагональных систем линейных уравнений. Применить его для решения задачи кубической сплайн-интерполяции.

Для проверки корректности использовать следующую схему. Выбрать отрезок $[a, b]$ и функцию $f(x)$, достаточно гладкую и не являющуюся кубическим полиномом на этом отрезке. Произвести сплайн-интерполяцию функции $f(x)$ на отрезке $[a, b]$ на равномерной сетке с граничными условиями $S''(a) = f''(a), S''(b) = f''(b)$. Размерность сетки n задается во входном файле. Провести численную оценку погрешности интерполяции функции и производной. Для этого найти величины $M = \max_{x\in[a,b]}|S(x) - f(x)|$ и $M' = \max_{x\in[a,b]}|S'(x) - f'(x)|$ на точках вспомогательной сетки с шагом в 4 раза меньше шага сетки, на которой производилась сплайн-интерполяция. Записать в выходной файл M, M'.

Формат входа: размерность сетки n.
Формат выхода: погрешность интерполяции функции и производной, вычисленная на вспомогательной сетке.
Способ проверки корректности: проверка теоретического порядка зависимости погрешностей от размерности сетки. Для этого погрешности на сетках размерностей n = 100, 1000, … для соседних значений n должны отличаться в $(8)^{\text{теоретический\_порядок}}$ – $(12)^{\text{теоретический\_порядок}}$ раз.
Ограничения на размер задач: размерность сетки не более 1 000 000.
Требования к масштабируемости: эффективность не менее 25%.

Задача 122

Реализовать метод блочной прогонки для решения трехдиагональных систем линейных уравнений. Применить его для решения задачи построения квадратичного сплайна.

Формат входа: аналогично задаче 121.
Формат выхода: аналогично задаче 121.
Способ проверки корректности: аналогично задаче 121.
Ограничения на размер задач: аналогично задаче 121.
Требования к масштабируемости: аналогично задаче 121.

Задача 123

Реализовать метод циклической редукции для решения трехдиагональных систем линейных уравнений. Применить его для решения динамической задачи теплопроводности. Использовать схему Кранка-Николсона.
Уравнение теплопроводности имеет вид:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), \qquad x \in [0, X], t \in [0, T].$$

Для проверки корректности использовать следующую схему. Выбрать значения X, T таким образом, чтобы выполнялись условия сходимости. Выбрать в качестве эталонного решения достаточно гладкую функцию $u^*(x,t)$, не являющуюся константой по x и t. Составить тестовую задачу, точным решением которой является $u^*(x,t)$. Для этого взять $f(x,t) = \frac{\partial u^*(x,t)}{\partial t} - \frac{\partial^2 u^*(x,t)}{\partial x^2}$, начальные и граничные условия взять в соответствии с функцией $u^*(x,t)$.

Решить данную задачу численно на сетке размерности nx по x и nt по t. Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением на последнем шаге по времени.

Формат входа: nx, nt.
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 121.
Ограничения на размер задач: произведение размерностей сетки не более 10 000 000.
Требования к масштабируемости: аналогично задаче 121.

Задача 124

Реализовать метод циклической редукции для решения трехдиагональных систем линейных уравнений. Применить его для решения динамической задачи теплопроводности. Использовать чисто неявную разностную схему.

Формат входа: аналогично задаче 123.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Блочные алгоритмы линейной алгебры

Задача 131

Реализовать блочный алгоритм решения системы линейных уравнений с треугольной матрицей и несколькими правыми частями: $LX = B$.

Формат входа: матрица системы L в стандартном плотном формате; для задачи с k неизвестными в качестве матрицы правых частей взять первые k столбцов единичной матрицы соответствующего размера.
Формат выхода: матрица решений X.
Способ проверки корректности: проверка, что норма невязки системы на найденном решении не превышает 1% нормы матрицы системы.
Ограничения на размер задач: количество элементов в матрице системы не более 100 000 000.
Требования к масштабируемости: эффективность не менее 75%.

Задача 132

Реализовать блочное LU-разложение для квадратной матрицы.

Формат входа: матрица в стандартном плотном матричном формате.
Формат выхода: в первом файле матрица L в стандартном плотном матричном формате, во втором файле матрица U в стандартном плотном матричном формате.
Способ проверки корректности: проверка, что норма разности произведения LU и исходной матрицы не превышает 1% нормы исходной матрицы.

Ограничения на размер задач: количество элементов во входных матрицах не более 100 000 000.
Требования к масштабируемости: эффективность не менее 75%.

Задача 133

Реализовать блочное разложение Холецкого для симметричной положительно определенной матрицы.

Формат входа: матрица в стандартном плотном матричном формате.
Формат выхода: матрица L в стандартном плотном матричном формате.
Способ проверки корректности: проверка, что норма разности произведения LL^T и исходной матрицы не превышает 1% нормы исходной матрицы.
Ограничения на размер задач: аналогично задаче 132.
Требования к масштабируемости: эффективность не менее 50%.

Итерационные методы решения СЛАУ с разреженной матрицей

Задача 141

Реализовать метод сопряженных градиентов для решения СЛАУ с разреженной матрицей: $Ax = b$, A – разреженная матрица, x, b – плотные векторы.

Для проверки корректности использовать следующую схему. На вход подается матрица системы A и положительное число ε. Необходимо выбрать вектор x^* и в качестве правой части системы взять вектор Ax^*. Для построенной таким образом системы x^* является точным решением (если пренебречь вычислительной погрешностью). Далее необходимо запустить метод для решения полученной системы с критерием останова $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, где $x^{(k)}$ и $x^{(k+1)}$ – приближения, полученные методом на итерациях k и $k + 1$. В выходной файл записать количество произведенных итераций и норму разности полученного решения и x^*.

Формат входа: в первом файле матрица A в формате mtx, во втором файле число ε (см. описание выше).
Формат выхода: количество произведенных итераций, норма невязки системы на полученном решении.
Способ проверки корректности: проверка, что норма невязки системы на полученном решении не превышает 1% нормы матрицы системы.
Ограничения на размер задач: количество ненулевых элементов в матрице системы не более 100 000 000.
Требования к масштабируемости: эффективность не менее 50%.

Задача 142

Реализовать метод верхней релаксации для решения СЛАУ с разреженной матрицей. В параллельной реализации построить модификацию по аналогии с методом Якоби.

Формат входа: аналогично задаче 141.
Формат выхода: аналогично задаче 141.
Способ проверки корректности: аналогично задаче 141.
Ограничения на размер задач: аналогично задаче 141.
Требования к масштабируемости: аналогично задаче 141.

Задача 143

Реализовать метод бисопряженных градиентов (BiCG) для решения СЛАУ с разреженной матрицей (матрица – произвольная квадратная).

Формат входа: аналогично задаче 141.
Формат выхода: аналогично задаче 141.
Способ проверки корректности: аналогично задаче 141.
Ограничения на размер задач: аналогично задаче 141.
Требования к масштабируемости: аналогично задаче 141.

Задача 144

Реализовать обобщенный метод минимальных невязок (GMRes) для решения СЛАУ с разреженной матрицей (матрица – произвольная квадратная).

Формат входа: аналогично задаче 141.
Формат выхода: аналогично задаче 141.
Способ проверки корректности: аналогично задаче 141.
Ограничения на размер задач: аналогично задаче 141.
Требования к масштабируемости: аналогично задаче 141.

Задача 145

Реализовать обобщенный метод минимальных невязок с перезапуском (GMRes(m)) для решения СЛАУ с разреженной матрицей (матрица – произвольная квадратная).

Формат входа: аналогично задаче 141.
Формат выхода: аналогично задаче 141.
Способ проверки корректности: аналогично задаче 141.
Ограничения на размер задач: аналогично задаче 141.
Требования к масштабируемости: аналогично задаче 141.

Задача 146

Реализовать стабилизированный метод бисопряженных градиентов (BiCG Stab) для решения СЛАУ с разреженной матрицей (матрица – произвольная квадратная).

Формат входа: аналогично задаче 141.
Формат выхода: аналогично задаче 141.
Способ проверки корректности: аналогично задаче 141.
Ограничения на размер задач: аналогично задаче 141.
Требования к масштабируемости: аналогично задаче 141.

Численное решение задач математической физики

Задача 151

Реализовать решение краевой задачи для стационарного двумерного уравнения теплопроводности с использованием быстрого преобразования Фурье, рассмотреть распределение температуры в стержне.

Двумерное стационарное уравнение теплопроводности имеет вид:

$$\frac{\partial^2 u(x,y)}{\partial x^2}+\frac{\partial^2 u(x,y)}{\partial y^2}=-f(x,y), \qquad x\in[0,X], y\in[0,Y].$$

Для проверки корректности использовать схему, аналогичную описанной в задаче 123 для одномерного нестационарного уравнения теплопроводности. Выбрать значения X , Y таким образом, чтобы выполнялись условия сходимости. Необходимо выбрать в качестве эталонного решения функцию $u^*(x,y)$, подобрать правую часть и граничные условия так, чтобы $u^*(x,y)$ было точным решением задачи. Решить данную задачу численно на сетке размерности *nx* по *x* и *ny* по *y*. Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением.

Формат входа: *nx*, *ny*.
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: эффективность не менее 50%.

Задача 152

Реализовать метод Рунге-Кутта 4-го порядка для решения динамического одномерного уравнения теплопроводности методом частичной дискретизации. Выбрать размер отрезков по x, t так, чтобы выполнялись условия сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123.

Формат входа: аналогично задаче 123.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Задача 153

Реализовать метод Рунге-Кутта 4-го порядка для решения одномерного уравнения колебаний методом частичной дискретизации. Выбрать размер отрезков по x, t так, чтобы выполнялись условия сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123 для уравнения теплопроводности.

Формат входа: аналогично задаче 123.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Задача 154

Реализовать решение краевой задачи для стационарного двумерного уравнения теплопроводности с использованием метода верхней релаксации с оптимальным параметром, при организации параллельных вычислений использовать волновую схему. Двумерное стационарное уравнение теплопроводности имеет вид:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -f(x,y), \qquad x \in [0,X], y \in [0,Y].$$

Для проверки корректности использовать схему, аналогичную описанной в задаче 123 для одномерного нестационарного уравнения теплопроводности. Выбрать значения X, Y таким образом, чтобы выполнялись условия сходимости. Необходимо выбрать в качестве эталонного решения функцию $u^*(x,y)$, подобрать правую часть и граничные условия так, чтобы $u^*(x,y)$ было точным решением задачи. Решить данную задачу численно на сетке размерности *nx* по *x* и *ny* по *y*. Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением.

Формат входа: *nx*, *ny*.
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: эффективность не менее 50%.

Задача 155

Реализовать решение краевой задачи для стационарного двумерного уравнения теплопроводности с использованием метода верхней релаксации с оптимальным параметром, при организации параллельных вычислений использовать блочную волновую схему, подобрать оптимальный (для вашей задачи) размер блока. Двумерное стационарное уравнение теплопроводности имеет вид:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -f(x,y), \qquad x \in [0,X], y \in [0,Y].$$

Для проверки корректности использовать схему, аналогичную описанной в задаче 123 для одномерного нестационарного уравнения теплопроводности. Выбрать значения X , Y таким образом, чтобы выполнялись условия сходимости. Необходимо выбрать в качестве эталонного решения функцию $u^*(x,y)$, подобрать правую часть и граничные условия так, чтобы $u^*(x,y)$ было точным решением задачи. Решить данную задачу численно на сетке размерности *nx* по *x* и *ny* по *y*. Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением.

Формат входа: *nx*, *ny*.
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: эффективность не менее 50%.

Задача 156

Реализовать решение краевой задачи для стационарного трехмерного уравнения теплопроводности с использованием метода верхней релаксации с оптимальным параметром, при организации параллельных вычислений использовать волновую схему. Трехмерное стационарное уравнение теплопроводности имеет вид:

$$\frac{\partial^2 u(x,y,z)}{\partial x^2} + \frac{\partial^2 u(x,y,z)}{\partial y^2} + \frac{\partial^2 u(x,y,z)}{\partial z^2} = -f(x,y,z), \qquad x \in [0,X], y \in [0,Y], z \in [0,Z].$$

Для проверки корректности использовать схему, аналогичную описанной в задаче 123 для одномерного нестационарного уравнения теплопроводности. Выбрать значения X, Y, Z таким образом, чтобы выполнялись условия сходимости Необходимо выбрать в качестве эталонного решения функцию $u^*(x, y, z)$, подобрать правую часть и граничные условия так, чтобы $u^*(x, y, z)$ было точным решением задачи. Решить данную задачу численно на сетке размерности nx по x и ny по y , nz по z . Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением.

Формат входа: *nx, ny, nz*
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: эффективность не менее 50%.

Задача 157

Реализовать решение краевой задачи для стационарного трехмерного уравнения теплопроводности с использованием метода верхней релаксации с оптимальным параметром, при организации параллельных вычислений использовать блочную волновую схему, подобрать оптимальный (для вашей задачи) размер блока.
Для проверки корректности использовать схему, аналогичную описанной в задаче 156.
Вывести погрешность решения – максимальное расхождение между найденным и эталонным решением.

Формат входа: *nx*, *ny, nz*
Формат выхода: погрешность решения.
Способ проверки корректности: по порядку зависимости погрешности от размерностей сетки, аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: эффективность не менее 50%.

Задача 158

Реализовать явную схему для решения динамического одномерного уравнения колебаний. Выбрать размер отрезков по x, t так, чтобы выполнялись условия сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123.

Формат входа: аналогично задаче 123.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Задача 159

Реализовать явную схему для решения динамического одномерного уравнения теплопроводности. Подобрать размер отрезков по $x,\ t$ так, чтобы выполнялось условие сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123.

Формат входа: аналогично задаче 123.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Задача 161

Реализовать явную схему для решения нестационарного двумерного уравнения теплопроводности. Подобрать шаги сетки так, чтобы выполнялось условие устойчивости. Двумерное нестационарное уравнение теплопроводности имеет вид:

$$\frac{\partial u(t,x,y)}{\partial t} = \frac{\partial^2 u(t,x,y)}{\partial x^2} + \frac{\partial^2 u(t,x,y)}{\partial y^2} + f(x,y,z), \qquad x \in [0,X], y \in [0,Y], t \in [0,T].$$

Подобрать размер отрезков по x, y, t так, чтобы выполнялось условие сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123.

Формат входа: *nx, ny, nt*.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Задача 162

Реализовать явную схему для решения нестационарного двумерного уравнения колебаний. Подобрать шаги сетки так, чтобы выполнялось условие устойчивости. Двумерное нестационарное уравнение колебаний имеет вид:

$$\frac{\partial^2 u(t,x,y)}{\partial t^2} = \frac{\partial^2 u(t,x,y)}{\partial x^2} + \frac{\partial^2 u(t,x,y)}{\partial y^2} + f(x,y,z), \qquad x \in [0,X], y \in [0,Y], t \in [0,T].$$

Подобрать размер отрезков по x, y, t так, чтобы выполнялось условие сходимости. Для проверки корректности использовать схему, аналогичную описанной в задаче 123.

Формат входа: *nx, ny, nt*.
Формат выхода: аналогично задаче 123.
Способ проверки корректности: аналогично задаче 123.
Ограничения на размер задач: аналогично задаче 123.
Требования к масштабируемости: аналогично задаче 123.

Предобуславливание для итерационных методов

Задача 163

Реализовать вычисление предобуславливателя M для разреженной матрицы методом неполного LU-разложения. Использовать ILU(0)-предобуславливатель.

Формат входа: матрица A в формате mtx.
Формат выхода: матрица-предобуславливатель M в формате mtx.
Способ проверки корректности: $A \in M$ (все элементы A есть в M); проверка, что число обусловленности A больше, чем число обусловленности $M^{-1}A$.
Ограничения на размер задач: количество ненулевых элементов матрицы не больше 100 000 000.
Требования к масштабируемости: эффективность не менее 25%.

Задача 164

Реализовать вычисление предобуславливателя для разреженной матрицы методом неполного LU-разложения. Использовать ILU(p)-предобуславливатель.
В программе использовать значение p = 1.

Формат входа: аналогично задаче 161.
Формат выхода: аналогично задаче 161.
Способ проверки корректности: аналогично задаче 161.
Ограничения на размер задач: аналогично задаче 161.
Требования к масштабируемости: эффективность не менее 25%.

Литература

1. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. 2003. – М.: МЦНМО, 2003.Стр. 78-83.

2. Кнут Д. Искусство программирования. – М.: Вильямс, 2007. Т.2, с. 465-468.

3. Черемушкин А.В. Лекции по арифметическим алгоритмам в крипто-графии. – М.: МЦНМО, 2002. Стр. 77-80.

4. Вержбицкий В.М. Численные методы: математический анализ и обык-новенные дифференциальные уравнения. – М.: Высшая школа, 2001. – 382 с.

5. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – Бином. Лаборатория знаний, 2008. – 640 с.