



Введение в методы параллельных вычислений

Разработчик:
А.В. Старченко, д.ф.-м.н., профессор
E-mail: starch@math.tsu.ru
Томский государственный университет

Направление 010300.68
«Фундаментальная информатика и информационные технологии»

Проект комиссии Президента по модернизации и техническому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»



Разработка курса выполнена в рамках Проекта комиссии Президента РФ по модернизации и техническому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»

Применение потенциала суперкомпьютерных технологий (СКТ) как значимой составляющей инновационного развития страны является задачей государственной важности, относится к приоритетному направлению и находится под постоянным контролем Президента и Правительства России. Одним из сдерживающих факторов развития страны в этом направлении является острая нехватка высококвалифицированных кадров в области СКТ, поскольку подготовка таких специалистов сейчас отсутствует как элемент системы высшего профессионального образования.

Стратегической целью проекта является создание национальной системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения.

<http://hpc-education.ru>.

© Московский государственный университет © Томский государственный университет Старченко А.В.



Содержание курса

- Введение
- Рекуррентные формулы
- Параллельные вычисления определенных и кратных интегралов
- Умножение матрицы на вектор. Умножение матриц
- Прямые методы решения систем линейных уравнений на многопроцессорных системах Организация межпроцессорных обменов
- **Трехдиагональные системы. Параллельная реализация прямых методов решения систем линейных уравнений**
- Параллельная реализация итерационных методов решения СЛАУ
- Параллельная реализация быстрого преобразования Фурье

© Московский государственный университет © Томский государственный университет Старченко А.В.



Содержание лекции

- Введение. О решении линейных систем с трехдиагональными матрицами
- Метод прогонки
- Другой вариант метода прогонки
- Метод встречных прогонок
- Параллельная реализация метода прогонки
- Оценка ускорения
- Метод циклической редукции
- Параллельная реализация метода циклической редукции
- Оценка ускорения и эффективности
- Метод циклической редукции без обратного хода
- Заключение

© Московский государственный университет © Томский государственный университет Старченко А.В.



О решении линейных систем с трехдиагональными матрицами

- Система линейных алгебраических уравнений с трехдиагональной матрицей имеет вид:

$$\begin{pmatrix} b_0 & -c_0 & & & 0 \\ -a_1 & b_1 & -c_1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & -a_{n-1} & b_{n-1} & -c_{n-1} & \\ & & -a_n & b_n & \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}$$

© Московский государственный университет © Томский государственный университет Старченко А.В.



К решению линейных систем с трехдиагональными матрицами

- Системы такого вида получаются в результате разностных аппроксимаций дифференциальных краевых задач, а также при построении кубических сплайнов.
- Экономичными прямыми методами решения таких систем уравнений на компьютерах с последовательной архитектурой являются специальные варианты метода исключения Гаусса – метод прогонки и методы редукции.
- К сожалению, многие методы, которые оказались наиболее эффективными в последовательных алгоритмах, не всегда сохраняют такое свойство при прямом использовании на параллельных компьютерах.

© Московский государственный университет © Томский государственный университет Старченко А.В.



Метод прогонки

- Пусть необходимо решить систему линейных уравнений:

$$\begin{cases} -a_i x_{i-1} + b_i x_i - c_i x_{i+1} = f_i, i = 0, \dots, n; \\ a_0 = 0, c_n = 0. \end{cases}$$

- Будем рассматривать только системы, имеющие единственное решение и у матриц которых есть диагональное преобладание:

$$|b_i| \geq |a_i| + |c_i|, i = 0, \dots, n;$$



Метод прогонки

- «Прямой ход» - вычисление прогоночных коэффициентов

$$P_0 = \frac{c_0}{b_0}; Q_0 = \frac{f_0}{b_0};$$

$$P_i = \frac{c_i}{b_i - a_i P_{i-1}}, Q_i = \frac{f_i + a_i Q_{i-1}}{b_i - a_i P_{i-1}}, i = 1, \dots, n-1;$$

- «Обратный ход» - вычисление неизвестных

$$x_n = \frac{f_n + a_n Q_{n-1}}{b_n - a_n P_{n-1}}; x_i = P_i x_{i+1} + Q_i, i = n-1, \dots, 0;$$

- Арифметических операций: $6(n-1) + 2n + 7 \approx 8n$.



Другой вариант метода прогонки

- «Прямой ход» - вычисление прогоночных коэффициентов

$$R_n = \frac{a_n}{b_n}, T_n = \frac{f_n}{b_n};$$

$$R_i = \frac{a_i}{b_i - c_i R_{i+1}}, T_i = \frac{f_i + c_i T_{i+1}}{b_i - c_i R_{i+1}}; i = n-1, \dots, 1;$$

- «Обратный ход» - вычисление неизвестных

$$x_0 = \frac{f_0 + c_0 T_1}{b_0 - c_0 R_1}; x_i = R_i x_{i-1} + T_i, i = 1, \dots, n.$$

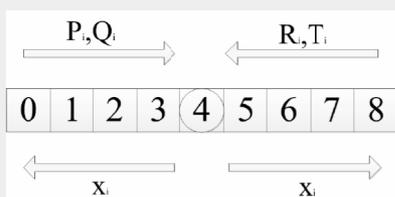


Метод встречных прогонок

- Если применять формулы первого варианта метода прогонки (расчет прогоночных коэффициентов $\{P_i\}, \{Q_i\}$) для $i=0, \dots, n/2-1$, а второго (расчет прогоночных коэффициентов $\{R_i\}, \{T_i\}$) для $i=n, \dots, n/2+1$, то можно получить вариант метода прогонки, в котором вычисления прогоночных коэффициентов могут проводиться одновременно и независимо.
- Далее, зная значение $x_{n/2}$, можно по формулам $x_i = P_i x_{i+1} + Q_i, i=n/2-1, \dots, 0; x_i = R_i x_{i-1} + T_i, i=n/2+1, \dots, n;$ одновременно и независимо вычислить все остальные неизвестные.



Метод встречных прогонок



$$x_{n/2} = \frac{f_{n/2} + a_{n/2} Q_{n/2-1} + c_{n/2} T_{n/2+1}}{b_{n/2} - a_{n/2} P_{n/2-1} - c_{n/2} R_{n/2+1}}.$$



Распараллеливание метода прогонки

- Идея метода встречных прогонок была обобщена в работе Н.Н. Яненко с сотрудниками на случай многопроцессорной системы.
- Яненко Н.Н., Коновалов А.Н., Бугров А.Н., Шустов Г.В. Об организации параллельных вычислений и "распараллеливании" прогонки // Численные методы механики сплошной среды. 1978. - №7. - С. 139-146.



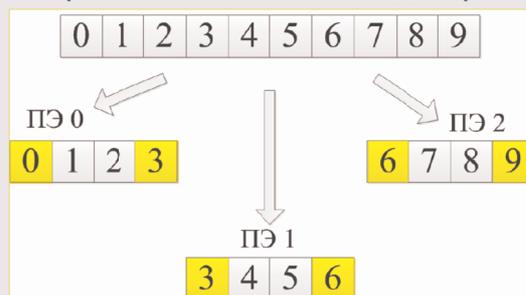


Распараллеливание метода прогонки

- Выделение части искомым неизвестных $\{x_i\}, i = 0, \dots, n$ в качестве параметрических $\{\bar{x}_\mu\}, \mu = 0, \dots, p-1$.
- Построение системы линейных уравнений относительно этих неизвестных и ее решение.
- Нахождение по $\{\bar{x}_\mu\}$ остальных неизвестных $\{x_i\}$.
- Для осуществления этой идеи система уравнений разбивается на подсистемы, количество которых будет совпадать с количеством используемых процессов p . Причем декомпозиция по данным осуществляется таким образом, что каждый μ -ый процесс обрабатывает $m+1$ уравнений ($m=n/p$).



Распараллеливание метода прогонки



- Пример декомпозиции неизвестных при $n=9, p=3$

$$x_j^{(\mu)} = v_j^{(\mu)} \bar{x}_\mu + z_j^{(\mu)} \bar{x}_{\mu+1} + w_j^{(\mu)}$$



Распараллеливание метода прогонки

- Для неизвестных системы вводятся локальные индексы для получения однородного алгоритма:
 $x_i = x_{j+\mu m} = x_j^{(\mu)}, i = 0, \dots, n; j = 0, \dots, m; \mu = 0, \dots, p-1;$
- Кроме того, принимается, что
 $x_m^{(\mu-1)} = x_0^{(\mu)} = \bar{x}_\mu, \mu = 1, \dots, p-1; \bar{x}_0 = x_0; \bar{x}_p = x_n;$
- Решение системы уравнений ищется в виде рекуррентных формул вида:
 $x_j^{(\mu)} = v_j^{(\mu)} \bar{x}_\mu + z_j^{(\mu)} \bar{x}_{\mu+1} + w_j^{(\mu)}; j = 0, \dots, m; \mu = 0, \dots, p-1;$



Распараллеливание метода прогонки

- Коэффициенты $\{v_j^{(\mu)}\}, \{z_j^{(\mu)}\}, \{w_j^{(\mu)}\}$ находятся из решения следующих систем:

$$\begin{cases} -a_{j+\mu m} v_{j-1}^{(\mu)} + b_{j+\mu m} v_j^{(\mu)} - c_{j+\mu m} v_{j+1}^{(\mu)} = 0; j = 1, \dots, m-1 \\ v_0^{(\mu)} = 1, v_m^{(\mu)} = 0. \end{cases}$$

$$\begin{cases} -a_{j+\mu m} z_{j-1}^{(\mu)} + b_{j+\mu m} z_j^{(\mu)} - c_{j+\mu m} z_{j+1}^{(\mu)} = 0; j = 1, \dots, m-1 \\ z_0^{(\mu)} = 0, z_m^{(\mu)} = 1. \end{cases}$$

$$\begin{cases} -a_{j+\mu m} w_{j-1}^{(\mu)} + b_{j+\mu m} w_j^{(\mu)} - c_{j+\mu m} w_{j+1}^{(\mu)} = f_{j+\mu m}; j = 1, \dots, m-1 \\ w_0^{(\mu)} = 0, w_m^{(\mu)} = 0. \end{cases}$$



Распараллеливание метода прогонки

- После вычисления коэффициентов производится построение системы уравнений для параметрических неизвестных $\{\bar{x}_\mu\}$.
- Для этого рассматриваются неиспользованные ранее уравнения для $i=0, m, 2m, \dots, pm$, в которых предварительно исключаются все «непараметрические» неизвестные.
- В результате имеем:

$$-a_{\mu m} v_{m-1}^{(\mu)} \bar{x}_{\mu-1} + (b_{\mu m} - a_{\mu m} z_{m-1}^{(\mu)} - c_{\mu m} v_1^{(\mu)}) \bar{x}_\mu - c_{\mu m} z_1^{(\mu)} \bar{x}_{\mu+1} = f_{\mu m} + a_{\mu m} w_{m-1}^{(\mu-1)} + c_{\mu m} w_1^{(\mu)}; \mu = 0, \dots, p$$
- Решая эту систему обычным методом прогонки, найдем $\{\bar{x}_\mu\}$



Параллельная реализация метода прогонки

- Рассмотренный вычислительный алгоритм решения СЛАУ с трехдиагональной матрицей обладает высокой степенью параллелизма – вычисления коэффициентов и нахождение неизвестных производятся с наивысшей для данной задачи степенью параллелизма p , тем не менее, решение СЛАУ для параметрических неизвестных выполняется последовательно только одним процессом при простаивающих остальных.
- Кроме того, при использовании многопроцессорных вычислительных систем с распределенной памятью вычисление коэффициентов и значений неизвестных также будет связано с дополнительными коммуникационными затратами, обусловленными необходимостью передачи данных из локальной памяти других процессов.



Оценка ускорения

- На первом этапе (вычисление коэффициентов) каждому процессу требуется около $13 \cdot m$ флопов, которые выполняются одновременно и независимо.
- Для вычисления коэффициентов системы для «параметрических» неизвестных и нахождения ее решения требуется около $18 \cdot p$ флопов. При этом потребуется передать $6 \cdot p$ чисел для расчета коэффициентов и $2 \cdot p$ чисел – «параметрических» неизвестных.
- На последнем этапе остальные «непараметрические» неизвестные вычисляются одновременно и независимо за $4 \cdot m$ флопов.



Оценка ускорения

- В итоге

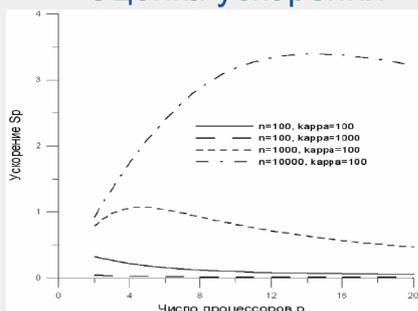
$$T_p \approx (17m + 18p)t_a + 8pt_{comm}$$

$$S_p = \frac{T_1}{T_p} \approx \frac{8p/17}{1 + 18p^2/17n + 8\kappa_{appa} \cdot p^2/17n};$$

$$\kappa_{appa} = \frac{t_{comm}}{t_a} \gg 1.$$



Оценка ускорения



- Оценка ускорения параллельной версии метода прогонки



Метод циклической редукции

- При решении системы с трехдиагональной матрицей методом циклической редукции требуется, чтобы $n=2^q$ и система имела следующий вид:

$$\begin{cases} x_0 = f_0, \\ -a_i x_{i-1} + b_i x_i - c_i x_{i+1} = f_i, i = 1, \dots, n-1; \\ x_n = f_n \end{cases}$$

- В случае невыполнения этих условий система дополняется тривиальными уравнениями вида $x_i = 0$.



Метод циклической редукции

- Идея метода: **циклически**, на каждом шаге $k=1, \dots, q-1$ ($q = \log_2 n$) с помощью линейных преобразований производится **исключение** неизвестных из системы таким образом, что оставались x_i с индексами, кратными 2^k .
- Например, на первом шаге ($k=1$) из системы исключаются неизвестные с нечетными индексами:

$$\begin{aligned} (a_i / b_{i-1}) \times & -a_{i-1}x_{i-2} + \underline{b_{i-1}x_{i-1}} - c_{i-1}x_i = f_{i-1}; \\ + & -a_i x_{i-1} + b_i x_i - c_i x_{i+1} = f_i; \\ (c_i / b_{i+1}) \times & -a_{i+1}x_i + \underline{b_{i+1}x_{i+1}} - c_{i+1}x_{i+2} = f_{i+1}; \end{aligned}$$



Метод циклической редукции

- В результате выполнения этого действия получим систему, содержащую неизвестные с четными индексами:

$$\begin{cases} x_0 = f_0, \\ -a_i^{(1)} x_{i-2} + b_i^{(1)} x_i - c_i^{(1)} x_{i+2} = f_i^{(1)}, i = 2, 4, \dots, n-2; \\ x_n = f_n. \end{cases}$$

- Продолжая редукцию неизвестных (прямой ход), на последнем ($q-1$)-ом шаге получим систему, содержащую только три уравнения:



Метод циклической редукции

$$\begin{cases} x_0 = f_0, \\ -a_{n/2}^{(q-1)} x_0 + b_{n/2}^{(q-1)} x_{n/2} - c_{n/2}^{(q-1)} x_n = f_{n/2}^{(q-1)}; \\ x_n = f_n, \end{cases}$$

- из которой следует

$$x_{n/2} = \frac{f_{n/2}^{(q-1)} + a_{n/2}^{(q-1)} f_0 + c_{n/2}^{(q-1)} f_n}{b_{n/2}^{(q-1)}}.$$



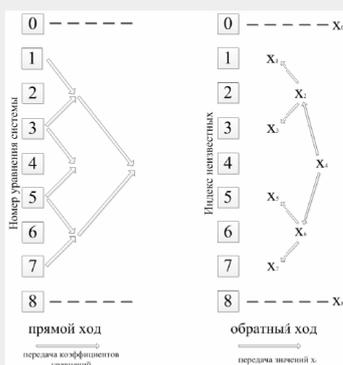
Метод циклической редукции

- Найденное значение $x_{n/2}$ может затем использоваться для определения значений неизвестных $x_{n/4}$ и $x_{3n/4}$ из редуцированной на $(q-2)$ -м шаге системы (обратный ход):

$$\begin{cases} x_0 = f_0, \\ -a_{n/4}^{(q-2)} x_0 + b_{n/4}^{(q-2)} x_{n/4} - c_{n/4}^{(q-2)} x_{n/2} = f_{n/4}^{(q-2)}; \\ -a_{n/2}^{(q-2)} x_{n/4} + b_{n/2}^{(q-2)} x_{n/2} - c_{n/2}^{(q-2)} x_{3n/4} = f_{n/2}^{(q-2)}; \\ -a_{3n/4}^{(q-2)} x_{n/2} + b_{3n/4}^{(q-2)} x_{3n/4} - c_{3n/4}^{(q-2)} x_n = f_{3n/4}^{(q-2)}; \\ x_n = f_n. \end{cases}$$



Метод циклической редукции



Метод циклической редукции

- Процедура циклической редукции включает вычисления коэффициентов и правых частей $k = 1, \dots, q-1$:

$$\begin{aligned} a_i^{(k)} &= P_i a_{i-2^{k-1}}^{(k-1)}; & b_i^{(k)} &= b_i^{(k-1)} - P_i c_{i-2^{k-1}}^{(k-1)} - Q_i a_{i+2^{k-1}}^{(k-1)}; \\ c_i^{(k)} &= Q_i c_{i+2^{k-1}}^{(k-1)}; & f_i^{(k)} &= f_i^{(k-1)} + P_i f_{i-2^{k-1}}^{(k-1)} + Q_i f_{i+2^{k-1}}^{(k-1)}; \\ P_i &= \frac{a_i^{(k-1)}}{b_{i-2^{k-1}}^{(k-1)}}; & Q_i &= \frac{c_i^{(k-1)}}{b_{i+2^{k-1}}^{(k-1)}}; & i &= 2^t, 2 \times 2^t, 3 \times 2^t, \dots, n-2^t. \end{aligned}$$

- Расчет неизвестных $k = q, q-1, \dots, 2, 1$ из

$$x_i = \frac{f_i^{(k-1)} + a_i^{(k-1)} x_{i-2^{k-1}} + c_i^{(k-1)} x_{i+2^{k-1}}}{b_i^{(k-1)}}; \quad i = 2^{t+1}, 2 \times 2^{t+1}, \dots, n-2^{t+1}.$$



Метод циклической редукции

- Для расчета всех коэффициентов одного уравнения редуцированной системы потребуется 12 флопов, всего таких расчетов необходимо сделать $(n/2-1) + (n/4-1) + \dots + (4-1) + (2-1) = n-q-1$.
- При вычислении значения одного неизвестного необходимо 5 флопов, а для всех это будет $5 \cdot (n-1)$ флопов.
- В итоге

$$T_1 \approx 12 \cdot (n-1-q) + 5 \cdot (n-1) = 17 \cdot n - 12 \cdot q - 17 = O(n).$$

- Метод циклической редукции – экономичный метод!



Параллельная реализация метода циклической редукции

- Алгоритм циклической редукции обладает высоким уровнем параллелизма:
 - во-первых, операции исключения независимы и могут выполняться параллельно.
 - во-вторых, расчет значений неизвестных проводится одновременно.
- Однако, если используется МВС с распределенной памятью и обычная схема декомпозиции данных по ПЭ, то на этапе редукции понадобится передача полученных коэффициентов в локальную память других ПЭ. Кроме того, на завершающих шагах прямого хода будет удваиваться число простаивающих процессоров.
- Похожая картина наблюдается при расчете значений неизвестных.



Параллельная реализация метода циклической редукции



Оценка ускорения и эффективности

- Оценим трудоёмкость полученной параллельной реализации алгоритма метода циклической редукции для случая $p=2^r \ll n=2^q$.
- Пусть при инициализации в памяти каждого процессорного элемента размещены по $m=n/p$ уравнений системы.
- Тогда на этапе редукции временные затраты на вычисление коэффициентов оцениваются как $12 \cdot m$ флопов, причем нужно добавить время на пересылку 4 коэффициентов для проведения редукции на каждом шаге.



Оценка ускорения и эффективности

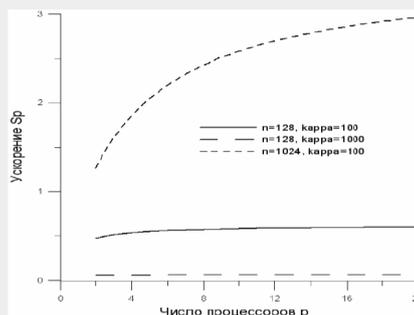
- Параллельное вычисление неизвестных будет произведено за $5 \cdot m$ флопов, которое будет сопровождаться одновременной рассылкой неизвестных процессорным элементам МВС.
- С учетом оценки времени выполнения параллельного алгоритма на этапе вычисления неизвестных получим

$$T_p \approx 17 \cdot m \cdot t_a + 5 \cdot q \cdot t_{comm};$$

$$S_p = \frac{T_1}{T_p} \approx \frac{p}{1 + \frac{5kp \log_2(n)}{17n}}; \quad E_p = \frac{S_p}{p} \approx \frac{1}{1 + \frac{5kp \log_2(n)}{17n}}$$



Оценка ускорения и эффективности



- Оценка ускорения параллельной версии метода циклической редукции



Оценка ускорения и эффективности

- Сравнивая графики изменения оценок ускорения рассмотренных алгоритмов, можно сделать вывод, что параллельный алгоритм метода циклической редукции при небольших размерах задачи ($n \sim 10^2$) и для многопроцессорных систем с медленной скоростью передачи данных между вычислительными узлами ($k \sim 10^3$) также, как и алгоритм параметрической прогонки, не является эффективным.
- При увеличении размера задачи до $n \sim 10^3$ рассмотренный параллельный вариант метода циклической редукции более предпочтителен, поскольку позволяет получить более высокие оценки величины ускорения параллельной программы.

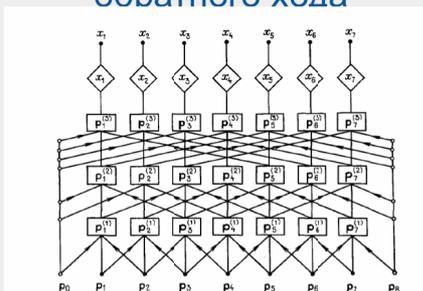


Метод циклической редукции без обратного хода

- На первом шаге из уравнений системы поочередно исключаются неизвестные с четными и нечетными номерами. В результате получаются две независимые подсистемы.
- На втором шаге редуцируется похожим образом каждая полученная подсистема, что приводит в результате к четырем независимым подсистемам, содержащим по $(n-1)/4$ неизвестных.
- ...
- На последнем шаге этапа система уравнений преобразуется к системе с диагональной матрицей (или к $n+1$ независимым уравнениям для отдельных неизвестных), решение которой можно получить, проводя одновременно вычисления по явным формулам.



Метод циклической редукции без обратного хода



- Диаграмма маршрутизации циклической редукции без обратного хода (взята из книги Хокни Р., Джессхоуп К. Параллельные ЭВМ)



Заключение

- Таким образом, рассматривая параллельные варианты методов прогонки и циклической редукции, следует отметить, что эти методы не следует использовать в качестве алгоритмов, требующих распараллеливания процедуры решения систем с трехдиагональными матрицами.
- В настоящее время в параллельных вычислениях произошел переход от мелкозернистых и с большими коммуникационными затратами параллельных алгоритмов к алгоритмам, которые имеют крупноблочный параллелизм с очень малым числом обменов за счет применения в параллельных ветвях алгоритма одновременно выполняющихся последовательных шагов.



Заключение

- Тем не менее, мелкозернистый параллелизм циклической редукции может все еще быть полезен для расчета специальных векторов и для решения специальных блочных трехдиагональных систем.
- Для блочных трехдиагональных матриц, размер которых на два-три порядка выше числа используемых процессоров, целесообразно рассматривать технологию решения, в которой трехдиагональные подсистемы целиком решаются одновременно и независимо, например, обычным методом прогонки (крупнозернистый параллелизм), а не технологию, в которой каждая отдельная трехдиагональная подсистема решалась бы параллельно (мелкозернистый параллелизм).



Вопросы для обсуждения

- При численном решении каких задач получаются системы с трехдиагональными матрицами?
- Оцените количество операций с плавающей точкой (флопов) в обычном методе прогонки. Относится ли метод прогонки к экономичным методам?
- Какие условия должны выполняться, чтобы обеспечить устойчивость метода прогонки?
- В чем заключается идея метода встречных прогонок? Что является «узким местом» этого метода?
- Какова основная идея параллельной версии метода прогонки Яненко-Коновалова-Бугрова-Шустова?
- Какие неизвестные в этом методе относятся к «параметрическим»?
- Какая рекуррентная формула лежит в основе параллельной версии метода прогонки?
- Укажите какие этапы параллельной версии метода прогонки обладают наивысшей степенью параллелизма.



Вопросы для обсуждения

- Где происходит потеря производительности параллельной версии метода прогонки?
- Что влияет на ускорение параллельной версии метода прогонки по сравнению с последовательной?
- Какие условия необходимо обеспечить для применения метода циклической редукции для решения трехдиагональных систем?
- В чем заключается идея метода циклической редукции?
- Нарисуйте диаграмму маршрутизации метода циклической редукции при $n=16$.
- Относится ли метод циклической редукции к экономичным методам? Дайте оценку количеству выполняемым в нем операций с плавающей точкой.
- Обладает ли скрытым параллелизмом метод циклической редукции?
- Что же является более предпочтительным при решении систем с трехдиагональными матрицами на многопроцессорных компьютерах? Почему?
- Стоит ли использовать мелкозернистый параллелизм прямых методов решения трехдиагональных систем?



Темы заданий для самостоятельной работы

- Оцените ускорение метода встречных прогонок по сравнению с обычным последовательным алгоритмом.
- Разработайте MPI-программу метода встречных прогонок для двухпроцессорной вычислительной системы.
- Оцените ускорение метода циклической редукции без обратного хода.
- Разработайте MPI-программу метода циклической редукции без обратного хода.



Основная литература

1. Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы. - М.: Радио и связь. 1986.
2. Самарский А.А., Николаев Е.Н. Методы решения сеточных уравнений. - М.:Наука, 1978.
3. Голуб Дж., Ван Лоун Ч. Матричные вычисления. – М.:Мир, 1999.
4. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. – М.:Мир, 1991.
5. Деммель Дж. Вычислительная линейная алгебра. – М.:Мир, 2001.
6. *Высокопроизводительные вычисления на кластерах.* – Томск: Изд-во Том. ун-та, 2008.



Следующая тема

- Введение
- Рекуррентные формулы
- Параллельные вычисления определенных и кратных интегралов
- Умножение матрицы на вектор. Умножение матриц
- Прямые методы решения систем линейных уравнений на многопроцессорных системах Организация межпроцессорных обменов
- Трехдиагональные системы. Параллельная реализация прямых методов решения систем линейных уравнений
- Параллельная реализация итерационных методов решения СЛАУ
- Параллельная реализация быстрого преобразования Фурье