УДК 004.272.2:519.63

О. А. ДМИТРИЕВА

Донецкий национальный технический университет, Украина

УПРАВЛЕНИЕ ШАГОМ ИНТЕГРИРОВАНИЯ ПРИ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ ОБОБЩЕННЫХ КОЛЛОКАЦИОННЫХ БЛОЧНЫХ МЕТОДОВ

В работе рассматриваются вопросы построения параллельных алгоритмов управления шагом интегрирования при решении задачи Коши для систем обыкновенных дифференциальных уравнений. Алгоритмы строятся на многошаговых коллокационных блочных методах со старшими производными, что обеспечивает векторизацию процедуры получения решения. Предложены параллельные алгоритмы управления шагом и порядком интегрирования для решения жестких систем, основанные на коллокационных одношаговых и многошаговых блочных методах со старшими производными. В разработанных алгоритмах локальные погрешности определяются и сопоставляются во всех коллокационных точках блока, в отличие от стадийных методов, где стадии, как правило, не совпадают, и сравнение ведется только по одной конечной расчетной точке. Вычисления реализуются независимо, и необходимость в обменах возникает только после получения конечных результатов для блоков расчетных точек.

Ключевые слова: *задача Коши, шаг интегрирования, показатели параллелизма, блок, точки коллокации, порядок аппроксимации*

Введение

Современное состояние разработок и исследований в области математического моделирования отмечается широким применением параллельных вычислительных систем [1]. Выделился класс фундаментальных научных и инженерных задач, называемый «большой вызов» [2], связанный с моделированием сложных динамических систем, которое стало возможным только с привлечением мощных вычислительных ресурсов. Сформировалась новая область исследований «технологии параллельного моделирования» (ParSimTech - Parallel Simulation Technology), стремительное развитие которой стимулируется как необходимостью решения традиционных проблем моделирования сложных динамических систем авиации и космонавтики, радиоэлектроники, военных комплексов, энергетики, химических технологий, так и сравнительно новых проблем защиты окружающей среды, метеорологии, биотехнологий, технологий сыпучих материалов, молекулярной динамики, биомеханики [3-4]. Следует отметить, что, как правило, модели, описывающие поведение таких объектов, представлены большими системами обыкновенных дифференциальных уравнений (СОДУ) с сосредоточенными параметрами. Поэтому разрешению подлежат проблемы, связанные с высокими размерностями задач, жесткостью, плохой обусловленностью, учетом большого числа параметров. Несмотря на то, что современные высокопроизводительные кластеры преодолели рубеж в петафлопс, а количество ядер, которое насчитывают такие системы, десятков и сотен тысяч, эффективность параллельных приложений на подобных системах по разным оценкам не превышает 15%, а в некоторых случаях или даже 3-5% от пиковой производительности [1-2]. Получается, что на данном этапе разработчики не в состоянии воспользоваться теми преимуществами, которые предоставляет современная элементная база. Именно поэтому в проблеме параллельных вычислений [5] следует выделить в качестве основного направление, связанное с разработкой современных численных методов и адаптацией существующих, ориентированных на параллельные вычисления, анализом их сходимости, устойчивости и вычислительной сложности.

1. Блочные методы интегрирования

Основная идея, на которой базируется конструирование блочных методов для решения СОДУ на параллельных компьютерах, заключается в одновременном получении приближений точного решения в равноотстоящих точках блока [6-7]. Однако если речь идет о численном интегрировании жестких дифференциальных уравнений, возникает необходимость в изменении шага интегрирования, что невозможно обеспечить внутри блока. С одной стороны, это является недостатком метода, но, с другой стороны, поскольку точки внутри блока расположены регулярно (рис. 1), есть возможность определения и сопоставления локальных погрешностей во

© О. А. Дмитриева

всех точках блока [8]. В этом заключается основное отличие блочных методов от стадийных, в которых стадии, как правило, не совпадают, и сравнение ведется только по одной конечной расчетной точке t_{n+1}. Также не особенно привлекательным является использование в компьютерах с распределенной памятью мелкозернистого параллелизма стадийных методов.

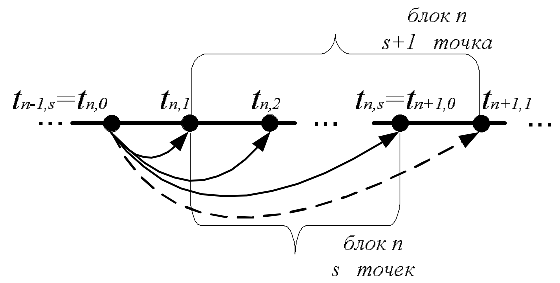


Рис. 1. Шаблон расчетной схемы блочного метода

Алгоритмы управления шагом базируются на использовании коллокационных блочных одношаговых и многошаговых методов. Параллельный счет осуществляется в пределах одного цикла для всех точек блоков с размерностями s и s+1. Две нити вычислений проходят независимо, и необходимость в обменах возникает только после получения конечных результатов для обоих блоков расчетных точек.

2. Управление шагом при интегрировании одношаговыми блочными методами

Рассматривается решение задачи Коши

$$x' = f(t, x(t)), \quad x(t_0) = x_0 \qquad (1)$$

одношаговым коллокационным блочным методом (2) с числом расчетных точек s

$$u_{n,j} = u_{n,0} + \tau \sum_{i=1}^{s} \tau^l a_{i,j}^{(l)} F_{n,i}^{(l)}, \quad j=1,2,\ldots,s, \qquad (2)$$

где $F_{n,i}^{(l)} = f^{(l)}(t_n + i\tau, u_{n,i})$ l-ая производная правой части,

$a_{i,j}^{(l)}$ - коэффициенты расчетной схемы.

Выделяются две системы процессорных узлов, на которых запускаются параллельные процессы. Первая система узлов обеспечивает параллельную реализацию одношагового коллокационного блочного s-точечного метода, что требует s процессорных элементов для получения значений $u_{n+1,i}$, i=1,2,…,s. На второй системе узлов осуществляется реализация одношагового коллокационного блочного s+1-точечного метода с получением значений $\nu_{n+1,i}$, i=1,2,…,s+1. Поскольку методы является неявными, каждый временной шаг подразумевает проведение некоторого количества итераций, обеспечивающих требуемую локальную точность. Реализация обоих методов осуществляется автономно, что обеспечивает крупнозернистость вычислений.

Размерность вычислительного поля может быть представлена двумя вариантами. В первом случае это могут быть кольцевые топологии, количество процессоров в которых совпадает с размерностью блоков s и s+1 соответственно, во втором случае вычислительные поля представляют собой решетки процессоров. Первая решетка с s (количество точек в блоке) столбцами и m (количество уравнений в системе) строками, у второй решетки размерность процессорного поля $(s+1) \times m$.

Если используется кольцевое вычислительное поле, размерность которого совпадает с размерностью блока, каждый процессор закрепляется за рассчитываемой точкой блока для всех уравнений системы. Для осуществления итераций по (2) в каждом i–ом процессоре, обеспечивающем вычисления s точечным методом, должны быть размещены соответствующие коэффициенты $a_{i,j}, b_i$, i=1,2,…,s, $j = 1,2,...,s$, а также значения элементов вектора правых частей системы в последней точке предшествующего блока, которые будут считаться начальными для следующего. Для группы процессоров, которые обеспечивают решение s+1 точечным методом, соответствующие коэффициенты $a_{i,j}, b_i$, $i=1,2,\ldots,s+1$, $j = 1,2,...,s+1$.

В случае, если используется решетка процессоров, позиция каждого элемента в строке определяет порядковый номер уравнения в системе, для которого ведется расчет, а номер элемента в столбце определяет рассчитываемую точку блока. Поскольку вычисления проводятся для блоков точек, которые расположены регулярно, есть возможность сопоставления решений, полученных методами с разными порядками точности в s совпадающих расчетных точках $t_{n,1}, t_{n,2}, ..., t_{n,s}$. Если норма вектора расхождений не превосходит заданную глобальную точность вычислений $\|u_{n,i} - \nu_{n,i}\| \le tol, i = 1,2,...,s$, за основу берется решение, полученное коллокационным блочным s+1- точечным методом $\nu_{n+1,i}$, $i=1,2,\ldots,s+1$.

Рассчитывается новое значение шага τ_{new}

$$\tau_{new} = \tau_{n+1} \min\left(\text{fax max}, \max\left(\text{fax min}, \omega\left(\frac{\varepsilon}{\|u_{i,i} - \nu_{i,i}\|}\right)^{\frac{1}{s+1}}\right)\right),$$

и по результатам расчета s+1- точечным методом формируется новый вектор опорных точек. Если норма вектора расхождений не превосходит заданную локальную точность вычислений, т.е. $tol < \left\| u_{n,i} - \nu_{n,i} \right\| < \varepsilon, i = 1, 2, ..., s$, за основу берется решение, полученное коллокационным блочным s-точечным методом $u_{n,i}$, i=1,2,…,s. Новый шаг τ_{new} и новый вектор опорных точек формируется по результатам расчета s - точечным методом. Если полученное решение не обеспечивает заданную локальную точность, от шага необходимо отказаться, сократив его на величину, задаваемую параметром faxmin. В качестве максимальных коэффициентов увеличения шага facmax и уменьшения faxmin, параметра ω, а также начальной длины шага принимаются значения, описанные в [9 - 10].

3. Управление шагом при интегрировании многошаговыми блочными методами

Параллельное управление шагом для задачи (1) многошаговым коллокационным блочным методом (3) с числом опорных точек m и расчетных s не будет иметь принципиальных различий с подходами, рассмотренными в п. 2. Однако возникает необходимость в формировании начальных данных для расчета очередного блока значений.

$$u_{n,j} = u_{n,0} + \tau \sum_{i=1}^{m} b_{i,j} F_{n-1,i}^{(0)} + \tau \sum_{l=0}^{p_j} \sum_{i=1}^{s} \tau^l a_{i,j}^{(l)} F_{n,i}^{(l)}. \quad (3)$$

Для выполнения следующего шага нужно обеспечить доступ каждого процессора к значениям правых частей в опорных точках блока. Кроме того, для осуществления итераций по (3) в каждом i–ом процессоре, обеспечивающем вычисления s точечным m шаговым методом, должны быть размещены соответствующие коэффициенты

$$a_{i,j}, b_{i,l},\ i=1,2,\ldots,s,\quad j = 1,2,...,s, l = 1,2,...,m,$$

а также значения элементов векторов правых частей системы в m точках предшествующего блока, которые будут считаться начальными для следующего. Для группы процессоров, которые формируют решение s+1 точечным методом, соответствующие коэффициенты

$$a_{i,j}, b_{i,l},\ i=1,2,\ldots,s+1,\ j = 1,2,...,s+1, l = 1,2,...,m.$$

Поскольку максимальный порядок p аппроксимации рассматриваемого m-шагового s-точечного разностного метода, который равен $2m + s - 1$, не обеспечивает абсолютной устойчивости метода [11], при генерации расчетных коэффициентов необходимо сократить предельное значение, введя параметр q<p. Тогда, в зависимости от значения нормы вектора расхождений, будет изменяться расчетная схема определения шага интегрирования. Если норма вектора расхождений не превосходит заданную глобальную точность вычислений

$$\left\| u_{n,i} - \nu_{n,i} \right\| \le tol, i = 1, 2, ..., s,$$

за основу берется решение, полученное коллокационным блочным s+1 - точечным методом $\nu_{n+1,i}$, i=1,2,…,s+1.

Рассчитывается новое значение шага τ_{new} с поправкой на порядок q аппроксимации рассматриваемого m-шагового s-точечного разностного метода, и по результатам расчета s+1-точечным m-шаговым методом формируется новый вектор опорных точек. Если норма вектора расхождений не превосходит заданную локальную точность вычислений, т.е. $tol < \left\| u_{n,i} - \nu_{n,i} \right\| < \varepsilon, i = 1, 2, ..., s$, за основу берется решение, полученное коллокационным блочным s - точечным m - шаговым методом. Рассчитывается новое значение шага τ_{new}, и новый вектор опорных точек формируется по результатам расчета s-точечным m-шаговым методом. Если полученное решение не обеспечивает заданную локальную точность, происходит отказ от шага.

4. Реализация тестовых задач на основе алгоритмов управления, основанных на блочных методах

В качестве тестовой рассматривается задача [12]

$$\begin{aligned} x_1' &= -x_1 + x_2, \\ x_2' &= -1000x_1 - x_2, \\ x_3' &= -100x_3 + x_4, \\ x_2' &= -10000x_3 - 100x_4 \end{aligned} \quad (4)$$

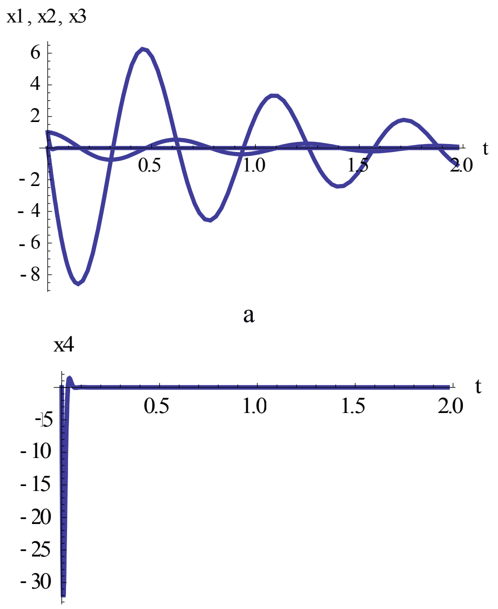
на интервале [0, 2] с начальными условиями:

$$x_1(0) = 1, x_2(0) = 0, x_3(0) = 1, x_4(0) = 0.$$

На графиках (рис. 2 а, б) показано поведение численного решения и распределения глобальных погрешностей. Разделение компонент x_1, x_2, x_3 и x_4 выполнено из-за принципиально разного масштаба решения.

Из численного решения видно, что пограничный слой этой системы расположен в начале интервала интегрирования, этим объясняются и малые размеры шагов, полученные с помощью параллельной реализации алгоритма управления шагом, построенного на основе одношаговых коллокационных блочных методов (рис. 3). Данные вычислительного эксперимента говорят о том, что автоматическое управление длиной шага и порядком параллельного коллокационного блочного метода действительно работает на практике и позволяет решать задачу (1) за короткое время и с приемлемой точностью.

Также в работе рассматривались реализации на основе одношаговых коллокационных блочных методов с числом расчетных точек 2 и 3, и многошаговых с числом опорных точек 2 и расчетных 3 и 4 (рис. 4).



б

Рис. 2. Численное решение системы (4)

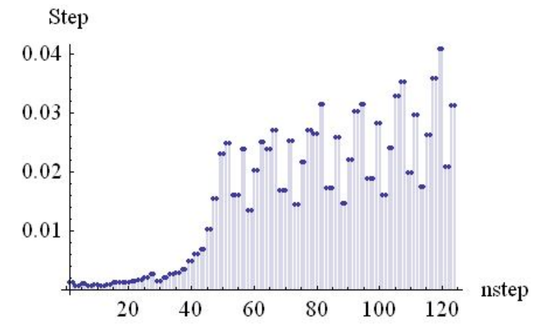


Рис. 3. Вариация шага для системы (4) с числом расчетных точек 2 и 3

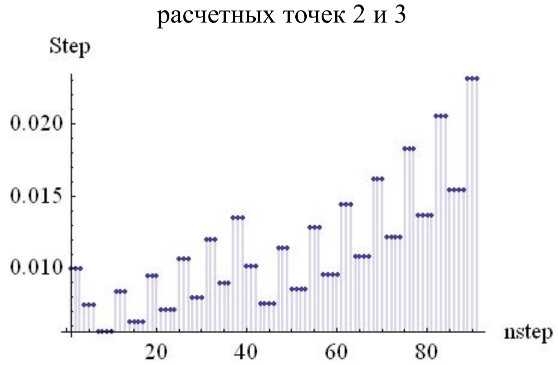


Рис. 4. Вариация шага для системы (4) с числом расчетных точек 3 и 4

Заключение

В работе приведены параллельные алгоритмы управления шагом интегрирования для решения жестких систем, основанные на коллокационных одношаговых и многошаговых блочных методах со старшими производными. В разработанных алгоритмах локальные погрешности определяются и сопоставляются во всех коллокационных точках блока, в отличие от стадийных методов, где стадии, как правило, не совпадают, и сравнение ведется только по одной конечной расчетной точке. Параллельный счет осуществляется в пределах одного цикла для всех точек блоков с размерностями s и s+1. Две нити вычислений реализовываются независимо, и необходимость в обменах возникает только после получения конечных результатов для блоков расчетных точек. Построение процедуры управления шагом позволяет продвигаться сразу на s или на s+1 расчетную точку в зависимости от нормированных величин расхождений, полученных при формировании очередного блока, что обеспечивает соответствующую величину ускорения даже при последовательной реализации. Введение дополнительных производных позволило значительно повысить порядок аппроксимации, не прибегая к наращиванию размерности системы, как это происходит при использовании многостадийных методов.

На известных тестовых задачах выполнена параллельная реализация всех предложенных в разделе алгоритмов управления шагом. В качестве тестовых выбирались плохо обусловленные, быстроосциллирующие, жесткие задачи. Также численные эксперименты проводились на тестах, позволяющих сформировать систему обыкновенных дифференциальных уравнений любой размерности. Параллельное управление шагом осуществлялось на двух линейках процессорных элементов. В качестве размера шага выбирался максимально возможный, обеспечивающий с учетом гарантийных факторов локальную точность. Полученные характеристики параллелизма свидетельствуют о высоких скоростных свойствах разработанных методов и способности получать решение с заданной точностью.

Литература

1. The Networking and Information Technology Research and Development Program fy 2013. [Электронный ресурс] // Reports National Coordination Office for Networking and Information Technology. – 2013. – Режим доступа: http://www.nitrd.gov/pubs /2013supplement/ FY13NITRDSupplement.pdf.

2. Grand Challenges: Science, Engineering and Societal Advances Requiring Networking and Information Technology [Электронный ресурс]// Report by Interagency Working Group on Information the Technology Research and Development. - Режим доступа: http://www.grandchallenges.org/ Pages/2011_grand_challenges.pdf.

3. FOSER - Future of Software Engineering Research Workshop. [Электронный ресурс] // Reports Federally Funded Research and Development in

Networking and Information Technology. – 2011. – Режим доступа: http://www.nitrd.gov/SUBCOMMITTEE/sdp/foser/FOSER+December+2011.pdf&pli=1.

4. Суперкомпьютерные технологии в науке, образовании и промышленности [Текст] / под ред. акад. В. А. Садовничего. – М. : Изд. МГУ, 2012. – 232 с.

5. Воеводин, В. В. Параллельные вычисления [Текст] / В. В. Воеводин, Вл. В. Воеводин. – СПб. : BHV-Санкт-Петербург, 2002. – 608 с.

6. Dmitrieva, O. Parallel Algorithms of Simulation. Increase of simulation of dynamic objects with the lumped parameters into parallel computer systems [Text] / O. Dmitrieva, A. Firsova. – Lambert Academic Publishing, 2012. – 192 p.

7. Дмитрієва, О. А. Паралельні різницеві методи розв'язання задачі Коші [Текст] / О. А. Дмитрієва. – Донецьк : ДонНТУ, 2011. – 265 с.

8. Дмитриева, О. А. Высокоэффективные алгоритмы управления шагом на основе параллельных коллокационных блочных методов [Текст] / О. А. Дмитрієва // Искусственный интеллект. – 2012. – № 4. – С. 77–88.

9. Dmitrieva O. Parallel Step Control. Development of parallel algorithms of the step variation for simulation of stiff dynamic systems [Text] / O. Dmitrieva, L. Feldman. – Lambert Academic Publ., 2013. – 72 p.

10. Дмитриева, О. А. Разработка параллельных алгоритмов управления шагом на основе вложенных стадийных методов [Текст] / О. А. Дмитриева // Наукові праці ДонНТУ. Серія «Проблеми моделювання та автоматизації проектування динамічних систем». – 2012. – № 1(10) –2(11). – С. 22–31.

11. Дмитриева, О. А. О модификации многошаговых коллокационных блочных методов при параллельном моделировании динамических объектов [Текст] / О. А. Дмитриева // Системи обробки інформації. – 2013. – № 14 (177). – С. 121-126.

12. Залеткин, С. Ф. Коллекция дифференциальных уравнений для тестирования вычислительных алгоритмов и программ [Текст] / С. Ф. Залеткин // Вопросы конструирования библиотек программ ; под ред. Е. А. Гребеникова, В. А. Морозова, О. Б. Арушаняна. – М. : Изд-во МГУ, 1983. – С. 54-71.

Поступила в редакцию 14.02.2014, рассмотрена на редколлегии 24.03.2014

Рецензент: д-р техн. наук, проф., проф. кафедры прикладной математики и информатики Л. П. Фельдман, Донецкий национальный технический университет, Донецк, Украина.

КЕРУВАННЯ КРОКОМ ІНТЕГРУВАННЯ ПРИ ПАРАЛЕЛЬНІЙ РЕАЛІЗАЦІЇ УЗАГАЛЬНЕНИХ КОЛОКАЦІЙНИХ БЛОКОВИХ МЕТОДІВ

О. А. Дмитрієва

У роботі розглядаються питання побудови паралельних алгоритмів керування кроком інтегрування при розв'язанні задачі Коші для систем звичайних диференціальних рівнянь. Алгоритми будуються на багатокрокових колокаційних блокових методах зі старшими похідними, що забезпечує векторизацію процедури одержання розв'язку. Запропоновано паралельні алгоритми керування кроком і порядком інтегрування для розв'язання жорстких систем, засновані на колокаційних однокрокових і багатокрокових блокових методах зі старшими похідними. У розроблених алгоритмах локальні похибки визначаються й зіставляються у всіх колокаційних точках блоку, на відміну від стадійних методів, де стадії, як правило, не збігаються, і порівняння ведеться тільки по одній кінцевій розрахунковій точці. Обчислення реалізуються незалежно, і необхідність в обмінах виникає тільки після одержання кінцевих результатів для блоків розрахункових точок.

Ключові слова: задача Коші, крок інтегрування, показники паралелізму, блок, точки колокації, порядок апроксимації

INTEGRATION STEP CONTROL IN PARALLEL IMPLEMENTATION OF GENERIC COLLOCATION BLOCK METHODS

O. A. Dmitrieva

The issues of parallel algorithms of the integration step control for solving the Cauchy problem for systems of ordinary differential equations are considered in the paper. The algorithms are based on collocation multistep block methods with higher derivatives, which provides vectoring of the procedures for obtaining solutions. The parallel algorithms for controlling the step and order of integration for solving stiff systems based on one-step and multistep collocation block methods with higher derivatives are proposed. In the developed algorithms the local errors are detected and compared in all collocation points of the block, in contrast to phasic methods wherein steps generally do not coincide, and the comparison is carried out only on one calculation end point. Calculations are implemented independently, and the need for exchanges occurs only after the final results for blocks of the calculation points.

Key words: Cauchy problem, integration step, rates of parallelism, the block, collocation points , the order of approximation

Дмитриева Ольга Анатольевна – д-р техн. наук, профессор кафедры прикладной математики и информатики Донецкого национального технического университета, Донецк, Украина.