

УДК 004.932.4

ЭКСТРАПОЛЯЦИОННЫЕ БЛОЧНЫЕ ОДНОШАГОВЫЕ МЕТОДЫ ЧИСЛЕННОГО ВЫСОКОТОЧНОГО РЕШЕНИЯ ЗАДАЧИ КОШИ

Кулаков В.В., Назарова И. А., Фельдман Л.П.

Донецкий национальный технический университет

Рассматриваются параллельные экстраполяционные алгоритмы решения нелинейной задачи Коши для систем обыкновенных дифференциальных уравнений, позволяющие получить высокоточное решение жестких задач. Разработанные алгоритмы реализованы на параллельных системах с распределенной памятью и топологией гиперкуб. Получены оценки времени выполнения и обменов, ускорения и эффективности параллельного решения.

Введение

Экстраполяционные методы Ричардсона предназначены для получения высокоточного решения задачи Коши и интегрирования систем обыкновенных дифференциальных уравнений (СОДУ) со сложными правыми частями. Практическое применение экстраполяционных методов затруднено в связи с большой вычислительной сложностью их последовательных реализаций. Кроме того, использование технологии локальной экстраполяции на основе явных опорных схем ограничивает использование методов областью нежестких задач. Поэтому построение эффективных параллельных блочных неявных методов локальной экстраполяции Ричардсона – один из наиболее реальных способов сокращения времени интегрирования многомерных жестких начальных задач.

Целью данной работы является разработка параллельных блочных экстраполяционных одношаговых методов численного решения жестких задач Коши и исследование эффективности их отображения на параллельные вычислительные структуры с распределенной памятью.

Решение задачи Коши для систем обыкновенных дифференциальных уравнений первого порядка:

$$\begin{cases} \frac{d\bar{y}(x)}{dx} = \bar{f}(x, \bar{y}(x)), \\ \bar{y}(x_0) = \bar{y}_0, \end{cases} \quad (1)$$

рассматривается при переходе из точки x_n в точку $x_{n+1} = x_n + H$, где H – базовая длина шага. Выбирается ряд натуральных чисел $P_i = \{n_1, n_2, \dots, n_k, \dots\}$, $n_1 < n_2 < \dots < n_k < \dots$ и, соответственно, последовательность шагов интегрирования: $h_1 > h_2 > \dots > h_k > \dots$, где $h_i = H / n_i$. Задается опорный численный метод порядка r_0 и вычисляются приближенные решения исходной задачи Коши в точке x_{n+1} : $T_{i,l} = \bar{y}_{h_i}(x_n + H) = \bar{y}_{n+1}^{(i)}$, $i = \overline{1, k}$. Выполнив вычисления для ряда последовательных значений i , по рекуррентному соотношению, определяют значения для произвольных i, j по схеме локальной полиномиальной экстраполяции Эйткена-Невилла (рис. 1):

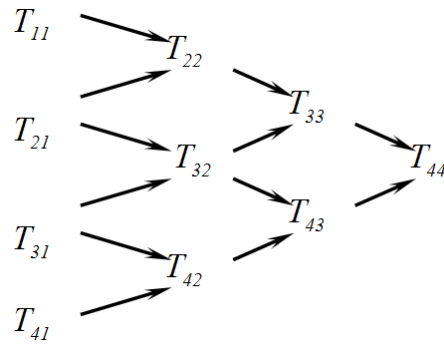


Рисунок 1 – Вычисления по схеме полиномиальной экстраполяции Эйткена-Невилла

$$T_{i,j+1} := T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{(n_i / n_{i-j})^b - 1}$$

Эффективный с точки зрения минимизации вычислительных затрат последовательный метод с использованием технологии Ричардсона описан в [1-2]. Потенциально вычисления по технологии локальной экстраполяции содержат три источника внутреннего параллелизма:

- системный параллелизм (ограничен размерностью СОДУ, m);
- параллелизм экстраполяции (ограничен размером таблицы экстраполяции, k);
- параллелизм опорного метода (малая степень параллелизма).

Реализация технологии локальной экстраполяции Ричардсона для блочных методов требует многократных вычислений на одном и том же интервале интегрирования с использованием опорного блочного k_0 -точечного метода порядка r_0 на сгущающихся равномерных сетках:

- 1) а) $\Omega_{h/n_1} = \{x_j\}, j = \overline{1, M_1}$ – с шагом $h_1 = h / n_1$ в N_1 блоках;
- 2) б) $\Omega_{h/n_2} = \{x_j\}, j = \overline{1, M_2}$ – с шагом $h_2 = h / n_2$ в N_2 блоках;
-
- 3) в) $\Omega_{h/n_k} = \{x_j\}, j = \overline{1, M_k}$ – с шагом $h_k = h / n_k$ в N_k блоках, где k – число строк экстраполяционной таблицы.

Базовый шаг интегрирования равен $h_1 = h, n_1 = 1$, то есть основой счета является сетка: Ω_h . Для генерации вспомогательных сеток выбирается гармонический ряд $P_1 = \{n_2, \dots, n_k, \dots\} = \{2, 3, 4, 5, \dots\}$ как наименее затратный в случае опорного метода произвольного типа. Блочный опорный метод должен иметь малый порядок точности:

$$T_{11,i} = y_{n,i} = y_{n,0} + ih \left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j} \right]; i = \overline{1, k_0}, k_0 \leq 4 \quad (2)$$

так как с ростом r_0 вычислительные затраты на технологию в целом существенно возрастают. Для неявных методов, к которым относятся и рассматриваемые блочные методы, это особенно важно, поскольку увеличение порядка, а, следовательно, и числа точек блока, влечет за собой увеличение порядка многократно решаемых СНАУ. Заметим, что для блочных методов базовый интервал интегрирования – это некоторый блок с номером n основной сетки длины $H = k_0 h$. Общее число точек, вычисляемое с разными шагами интегрирования $h_i = h / i, i = \overline{1, k}$ и определяющие узлы сеток $i = \overline{1, k}$,

равно $M_i = H / h_i = k_0 i$, количество блоков i -той сетки $N_i = M_i / k_0 = i$ и $n_i = N_i, i = \overline{1, k}$.

Поскольку экстраполируются значения в узлах базовой сетки, необходимо установить механизм соответствия узлов сетки Ω_h основного счета узлам сеток для экстраполяции $\Omega_{h_i}, i = \overline{2, k}$. Для двухточечного опорного блочного метода и гармонического ряда имеем:

- 1) i – четное число: $x_{n,1}(h) \Leftrightarrow x_{n+\frac{i}{2}-1,2}(h_i)$;
- 2) i – нечетное число: $x_{n,1}(h) \Leftrightarrow x_{n+\lceil \frac{i}{2} \rceil,1}(h_i)$;
- 3) для любых i : $x_{n,2}(h) \Leftrightarrow x_{n+i-1,2}(h_i)$.

Значения первого столбца экстраполяционной таблицы равны:

$$\left\{ \begin{array}{l} T_{11,n,i} = y_{n,i}^{(l)} = y_{n,0}^{(l)} + ih \cdot \left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j} \right]; i = \overline{1, k_0}; n = \overline{1, N_1}, \\ \dots \\ T_{k1,n,i} = y_{n,i}^{(k)} = y_{n,0}^{(k)} + i \frac{h}{k} \cdot \left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j} \right]; i = \overline{1, k_0}; n = \overline{1, N_k}. \end{array} \right. \quad (3)$$

Каждая из аппроксимаций решения получается за счет N_i раз примененной схемы одношагового блочного k_0 -точечного метода с разными шагами интегрирования:

$$\left\{ \begin{array}{l} T_{T_{11}} = T_{\bar{y}_{n,i}^{(1)}(h)} = N_1 \cdot T_1^{r_0}(h), \\ \dots \\ T_{T_{k1}} = T_{\bar{y}_{n,i}^{(k)}\left(\frac{h}{k}\right)} = N_k \cdot T_1^{r_0}(h/k), \end{array} \right. \quad (4)$$

где $T_1^{r_0}(h/i), i = \overline{1, k}$ – время, необходимое для решения задачи Коши для ОДУ опорным методом порядка r_0 с шагом h_i . Затем по формуле Эйткена-Невилла вычисляются приближения T_{22}, T_{33}, \dots и $T_{k,k}$. Организация параллельных вычислений может производиться двумя способами. Первый вариант подразумевает использование только системного параллелизма, второй – комбинацию параллелизма экстраполяции и системного.

Рассмотрим различные варианты организации параллельных вычислений экстраполяционной таблицы по схеме Эйткена-Невилла при равномерном распределении [3-4]. После параллельного вычисления k аппроксимаций решения на k группах процессоров, имеется следующее распределение исходных данных для вычисления экстраполированных значений. Первая группа процессоров содержит вектор $T_{11,j}; j = \overline{1, m_1}$, вторая – $T_{21,j}; j = \overline{1, m_2}$ и, соответственно, k -тая – $T_{k1,j}; j = \overline{1, m_k}$, каждый процессор в группах содержит одинаковое m_i число компонент. Для вычислений элементов экстраполяционной таблицы могут быть реализованы два варианта. Первый состоит в использовании системного параллелизма, каждый процессор группы передает соответствующему процессору соседней группы все компоненты своего вектора аппроксимации решения. Так первый процессор первой группы передаст первому процессору второй группы подвектор: $T_{11,j}, j = \overline{1, m_1}$ для вычисления такого же подвектора экстраполированного значения: $T_{22,j}, j = \overline{1, m_2}$, второй процессор первой группы передает $T_{11,j}, j = \overline{m_1+1, 2m_1}$ второму процессору второй группы для

вычисления $T_{22,j}$, $j = \overline{m_2 + 1, 2m_2}$ и так далее, первый процессор $(k-1)$ группы передает первому процессору k группы значения подвектора $T_{k-1,1,j}$, $j = \overline{1, m_{k-1}}$ для вычисления $T_{k2,j}$, $j = \overline{1, m_k}$. Высота такого параллельного алгоритма равна: $k-1$. То есть на первом шаге вычисляются подвектора второй строки таблицы экстраполяции, на последнем $k-1$ шаге вычисляются соответствующие подвектора k -той строки таблицы экстраполяции.

Второй алгоритм вычисления экстраполяционной таблицы состоит в следующем: в каждой группе процессоров ровно один процессор будет отвечать за передачу данных между группами. Для этого после подсчета элементов первого столбца таблицы в каждой группе процессоров производится обмен значениями по типу “все-всем”, в результате каждый процессор в группе будет содержать весь вектор аппроксимации решения, а не его часть. Процессоры первой группы – \bar{T}_{11} , второй – \bar{T}_{21} и k -той – \bar{T}_{k1} . Затем i -тый процессор каждой группы кроме последней, передает всем процессорам соседней группы вектор решений для подсчета экстраполированного значения. Далее каждый процессор каждой группы кроме первой (ширина алгоритма уменьшается с каждым шагом) считает свой подвектор (длиной m_i) своего вектора экстраполированного значения: процессоры второй группы – \bar{T}_{22} , третьей – \bar{T}_{32} и k -той – \bar{T}_{k2} . На последнем $k-1$ шаге будут работать только процессоры последней группы, и каждый будет считать m_k значений вектора \bar{T}_{kk} . Чтобы увеличить балансировку загрузки разобьем процессорное поле на такое же число групп, но неравномерно, а пропорционально числам n_i , $i = \overline{1, k}$: $p_1 : p_2 : \dots : p_k = n_1 : n_2 : \dots : n_k$, $\sum_{i=1}^k p_i = p$, $p_i = \frac{p}{N(k)} \cdot n_i$. Алгоритм разбиения на группы следующий, если число процессоров не кратно $N(k)$, то берется $p_i = \left\lfloor \frac{p}{N(k)} \cdot n_i \right\rfloor$, а затем оставшиеся процессоры по одному добавляются в каждую из групп, начиная с последней. Таким образом, каждая группа будет содержать тем большее число процессоров, чем больше шагов интегрирования на базовом шаге ей предстоит выполнить для получения аппроксимации решения.

Из двух рассмотренных вариантов получения экстраполяционной таблицы, именно второй подходит для пропорционального разбиения процессоров, поскольку не требует дополнительной сортировки данных из-за неравномерности их распределения. В общем виде проблема разбиения процессорного поля на группы, обеспечивающие оптимальное решение задачи балансировки загрузки многопроцессорной системы сформулирована в терминах комбинаторной оптимизации. Точное решение задач этого класса за приемлемое время невозможно, так как задача является NP-сложной, поэтому предлагаются близкие к оптимальному рациональные решения приближенными эвристическими методами.

Сократить время простоя можно при использовании комбинационного способа. Идея комбинационного метода состоит в использовании $\lceil k/2 \rceil$ групп процессоров, причем i -тая группа вычисляет i -тую и $(k - i + 1)$ -ю аппроксимации решения. Для получения i -той аппроксимации необходимо n_i раз выполнить обращение к опорному методу решения, для $(k - i + 1)$ -й – n_{k-i+1} раз. То есть первая группа должна $n_1 + n_k$ раз обратиться к вычислению решения по опорному методу, вторая группа – $n_2 + n_{k-1}$ и, наконец, $k_1 = \lceil k/2 \rceil$ группа – $n_{k_1} + n_{k_1+1}$.

Для разработанных параллельных алгоритмов метода ЛЭР построены схемы

отображения на параллельные структуры разных топологий: кольцо, тор и гиперкуб. Проведенный сравнительный анализ эффективности использования топологий и вычислительный эксперимент дают основания утверждать, что менее затратными являются обмены в топологии гиперкуб.

Определяющим параметром для рассмотренных вычислительных схем методов локальной экстраполяции является сложность правой части исходной СОДУ, обеспечивающая доминирование вычислений над обменами. Для несложных правых частей третья вычислительная схема, объединяющая параллелизм системы и экстраполяции, обладает явным преимуществом, и этот эффект с ростом числа процессоров увеличивается. Для SIMD-систем, как и ранее, может быть реализована только первая схема, которая имеет низкие показатели ускорения и эффективности.

Выводы

Численный эксперимент на базе тестов для СОДУ и проведенный сравнительный анализ динамических характеристик параллельных алгоритмов по технологии локальной экстраполяции показал, что большая вычислительная сложность последовательных реализаций этих методов и высокая степень внутреннего параллелизма являются основанием для эффективного использования их в высокопроизводительных мультипроцессорных системах. Апробация параллельных алгоритмов проводилась на моделях ВС кластерной архитектуры с различными топологиями. В частности, наилучшие характеристики распараллеливания достигаются при использовании топологии гиперкуб. Особенно важным является тот факт, что четко выявлена область применения параллельных экстраполяционных методов, где они практически не имеют конкурентов – высокоточное решение задачи Коши для СОДУ со сложными правыми частями.

Перспективным направлением дальнейших исследований является применение технологии локальной экстраполяции для разработки параллельных методов решения жестких задач. Для достижения этой цели необходимо разработать алгоритмы с использованием опорных методов, имеющих достаточные характеристики устойчивости: среди одношаговых методов к таким методам относятся многоточечные блочные и полностью неявные методы типа Рунге-Кутты.

Список источников

- [1] Хайрер Э., Нёрсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи: Пер. с англ. – М.: Мир, 1990. – 512 с.
- [2] Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. – М.: Мир, 1999. – 685 с.
- [3] Назарова И.А. Эффективность применения технологии локальной экстраполяции в параллельных алгоритмах численного решения задачи Коши // Научно-теоретический журнал ИПИИ НАН Украины «Искусственный интеллект», № 3, 2006. – Донецк: ИПИИ, 2006. – С. 92-202.

- [4] Фельдман Л.П., Назарова И.А. Применение технологии локальной экстраполяции для высокоточного решения задачи Коши на SIMD-структурах // Научные труды Донецкого национального технического университета. Выпуск 70. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003) – Донецк: ДонНТУ, 2003. – С. 98-107.