

ЭФФЕКТИВНОСТЬ СПОСОБОВ ОЦЕНКИ АПОСТЕРИОРНОЙ ЛОКАЛЬНОЙ ПОГРЕШНОСТИ ПРИ ПАРАЛЛЕЛЬНОМ РЕШЕНИИ СИСТЕМ ЛИНЕЙНЫХ ОДНОРОДНЫХ ОДУ

Л.П.Фельдман , И.А.Назарова
Кафедра ПМИИ, ДонНТУ
nazarova@r5.dgtu.donetsk.ua

Фельдман Л.П., Назарова І.А. Ефективність засобів оцінки апостеріорної локальної похибки при паралельнім вирішенні лінійних однорідних ЗДР. В статті розглянута ефективність альтернативних засобів оцінки локальної похибки: правило Рунге, локальна екстраполяція та вкладені методи Рунге-Кутти при паралельному вирішенні систем лінійних однорідних диференціальних рівнянь. Розроблені обчислювальні схеми відображення методів на SIMD-структури з матричною топологією та розподіленою пам'яттю. Отримані порівняльні характеристики потенційного та реального паралелізму.

Введение

Моделирование реальных экономических, технических и других процессов, описываемых системами обыкновенных дифференциальных уравнений (СОДУ), представляет собой обширный класс задач, для решения которых применение высокопроизводительной вычислительной техники не только оправдано, но и необходимо. Об этом свидетельствует знаменитый список проблем “большой вызов”, в котором такие задачи занимают одно из ведущих мест[1]. Как показала практика, наиболее эксплуатируемым способом создания параллельных методов является распараллеливание хорошо исследованных и многократно апробированных последовательных численных алгоритмов[2]. Современный качественный численный алгоритм решения СОДУ обязательно должен содержать механизм управления шагом интегрирования на основе информации о погрешности решения на каждом шаге интегрирования[3].

В данной статье рассматривается эффективность альтернативных методов оценки апостериорной локальной погрешности на базе параллельных алгоритмов решения линейных однородных СОДУ с постоянными коэффициентами для многопроцессорных ВС SIMD-архитектуры с распределенной памятью.

1.Общая характеристика способов оценки апостериорной локальной погрешности

Известными методами оценки локальной погрешности решения СОДУ являются: правило Рунге; вложенные методы Рунге-Кутты (ВМРК); метод локальной экстраполяции Ричардсона.

Наиболее простой способ достижения поставленной цели – это удвоение или дублирование шага по правилу Рунге. На основе одной и той же формулы Рунге-Кутты порядка p вычисляют два приближения к решению в одной и той же точке сначала с удвоенным шагом: $2h$, а затем независимо, дважды с половинным шагом: h . Для точного решения при переходе из точки x_n к точке $x_n+2\cdot h, y(x_n+2\cdot h)$ получают две аппроксимации y_1 (1 шаг $2\cdot h$) и y_2 (2 шага h). Разность между этими значениями используется в качестве оценки локальной погрешности интегрирования, в качестве решения принимается аппроксимация, полученная с половинным шагом, как наиболее точная:

$$\begin{aligned} y(x+2h) &\cong y_1 + (2h)^p \varphi + O(h^p) \\ y(x+2h) &\cong y_2 + 2(h^p) \varphi + O(h^p) \\ \Delta y &= y_2 - y_1 \\ y(x+2h) &\cong y_3 = y_2 + \frac{\Delta y}{2^p - 1} + O(h^{p+1}). \end{aligned} \quad (1)$$

Процесс уточнения решения с половинным шагом y_2 , как видно из (1), повышает точность решения на единицу и получил название локальной экстраполяции. Данный способ оценки погрешности достаточно прост, однако имеет большие накладные расходы: объем вычислений на один узел сетки возрастает почти втрое.

Второй способ оценки локальной погрешности метода – вложенные методы Рунге-Кутты. Этот способ основан на использовании двух приближенных значений решения в одной точке, но в отличие от правила Рунге приближения вычисляются не по одной, а по двум формулам различных порядков точности p и \hat{p} с одним и тем же шагом [3-4]:

$$\begin{cases} y_{n+1} = y_n + h_n \cdot \sum_{l=1}^s b_l \cdot k_l; \\ \hat{y}_{n+1} = y_n + h_n \cdot \sum_{l=1}^{\hat{s}} \hat{b}_l \cdot k_l; \\ k_l = f(x_n + c_l \cdot h_n; y_l + h_n \cdot \sum_{i=1}^{l-1} a_{li} \cdot k_i); l=1, \dots, s; \\ d_{n+1} = |(\hat{y}_{n+1} - y_{n+1})| \end{cases} \quad (2)$$

Для определения локальной погрешности менее точного результата и управления величиной шага интегрирования используется величина d^{n+1} . Вложенный метод Рунге-Кутты $p(\hat{p})$ – это схема, в которой метод

\hat{p} – го порядка (“оценщика погрешности”) получается как побочный продукт метода p -го порядка. Порядок аппроксимаций: y_n и \hat{y}_n обычно отличаются на 1, т.е. $p = \hat{p} + 1$ или $p = \hat{p} - 1$. Оба приближения используют практически одни и те же значения шаговых коэффициентов $k_i, i = 1, \dots, s$, но комбинируют их по-разному. Это позволяет уменьшить количество вычислений функции $f(x, y)$. Для s - стадийных вложенных методов эта величина составит $s + c$, где c – некоторая константа (обычно $c = 1$ или $c = 2$). Для сравнения приведем эти оценки для $p = 4$: накладные расходы для метода Рунге-Кутты с правилом Рунге, а именно число вычислений правой части равно 11, а для вложенного метода Фельберга того же порядка точности всего 6.

Метод локальной экстраполяции Ричардсона является обобщением технологии удвоения шага по правилу Рунге [3-5]. Идея этого метода заключается в многократном измельчении шага интегрирования, и также в многократном применении процесса вычисления, названного локальной экстраполяцией. Решение задачи Коши рассматривается при переходе из точки x_n в точку $x_{n+1} = x_n + H$, H – базовая длина шага, $H > 0$. Выбирается ряд натуральных чисел такой, что: $n_1 < n_2 < \dots < n_{k-1} < n_k < \dots$ и, соответственно, последовательность шагов: $h_1 > h_2 > \dots > h_{k-1} > h_k > \dots$, где $h_i = H / n_i$. Задается опорный численный метод порядка p и, выполняя n_i шагов интегрирования длиной h_i , вычисляют приближенное решение исходной задачи:

$$T_{i,l} := y_{h_i}(x_0 + H). \quad (3)$$

Выполнив вычисления для ряда последовательных значений i , по рекуррентному соотношению (2) определяют экстраполированные значения $T_{i,j+1}$ для произвольных i, j по формуле (4). Этот процесс получил название локальная полиномиальная экстраполяция:

$$T_{i,j+1} := T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{(n_i / n_{i-j})^b - 1}. \quad (4)$$

Здесь величина b равна единице в общем случае, в тоже время для симметричных опорных методов, имеющих разложение погрешности по степеням h^2 , b равно двум (каждая экстраполяция исключает две степени h вместо одной).

Таблица 1 – Экстраполяционная таблица

p	$p+b$	$p+2b$		$p+(k-2)b$	$p+(k-1)b$
T_{11}					
T_{21}	T_{22}				
T_{31}	T_{32}	T_{33}			
....	$T_{k-1,k-1}$	
T_{k1}	T_{k2}	T_{k3}	...	T_{k-1k}	T_{kk}

В таблице (1) T_{ij} - есть приближенное решение задачи Коши, полученное численным методом порядка $p+(j-1) \cdot b$ с шагом h_j . Величина $T_{k,k}$ соответствует аппроксимации наивысшего порядка, равного $2k$, в случае, если вычислены первые k строк экстраполяционной таблицы, а величина $T_{k-1,k}$ соответствует аппроксимации порядка $2k-2$. Для управления шагом интегрирования естественно использовать выражение:

$$\|T_{k-1,k} - T_{k,k}\|.$$

Таким образом, экстраполяционная технология Ричардсона включает численный метод решения задачи Коши, последовательность сеток, рекуррентное правило вычисления значений приближенного решения. Эффективность применения технологии локальной экстраполяции напрямую зависит от правильного выбора и сочетания всех трех составляющих этого метода.

2. Анализ вычислительной сложности последовательных методов решения СЛОДУ с постоянными коэффициентами

Исследование эффективности предложенных альтернативных способов оценки локальной погрешности проводилось на следующем множестве методов одного и того же порядка p :

- 1 – явный метод Рунге-Кутты и правило Рунге;
- 2 – экспоненциальный метод и правило Рунге;
- 3 – вложенные методы Рунге-Кутты;
- 4 – вложенные методы на основе экспоненциального;
- 5 – метод Грэгга-Булирша-Штера;
- 6 – экспоненциальный метод с локальной экстраполяцией.

Экспоненциальный метод относится к специальным методам численного интегрирования линейных СОДУ с постоянными коэффициентами, основанных на точном представлении решения в аналитической форме и вычислении матричной экспоненты.

Точным решением задачи Коши вида:

$$\begin{cases} y' = A \cdot y \\ y(x_0) = y_0 \end{cases} \quad (6)$$

является матричная экспонента

$$y(x_0+h) = e^{Ah} \cdot y_0, \quad e^{Ah} = \sum_{k=0}^{\infty} \frac{(hA)^k}{k!}.$$

Приближенное решение (6) можно построить, аппроксимировав матричную экспоненту отрезком ряда Тейлора :

$$e^{Ah} \approx \sum_{k=0}^p \frac{(hA)^k}{k!}, \quad (7)$$

причем, (7) определяет численный метод решения (6) порядка p [6].

Наиболее эффективным последовательным методом, реализующим технологию локальной полиномиальной экстраполяции считается алгоритм Грегга-Булирша-Штера(ГБШ), базирующийся на модифицированном методе средней точки [3]:

$$\begin{aligned} \bar{y}_1 &= \bar{y}_0 + h_k \cdot \bar{f}(x_0, \bar{y}_0) \\ \bar{y}_{i+1} &= \bar{y}_{i-1} + 2h_k \cdot \bar{f}(x_i, \bar{y}_i), i=1, \dots, n_k - 1 \\ \bar{y}(x+H) &\cong \frac{1}{2} [\bar{y}_{n_k} + \bar{y}_{n_k-1} + h_k \cdot \bar{f}(x+H, \bar{y}_{n_k})] h_k = H/n_k. \end{aligned} \quad (8)$$

На рисунке 1 приведены оценки вычислительных затрат при последовательной реализации множества алгоритмов 1–6. При этом предполагалось, что время выполнения любых арифметических операций является одинаковым и полагается равным 1 в тех или иных единицах измерения.

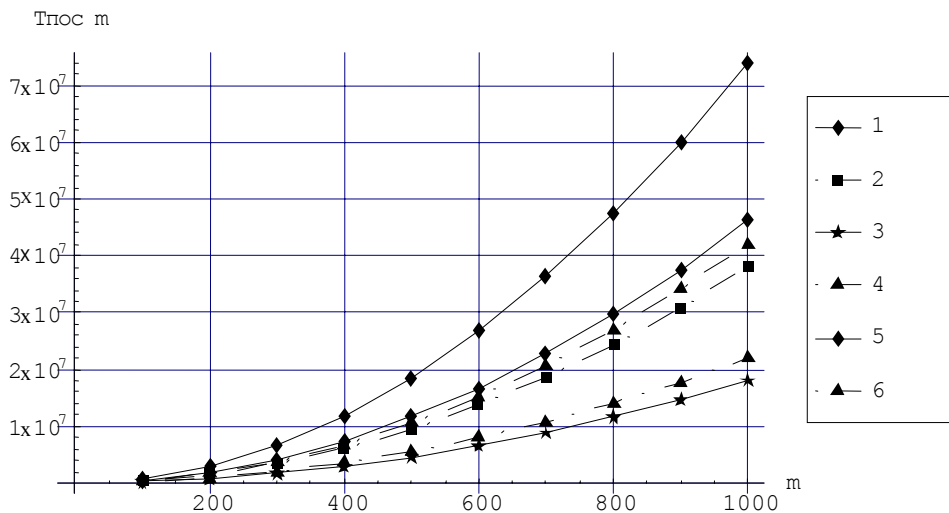


Рисунок 1-График зависимости вычислительной сложности последовательных алгоритмов от размерности задачи

Наиболее трудоемким из рассматриваемых алгоритмов является метод ГБШ, число арифметических операций для него имеет порядок $O([p^2 + p + 2] \cdot m^2)$, где m – размерность системы уравнений, а p – порядок метода. Наименее трудоемким является метод вложенных форм Рунге-

Кутты: $O([2p+2] \cdot m^2)$ и вложенный метод на основе экспоненты: $O([2p+6] \cdot m^2)$.

3. Анализ эффективности параллельных алгоритмов решения СЛОДУ с постоянными коэффициентами с учетом локальной погрешности

Рассмотрим отображение алгоритмических схем приведенных методов на многопроцессорные вычислительные системы SIMD-структуры с распределенной памятью. Конфигурацию системы считаем фиксированной: число процессорных элементов и схема их соединения не изменяются в процессе счета. Каждый процессор может выполнить любую арифметическую операцию за один такт, временные затраты, связанные с обращением к памяти отсутствуют.

Параллельная реализация перечисленных методов требует распараллеливания следующих базовых операций: матричное и векторное умножение и сложение, а также умножение вектора и матрицы на скаляр. Наиболее оптимальное топологическое решение – это квадратная сетка $m \times m$ или ее замкнутый эквивалент – тор. На такой топологической схеме достаточно эффективно выполняются матричные операции. Для простоты изложения рассмотрим случай, когда количество процессорных элементов в строке или столбце матрицы совпадает с размерностью задачи, m . Вычисление матричного умножения и умножения матрицы на вектор может быть выполнено по систолическому алгоритму, который является наиболее эффективным для SIMD-систем [7]. Процесс выполнения систолического умножения матриц состоит из предварительного косоого сдвига левого сомножителя по строкам, влево и правого - вверх, по столбцам. Затем пошагово выполняются: m умножений элементов, $(m-1)$ - одиночный сдвиг и $(m-1)$ сложение:

$$T_{AxA}^{SYS} = mt_{ум} + (m-1)t_{сл} + 3(m-1)t_{сд}, \quad (11)$$

где $t_{ум}, t_{сл}, t_{сд}$ - времена выполнения одиночных операций умножения, сложения и сдвига. Вычисление систолического умножения матрицы на вектор на базе алгоритма сдваивания, подробно описано в [7] и требует следующего времени выполнения:

$$T_{AxY}^{SYS} = t_{ум} + (m + \hat{m} - 2)t_{сд} + \log_2 \hat{m} t_{сл}, \quad (12)$$

где $\hat{m} = 2^l - 1$ и $l = \lceil \log_2 m \rceil$.

Вычислительные схемы параллельных алгоритмов с учетом отображения на структуру многопроцессорной ВС получены с помощью аппарата графов влияния [2]. В рамках статьи приведем результаты анализа эффективности использования потенциального параллелизма (без учета обменных операций) и реального (с учетом операций обмена между

процессорами). На рисунке 2 приведены графики зависимости количества арифметических операций от размерности исходной системы уравнений. Наименее трудоемкими оказались экспоненциальный метод с оценкой погрешности по правилу Рунге, для которого число арифметических операций имеет порядок: $O(3 \cdot \text{Log}_2 m)$ и вложенный-экспоненциальный метод: $O(2 \cdot \text{Log}_2 m)$. Худшие показатели потенциального параллелизма у метода Рунге-Кутты с правилом Рунге и метода ГБШ.

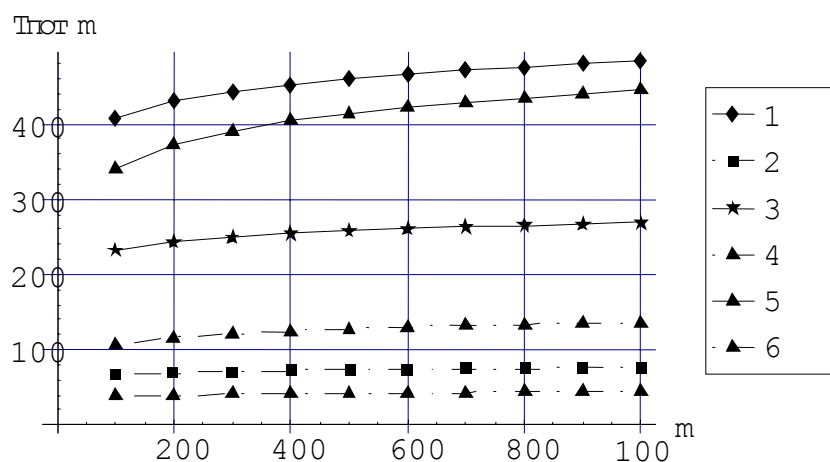


Рисунок 2 – График зависимости потенциального параллелизма от размерности системы

Общепризнанным приемом повышения эффективности параллельных вычислений служит сокращение межпроцессорных обменов данными. Уменьшение времени выполнения обменных операций возможно, как за счет повышения пропускной способности линков, так путем целенаправленного программирования решаемых задач, ориентированного на повышение отношения времени обработки к времени передачи данных в ВС. Эта задача тем более актуальна в связи с тем, что скорость обработки данных в микропроцессорах растет существенно быстрее увеличения пропускной способности интерфейсов микропроцессоров [2]. Оценки трудоемкости обменных операций для исследуемых алгоритмов приведены в таблице 1. Наибольшей коммуникационной сложностью обладает метод ГБШ, наименьшей - вложенный экспоненциальный и экспоненциальный на базе локальной экстраполяции.

Таблица 1– Характеристика алгоритмов по трудоемкости обменных операций

Параллельный алгоритм	Количество обменных операций
1.Метод Рунге-Кутты и правило Рунге	$(2 \cdot m - 2)(3 \cdot p - 1)$
2.Экспоненциальный метод и правило Рунге	$9(m - 1)$
3.Вложенные методы Рунге-Кутты	$\mathbf{B}F_{n-1} + \mathbf{A}f_n^{(s)} - u_n^{(s)}$
4.Вложенный экспоненциальный метод	$6(m - 1)$
5.Метод ГБШ	$3(m - 1)(p^2 + 2p + 4)/4$
6.Экспоненциальный метод с локальной экстраполяцией Ричардсона	$(m - 1)(p - 1)$

Анализ зависимости отношения времени обменов ко времени выполнения арифметических операций приведен на рисунке 3.

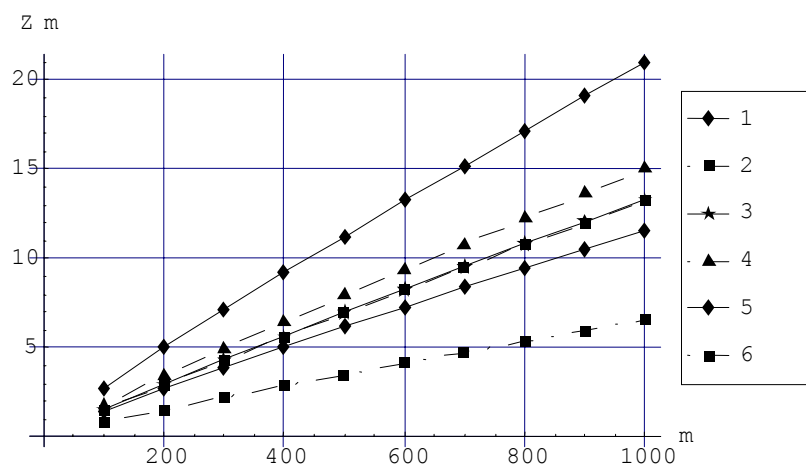


Рисунок 3 – График зависимости отношения обменных операций к арифметическим от размерности задачи

В целом характеристики реального параллелизма:

1)лучшие методы:

- 6-экспонента и локальная экстраполяция : $O(0.1m)$;
- 4- вложенная экспонента: $O(0.6m)$;

2)худший метод – метод ГБШ: $O(0.3 \cdot [p^2 + 2p + 4] \cdot m)$.

В качестве показателей эффективности параллельных алгоритмов применялись такие характеристики, как коэффициенты ускорения и эффективности. Коэффициент ускорения, получаемого при использовании

параллельного алгоритма для p процессоров, по сравнению с последовательным вариантом выполнения вычислений определяется:

$$K_p(m) = T_1(m) / T_p(m),$$

как отношение времени решения на скалярной ЭВМ: $T_1(m)$ к времени выполнения параллельного алгоритма: $T_p(m)$. Величина m - параметр, количество входных данных задачи, в данном конкретном случае порядок СОДУ.

На рисунке 4 приведены коэффициенты ускорения для оценки реального параллелизма с учетом обмена, как функции размерности задачи.

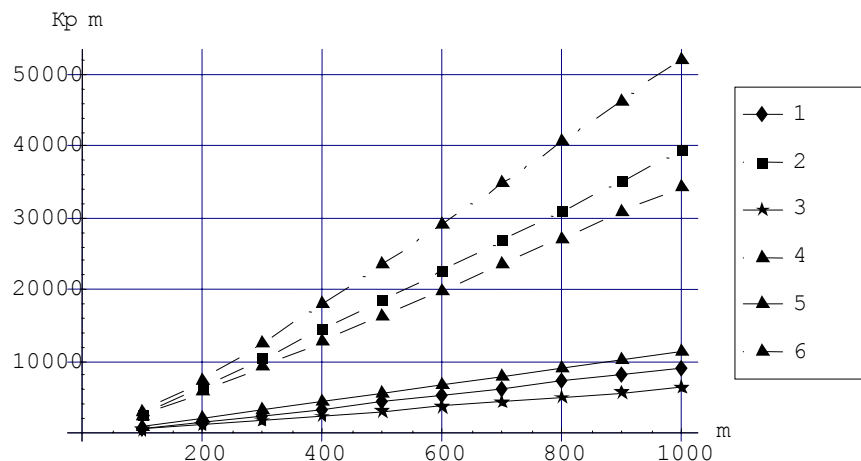


Рисунок 4 – Зависимость коэффициентов ускорения от порядка системы с учетом операций обмена

Определение характеристик параллелизма осуществлялось с помощью пакета *Mathematica*® (Wolfram Research Inc.), численный эксперимент проводился на базе тестов для СОДУ, разработанных в НИВЦ МГУ [8].

Анализ полученных результатов позволяет сделать следующие выводы:

- самым эффективным из рассмотренных последовательных методов является вложенный метод с контролем погрешности на шаге по правилу Рунге;

- лучшими параллельными методами с точки зрения трудоемкости являются вложенный экспоненциальный метод и экспоненциальный с локальной экстраполяцией;

- преимущества методов на базе локальной экстраполяции проявляются при получении высокоточных решений ($10^{-15} - 10^{-20}$) и для сложных правых частей.

Заключение

Анализ вычислительной сложности различных технологий определения апостериорной локальной погрешности при решении задачи Коши имел своей целью управление шагом интегрирования в достижение некоторой заданной точности с минимальными вычислительными затратами. Перспективным направлением дальнейших исследований является оценка влияния различных топологий многопроцессорных ВС на характеристики качества параллельных алгоритмов, исследование устойчивости полученных методов.

Литература

1. Rajkumar Buyya. High Performance Cluster Computing. Volume 1: Architectures and Systems. Volume 2: Programming and Applications. Prentice Hall PTR, Prentice-Hall Inc., 1999.
2. В.В.Воеводин, Вл.В.Воеводин. Параллельные вычисления. –СПб.: БХВ-Петербург, 2002. –608с.
3. Хайрер Э., Нёрсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи: Пер. с англ. – М.: Мир, 1990. – 512с.
4. Bergman S., Rauber T., Runger G. Parallel execution of embedded Runge-Kutta methods. International Journal of Supercomputer Applications, 10(1):62-90, 1996.
5. Houwen P.J., Sommeijer B.P. Parallel ODE solver.// Proceedings of the International Conference on Supercomputing. –ACM Press, 1990, p.71-81.
6. Арушанян О.Б., Залеткин С.Ф. Численное решение обыкновенных дифференциальных уравнений на Фортране.– М.: МГУ, 1990.–336с.
7. Бройнль Т. Параллельне програмування: Початковий курс: Навч. посібник/ Вступ.слово А.Ройтера;Пер. з нім. В.А.Святного. – К.: Вища шк., 1997.-358с.
8. Арушунян О.Б., Залеткин С.Ф., Калиткин Н.Н. Тесты для вычислительного практикума по обыкновенным дифференциальным уравнениям.// Вычислительные методы и программирование, 2002, т.3, с.11-19.

Поступила в редакцию 11 января 2004 года