

УДК 519.3

Параллельные алгоритмы экстраполяционных методов решения задачи Коши для компьютеров с распределенной памятью

Л.П. Фельдман, И.А. Назарова

Донецкий национальный технический университет
feldman@r5.dgtu.donetsk.ua, nazar451@gmail.com

Abstract

Feldman L.P., Nazarova I.A. Parallel Algorithms of Extrapolation Methods for Solving Cauchy Problem on Distributed Memory Computers. Extrapolation methods for solving Cauchy problem for ordinary differential equations possess a high degree of potential parallelism. This article is dedicated to the design and analysis of the efficiency of parallel algorithms for local extrapolation based on explicit based methods. The developed algorithms are implemented on parallel systems with distributed memory and the topology of the hypercube. The estimations of the runtime and exchanges, total overhead of parallelism, speedup and efficiency of parallel solutions are defined.

Введение

Моделирование реальных многомерных динамических процессов, описываемых системами обыкновенных дифференциальных уравнений (СОДУ), представляет собой класс задач, при реализации которых использование параллельных или распределенных компьютеров не только оправдано, но и необходимо.

Данная статья посвящена разработке параллельных алгоритмов решения задачи Коши с встроенным способом оценки апостериорной погрешности на основе технологии локальной экстраполяции Ричардсона, а также исследованию эффективности их отображений на параллельные системы с распределенной памятью.

Экстраполяционные методы Ричардсона предназначены для получения высокоточного решения задачи Коши и интегрирования СОДУ со сложными правыми частями. Практическое применение методов затруднено в связи с большой вычислительной сложностью их последовательных реализаций. Поэтому построение эффективных параллельных методов локальной экстраполяции Ричардсона – один из наиболее реальных способов сокращения времени интегрирования многомерных начальных задач для систем обыкновенных дифференциальных уравнений.

1. Технология метода локальной экстраполяции Ричардсона

Метод локальной экстраполяции Ричардсона (ЛЭР) является обобщением технологии удвоения шага по правилу Рунге [1-2]. Идея метода заключается в многократном измельчении шага интегрирования, и также в

многократном применении процесса вычисления, названного локальной экстраполяцией.

Решение задачи Коши для систем обыкновенных дифференциальных уравнений первого порядка:

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

рассматривается при переходе из точки x_n в точку $x_{n+1} = x_n + H$, где H – базовая длина шага (рис. 1). 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: $h_1 > h_2 > \dots > h_{k-1} > h_k > \dots$, 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} :

$$T_{i,1} = \bar{y}_{h_i}(x_n + H) = \bar{y}_{n+1}^{(i)}, i = \overline{1, k}.$$

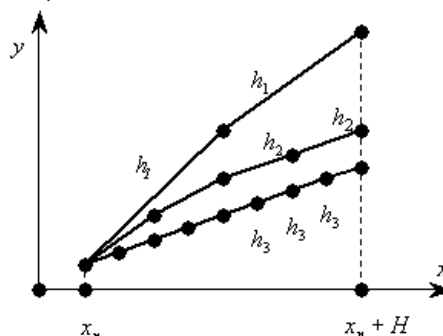


Рисунок 1 – Схема технологии локальной экстраполяции Ричардсона

Выполнив вычисления для ряда последовательных значений i , по рекуррентному

соотношению, определяют экстраполированные значения для произвольных i, j по схеме локальной полиномиальной экстраполяции Эйткена-Невилла:

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

Величина b равна 1 в общем случае, для симметричных опорных методов b равно двум. T_{ij} – приближенное решение задачи Коши, полученное численным методом порядка $r_0 + (j-1) \cdot b$ с шагом h_i .

Достоинство этого метода состоит в том, что он дает целую таблицу результатов вычислений (табл. 1), которые образуют последовательность вложенных методов, позволяют оценить локальную погрешность и выбрать стратегию для управления шагом интегрирования.

Таблица 1. Экстраполяционная таблица

r_0	$r_0 + b$...	$r_0 + (k-1)b$
T_{11}		...	
T_{21}	T_{22}	...	
...	
T_{k1}	T_{k2}	...	T_{kk}

2. Эффективный последовательный алгоритм технологии локальной экстраполяции для решения СОДУ

Построим эффективный с точки зрения минимизации вычислительных затрат последовательный метод с использованием технологии Ричардсона. Так как экстраполяционная технология включает: опорный численный метод решения задачи Коши, последовательность сеток интегрирования, рекуррентное правило вычисления значений приближенного решения, то эффективность ее применения напрямую зависит от правильного выбора и сочетания всех трех составляющих.

Для получения множества шагов интегрирования применяются числовые последовательности, образованные гармоническим рядом, рядом четных чисел, степенями двойки [1-3]:

- 1) $P_1 = \{1, 2, 3, 4, 5, 6, \dots\}$;
- 2) $P_2 = \{1, 2, 4, 6, 8, 10, 12, \dots\}$;
- 3) $P_3 = \{1, 2, 4, 8, 16, 32, 64, \dots\}$.

Вычислительная сложность использования последовательностей натуральных чисел для произвольных опорных методов определяется числом обращений к функции \bar{f} – правой части системы исходных ОДУ:

$$N(k) = n_1 + n_2 + \dots + n_k,$$

$$P_1: N(k) = (k^2 + k) / 2,$$

$$P_2: N(k) = k^2 - k + 1,$$

$$P_3: N(k) = 2^k - 1,$$

где k – число строк экстраполяционной таблицы.

При одном и том же опорном методе наиболее затратной является последовательность P_3 , которая фактически повторяет процесс удвоения шага; последовательности P_1 и P_2 имеют приблизительно одинаковую вычислительную сложность.

Выбор опорного метода интегрирования базируется на исследовании влияния порядка метода и его свойств на объем необходимых вычислений. Пусть опорный метод имеет порядок – r_0 , длина экстраполяционной таблицы равна k , а число стадий явного метода – s , определим необходимый размер экстраполяционной таблицы для получения аппроксимации порядка r .

При условии симметричности опорного метода, он имеет асимптотическое разложение по степеням h^2 , и каждое однократное применение процедуры локальной экстраполяции исключает две степени h [1]. Для произвольного опорного метода каждая экстраполяция исключает одну степень h . Тогда, количество строк экстраполяционной таблицы, определяется одним из следующих соотношений:

$$1) r = r_0 + (k-1) - \text{ для произвольного}$$

опорного метода;

$$2) r = r_0 + 2(k-1) - \text{ для симметричного}$$

опорного метода.

Следовательно, если вычислены k строк экстраполяционной таблицы, то для симметричных опорных методов имеем T_{kk} в качестве аппроксимации наивысшего порядка точности равного $2k$ и, кроме того, точность аппроксимации $T_{k-1,k}$ имеет порядок – $(2k-2)$. Соответственно, для произвольных опорных методов имеем: $T_{kk} - (k-1)$ и $T_{k-1,k} - (k-2)$ порядка точности. Для определения апостериорной локальной погрешности и управления шагом интегрирования используется величина: $\|T_{k-1,k} - T_{k,k}\|$.

Определим вычислительные затраты на формирование последовательности сеток интегрирования на основе гармонического ряда (P_1) и лучшей из четных последовательностей (P_2). Формирование экстраполяционной таблицы для симметричных методов на основе четных рядов требует $k = \left\lfloor (r - r_0) / 2 + 1 \right\rfloor$ строк для достижения точности порядка r , а для произвольных методов – $k = r - r_0 + 1$. Таким

образом, h^2 -экстраполяция на основе симметричных опорных методов и второй четной последовательности для генерации сетки интегрирования является наиболее эффективной с точки зрения вычислительных затрат [4].

Например, если в качестве опорного метода выбирается явный одношаговый симметричный метод второго порядка точности, $r_0 = 2$, то оптимальной последовательностью является вторая четная последовательность с числом вычислений f , равным:

$$N(r/2) = (r^2 - 2 \cdot r + 4) / 4.$$

Для явных опорных методов, не обладающих свойством симметричности, оптимальной является гармонический ряд с числом обращений к правой части СОДУ:

$$N(r-1) = (r^2 - r) / 2.$$

Для исследования влияния порядка опорного метода на накладные расходы в расчете на один шаг интегрирования определяется время выполнения одного шага последовательного алгоритма по технологии локальной экстраполяции. На рис. 2 приведены графики зависимости времени выполнения одного шага интегрирования от порядка получаемого экстраполяционного метода при постоянной размерности задачи для явных опорных методов порядков $r_0 = 2 \div 10$, причем время изображено в логарифмическом масштабе.

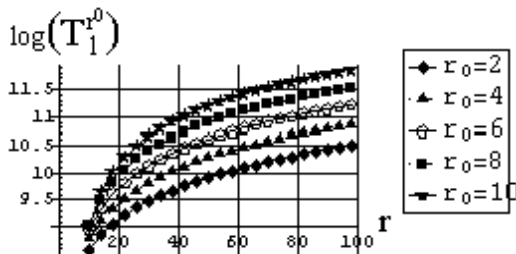


Рисунок 2 – Время выполнения одного шага интегрирования по технологии локальной экстраполяции с явными опорными методами

Пусть $T_1^{r_0}$ - время выполнения последовательного алгоритма ЛЭР на основе

явного опорного метода порядка r_0 , T_F - время обращения к правой части СОДУ, t_{op} - время реализации одной операции с плавающей точкой.

Анализ вычислительной сложности полученного алгоритма позволяет сделать следующие выводы:

1) вычислительная сложность одного шага интегрирования по технологии локальной экстраполяции при одних и тех же способах получения сеток интегрирования и порядке экстраполяционного метода r , существенно зависят от порядка опорного метода и его свойств, от времени вычисления правой части СОДУ и времени на одну арифметическую операцию;

2) длина экстраполяционной таблицы для получения метода порядка r линейно уменьшается с ростом порядка опорного метода: $r_0 \uparrow \Rightarrow k \downarrow$;

3) не смотря на уменьшение количества строк экстраполяционной таблицы, вычислительная сложность метода локальной экстраполяции Ричардсона возрастает с ростом порядка опорного метода: $r_0 \uparrow \Rightarrow T_I^{r_0} \uparrow$, причем

при сложных правых частях СОДУ: $T_F \gg t_{op}$ в $1,55 \div 2$ раза для методов смежных порядков, а при тривиальных правых частях: $T_F \approx t_{op}$ накладные расходы возрастают с ростом порядка r_0 существенно, в десятки/сотни раз;

4) накладные расходы на экстраполяцию возрастают с ростом размерности начальной задачи: $m \uparrow \Rightarrow T_I^{r_0} \uparrow$.

Суммируя все полученные результаты, можно сделать вывод, что для уменьшения накладных вычислительных расходов при применении технологии локальной экстраполяции, следует выбирать опорный метод малого порядка с достаточными свойствами устойчивости. Особенностью интегрирования задачи Коши для СОДУ по технологии ЛЭР является тот факт, что экстраполяционная таблица вычисляется для каждой размерности исходной системы. Теоретически нет никаких ограничений на длину экстраполяционной таблицы, но конечность памяти машины и накапливание ошибок округления, ограничивают длину экстраполяционной таблицы сверху и, как правило, используется: $k_{max} \leq 10$. Поэтому для получения высокоточных приложений ($10^{-15} - 10^{-20}$), где вычисления правой части СОДУ достаточно сложны, не имеется другой возможности, как использовать технологию локальной экстраполяции Ричардсона на базе опорного метода высокого порядка точности.

3. Параллельные алгоритмы технологии локальной экстраполяции для решения СОДУ

Потенциально вычисления по технологии локальной экстраполяции содержат три источника внутреннего параллелизма:

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

Построение параллельных алгоритмов решения нелинейной задачи Коши для СОДУ по технологии Ричардсона базируется на иерархической декомпозиционной методике. После того, как определены все составляющие технологии экстраполяции, процесс решения можно разбить на две последовательно выполняемые подзадачи:

1) вычисление k аппроксимаций решения в точке x_{n+1} с разными шагами интегрирования:

$$\bar{y}_{h_i}(x_n + H) = \bar{y}_{n+1}^{(i)};$$

2) построение экстраполяционной таблицы.

Организация параллельных вычислений может производиться двумя способами, отсюда и две различные макрооперационные схемы алгоритма. Первый вариант подразумевает использование только системного параллелизма, второй – комбинацию параллелизма экстраполяции и системного. Рассмотрим первую подзадачу алгоритма. В качестве основной макрооперации для обоих вариантов вводится – вычисление аппроксимации точного решения в точке сетки с заданным шагом интегрирования на базе явного опорного метода. Граф влияния для первого варианта макрооперационной вычислительной схемы приведен на рисунке 3.

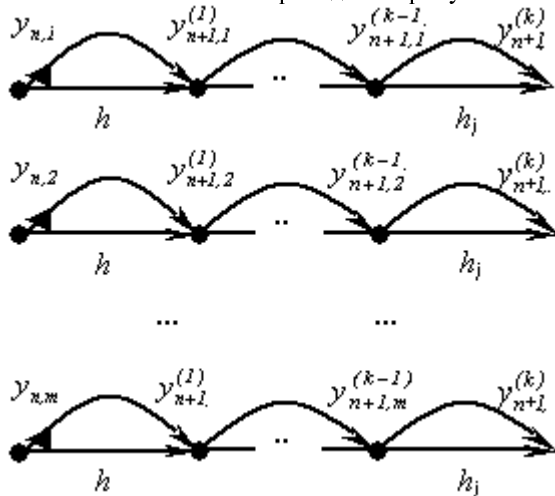


Рисунок 3 – Граф влияния первой макрооперационной схемы ЛЭР (системный параллелизм)

При таком варианте все аппроксимации решения вычисляются последовательно одновременно для каждой размерности системы.

По второй макрооперационной схеме параллельно вычисляется k независимых аппроксимаций решения в точке $x_n + H$, граф влияния приведен на рис. 4. Второй вариант алгоритма имеет большую степень параллелизма, равную $Dop = m \cdot k$, в отличие от первого для которого: $Dop = m$. Однако, этот недостаток схемы №1, а именно, последовательное выполнение макроопераций, может быть компенсирован лучшей балансировкой загрузки многопроцессорной ВС, а также низкой интенсивностью обменов.

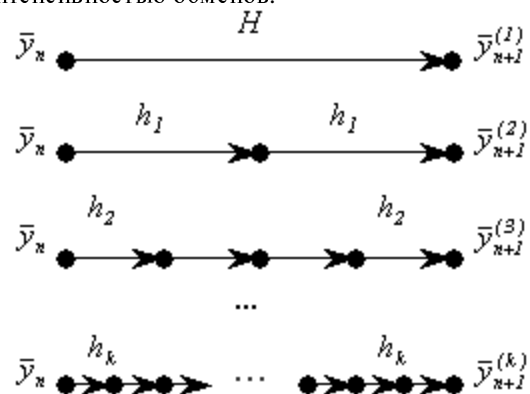


Рисунок 4 – Граф влияния второй макрооперационной схемы ЛЭР (параллелизм экстраполяции)

Вторая подзадача, а именно вычисление экстраполяционной таблицы, в зависимости от способа решения первой подзадачи, также может быть распараллелена разными способами. В качестве макрооперации для этой подзадачи вводится вычисление экстраполированного значения $T_{i,l,j}, j=1, m, i=2, k, l=i, k$, вычисленного по формуле Эйткена-Невилла (рис. 5).

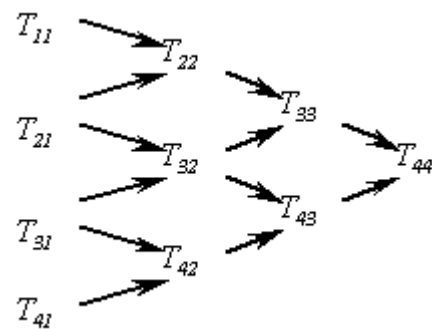


Рисунок 5 – Вычисления по схеме полиномиальной экстраполяции Эйткена-Невилла, $r = 4$

Распараллеливание построения таблицы экстраполяции может быть осуществлено за счет независимого выполнения вычислений по формуле полиномиальной экстраполяции для каждой размерности системы ($Dop = m$), а также за счет распределенного вычисления каждой строки таблицы ($Dop = k - 1$). Как правило, $m > k$, поэтому появляется возможность совместить оба вида параллелизма. На рассмотренном этапе детализации дан предварительный анализ эффективности каждой из вычислительных схем, перейдем к использованию внутреннего параллелизма метода.

Первая макрооперационная схема алгоритма локальной экстраполяции практически повторяет макрооперационную схему для правила Рунге, с тем различием, что теперь у нас не три, а k последовательно вычисляемых аппроксимаций по ЯМРК. Заметим, что, каждую макрооперацию схемы локальной экстраполяции можно представить, как n_i раз используемую макрооперацию однократного применения явной численной схемы одношагового метода.

Далее по декомпозиционной технологии необходимо распределить данные по процессорам. Если размерность процессорного поля велика и больше, чем длина экстраполяционной таблицы, появляется возможность выполнять макрооперации первой подзадачи параллельно. Для этого необходимо разбить процессоры на группы таким образом, чтобы обеспечить наиболее рациональную балансировку загрузки многопроцессорной системы.

Рассмотрим следующие способы разбиения процессоров на группы, а соответственно и распределения данных по процессорам [5-6]:

- 1) равномерный,
- 2) пропорциональный,
- 3) комбинационный.

При наиболее простом, равномерном способе, каждый процессор или процессорная группа будут отвечать за вычисление равного количества аппроксимаций. Если размерность процессорного поля равна p , количество строк в экстраполяционной таблице – k , то при таком способе будет k групп по $p_i = \lceil p/k \rceil$, $i = \overline{1, k}$ процессоров в каждой и каждый процессор в i -той группе будет содержать $m_i = \lceil m/p_i \rceil$ компонент соответствующей аппроксимации решения и соответствующего экстраполированного значения. Для равномерного разбиения: $p_1 = p_2 = \dots = p_k$ и, соответственно, $m_1 = m_2 = \dots = m_k$. Каждая группа будет

отвечать за вычисление определенной аппроксимации решения: $T_{j1}, j = \overline{1, k}$, время решения определяется по $\max_{j=1}^k T_{j1}$, и все группы, кроме последней, будут простаивать некоторое время.

Рассмотрим различные варианты организации параллельных вычислений экстраполяционной таблицы по схеме Эйткена-Невилла при равномерном распределении для вычислительной схемы №2. После параллельного вычисления k аппроксимаций решения на k группах процессоров, имеется следующее распределение исходных данных для вычисления экстраполированных значений. Первая группа процессоров содержит вектор $T_{11,j}; j = \overline{1, m}$, вторая – $T_{21,j}; j = \overline{1, m}$ и, соответственно, k -тая – $T_{k1,j}; j = \overline{1, m}$, 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 $n_i, i = \overline{1, k}$: $p_1 : p_2 : \dots : p_k = n_1 : n_2 : \dots : n_k$,

$$\sum_{i=1}^k p_i = p, \quad p_i = \frac{p}{N(k)} \cdot n_i.$$

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 групп, начиная с последней. Таким образом, каждая группа будет содержать тем больше 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 МПВС, может быть сформулирована в терминах комбинаторной оптимизации. Точное решение задач этого класса за приемлемое время невозможно, так как задача является NP-сложной. Поэтому будем искать близкое к оптимальному рациональное решение 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 $(k-i+1)$ -тую аппроксимации решения. Для получения i -той аппроксимации необходимо n_i раз выполнить обращение к опорному методу решения, для $(k-i+1)$ -той – 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, $kI = \lceil k/2 \rceil$ группа – $n_{kI} + n_{kI+1}$. Напомним, что для $P_2 : \{1, 2, 4, 6, 8, 10, \dots\}$.

В таблице 2 приведены данные разбиения процессорного поля на группы в случае, если длина экстраполяционной таблицы – четное число и используется вторая четная последовательность, формирующая сетки интегрирования.

Таблица 2. Разбиение процессоров на группы для P_2 и четной длины экстраполяционной таблицы

$k \backslash N_2$	1	2	3
6	11	10	10
8	15	14	14
10	19	18	18
12	23	22	22

Столбцы таблицы 2 соответствуют номеру группы процессоров, строки – длине таблицы экстраполяции. Очевидно, что имеется практически равномерное обеспечение загрузки процессорных групп, кроме первой, которая отличается от остальных вне зависимости от длины таблицы на единицу. В случае нечетной длины таблицы разбиение становится менее равномерным за счет непарной последней группы. В этом случае последняя группа может быть расформирована за счет равномерного распределения по остальным группам.

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 (рис. 6).

Проведем сравнение вычислительных схем на основе реальных коэффициентов ускорения и эффективности для топологии гиперкуб, как наиболее эффективной среди рассмотренных. Равномерное разбиение процессоров на группы, несмотря на простоту реализации, обладает большими накладными расходами, в том числе и

непроизводительными, и имеет наиболее худшие характеристики параллелизма.



Рисунок 6 – Эффективность обменных операций метода ЛЭР для различных топологий

Первая схема, использующая только системный параллелизм, имеет определенные преимущества, она может быть реализована на параллельном компьютере любого типа и обладает хорошей балансировкой загрузки процессоров. Первая и третья схемы при сложных правых частях имеют хорошие показатели эффективности для систем, как с высокоскоростными каналами передачи информации, так и низкоскоростными (рис. 7).

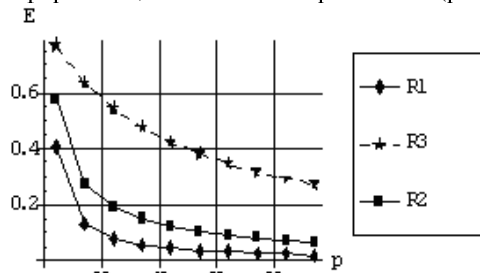


Рисунок 7 – Коэффициенты реальной эффективности методов от числа процессоров,

$$r = 8, m = 1000; T_f \sim t_{op}$$

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.

Заключение

Численный эксперимент на базе тестов для СОДУ [7] 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 основным для эффективного использования их в высокопроизводительных мультипроцессорных системах. Апробация параллельных алгоритмов проводилась на моделях ВС SIMD, MIMD, 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.

Литература

1. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи / Э.Хайрер, С.Нёрсет, Г.Ваннер. - М.: - Мир, 1990. -512с.
2. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Жесткие задачи / Э. Хайрер, Г.Ваннер. - М.: Мир, 1999. - 685с.
3. Дж. Холл Современные численные методы решения обыкновенных дифференциальных уравнений / Дж. Холл, Дж. Уатт. - М.: Мир, 1979. – 312с.
- 4.Фельдман Л.П. Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений / Л.П.Фельдман, И.А.Назарова // Математическое моделирование. – 2006. – Т.18. – № 9 – С.17-31.
5. Фельдман Л.П. Эффективность параллельных алгоритмов оценки локальной апостериорной погрешности для численного решения задачи Коши / Л.П.Фельдман, И.А.Назарова // Электронное моделирование. – 2007. – Т. 29. – № 3. - С.11-25.
6. Назарова И.А. Эффективность применения технологии локальной экстраполяции в параллельных алгоритмах численного решения задачи Коши /И.А. Назарова // Научно-теоретический журнал ИПИИ НАН Украины. Искусственный интеллект. – 2006. – №3. – С. 192–202.
7. Арушунян О.Б. Тесты для вычислительного практикума по обыкновенным дифференциальным уравнениям / О.Б. Арушунян, С.Ф.Залеткин, Н.Н. Калиткин // Вычислительные методы и программирование. – 2002. – Т.3. – С. 11-19.

Поступила в редакцию 30.03.2010