

---

УДК 681.3

**Л. П. Фельдман**, д-р техн. наук, **И. А. Назарова**  
Донецкий национальный технический университет  
(Украина, Донецк, ул. Артема, 66,  
тел.: (062) 3010856,  
E-mail: feldman@r5.dgtu.donetsk.ua; nazarova@r5.dgtu.donetsk.ua)

## **Эффективность параллельных алгоритмов оценки локальной апостериорной погрешности для численного решения задачи Коши**

Предложены параллельные численные алгоритмы оценки апостериорной локальной погрешности на основе явных и полностью неявных разностных схем, исследованы особенности применения алгоритмов для решения линейной задачи Коши. Разработаны вычислительные схемы отображения методов на параллельные структуры различной топологии: линейка — кольцо, решетка — тор, гиперкуб. Приведены сравнительные характеристики потенциального и реального параллелизма, а также результаты численных экспериментов на системе тестов.

Запропоновано паралельні чисельні алгоритми оцінки апостеріорної локальної похибки на базі явних та повністю неявних різницевих схем, досліджено особливості застосування алгоритмів для розв'язування лінійної задачі Коші. Розроблено обчислювальні схеми відображення методів на паралельні структури різної топології: лінійка — кільце, сітка — тор, гіперкуб. Наведено порівняльні характеристики потенційного та реального паралелізму, а також результати числових експериментів на системі тестів.

*К л ю ч е в ы е с л о в а:* апостериорная шаговая погрешность, параллельный алгоритм, вложенный метод, локальная экстраполяция.

Одной из главных проблем, возникающих при численном решении задачи Коши для систем обыкновенных дифференциальных уравнений (СОДУ), является оценка погрешности приближенного решения. Априорная оценка глобальной погрешности разностного метода позволяет судить о сходимости приближенного решения задачи к точному и, следовательно, о его применимости. Апостериорная оценка локальной погрешности, получаемая на каждом шаге вычислений, позволяет автоматически выбирать шаг интегрирования, обеспечивающий заданную точность приближенного решения. Одношаговые методы, построенные способом Рунге—Кутты, как и многошаговые, в отличие от вычислительных правил, полученных на основе разложения решения в ряд по последовательным главным частям, не позволяют судить о локальной точности найденного значения прибли-

женного решения непосредственно по результатам промежуточных вычислений. Традиционное оценивание погрешности с использованием разложения в ряд Тейлора не представляет практического интереса, поскольку требует вычисления частных производных высоких порядков.

Известными и общепринятыми методами оценки локальной апостериорной погрешности решения СОДУ являются следующие:

- дублирование шага по правилу Рунге;
- технология локальной экстраполяции Ричардсона;
- вложенные методы или методы вложенных форм.

Оценки локальной погрешности необходимы, с одной стороны, чтобы обеспечить длину шага интегрирования, достаточно малую для достижения требуемой точности вычисляемых результатов, а с другой стороны, чтобы гарантировать достаточно большую длину шага во избежание бесполезной вычислительной работы. Поэтому получение надежных и, в то же время простых и эффективных способов оценки шаговой погрешности — одна из основных проблем, возникающих при разработке современного качественного численного алгоритма решения задачи Коши.

Предлагаемые алгоритмы решения линейных задач ориентированы на использование в многопроцессорных вычислительных системах SIMD, MIMD и кластерной архитектуры с различными топологиями соединения процессорных элементов. Численно решается задача Коши для СОДУ (в общем случае нелинейных уравнений) первого порядка размерности  $m$  с известными начальными условиями:

$$\mathbf{y}' = F(x, \mathbf{y}), \mathbf{y}(x_0) = \mathbf{y}_0, \quad (1)$$

где  $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$ ,  $\mathbf{y}_0 = (y_{10}, y_{20}, \dots, y_{m0})^T$ ,  $F = (f_1, f_2, \dots, f_m)^T$ .

Современные теоретические исследования и вычислительные эксперименты в области параллельных вычислений свидетельствуют о том, что наиболее часто используемым способом создания параллельных методов является распараллеливание хорошо известных и многократно апробированных последовательных численных алгоритмов [1]. Развитым и общепризнанным математическим аппаратом разработки параллельных алгоритмов являются графовые модели. Однако для решения реальных задач применение графовой модели сопряжено с большими трудностями. Поэтому для разработки параллельных алгоритмов решения задач практического уровня сложности используется многоэтапный технологический процесс — иерархическая декомпозиционная методика. Графы влияния могут быть использованы для построения параллельного алгоритма, как на верхнем уровне для анализа взаимодействия подзадач, так и на нижнем — для распараллеливания отдельных макроопераций, соответствующих

определенным подзадачам. Выбор декомпозиционной методики распараллеливания обусловлен преимуществами этого подхода, а именно: взаимозависимостью и итеративностью этапов построения параллельного алгоритма и возможностью обеспечить необходимый уровень масштабируемости вычислений за счет варьирования детальности декомпозиции.

При параллельных вычислениях важно учитывать сложности межпроцессорных связей. Поскольку для параллельных методов решения задачи Коши информационное взаимодействие по типу коммуникаций является структурным, существует возможность разработать единые коммуникационные примитивы различных операций передачи данных в различных топологиях. Наиболее часто используются три топологии, линейка—кольцо, решетка—тор, гиперкуб, и две коммуникационные операции: передача данных между двумя процессорами сети (операция типа «точка—точка») и передача данных от всех процессоров сети всем процессорам сети (множественная пересылка «все—всем»).

Оценка времени выполнения операции передачи одного сообщения объемом  $V$  байт между двумя процессами, локализованными на различных процессорах при распределенной памяти, проводилась на основании следующей модели:

$$T_{T-T} = t_s + t_w V / l; \quad t_w = y / B, \quad (2)$$

где  $t_s$  — длительность подготовки сообщения для передачи;  $l$  — длина маршрута;  $t_w$  — длительность передачи одного байта;  $y$  — число байт в слове;  $B$  — пропускная способность канала передачи данных (байт/с). Трудоемкость одиночной операции пересылки данных между двумя процессорами может быть определена подстановкой длины диаметра сети в выражение (2). Для вычисления времени выполнения множественной пересылки в условиях той же модели необходимо определиться с выбором алгоритма маршрутизации.

К числу наиболее распространенных оптимальных алгоритмов передачи данных относится класс методов покоординатной маршрутизации. Идея этих методов заключается в том, что поиск путей передачи данных осуществляется последовательно для каждой размерности рассматриваемой топологии. Сокращение времени передачи данных может быть достигнуто и при использовании встречных обменов, когда данные передаются одновременно в обе стороны. Это позволяет добиться почти двойного уменьшения времени обмена данными по сравнению с однонаправленными обменами. Таким образом, в условиях существования двунаправленных линков для процессорного поля размерности  $p$  получаем

при кольцевой топологии —

$$T_{T-T}^R = t_s + Vt_w \lfloor p/2 \rfloor;$$

$$T_{B-B}^R = (t_s + Vt_w)(p-1),$$

при топологии решетка —

$$T_{T-T}^M = t_s + 2Vt_w \lfloor \sqrt{p}/2 \rfloor;$$

$$T_{B-B}^M = 2t_s(\sqrt{p}-1) + Vt_w(p-1),$$

при топологии гиперкуб —

$$T_{T-T}^H = t_s + Vt_w \log_2 p;$$

$$T_{B-B}^H = t_s \log_2 p + t_w V(p-1).$$

**Разработка и исследование эффективности параллельных методов оценки шаговой погрешности на основе явных одношаговых схем.** Явные формулы Рунге—Кутты (ЯМРК) входят в число старейших и хорошо изученных схем численного анализа. Однако, несмотря на наличие огромных и всесторонних знаний в этой области, ЯМРК продолжают быть объектом активных исследований, особенно в таком новом научном направлении как параллельные вычисления. Потенциально ЯМРК содержат два вида параллелизма: параллелизм метода и параллелизм системы.

Системный параллелизм реализуется через распределенные вычисления различных компонент системы, т.е. отдельных компонент шаговых коэффициентов и векторов решений. Параллелизм метода связан с выделением независимых субвычислений внутри одного шага метода и, как правило, значительно меньше системного. Для СОДУ размерности  $m$  явный  $s$ -стадийный метод Рунге—Кутты имеет следующий вид:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{k}_i;$$

$$\mathbf{k}_i = F(x_n + c_i h; \mathbf{g}_i), \quad i = \overline{1, s};$$

$$\mathbf{g}_i = \mathbf{y}_n + h \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j.$$

Вычисление любой аппроксимации решения состоит в вычислении множества коэффициентов  $\mathbf{k}_i = (k_{i1}, k_{i2}, \dots, k_{im})$ ,  $i = \overline{1, s}$ , и собственно приближенного решения  $\mathbf{y}_{n+1}(h)$ . Для явных схем вычисление шаговых коэффициентов есть сугубо последовательный процесс. Равенство нулю элементов  $a_{ij}$ ,  $i = 2, s$ ;  $j = 1, s-1$ , в матрице Батчера для ЯМРК определяет

возможность одновременного вычисления некоторых коэффициентов, однако обеспечивает очень незначительную степень параллелизма. Для разработки параллельного алгоритма был использован математический аппарат графов влияния [1].

Задача распараллеливания в такой постановке сводится к отысканию максимального независимого множества вершин орграфа, причем вершинам графа сопоставляются выполняемые операции, и вершины соединяются дугами тогда и только тогда, когда результат выполнения одной операции влияет на результат выполнения смежной.

В общем случае для каждого коэффициента  $\mathbf{k}_i$ ,  $i = \overline{1, s}$ ,  $l$  компонент вектора  $\mathbf{g}_i$ ,  $i = \overline{1, s}$ , могут быть вычислены параллельно:

- 1)  $l = m$  компонент при  $m = p$ ;
- 2)  $l = \lceil m/p \rceil$  компонент при  $m > p$ .

Параллельные алгоритмы решения задачи Коши, разработанные на основе явных численных схем, содержат встроенные альтернативные способы определения локальной апостериорной погрешности, а именно: дублирование шага по правилу Рунге; вложенные методы Рунге—Кутты (ВМРК); технология локальной экстраполяции. Принцип удвоения шага — наиболее простой способ получения шаговой погрешности, однако он имеет большие накладные расходы: объем вычислений на один узел сетки возрастает почти втрое.

Метод локальной экстраполяции является обобщением технологии удвоения шага по правилу Рунге. Идея метода заключается в многократном измельчении шага интегрирования, и также в многократном применении процесса локальной экстраполяции. Решение задачи Коши выполняется при переходе из точки  $x_n$  в точку  $x_{n+1} = x_n + H$ , где  $H$  — базовая длина шага,  $H > 0$ . Выбирают ряд натуральных чисел, такой, что:  $n_1 < n_2 < \dots < n_{k-1} < n_k < \dots$ , и соответственно последовательность  $h_1 > \dots > h_{k-1} > h_k > \dots$ , где  $h_i = H / n_i$ . Задают опорный численный метод порядка  $r^0$  и, выполняя  $n_i$  шагов интегрирования длиной  $h_i$ , находят приближенное решение исходной задачи:  $T_{i,1} := y_{h_i}(x_n + H)$ . Выполнив вычисления для ряда последовательных значений  $i$ , по рекуррентному соотношению определяют экстраполированные значения  $T_{i,j+1}$  для произвольных  $i, j$  [2]. Этот процесс получил название локальной полиномиальной экстраполяции:

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

Здесь  $b = 1$  в общем случае, а для симметричных опорных методов  $b = 2$ .

Достоинство этого метода состоит в том, что с его помощью можно получить полную таблицу результатов вычислений, которые образуют

последовательность вложенных методов и позволяют оценить локальную погрешность, выбрать стратегию для методов переменного порядка. Преимущество экстраполяционных методов заключается в том, что у них на каждом шаге можно менять не только длину шага, но и порядок метода.

Для получения множества шагов интегрирования используем числовые последовательности, образованные гармоническим рядом, степенными двойками, различными четными рядами чисел:

$$P_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, \dots\} - n_k = k;$$

$$P_2 = \{1, 2, 4, 6, 8, 10, 12, \dots\} - n_k = 2k;$$

$$P_3 = \{1, 2, 4, 8, 16, 32, \dots\} - n_k = 2^k.$$

Вычислительная сложность формирования сеток интегрирования для произвольных опорных методов определяется числом обращений к функции  $\mathbf{F}$  — правой части СОДУ:

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

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

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

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

где  $k$  — число строк экстраполяционной таблицы. Необходимый размер экстраполяционной таблицы для получения решения методом порядка  $r$ , если исходный опорный метод имеет порядок  $r^0$ , зависит от свойств опорного метода. При условии симметричности опорный метод имеет асимптотическое разложение по степеням  $h^2$  и каждая экстраполяция исключает две степени  $h$  [2]. Для произвольного опорного метода каждая экстраполяция исключает одну степень  $h$ . Тогда число строк экстраполяционной таблицы, определяется следующими соотношениями:

$$r = r^0 + k - 1 \text{ для произвольного опорного метода;}$$

$$r = r^0 + 2(k - 1) \text{ для симметричного опорного метода.}$$

Следовательно, если вычислены  $k$  строк экстраполяционной таблицы, то в случае симметричности опорного метода получаем аппроксимацию решения  $T_{kk}$ , имеющую наивысший порядок точности, равный  $2k$ . При этом величина шага интегрирования выбирается с учетом величины  $\|T_{k-1,k} - T_{k,k}\|$ . Накладные расходы на формирование последовательности сеток интегрирования определяем исходя из того, что при формировании экстраполяционной таблицы для достижения точности порядка  $r$  симметричные методы требуют  $r/2$  строк, а произвольные методы —  $(r - 1)$  строк. Например, если в качестве опорного метода выбран симметричный явный

метод Рунге—Кутты 2-го порядка, то оптимальной последовательностью является вторая четная последовательность. Число обращений к правой части СОДУ при этом составляет  $N(r/2) = (r^2 + 2r + 4)/4$ . Для произвольных опорных методов наименее затратным является гармонический ряд.

Таким образом, экстраполяционная технология имеет три составляющие: опорный численный метод решения задачи Коши, последовательность сеток, рекуррентное правило вычисления значений приближенного решения. Ее эффективность зависит от правильного выбора и сочетания этих составляющих. Следует заметить, что при разработке параллельных вычислительных схем для технологии локальной экстраполяции и дублирования шага появляется необходимость исследовать:

а) способы разбиения процессоров на группы, вычисляющие определенную аппроксимацию решения: равномерный, пропорциональный и комбинационный;

б) степень влияния порядка опорного метода на характеристики алгоритма в целом.

Результаты теоретического анализа и численного эксперимента позволяют сделать следующие выводы:

1) накладные расходы на один шаг интегрирования по технологии локальной экстраполяции при одних и тех же способах получения сетки интегрирования  $P_i$  и при одинаковом порядке полученного экстраполяционного метода  $r$  существенно зависят от порядка опорного метода и его свойств, от времени вычисления правой части  $T_F$  СОДУ, времени на одну арифметическую операцию  $t_{оп}$  и коммуникационных констант;

2) длина экстраполяционной таблицы линейно уменьшается с возрастанием порядка опорного метода;

3) несмотря на уменьшение числа строк экстраполяционной таблицы, накладные расходы увеличиваются с возрастанием порядка опорного метода, причем при  $T_F \gg t_{оп} = 1,5 \div 2$  раза для методов смежных порядков, и это различие уменьшается с увеличением порядка; при  $T_F \approx t_{оп}$  накладные расходы существенно возрастают с увеличением порядка  $r^0$ , т. е. в десятки—сотни раз;

4) накладные расходы возрастают с увеличением размерности задачи.

Суммируя все полученные результаты, можно сделать вывод о том, что для уменьшения накладных расходов при применении технологии локальной экстраполяции следует выбирать опорный метод малого порядка, в частности второго, с достаточными свойствами устойчивости. Однако есть еще один момент, требующий анализа. Теоретически нет никаких ограничений на длину экстраполяционной таблицы, но конечность памяти машины и накапливание ошибок округления, ограничивают длину экстра-

поляционной таблицы сверху и, как правило,  $k_{\max} \leq 10$ . Поэтому для получения высокоточных приложений ( $10^{-15} - 10^{-20}$ ), где вычисления правой части СОДУ достаточно сложны, можно использовать только технологию локальной экстраполяции Ричардсона на базе опорного метода высокого порядка.

Альтернативным способом определения локальной апостериорной погрешности при решении задачи Коши являются ВМРК или методы вложенных форм. 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  $r$  и  $\hat{r}$  с одним и тем же шагом [2]:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \sum_{l=1}^s b_l \mathbf{k}_l; \quad \hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + h_n \sum_{l=1}^{s'} \hat{b}_l \mathbf{k}_l;$$

$$\mathbf{k}_l = \mathbf{f}(x_n + c_l h_n; \mathbf{y}_n + h_n \sum_{l=1}^{l-1} a_{li} \mathbf{k}_i); \quad l = 1, \dots, s;$$

$$d_{n+1} = \|(\hat{\mathbf{y}}_{n+1} - \mathbf{y}_{n+1})\|.$$

Анализ эффективности полученных параллельных алгоритмов проведен на основе следующих показателей:

время решения с помощью последовательного алгоритма  $T_1$ ;

время решения с помощью параллельного алгоритма без учета и с учетом обменных операций:  $T_p = T_{p\text{выч}} + T_{p\text{об}}$  (потенциальный и реальный параллелизм);

коммуникационная сложность алгоритма в зависимости от выбранной топологии соединения процессоров и модели передачи данных;

максимальная степень параллелизма  $D$ ;

коэффициенты потенциального и реального ускорения  $S$ , эффективность  $E$  параллельного алгоритма и масштабируемость на основе функции изоэффективности  $f_E$ .

Динамические характеристики параллельных алгоритмов существенно зависят от многих факторов: размера задачи и времени обращения к правой части исходной СОДУ, типа параллельного компьютера, размерности процессорных полей, времени выполнения арифметических операций, временных коммуникационных констант ( $t_s$  и  $t_w$ ), топологии межпроцессорного соединения. При расчете динамических характеристик алгоритмов любая арифметическая операция с плавающей точкой выполняется за одно и то же время независимо от вида операции. Это предположение справедливо для большинства современных компьютеров RISC-архитектуры.



Под масштабируемостью понимаем возможность алгоритма обеспечить постоянную эффективность вычислений при увеличении числа процессоров (при увеличении размерности задачи). В качестве оценки масштабируемости принимаем функцию изоэффективности [3], определяющую зависимость числа используемых процессоров для обеспечения постоянного уровня эффективности параллельных вычислений от размерности задачи. Тогда показатель вычислительной сложности задачи (число операций) запишем в виде  $W = Kf_E(p)$ , где  $K$  — коэффициент, зависящий только от значения показателя эффективности. Построение функции изоэффективности — это попытка соединить в едином аналитическом выражении все перечисленные характеристики и оценить степень их влияния на качество параллельного алгоритма.

Исследование теоретического выполнения параллельных методов решения нелинейной задачи Коши на основе ЯМРК и проведенный численный эксперимент позволяют сделать следующие выводы:

наиболее оптимальными параллельными методами по эффективности и ускорению являются вложенные методы;

преимущества методов, основанных на локальной экстраполяции, проявляются при получении высокоточных решений ( $10^{-15}$ – $10^{-20}$ ) и для СОДУ со сложными правыми частями.

Лучшие характеристики параллелизма ЯМРК, близкие к потенциальным (линейное ускорение и единичная эффективность), достигаются при сочетании нескольких факторов: сложной правой части (при  $T_F \gg t_{оп}$ ), выполнении условия  $p \leq D = m$ , использовании топологии гиперкуб с синхронным или асинхронным выполнением обменов.

Определение реальных характеристик параллелизма осуществлялось с помощью пакета Mathematica® (Wolfram Research Inc.), численный эксперимент проводился на базе тестов для СОДУ с использованием библиотеки MPI.

**Параллельные методы решения жестких задач Коши с использованием полностью неявных разностных схем.** Исследование численных алгоритмов решения задачи Коши, основанных на конечно-разностных схемах, показало, что параллельные свойства таких алгоритмов во многом определяются видом лежащей в их основе численной схемы. Большим потенциальным параллелизмом обладают явные методы, однако присущие этим схемам недостатки, одним из которых является их условная устойчивость, ограничивает область применения таких алгоритмов. В этой связи значительный интерес представляют неявные схемы, которые, несмотря на большую вычислительную сложность, не имеют альтернативы среди одношаговых методов при решении жестких задач.

Численное решение (1) полностью неявным  $s$ -стадийным методом типа Рунге—Кутты (ПНМРК) можно получить последовательно по шагам с помощью следующей формулы:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{k}_i; \quad (3)$$

$$\mathbf{k}_i = F(x_n + c_i h; \mathbf{y}_n + h \sum_{j=1}^s a_{ij} \mathbf{k}_j), \quad i = \overline{1, s}.$$

К достоинствам полностью неявных, одношаговых методов общего вида (на основе квадратурных формул Радо и Лобатто) следует отнести характеристики устойчивости и точности, достаточные для решения жестких задач. Так, например,  $s$  — стадийный метод РадоIA имеет порядок практически в два раза больше, чем число стадий и обладает  $A$ -устойчивостью [4]. Однако при решении ОДУ с использованием ПНМРК  $m$ s неизвестных  $K = \|k_{ij}\|, i = 1, \dots, s, j = 1, \dots, m$ , следует определять одновременно, что существенно усложняет задачу.

Для решения системы (3) используем метод функциональной итерации:

$$\mathbf{g}_i^{(0)} = 0; \quad \mathbf{k}_i^{(0)} = F(x_n + c_i h, \mathbf{y}_n);$$

$$\mathbf{g}_i^{(l)} = a_{i1} \mathbf{k}_1^{(l)} + a_{i2} \mathbf{k}_2^{(l)} + \dots + a_{is} \mathbf{k}_s^{(l)};$$

$$\mathbf{k}_i^{(l)} = F[x_n + c_i h, \mathbf{y}_n + h \mathbf{g}_i^{(l-1)}], \quad i = 1, \dots, s; \quad l = 1, \dots, N.$$

В результате итерационного процесса, повторенного  $l$  раз, получено  $\mathbf{k}_i^{(l)}, i = 1, s$ , в виде  $l$ -й аппроксимации для вектора шагового коэффициента  $k_i$ . Скорость сходимости итерационного процесса можно определить следующим образом:  $r^* = \min(r, N + 1)$ , где  $r$  — порядок используемого ПНМРК. Как и для всех рассмотренных методов, распараллеливание ПНМРК основано на выполнении одного шага интегрирования. Все множество процессоров разбиваем на  $s$  групп по числу шаговых векторов, каждый процессор в группе вычисляет всего  $\lceil m/p \rceil$  компонент  $\mathbf{k}_i$  и затем передает их всем процессорам группы. Вычисление решения в следующей точке требует реализации операции «все—всем».

Межгрупповой обмен может быть осуществлен двумя способами:

1) каждый первый процессор в группе передает вектор  $\mathbf{k}_i$  в первый элемент каждой другой группы плюс параллельный групповой обмен в каждой группе;

2) межгрупповая передача все—всем.

Определение локальной погрешности для неявных схем проводилось на основе локальной экстраполяции Ричардсона и методов вложенных форм. Разработка и анализ эффективности полученных параллельных

алгоритмов ПНМРК и ЯМРК аналогичны. Трудоемкость неявных ВМРК и ПНМРК существенно зависит от сложности функции вектора  $F$ . Для выполнения сложных правых частей уравнений явные ВМРК требуют меньше времени, чем неявные. Этот эффект увеличивается в случае, если время, расходуемое на арифметические операции, больше времени обмена. В общем случае вложенные ЯМРК имеют меньшую вычислительную сложность, чем ПНМРК и этот эффект увеличивается с увеличением порядка метода. Для ВМРК трудоемкость коммуникационных операций зависит от числа этапов метода и от числа итераций, причем с увеличением порядка метода число этапов возрастает быстрее, чем число итераций. При  $T_F \gg t_{оп}$  потенциальный коэффициент ускорения  $S_{п}$  равен  $O(s)$ , а если  $T_F \approx t_{оп}$ , то  $S_{п} = O(p)$ .

**Особенности оценки апостериорной локальной погрешности при параллельном решении линейной задачи Коши.** Решение задачи Коши для систем линейных обыкновенных дифференциальных уравнений (СЛОДУ) можно получить, используя специальный экспоненциальный метод, основанный на точном представлении решения в аналитической форме и вычислении матричной экспоненты.

Общий вид задачи Коши для однородных СЛОДУ с постоянными коэффициентами следующий:

$$\begin{aligned} \mathbf{y}'(x) &= A \mathbf{y}(x); \\ \mathbf{y}(x_0) &= \mathbf{y}_0; \\ A &= \|a_{ij}\|, i, j = \overline{1, m}, A = \text{const}, \end{aligned} \quad (4)$$

где  $\mathbf{y}$  и  $\mathbf{y}_0$  — вектор неизвестных и вектор начальных условий;  $A$  — матрица коэффициентов линейной системы.

Точным решением задачи Коши вида (4) является матричная экспонента:

$$\mathbf{y}(x_0 + h) = F(hA)\mathbf{y}_0, F(hA) = e^{Ah} = \sum_{k=0}^{\infty} \frac{(hA)^k}{k!}.$$

Приближенное решение (4) построим, аппроксимировав матричную экспоненту отрезком ряда Тейлора при малом значении  $h$ :

$$F(hA) \cong \sum_{k=0}^r \frac{(hA)^k}{k!}, \mathbf{y}_{n+1} = F^n(Ah)\mathbf{y}_0.$$

Затем, используя некоторый алгоритм умножения матриц, вычислим численное решение. Аналогично решаем задачу Коши для неоднородной системы

$$\begin{aligned} \mathbf{y}'(x) &= A\mathbf{y}(x) + \mathbf{b}; \quad \mathbf{y}(x_0) = \mathbf{y}_0; \\ A &= \|a_{ij}\|, i, j = \overline{1, m}, A, \mathbf{b} = \text{const}. \end{aligned}$$

Ее решение имеет вид  $\mathbf{y}_{n+1} = \mathbf{y}_n + R(h)(A\mathbf{y}_n + b)$ , где  $R(h)$  — матричная функция,

$$R(h) = \sum_{i=1}^{\infty} \frac{1}{i!} A^{i-1} h^i,$$

которая также может быть аппроксимирована отрезком ряда Тейлора:

$$R(h) \cong \sum_{i=1}^r \frac{1}{i!} A^{i-1} h^i.$$

Основной особенностью метода является возможность введения подготовительного этапа, выполняемого до начала интегрирования, в который можно включить наиболее ресурсоемкие операции нахождения матричных форм, а именно матричное умножение, не зависящее от шага интегрирования. Матричная функция  $F(h)$  обладает свойством  $F(h) = F(h/2)F(h/2)$ , что позволяет достаточно легко определять локальную погрешность на основе дублирования шага и локальной экстраполяции. Применение экспоненциального метода для вложенных форм делает практически ненужными вычисления по формуле высшего порядка:

$$\hat{\mathbf{y}}^{n+1} = \left[ E + hA + \frac{h^2 A^2}{2} + \dots + \frac{h^r A^r}{r!} + \frac{h^{r+1} A^{r+1}}{(r+1)!} \right] \mathbf{y}^n;$$

$$\mathbf{y}^{n+1} = \left[ E + hA + \frac{h^2 A^2}{2} + \dots + \frac{h^r A^r}{r!} \right] \mathbf{y}^n;$$

$$\|\hat{\mathbf{y}}^{n+1} - \mathbf{y}^{n+1}\| = \frac{h^{r+1} A^{r+1}}{(r+1)!} \mathbf{y}^n.$$

Для плотнозаполненных матриц имеется два принципиально различных класса последовательных алгоритмов умножения матриц: традиционные и рекурсивные методы быстрого умножения на базе алгоритмов Штрассена—Винограда [5]. Вычислительная сложность матричного умножения  $C = A \times B$  для исходных данных размерности  $m \times m$  составляет:

$O(m^3)$  — для традиционных алгоритмов;

$O(m^{2.81})$  — для быстрого умножения по Штрассену;

$O(m^{2.376})$  — по методу Винограда.

Метод Штрассена—Винограда состоит из семи блочных умножений матриц и 15-ти блочных сложений (вычитаний) матриц:

$$S_1 = A_{21} + A_{22}, M_1 = S_2 S_6, T_1 = M_1 + M_3,$$

$$S_2 = S_1 - A_{11}, M_2 = A_{11} B_{11}, T_2 = T_1 + M_4,$$

$$\begin{aligned}
 S_3 &= A_{21} - A_{12}, M_3 = A_{12}B_{21}, T_3 = M_5 + M_6, \\
 S_4 &= A_{12} - S_2, M_4 = S_3S_7, C_{11} = M_2 + M_3, \\
 S_5 &= B_{12} - B_{11}, M_5 = S_1S_5, C_{12} = T_1 + T_3, \\
 S_6 &= B_{22} - S_5, M_6 = S_4B_{22}, C_{21} = T_2 - M_7, \\
 S_7 &= B_{22} - B_{12}, M_7 = A_{22}S_8, C_{22} = T_2 + M_5, \\
 S_8 &= S_6 - B_{12}; \\
 A &= \left\langle \begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right\rangle, B = \left\langle \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right\rangle, C = \left\langle \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right\rangle.
 \end{aligned}$$

В оригинале алгоритм Штрассена—Винограда — это алгоритм рекурсивного умножения блочных матриц половинного размера, где каждый блок квадратный, т.е. размерности матриц должны быть четными числами. Идея Штрассена может быть применена рекурсивно: если исходные матрицы  $A$  и  $B$  имеют размеры  $m \times m$ , то алгоритм быстрого умножения можно применять многократно (на самом нижнем уровне получим блоки  $1 \times 1$ ).

В классическом варианте, как известно, алгоритмы рекурсивного умножения матриц очень чувствительны к ошибкам округления, т.е. плохо обусловлены, что ограничило их область применения для последовательных вычислительных систем. В связи с этим предлагается использовать комбинации алгоритмов рекурсивного и блочного систолического умножения, что позволяет избежать указанных недостатков. Процесс выполнения систолического умножения матриц заключается в предварительном косом сдвиге левого сомножителя по строкам влево и правого — вверх по столбцам. Затем пошагово выполняются  $m$  умножений элементов,  $(m - 1)$  одиночный сдвиг и  $(m - 1)$  сложение:  $T_{A \times A} = mt_y + (m - 1)t_{сл} + 3(m - 1)T_{T-T}$ , где  $t_y$ ,  $t_{сл}$ ,  $T_{T-T}$  — длительность выполнения одиночных операций умножения, сложения и обмена типа точка—точка.

Алгоритм систолического умножения является одним из наиболее эффективных при реализации на синхронных параллельных структурах типа SIMD. В то же время для асинхронных вычислительных систем с целью улучшения динамических характеристик необходимо увеличить зернистость алгоритма, поэтому предлагается использовать блочные модификации метода.

На основе экспоненциального метода разработаны три параллельных алгоритма с учетом альтернативных способов оценки локальной погрешности:

- экспоненциальный и правило Рунге;
- вложенный экспоненциальный;
- экспоненциальный с локальной экстраполяцией.

Для построения и исследования эффективности параллельных алгоритмов решения использована та же технология, что и для ЯМРК и НМРК общего вида. Результаты проведенных исследований позволяют сделать следующие выводы.

**Выводы.** Решение линейной задачи Коши на основе модифицированного экспоненциального метода с использованием локальной экстраполяции и вложенных форм имеет практически одинаковую вычислительную сложность, хотя локальная экстраполяция обеспечивает целую таблицу вложенных решений, а также возможность увеличения точности полученного решения при увеличении порядка опорного метода и  $h^2$ -экстраполяции.

Наиболее эффективной топологией для реализации методов является решетка и ее замкнутый эквивалент — тор, поскольку на такой топологической схеме наиболее эффективно выполняются матричные операции. Этот способ решения требует меньшего объема вычислений, чем стандартные методы и позволяет построить более простые и эффективные параллельные алгоритмы решения линейной задачи Коши. Предлагаемый метод особенно эффективен для решения систем с большой константой Липшица, в частности для жестких систем уравнений.

Таким образом, проведенные теоретические исследования и результаты вычислительного эксперимента [6—11] свидетельствуют о том, что несмотря на большие накладные расходы, алгоритмы решения нелинейной задачи Коши для СОДУ большой размерности на основе одношаговых методов при тщательном подборе и правильном сочетании аргументов задачи и параметров вычислительной системы могут быть достаточно эффективно использованы на параллельных структурах с топологиями гиперкуб и тор при синхронной и асинхронной передаче данных.

Перспективным направлением дальнейших исследований является разработка масштабируемых алгоритмов решения задачи Коши с встроенными методами оценки апостериорной локальной погрешности для блочных параллельных методов, а также исследование влияния параметров параллельной системы на динамические характеристики полученных параллельных алгоритмов.

Parallel numerical algorithms for estimation of a posterior local error are proposed on the basis of explicit and totally implicit difference scheme. The peculiarities of the algorithm application for solving linear Cauchy problem are studied. The calculation schemes of the method reflection on parallel structures of different topologies, such as line — ring, matrix — tore, hypercube are developed. Comparative characteristics of potential and real parallelism are given. The results of numerical experiments in test systems are also obtained.

1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб. : БХВ — Петербург, 2002. — 608 с.
2. Хайрер Э., Нёрсентт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи: Пер. с англ. — М. : Мир, 1990. — 512 с.
3. Kumar V., Gupta A. Analyzing scalability of parallel algorithms and architectures // J. of Parallel and Distributed Computing. — 1994. — **22**, № 3. — P. 379—391.
4. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. — М. : Мир, 1999. — 685с.
5. Голуб Дж., Ван Лоун Ч. Матричные вычисления: Пер. с англ. — М. : Мир, 1999. — 548 с.
6. Фельдман Л. П., Назарова И. А. Применение технологии локальной экстраполяции для высокоточного решения задачи Коши на SIMD-структурах// Науч. тр. Донецкого национального технического университета. Вып. 70. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003) — Донецк : ДонНТУ, 2003. — С. 98—107.
7. Назарова И. А., Фельдман Л. П. Масштабируемый параллельный алгоритм численного решения линейных СОРУ для компьютеров с распределенной памятью // Тез. докл. V Международной конф. по неравновесным процессам в соплах и струях (NPNJ-2004), Самара, 5—10 июля 2004 г. — М. : Вузовская книга, 2004. — С. 153—155.
8. Назарова И. А., Фельдман Л. П. Эффективность параллельного численного решения нежестких систем обыкновенных дифференциальных уравнений с контролем локальной погрешности// Тез. докл. XX Международного семинара по струйным, отрывным и нестационарным течениям. — СПб. : ИПЦ СПбГУТД, 2004. — С. 201—202.
9. Назарова И. А. Эффективность численного решения нежестких СОРУ с контролем локальной погрешности для компьютеров с распределенной памятью // Искусственный интеллект. — 2004. — № 3. — С. 212—216.
10. Назарова И. А. Параллельные полностью неявные методы численного решения жестких задач для СОРУ // Там же. — 2005. — № 3. — С. 185—193.
11. Фельдман Л. П., Назарова И. А. Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений // Математическое моделирование. — 2006. — **18**, № 9. — С. 17—31.

Поступила 04.10.06

*ФЕЛЬДМАН Лев Петрович, д-р техн. наук, профессор кафедры прикладной математики Донецкого национального технического университета. В 1951 г. окончил Московский госуниверситет. Область научных исследований — параллельные вычисления и моделирование вычислительных систем.*

*НАЗАРОВА Ирина Акоповна, ст. преподаватель кафедры прикладной математики Донецкого национального технического университета. В 1981 г. окончила Донецкий политехнический ин-т. Область научных исследований — параллельные вычисления.*