

УДК 681.3

*И.А. Назарова*

Донецкий национальный технический университет, Украина, Донецк,  
nazarova@r5.dgtu.donetsk.ua

## Параллельные полностью неявные методы численного решения жестких задач для СОДУ

Предложены параллельные численные алгоритмы неявных методов Рунге – Кутта для решения жестких задач Коши. Разработаны вычислительные схемы отображения методов на параллельные структуры различной топологии: линейка/кольцо, решетка/тор, гиперкуб. Получены сравнительные характеристики потенциального и реального параллелизма, проведены численные эксперименты на системе тестов.

Широко распространенным способом создания параллельных методов является распараллеливание хорошо исследованных и многократно апробированных последовательных численных алгоритмов [1]. Если ограничиться рассмотрением численных алгоритмов решения задачи Коши, основанных на конечно-разностных схемах, то параллельные свойства таких алгоритмов во многом определяются видом лежащей в их основе численной схемы. Наибольшим потенциальным параллелизмом обладают явные методы [2-4], однако присущие этим схемам недостатки, одним из которых является их условная устойчивость, ограничивают область применения таких алгоритмов. В этой связи значительный интерес представляют неявные схемы, так как, несмотря на большую вычислительную сложность и слабый внутренний параллелизм, для решения жестких задач эти методы не имеют альтернативы.

Численно решается задача Коши для обыкновенного дифференциального уравнения первого порядка с известными начальными условиями:

$$\begin{cases} y' = f(x, y); \\ y(x_0) = y_0. \end{cases} \quad (1)$$

Методы Рунге – Кутта с числом стадий, равным  $s$ , в общем виде определяются формулами (2) и (3):

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j), \quad i = \overline{1, s} \quad (2)$$

$$y_{n+1} = y_n + h \cdot \sum_{i=1}^s b_i \cdot k_i. \quad (3)$$

Коэффициенты  $a_{ij}, c_i, b_i$  определяют уникальный вариант метода Рунге – Кутта и выбираются из соображений точности. Если  $a_{ij} = 0$  при  $i \leq j$ , то метод Рунге – Кутта

является явным. В том случае, если  $a_{ij} = 0$  при  $i < j$  (нижняя треугольная матрица) и хотя бы одно значение  $a_{ii} \neq 0$ , то  $k_i$  определяются неявно из уравнения

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^i a_{ij} \cdot k_j), i = \overline{1, s}. \quad (4)$$

Такой метод называют диагонально-неявным (ДНРК-метод).

Если в диагонально-неявном методе Рунге – Кутта все диагональные элементы одинаковы:  $a_{ii} = \gamma$ , то такой метод называют однократно диагонально неявным (ОДНРК-метод). Во всех остальных случаях имеем полностью неявный метод Рунге – Кутта (ПНРК), его представление в виде таблицы Батчера изображено на рис. 1.

$c_1$	$a_{11}$	$a_{12}$	...	$a_{1s}$
$c_2$	$a_{21}$	$a_{22}$	...	$a_{2s}$
...	...	...	...	...
$c_s$	$a_{s1}$	$a_{s2}$	...	$a_{ss}$
	$b_1$	$b_2$	...	$b_s$

Рисунок 1 – Таблица Батчера для s-стадийного полностью неявного метода Рунге – Кутта

Рассмотрим параллельный алгоритм численного решения одного дифференциального уравнения полностью неявным методом Рунге – Кутта на основе квадратурных формул Радо и Лобатто [5]. Именно эти неявные методы имеют хорошие характеристики устойчивости и точности. Так, например, s-стадийный метод РадоIA имеет порядок практически в 2 раза больше, чем число стадий, и обладает А-устойчивостью. Однако характеристики параллелизма для всех полностью неявных методов не зависят от того, какой конкретно метод используется, поэтому далее процесс распараллеливания показан в общем виде:

$$\begin{cases} k_1 = f[x_n + c_1 h, y_n + h(a_{11}k_1 + a_{12}k_2 + \dots + a_{1s}k_s)]; \\ k_2 = f[x_n + c_2 h, y_n + h(a_{21}k_1 + a_{22}k_2 + \dots + a_{2s}k_s)]; \\ \dots \\ k_s = f[x_n + c_s h, y_n + h(a_{s1}k_1 + a_{s2}k_2 + \dots + a_{ss}k_s)]. \end{cases} \quad (5)$$

Применение полностью неявного метода Рунге – Кутта для решения обыкновенных дифференциальных уравнений (ОДУ) требует определения шаговых  $k_i, i = 1, \dots, s$  коэффициентов, которые связаны системой s нелинейных уравнений.

Для решения такой системы используем метод последовательных итераций:

$$\begin{cases} g_i^{(0)} = 0; \\ g_i^{(l)} = a_{i1}k_1^{(l)} + a_{i2}k_2^{(l)} + \dots + a_{is}k_s^{(l)}; \\ i = 1, \dots, s. \end{cases}$$

$$\left\{ \begin{array}{l} k_1^{(0)} = f(x_n + c_1 h, y_n + h \cdot g_1^{(0)}); \\ k_2^{(0)} = f(x_n + c_2 h, y_n + h \cdot g_2^{(0)}); \\ \dots \\ k_s^{(0)} = f(x_n + c_s h, y_n + h \cdot g_s^{(0)}). \end{array} \right. \dots \left\{ \begin{array}{l} k_1^{(N+1)} = f[x_n + c_1 h, y_n + h \cdot g_1^{(N)}]; \\ k_2^{(N+1)} = f[x_n + c_2 h, y_n + h \cdot g_2^{(N)}]; \\ \dots \\ k_s^{(N+1)} = f[x_n + c_s h, y_n + h \cdot g_s^{(N)}]. \end{array} \right.$$

Анализ эффективности полученных параллельных алгоритмов производился на основе следующих показателей:

- теоретической оценки сложности решения задачи;
- времени решения при помощи последовательного алгоритма;
- времени решения при помощи параллельного алгоритма без учета и с учетом обменных операций (потенциальный и реальный параллелизм);
- анализа коммуникационной сложности алгоритма в зависимости от выбранной топологии соединения процессоров и модели передачи данных;
- коэффициенты ускорения, эффективности параллельного алгоритма.

Время последовательного алгоритма для одного уравнения при реализации полностью неявного метода Рунге – Кутта может быть описано следующим соотношением:

$$T_{\text{посл.}} = s \cdot T_f + N \cdot [s^2 \cdot t_{\text{умн}} + s^2 \cdot t_{\text{сл}} + s \cdot T_f] + s \cdot t_{\text{умн}} + s \cdot t_{\text{сл}},$$

где  $T_f$  – время вычисления функции  $f$  – правой части исходного дифференциального уравнения;  $t_{\text{умн}}$  – время выполнения операции одиночного умножения;  $t_{\text{сл}}$  – время выполнения операции одиночного сложения;  $N$  – число итераций в методе простых итераций. Для представленного параллельного алгоритма (рис. 2) время выполнения на  $p$  процессорах без учета обменов и других накладных расходов задается соотношением

$$T_{\text{нар}} = T_f + N \cdot [s \cdot t_{\text{умн}} + s \cdot t_{\text{сл}} + T_f] + t_{\text{умн}} + t_{\text{сл}}.$$

Соответственно, коэффициент ускорения равен

$$K_{\text{уск}} = T_{\text{посл.}} / T_{\text{нар}} = \frac{s \cdot T_f + N \cdot [2s^2 \cdot t_{\text{он}} + s \cdot T_f] + 2s \cdot t_{\text{он}}}{T_f + N \cdot [2s \cdot t_{\text{он}} + T_f] + 2 \cdot t_{\text{он}}}.$$

При подсчете коэффициента ускорения считалось, что  $t_{\text{сл}} = t_{\text{умн}} = t_{\text{он}}$  – любая арифметическая операция с плавающей точкой выполняется за одно и то же время независимо от вида операции (флоп). Это предположение справедливо для большинства современных компьютеров RISC архитектуры. Для неявных методов Рунге – Кутта время интегрирования в основном определяется числом обращений к функции правой части уравнения (1) и соотношением между временем вычисления функции  $f$  и флопом. Проведенные эксперименты показали, что для сложных функций правой части (1) (время вычисления функции  $f$  значительно превышает время одного флопа) коэффициент ускорения практически равен числу процессоров, а коэффициент эффективности – единице:

$$K_{\text{уск}} \approx \frac{s \cdot T_f + N \cdot s \cdot T_f}{T_f + N \cdot T_f} = s = p.$$

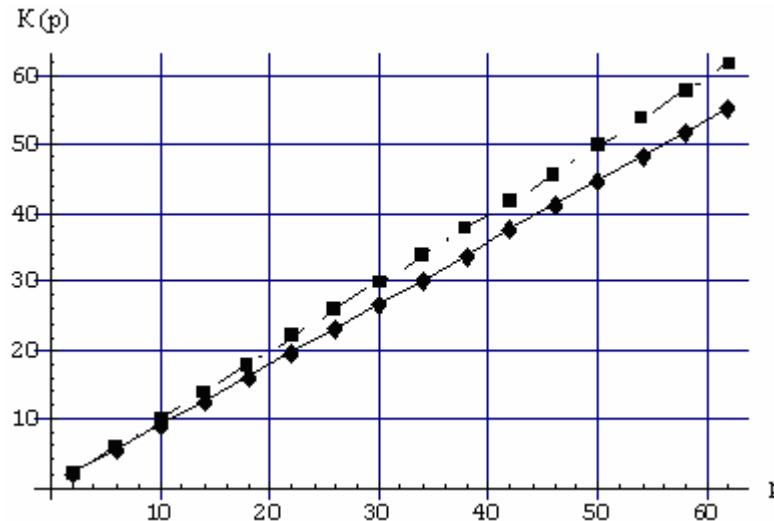


Рисунок 2 – Зависимость коэффициента ускорения параллельного алгоритма решения ОДУ полностью неявным методом Рунге – Кутта от количества процессоров при различном числе итераций

Эффективность параллельных вычислений во многом определяется трудоемкостью коммуникационных операций, выполняемых в параллельных алгоритмах. В этой связи важным представляется адекватность моделей и точность в оценке сложности операций передачи данных в многопроцессорные вычислительные системы (МПВС). Общее время на обмен данными между процессорами при реализации неявного метода Рунге – Кутта определяется следующими факторами. Во-первых, оно существенно зависит от выбранной топологии соединения процессоров; безусловно, от числа используемых процессоров; от коммуникационных констант, которые, в свою очередь, различаются для синхронной и асинхронной моделей вычислительные системы (ВС); метода передачи данных: сообщений или пакетов; и конечно, от характеристик самой задачи и алгоритма.

Для оценки времени выполнения операции передачи одного сообщения объемом  $n$  байт между двумя задачами, локализованными на различных процессорах (обмен «точка – точка»), при распределенной памяти использовалась следующая модель [6]:

$$T_{m-m} = t_n + t_k \cdot n \cdot l, \quad t_k = \frac{y}{B}, \quad (6)$$

где  $t_n$  – латентность, длительность подготовки сообщения для передачи;  $l$  – длина маршрута;  $t_k$  – время передачи одного байта;  $y$  – число байт в слове;  $B$  – пропускная способность канала передачи данных (байт/секунда).

Эта модель подразумевает использование способа передачи неделимых блоков информации, т.е. сообщений, поскольку объемы пересылаемых данных невелики (слово или 4 байта).

При реализации параллельного алгоритма для одного ОДУ требуется  $N + 2$  операции передачи данных по типу «все – всем» с учетом модели (6). Рассмотрим широко используемые топологии, такие, как линейка/кольцо, решетка/тор и гиперкуб. К числу наиболее распространенных оптимальных алгоритмов передачи данных относится класс методов по координатной маршрутизации [7], [8]. Идея этих методов заключается в том, что поиск путей передачи данных осуществляется последовательно для каждой размерности рассматриваемой топологии.

Для кольцевой топологии каждый процессор может инициировать рассылку своего сообщения в каком-либо выбранном направлении по кольцу. Без существенных ограничений предполагаем, что любой процессор имеет возможность осуществлять одновременно прием и передачу данных (двунаправленные линки) (рис. 3). Таким образом, для топологии кольцо коммуникационная составляющая первого алгоритма равна  $T_{обм} = (N + 2)(p - 1) \cdot T_{m-m}$ .

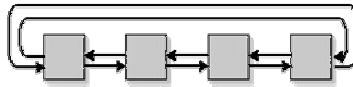


Рисунок 3 – Топология кольцо, или 1D-тор

Для топологии «решетка/тор» (рис. 4) множественная рассылка по типу «все – всем» может быть осуществлена обобщением способа передачи для кольцевой структуры, вначале организуется передача данных по горизонтали, а затем – по вертикали решетки:  $T_{обм} = 2(N + 2) \cdot (\sqrt{p} - 1) \cdot t_n + (N + 2) \cdot (p - 1)t_k$ .

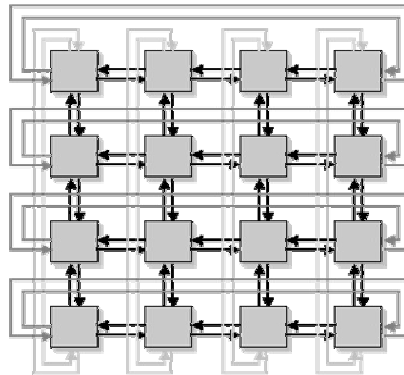


Рисунок 4 – Топология решетка, или 2D-тор

Для гиперкуба (рис. 5) алгоритм множественной рассылки может быть получен обобщением способа передачи «все – всем» для решетки на размерность куба:  $lp = \log_2 p$ .

$$T_{обм} = (N + 2) \cdot \sum_{i=1}^{lp} (t_n + 2^{i-1} t_k) = t_n \cdot (N + 2) \cdot \log_2 p + t_k \cdot (p - 1)(N + 2).$$

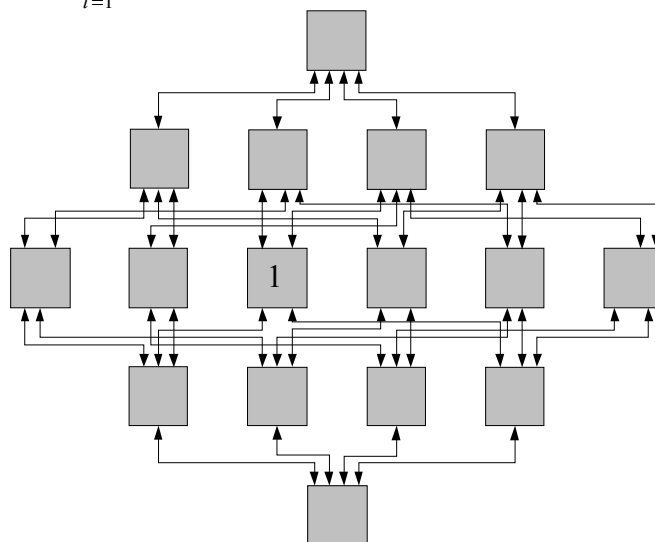


Рисунок 5 – Топология гиперкуб

Более информативной величиной является отношение коммуникационных затрат к общим накладным расходам параллельного алгоритма (рис. 6):

$$Z = \frac{T_{\text{обмена}}}{T_{\text{общ}}} = \frac{T_{\text{обмена}}}{T_{\text{обмена}} + T_{\text{выч}}}$$

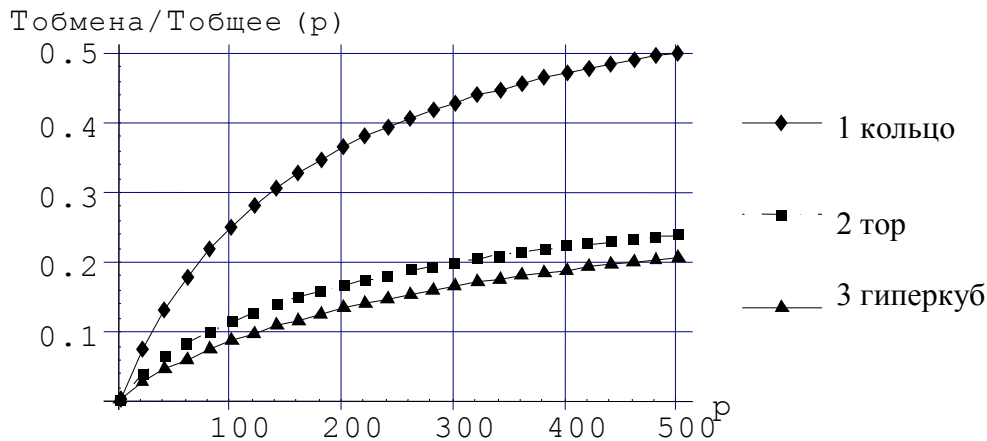


Рисунок 6 – Сравнительный анализ долей коммуникационных операций к общим накладным расходам при различных топологиях для синхронной модели

Выберем оптимальную топологию соединения процессоров, если таковая существует при варьировании всех остальных характеристик. Выбор топологии производился при анализе множества следующих характеристик:  $T_{\text{обм}}$ ,  $Z$ ,  $K_{\text{уск}}$ ,  $E$  как некоторых функций  $\varphi(p, t_n, t_k, t_{\text{он}}, T_f)$ .

Анализ предложенных топологий показал (рис. 7), что, безусловно, наихудшим вариантом для рассматриваемого алгоритма является соединение линейка/кольцо, так как при использовании характеристик многопроцессорной системы iPSC/860, ( $t_n = 175 \mu\text{с}$ ,  $t_k = 0,39 \mu\text{с}$ ) [8] имеем 70 % операций обменов и эффективность параллельных вычислений, равную 0,2.

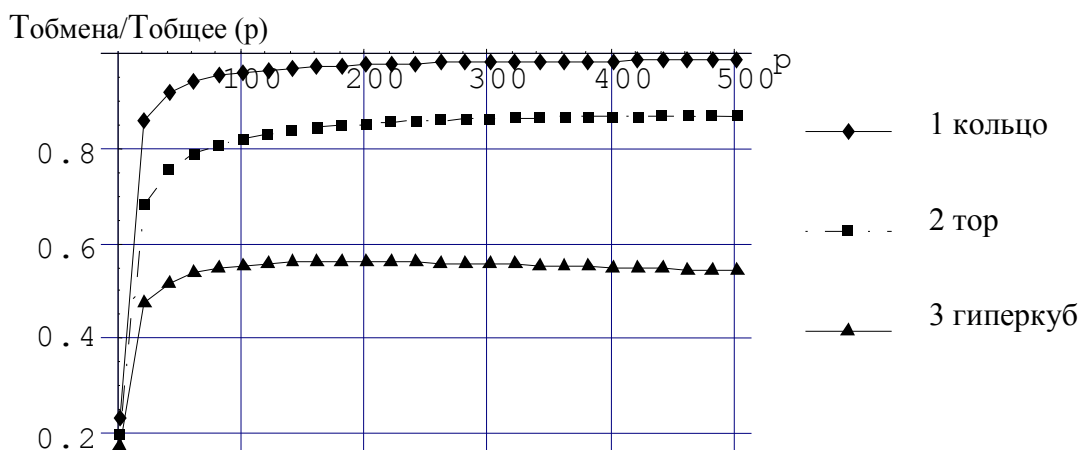


Рисунок 7 – Сравнительный анализ долей коммуникационных операций к общим накладным расходам, при различных топологиях для асинхронной модели

Топологии гиперкуб и тор часто дают достаточно близкие характеристики, но даже при худшей комбинации всех аргументов (асинхронная передача, малая сложность вычисления функции  $f$ ) соединение процессоров по типу гиперкуб предпочтительнее. Так, например, при синхронной передаче данных с коммуникационными характеристиками по типу СМ-2 ( $t_n = 3\mu\text{с}, t_k = 0,5\mu\text{с}$ ) [9] и достаточно сложной правой части уравнения (1) ( $T_f = 1000t_{on}$ ) время передачи данных составляет всего 15 % от общих накладных расходов, а коэффициент эффективности стремится к 0,9 при увеличении числа процессоров (рис. 8).

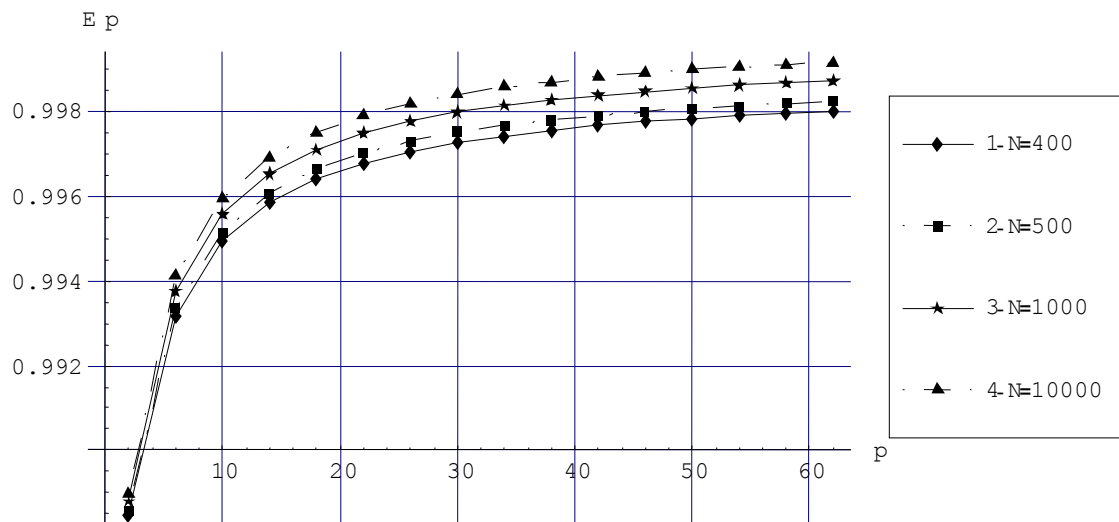


Рисунок 8 – Зависимость коэффициента эффективности  $E$  от числа процессоров для различного числа итераций при решении полностью неявным методом Рунге – Кутты

Для системы из  $m$  ОДУ:

$$\begin{cases} \bar{y}' = F(x, \bar{y}); \\ \bar{y}(x_0) = \bar{y}_0. \end{cases}$$

где  $\bar{y} = (y_1, y_2, \dots, y_m)^T$ ,  $F = (f_1, f_2, \dots, f_m)^T$ , полностью неявный метод Рунге – Кутты принимает вид

$$\bar{y}_{n+1} = \bar{y}_n + h \cdot \sum_{i=1}^s b_i \cdot \bar{k}_i,$$

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

При решении СОДУ с использованием полностью неявных методов Рунге – Кутта все  $m \cdot s$  неизвестных должны определяться одновременно, что существенно усложняет задачу (всего шаговых коэффициентов  $s$  и размерность каждого вектора  $\bar{k}_i$  равна  $m$ :

$$\bar{k}_i = (k_{i1}, k_{i2}, \dots, k_{im}).$$

$$K = \|k_{ij}\|, \quad i = 1, \dots, s; \quad j = 1, \dots, m$$

Времена выполнения последовательного и потенциального параллельного алгоритмов ПНМРК для СОДУ включают две составляющие: время на определение матрицы шаговых коэффициентов на основе итерационного процесса и собственно время вычисления решения на следующем шаге:

$$\begin{aligned} T_{\text{носл.}} &= T_{\bar{k}} + T_{\bar{y}}, \\ T_{\bar{k}} &= Ns \cdot [ms(t_{\text{ym}} + t_{\text{cl}}) + \sum_{i=1}^m T_{f_i}], \\ T_{\bar{y}} &= m(s+1) \cdot t_{\text{ym}} + [m(s-1) + 1] \cdot t_{\text{cl}}. \end{aligned}$$

При условии, что  $t_{\text{ym}} = t_{\text{cl}} = t_{\text{on}}$

$$\begin{aligned} T_{\text{носл.}} &= (2Nms^2 + 2ms + m) \cdot t_{\text{on}} + NsmT_f \\ T_{\text{нар}} &= N \left[ s \left\lfloor \frac{ms}{p} \right\rfloor (t_{\text{ym}} + t_{\text{cl}}) + \sum_{i=1}^m T_{f_i} \right] + \left\lfloor \frac{m}{p} \right\rfloor [(s+1)t_{\text{ym}} + st_{\text{cl}}] = \\ &= \left[ 2Ns \cdot \left\lfloor \frac{ms}{p} \right\rfloor + (2s+1) \cdot \left\lfloor \frac{m}{p} \right\rfloor \right] \cdot t_{\text{on}} + NmT_f \end{aligned}$$

Накладные расходы и обменные операции составляют:

$$T_{\text{обм}} = N \cdot \left\lfloor \frac{ms}{p} \right\rfloor \cdot T_{\text{внутригруп.}} + NsT_{\text{межгруп.}}$$

Межгрупповой обмен может быть осуществлен двумя способами:

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

2) межгрупповая передача по типу «все – всем».

Коэффициент потенциального ускорения равен

$$K_{\text{ном}} = \frac{(2Nms^2 + 2ms + m) \cdot t_{\text{on}} + NsmT_f}{\left[ 2Ns \cdot \left\lfloor \frac{ms}{p} \right\rfloor + (2s+1) \cdot \left\lfloor \frac{m}{p} \right\rfloor \right] \cdot t_{\text{on}} + NmT_f}.$$

Для сложных правых частей:  $T_f \gg t_{\text{on}}$ , коэффициент ускорения  $K_{\text{ном}} = O(s)$ ; если трудоемкость правой части порядка флопа,  $T_f \approx t_{\text{on}}$ :  $K_{\text{ном}} = O(p)$ .

Определение характеристик параллелизма осуществлялось с помощью пакета *Mathematica*® (Wolfram Research Inc.).

## Заключение

Численный эксперимент и проведенный сравнительный анализ динамических характеристик параллельного алгоритма на основе неявных методов Рунге – Кутты показал, что достаточно большая по сравнению с явными методами вычислительная сложность этих методов не является препятствием для использования их в



высокопроизводительных мультипроцессорных системах. Однако требуется тщательный учет всех составляющих параллельной системы для того, чтобы такое решение было масштабируемым и эффективным. В частности, наилучшие характеристики распараллеливания – практически линейное ускорение и близкая к единичной эффективность – достигаются только при использовании синхронной передачи данных в топологии «гиперкуб».

Перспективным направлением дальнейших исследований является разработка и исследование особенностей распараллеливания диагонально неявных и однократно диагонально неявных методов Рунге – Кутты, а также исследование влияния альтернативных механизмов определения локальной погрешности на динамические характеристики полученных параллельных алгоритмов.

## Литература

1. Rajkumar Buaya. High Performance Cluster Computing. – Prentice Hall PTR, Prentice-Hall Inc., 1999.
2. Houwen P.J., Sommeijer B.P. Parallel ODE solver // Proc. International Conf. on Supercomputing. – ACM Press. – 2001. – P. 71-81.
3. Jackson K.R., Norsett S.P. The potential for parallelism in Runge-Kutta methods. Part 1: R-K formulas in standard form // SIAM J. Numer. Anal. – 1999. – Vol. 32. – P. 49-82
4. Houwen P.J., Sommeijer B.P. CWI Contribution to the development of parallel Runge-Kutta methods: Preprint NM-R9506. – CWI. – Amsterdam. – 2003. – 125 p.
5. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. – М.: Мир, 1999. – 685 с.
6. Foster I. Designing and Building Parallel Programs. – Addison-Wesley, 1999. – 302 с.
7. Немюгин С.А., Стесик Р.Г. Параллельное программирование для многопроцессорных ВС. – СПб.: БХВ-Петербург, 2002. – 400 с
8. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных машин. – Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2000. – 176 с.
9. 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.

*І.А. Назарова*

### **Паралельні повністю неявні методи чисельного розв'язання жорстких задач для СЗДР**

Запропоновані паралельні чисельні алгоритми неявних методів Рунге – Кутта для розв'язання жорстких задач Коші. Розроблені обчислювальні схеми відображення методів на паралельні структури різної топології: ланцюг/кільце, ґратка/тор, гіперкуб. Отримані порівняльні характеристики потенційного та реального паралелізму, проведені чисельні експерименти на системі тестів.

*I.A. Nazarova*

### **Parallel Implicit Methods of Numerical Solution Stiff Problems for SODE**

There are realized parallel method of numerical solution implicit methods Runge-Kutta for stiff Cauchy's problem. The developed calculable schemes of methods reflection on parallel structures, with different topologies: ring, matrix/tore, hypercube. The got comparative descriptions potential and real parallelism, the conducted numeral experiments on the system of tests.

*Статья поступила в редакцию 01.07.2005.*