

## ПАРАЛЛЕЛЬНЫЙ (3,2)-МЕТОД РАСЧЕТА ЭЛЕКТРИЧЕСКИХ ЦЕПЕЙ

Новиков Е.А.<sup>1</sup>, Ващенко Г.В.<sup>2</sup>

<sup>1</sup>ФГБУН «Институт вычислительного моделирования СО РАН», Красноярск, Россия (660036, г. Красноярск, Академгородок, ИВМ СО РАН), e-mail: [novikov@icm.krasn.ru](mailto:novikov@icm.krasn.ru)

<sup>2</sup>ФГБОУ ВПО «Сибирский государственный технологический университет», Красноярск, Россия (660049, г. Красноярск, пр. Мира, 82), e-mail: [algo\\_v@mail.ru](mailto:algo_v@mail.ru)

---

Разработан  $L$ -устойчивый (3,2)-метод третьего порядка точности, который предлагается для расчета электрических цепей и сетей. Этот метод входит в семейство  $(m, k)$ -методов, введенных в работе [5]. Эти методы просты в реализации и обладают хорошими свойствами по точности. Эффективность предлагаемого метода может быть повышена за счет использования алгоритмов с контролем точности вычислений на шаге. Для реализации пошагового контроля нами построен критерий, в основе которого лежит оценка аналога глобальной ошибки. Эта оценка осуществляется с привлечением ранее вычисленных стадий численной схемы, что обеспечивает возможность выбора величины шага интегрирования фактически без увеличения вычислительных затрат. Сформулированы и представлены последовательный и параллельный алгоритмы (3,2)-метода с контролем точности вычислений на шаге.

Ключевые слова:  $(m,k)$ -метод, контроль точности, переменный шаг, параллельные вычисления, электрические цепи.

## PARALLEL (3,2)-METHOD FOR ELECTRICAL CIRCUITS

Novikov E.A.<sup>1</sup>, Vashchenko G.V.<sup>2</sup>

<sup>1</sup>FGBUN «Institute of computational modeling SB RAS», Krasnoyarsk, Russia (660036, Krasnoyarsk, Akademgorodok, ICM SB RAS), e-mail: [novikov@icm.krasn.ru](mailto:novikov@icm.krasn.ru)

<sup>2</sup>FGBOU VPO «Siberian State Technological University», Krasnoyarsk, Russia (660049, Krasnoyarsk, av. Mira, 82), e-mail: [algo\\_v@mail.ru](mailto:algo_v@mail.ru)

---

A new class of one-step numerical schemes, which were called  $(m, k)$ -methods have been presented in [5]. They are simple in implementation and have good properties of accuracy. Our goal in this paper is to present  $L$ -stable (3,2)-method of the third order for stiff ODEs. This method we propose for the calculation of electric circuits and networks. The efficiency of the  $L$ -stable (3,2)-method can be increased by means of an algorithm with stepsize control. We have obtained an accuracy criterion for the stepsize control. This criterion is based on an estimate of analog of the global error. It is shown that when a variable step size of integration is used the estimation is carried out with involvement of the previously computed stages allowing you to choose the step of integrating with no increase in computational cost. The primary objective of this work is to illustrate a definite potential for the parallel performance. Serial and parallel algorithm of  $L$ -stable (3,2)-method with stepsize control are formulated.

Keywords:  $(m,k)$ -method, accuracy control, variable step, parallel computing, electrical circuits.

### Введение

Для численного решения жестких систем обыкновенных дифференциальных уравнений обычно применяются  $L$ -устойчивые методы [6]. При реализации таких численных схем на каждом шаге несколько раз решается линейная система алгебраических уравнений с применением  $LU$ -разложения матрицы Якоби. Решение осуществляется с выбором главного элемента по строке или столбцу, а иногда и по всей матрице. При большой размерности исходной задачи общие вычислительные затраты фактически полностью определяются временем декомпозиции данной матрицы. Сокращения вычислительных затрат достигают за счет применения одной матрицы на нескольких шагах [3; 4; 6]. Наиболее естественно это осуществляется в итерационных методах, где данная матрица только определяет скорость

сходимости итерационного процесса [6]. Для безытерационных методов типа Розенброка [7] и их модификаций [5; 6] вопрос о замораживании матрицы Якоби более сложный. В таких методах эта матрица включена в численную схему, и поэтому ее аппроксимация приводит к потере порядка точности численной формулы. Максимальный порядок методов типа Розенброка с замораживанием матрицы Якоби равен двум [3]. Безытерационные методы просты с точки зрения реализации и, как следствие, привлекательны для вычислителей. В [5] предложен новый класс одношаговых численных схем, которые были названы  $(m,k)$ -методами. Они столь же просты при реализации, как и методы типа Розенброка, но обладают более хорошими свойствами точности и устойчивости. Более существенно, они достаточно просто реализуются с замораживанием матрицы Якоби.

Здесь разработан  $L$ -устойчивый  $(3,2)$ -метод третьего порядка для решения жестких задач. На основе оценки глобальной ошибки построено неравенство для контроля точности вычислений. Оценка осуществляется через ранее вычисленные стадии. Сформулирован параллельный алгоритм для вычислительных систем кластерной архитектуры.

### 1. Класс $(m, k)$ -методов решения жестких задач

Рассмотрим задачу Коши для системы дифференциальных уравнений вида

$$y' = f(y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_k, \quad (1)$$

где  $y$  и  $f$  – вещественные  $N$ -мерные вектор-функции,  $t$  – независимая переменная. Пусть  $Z$  – множество целых чисел, и заданы числа  $m, k \in Z, k \leq m$ . Обозначим через  $M_m$  множество чисел  $\{i \in Z, 1 \leq i \leq m\}$ , а через  $M_k$  и  $J_i, 1 < i \leq m$ , подмножества из  $M_m$  вида

$$M_k = \{m_i \in M_m \mid 1 = m_1 < m_2 < \dots < m_k \leq m\},$$

$$J_i = \{m_{j-1} \in M_m \mid j > 1, m_j \in M_k, m_j \leq i\}, \quad 1 < i \leq m.$$

Тогда класс  $(m,k)$ -методов записывается следующим образом:

$$y_{n+1} = y_n + \sum_{i=1}^m p_i k_i, \quad D_n = E - ahf'_n,$$

$$D_n k_i = hf(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j) + \sum_{j \in J_i} \alpha_{ij} k_j + hf'_n \sum_{j=1}^{i-1} c_{ij} k_j, \quad i \in M_k, \quad (2)$$

$$D_n k_i = k_{i-1} + \sum_{j \in J_i} \alpha_{ij} k_j + hf'_n \sum_{j=1}^{i-1} c_{ij} k_j, \quad i \in M_m \setminus M_k,$$

где  $k_i, 1 \leq i \leq m$ , – стадии метода,  $a, p_i, \beta_{ij}, \alpha_{ij}$  и  $c_{ij}$  – постоянные коэффициенты,  $h$  – шаг интегрирования,  $E$  – единичная матрица,  $f'_n = \partial f(y_n) / \partial y$  – матрица Якоби системы (1),  $k$  – количество вычислений функции  $f$  на шаге,  $m$  – число стадий. На каждом шаге интегрирования осуществляются одно вычисление матрицы Якоби и одна декомпозиция матрицы  $D_n$ . Так как  $k$  и  $m$  полностью определяют затраты на шаг, а набор чисел  $m_1, \dots, m_k$  из

множества  $M_k$  только распределяет их внутри шага, то методы типа (2) названы  $(m,k)$ -методами. При  $k=m$  и  $\alpha_{ij}=c_{ij}=0$  схемы (2) совпадают с методами типа Розенброка [7].

## 2. Исследование (3, 2)-метода

Рассмотрим численную формулу вида

$$\begin{aligned} y_{n+1} &= y_n + p_1 k_1 + p_2 k_2 + p_3 k_3, \quad D_n = E + ahf'_n, \\ D_n k_1 &= hf(y_n), \quad D_n k_2 = k_1, \quad D_n k_3 = hf(y_n + \beta_{31} k_1 + \beta_{32} k_2) + \alpha_{32} k_2. \end{aligned} \quad (3)$$

Разлагая стадии  $k_i$ ,  $1 \leq i \leq 3$ , в ряды Тейлора и подставляя в первую формулу (3), получим ряд Тейлора для приближенного решения  $y_{n+1}$ . Полагая  $y_n = y(t_n)$  и сравнивая ряды для точного и приближенного решений, получим условия третьего порядка точности схемы (3), то есть:

$$\begin{aligned} p_1 + p_2 + (1 + \alpha_{32}) p_3 &= 1, \quad ap_1 + 2ap_2 + (a + \beta_{31} + \beta_{32} + 3a\alpha_{32}) p_3 = 1/2, \\ a^2 p_1 + 3a^2 p_2 + (a^2 + 2a\beta_{31} + 3a\beta_{32} + 6a^2 \alpha_{32}) p_3 &= 1/6, \quad (\beta_{31} + \beta_{32})^2 p_3 = 1/3. \end{aligned}$$

Положим  $a$ ,  $\beta_{31}$  и  $\beta_{32}$  свободными и исследуем эту систему на совместность. Получим:

$$\begin{aligned} p_1 &= \{\beta^2(18a^2 - 9a + 1) + 2a(\beta_{31} - a)\} / \{6a^2 \beta^2\}, \\ p_2 &= \{\beta^2(-18a^2 + 15a - 2) + 2a(\beta_{32} - \beta_{31})\} / \{6a^2 \beta^2\}, \\ p_3 &= 1 / \{3\beta^2\}, \quad \alpha_{32} = 0.5\{\beta^2(6a^2 - 6a + 1) - 2a\beta_{32}\} / a^2, \end{aligned} \quad (4)$$

где  $\beta = \beta_{31} + \beta_{32}$ . Исследуем устойчивость схемы (3) на линейном скалярном уравнении  $y' = \lambda y$ , где смысл комплексного числа  $\lambda$ ,  $\text{Re}(\lambda) < 0$ , – некоторое собственное число матрицы Якоби задачи (1). Применяя (3) для решения этого уравнения, получим  $y_{n+1} = Q(z)y_n$ , где  $z = h\lambda$ , а функция устойчивости  $Q(z)$  записывается следующим образом:

$$\begin{aligned} Q(z) &= \{a[a(p_1 + p_3) - a^2 - \beta_{31} p_3]z^3 + [3a^2 - 2ap_1 - ap_2 + (\beta_{31} + \beta_{32} - 2a)p_3]z^2 + \\ &+ [p_1 + p_2 - 3a + (1 + \alpha_{32})p_3]z + 1\} / \{1 - az\}^3. \end{aligned}$$

Из вида  $Q(z)$  следует, что для  $L$ -устойчивости схемы (3) необходимо выполнение соотношения  $a^2 - a(p_1 + p_3) + \beta_{31} p_3 = 0$ . Подставляя сюда коэффициенты (4), получим уравнение  $a^3 - 3a^2 + 1.5a - 1/6 = 0$ . Далее, сравнивая представление приближенного и точного решений до членов с  $h^4$  включительно, видим, что слагаемые с элементарными дифференциалами  $f'''f^3$  и  $f''ff^2$  в главном члене локальной ошибки будут отсутствовать, если

$$(\beta_{31} + \beta_{32})^3 p_3 = 1/4, \quad a(\beta_{31} + \beta_{32})(\beta_{31} + 2\beta_{32}) p_3 = 1/24.$$

Теперь отсюда и (4) получим набор коэффициентов

$$\begin{aligned} p_1 &= \{130a^2 - 33a + 6\} / \{54a^2\}, \quad p_2 = \{21a - 54a^2 - 4\} / \{18a^2\}, \quad p_3 = 16/27, \\ \beta_{31} &= \{48a - 3\} / \{32a\}, \quad \beta_{32} = \{3 - 24a\} / \{32a\}, \quad \alpha_{32} = \{54a^2 - 30a + 6\} / \{32a^2\}, \end{aligned}$$

при которых локальная ошибка  $\delta_{n,3}$  схемы (3) имеет вид

$$\delta_{n,3} = \{[1 - 12a + 36a^2 - 24a^3]h^4 f'^3 f + [1 - 4a]h^4 f f'^2\} / 24 + O(h^5),$$

где значение  $a$  определяется из условия  $L$ -устойчивости  $a^3 - 3a^2 + 1.5a - 1/6 = 0$ . Это уравнение имеет три вещественных корня. Согласно [1] требование  $A$ -устойчивости схемы (3) имеет вид  $1/3 \leq a \leq 1/0685790$ , поэтому выбираем корень  $a = 0.435866521508459$ .

В жестких задачах поведение ошибки определяется элементарным дифференциалом  $f^3 f$  [4]. Поэтому при построении оценки аналога глобальной ошибки будем учитывать только первое слагаемое в локальной ошибке. Для контроля точности вычислений и автоматического выбора величины шага интегрирования используем идею вложенных методов. Для этого рассмотрим двухстадийный метод (2) вида

$$y_{n+1,1} = y_n + b_1 k_1 + b_2 k_2, \quad D_n k_1 = hf(y_n), \quad D_n k_2 = k_1, \quad (5)$$

где  $y_n$  вычислено по формуле (3). В численной формуле (5) применяются стадии метода (3), и поэтому она практически не приводит к увеличению вычислительных затрат. Нетрудно видеть, что при коэффициентах  $b_1 = 0.5(4a-1)/a$  и  $b_2 = 0.5(1-2a)/a$  схема (5) имеет второй порядок точности, а ее локальная ошибка имеет вид  $\delta_{n,2} = (6a^2 - 6a + 1)h^3 f^2 f / 6 + O(h^4)$ . Тогда в неравенстве для контроля точности можно применять оценку ошибки  $\varepsilon_n(j_n)$  вида [4]

$$\varepsilon_n(j_n) = c^{-1} D_n^{1-j_n} (y_{n+1} - y_{n+1,1}). \quad (6)$$

где  $c = 4 \cdot |6a^2 - 6a + 1| / |1 - 12a + 36a^2 - 24a^3| \approx 3$ . При  $j_n = 1$  оценка  $\varepsilon_n(j_n)$  будет  $A$ -устойчивой, а при  $j_n = 2$  –  $L$ -устойчивой. Теперь неравенство для контроля точности имеет вид

$$\| D_n^{1-j_n} (y_{n+1} - y_{n+1,1}) \| \leq c\varepsilon, \quad 1 \leq j_n \leq 2, \quad (7)$$

где  $\varepsilon$  – требуемая точность интегрирования, а параметр  $j_n$  выбирается наименьшим, при котором выполняется неравенство (7). Норма  $\|\xi\|$  в (7) вычисляется по формуле

$$\|\xi\| = \max_{1 \leq i \leq N} \{ |\xi_i| / [ |y_n^i| + \nu ] \}.$$

В случае выполнения неравенства  $|y_n^i| < \nu$  по  $i$ -й компоненте решения контролируется абсолютная ошибка  $\nu\varepsilon$ , в противном случае контролируется относительная ошибка  $\varepsilon$ .

### 3. Последовательный и параллельный алгоритмы интегрирования

Запишем схему (3) в покомпонентной форме, имеем

$$y_i^{(n+1)} = y_i^{(n)} + p_1 k_{1,i}^{(n)} + p_2 k_{2,i}^{(n)} + p_3 k_{3,i}^{(n)}, \quad (8)$$

$$\sum_{j=1}^N d_{ij}^{(n)} k_{1,j}^{(n)} = g_i^{(n)}, \quad \sum_{j=1}^N d_{ij}^{(n)} k_{2,j}^{(n)} = k_{1,i}^{(n)}, \quad \sum_{j=1}^N d_{ij}^{(n)} k_{3,j}^{(n)} = w_i^{(n)},$$

где  $1 \leq i \leq N$ , элементы  $k_{1,j}^{(n)}$ ,  $k_{2,j}^{(n)}$  и  $k_{3,j}^{(n)}$  есть компоненты векторов приращений  $K_1^{(n)} = \{k_{1,j}^{(n)}\}$ ,  $K_2^{(n)} = \{k_{2,j}^{(n)}\}$  и  $K_3^{(n)} = \{k_{3,j}^{(n)}\}$ , значения  $d_{ij}^{(n)}$ ,  $g_i^{(n)}$ ,  $\sigma_i^{(n)}$  и  $w_i^{(n)}$  есть элементы матрицы  $D_n$  и векторов  $g^{(n)}$ ,  $\sigma^{(n)}$  и  $w^{(n)}$ ,  $g^{(n)} = \{g_i^{(n)}\} = \{h_n f_i^{(n)}\}$ ,  $\sigma^{(n)} = \{\sigma_i^{(n)}\} = \{y_i^{(n)} + \beta_{31} k_{1,i}^{(n)} + \beta_{32} k_{2,i}^{(n)}\}$ ,  $w^{(n)} = \{w_i^{(n)}\} = \{h_n f_i^{(n)}(\sigma^{(n)}) + \alpha_{32} k_{2,i}^{(n)}\}$ , причем  $d_{ij}^{(n)} = 1 - ah_n \phi_{ij}^{(n)}$  при  $i = j$  и  $d_{ij}^{(n)} = -ah_n \phi_{ij}^{(n)}$  при

$i \neq j$ . Здесь  $\varphi_{ij}^{(n)}$  – элементы матрицы Якоби  $f'_n$  задачи (1), вычисленные на решении  $y^{(n)}$ ,  $f_i^{(n)}$  – элементы вектора правой части  $f(y)$ . Применение  $LU$ -разложения [2] приводит к системам уравнений для нахождения стадий  $k_{1,j}^{(n)}$ ,  $k_{2,j}^{(n)}$  и  $k_{3,j}^{(n)}$ , то есть

$$\begin{aligned} \sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} &= g_i^{(n)}, \quad \sum_{j=1}^N u_{ij}^{(n)} k_{1,j}^{(n)} = z_i^{(n)}, \quad 1 \leq i \leq N, \\ \sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} &= k_{1,i}^{(n)}, \quad \sum_{j=1}^N u_{ij}^{(n)} k_{2,j}^{(n)} = z_i^{(n)}, \quad 1 \leq i \leq N, \\ \sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} &= w_i^{(n)}, \quad \sum_{j=1}^N u_{ij}^{(n)} k_{3,j}^{(n)} = z_i^{(n)}, \quad 1 \leq i \leq N, \end{aligned} \quad (9)$$

где  $u_{ij}^{(n)} = d_{ij}^{(n)} - \sum_{m=1}^{i-1} l_{im}^{(n)} u_{mj}^{(n)}$  при  $i \leq j$  и  $l_{ij}^{(n)} = \{d_{ij}^{(n)} - \sum_{m=1}^{i-1} l_{im}^{(n)} u_{mj}^{(n)}\} / u_{jj}^{(n)}$  при  $i > j$ .

В приводимом ниже алгоритме интегрирования используются обозначения программ  $LU\_Decompos()$  для декомпозиции матрицы  $D_n$  и  $LU\_Solution()$  для вычисления вектора приращения. С применением обозначения элементов обратной матрицы  $D_n^{-1}$  через  $\tilde{d}_{ij}$ , нормы ошибок  $\|\varepsilon_n(1)\|$  и  $\|\varepsilon_n(2)\|$  вычисляются по формулам

$$\|\varepsilon_n(1)\| = \max_{1 \leq i \leq N} \left\{ \left| ((p_1 - b_1)k_{1,i}^{(n)} + (p_2 - b_2)k_{2,i}^{(n)} + p_3 k_{3,i}^{(n)}) \right| / \left( |y_i^{(n)}| + v \right) \right\}, \quad (10)$$

$$\|\varepsilon_n(2)\| = \max_{1 \leq i \leq N} \sum_{j=1}^N \left\{ \tilde{d}_{ij} \left| ((p_1 - b_1)k_{1,j}^{(n)} + (p_2 - b_2)k_{2,j}^{(n)} + p_3 k_{3,j}^{(n)}) \right| / \left( |y_i^{(n)}| + v \right) \right\}. \quad (11)$$

Пусть приближение  $y_n$  к решению  $y(t_n)$  задачи (1) вычислено в точке  $t_n$  с шагом  $h_n$ . Тогда учитывая, что имеет место  $\varepsilon_n(j_n) = O(h_n^3)$ ,  $1 \leq j_n \leq 2$ , последовательный алгоритм интегрирования формулируется следующим образом.

**Шаг 1.** Вычислить матрицу Якоби  $f'_n$ .

**Шаг 2.** Сформировать матрицу  $D_n = E - ah_n f'_n$ .

**Шаг 3.** Выполнить декомпозицию матрицы  $D_n = LU\_Decompos()$ .

**Шаг 4.** Вычислить  $K_1^{(n)} = LU\_Solution()$ .

**Шаг 5.** Вычислить  $K_2^{(n)} = LU\_Solution()$ .

**Шаг 6.** Вычислить  $\sigma^{(n)} = \{\sigma_i^{(n)}\} = y_i^{(n)} + \beta_{31} k_{1,i}^{(n)} + \beta_{32} k_{2,i}^{(n)}$ .

**Шаг 7.** Вычислить  $w^{(n)} = \{w_i^{(n)}\} = \{h_n f_i^{(n)}(\sigma^{(n)}) + \alpha_{32} k_{2,i}^{(n)}\}$ .

**Шаг 8.** Вычислить  $K_3^{(n)} = LU\_Solution()$ .

**Шаг 9.** Вычислить норму ошибки  $\|\varepsilon_n(1)\|$  по формуле (10).

**Шаг 10.** Вычислить величину  $q_1$  по формуле  $q_1^3 \|\varepsilon_n(1)\| = c\varepsilon$ .

**Шаг 11.** Если  $q_1 < 1$ , то есть требуемая точность не достигнута, то вычисляется  $\|\varepsilon_n(2)\|$  по формуле (11). В противном случае  $\varepsilon_n(2)$  полагается равным  $\varepsilon_n(1)$ .

**Шаг 12.** Вычислить значение параметра  $q_2$  по формуле  $q_2^3 \|\varepsilon_n(2)\| = c\varepsilon$ .

**Шаг 13.** Если  $q_2 < 1$ , то  $h_n$  полагается равным  $q_2 h_n$ , и происходит повторное вычисление решения – возврат на шаг 2.

**Шаг 14.** Вычислить приближение к решению в точке  $t_{n+1}$  по формуле (8), то есть

$$y^{(n+1)} = y^{(n)} + p_1 K_1^{(n)} + p_2 K_2^{(n)} + p_3 K_3^{(n)}.$$

**Шаг 15.** Вычислить значение параметра  $h_{n+1}$  по формуле  $h_{n+1} = \min(q_1, q_2) h_n$ .

**Шаг 16.** Выполнить следующий шаг интегрирования.

Для  $LU$ -разложения матрицы  $D_n$  используется алгоритм с замещением, что дает возможность применять один массив на шаге интегрирования. Далее предположим, что размерность  $N$  системы (1) связана с размером  $p$  вычислительной системы соотношением  $N = s \cdot p$ . Для  $D_n$  выберем блочно-строчную схему хранения,  $d_{ij}^{(n)} \in \text{proc}(j)$ ,  $j = 1, 2, \dots, p$ ,  $s_j = (j-1) \cdot s + 1, \dots, j \cdot s$ . В результате на каждом процессоре будет размещено  $s$  строк матрицы  $D_n$  и  $s$  элементов вектора  $y^{(n)}$ . Тогда параллельный аналог (3,2)-метода (3) запишется в виде

$$\begin{aligned} y_{s_j}^{(n+1)} &= y_{s_j}^{(n)} + p_1 k_{1,s_j}^{(n)} + p_2 k_{2,s_j}^{(n)} + p_3 k_{3,s_j}^{(n)}, \\ \sum_{m=1}^N d_{s_j,m}^{(n)} k_{1,m}^{(n)} &= g_{s_j}^{(n)}, \quad \sum_{m=1}^N d_{s_j,m}^{(n)} k_{2,m}^{(n)} = k_{1,s_j}^{(n)}, \quad \sum_{m=1}^N d_{s_j,m}^{(n)} k_{3,m}^{(n)} = w_{s_j}^{(n)}, \end{aligned} \quad (12)$$

где  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$ . Параллельные аналоги (9) имеют вид

$$\begin{aligned} \sum_{m=(j-1)s+1}^{j \cdot s} l_{s_j,m}^{(n)} z_m^{(n)} &= g_{s_j}^{(n)}, \quad \sum_{m=(j-1)s+1}^{j \cdot s} u_{s_j,m}^{(n)} k_{1,m}^{(n)} = z_{s_j}^{(n)}, \\ \sum_{m=(j-1)s+1}^{j \cdot s} l_{s_j,m}^{(n)} z_m^{(n)} &= k_{1,s_j}^{(n)}, \quad \sum_{m=(j-1)s+1}^{j \cdot s} u_{s_j,m}^{(n)} k_{2,m}^{(n)} = z_{s_j}^{(n)}, \\ \sum_{m=(j-1)s+1}^{j \cdot s} l_{s_j,m}^{(n)} z_m^{(n)} &= w_{s_j}^{(n)}, \quad \sum_{m=(j-1)s+1}^{j \cdot s} u_{s_j,m}^{(n)} k_{3,m}^{(n)} = z_{s_j}^{(n)}, \end{aligned} \quad (13)$$

где  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$ . Сформулируем параллельный алгоритм интегрирования.

Для этого введем функции  $Par\_LU\_Decompos()$  и  $Par\_LU\_Solution()$ , реализующие декомпозицию матрицы  $D_n$  и нахождение векторов  $K_1^{(n)}$ ,  $K_2^{(n)}$  и  $K_3^{(n)}$ . Для контроля точности (12) используем процессор  $\text{proc}(1)$ . Пусть известно приближение  $y^{(n)}$  в точке  $t_n$  с шагом  $h_n$ . Тогда для вычисления  $y^{(n+1)}$  в точке  $t_{n+1}$  справедлив параллельный алгоритм, в котором на каждом процессоре  $\text{proc}(j)$  формируется своя  $j$ -я часть  $y_{s_j}^{(n+1)}$  вектора решения.

- Шаг 1.** В каждом  $\mathit{proc}(j)$ ,  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$  : а) выполнить  $\mathit{recv}(y_{s_{1\dots p}}^{(n+1)}, h; 1, \dots, p)$  ; б) вычислить  $f_{s_j}^{(n)}(y^{(n)})$  ; в) вычислить матрицу Якоби.
- Шаг 2.** Сформировать матрицу  $D_n = E - ah_n f'_n$ .
- Шаг 4.** Разложить матрицу  $D_n = LU\_Decompos()$ .
- Шаг 5.** Вычислить  $K_1^{(n)} = LU\_Solution()$ .
- Шаг 6.** Вычислить  $K_2^{(n)} = LU\_Solution()$ .
- Шаг 7.** В каждом  $\mathit{proc}(j)$ ,  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$  : а) выполнить  $\mathit{recv}(y_{s_{1\dots p}}^{(n)}, k_{1,s_{1\dots p}}^{(n)}, k_{2,s_{1\dots p}}^{(n)}; 1, \dots, p)$  ; б) вычислить  $\sigma_{s_j}^{(n)} = \{y_{s_j}^{(n)} + \beta_{31} k_{1,s_j}^{(n)} + \beta_{32} k_{2,s_j}^{(n)}\}$  ; в) выполнить  $\mathit{send}(\sigma_{s_j}^{(n)}; 1, \dots, p)$ .
- Шаг 8.** В каждом  $\mathit{proc}(j)$ ,  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$  : а) выполнить  $\mathit{recv}(\sigma_{s_j}^{(n)}, k_{2,s_j}^{(n)}, 1, \dots, p)$  ; б) сформировать  $\sigma^{(n)} = \{\sigma_{s_j}^{(n)}\}$  ; в) вычислить  $f_{s_j}^{(n)}(\sigma^{(n)})$  ; г) вычислить  $w_{s_j}^{(n)} = \{h_n f_{s_j}^{(n)}(\sigma^{(n)}) + \alpha_{32} k_{2,s_j}^{(n)}\}$ .
- Шаг 9.** Вычислить  $K_3^{(n)} = Par\_LU\_Solution()$ .
- Шаг 10.** В каждом  $\mathit{proc}(j)$ ,  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$  : а) вычислить  $|\mathcal{E}^{(n)}|_{s_j} = |(p_1 - b_1)k_{1,s_j}^{(n)} + (p_2 - b_2)k_{2,s_j}^{(n)} + p_3 k_{3,s_j}^{(n)}| / \{|y_{s_j}^{(n)}| + v\}$  ; б) вычислить  $\|\mathcal{E}^{(n)}(1)\|_j = \max\{|\mathcal{E}_{(j-1) \cdot s + 1}^{(n)}|, |\mathcal{E}_{(j-1) \cdot s + 2}^{(n)}|, \dots, |\mathcal{E}_{j \cdot s}^{(n)}|\}$  ; в) вычислить  $\|\mathcal{E}^{(n)}(2)\|_{s_j} = \max_{(j-1) \cdot s + 1 \leq j \leq j \cdot s} \sum_{i=1}^N |\tilde{d}_{ij} [(p_1 - b_1)k_{1,j}^{(n)} + (p_2 - b_2)k_{2,j}^{(n)} + p_3 k_{3,j}^{(n)}]| / \{|y_{s_j}^{(n)}| + v\}$  ; г) вычислить  $\|\mathcal{E}^{(n)}(2)\|_j = \max\{|\mathcal{E}_{(j-1) \cdot s + 1}^{(n)}|, |\mathcal{E}_{(j-1) \cdot s + 2}^{(n)}|, \dots, |\mathcal{E}_{j \cdot s}^{(n)}|\}$  ; д) выполнить  $\mathit{send}(\|\mathcal{E}^{(n)}(1)\|_j, \|\mathcal{E}^{(n)}(2)\|_j, 1)$ .
- Шаг 11.** В  $\mathit{proc}(1)$ : а) выполнить  $\mathit{recv}(\|\mathcal{E}^{(n)}(1)\|_j, \|\mathcal{E}^{(n)}(2)\|_j, 1 \leq j \leq p)$  ; б) определить  $\|\mathcal{E}^{(n)}(1)\| = \max_{1 \leq j \leq p} \{\|\mathcal{E}^{(n)}(1)\|_j\}$  ; в) определить  $\|\mathcal{E}^{(n)}(2)\| = \max_{1 \leq j \leq p} \{\|\mathcal{E}^{(n)}(2)\|_j\}$  ; г) вычислить  $q_1 = \sqrt[3]{c\mathcal{E} / \|\mathcal{E}^{(n)}(1)\|}$  ; д) если  $q_1 < 1$ , то есть требуемая точность не достигнута, то переход на пункт е), иначе  $\varepsilon_n(2)$  полагается равным  $\varepsilon_n(1)$  и переход на е) ; е) вычислить  $q_2 = \sqrt[3]{c\mathcal{E} / \|\mathcal{E}^{(n)}(2)\|}$  ; ж) если  $q_2 < 1$ , то  $h_n$  полагается равным  $q_2 h_n$  ; з) выполнить  $\mathit{send}(h_n, rp; 1, \dots, p)$ . При  $rp=1$  происходит повторное вычисление решения – возврат на шаг 2, при  $rp=0$  – переход на шаг 12.
- Шаг 12.** В каждом  $\mathit{proc}(j)$ ,  $1 \leq j \leq p$ ,  $(j-1) \cdot s + 1 \leq s_j \leq j \cdot s$  : а) выполнить  $\mathit{recv}(h_n, rp; 1)$  ;

б) вычислить  $y_{s_j}^{(n+1)} = y_{s_j}^{(n)} + p_1 k_{1,s_j}^{(n)} + p_2 k_{2,s_j}^{(n)} + p_3 k_{3,s_j}^{(n)}$ ; б) выполнить *send* ( $y_{s_{1\dots p}}^{(n+1)}; 1, \dots, p$ ).

**Шаг 13.** Выполнить следующий шаг интегрирования.

Необходимость решения жестких задач возникает во многих приложениях. Часто методом прямых уравнения математической физики в частных производных сводятся к жестким системам. Основные тенденции при построении численных методов связаны с расширением их возможностей для решения систем большой размерности. Здесь предложен параллельный алгоритм для вычислительных систем кластерной архитектуры.

### Список литературы

1. Демидов Г.В., Юматова Л.А. Исследование некоторых аппроксимаций в связи с А-устойчивостью полуявных методов // Численные методы механики сплошной среды. – 1977. – Т. 8. – № 3. – С. 68–79.
2. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения. – М. : Мир, 2001. – 429 с.
3. Новиков В.А., Новиков Е.А., Юматова Л.А. Замораживание матрицы Якоби в методе типа Розенброка второго порядка точности // ЖВМ и МФ. – 1987. – Т. 27. – № 3. – С. 385–390.
4. Новиков Е.А., Шорников Ю.В. Компьютерное моделирование жестких гибридных систем. – Новосибирск : Изд-во НГТУ, 2012. – 450 с.
5. Новиков Е.А., Шитов Ю.А., Шокин Ю.И. Одношаговые безытерационные методы решения жестких систем // ДАН СССР. – 1988. – Т. 301. – № 6. – С. 1310–1314.
6. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. – М. : Мир, 1999. – 685 с.
7. Rosenbrock H.H. Some general implicit processes for the numerical solution of differential equations // Computer. – 1963. – № 5. – P. 329–330.

*Работа выполнена при финансовой поддержке РФФИ (проект 11-01-00106).*

### Рецензенты

Богульский Игорь Олегович, д.ф.-м.н., профессор кафедры сопротивления материалов и теоретической механики Института управления инженерными системами Красноярского государственного аграрного университета, г. Красноярск.

Нужин Яков Нифантьевич, д.ф.-м.н., профессор кафедры математического обеспечения дискретных устройств и систем Института фундаментальной подготовки Сибирского федерального университета, г. Красноярск.