УДК 519.622.852

L-УСТОЙЧИВЫЙ МЕТОД ТРЕТЬЕГО ПОРЯДКА ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ ЖЕСТКИХ ЗАДАЧ

[1]Новиков Е.А., [2]Ващенко Г.В.

[1]ФГБУН «Институт вычислительного моделирования СО РАН», Красноярск, e-mail: novikov@icm.krasn.ru;

[2]ФГБОУ ВПО «Сибирский государственный технологический университет», Красноярск, e-mail: algo_v@mail.ru

Представлен L-устойчивый метод третьего порядка точности для численного решения жестких задач. Этот метод достаточно прост в реализации и обладает 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 алгоритм и параллельная версия MPI-алгоритма с контролем точности вычислений на шаге. Данный метод входит в семейство (m, k)-методов, введенных в работе [5]. В (m,k)-методах стадия метода не связывается с обязательным вычислением правой части исходной задачи. За счет этого их свойства могут быть улучшены. Данные схемы можно рассматривать как способ реализации неявных методов типа Рунге – Кутты. Важно, что при такой реализации не требуются итерации метода Ньютона.

Ключевые слова: контроль точности, переменный шаг, (3,2)-метод, параллельный MPI-алгоритм

AN L-STABLE METHOD OF THIRD ORDER FOR THE NUMERICAL INTEGRATION STIFF PROBLEMS

[1]Novikov E.A., [2]Vaschenko G.V.

[1]Institute of computational modeling SB RAS, Krasnoyarsk, e-mail: novikov@icm.krasn.ru;

[2]Siberian State Technological University, Krasnoyarsk, e-mail: algo_v@mail.ru

The aim of this paper is to present L-stable method of the third order for stiff problems. It the efficiency can be increased by means of an algorithm with step size control. This method has good accuracy. We have obtained an accuracy criterion for the step size 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 choosing a integration step with no increase in computational cost. Serial and parallel MPI-algorithms of L-stable method with step size control are formulated. This method is included in the class (m, k)-methods introduced in [5]. (m, k)-methods can be seen as a way to implement implicit Runge-Kutta methods. It is important that when such implementation is not required iteration of Newton's method.

Keywords: accuracy control, variable step, (3,2)-method, parallel MPI-algorithm

Для численного решения жестких систем обыкновенных дифференциальных уравнений обычно применяются L-устойчивые методы [9]. При реализации таких численных схем на каждом шаге несколько раз решается линейная система алгебраических уравнений с применением LU-разложения матрицы Якоби. При большой размерности исходной задачи общие вычислительные затраты фактически полностью определяются временем декомпозиции данной матрицы. Сокращения затрат достигают за счет применения одной матрицы на нескольких шагах [3, 4, 9]. Наиболее естественно это осуществляется в итерационных методах, где данная матрица только определяет скорость сходимости итерационного процесса [9]. Для безитерационных методов типа Розенброка [10] и их модификаций [5, 9] 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. Некоторым аналогом замораживания матрицы Якоби является применение в расчетах алгоритмов интегрирования на основе явных и L-устойчивых методов с автоматическим выбором численной схемы. В этом случае эффективность алгоритма может быть повышена за счет расчета переходного участка, соответствующего максимальному собственному числу матрицы Якоби, явным методом. В качестве критерия выбора эффективной численной формулы естественно применять неравенство для контроля устойчивости [6, 11]. Следует отметить, что применение таких комбинированных алгоритмов полностью не снимает проблему замораживания матрицы Якоби, потому что явным методом можно просчитать, вообще говоря, погранслойное решение, соответствующее максимальному собственному числу матрицы Якоби. 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 точности для решения жестких задач. Построено неравенство для контроля точности вычислений, основанное на оценке аналога глобальной ошибки. Оценка осуществляется с привлечением ранее вычисленных стадий, что позволяет выбирать величину шага интегрирования фактически без увеличения вычислительных затрат. Сформулирован последовательный алгоритм и его параллельный аналог – MPI-алгоритм.

1. Класс $(\boldsymbol{m}, \boldsymbol{k})$-методов решения жестких задач

Рассмотрим задачу Коши для системы дифференциальных уравнений вида

$$y' = f(y);\ y(t_0) = y_0;\ t_0 \le t \le t_k, \tag{1}$$

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

$$M_k = \left\{ m_i \in M_m \,\middle|\, 1 = m_1 < m_1 < \ldots < m_k \le m \right\};$$

$$J_i = \left\{ m_{j-1} \in M_m \,\middle|\, j > 1, m_j \in M_k, m_j \le i \right\},$$

где $1 \le i \le m$. Тогда класс (m, k)-методов записывается следующим образом [5]:

$$\begin{gathered} y_{n+1} = y_n + \sum\nolimits_{i=1}^{m} p_i k_i;\quad D_n = E - ahf_n', \\ D_n k_i = hf_n\left(y_n + \sum\nolimits_{j=1}^{i-1} \beta_{ij} k_j\right) + \sum\nolimits_{j\in J_i} \alpha_{ij} k_j + hf_n' \sum\nolimits_{j=1}^{i-1} c_{ij} k_j,\quad i \in M_k, \\ D_n k_i = k_{i-1} + \sum\nolimits_{j\in J_i} \alpha_{ij} k_j + hf_n' \sum\nolimits_{j=1}^{i-1} c_{ij} k_j,\ i \in M_m \backslash M_k, \end{gathered} \tag{2}$$

где k_i, $1 \le i \le m$, – стадии метода; a, p_i, β_{ij}, α_{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 $m_1, \ldots, m_k$ из множества M_k только распределяет их внутри шага, то методы типа (2) названы (m, k)-методами. Основное отличие приведенных методов (2) от существующих численных формул состоит в том, что в них стадия метода не связывается с обязательным вычислением правой части исходной задачи (1), за счет этого свойства методов могут быть улучшены.

При $k = m$ и $\alpha_{ij} = c_{ij} = 0$ схемы (2) совпадают с методами типа Розенброка [10], а при $k = m$ и $\alpha_{ij} = 0$ – с ROW-методами [9]. В отличие от ROW-методов в численных формулах (2) более точно определены затраты на шаг интегрирования и более правильно описана область определения коэффициентов численных формул, что упрощает их исследование и делает их предпочтительнее. При рассмотрении методов такого типа в основном изучался случай $k = m$, то есть когда число стадий и количество вычислений правой части задачи (1) совпадают. В этом случае k-стадийную схему (2) можно поставить в соответствие k-стадийной полуявной формуле типа Рунге – Кутты, при реализации которой на каждом шаге используется одна матрица размерности N. Относительно таких численных формул известно, что нельзя построить k-стадийную схему выше $(k + 1)$-го порядка точности, причем схема максимального порядка A-устойчивая. Если рассматривать схемы (2) при $m > k$, то можно построить численные схемы более высокого порядка точности [5].

2. Исследование (3, 2)-метода

Рассмотрим численную формулу вида

$$\begin{gathered} y_{n+1} = y_n + p_1 k_1 + p_2 k_2 + p_3 k_3; \\ D_n = E - ahf_n';\ D_n k_1 = hf(y_n);\quad D_n k_2 = k_1; \\ D_n k_3 = hf(y_n + \beta_{31} k_1 + \beta_{32} k_2) + \alpha_{32} k_2. \end{gathered} \tag{3}$$

Разлагая стадии k_i, $1 \le i \le 3$ 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

$$p_1 + p_2 + (1+\alpha_{32})p_3 = 1; \quad ap_1 + 2ap_2 + (a + \beta_{31} + \beta_{32} + 3a\alpha_{32})p_3 = \frac{1}{2};$$

$$a^2 p_1 + 3a^2 p_2 + (a^2 + 2a\beta_{31} + 3a\beta_{32} + 6a^2\alpha_{32})p_3 = \frac{1}{6}; (\beta_{31} + \beta_{32})^2 p_3 = \frac{1}{3}.$$

Положим a, β_{31} и β_{32} свободными и исследуем эту систему на совместность. Получим

$$\begin{aligned} p_1 &= \{\beta^2(18a^2 - 9a + 1) + 2a(\beta_{31} - a)\} / \{6a^2\beta^2\}; \\ p_2 &= \{\beta^2(-18a^2 + 15a - 2) + 2a(\beta_{32} - \beta_{31})\} / \{6a^2\beta^2\}; \\ p_3 &= 1/\{3\beta^2\}; \quad \alpha_{32} = 0{,}5\{\beta^2(6a^2 - 6a + 1) - 2a\beta_{32}\} / a^2. \end{aligned} \tag{4}$$

где $\beta = \beta_{31} + \beta_{32}$. Исследуем устойчивость схемы (3) на линейном скалярном уравнении $y' = \lambda y$, где смысл комплексного числа λ, $\mathrm{Re}(\lambda) < 0$, – 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:

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

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 $f''f'f^2$ в главном члене локальной ошибки будут отсутствовать, если

$$(\beta_{31} + \beta_{32})^3 p_3 = \frac{1}{4};$$

$$a(\beta_{31} + \beta_{32})(\beta_{31} + 2\beta_{32})p_3 = \frac{1}{24}.$$

Теперь отсюда и (4) получим набор коэффициентов

$$p_1 = \{130a^2 - 33a + 6\} \big/ \{54a^2\};$$

$$p_2 = \{21a - 54a^2 - 4\} \big/ \{18a^2\};$$

$$p_3 = \frac{16}{27};$$

$$\beta_{31} = \{48a - 3\} / \{32a\};$$

$$\beta_{32} = \{3 - 24a\} / \{32a\};$$

$$\alpha_{32} = \{54a^2 - 30a + 6\} \big/ \{32a^2\},$$

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

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

где значение a определяется из условия L-устойчивости

$$a^3 - 3a^2 + 1{,}5a - 1/6 = 0.$$

Это уравнение имеет три вещественных корня. Согласно [2] требование A-устойчивости схемы (3) имеет вид $1/3 \le a \le 1{,}0685790$, поэтому выбираем корень $a = 0{,}435866521508459$.

В жестких задачах поведение ошибки определяется элементарным дифференциалом $f'^3 f$ [5]. 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

$$y_{n+1,1} = y_n + b_1 k_1 + b_2 k_2;$$

$$D_n k_1 = hf(y_n);\ D_n k_2 = k_1, \qquad (5)$$

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 [5]

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

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

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

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

$$\| \xi \| = \max_{1 \le i \le N} \left\{ |\xi_i| \big/ \left[|y_n^i| + v \right] \right\}.$$

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

3. Последовательный алгоритм интегрирования и его параллельная версия

Запишем схему (3) в покомпонентной форме, имеем

$$y_i^{(n+1)} = y_i^{(n)} + p_1 k_{1,i}^{(n)} + p_2 k_{2,i}^{(n)} + p_3 k_{3,i}^{(n)};$$
$$\sum_{j=1}^N d_{ij}^{(n)} k_{1,j}^{(n)} = g_i^{(n)}; \qquad (8)$$
$$\sum_{j=1}^N d_{ij}^{(n)} k_{2,j}^{(n)} = k_{1,i}^{(n)};\ \sum_{j=1}^N d_{ij}^{(n)} k_{3,j}^{(n)} = w_i^{(n)},$$

где $1 \le i \le N$, $k_{1,j}^{(n)}$, $k_{2,j}^{(n)}$ и $k_{3,j}^{(n)}$ есть элементы векторов приращений $K_1^{(n)} = \{k_{1,j}^{(n)}\}$; $K_2^{(n)} = \{k_{2,j}^{(n)}\}$ и $K_3^{(n)} = \{k_{3,j}^{(n)}\}$; $d_{ij}^{(n)}$, $g_i^{(n)}$, $\sigma_i^{(n)}$ и $w_i^{(n)}$ есть элементы матрицы D_n и векторов $g^{(n)}$, $\sigma^{(n)}$, $w^{(n)}$;

$$g_i^{(n)} = \{h_n f_i^{(n)}\};$$

$$\sigma_i^{(n)} = \{y_i^{(n)} + \beta_{31} k_{1,i}^{(n)} + \beta_{32} k_{2,i}^{(n)}\};$$

$$w_i^{(n)} = \left\{ h_n f_i^{(n)}\left(\sigma^{(n)}\right) + \alpha_{32} k_{2,i}^{(n)} \right\};$$
$$d_{ij}^{(n)} = 1 - a h_n \varphi_{ij}^{(n)}$$

при $i = j$ и $d_{ij}^{(n)} = -a h_n \varphi_{ij}^{(n)}$ при $i \ne j$. 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 [8] приводит к системам уравнений для нахождения стадий $k_{1,j}^{(n)}$, $k_{2,j}^{(n)}$ и $k_{3,j}^{(n)}$, то есть

$$\sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} = g_i^{(n)};\quad \sum_{j=1}^N u_{ij}^{(n)} k_{1,j}^{(n)} = z_i^{(n)},\ 1 \le i \le N;$$
$$\sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} = k_{1,i}^{(n)};\quad \sum_{j=1}^N u_{ij}^{(n)} k_{2,j}^{(n)} = z_i^{(n)},\ 1 \le i \le N; \qquad (9)$$
$$\sum_{j=1}^N l_{ij}^{(n)} z_j^{(n)} = w_i^{(n)};\quad \sum_{j=1}^N u_{ij}^{(n)} k_{3,j}^{(n)} = z_i^{(n)},\ 1 \le i \le N,$$

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

Используя обозначения элементов обратной матрицы D_n^{-1} через $\tilde{d}_{ij}$, нормы ошибок вычисляем по формулам

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

$$\left\|\varepsilon_n(2)\right\| = \max_{1\le i\le N}\left\{\left|\sum\nolimits_{j=1}^{N}\tilde{d}_{ij}\left[(p_1-b_1)k_{1,j}^{(n)}+(p_2-b_2)k_{2,j}^{(n)}+p_3k_{3,j}^{(n)}\right]\right|\Big/\left[\left|y_i^{(n)}\right|+v\right]\right\}. \quad (11)$$

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 $\varepsilon_n(j_n)=O\left(h_n^3\right)$, $1\le j_n\le 2$, последовательный алгоритм интегрирования формулируется следующим образом.

Шаг 1, 2. Вычислить матрицу Якоби f_n' и сформировать матрицу $D_n = E - ah_nf_n'$.

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

Шаг 4, 5. Вычислить

$$K_1^{(n)} = LU\_Solution(),$$

$$K_2^{(n)} = LU\_Solution().$$

Шаг 6, 7. Вычислить

$$\sigma^{(n)} = \{\sigma_i^{(n)}\} = y_i^{(n)} + \beta_{31}k_{1,i}^{(n)} + \beta_{32}k_{2,i}^{(n)};$$

$$w^{(n)} = \left\{w_i^{(n)}\right\} = \left\{h_nf_i^{(n)}\left(\sigma^{(n)}\right)+\alpha_{32}k_{2,i}^{(n)}\right\}.$$

Шаг 8. Вычислить

$$K_3^{(n)} = LU\_Solution().$$

Шаг 9, 10. Вычислить норму ошибки $\left\|\varepsilon_n(1)\right\|$ по формуле (10) и q_1 по формуле

$$q_1^3\left\|\varepsilon_n(1)\right\| = c\varepsilon.$$

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

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

Шаг 13. Если $q_2 < 1$, то h_n полагается равным q_2h_n и возврат на шаг 2.

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

$$y^{(n+1)} = y^{(n)} + p_1K_1^{(n)} + p_2K_2^{(n)} + p_3K_3^{(n)}.$$

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

Шаг 16. Выполнить следующий шаг интегрирования.

Замечание

При численном вычислении матрицы Якоби шаг численного дифференцирования r_i, $1 \le i \le N$, выбирается по формуле [6]

$$r_j = \max\left\{r_{\min}, r_{\min}^{0,5}\,|\,y_n^j\,|\right\},\ 1\le j\le N,$$

где r_{min} – минимальный шаг численного дифференцирования, зависит от разрядности вычислительной системы. При расчетах с двойной точностью величину r_{min} следует принять равной 10^{-14}. Тогда j-й столбец a_n^j численной матрицы вычисляется по формуле

$$a_n^j = \left[f\left(y_1,\cdots,y_j+r_j,\cdots,y_N\right)-f\left(y_1,\cdots,y_j,\cdots,y_N\right)\right]\Big/r_j,$$

то есть для задания матрицы требуется N вычислений правой части системы дифференциальных уравнений (1).

Если рассматривать алгоритм (3, 2)-метода (3), (8) как объект для распараллеливания, то его последовательный вариант выглядит как процесс вычисления векторов приращений $K_i^{(n)}$, $1 \le i \le 3$. При этом на каждом n-м шаге вычисления имеют последовательный порядок, $K_1^{(n)} \to K_2^{(n)} \to K_3^{(n)}$. При построении параллельного алгоритма необходимо сохранить этот порядок вычисления. Элементы каждого из векторов приращений получаются из решения систем линейных уравнений с одинаковой матрицей D_n и разными правыми частями.

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. Тогда параллельный аналог (3,2)-метода (3) запишется в виде [7]

$$y_{s_j}^{(n+1)} = y_{s_j}^{(n)} + p_1k_{1,s_j}^{(n)} + p_2k_{2,s_j}^{(n)} + p_3k_{3,s_j}^{(n)};$$

$$\sum\nolimits_{m=1}^{N} d_{s_j,m}^{(n)}k_{1,m}^{(n)} = g_{s_j}^{(n)}; \quad (12)$$

$$\sum\nolimits_{m=1}^{N} d_{s_j,m}^{(n)}k_{2,m}^{(n)} = k_{1,s_j}^{(n)};\quad \sum\nolimits_{m=1}^{N} d_{s_j,m}^{(n)}k_{3,m}^{(n)} = w_{s_j}^{(n)},$$

где $1\le i\le p$, $(j-1)\cdot s+1\le s_j\le j\cdot s$.

Теперь сформулируем параллельный алгоритм вычисления приближенного решения с контролем точности вычислений. Для контроля точности в (12) используем процессор $pr(1)$. 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 pr(*j*) формируется своя s_j-я часть матрицы D_n, векторов $k^{(n)}_{1,s_j}, k^{(n)}_{2,s_j}, k^{(n)}_{3,s_j}$ и вектора решения $y^{(n+1)}_{s_j}$.

Шаг 1. В каждом *pr*(*j*), $1 \le i \le p$, $(j-1)\cdot s+1 \le s_j \le j\cdot s$:

а) выполнить $recv\left(y^{(n+1)}_{s_{1,\ldots,p}}, h; 1,\ldots, p\right)$;

б) вычислить $f^{(n)}_{s_j}\left(y^{(n)}\right)$;

в) вычислить матрицу Якоби.

Шаг 2, 3. Сформировать матрицу $D_n = E - ah_n f'_n$ и разложить

$$D_n = ParLU\_Decompos().$$

Шаг 4, 5. Вычислить

$$K^{(n)}_1 = ParLU\_Solution()$$

$$K^{(n)}_2 = ParLU\_Solution().$$

Шаг 6. В каждом *pr*(*j*), $1 \le i \le p$, $(j-1)\cdot s+1 \le s_j \le j\cdot s$:

а) выполнить

$$recv\left(y^{(n)}_{s_{1,\ldots,p}}, k^{(n)}_{1,s_{1,\ldots,p}}, k^{(n)}_{2,s_{1,\ldots,p}}; 1,\ldots, p\right);$$

б) вычислить

$$\sigma^{(n)}_{s_j} = \left\{y^{(n)}_{s_j} + \beta_{31}k^{(n)}_{1,s_j} + \beta_{32}k^{(n)}_{2,s_j}\right\};$$

в) выполнить $send\left(\sigma^{(n)}_{s_j}; 1,\ldots, p\right)$.

Шаг 7. В каждом *pr*(*j*), $1 \le j \le p$, $(j-1)\cdot s+1 \le s_j \le j\cdot s$:

а) выполнить $recv\left(\sigma^{(n)}_{s_j}, k^{(n)}_{2,s_j}, 1,\ldots, p\right)$;

б) сформировать $\sigma^{(n)} = \left\{\sigma^{(n)}_{s_j}\right\}$;

в) вычислить $f^{(n)}_{s_j}\left(\sigma^{(n)}\right)$ и $w^{(n)}_{s_j} = \left\{h_n f^{(n)}_{s_j}\left(\sigma^{(n)}\right) + \alpha_{32}k^{(n)}_{2,s_j}\right\}$.

Шаг 8. Вычислить

$$K^{(n)}_3 = ParLU\_Solution().$$

Шаг 9. В каждом *pr*(*j*), $1 \le j \le p$, $(j-1)\cdot s+1 \le s_j \le j\cdot s$:

а) вычислить

$$\left|\varepsilon^{(n)}\right|_{s_j} = \left|(p_1-b_1)k^{(n)}_{1,s_j} + (p_2-b_2)k^{(n)}_{2,s_j} + p_3k^{(n)}_{3,s_j}\right| \Big/ \left[\left|y^{(n)}_{s_j}\right| + v\right];$$

б) вычислить

$$\left\|\varepsilon^{(n)}(1)\right\|_j = \max\left\{\left|\varepsilon^{(n)}_{(j-1)\cdot s+1}\right|, \left|\varepsilon^{(n)}_{(j-1)\cdot s+2}\right|, \ldots, \left|\varepsilon^{(n)}_{j\cdot s}\right|\right\};$$

в) вычислить

$$\|\varepsilon^{(n)}(2)\|_{s_j} = \max_{(j-1)\cdot s+1\le s_j\le j\cdot s}\left\{\left|\sum\nolimits_{j=1}^{N}\tilde{d}_{ij}\left[(p_1-b_1)k^{(n)}_{1,j} + (p_2-b_2)k^{(n)}_{2,j} + p_3k^{(n)}_{3,j}\right]\right| \Big/ \left[|\,y^{(n)}_{s_j}\,| + v\right]\right\};$$

г) вычислить

$$\left\|\varepsilon^{(n)}(2)\right\|_j = \max\left\{\left|\varepsilon^{(n)}_{(j-1)\cdot s+1}\right|, \left|\varepsilon^{(n)}_{(j-1)\cdot s+2}\right|, \ldots, \left|\varepsilon^{(n)}_{j\cdot s}\right|\right\};$$

д) send $\left(\left\|\varepsilon^{(n)}(1)\right\|_j\ \left\|\varepsilon^{(n)}(2)\right\|_j, 1\right)$.

Шаг 10. В *pr*(1):

а) выполнить

$$recv\left(\left\|\varepsilon^{(n)}(1)\right\|_j, \left\|\varepsilon^{(n)}(2)\right\|_j, 1 \le j \le p\right);$$

выполняется последовательность действий по контролю точности и вычисление значения h_n;

б) выполнить send (h_n, *rp*; 1, …, *p*). При *rp* = 1 – возврат на шаг 2, при *rp* = 0 – переход на шаг 11.

Шаг 11. В каждом *pr*(*j*), $1 \le j \le p$, $(j-1)\cdot s+1 \le s_j \le j\cdot s$:

а) выполнить recv (h_n, *rp*; 1);

б) вычислить $y^{(n+1)}_{s_j} = y^{(n)}_{s_j} + p_1k^{(n)}_{1,s_j} + p_2k^{(n)}_{2,s_j} + p_3k^{(n)}_{3,s_j}$;

в) выполнить send $\left(y^{(n+1)}_{s_{1,\ldots,p}}; 1,\ldots, p\right)$.

Шаг 12. В $pr(1)$:

а) вычислить h_{n+1} по формуле

$$h_{n+1} = \min(q_1, q_2)h_n;$$

б) send (h_{n+1}; 1, …, p).

Шаг 13. Выполнить следующий шаг интегрирования.

Замечание

Представленный алгоритм является параллельно-последовательным. В нем не учтены фрагменты, относящиеся к вычислению правой части (1) и матрицы Якоби, а также обращение матрицы Якоби, необходимое при невыполнении неравенства по точности. Для LU-разложения используется частичный выбор по столбцу.

В (m, k)-методах стадия метода не связывается с обязательным вычислением правой части исходной задачи. За счет этого их свойства могут быть улучшены. Данные схемы можно рассматривать как способ реализации неявных методов типа Рунге – Кутты. Важно, что при такой реализации не требуются итерации метода Ньютона, а все проблемы решаются за счет выбора шага интегрирования. Построена параллельная MPI-версия алгоритма интегрирования переменного шага, ориентированная на кластерные системы и топологию полного графа. В [1] построено соотношение изоэффективности, которое может быть использовано для сравнения различных параллельных алгоритмов решения одной и той же задачи Коши на основе (3,2)-метода, а также подходов к выбору и построению алгоритмов вычисления матрицы Якоби и алгоритмов ее факторизации. В особенности это относится к оценке коммуникационных затрат при организации межпроцессорных обменов.

Работа выполнена при финансовой поддержке РФФИ (проект 14-01-00047).

Список литературы

1. Ващенко Г.В. Построение соотношения изоэффективности параллельного (3,2)-метода для численного решения задачи Коши // Современное состояние естественных и технических наук: мат. X Международной конф. – М.: Изд-во «Спутник+», 2013. – С. 36–41.

2. 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.

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. 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.

7. Новиков Е.А. Явные методы для жестких систем. – Новосибирск: Наука, 1997. – 197 с.

8. Новиков Е.А. Ващенко Г.В. Design of Parallel Algorithm (3,2)-method for Stiff Problems //Second International Conference on Cluster Computing (CC 2013), Lviv, Ukraine, June 3–5, Proceedings, p.165, 2013.

9. Demmel J. Applied numerical linear algebra. SIAM, 1997.

10. Hairer E. and Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems. Springer-Verlag, second revised edition, 1996.

11. Rosenbrock H.H. Some general implicit processes for the numerical solution of differential equations // Computer. – 1963. – № 5. – P. 329–330.

12. Shampine L.M. Implementation of Rosenbrock methods // ACM Transaction on Mathematical Software. – 1982. – Vol. 8, № 5. – P. 93–113.

References

1. Vashchenko G.V. Postroenie sootnoshenia izoeffektivnosti parallelnogo (3,2)metoda dlya chislennogo resheniya zadachi Koshy Mat.X Mezdunarodnoi konferencii Sovremennoe sostoyanie estestvennyh I tehnicheskih nauk, Mart, Moskva, Sputnik+, 2013, pp. 36–41.

2. Demidov G.V., Yumatova L.A. Issledovanie nekotoryh approksimaciy v svayzi s A-ustoichivostuy poluyavnyh metodov Numerical methods of continuum mechanics, 1977, Vol. 8, no.3, pp. 68–79.

3. Novikov V.A., Novikov E.A., Yumatova L.A. Zamorajivanie matrici Jacobi v metode Rosenbrock vtorogo poryadka tochnosti GVMiMF.1987, Vol. 27, no. 3, p. 385–390.

4. Novikov E.A., Shornikov Yu.V. Komputernoe modelirovanie gibridnyh system. Novosibirsk, NSTU Publishers, 2012.

5. Novikov E.A., Shitov Yu.A., Shokin Yu.I. Odnoshagovye beziteracinnye metody resheniya gestkih zadach DAN USSR, 1988, Vol. 301, no. 6, pp. 1310–1314.

6. Demidov G.V., Jumatova L.A. Issledovanie nekotoryh approksimacij v svjazi s A–ustojchivost'ju polujavnyh metodov // Chislennye metody mehaniki sploshnoj sredy. 1977. T.8. no. 3. pp. 68–79.

7. Novikov E.A. Javnye metody dlya gestkih system. Novosibirsk,Nauka,1997.

8. Novikov E.A., Vashchenko G.V. Design of Parallel Algorithm (3,2)-method for Stiff_Problems // Second International Conference on Cluster Computing (CC 2013), Lviv, Ukraine, June 3–5, Proceedings, pp. 165, 2013.

9. Demmel J. Applied numerical linear algebra. SIAM, 1997.

10. Hairer E. and Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems. Springer-Verlag, second revised edition, 1996.

11. Rosenbrock H.H. Some general implicit processes for the numerical solution of differential equations // Computer,1963, no. 5, pp. 329–330.

12. Shampine L.M. Implementation of Rosenbrock methods // ACM Transaction on Mathematical Software, 1982, Vol. 8, no. 5, pp. 93–113.

Рецензенты:

Белолипецкий В.М., д.ф.-м.н., профессор, главный научный сотрудник, ФГБУН «Институт вычислительного моделирования» СО РАН, г. Красноярск;

Плотников С.М., д.т.н., профессор, ФГБОУ ВПО «Сибирский государственный технологический университет», г. Красноярск.

Работа поступила в редакцию 10.07.2014.