УДК 519. 622. 852

ПОСЛЕДОВАТЕЛЬНЫЙ И ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМЫ ПЕРЕМЕННОГО ШАГА НА ОСНОВЕ (2,2)-МЕТОДА

Новиков Е.А., Ващенко Г.В.

Институт вычислительного моделирования СО РАН, Красноярск, Россия, e-mail: novikov@icm.krasn.ru
Сибирский государственный технологический университет, Красноярск, Россия, e-mail: algo_v@mail.ru

Представлены последовательный и параллельный алгоритмы *L*-устойчивого (2,2)-метода переменного шага. Изменение величины шага построено на основе контроля точности численной схемы.

Ключевые слова: (2,2)-метод, последовательный алгоритм, параллельный алгоритм, контроль точности.

SERIAL AND PARALLEL ALGORITHM BASED ON VARIABLE STEP (2,2)−METHOD

Novikov E.A., Vashchenko G.V.

Institute of computational modeling SB RAS, Krasnoyarsk, Russia, e-mail: novikov@icm.krasn.ru
Siberian State Technological University, Krasnoyarsk, Russia, e-mail: algo_v@mail.ru

The serial and parallel algorithm of *L*-stable (2,2)-method with variable step are presented. Adjust step size is based on monitoring of the accuracy of the numerical scheme.

Keywords: (2,2)-method, serial algorithm, parallel algorithm, accuracy control.

Введение. Основные тенденции при построении эффективных численных методов решения задачи Коши для систем обыкновенных дифференциальных уравнений связаны с проблемами жесткости и большой размерности практических задач. Для решения жестких задач обычно применяются L-устойчивые методы [1], общие вычислительные затраты в которых фактически полностью определяются временем декомпозиции матрицы Якоби исходной системы. Обычно декомпозиция выполняется с выбором главного элемента по строке или столбцу, а иногда и по всей матрице. Сокращения вычислительных затрат достигают за счет применения одной матрицы на нескольких шагах интегрирования, а также за счет параллельных вычислений. В неявных и полуявных численных схемах алгоритмы с замораживанием матрицы Якоби реализуются достаточно просто потому, что данная матрица влияет только на скорость сходимости итерационного процесса.

Широкую известность получили безытерационные численные схемы за счет хороших свойств точности и устойчивости, а также за счет простоты реализации. К безытерационным методам относятся схемы типа Розенброка и различные их модификации. Однако в данных методах матрица Якоби включена непосредственно в вычислительные формулы, что приводит к определенным сложностям с замораживанием матрицы Якоби, то есть с применением одной матрицы на нескольких шагах интегрирования. В [2] доказана теорема о том, что максимальный порядок точности методов типа Розенброка равен двум, если они реализуются с применением одной матрицы на нескольких шагах интегрирования. Отметим, что данные численные формулы достаточно просты с точки зрения реализации, и как следствие, привлекательны для многих пользователей. В [3–5] предложен новый класс одношаговых безытерационных численных схем. Они обладают хорошими свойствами точности и устойчивости, а также достаточно просто реализуются с замораживанием матрицы Якоби. В дальнейшем эти численные схемы стали называть (m, k)-методами, и они определяются следующим образом.

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений:

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

где $y:[t_0,t_k] \to R^N$, $f:[t_0,t_k] \times R^N \to R^N$. Пусть заданы целые числа m и k, $k \le m$. Определим множества:

$$M_m = \{1, 2, \dots, m\},$$

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

$$J_i = \{m_j - 1 \in M_m \mid j > 1,\ m_j \in M_k,\ m_j \le i\},\ 2 \le i \le m.$$

Тогда (m,k)-методы имеют вид [3−5]:

$$y_{n+1} = y_n + \sum_{i=1}^{m} p_i K_i^{(n)},\ D_n = E - a h_n f_n',$$

$$D_n K_i^{(n)} = h_n f\Big(y_n + \sum_{j=1}^{i\text{-}1} \beta_{ij} K_j^{(n)}\Big) + \sum_{j \in J_i} \alpha_{ij} K_j^{(n)},\ i \in M_k, \tag{2}$$

$$D_n K_i^{(n)} = K_{i-1}^{(n)} + \sum_{j \in J_i} \alpha_{ij} K_j^{(n)},\ i \in M_m \setminus M_k,$$

где $K_i^{(n)} = \left(k_{i,1}^{(n)}, \dots, k_{i,N}^{(n)}\right)^T$, $1 \le i \le m$, − векторы приращений, E – единичная матрица, $f_n' = \partial f(y_n)/\partial y$ – матрица Якоби системы (1), h_n – шаг интегрирования, a, p_i, α_{ij} и β_{ij} – вещественные константы. Затраты на шаг в методах (2) следующие – один раз вычисляется матрица Якоби и осуществляется декомпозиция матрицы D_n, k раз вычисляется функция f и m раз реализуется обратный ход метода Гаусса.

Здесь представлены вычислительная схема (2,2)-метода, ее последовательный и параллельный алгоритмы интегрирования, в которых изменение величины шага основывается на оценке аналога глобальной ошибки. Формирование параллельного алгоритма (2,2)-метода состоит в использовании декомпозиции на подзадачи и установлении взаимосвязи между ними [6–7].

Постановка задачи. Для решения задачи (1) будем применять (2,2)-схему вида

$$y^{(n+1)} = y^{(n)} + p_1 K_1^{(n)} + p_2 K_2^{(n)},$$

$$D_n K_1^{(n)} = h_n f(y_n), \tag{3}$$

$$D_n K_2^{(n)} = h_n f(y_n + \beta K_1^{(n)}) + \alpha K_1^{(n)}.$$

Здесь $K_1^{(n)} = \left\{k_{1,j}^{(n)}\right\}$ и $K_2^{(n)} = \left\{k_{2,j}^{(n)}\right\}$, $1 \le j \le N$, − векторы приращений, $D_n = E - a h_n A_n$, матрица A_n представима в виде

$$A_n = f_n' + h_n B_n + O(h^2),$$

а матрица B_n не зависит от размера шага интегрирования. Такое представление A_n позволяет применять схему (3) с замораживанием как численной, так и аналитической матрицы Якоби. В случае использования схемы (3) с замораживанием матрицы Якоби, то есть в случае применения матрицы, вычисленной k шагов назад, получим:

$$A_n = \partial f(y_{n-k})/\partial y = f_n' - k h f_n'' f_n + O(h^2),$$

где имеет место

$$B_n = -k f_n'' f_n,\ 0 \le k \le Q_f,$$

Q_f – максимальное число шагов с замороженной матрицей Якоби,

$$f_n'' f_n = \sum_{i=1}^{N} \frac{\partial^2 f(y_n)}{\partial^2 y_i} f_i(y_n).$$

Если матрица Якоби вычисляется численно с шагом $r_j = c_j h$, где c_j, $1 \le j \le N$, – постоянные числа, то запишем:

$$A_n = f_n' + hG_n + O(h^2),\ B_n = G_n,$$

где элементы g_n^{ij} матрицы G_n определяются по формулам:

$$g_n^{ij} = \frac{1}{2} c_j \frac{\partial^2 f_i(y_n)}{\partial^2 y_j},\ 1 \le i, j \le N.$$

В случае замораживания численной матрицы Якоби можно записать:

$$A_n = f_n' + h(G_n - kf_n'' f_n) + O(h^2),$$

$$B_n = G_n - kf_n'' f_n.$$

Из приведенных выше формул следует, что вопрос о замораживании и численной аппроксимации матрицы Якоби можно рассматривать одновременно.

Требование второго порядка точности и учет дополнительной ошибки за счет аппроксимации матрицы Якоби приводит к коэффициентам:

$$a = 1 - 0{,}5\sqrt{2},\quad \alpha = -4/3,\quad p_1 = 1{,}25,\quad p_2 = 0{,}75,\quad \beta = 2/3.$$

Изменение величины шага основано на оценке аналога глобальной ошибки ε_n [5]. Учитывая очевидное соотношение

$$\varepsilon_n = \| K_1^{(n)} - K_2^{(n)} \| = O(h^2),$$

новый шаг h_{new} будем определять по формуле $h_{new} = qh_n$, где значение q находится из уравнения [4]:

$$q^2 \| \varepsilon_n \| = 7\varepsilon.$$

Если $q < 1$, то есть требуемая точность не выполняется, то происходит повторное вычисление решения с шагом $h_n = h_{new}$. Если имеет место неравенство $q \ge 1$, то вычисляется решение в следующей точке с шагом интегрирования $h_{n+1} = h_{new}$.

Ниже при записи алгоритмов будем использовать обозначения *LU_Decompos*() и *LU_Solution*() для функций, реализующих *LU*-факторизацию и решения систем алгебраических уравнений относительно приращений $K_1^{(n)}$ и $K_2^{(n)}$. Аналоги этих функций для параллельного алгоритма обозначим *Par_LU_Decompos*() и *Par_LU_Solution*().

Последовательный алгоритм. Последовательный алгоритм интегрирования с контролем точности вычислений формулируется следующим образом. Пусть для численного решения задачи (1) используется (2,2)-схема (3) и известно приближенное решение $y^{(n)}$ в точке t_n с шагом h_n. Тогда для получения приближенного решения $y^{(n+1)}$ в точке t_{n+1} справедлив следующий алгоритм.

Шаг 1. Вычислить $f_n = f(y^{(n)})$.

Шаг 2. Вычислить матрицу Якоби $f_n' = \partial f(y_n)/\partial y$.

Шаг 3. Сформировать матрицу $D_n = E - ah_n f_n'$.

Шаг 4. Разложить матрицу D_n, то есть $D_n = LU\_Decompos()$.

Шаг 5. Вычислить $K_1^{(n)}$, то есть $K_1^{(n)} = LU\_Solution()$.

Шаг 6. Вычислить $f_n = f(y^{(n)} + \beta K_1^{(n)}) + \alpha K_1^{(n)}$.

Шаг 7. Вычислить $K_2^{(n)}$, то есть $K_2^{(n)} = LU\_Solution()$.

Шаг 8. Вычислить величину $\varepsilon_n = \| K_1^{(n)} - K_2^{(n)} \|$.

Шаг 9. Вычислить значение q по формуле $q = (7\varepsilon / \| \varepsilon_n \|)^{0.5}$ и $h_{new} = qh_n$.

Если $q<1$, то возврат на шаг 3 с шагом интегрирования $h_n = h_{new}$.

Шаг 10. Вычислить приближенное решение по формуле (3):

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

Шаг 11. Вычислить решение в следующей точке с шагом интегрирования $h_{n+1} = h_{new}$.

Практическая реализация вычислений по данному алгоритму зависит от системы программирования, способа организации вычислений правой части системы (1) и способа факторизации (способа выбора ведущих элементов), а также от подходов к формированию числовых массивов и доступа к их элементам.

Параллельный алгоритм. При записи параллельного алгоритма предполагаем, что имеется вычислительная система из p процессоров $proc(j)$, $1 \le j \le p$. Матрица D_n размещена строчными блоками размером s, причем $s = N/p$, если N кратно p, и $s=[N/p]+q$ в противном случае. Для контроля точности численной схемы (3) введем функцию $accur\_control()$, а для ее выполнения назначим процессор $proc(1)$. Параллельный алгоритм вычисления приближенного решения $y^{(n+1)}$ с переменным шагом интегрирования формулируем следующим образом.

Пусть известно решение $y^{(n)}$в точке t_n с шагом h_n. Тогда для получения значения $y^{(n+1)}$ в точке t_{n+1} справедлив параллельный алгоритм, в котором на каждом процессоре $proc(j)$ формируется своя j-я часть $y_{s_j}^{(n+1)}$ вектора решения.

Шаг 1. В каждом $proc(j)$, $1 \le j \le p$, $(j-1)\cdot s + 1 \le s_j \le j \cdot s$,

выполнить $recv(y_{s_{1,\dots,p}}^{(n+1)},\ h;\ 1,\ \dots,\ p)$,

вычислить $f_{s_j}^{(n)}(y^{(n)})$ и матрицу Якоби J_j, $1 \le j \le p$.

Шаг 2. Сформировать матрицу $D_n = E - ah_n f_n'$.

Шаг 3. Разложить матрицу D_n, $D_n = Par\_LU\_Decompos()$.

Шаг 4. Вычислить $K_1^{(n)}, K_1^{(n)} = Par\_LU\_Solution()$.

Шаг 5. В каждом $proc(j)$, $1 \le j \le p$, $(j-1)\cdot s + 1 \le s_j \le j \cdot s$,

выполнить $recv(y_{s_{1,\dots,p}}^{(n)}, k_{1,s_{1,\dots,p}}^{(n)};\ 1,\dots,\ p)$

и вычислить $f_{s_j}^{(n)}(y^{(n)} + \beta K_1^{(n)}) + \alpha k_{1,s_j}^{(n)}$.

Шаг 6. Вычислить $K_2^{(n)}$, то есть $K_2^{(n)} = Par\_LU\_Solution()$.

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

вычислить локальные нормы погрешности:

$$|\varepsilon^{(n)}|_{s_j} = |k_{2,s_j}^{(n)} - k_{1,s_j}^{(n)}| / (|y_{s_j}^{(n)}| + r),$$

а затем вычислить глобальную норму:

$$\|\varepsilon^{(n)}\|_j = \max\{|\varepsilon_{(j-1)\cdot s+1}^{(n)}|, |\varepsilon_{(j-1)\cdot s+2}^{(n)}|, \dots, |\varepsilon_{j\cdot s}^{(n)}|\},$$

$$1 \le j \le p.$$

Шаг 8. В $proc(1)$ выполнить $accur\_control()$: $q = \sqrt{7\varepsilon / \|\varepsilon^{(n)}\|}$, $h_{new} = qh_n$.

Если $q<1$, то возврат на шаг 2 с шагом интегрирования $h_n = h_{new}$.

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

вычислить:

$$y_{s_j}^{(n+1)} = y_{s_j}^{(n)} + p_1 k_{1,\,s_j}^{(n)} + p_2 k_{2,\,s_j}^{(n)}$$

и выполнить $send(y_{s_{1,\dots,p}}^{(n+1)};\ 1,\dots,\ p)$.

Шаг 10. Вычислить решение в следующей точке с шагом интегрирования $h_{n+1} = h_{new}$.

Аналогичный алгоритм может быть сформулирован и для столбцовой схемы размещения матрицы D_n**.** Теоретические и экспериментальные оценки показали, что подходящими топологиями для выполнения алгоритма являются полный граф и гиперкуб. Реализация алгоритма осуществлялась на 99-процессорном кластере ИВМ СО РАН [8] с использованием языка $C++$ и функций библиотеки MPI.

Заключение. На примере (2,2)-схемы рассмотрены основные особенности разработки параллельных алгоритмов (*m*,*k*)-методов для многопроцессорных вычислительных систем кластерной архитектуры с использованием библиотеки MPI. Численные эксперименты на модельных задачах подтвердили, что количество пересылок и объем пересылаемых данных связан со структурой правой части системы. Подходящими топологиями для реализации являются полный граф и гиперкуб.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (*проект* 11-01-00106).

Список литературы

1. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи : монография / Э. Хайрер, Г. Ваннер. – М. : Мир, 1999. – 685 с.
2. Новиков В.А. Замораживание матрицы Якоби в методе типа Розенброка второго порядка точности / В.А. Новиков, Е.А. Новиков, Л.А. Юматова // ЖВМ и МФ. – 1987. – Т. 27. – № 3. – С. 385–390.
3. Новиков Е.А. Одношаговые безытерационные методы решения жестких систем / Е.А. Новиков, Ю.А. Шитов, Ю.И Шокин // ДАН СССР. – 1988. – Т. 301. – № 6. – С. 1310–1314.
4. Новиков Е.А. О классе (*m*,*k*)-методов решения жестких систем / Е.А. Новиков, Ю.А. Шитов, Ю.И Шокин // ЖВМ и МФ. – 1989. – Т. 29. – № 2. – С. 194–201.
5. Новиков Е.А. Алгоритм интегрирования переменной структуры для решения жестких задач на основе явного и *L*-устойчивого методов // Вестник СибГАУ. – 2008. – № 1 (18). – С. 75–78.
6. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – Спб. : БХВ – Петербург, 2002. – 806 с.
7. Hendrickson B. Graph partitioning models for parallel computing/ B. Hendrickson, Tamara G. Kolda // Parallel Computing. – 2002. – V. 26. – № 12. – P. 181–197.
8. Исаев С.В. Развитие Красноярского центра параллельных вычислений / С.В. Исаев, А.В. Малышев, В.В. Шайдуров // Вычислительные технологии. – 2006. – Т. 11. – С. 28–33.

Рецензенты:

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

Нужин Я.Н., д.ф.-м.н., профессор кафедры математического обеспечения дискретных устройств и систем (МОДУС) Института фундаментальной подготовки (ИФП) Сибирского федерального университета, г. Красноярск.

Кирьянов Б.Ф., д.т.н., профессор кафедры прикладной математики и информатики Новгородского государственного университета имени Ярослава Мудрого Министерства образования и науки РФ.
Работа получена 12.11.2011