УДК 519.852.6

ЯВНЫЕ МЕТОДЫ ТИПА РУНГЕ-КУТТЫ И ИХ ПАРАЛЛЕЛЬНЫЕ АНАЛОГИ

Ващенко Г.В.
Сибирский государственный технологический университет
г. Красноярск, Россия

Для численного решения задачи Коши рассматриваются параллельные аналоги алгоритмов явных методов типа Рунге-Кутты для систем обыкновен ных дифференциальных уравнений первого порядка. Предложены параллель ные вычислительные схемы методов, ориентированных на применение в много процессорных вычислительных системах кластерной архитектуры.

Рассматривается задачи Коши для систем обыкновенных дифференциаль ных уравнений первого порядка

$$y' = f(t, y),\, y(t_0) = y^0. \tag{1}$$

Для численного решения задачи (1) применяется явные s-стадийные методы типа Рунге-Кутты, $(n+1)$ -й шаг в которых задается формулами

$$y^{(n+1)} = y^{(n)} + h_{n+1}\sum_{i=1}^{s} b_i K_i^{(n)}, \tag{2}$$

где $K_i^{(n)} = f(t_n + c_i h_{n+1}, y^{(n)} + h_{n+1}\sum_{j=1}^{i-1} a_{ij} K_j^{(n)}), i = 1, 2, \mathbf{K}, s.$

Конкретный метод Рунге – Кутты определяется набором коэффициентов b_i, c_i, a_{ij} , $1 \le i \le s$, $2 \le j \le (i-1)$ [2-4].

Основной подход при конструировании параллельных методов состоял в распараллеливании последовательных численных алгоритмов, использовании декомпозиции и анализе информационных взаимосвязей между подзадачами [1,5].

1. Последовательная вычислительная схема

Для определенности зададимся некоторым отрезком $[t_0, T]$, введем равномерную сетку $w_n = (t_0, t_1, \ldots, t_n)$ с величиной шага $h_{n+1} = h$ и на сетке w_n в начальный момент времени t_0 в качестве начального условия зададим вектор $y^{(0)} = (y_1^{(0)}, y_2^{(0)}, \mathbf{K}, y_N^{(0)})$. Определение значений компонент вектора приближенного решения $\tilde{y}^{(n+1)} = (\tilde{y}_1^{(n+1)}, \tilde{y}_2^{(n+1)}, \mathbf{K}, \tilde{y}_N^{(n+1)})$ осуществляется по формуле (2), записанной для вычисления каждой компоненты вектора $\tilde{y}^{(n+1)}$

$$\left.\begin{aligned} &\tilde{y}_j^{(n+1)} = \tilde{y}_j^{(n)} + h\,[b_1 K_{1,j}^{(n)} + b_2 K_{2,j}^{(n)} + \mathbf{K} + b_s K_{s,j}^{(n)}],\\ &y_j^{(0)} = y_j(t_0), \quad j = 1, 2, \mathbf{K}, N, \end{aligned}\right\} \tag{3}$$

где $K_m^{(n)} = f_j(t_n + c_s h, \tilde{y}^{\ (n)} + h\sum_{k=1}^{m-1} a_{m,k} K_k^{(n)}),\ m=1,2,\mathbf{K},s.$

Формулы (2) и (3) показывают, что определение значения $\tilde{y}^{(n+1)}$ сводится к строго последовательному вычислению коэффициентов $K_1^{(n)}, K_2^{(n)}, \mathbf{K}, K_s^{(n)}$, их умножению на коэффициенты b_i, i = 1,2, …, s , соответственно, и последую щему суммированию.

2 Параллельная явная s –стадийная вычислительная схема

Последовательная схема (3) дает основание для организации двух типов параллельных вычислений:

а) вычисление отдельных компонент векторов коэффициентов $K_1^{(n)}$, $K_2^{(n)}$, $\mathbf{K}, K_s^{(n)}$ и вектора численного решения $\tilde{y}^{(n+1)} = (\tilde{y}_1^{(n+1)}, \tilde{y}_2^{(n+1)},\ \ldots,\ \mathbf{K}, \tilde{y}_N^{(n+1)})$;

б) вычисление отдельных операций внутри одного шага метода.

Вычисление отдельных операций внутри одного шага метода обеспечивает небольшую степень параллелизма и поэтому не рассматривается.

С целью выявления максимального независимого набора операций при вычислении коэффициентов $K_i^{(n)}$, i = 1, 2, …, s используется аппарат графов зависимости. Анализ графа зависимостей, в предположении, что размерность N исходной системы (1) кратна числу компьютеров p, $N = kp$ и схема размещения блочная обеспечивает возможность записи параллельных схем. В каждом компьютере размещено и вычисляется последовательно по k компонент векторов коэффициентов, $K_1^{(n)}$, $K_2^{(n)}$,…, $K_s^{(n)}$. Вычисления $\tilde{y}^{(n+1)}$ в n-ом узле сетки w_n реализуются по правилу, показанному на рис.

Алгоритм. Пусть задана система (1), правая часть которой гладкая по всем аргументам y_i, $1 \le i \le N$ и пусть на заданном отрезке $[t_0, T]$ определена равномерная сетка w_n. Тогда для решения задачи Коши вычислительной системе из $p = N/k$ компьютеров, $Comp(i)$, $1 \le i \le N$, и блочной схемой хранения, справедлив параллельный алгоритм.

Шаг 1. Вычислить $K_{1,l_z}^{(n)}$ и $\alpha_{2,l_z}^{(n)}$ по формулам

$$\alpha_{2,l_z}^{(n)} = y_{l_z}^{\ (n)} + h a_{m,1} K_{1,l_z}^{(n)}\ z=1,2,\mathbf{K},p; l_z=(z-1)k+1,2,\mathbf{K},zk,$$

и переслать $K_{1,l_z}^{(n)}$ и $\alpha_{2,l_z}^{(n)}$ всем $(p–1)$ компьютерам.

Шаг 2. Вычислить $K_{2,l_z}^{(n)}$ и $\alpha_{3,l_z}^{(n)}$ по формулам

$$\alpha_{3,l_z}^{(n)} = y_{l_z}^{\ (n)} + h\sum_{j=1}^{2} a_{m,j} K_{j,l_z}^{(n)}, z=1,2,\mathbf{K},p; l_z=(z-1)k+1,2,\mathbf{K},zk$$

и переслать $K^{(n)}_{2,l_z}$ и $\alpha^{(n)}_{3,l_z}$ всем $(p–1)$ компьютерам.

.. . .

Шаг *s*. Вычислить $K^{(n)}_{s,l_z}$.

Шаг (*s*+1). Вычислить

$$y^{(n+1)}_{l_z} = y^{(n)}_{l_z} + \sum_{m=1}^{s} b_m K^{(n)}_{m,l_z},\ z=1,2,\mathbf{K},p;\ l_z=(z-1)k+1,2,\mathbf{K},zk. \quad .$$

Шаг (*s*+2). Сохранить $\tilde{y}^{(n)}_{l_z} = \tilde{y}^{(n+1)}_{l_z}$.

Шаг (*s*+3). Переслать $\tilde{y}^{(n+1)}_{l_z}$ всем $(p–1)$ компьютерам.

Рис. Вычисление коэффициентов и вектора
приближенного решения при $N = kp$

На последнем шаге в каждом компьютере вычисляется и сохраняется своя часть вектора $\tilde{y}^{(n+1)}$. Таким образом, после параллельного вычисления коэффициентов параллельно определяется такое же количество компонент вектора приближенного решения $\tilde{y}^{(n+1)}$. Число пересылок для одного узла сетки w_n составит $s(p–1)^2 \approx O(sp^2)$.

Параллельный аналог формулы (3) для равномерной сетки может быть записан в виде

$$y^{(n+1)}_{l_z} = y^{(n)}_{l_z} + \sum_{m=1}^{s} b_m K^{(n)}_{m,l_z},\ z=1,2,\mathbf{K},p;\ l_z=(z-1)k+1,2,\mathbf{K},zk,$$

где $K^{(n)}_{m,l_z} = f_{l_z}\left(t_n + c_s h, \tilde{y}^{(n)} + h\sum_{k=1}^{m-1} a_{m,k} K^{(n)}_{k,l_z}\right),\ m=1,2,\mathbf{K},s$.

Заметим, что в отдельных случаях, зависящих от типа и вида исходной системы (1), может быть достигнута максимальная степень параллелизма и, соответственно, сокращение временных затрат выполнения вычислений по разработанному алгоритму. Так, например, для двустадийной схемы Рунге-Кутты параллельный алгоритм записывается следующим образом.

Шаг 1. Вычислить $K^{(n)}_{1,l_z}$ и $\alpha^{(n)}_{2,l_z}$ по формулам

$$\alpha^{(n)}_{2,l_z} = y_{l_z}{}^{(n)} + h a_{m,1} K^{(n)}_{1,l_z}\ z=1,2,\mathbf{K},p;\ l_z=(z-1)k+1,2,\mathbf{K},zk,$$

и переслать $K^{(n)}_{1,l_z}$ и $\alpha^{(n)}_{2,l_z}$ всем $(p–1)$ компьютерам.

Шаг 2. Вычислить $K^{(n)}_{2,l_z}$ и $\alpha^{(n)}_{3,l_z}$ по формулам

$$\alpha^{(n)}_{3,l_z} = y_{l_z}{}^{(n)} + h\sum_{j=1}^{2} a_{m,j} K^{(n)}_{j,l_z},\ z=1,2,\mathbf{K},p;\ l_z=(z-1)k+1,2,\mathbf{K},zk$$

Шаг 3. Вычислить

$$y_{l_z}^{(n+1)} = y_{l_z}^{(n)} + \sum_{m=1}^{2} b_m K_{m,l_z}^{(n)}, z = 1, 2, \mathbf{K}, p; l_z = (z-1)k + 1, 2, \mathbf{K}, zk. \quad .$$

Шаг 4. Сохранить $\tilde{y}_{l_z}^{(n)} = \tilde{y}_{l_z}^{(n+1)}$.

Шаг 5. Переслать $\tilde{y}_{l_z}^{(n+1)}$ всем $(p–1)$ компьютерам.

Рассмотренные параллельные схемы явных методов типа Рунге-Кутты ориентированы на реализацию в многопроцессорных вычислительных системах кластерной архитектуры с использованием технологии MPI. MPI имеет в составе коммуникационные операции попарные и коллективные обмены, средства организации виртуальных топологий. Исследования представленных параллельных схем показали, что для их реализация наиболее подходящими могут быть топологии кольцо, линейка, решетка и гиперкуб. Разработанные схемы могут служить основой для разработки параллельных алгоритмов решения задачи Коши явными методами с контролем точности и устойчивости, алгоритмов переменного порядка и шага, а также возможной автоматизации построения методов интегрирования с адаптивной областью устойчивости.

Литература

[1] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления / Спб.: БХВ – Петербург, 2002. – 806 с.

[2] Новиков Е.А. Явные методы для жестких систем / Новосибирск: Наука, 1997. – 197с.

[3] Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи / М.: Мир, 1999. – 685с.

[4] Jackson K., Norsett S. The potential for parallelism in Runge -Kutta methods. Part I: RK formulas in standart form // SIAM J. Numer. Anal., v. 32, 1996. – p. 49–82

[5] Hendrickson B., Kolda Tamara G. Graph partitioning models for parallel computing // Parallel Computing, т. 26, № 12, 2002. – p. 181–197.