

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧИ КОШИ

Л.П. Фельдман, О.А. Дмитриева
Донецкий национальный технический университет
З. Гербер
Лейпцигский университет, Институт информатики

Л.П. Фельдман, О.А. Дмитриева, З. Гербер. Високопродуктивні алгоритми чисельного рішення задачі Коши. Робота, що пропонується, спирається на дослідження, результати яких раніше опубліковані авторами в [2-4, 6-8] і присвячена проблемам побудови ефективних алгоритмів рішення задачі Коши для звичайних диференціальних рівнянь. Розроблені високопродуктивні алгоритми, орієнтовані на реалізацію в обчислювальних системах з паралельною архітектурою. Гранична точність отриманих методів значно перевершує точність відомих методів рішення при тих же значеннях кроку інтегрування.

Введение

Практическое использование большинства методов численного интегрирования дифференциальных уравнений, которые ориентированы на вычислительные системы с параллельной архитектурой не всегда оправдано, поскольку многие из них либо обладают численной неустойчивостью, либо имеют очень сложную структуру, приводящую к потере эффективности. К методам, лишенным указанных недостатков, можно отнести блочные [1]. Блочным будем называть метод, при котором для блока из k точек новые k значений функции вычисляются одновременно. Эта особенность методов, во-первых, согласуется с архитектурой параллельных вычислительных систем, а, во-вторых, позволяет вычислять коэффициенты разностных формул не в процессе интегрирования, а на этапе разработки метода, что значительно увеличивает эффективность счета. Данная работа опирается на исследования, результаты которых ранее опубликованы авторами в [2-4, 6-8] и посвящена проблемам построения эффективных алгоритмов решения задачи Коши для обыкновенных дифференциальных уравнений.

1. Алгоритмы численного решения обыкновенных дифференциальных уравнений

1.1. Алгоритмы параллельного решения нелинейной разностной задачи

Решение задачи Коши

$$\frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0 \quad (1)$$

блочными методами подробно рассмотрено в [2-4]. Вопросы оценки погрешности аппроксимации многошаговых блочных методов и сходимости представлены в [6-8].

Для вычисления приближенных значений решения задачи Коши (1) необходимо решить нелинейную систему уравнений

$$\frac{u_{n,i} - u_{n,0}}{i\tau} = \sum_{j=1}^k b_{i,j} F_{n-1,j} + \sum_{j=1}^k a_{i,j} F_{n,j}, \quad i = \overline{1, k}, \quad n = 1, 2, \dots, N \quad (2)$$

где $F_{n,j} = f(t_n + j\tau, u(t_n + j\tau))$.

Используем для этого следующий итерационный процесс

$$u_{n,i,0} = u_{n,0} + i\tau \sum_{j=1}^k c_{i,j} F_{n-1,j}, \quad i = \overline{1, k}, \quad n = 1, 2, \dots, N$$

$$u_{n,i,s+1} = u_{n,0} + i\tau \left(\sum_{j=1}^k b_{i,j} F_{n-1,j} + \sum_{j=1}^k a_{i,j} F_{n,j,s} \right), \quad s = \overline{0, k-1}, \quad (3)$$

который позволяет проводить вычисления параллельно для каждого узла блока. Покажем, что после выполнения k шагов вычислений по формулам (3) локальная ошибка в узлах блока будет иметь порядок $O(\tau^{2k+1})$.

Обозначим через $u_{n,i}^{[p]}$ значение решения в точке $t_{n,i}$, найденное с локальной погрешностью $O(\tau^p)$, а через

$$F_{n,i}^{[p]} = f(t_{n,i}, u_{n,i}^{[p]}),$$

вычисленное при этом значение правой части уравнения (1). Допустим, что значения $u_{n,0}$ вычислены с локальной ошибкой $O(\tau^{2k+1})$, тогда и правая часть уравнения может быть вычислена с такой же погрешностью, т. е. имеем $F_{n,0}^{[2k+1]}$. Выполнив первый шаг вычислений по первой формуле (3), получим

$$u_{n,i,0}^{[k+1]} = u_{n,0}^{[2k+1]} + i\tau \sum_{j=1}^k c_{i,j} F_{n,j,0}^{[2k+1]}, \quad i = \overline{1, k},$$

так как локальная ошибка экстраполяционной формулы Адамса имеет порядок $O(\tau^{k+1})$ [7]. Параллельные вычисления по второй формуле (3) при $s=0$ дадут следующие результаты:

$$u_{n,i,1}^{[k+2]} = u_{n,0}^{[2k+1]} + i\tau \left(\sum_{j=1}^k b_{i,j} F_{n-1,j}^{[2k+1]} + \sum_{j=1}^k a_{i,j} F_{n,j,0}^{[k+1]} \right), i = \overline{1, k},$$

так как функции $F_{n,j,0}^{[k+1]} = f(t_{n,j}, u_{n,j,0}^{[k+1]})$, $j = \overline{1, k}$ могут быть вычислены с локальной ошибкой $O(\tau^{k+1})$. Вычисления на следующем шаге $s=1$ улучшат точность результатов на один порядок. Продолжая вычисления до $s=k-1$, получим результаты, соответствующие локальным предельным точностям приближенных формул (3), т. е.

$$u_{n,i,k}^{[2k+1]} = u_{n,0}^{[2k+1]} + i\tau \left(\sum_{j=1}^k b_{i,j} F_{n-1,j}^{[2k+1]} + \sum_{j=1}^k a_{i,j} F_{n,j,2k-1}^{[2k]} \right), i = \overline{1, k},$$

поскольку разностные уравнения (2) аппроксимируют дифференциальное уравнение (1) с порядком $O(\tau^{2k})$, дальнейшие шаги не дадут повышения порядка точности результатов.

Для одношаговых блочных методов

$$\frac{u_{n,i} - u_{n,0}}{i\tau} = b_i F_{n,0} + \sum_{j=1}^k a_{i,j} F_{n,j}, i = \overline{1, k}, n = 1, 2, \dots, N \quad (4)$$

формулы итерационного процесса могут быть получены в следующем виде

$$u_{n,i,0} = u_{n,0} + i\tau F_{n,0}, i = \overline{1, k}, n = 1, 2, \dots, N$$

$$u_{n,i,s+1} = u_{n,0} + i\tau \left(b_i F_{n,0} + \sum_{j=1}^k a_{i,j} F_{n,j,s} \right), s = \overline{0, k-1}. \quad (5)$$

Покажем, что после выполнения $k+1$ шагов вычислений по формулам (5) локальная ошибка в узлах блока будет иметь порядок $O(\tau^{k+2})$. Допустим, что значения $u_{n,0}$ вычислены с локальной ошибкой $O(\tau^{k+2})$, тогда и правая часть уравнения может быть вычислена с такой же погрешностью, т. е. имеем $F_{n,0}^{[k+2]}$. Выполнив первый шаг вычислений по первой формуле (5), получим $u_{n,i,0}^{[2]} = u_{n,0}^{[k+2]} + i\tau F_{n,i,0}^{[k+2]}$, $i = \overline{1, k}$, так как локальная ошибка формулы Эйлера имеет порядок $O(\tau^2)$. Параллельные вычисления по второй формуле (5) при $s=0$ дадут следующие результаты:

$$u_{n,i,1}^{[3]} = u_{n,0}^{[k+2]} + i\tau \left(b_i F_{n,0}^{[k+2]} + \sum_{j=1}^k a_{i,j} F_{n,j,0}^{[2]} \right), i = \overline{1, k},$$

так как функции **A** могут быть вычислены с локальной ошибкой $O(\tau^2)$. Вычисления на следующем шаге $s=1$ улучшат точность результатов на один порядок. Продолжая вычисления до $s=k-1$, получим результаты соответствующие локальным предельным точностям приближенных формул (5), т. е.

$$u_{n,i,k}^{[k+2]} = u_{n,0}^{[k+2]} + i\tau (b_i F_{n,0}^{[k+2]} + \sum_{j=1}^k a_{i,j} F_{n,j,k-1}^{[k+1]}), i = \overline{1, k},$$

поскольку разностные уравнения (4) аппроксимируют дифференциальное уравнение (1) с порядком $O(\tau^{k+1})$, последующие итерации повышения порядка точности результатов не дадут.

1.2 Алгоритм решения нелинейной разностной задачи методом Ньютона.

Систему нелинейных уравнений (2) с неизвестными $u_{n,i}, i = \overline{1, k}$ в фиксированном блоке n перепишем в виде

$$u_{n,i} = u_{n,0} + \tau i \sum_{j=1}^k b_{i,j} F_{n-1,j} + \tau i \sum_{j=1}^k a_{i,j} F_{n,j}, i = \overline{1, k}, n = 1, 2, \dots, N. \quad (6)$$

Определим начальные приближения для неизвестных

$$u_{n,i}^{(0)} = u_{n,0} + i\tau \sum_{j=1}^k c_{i,j} F_{n-1,j}, i = \overline{1, k}, n = 1, 2, \dots, N.$$

Предположим теперь, что найдены на шаге s приближенные значения решения уравнений $u_{n,i}^{(s)}, i = \overline{1, k}$. Найдем поправки $\varepsilon_{n,i}^{(s)}$, уточняющие полученные приближенные значения, т. е. следующие приближенные значения будут равны

$$u_{n,i}^{(s+1)} = u_{n,i}^{(s)} + \varepsilon_{n,i}^{(s)}, i = \overline{1, k}. \quad (7)$$

После подстановки (7) в (6) и, ограничиваясь линейными членами разложений в ряды по приращениям $\varepsilon_{n,i}^{(s)}$, получим

$$u_{n,i}^{(s)} + \varepsilon_{n,i}^{(s)} = u_{n,0} + \tau i \sum_{j=1}^k b_{i,j} F_{n-1,j} + \tau i \sum_{j=1}^k a_{i,j} F_{n,j}^{(s)} + \tau i \sum_{j=1}^k a_{i,j} \varepsilon_{n,j}^{(s)} \frac{\partial F_{n,j}^{(s)}}{\partial u_{n,j}},$$

где $F_{n,j}^{(s)} = f(t_{n,j}, u_{n,j}^{(s)})$, $\frac{\partial F_{n,j}^{(s)}}{\partial u_{n,j}} = \frac{\partial f(t_{n,j}, u_{n,j}^{(s)})}{\partial u_{n,j}}$.

Таким образом, для определения поправок $\varepsilon_{n,i}^{(s)}$ получена следующая система линейных алгебраических уравнений

$$\varepsilon_{n,i}^{(s)} - \tau i \sum_{j=1}^k a_{i,j} \varepsilon_{n,j}^{(s)} \frac{\partial F_{n,j}^{(s)}}{\partial u_{n,j}} = u_{n,0} + \tau i \sum_{j=1}^k b_{i,j} F_{n-1,j} + \tau i \sum_{j=1}^k a_{i,j} F_{n,j}^{(s)} - u_{n,i}^{(s)} \quad (8)$$

Запишем (8) в матричной форме

$$(E - \tau DA \partial F_n^{(s)}) \varepsilon_n^{(s)} = u_{n,0} \mathbf{e} + \tau D (\mathbf{B} F_{n-1} + \mathbf{A} f_n^{(s)}) - u_n^{(s)}. \quad (9)$$

Здесь дополнительно введены обозначения:

$f_n^{(s)}$ - вектор столбец с компонентами $f(t_{n,i}, u_{n,i}^{(s)})$, $i = \overline{1, k}$,

$\partial F_n^{(s)}$ - диагональная матрица, элементы которой $\frac{\partial F_{n,i}^{(s)}}{\partial u_{n,i}}$, $i = \overline{1, k}$.

Тогда погрешность решения задачи на s - том шаге

$$\varepsilon_n^{(s)} = (\mathbf{E} - \tau \mathbf{D} \mathbf{A} \partial F_n^{(s)})^{-1} * u_{n,0} \mathbf{e} + \tau \mathbf{D} (\mathbf{B} F_{n-1} + \mathbf{A} f_n^{(s)}) - u_n^{(s)}. \quad (10)$$

Для одношаговых разностных методов (4) уравнение погрешности на на s - том шаге можно представить в виде

$$\varepsilon_n^{(s)} = (\mathbf{E} - \tau \mathbf{D} \mathbf{A} \partial F_n^{(s)})^{-1} * (u_{n,0} \mathbf{e} + \tau \mathbf{D} (F_{n,0} \mathbf{b} + \mathbf{A} f_n^{(s)}) - u_n^{(s)}). \quad (11)$$

1.3 Сравнительные характеристики численного решения тестовых задач блочными методами

Выберем в качестве иллюстрации решения блочными методами умеренно устойчивую задачу [9]

$$x' = -10(t-1)x, \quad x(0) = 1, \quad (12)$$

Аналитическое решение рассматриваемой задачи имеет вид

$$x(t) = \text{Exp}(-5t(t-2)). \quad (13)$$

Найдем решение задачи (12) одношаговым (с размерностью блока $k=4$), многошаговым (с блоками 4×4) блочными методами и методом Ньютона (с размерностью блоков 4×4). Шаги интегрирования τ_1 для метода (2) и τ_2 для метода (4) выберем из [2]. Для этого потребуются значения норм матриц \mathbf{A} , приведенных в [4] Оценим значение константы L , найдем для уравнения (12) $\text{Sup} \left| \frac{\partial f}{\partial x} \right|$ на отрезке интегрирования $0 \leq t \leq 2$. Примем $L=10$.

Решение выполним в системе *Mathematica*®.

Для одношагового итерационного четырехточечного метода

$$\|A\|_c = 1.4375, \quad \tau_0 = \frac{1}{4*10*1.4375} = 0.0174.$$

Для многошагового итерационного четырехточечного метода

$$\|A\|_c = 0.9857, \quad \tau_0 = \frac{1}{4*10*0.9857} = 0.02536.$$

Шаг для метода Ньютона выберем таким же, как и для итерационного многошагового метода. Поскольку точное решение известно (13), то по результатам приближенного решения можно оценить максимальную ошибку, которая всеми методами достигается в точке $t=1$. Сведем полученные результаты решения в таблицу, характеризующую зависимость ошибки от числа выполненных итераций.

Таблица 1. Ошибка в критической точке $t=1$

Число итераций	Максимальная ошибка одношаговой разностной схемы при решении итерационным методом		Максимальная ошибка многошаговой разностной схемы при решении итерационным методом		Максимальная ошибка многошаговой разностной схемы (метод Ньютона)
	Априорн.	Экспер.	Априорн.	Экспер.	Экспер.
1	21.445	15	0.4285	0.2987	0.2987
2	4.8223	2.3	$4.78 \cdot 10^{-3}$	$3.35 \cdot 10^{-3}$	$9.04 \cdot 10^{-3}$
3	0.4311	0.25	$7.29 \cdot 10^{-4}$	$1.05 \cdot 10^{-4}$	$1.52 \cdot 10^{-4}$
4	$2.78 \cdot 10^{-2}$	0.025	$1.49 \cdot 10^{-5}$	$2.93 \cdot 10^{-6}$	$2.01 \cdot 10^{-6}$
5	$3.59 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$	$2.80 \cdot 10^{-6}$	$7.58 \cdot 10^{-8}$	$2.4234 \cdot 10^{-7}$

Решение задачи (12) методом Рунге- Кутты четвертого порядка точности шагом τ_1 дает максимальную ошибку равную $\varepsilon=0.00168$, что на порядок хуже чем в одношаговом блочном методе, максимальная ошибка для которого самая высокая среди приведенных в таблице значений. Воспользуемся формулой априорной оценки погрешности [2], чтобы определить шаг интегрирования обеспечивающий ту же точность $\varepsilon=0.00014$, которая была получена при численном решении в предыдущем примере. Для рассматриваемого случая эта формула примет вид

$$\|z\|_c \leq \frac{3 \max |x^{(6)}(t)|}{160} \tau^5 \frac{E^{T(\|b\|_c + L\|A\|_c)} - 1}{(\|b\|_c + L\|A\|_c)}, \quad 0 \leq t \leq 2.$$

Задав требуемую точность, получим необходимый для этого шаг

$$\tau_0 = \sqrt[5]{\frac{160\varepsilon(\|b\|_c + L\|A\|_c)}{3M_6 E^{T(\|b\|_c + L\|A\|_c)}}}, \quad M_6 = \max |x^{(6)}(t)|, \quad 0 \leq t \leq 2. \quad (14)$$

Найдем M_6 , используя средства *Mathematica*®. Графическая интерпретация решения приведена на рис. 1. Из рисунка видно, что максимум модуля $x^{(6)}(t)$ достигает при $t=1$. Задав $\varepsilon=0.00014$, получим $\tau_0 = 0.000146$, что в сто раз меньше того шага, с которым фактически интегрировали задачу. Таким образом, априорная оценка дает очень завышенную оценку погрешности.

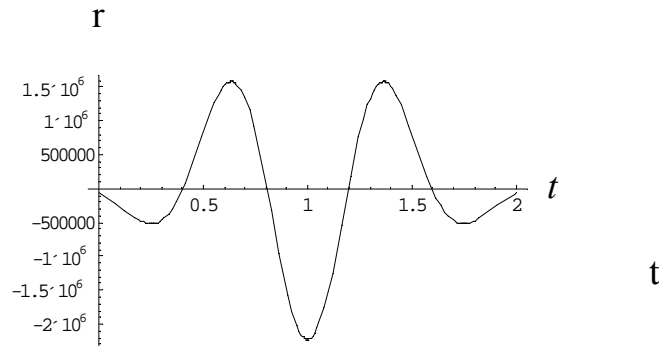


Рис. 1 Распределение погрешностей решения одношаговым методом

При численном решении задачи Коши для сравнительной характеристики методов можно рассматривать различные показатели. В случае произвольной правой части уравнения о трудоемкости метода естественно судить по числу обращений для вычисления значений правой части уравнения на каждый узел сетки.

Оценим время выполнения алгоритма, определяемого формулами (3) и (5) на одном процессоре. Обозначим через t_f – время вычисления значения функции $f(t,x)$, t_{ad} , t_{mul} – время выполнения операции сложения и умножения соответственно. Время вычисления приближенных значений решения по формулам (3) с локальной точностью $O(\tau^{2k+1})$ составит

$$T_{\text{многошаг.}} = k(2k+1) t_f + 4k^3 t_{ad} + 2k^2(2k+1) t_{mu}.$$

Время вычисления с локальной точностью $O(\tau^{k+2})$ во всех k узлах блока составит по формулам (5) составит

$$T_{\text{одношаг.}} = (k^2+1) t_f + k[k(k-1)+2] t_{ad} + (k^3+2k+1) t_{mu}.$$

Для параллельного выполнения вычислений по формулам (3) и (5) будем использовать вычислительную систему SIMD структуры с линейкой процессорных элементов. При реализации алгоритма каждый процессорный элемент может выполнить любую арифметическую операцию за один такт; временные затраты, связанные с обращением к запоминающему устройству, отсутствуют. Для простоты изложения рассматривается случай, когда количество процессорных элементов совпадает с размерностью блока. Закрепим за каждым узлом блока процессор. При реализации алгоритма на k процессорах можно одновременно вычислять значения $F_{n,i,s}$, а затем также одновременно получить по формулам значения $u_{n,i,s}$ для каждого фиксированного s . Объединим процессоры в кольцо, чтобы иметь возможность одновременной передачи данных соседним процессорам. Обозначим через t_{ta} – время передачи числа соседнему процессору. Время параллельного вычисления приближенных значений решения с той же точностью для всех узлов блока составит в первом случае

$$T_{\text{многошаг.}} = 2(k+1) t_f + 2[k(2k+1)+1] t_{ad} + 3[k(2k+1)+1] t_{mu} + k(2k+1) t_{ta}.$$

$$T_{\text{одношаг.}} = (k+1) t_f + (k^2-k+3) t_{ad} + (k^2+3) t_{mu} + k(k-1) t_{ta}.$$

Ускорение параллельных многоточечных алгоритмов можно будет теперь вычислить по формуле $W(k) = T_s / T_p$. Если учитывать только время вычислений правой части уравнений, т.к. времена выполнения арифметических операций и обмена значительно меньше времени вычисления правой части, то ускорение k точечного многошагового параллельного алгоритма можно считать приближенно равным

$$W(k)_{\text{многошаг.}} = k(2k+1)/(2k+1).$$

Ускорение параллельного одношагового k точечного алгоритма можно будет теперь считать приближенно равным

$$W(k)_{\text{одношаг.}} = (k^2 + 1)/(k+1).$$

В качестве еще одной оценки степени параллелизма алгоритма используется коэффициент эффективности, который определяется отношением ускорения к числу используемых процессорных элементов. Нетрудно видеть, что для двух типов рассмотренных в статье блочных методов этот коэффициент близок к единице.

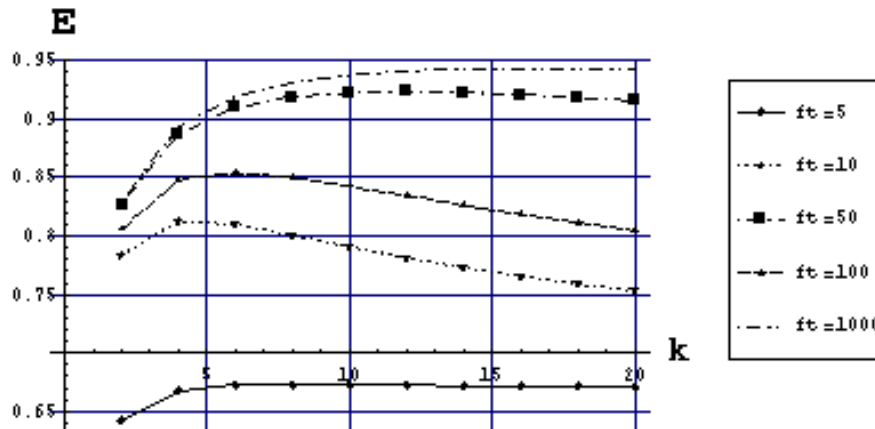


Рис. 2. Эффективность многошаговых многоточечных блочных методов в зависимости от размерности блока k и трудоемкости вычисления правой части tf

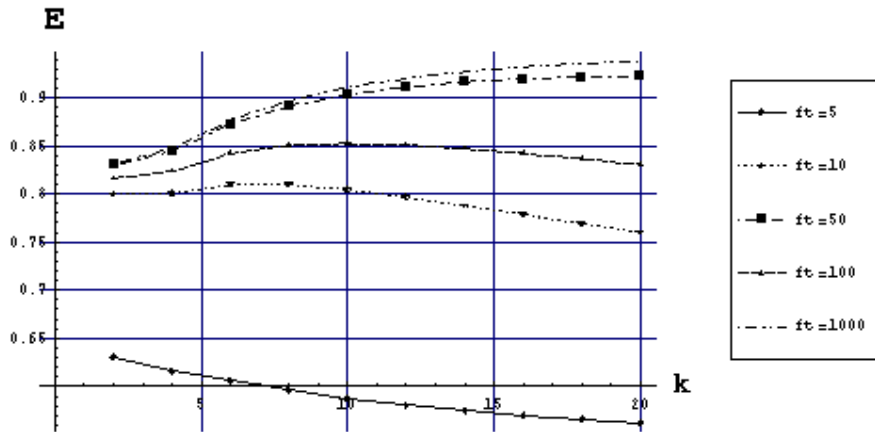


Рис. 3. Эффективность одношаговых многоточечных блочных методов в зависимости от размерности блока k и трудоемкости вычисления правой части tf

2. Параллельные блочные многошаговые алгоритмы численного решения систем обыкновенных дифференциальных уравнений

Разработанные и обоснованные выше многошаговые многоточечные блочные методы решения обыкновенных дифференциальных уравнений можно распространить и на решение систем ОДУ. Рассмотрим решение задачи Коши для системы обыкновенных дифференциальных уравнений

$$\frac{dx}{dt} = \Phi(t, x(t)), \quad x(t_0) = x^0 = (x_1^0, x_2^0, \dots, x_m^0)^T, \quad (15)$$

где

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{pmatrix} \begin{matrix} \text{вектор функция} \\ \text{неизвестных} \end{matrix}, \quad \Phi = \begin{pmatrix} f_1(t, x_1, x_2, \dots, x_m) \\ f_2(t, x_1, x_2, \dots, x_m) \\ \dots \\ f_m(t, x_1, x_2, \dots, x_m) \end{pmatrix} \begin{matrix} \text{вектор функция} \\ \text{правой части} \end{matrix}$$

k -точечным многошаговым блочным методом.

В общем случае уравнения многошаговых разностных методов для блока из n точек с учетом введенных обозначений можно записать в виде

$$\frac{u_{n,q,i} - u_{n,q,0}}{i\tau} = \sum_{j=1}^k b_{i,j} F_{n-1,q,j} + \sum_{j=1}^k a_{i,j} F_{n,q,j}, \quad i = \overline{1, k}, \quad q = \overline{1, m}, \quad n = 1, 2, \dots, N, \quad (16)$$

где $F_{n,q,j} = f_q(t_n + j\tau, u_1(t_n + j\tau), u_2(t_n + j\tau), \dots, u_q(t_n + j\tau))$.

Индекс q означает порядковый номер дифференциального уравнения в системе.

Для вычисления приближенных значений решения задачи Коши (15) необходимо решить нелинейную систему уравнений (16). Итерационный процесс будет выглядеть следующим образом

$$u_{n,q,i,0} = u_{n,0} + i\tau \sum_{j=1}^k c_{i,j} F_{n-1,q,j}, \quad i = \overline{1,k}, \quad q = \overline{1,m}, \quad n = 1, 2, \dots, N$$

$$u_{n,q,i,s+1} = u_{n,q,0} + i\tau \left(\sum_{j=1}^k b_{i,j} F_{n-1,q,j} + \sum_{j=1}^k a_{i,j} F_{n,q,j,s} \right), \quad i = \overline{1,k}, \quad s = \overline{0, k-1}, \quad (17)$$

который позволяет проводить вычисления параллельно для каждого узла блока. Выше было показано, что после выполнения k шагов локальная ошибка в узлах блока будет иметь порядок $O(\tau^{2k+1})$.

Заключение

В представляемой статье предложены высокопроизводительные алгоритмы решения задачи Коши для обыкновенных дифференциальных уравнений, не требующие вычисления значений в промежуточных точках, что значительно повышает эффективность счета. Осуществлена модификация алгоритмов, реализующих разностные блочные методы распараллеливания для решения систем обыкновенных дифференциальных уравнений, которая с точки зрения временных затрат позволяет с большей эффективностью по сравнению с существующими алгоритмами находить численное решение задачи Коши. Предложены итерационные параллельные алгоритмы численного решения нелинейной разностной задачи Коши, позволяющие получать результаты с заданной степенью точности. Приведены сравнительные характеристики численного решения тестовых задач и оценки параллелизма. Решение тестовых задач показало, что экспериментальные оценки ускорения и эффективности близки к потенциальным. Обоснована нецелесообразность применения для нахождения значений на начальном отрезке методов последовательных сближений и одношаговых методов, поскольку при их использовании число операций, затраченных на разгонку, может превзойти трудоемкость решения всей задачи многошаговым блочным методом. Для каждого из приведенных алгоритмов рассчитаны трудоемкости получения численного решения, получены характеристики параллелизма, которые достаточно убедительно свидетельствуют о высокой эффективности предложенных методов. Разработана параллельная реализация многошаговых методов Ньютона для решения нелинейной разностной задачи. Предельная точность полученных методов значительно превосходит точность известных методов решения при тех же значениях шага интегрирования.

Литература

1. Systems of parallel processing: Transl. from engl./ Ed. by Ivens D. - M.: «Mir», 1985-416 p.
2. Feldman L.P., Dmitrieva O.A. Effective methods of multisequencing of caushi problem's numeral decision for ordinary differential equations // Mathematical modeling 2001, v.13, № 7, 2001. – P.66-72.
3. Feldman L.P., Dmitrieva O.A., Gerber S.: Abbildung der blockartigen Algorithmen auf Parallelrechnerarchitekture. In: Tavangarian, D., Grützner, R. (Hrsg.): Tagungsband 15. ASIM-Symposium Simulationstechnik in Rostock, September 2002, SCS-Europe BVBA, Ghent/Belgium 2002, S.359-364
4. Feldman L.P. Convergence and error estimation of parallel one-step block methods of simulation of dynamic lumped-parameters systems.// Scientific Works of Donetsk State Technical University. Issue: Series: Modeling Problems and designing automations of dynamic systems, v. 15: - Donetsk: 2000, P. 34-39.
5. Samarsky A.A., Gulin A.V. Numerical methods. - M.: Head. red. phis.-matem. liter., 1989.-432 p.
6. Feldmann L.P. Implementierung und Effizienzanalyse von parallelen blockartigen Simulationsalgorithmen für dynamische Systeme mit konzentrierten Parametern. In: Möller, D.P.F. (Hrsg.): Tagungsband 14. ASIM-Symposium Simulationstechnik in Hamburg, September 2000, SCS-Europe BVBA, Ghent/Belgium 2000, S. 241-246.
7. Dmitrieva O.A. The analysis of parallel algorithms for the numerical systems decision of ordinary differential equations systems by Adams-Bachfort-methods and Adams-Moulton-methods.// Mathematical modeling 2000, v. 12, № 5, P. 81-86.
8. Dmitrieva O.A. Increase of calculation effectiveness of starting piece for block multysteps task decision methods Cauchy on SIMD structures. // Scientific Works of Donetsk State Technical University. Issue: Series: Modeling Problems and designing automations of dynamic systems, v. 29: - Donetsk: 2001, P. 121-128.
9. Kahaner D., Moler C., Nash S. Numerical methods and Software: Transl. with engl. - M.: «Mir», 1998.- 575 p.

Поступила в редакцию 20.12.2003