

О.А. Дмитриева, к.т.н., доц.
Донецкий национальный технический университет
dmitriv@r5.dgtu.donetsk.ua

Организация параллельных вычислений при моделировании динамических объектов с автоматическим выбором шага и порядка

Предложены параллельные алгоритмы моделирования динамических объектов, ориентированные на автоматический выбор оптимального размера шага и порядка в каждой точке сетки. Разработанные алгоритмы базируются на явных и неявных экстраполяционных методах и ориентированы на минимизацию вычислительной работы за единичный шаг. При использовании неявных методов экстраполяции рекурсивное оценивание коэффициента главного члена глобальной ошибки интегрирования осуществлялось параллельно с проведением итерирования системы нелинейных алгебраических уравнений. Такая организация вычислительного процесса позволила значительно сократить число дополнительных арифметических действий. Параллельная реализация ориентирована на мультиосновные машины и кластерную вычислительную систему типа MIMD.

Ключевые слова: параллельное моделирование, стадийный метод, явная и неявная экстраполяция, адаптация шага, кластерная архитектура.

Введение

Применение параллельных вычислительных систем для моделирования динамических объектов с сосредоточенными параметрами является одним из основных направлений развития современной компьютерной индустрии [1]. Наибольшая необходимость в эффективных параллельных вычислениях возникает при моделировании динамических процессов, описываемых системами обыкновенных дифференциальных уравнений. При решении задач этого класса использование высокопроизводительной вычислительной техники не только оправдано, но и необходимо. Об этом свидетельствует знаменитый список проблем «большой вызов» [2], в котором такие задачи занимают одно из ведущих мест.

Эффективность численных методов моделирования динамических систем может быть значительно повышена за счет автоматического изменения шага интегрирования [3]. При этом актуальными являются вопросы соотношения погрешностей результатов и времени, затрачиваемого на получение решения. Изменяя размер шага, можно на порядки увеличивать эффективность без существенной потери точности [4-5]. Особую актуальность алгоритмы изменения шага интегрирования приобретают при численном решении жестких дифференциальных уравнений, поскольку, с одной стороны, необходимо обеспечить длину шага, достаточно малую для достижения требуемой точности в пограничном слое, а, с другой стороны, нужно гарантировать

достаточно большую длину шага во избежание бесполезной вычислительной работы на участках медленного изменения искомой функции [6-8].

В настоящее время к наиболее перспективным алгоритмам управления шагом относят экстраполяционные методы [4-5], которые позволяют не только управлять локальной ошибкой, но и существенно сократить время интегрирования за счет изменения порядка метода и размера шага, т.е. они способны учитывать особенности поведения решения задачи в автоматическом режиме. [3-5, 9-11].

1. Генерация параллельных схем

Рассмотрим решение задачи Коши

$$x' = f(t, x(t)), \quad x(t_0) = x_0, \quad (1)$$

Введем на отрезке $[t_0, t_0 + T]$ произвольную неравномерную сетку $\omega_n = \{t_{n+1} = t_n + \tau_n, n = 0, 1, \dots, N - 1, t_N = t_0 + T\}$. Задав базовой длиной шага τ_n , выберем монотонно возрастающую последовательность натуральных чисел

$$k_1 < k_2 < \dots < k_m \quad (2)$$

и определим соответствующие длины шагов

$$\tau_n^1 > \tau_n^2 > \dots > \tau_n^m,$$

где $\tau_n^i = \frac{\tau_n}{k_i}, i = 1, 2, \dots, m$.

Выберем s стадийный численный метод порядка p

$$x_{n+1} = x_n + \tau_n * \sum_{i=1}^s b_i q_i ;$$

$$q_i = f(t_n + c_i \tau_n, g_i);$$

$$g_i = x_n + \tau_n * \sum_{j=1}^s a_{ij} q_j, \quad i = 1, 2, \dots, s.$$

и, выполняя n_i шагов длины τ_n^i , вычислим приближенное решение задачи Коши (1), которое обозначим через

$$u_{\tau_n^i}(t_n + \tau_n) = u_{i,1}.$$

Поскольку используется численный метод порядка p , то разложение в ряд глобальной погрешности имеет вид:

$$u_{\tau_n^i}(t_n + \tau_n) - x(t_n + \tau_n) = a_1(\tau_n^i)^p +$$

$$a_2(\tau_n^i)^{p+1} + \dots + a_m(\tau_n^i)^{p+m-1} + O((\tau_n^i)^{p+m}),$$

где $x(t_n + \tau_n) = x(t_{n+1})$ - значение точного решения в узле $t_n + \tau_n$.

Проведя вычисления $u_{i,1}$ для ряда последовательных значений $i = 1, 2, \dots, m+1$, получим систему уравнений

$$x_{n+1}^* + a_1(\tau_n^i)^p + a_2(\tau_n^i)^{p+1} + \dots + a_m(\tau_n^i)^{p+m-1} = u_{i,1}, \quad i = 1, 2, \dots, m+1$$

с неизвестными $x_{n+1}^*, a_1, a_2, \dots, a_m$. Решив полученную систему уравнений, матрица которой невырождена, найдем уточненное значение x_{n+1}^* приближенного решения уравнения (1), погрешность которого имеет порядок $O((\tau_n)^{p+m})$, т. е.

$$x_{n+1}^* - x(t_n + \tau_n) = O((\tau_n)^{p+m}).$$

Упростим получение приближенного решения задачи (1). Построим по вычисленным стадийным методом значениям $u_{i,1}$, $i = 1, 2, \dots, m$ интерполяционный многочлен вида

$$L(\tau_n) = a_0 + a_1 \tau_n + a_2 \tau_n^2 + \dots + a_{m-1} \tau_n^{m-1}, \quad (3)$$

для которого $L(\tau_n^i) = u_{i,1}$ и $L(0) = a_0$ есть уточненное значение x_{n+1}^* приближенного решения задачи (1), а погрешность имеет порядок $O((\tau_n)^m)$. Таким образом, для нахождения уточненного значения x_{n+1}^* надо вычислить значение интерполяционного многочлена (3) при $\tau_n = 0$. Вначале построим многочлен по двум точкам $(\tau_n^1, u_{1,1})$ и $(\tau_n^2, u_{2,1})$

$$L_{12}(\tau_n) = \frac{1}{\tau_n^2 - \tau_n^1} \begin{bmatrix} u_{1,1} & \tau_n^1 - \tau_n \\ u_{2,1} & \tau_n^2 - \tau_n \end{bmatrix}$$

и найдем значение $L_{12}(0)$, обозначив его через $u_{2,2}$,

$$L_{12}(0) = u_{2,2} = u_{2,1} + \frac{u_{2,1} - u_{1,1}}{\tau_n^1/\tau_n^2 - 1},$$

которое дает уточнение приближенного решения с порядком $O((\tau_n)^2)$. Аналогично

$$L_{i,i+1}(\tau_n) = \frac{1}{\tau_n^{i+1} - \tau_n^i} \begin{bmatrix} u_{i,1} & \tau_n^i - \tau_n \\ u_{i+1,1} & \tau_n^{i+1} - \tau_n \end{bmatrix}$$

и получим при $\tau_n = 0$

$$L_{i,i+1}(0) = u_{i+1,2} = u_{i+1,1} + \frac{u_{i+1,1} - u_{i,1}}{\tau_n^i/\tau_n^{i+1} - 1}.$$

Затем по схеме Эйткена построим многочлен по трем точкам $(\tau_n^1, u_{1,1})$, $(\tau_n^2, u_{2,1})$, $(\tau_n^3, u_{3,1})$

$$L_{123}(\tau_n) = \frac{1}{\tau_n^3 - \tau_n^1} \begin{bmatrix} L_{12}(\tau_n) & \tau_n^1 - \tau_n \\ L_{23}(\tau_n) & \tau_n^3 - \tau_n \end{bmatrix}$$

и при $\tau_n = 0$

$$L_{123}(0) = u_{3,3} = u_{3,2} + \frac{u_{3,2} - u_{2,2}}{\tau_n^1/\tau_n^3 - 1}.$$

Продолжая так рассуждать и далее, придем к рекуррентной формуле последовательных уточнений приближенного решения

$$u_{i,l+1} = u_{i,l} + \frac{u_{i,l} - u_{i-1,l}}{\frac{\tau_n^{i-l}}{\tau_n^i} - 1},$$

$$i = 2, 3, \dots, m, \quad l = 1, 2, \dots, i - 1.$$

При проведении экстраполяции в качестве наиболее экономичных использовались последовательности Ромберга, Булирша гармоническая. Такие последовательности при интегрировании задачи (1) позволяют повторно использовать значения функции для меньших τ_n^i .

2. Оптимизация вычислительной работы

Практически все известные в настоящее время численные методы с автоматическим выбором шага интегрирования основаны на вычислении главного члена локальной ошибки и последующем выборе такого размера для очередного шага, который является максимальным для заданного предела локальной ошибки [5]. Решение первой задачи, т.е. нахождение главного члена локальной погрешности, в нашем случае может быть легко осуществлено с помощью оценки ошибки

$$\Delta_i = \|u_{i,i} - u_{i,i-1}\|, \quad (4)$$

которая получается после каждого экстраполяционного подхода. Кроме того, само число таких экстраполяционных подходов может регламентироваться получаемыми значениями (4) и варьироваться от шага к шагу.

Основываясь на концепции минимизации вычислительной работы за единичный шаг, в качестве меры вычислительной работы возьмем отношение числа операций к длине шага [3]

$$W_n = \frac{A_{m,n}}{\tau_n},$$

где $A_n = A_{m,n-1} + k_n$,

$$A_1 = k_1 + 1.$$

Оптимальный порядок экстраполяционного метода в данной точке численного интегрирования должен минимизировать вычислительную работу на шаге, под которой будем понимать число вычислений правой части задачи (1). Предполагается, что такой алгоритм будет находить приближенное

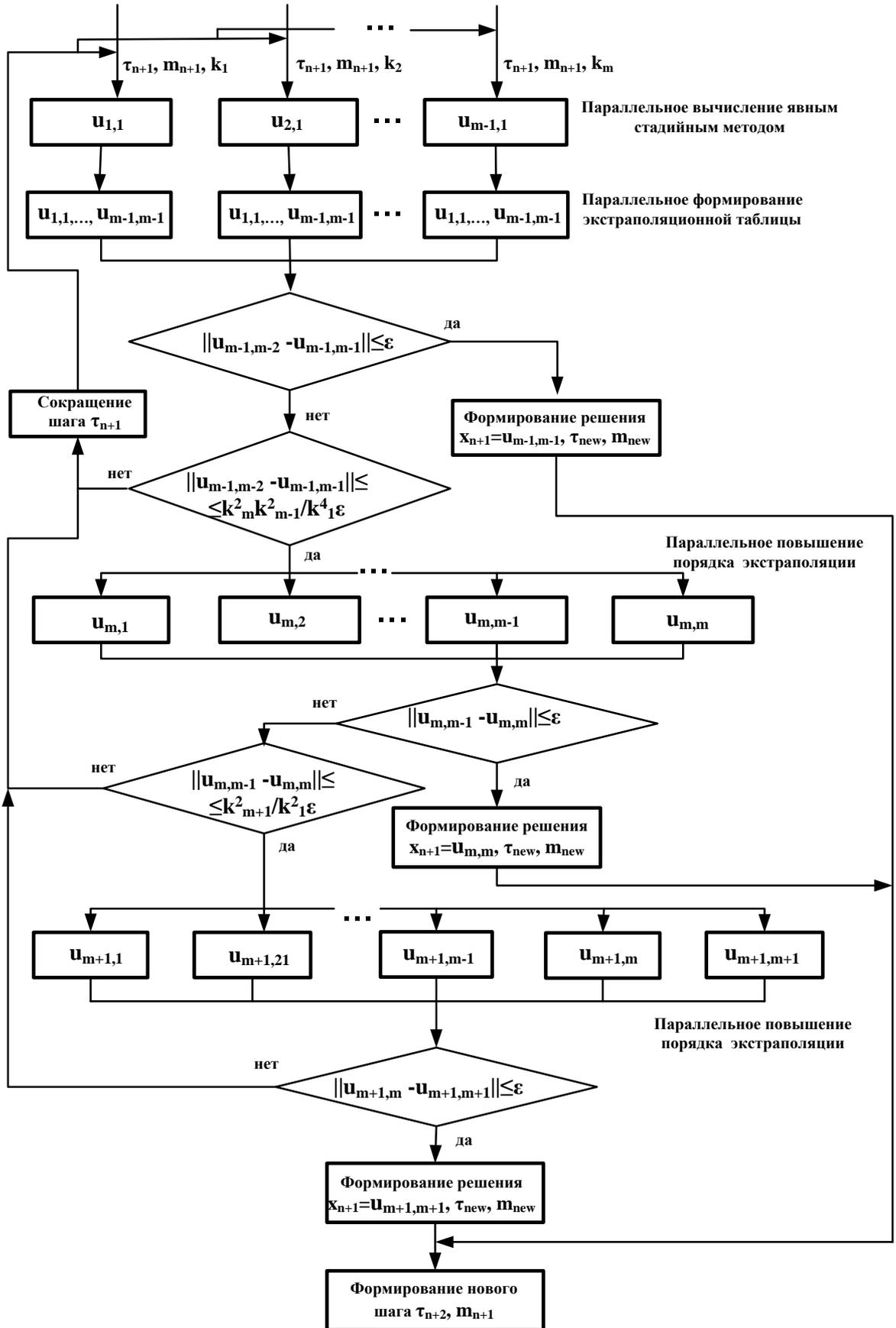


Рисунок 1 - Параллельный алгоритм управления шагом и порядком на основе явных стадийных методов

решение для фиксированного допуска на локальную ошибку за минимальное время.

Управление размером шага основано на том же принципе, что и во вложенных методах [12], когда оптимальный шаг выбирается как максимальный, обеспечивающий с учетом гарантийных факторов локальную точность.

3 Параллельные алгоритмы управления шагом на экстраполяционных схемах

Выбор первоначального шага осуществляется из следующих соображений. Предположим, что величина шага зависит от задаваемой точности ε , правых частей интегрируемых функций $f(t, x(t))$, порядка стадийного метода интегрирования p , начальных условий и границ интервала $t_0, t_N = t_0 + T$. В работе за основу принят следующий алгоритм выбора начального шага [3]. Вычисляются правые части (1) $f(t_0, x(t_0))$ в начальной точке. Принимая во внимание границы временного интервала, точность использованного метода и норму оценки правых частей, вводится параметр

$$par = \left(\frac{1}{\max(|t_0|, |t_N|)} \right)^{p+1} + \|f\|^{p+1}.$$

Начальное приближение к оценке шага производится как

$$\tau_0^1 = \left(\frac{\varepsilon}{par} \right)^{1/(p+1)}$$

В некоторых задачах начальные условия могут привести к нулевым компонентам $f(t_0, x(t_0))$, поэтому имеет смысл уточнить первичное полученное значение τ_0^1 . Выполняется один шаг стадийным методом с длиной шага τ_0^1 и определяются $f(t_0 + \tau_0^1, x(t_0 + \tau_0^1))$. Параметр par пересчитывается с корректировкой нормы $f(t_0 + \tau_0^1, x(t_0 + \tau_0^1))$, на основе чего определяется еще одно значение шага τ_0^2 . Начальный шаг определяется как

$$\tau_0 = \min(\tau_0^1, \tau_0^2).$$

Первичные экстраполяционные таблицы, значения в которых получены параллельно s -стадийным методом с шагами $\tau_n^1, \tau_n^2, \dots, \tau_n^i$

$$\begin{pmatrix} u_{1,1} & 0 & 0 & \dots & 0 \\ u_{2,1} & 0 & 0 & \dots & 0 \\ u_{3,1} & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ u_{i,1} & 0 & 0 & \dots & 0 \end{pmatrix}$$

для вычисленных i строк, преобразуются к виду

$$\begin{pmatrix} u_{1,1} & 0 & 0 & \dots & 0 \\ u_{2,1} & u_{2,2} & 0 & \dots & 0 \\ u_{3,1} & u_{3,2} & u_{3,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ u_{i,1} & u_{i,2} & u_{i,3} & \dots & u_{i,i} \end{pmatrix},$$

где $u_{i,i}$ соответствует уточнению приближенного решения с наивысшим порядком i . Количество

таких таблиц совпадает с размерностью системы, и значения в этих таблицах могут быть получены параллельно без обменов на стадиях, если для экстраполяции используется явный стадийный метод число процессоров в системе не меньше, чем количество требуемых экстраполяционных просчетов i . Схема параллельной реализации алгоритма управления шагом, основанного на явных многостадийных методах приведена на рис. 1.

Для выбора оптимального порядка необходимо иметь возможность оценить вычислительную работу, затрачиваемую на вычисление $u_{i,i}$. Она может измеряться количеством вычислений правой части уравнения. Большое число вычислений функции может быть компенсировано большей длиной шага τ_n .

Расчет значений для новой точки $x_{n+1}, n = 0, 1, 2, \dots$ начинается с параллельного формирования начального приближения к решению с помощью s -стадийного метода порядка p . При этом начальный шаг уже сформирован. Параллельный счет без обменов запускается на $m+1$ процессоре для последовательности значений k_1, k_2, \dots, k_{m+1} . Понятно, что с увеличением числа разбиений интервала увеличивается и время поиска решения, но для формирования первичной экстраполяционной таблицы достаточно дождаться получения результатов от процессоров, которые выполнили просчеты с шагами $\tau_n^1, \tau_n^2, \dots, \tau_n^{m-1}$, два оставшихся процессора, которые формируют решение с шагами τ_n^m и τ_n^{m+1} будут продолжать работу до получения результатов.

В сформированной первичной экстраполяционной таблице проверяется сходимость решения в строке $m-1$. Если $\|u_{m-1,m-2} - u_{m-1,m-1}\| \leq \varepsilon$, то результат формируется на основе уже полученных значений $u_{m-1,m-1}$, и результаты вычислений с шагами τ_n^m и τ_n^{m+1} не потребуются. Рассчитывается новое значение шага τ_{new} и новый размер экстраполяционной таблицы m_{new} .

Если $\|u_{m-1,m-2} - u_{m-1,m-1}\| > \varepsilon$, выполняется проверка на возможную сходимость в строке $m+1$. Если локальная ошибка превышает асимптотическую оценку

$$\|u_{m-1,m-2} - u_{m-1,m-1}\| > \left(\frac{k_{m+1}k_m}{k_1^2} \right)^2 \varepsilon, \quad (5)$$

текущий шаг τ_{n+1} отбрасывается, и значение τ_{n+1} пересчитывается заново. Если соотношение (5) не выполнилось, используется следующая, m -ая строка экстраполяционной таблицы, которая сформирована на основании уже имеющихся значений и просчета, выполненного с шагом τ_n^m . Если для рассчитанных значений

$$\|u_{m,m-1} - u_{m,m}\| \leq \varepsilon,$$

то результат формируется на основе значений $u_{m,m}$.

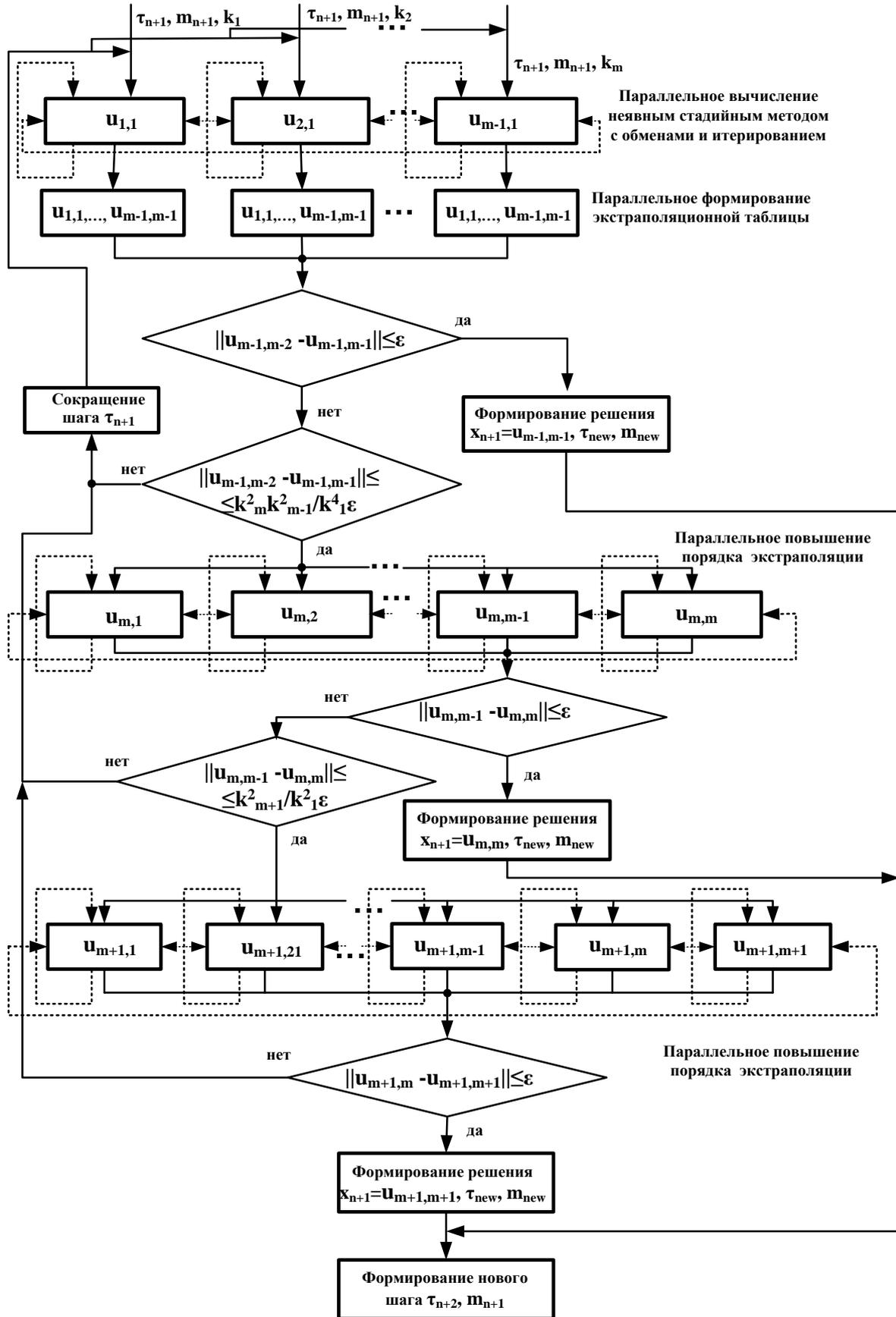


Рисунок 2 - Параллельный алгоритм управления шагом и порядком на основе неявных стадийных методов

Рассчитывается новое значение шага τ_{new} и новый размер экстраполяционной таблицы m_{new} . Если локальная ошибка превышает асимптотическую оценку

$$\|u_{m,m-1} - u_{m,m}\| > \left(\frac{k_{m+1}}{k_1}\right)^2 \varepsilon,$$

осуществляется еще одна попытка повысить порядок экстраполяции с сохранением текущего шага. Решение x_{n+1} формируется с текущим шагом τ_{n+1} на основе значений $u_{m+1,m+1}$, если $\|u_{m+1,m} - u_{m+1,m+1}\| \leq \varepsilon$, в противном случае происходит сокращение текущего шага и процесс повторяется с меньшим значением τ_{n+1} . После отбрасывания шага порядок экстраполяции не увеличивается. Количество просчетов на каждом шаге может быть существенно сокращено, если в качестве стадийного метода выбрать неявный вариант. Но при этом на каждой итерации при получении первичной таблицы возникает необходимость в обменах. С одной стороны, рост количества обменов приводит к увеличению времени реализации алгоритма поиска решения, но, как показывают эксперименты, заданная точность при этом достигается на меньших порядках экстраполяции и интегрирование ведется с большим шагом. Схема параллельной реализации алгоритма управления шагом, основанного на явных многостадийных методах приведена на рис. 2.

4 Параллельная реализация на основе стадийных схем

Для экспериментов на мультиосновных машинах использовался программный интерфейс MathLink среды параллельных вычислений в Mathematica Wolfram Research, обеспечивающий возможность параллельного поиска решений как на локальной машине, так и в сетевом масштабе. Также параллельная реализация осуществлялась на кластере NeClus MIMD-архитектуры с распределенной памятью. Кластер состоит из 93-х вычислительных узлов, узла управления (Front Node), системы коммутации в составе двух гигабитных Ethernet коммутаторов (Switch HP Procurve).

В качестве тестовой выбрана задача из [3]

$$\begin{aligned} x_1' &= 2t * \sqrt{x_2} * x_4, \\ x_2' &= 10t * e^{5*(x_3-1)} * x_4, \\ x_3' &= 2t * x_4, \\ x_4' &= -2t * \ln x_4 \end{aligned} \quad (6)$$

с начальными условиями

$$x_1(0) = x_2(0) = x_3(0) = x_4(0) = 1$$

и с известными точными решениями

$$\begin{aligned} x_1(t) &= e^{\sin(t^2)}, \\ x_2(t) &= e^{5\sin(t^2)}, \\ x_3(t) &= \sin(t^2) + 1, \\ x_4(t) &= \cos(t^2). \end{aligned}$$

Поиск решения проводился на интервале $[0, 2.5]$.

Точные решения использованы для вычисления глобальных ошибок численных решений задачи (6). Рассматривались реализации на основе стадийных методов, в качестве (2) использовалась гармоническая последовательность, а также последовательности Ромберга и Булирша. Результаты параллельного экстраполирования с использованием явных стадийных схем приведены в таблице 1.

Таблица 1. Результаты параллельного экстраполирования на явных методах с минимизацией вычислительной работы на единичном шаге

Допуск на локальную ошибку	Глобальная ошибка	Число принятых шагов	Число отброшенных шагов	Максимальный порядок экстраполяции	Время работы алгоритма, $c * 10^{-3}$
10^{-2}	10^{-2}	22	4	4	0.0624
10^{-3}	$4 * 10^{-4}$	27	5	7	0.078
10^{-4}	$1.5 * 10^{-5}$	59	4	8	0.1248
10^{-5}	$3.7 * 10^{-6}$	62	3	8	0.1908
10^{-6}	$1.4 * 10^{-7}$	71	5	10	0.3166
10^{-7}	$2.2 * 10^{-8}$	87	6	12	0.5732
10^{-8}	$3.1 * 10^{-9}$	96	6	14	0.6941

На графиках (рис. 3 - 6) показано поведение численного решения и распределения глобальных погрешностей при использовании неявных схем.

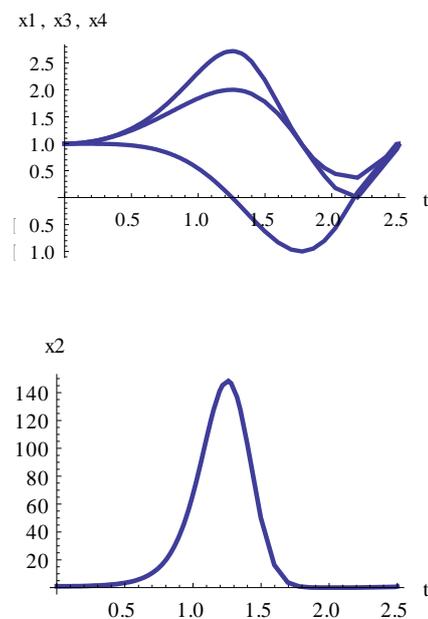


Рисунок 3 - Численное решение задачи (6)

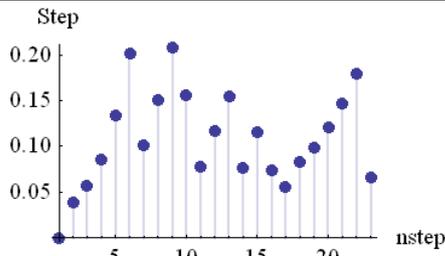


Рисунок 4 - Адаптация шага интегрирования

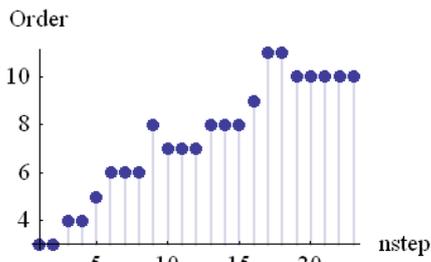
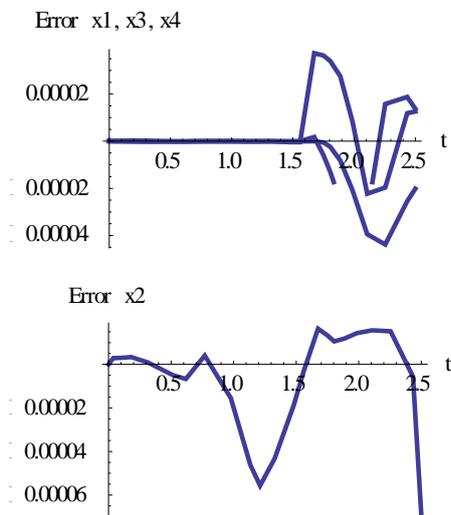


Рисунок 5 - Изменение порядка экстраполяции

Рисунок 6 - Распределение глобальных ошибок при численном решении задачи (6) с точностью 10^{-4}

Разделение компонент x_1 , x_3 , x_4 и x_2 выполнено из-за принципиально разного масштаба решения. Данные вычислительного эксперимента говорят о том, что автоматическое управление длиной шага и порядком параллельного экстраполяционного алгоритма, построенного на стадийных методах действительно работает на практике и позволяет решать задачу (1) за очень короткое время и с приемлемой точностью.

Выводы

В работе рассмотрены вопросы параллельного моделирования динамических объектов, которые описываются жесткими системами обыкновенных дифференциальных уравнений. Повышение эффективности достигается как за счет автоматического управления шагом интегрирования, так и за счет использования явных и неявных стадийных экстраполяционных схем переменного порядка. В качестве критерия выбора численной схемы используются неравенства для контроля точности и устойчивости. При решении жестких задач это позволяет на каждом шаге выбирать оптимальную с точки зрения вычислительных затрат численную схему. Параллельное управление шагом на основе явных и неявных экстраполяционных методов проводилось, основываясь на концепции минимизации вычислительной работы за единицу шага.

Список использованной литературы

1. Moulitsas, I., Karypis G. Architecture Aware Partitioning Algorithms// Algorithms and Architectures for Parallel Processing: Proceedings of 8th International Conference, ICA3PP 2008, Cyprus, June 9-11, 2008.— Berlin: Springer Verlag, 2008.— Vol. 5022/2008.— P. 42-53.
2. Grand Challenges: High performance computing and communications // A report by the Committee on Physical, Mathematical and Engineering Science, NSF/CISE, Washington, DC 20550, 2001.
3. Хайпер Э., Ваннер. Г. Решение обыкновенных дифференциальных уравнений. Жесткие задачи. - М.: Мир, 1999.- 685с.
4. Куликов Г. Ю., Хрусталева Е.Ю. Об автоматическом управлении размером шага и порядком в явных одношаговых экстраполяционных методах.// Журнал вычислительной математики и математической физики, 2008, Т. 48, № 8. - С. 1392-1405.
5. Куликов Г. Ю., Хрусталева Е.Ю. Об автоматическом управлении размером шага и порядком в неявных одношаговых экстраполяционных методах.// Журнал вычислительной математики и математической физики, 2008, Т. 48, № 9. - С. 1580-1606.
6. Butcher J. C. Numerical methods for ordinary differential equations - Wiley-VCH Verlag, 2003. - 418 p.
7. Дмитрієва О.А. Паралельні різницьеві методи розв'язання задачі Коші - Донецьк: ДонНТУ, 2011. 265 с.

8. Zanariah A. M., Suleiman M.B Solving Large Systems of Ordinary Differential Equations with Four Point Block Method on Parallel Computer// European Journ. of Scientific Research. -2009. - Vol. 29. - № 4. - P. 491-501.
9. Holsapple R., Iyer R., Doman D. Variable step-size selection methods for implicit integration schemes for ODEs// International journal of numerical analysis and modeling. Computing and Information. Vol. 4, Numb. 2, 2007. – P. 210-240.
10. Firsova A., Dmitrieva O., Dynamic System Simulation. Robust algorithms of state estimation of dynamic lumped parameters systems. – LAP Lambert Academic Publishing, 2011. – 92 p.
11. Soderlind G., WANG L. Adaptive time-stepping and computational stability. //J. Comp. and Appl. Math., Vol.185 Numb. 2, 2005. – P. 244-260.
12. Дмитриева О.А. Параллельное моделирование жестких систем на основе диагонализации полной матрицы.// Искусственный интеллект, - № 4. – 2011. - С. 46-53.

Надійшла до редколегії 30.03.2012

О.А ДМИТРИЄВА

Донецький національний технічний університет

Організація паралельних обчислень при моделюванні динамічних об'єктів з автоматичним вибором кроку і порядку

Запропоновано паралельні алгоритми моделювання динамічних об'єктів, орієнтовані на автоматичний вибір оптимального розміру кроку й порядку в кожній точці сітки. Розроблені алгоритми базуються на явних й неявних екстраполяційних методах і орієнтовані на мінімізацію обчислювальної роботи за одиничний крок. При використанні неявних методів екстраполяції рекурсивне оцінювання коефіцієнта головного члена глобальної помилки інтегрування здійснювалося паралельно із проведенням ітераційного процесу для системи нелінійних алгебраїчних рівнянь. Така організація обчислювального процесу дозволила значно скоротити число додаткових арифметичних дій. Паралельна реалізація орієнтована на мультиосновні машини і кластерну обчислювальну систему типу MIMD.

Ключові слова: паралельне моделювання, стадійний метод, явна й неявна екстраполяція, адаптація кроку, кластерна архітектура.

O.A. DMITRIEVA

Donetsk national technical university

Organization of parallel calculations when simulation of dynamic objects with automatic step and order selection

The parallel algorithms for simulation of dynamic objects, which are focused on automatic selection of the optimal step size and order in each grid point, are suggested. These algorithms are based on explicit and implicit extrapolation methods and focused on minimizing of the computer operations per unit step. When using the implicit extrapolation methods the recursive estimation of the coefficient of the leading term of the global integration error was carried out in parallel with iterating of a system of nonlinear algebraic equations. This organization of the computational process significantly reduced the number of additional arithmetic operations. The parallel implementation is focused on a cluster computing system, such as MIMD.

Keywords: parallel simulation, stepwise method, the explicit and implicit extrapolation, step adaptation, the cluster.