АЛГОРИТМИЧЕСКИЕ МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ МОДЕЛИРОВАНИЯ ЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Дмитриева О.А.
Кафедра ПМИ ДонНТУ
dmitriv@r5.dgtu.donetsk.ua

Abstract

Dmitrieva O. The algoritmical metods for effectiveness rising in linear dynamic systems. *In presented article explored possibilities multisequencings known consequent algorithm`s of numerical decision of the systems of the ordinary differential equations and transfer them on parallel computing structures to achieve maximum real productiveness. The offered approaches, allowing avoid the consequent area of the work multiprocessor computing SIMD systems. The revealed correlations between order of inaccuracy of the methods numerical integration and methods of interpolation, which allow to define the optimum choice of the method to parallel realization of the right parts. All got correlations equitable for linear systems of the common differential equations.*

Введение

Существует целый ряд обстоятельств, обуславливающих необходимость адаптации известных численных методов решения систем обыкновенных дифференциальных уравнений (ОДУ) в вычислительных системах с параллельной архитектурой. В первую очередь к таким обстоятельствам можно отнести тот факт, что более высокая скорость решения может достигаться за счет рациональной организации распараллеливания вычислений [1]. Эта задача становится особенно актуальной при реализации в параллельных средах систем большой размерности [2-3]. Причем рассматриваются подходы, основанные как на распараллеливании известных численных методов решения [4-5], так и ориентированные на создание новых методов [6-8].

Данная статья является продолжением работ [8-10], которые посвящены разработке и исследованию новых параллельных алгоритмов численного решения систем ОДУ, используемых для моделирования сложных динамических систем с сосредоточенными параметрами. Предлагаемые алгоритмы ориентированы на использование в многопроцессорных вычислительных системах SIMD (*single instruction stream - multiple data stream*) структуры с решеткой или линейкой процессорных элементов. Набор процессоров известен до начала вычислений и не меняется в процессе счета, при этом каждый процессорный элемент может выполнить любую арифметическую операцию за один такт, временные затраты, связанные с обращением к запоминающему устройству, отсутствуют.

1. Методы реализации, основанные на построении оператора перехода

Пусть математическую модель динамической системы можно представить в виде системы ОДУ с постоянными коэффициентами и начальными условиями

$$\frac{d\bar{x}}{dt} = A\bar{x} + \bar{f}(t), \quad \bar{x}(t_0) = \bar{x}^0 = (x_1^0, x_2^0, \ldots, x_m^0)^t, \tag{1}$$

где $\bar{x}$ - вектор неизвестных сигналов,

$\bar{f}(t)$ - вектор воздействий, $t \in [0,T]$,

A - матрица коэффициентов системы.

В этом случае решение можно получить последовательно по шагам с помощью численных методов заданного порядка точности.

Здесь вычисление значения вектора неизвестных $\bar{x}^{n+1}$ на очередном шаге требует предварительного определения значений $\bar{x}^n$. В [9] рассмотрены вопросы, связанные с возможностью параллельной реализации таких алгоритмов. В частности, если система (1) является однородной, т.е. $f_i(t) = 0$, $i=1,2,\ldots m$, тогда, в зависимости от выбранного метода интегрирования, можно искать решение в виде

$$\bar{x}^{n+1} = G\bar{x}^n, \tag{2}$$

где G - оператор (матрица) переходов.

Полученный оператор перехода G, который необходимо определить один раз до начала вычислений, позволяет вычислять значения вектора неизвестных параллельно [6,9-11]. Для методов Рунге-Кутты, например, этот оператор может быть представлен, в зависимости от точности метода, как

$$G = E + \tau A(E + \frac{\tau A}{2}(E + \frac{\tau A}{3}(E + \frac{\tau A}{4}))), \tag{3}$$

который обеспечивает точность 4-го порядка, или, например, как

$$G = E + \tau A(E + \frac{\tau A}{2}(E + \frac{\tau A}{3}(E + \frac{\tau A}{4}(E - \frac{\tau A}{5}(E - \frac{\tau A}{4}))))), \tag{4}$$

точность которого оценивается 6-ым порядком.

При решении неоднородной системы необходимо дополнительно вычислить на каждом шаге значения всех функций $f_i(t)$, $i=\overline{1,m}$ в нескольких промежуточных точках. Поскольку все эти функции могут быть различными, одновременное вычисление их на SIMD компьютере невозможно.

В связи с этим можно предложить два основных подхода, первый из которых состоит в том, что все промежуточные значения функций $\overline{f}_i(t)$ могут быть вычислены заранее. При этом, если количество промежуточных точек метода определяется как r, то можно оценить количество тактов расчета на топологических структурах линейке из m процессоров и решетке из $m \times m$ процессоров (размерность процессорного поля выбрана совпадающей с размерностью системы уравнений исключительно для удобства изложения). Определим для этого трудоемкость реализации правых частей системы как Θ_{f_i} и при расчете будем оперировать с максимальным значением, которое обозначим как $\Theta_f = \max_i \{\Theta_{f_i}\}$. Если расчет осуществляется для общего количества узлов, равного N, то общее число тактов работы на линейке процессоров составит для одного уравнения ближайшее целое сверху соотношения $\dfrac{r * \Theta_f * N}{m}$ или $\left[\dfrac{r * \Theta_f * N}{m}\right] + 1$. Тогда вся система может быть рассчитана за $r * \Theta_f * N + m$ тактов. На решетке процессоров одно уравнение будет решаться за ближайшее целое сверху к $\dfrac{r * \Theta_f * N}{m^2}$ тактов, а время, которое потребуется для расчета всей системы составит $\left[\dfrac{r * \Theta_f * N}{m}\right] + 1$. По каждому уравнению системы придется хранить двумерные массивы размерностью $r \times N$ и использовать коэффициенты с нужными индексами при расчете.

2. Методы реализации, основанные на интерполяции правой части

Второй подход, который позволит избежать последовательных участков при параллельной реализации системы, основывается на предварительном интерполировании правых частей (1). В вычислительной практике с таким подходом сталкиваются, если приходится заменять одну функцию $f(t)$ (известную, неизвестную или частично известную) некоторой функцией $\varphi(t)$, близкой к $f(t)$ и обладающей определенными свойствами, позволяющими производить над нею те или иные аналитические или вычислительные операции. Такую замену называют *приближением* функции $f(t)$. Тогда при решении задачи вместо функции $f(t)$ оперируют с функцией $\varphi(t)$, а задача построения функции $\varphi(t)$ называется задачей приближения. Исходя из проблематики задачи, т.е., принимая во внимание большое количество узлов, которые будут участвовать в расчете, и задаваясь требуемой точностью приближения функций, можно утверждать, что наиболее перспективным является случай, когда используются кусочно-полиномиальная аппроксимация, или сплайны, так как при этом интерполяционный многочлен строится не на весь интервал решения задачи, а на подинтервалах, что позволит избежать накопления ошибок приближения.

Основная идея такого подхода заключается в следующем: исходный отрезок решения для (1) $[0,T]$ разбивается на несколько подинтервалов V с шагом, определяющимся из соотношения точности методов численного интерполирования и интегрирования, а затем на каждом таком интервале строится интерполяционный многочлен. Поскольку в качестве интерполяционной функции обычно выбирают многочлены степени не выше 3 - 4-ой, что соответственным образом влияет на точность интерполяции, то необходимо предварительно согласовать порядки точности методов численного интегрирования и предварительного интерполирования.

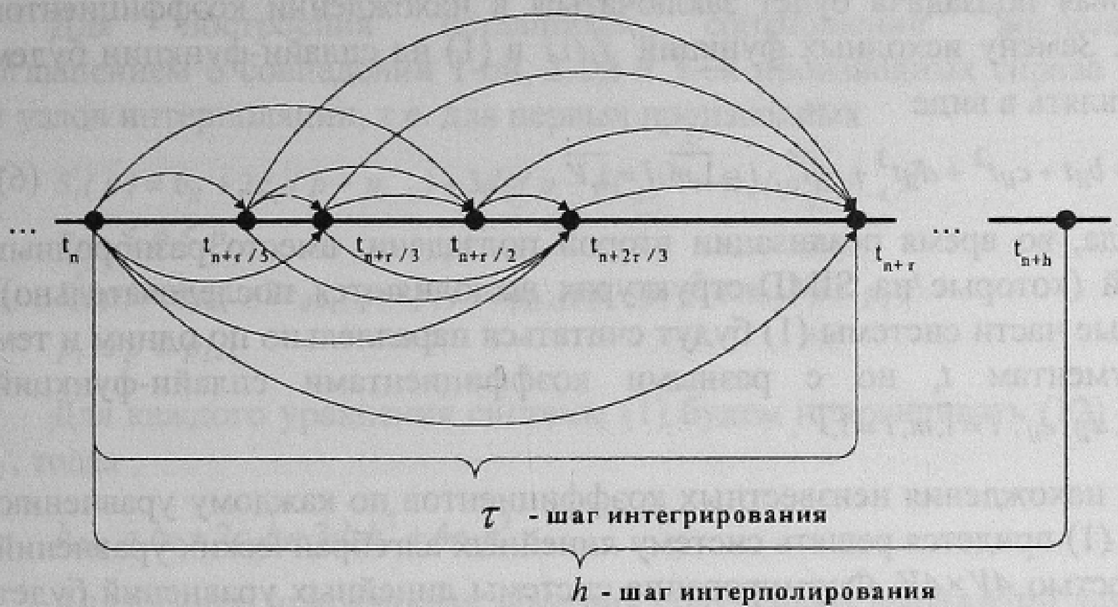

Рисунок 1 - Пример схемы расчета правых частей (1) с промежуточными точками

Если порядок метода численного интегрирования (1) определяется как $O(\tau^{\nu})$, а порядок сплайна как $O(h^4)$, то между шагами двух решающихся задач должно выполняться соотношение $\tau^{\nu}=h^4$. Если порядок точности метода интегрирования $\nu=4$ или более, т.е. между количеством узлов задач интерполирования и интегрирования выполняется соотношение $V\geq N$, то использование интерполирования для восстановления значений правых частей является нерациональным. Проще заранее вычислить значения правых частей на промежутке $[0,T]$. Если же речь идет о методах интегрирования, которые имеют порядок погрешности ниже 4, то тогда $V<N$ и при этом значения V и N можно связать с помощью некоторого коэффициента β, т.е.

$$N = \beta * V \text{, где } \beta >> 1. \qquad (5)$$

При этом желательно выбирать множитель β целым, т.к. предпочтительнее, чтобы на одном такте расчета использовались коэффициенты одного интервала сплайна, что значительно упростит алгоритм вычисления и выбор нужного интервала по заданному аргументу.

Если исходить из (5), то узлов интерполирования будет в β раз меньше, чем узлов интегрирования. К тому же оценку погрешности для сплайна порядка $O(h^4)$ можно считать завышенной. Тогда исходная задача может быть сведена к двум подзадачам, каждая из которых легко распараллеливается.

Первая подзадача будет заключаться в нахождении коэффициентов сплайна. Замену исходных функций $f_i(t)$ в (1) на сплайн-функции будем осуществлять в виде

$$a_{il} + b_{il}t + c_{il}t^2 + d_{il}t^3 + e_{il}t^4, \; i = \overline{1,m}, l = \overline{1,V}. \qquad (6)$$

Тогда, во время реализации второй подзадачи, вместо разнородных операций (которые на SIMD-структурах выполняются последовательно), все правые части системы (1) будут считаться параллельно по одним и тем же аргументам t, но с разными коэффициентами сплайн-функций $a_{il}, b_{il}, c_{il}, d_{il}, e_{il}, \; i = \overline{1,m}, l = \overline{1,V}$.

Для нахождения неизвестных коэффициентов по каждому уравнению системы (1) придется решать систему линейных алгебраических уравнений размерностью $4V \times 4V$. Формирование системы линейных уравнений будет исходить из принципов совпадения значений функции, ее первых, вторых и третьих производных на соседних подинтервалах слева и справа от узла интерполяции.

Пусть $\{p_l\}$ - множество узлов интерполяции, $l = \overline{0,V}$. Причем, для любой правой части системы (1) это множество узлов будет постоянным.

Между узлами p_{l-1} и p_l будем представлять функцию i- го уравнения системы в виде

$$S_i(p) = a_{il} + b_{il}(p - p_{l-1}) + c_{il}(p - p_{l-1})^2 + d_{il}(p - p_{l-1})^3 + e_{il}(p - p_{l-1})^4 \qquad (7)$$

$$p_{l-1} \le p \le p_l, \; i = \overline{1,m}, l = \overline{1,V}$$

Исходя из постановки задачи интерполирования, в узлах интерполяции значения исходной функции и интерполяционного многочлена должны совпадать, т.е.

$$S_i(p_l) = f_{il} \quad i = \overline{1,m},\, l = \overline{0,V} \tag{8}$$

С другой стороны, если аргумент сплайна совпадает с узлом интерполяции, тогда из (7) и (8)

$$S_i(p_{l-1}) = a_{il} = f_{il-1}, \quad i = \overline{1,m},\, l = \overline{1,V} \tag{9}$$

Поскольку интерполяционный многочлен строится для равностоящих узлов, т.е.

$$p_l - p_{l-1} = h, \tag{10}$$

то, используя полученные соотношения, можно переписать формулу для сплайна в l–ом узле в следующем виде

$$f_{il} = f_{il-1} + b_{il}h + c_{il}h^2 + d_{il}h^3 + e_{il}h^4 \tag{11}$$

Для построения оставшихся соотношений воспользуемся соглашением о совпадении 1-ой, 2-ой и 3-ей производных справа и слева от узлов интерполяции, т.е. для первых производных

$$\begin{aligned} &S_i'(p) = b_{il} + 2c_{il}(p - p_{l-1}) + 3d_{il}(p - p_{l-1})^2 + 4e_{il}(p - p_{l-1})^3 \\ &p_{l-1} \le p \le p_l \\ &S_i'(p) = b_{il+1} + 2c_{il+1}(p - p_l) + 3d_{il+1}(p - p_l)^2 + 4e_{il}(p - p_l)^3 \\ &p_l \le p \le p_{l+1} \end{aligned} \tag{12}$$

Для каждого уравнения системы (1) будем приравнивать (12) в точке p_l, тогда

$$b_{il+1} = b_{il} + 2c_{il}h + 3d_{il}h^2 + 4e_{il}h^3 \tag{13}$$

Аналогично приравнивая значения полученных выражений для 2-ой и 3-ей производных в узлах интерполяции и добавив граничные условия, получаем m систем линейных уравнений.

Сформированные системы относительно вектора неизвестных коэффициентов можно представить в общем виде как

$$Q_i \overline{y}_i = \overline{g}_i, \quad i = \overline{1,m} \tag{14}$$

Вектора неизвестных будут иметь вид

$$\overline{y}_i = (a_{i1}, b_{i1}, c_{i1}, d_{i1}, e_{i1}, \ldots, a_{iV}, b_{iV}, c_{iV}, d_{iV}, e_{iV}), \quad i = \overline{1,m}\,. \tag{15}$$

Особенностью полученных систем является ленточный вид матрицы Q, т.к. каждое уравнение системы (за исключением первого и последнего) будет содержать только 4 неизвестных. В этом случае систему можно преобразовать так, чтобы ее можно было решать методом встречной прогонки [12].

Трудоемкость решения таких систем на параллельных SIMD структурах линейно зависит от размерности решаемой системы и для системы размерностью k оценивается как $O(k)$. Тогда для нашего случая трудоемкость нахождения коэффициентов сплайн-функции для одного уравнения системы будет оцениваться на уровне $O(4V)$. Тогда для исходной задачи (1) число операций приближенно будет оцениваться на уровне $4*V*m$.

Еще один возможный подход к решению полученных систем (14), которые имеют большую размерность и разреженную матрицу коэффициентов, заключается в приведении ее к блочно-диагональной форме с обрамлением [13,14] и формировании вспомогательной системы значительно меньшей размерности, которая определит вектор определяющих величин, или переменных связи.

Трудоемкость реализации такого подхода на параллельных вычислительных структурах будет, как и в предыдущем случае, линейно зависеть от размерности системы.

Кроме того, для интерполирования необходимо предварительно вычислить значения правых частей в V точках, которые будут использоваться в качестве исходных данных для построения интерполяционного многочлена, тогда для системы из m уравнений потребуется $m*(4V+V*\Theta_f)$ тактов.

Также возникает необходимость в восстановлении значений правых частей по полученным коэффициентам интерполяционных многочленов в N основных и r вспомогательных узлах интегрирования. Последовательность выполняемых операций при этом составит для многочлена общего вида $a+bt+ct^2+dt^3+et^4$ на SIMD-структуре

умножения

1-ый такт – $t*t$

2-ой такт - t^2*t, t^2*t^2 - параллельно

3-ий такт - $b*t, c*t^2, d*t^3, e*t^4$ - параллельно

сложения

4-ый такт – $a+b*t, c*t^2+d*t^3$

5-ый такт - $a+b*t+c*t^2+d*t^3+e*t^4$

Таким образом, на определение одного значения правой части необходимо 5 временных тактов. Если учесть, что число точек, в которых необходимо восстанавливать значение правой части каждого уравнения определяется как $r*N$, то всего на восстановление значений функции по интерполяционному многочлену для системы потребуется $5*m*r*N$ временных тактов.

Сведем полученные приближенные результаты для 2-х описанных способов реализации правых частей в следующую таблицу

Таблица 1

	Предварительный расчет	Интерполирование
Общее число операций	$r*\Theta_f*N*m$	$m*(4V+V*\Theta_f+5*r*N)$
Число операций на линейке процессоров	$\lfloor r*\Theta_f*N \rfloor+1$	$4V+V*\Theta_f+5*r*N$
Число операций на решетке процессоров	$\left\lceil \frac{r*\Theta_f*N}{m} \right\rceil+1$	$(4V+V*\Theta_f+5*r*N)/m$

Поскольку изначально предполагалось, что трудоемкости вычисления правых частей Θ_f являются высокими [6], то оценка трудоемкости всего алгоритма может осуществляться относительно этих значений. Тогда очевидно, что в случае выполнения соотношения (5), предпочтительнее интерполировать правые части, хотя этот подход и сопряжен с алгоритмическими сложностями, но имеет безусловные преимущества.

Заключение

Таким образом, в представляемой работе исследованы возможности распараллеливания известных последовательных алгоритмов численного решения систем обыкновенных дифференциальных уравнений и переноса их на параллельные вычислительные структуры с целью получения максимальной реальной производительности.

Предложены подходы, позволяющие избегать последовательных участков работы многопроцессорных вычислительных SIMD систем, один из этих подходов основан на предварительном вычислении правых частей неоднородной линейной системы ОДУ, который предпочтительнее использовать, если точность метода интегрирования, с помощью которого решается задача, высока.

Разработан также подход, исключающий последовательные вычисления, который основывается на предварительном интерполировании правых частей системы (1) с помощью сплайнов.

Выявлены соотношения между порядками погрешностей методов численного интегрирования системы (1) и методами интерполирования, которые позволяют определить оптимальный выбор метода параллельной реализации правых частей. Все полученные соотношения справедливы, как отмечалось ранее, для линейных систем обыкновенных дифференциальных уравнений. Поэтому в качестве перспективной направленности данной работы необходимо определить возможность такого распараллеливания и для нелинейных систем ОДУ.

Литература

1. Tisseur F., Dongarra J. Parallelizing the Divide and Conquer Algorithm for the Symmetric Tridiagonal Eigenvalue Problem on Distributed Memory Architectures.// SIAM Journal on Scientific Computing. 1999. Vol. 6, № 20, P. 2223-2236.
2. J. Ortega, R. Gasca, Miguel Toro: Including Qualitative Knowledge in Semiqualitative Dynamical Systems. IEA/AIE, 1998. Vol. 1, P. 329-335.
3. R. Wismuller, M. Oberhuber, J. Krammer, and O. Hansen: Interactive debugging and performance analysis of massively parallel applications. Parallel Computing 22(3), 1996. №3, P. 415-442.
4. Henri Casanova, Michael Thomason, and Jack Dongarra. Stochastic Performance Prediction for Iterative Algorithms in Distributed Environments Journal of Parallel and Distributed Computing. 1999. Vol. 98, № 1, P. 68-91.
5. Дмитриева О.А. Анализ параллельных алгоритмов численного решения систем обыкновенных дифференциальных уравнений методами Адамса-Башфорта и Адамса-Моултона// Математическое моделирование, том 12, № 5, 2000. - С. 81-86
6. Фельдман Л.П., Дмитриева О.А. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений. //Математическое моделирование, том 13, № 7, 2001. – С.66-72.
7. Feldman L., Dmitriewa O.A., Gerber S. Abbildung der blockartigen Algoritmen auf die Parallelrechnerarchitekture /Simulationstechnik, 17.

Symposium in Magdeburg, Sept. 2003: SCS-Europe BVBA (ISBN 3-936150-27-3), Magdeburg, Germany, 2003.- p. 359-364.

8. Feldman L., Svjatnyj V., Dmitriewa O.A. Stabilität von parallelen Simulationsverfahren fur dynamische Systeme mit konzentrierten Parametern /Simulationstechnik, 17. Symposium in Magdeburg, Sept. 2003: SCS-Europe BVBA (ISBN 3-936150-27-3), Magdeburg, Germany, 2003.- p. 105-110.
9. Дмитриева О.А. Проблемы распараллеливания многошаговых алгоритмов типа Адамса. // Научные труды Донецкого национального технического университета. Выпуск 70. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003) – Донецк: ДонНТУ, 2003. С.128-137.
10. Дмитриева О.А. Отображение высокоточных многошаговых алгоритмов на параллельные вычислительные структуры с топологией гиперкуб.// Вісник Східноукраїнського національного університету імені Володимира Даля. №12 (82) , 2004 р. С. 79-86.
11. Дмитриева О.А. Параллельное моделирование динамических объектов с сосредоточенными параметрами. Тезисы докладов XII Юбилейной международной конференции по вычислительной механике и современным прикладным программным средствам. - М.:МГИУ, 2003.
12. Гаранжа В.А., Коньшин В.Н. Прикладные аспекты параллельных высокоточных алгоритмов решения задач вычислительной гидродинамики.// Тезисы докладов Всероссийской научной конференции «Фундаментальные и прикладные аспекты разработки больших распределенных программных комплексов». – М.: Изд-во МГУ. – 1998. – С. 34-38.
13. Джорт А., Лю Д. Численное решение больших разреженных систем уравнений. – М.: Мир, 1984. – 333 с.
14. Дмитриева О.А. Анализ параллельных алгоритмов численного решения систем линейных уравнений итерационными методами. //Научн. тр. ДонГТУ. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, випуск 10: - Донецк:, 2000. - С. 15-22.

Дата надходження до редакції 05.06.2005 р.