

УЧЕТ НЕОДНОРОДНОСТИ ПРИ ПАРАЛЛЕЛЬНОМ МОДЕЛИРОВАНИИ ДИНАМИЧЕСКИХ СИСТЕМ

Дмитриева О.А.

Донецкий национальный технический университет

Запропоновано підходи, що дозволяють уникати послідовних ділянок роботи багатопроцесорних обчислювальних систем. Перший підхід заснований на попереднім обчисленні правих частин неоднорідної лінійної системи звичайних диференціальних рівнянь, другий - на попередній інтерполяції правих частин системи за допомогою сплайнів. Виявлено співвідношення між порядками похибок методів чисельного інтегрування і методами інтерполяції.

Данная статья посвящена разработке и исследованию параллельных алгоритмов численного решения систем ОДУ, используемых для моделирования сложных динамических систем с сосредоточенными параметрами. Предлагаемые алгоритмы ориентированы на использование в многопроцессорных вычислительных системах типа SIMD с решеткой или линейкой процессорных элементов. Набор процессоров известен до начала вычислений и не меняется в процессе счета. Пусть математическая модель динамической объекта представлена в виде системы с постоянными коэффициентами и начальными условиями

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

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.

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 $\frac{r * \Theta_f * N}{m}$ или $\left[\frac{r * \Theta_f * N}{m} \right] + 1$. 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 $\frac{r * \Theta_f * N}{m^2}$ тактов, а время, которое потребуется для расчета всей системы составит $\left[\frac{r * \Theta_f * N}{m} \right] + 1$. По каждому уравнению системы придется хранить двумерные массивы размерностью $r \times N$ и использовать коэффициенты с нужными индексами при расчете.

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 подынтервалах, что позволит избежать накопления ошибок приближения.

Основная идея такого подхода заключается в следующем: исходный отрезок решения для (1) $[0, T]$ разбивается на несколько подынтервалов 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.

Если порядок метода численного интегрирования (1) определяется как $O(\tau^v)$, а порядок сплайна как $O(h^4)$, то между шагами двух решаемых задач должно выполняться соотношение $\tau^v = h^4$. Если порядок точности метода интегрирования $v=4$ 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.

$$N = \beta * V, \text{ где } \beta \gg 1. \quad (2)$$

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 (2), 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

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

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 подынтервалах слева и справа от узла интерполяции.

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. 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 приблизительно будет оцениваться на уровне $4*V*m$.

Еще один возможный подход к решению полученных систем, которые имеют большую размерность и разреженную матрицу коэффициентов, заключается в приведении ее к блочно-диагональной форме с обрамлением 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. 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 $r \cdot N$, то всего на восстановление значений функции по интерполяционному многочлену для системы потребуется $5 \cdot m \cdot r \cdot N$ 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	$r \cdot \Theta_f \cdot N \cdot m$	$m \cdot (4V + V \cdot \Theta_f + 5 \cdot r \cdot N)$
1D-тор	$\lceil r \cdot \Theta_f \cdot N \rceil + 1$	$4V + V \cdot \Theta_f + 5 \cdot r \cdot N$
2D-тор	$\left\lceil \frac{r \cdot \Theta_f \cdot N}{m} \right\rceil + 1$	$(4V + V \cdot \Theta_f + 5 \cdot r \cdot N) / m$

Поскольку изначально предполагалось, что трудоемкости вычисления правых частей Θ_f являются высокими, то оценка трудоемкости всего алгоритма может осуществляться относительно этих значений. Тогда очевидно, что в случае выполнения соотношения (2), предпочтительнее интерполировать правые части, хотя этот подход и сопряжен с алгоритмическими сложностями, но имеет безусловные преимущества.

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.