

EDUCATIONAL AND PEDAGOGICAL STUDIES

ПЕДАГОГИЧЕСКИЕ ИССЛЕДОВАНИЯ

УДК 378.147

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ЧИСЛЕННОГО РЕШЕНИЯ ОДНОЙ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Аглямзянова Г.Н., Гумерова Л.З.

В данной статье представлено численное решение математической модели, а именно модели мозга, представленной системой дифференциальных уравнений. Приближенные решения получены методами параллельного программирования.

Ключевые слова: модель мозга; параллельное программирование; система дифференциальных уравнений; численное решение

PARALLEL IMPLEMENTATION OF NUMERICAL SOLUTION OF ONE SYSTEM OF DIFFERENTIAL EQUATIONS

Aglyamzyanova G.N., Gumerova L.Z.

This article presents a numerical solution of a mathematical model, namely a brain model represented by a system of differential equations. Approximate solutions are obtained by parallel programming.

Keywords: brain model; parallel programming; system of differential equations; numerical solution

Введение

Большое количество работ ученых посвящено математическому моделированию нейродинамических систем. Интенсивное внедрение новых информационных технологий позволяет применять разные подходы к их численному решению. Для опи-

сания модели мозга будем использовать модель мозга типа Ходжина-Хаксли [2].

$$C_m \frac{dV_i}{dt} = I_l + I_{Na} + I_k + I_i^{ext} + I_i^{sin}, \quad i = 1, \dots, N \quad (1)$$

Здесь V_i – потенциал нейрона с номером i , C_m – параметр, определяющий емкость мембраны, I_l – ток утечки, I_{Na} – ток, обусловленный протеканием ионов натрия через мембрану, I_k – ток, обусловленный протеканием ионов калия через мембрану, I_i^{ext} – параметр, определяющий внешний ток, I_i^{sin} – синаптический ток, возникающий в результате взаимодействия между нейронами, N – общее число нейронов в сети. Ионные токи каждого отдельного нейрона записываются следующим образом:

$$\begin{aligned} I_{Na} &= m_i^2 h_i g_{Na} (V_{Na} - V_i) \\ I_k &= n_i^4 g_k (V_k - V_i) \\ I_l &= g_l (V_l - V_i) \end{aligned}$$

Здесь $V_{Na}, V_k, V_l, g_{Na}, g_l$ являются параметрами модели (константы), которые определяют равновесные потенциалы ионных каналов, а также их максимальные проводимости соответственно. Сложная динамика в модели возникает в результате того, что проводимости ионных каналов имеют нелинейную зависимость от трансмембранного потенциала. Для описания данного эффекта в модель введены так называемые воротные переменные m, h, n , изменение во времени которых описываются дифференциальными уравнениями первого порядка:

$$\begin{aligned} \frac{dm_i}{dt} &= a_m(V_i)(1 - m_i) - \beta_m(V_i)m_i \\ \frac{dh_i}{dt} &= a_h(V_i)(1 - h_i) - \beta_h(V_i)h_i \\ \frac{dn_i}{dt} &= a_n(V_i)(1 - n_i) - \beta_n(V_i)n_i \end{aligned} \quad (2)$$

Функции $a_m(V_i), a_h(V_i), a_n(V_i), \beta_m(V_i), \beta_h(V_i), \beta_n(V_i)$ являются нелинейными функциями, зависящими от потенциала V_i . Таким образом, в случае $I_i^{sin} = 0$ уравнения (1), (2) описывают динамику отдельного нейрона. Рассмотрим теперь взаимодействие между нейронами. Как уже говорилось, нейроны связываются друг с другом посредством

синаптических связей и передающихся по ним нервных импульсов. Формально в модели взаимодействие описывается с помощью слагаемого в правой части уравнения (1):

$$I_i^{syn} = \sum G_{ij} r_j(t) (V_{syn} - V_i), j \neq i, j = 1, \dots, N$$

Константа G_{ij} определяет максимальную проводимость синаптической связи от нейрона с номером j к нейрону с номером i , V_{syn} является параметром в модели, определяющим равновесный потенциал синаптической связи. Динамические переменные r_i описывают активацию синаптической связи в результате изменения мембранного потенциала V_i :

$$\frac{dr_i}{dt} = \alpha F(V_i)(1 - r_i) - \beta r_j$$

где $F(V_i) = \frac{1}{1 + e^{-\frac{10 - V_i}{z}}}$ – нелинейная активационная функция; α , β константы.

Таким образом, требуется найти решение задачи Коши для системы обыкновенных дифференциальных уравнений первого порядка, записанной в векторной форме следующим образом:

$$U' = F(X, U), U(x^0) = U^0 \quad (3)$$

Сам метод представлен следующим образом:

$$V^{j+1} = V^j + hF(x^j, V^j) \quad (4)$$

где верхним индексом обозначен номер итерации метода, вектор V_i является приближенным значением точного решения $U(x^j)$, h – шаг метода.

Материалы и методы исследования

Для распараллеливания программы используется библиотека `openmp`, позволяющая несложным способом распараллелить отдельные циклы, т.к. большинство параллельных вычислений производятся в циклах.

Наибольшее время работы программы тратится на непосредственно само вычисление системы дифференциальных уравнений, а именно каждой итерации метода Эйлера для каждого нейрона. В программном коде данное вычисление производится в функции `Solve`. В начале выполнения функции задается количество процессов для `openmp` с помощью метода:

```
omp_set_num_threads(numOfThreads);
```

Далее распараллеливается цикл по нейронам, заключив цикл в отдельный блок для параллельных вычислений `#pragma omp parallel { }` и добавив перед циклом следующую директиву:

```
#pragma omp for schedule(guided, 10)
```

Гонок данных в программе нет – все изменяемые переменные меняются потоками независимо. Однако результаты на выходе программы получаются некорректными. Проблема состоит в том, что гонка данных образуется в генераторе псевдослучайных чисел. В программе генератор реализован при помощи функции `rand()`. Начало последовательности ПСЧ определяется вызовом функции `srand(seed)`. Псевдослучайное число хранится в глобальной переменной. Следующее псевдослучайное число создается на основе предыдущего, поэтому в многопоточной программе это немедленно приведет к гонке данных. Чтобы этого избежать, функция `rand()` реализуется одним потоком. Для этого генерацию случайных чисел определим в отдельной функции вне параллельного процесса:

```
Void GenIExt()  
{  
    inti;  
    for (i = 0; i < N; i++)  
        I_ext[i] = -10.0 - (double)rand() / ((double)RAND_MAX);  
}
```

А вызов данной функции будет производиться непосредственно перед параллельным процессом, а именно перед циклом по нейронам.

Результаты исследования

Протестируем полученную распараллеленную программу. Выполним измерения времени выполнения приложения для набора нейронов: $N = 50, 500, 1000, 2000$. Количество итераций (шагов) по времени примем равным 20000. Также для эксперимента проведем вычисления времени (сек) для разного числа потоков, равного 1, 2, 4. Количество нитей задается пользователем в консоли при запуске программы. Результаты представлены в таблице 1.

Таблица 1.

Измерение времени параллельной программы

N	Нити		
	1	2	4
50	0,142	0,206	0,679
500	6,643	3,778	2,376
1000	25,985	14,192	8,868
2000	118,056	54,876	46,646

На основе вычислительных экспериментов вычислим достигнутое ускорение и занесем данные в таблицу 2.

Таблица 2.

Измерение ускорений параллельной программы

N	Нити		
	1	2	4
50	0	-1.4507	-4.7817
500	0	1.7583	2.7959
1000	0	1.8310	2.9302
2000	0	2.1513	2.5309

Заключение

Таким образом, с помощью использования средств библиотеки `orenpnr` разработан алгоритм распараллеливания системы дифференциальных уравнений модели мозга. Полученная параллельная программа была протестирована на 2 и 4 нитях. В результате при большем количестве нейронов время выполнения программы на 2 нитях ускорилось примерно в 2 раза, на 4 нитях примерно в 2,5 раза.

Список литературы

1. Баркалов К.А. Образовательный комплекс «Параллельные численные методы» / Лекционные мат-лы. Раздел 5. Методы решения систем обыкновенных дифференциальных уравнений. Н. Новгород, 2011.
2. Образовательный комплекс «Параллельные численные методы» / Лабораторная работа «Численное решение систем обыкновенных дифференциальных уравнений на примере решения задачи моделирования мозга» / Мееров И.Б. и др. Нижний Новгород, 2011.

References

1. Barkalov K.A. Educational complex “Parallel numerical methods” / Lecture materials. Section 5. Methods for solving systems of ordinary differential equations. Nizhny Novgorod, 2011.
2. Educational complex “Parallel numerical methods” / Laboratory work “Numerical solution of systems of ordinary differential equations by way of example of solution of a brain modeling problem” / Meerov I.B. et al. Nizhni Novgorod, 2011.

ДАННЫЕ ОБ АВТОРАХ

Аглямзянова Гульшат Накиповна, доцент кафедры информатики и вычислительной математики, кандидат физико-математических наук
*Набережночелнинский госуд-ый педагогический университет
ул. Низаметдинова, 28, Набережные Челны, 423806, Россия
dina.airat@mail.ru*

Гумерова Лилия Зуфаровна, доцент кафедры системного анализа и информатики, кандидат педагогических наук
*Казанский (Приволжский) государственный университет
ул. Кремлёвская, 18, г. Казань, 420008, Россия
gum9370@mail.ru*

DATA ABOUT THE AUTHORS

Aglyamzyanova Gulshat N., PhD in Physics and Mathematics, Associate Professor of the Department of Computer Science and Computational Mathematics
*Naberezhnye Chelny Pedagogical University
28, Nizametdinov Str., Naberezhnye Chelny, 423806, Russia
dina.airat@mail.ru
ORCID: 0000-0002-8003-5427*

Gumerova Liliya Z., PhD in Education, Associate Professor of the Department of System Analysis and Informatics
*Kazan Federal University
18, Kremlevskaya Str., Kazan, 420008, Russia
gum9370@mail.ru
ORCID: 0000-0002-8865-7589*