

3. *Золоторевич Л.А.* Экспериментальное исследование базовых методов моделирования в системе ModelSim / Л.А. Золоторевич, А.В. Ильинкова // The International Conference Computer– Aided Design of Diskrete Devices (CAD-DD'07). Vol.1. Minsk. 2007. P. 245–251.
4. *Золоторевич Л.А.* Анализ полноты тестов контроля цифровых устройств на основе решения задачи выполнимости // Л.А. Золоторевич, А.В. Ильинкова // Материалы Междунар. науч.-практ. конф. «Современная радиоэлектроника: научные исследования и подготовка кадров». Минск, 23–24 апреля 2008. С. 25–27.
5. *Золоторевич Л.А.* Анализ тестов контроля цифровых устройств по отрицательной нормальной форме реализуемой функции // Л.А. Золоторевич, А.В. Ильинкова // The Fifth International Scientific Conference «Information technologies in industry» ITI* 2008. P. 107–108.
- Ibarra O. H., Sahni S.* Polynomially Complete Fault Detection Problems // IEEE Transactions on Computers. 1975. V. C. 24. No. 3. P. 242–249.
7. *Murray B. T., Hayes J. P.* Hierarchical Test Generation Using Precomputed Tests for Modules // International Test Conference. 1988. P. 221–229.
8. *Ubar R.* Test Synthesis with Alternative Graphs // IEEE Design and Test of Computers. 1996. V. 13. No. 1. P. 48–57.
9. *Jervan G., Peng Z., Ubar R.* Test Cost Minimization for Hybrid BIST // IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'00). 2000. P. 283–291.
10. *Roth J.P.* Diagnosis of automata failures: a calculus and a method // IEEE Trans. on Computers. 1966. V. 15. No. 7. P. 278–291.

ЧИСЛЕННЫЙ АНАЛИЗ ОДНОГО ПАРАРЕАЛЬНОГО АЛГОРИТМА РЕШЕНИЯ СИСТЕМ ОДУ

В. А. Карпейчик

В настоящее время особенно актуальным становится вопрос наличия алгоритмов, способных выполняться параллельно несколькими вычислительными модулями, так как при исполнении последовательного алгоритма на вычислительной системе, использующей несколько вычислительных модулей, вычисления будут производиться только на одном из них. Важной проблемой при моделировании и решении сложных задач является решение дифференциальных уравнений и их систем. В данной работе проведено численное исследование одного парареального метода решения систем ОДУ, впервые представленного в 2001 году Габриэлем Туриничи и Айвоном Мэдзем[1]. Парареальным они называют алгоритм решения исходной задачи, позволяющий производить параллельные вычисления на отрезках временной оси дифференциальной задачи.

Пусть имеем дифференциальную задачу

$$\begin{aligned} u'(t) &= f(u(t)), \\ u(T_0) &= u^0, \end{aligned} \tag{1}$$

где $t \in [T_0, T] = \Omega$, $u \in \Omega \rightarrow R^n$, $f \in R^n \rightarrow R^n$.

Для построения парареального алгоритма разобьем Ω на N отрезков $\Omega_n = [T_n, T_{n+1}]$, где $n = 0, 1, \dots, N-1$ и решаем на каждом Ω_n задачу[2]

$$\begin{aligned} u_n'(t) &= f(u_n(t)), \\ u_n(t) &= U_n, \end{aligned} \quad (2)$$

где $t \in \Omega_n$, а значения $U_0, \dots, U_n$ должны выбираться так, чтобы $u_i(T_{i+1}) = u_{i+1}(T_{i+1}), i = 0, \overline{N-2}$.

То есть U_n должны удовлетворять системе уравнений

$$\begin{aligned} U_0 &= u^0, \\ U_n &= \varphi_{\Omega_{n-1}}(U_{n-1}), \end{aligned} \quad (3)$$

где $n = 0, 1, \dots, N-1$, $\varphi_{\Omega_{n-1}}(U_{n-1})$ – это точное решение задачи вида (2) на отрезке Ω_{n-1} с начальным условием равным U_{n-1} . Тогда систему (3) можно переписать в виде:

$$\begin{aligned} U_0 - u^0 &= 0, \\ U_1 - \varphi_{\Omega_0}(U_0) &= 0, \\ U_{N-1} - \varphi_{\Omega_{N-2}}(U_{N-2}) &= 0. \end{aligned} \quad (4)$$

Система (4) позволяет определить необходимые значения U_n для каждого из отрезков Ω_n . Решать ее можно, например, при помощи модифицированного метода Ньютона. Тогда запишем:

$$\begin{aligned} U_0^k &= u^0, k = 0, 1, \dots, \\ U_n^{k+1} &= \varphi_{\Omega_{n-1}}(U_{n-1}^k) + \varphi_{\Omega_{n-1}}'(U_{n-1}^k)(U_{n-1}^{k+1} - U_{n-1}^k), \\ k &= 0, 1, \dots; n = 1, \dots, N-1. \end{aligned} \quad (5)$$

Теперь, пусть $F(T_{n-1}, T_n, U_{n-1})$ – приближенное решение к $\varphi_{\Omega_{n-1}}(U_{n-1})$ порядка p_0 , а $G(T_{n-1}, T_n, U_{n-1})$ – приближенное решение к $\varphi_{\Omega_{n-1}}'(U_{n-1})$ порядка p_1 , причем $p_0 > p_1$. Тогда справедливо[2]

$$\begin{aligned} \varphi_{\Omega_{n-1}}(U_{n-1}^k) &\approx F(T_{n-1}, T_n, U_{n-1}^k), \\ \varphi_{\Omega_{n-1}}'(U_{n-1}^k)(U_{n-1}^{k+1} - U_{n-1}^k) &\approx G(T_{n-1}, T_n, U_{n-1}^{k+1}) - G(T_{n-1}, T_n, U_{n-1}^k). \end{aligned}$$

Тогда из (5) можно записать парареальный алгоритм решения исходной задачи:

$$U_0^{k+1} = u^0,$$

$$U_n^{k+1} = F(T_{n-1}, T_n, U_{n-1}^k) + G(T_{n-1}, T_n, U_{n-1}^{k+1}) - G(T_{n-1}, T_n, U_{n-1}^k),$$

$$k = 0, 1, \dots; n = 1, \dots, N-1,$$

где $U_n^0 = G(T_n, T_{n-1}, U_{n-1}^0)$.

Представим общий вид параллельного алгоритма при исполнении на N процессорах[1].

$$U_0^0 \leftarrow u^0$$

for i=0:N-1 do

$$U_{i+1}^0 \leftarrow G(T_i, T_{i+1}, U_i^0)$$

end for

Разбиваем Ω на N отрезков и вычисляем $F(T_n, T_{n+1}, U_{n-1}^0)$ параллельно на N процессорах, по одному отрезку на процессор.

$$k \leftarrow 0$$

while true do

$$U_0^{k+1} \leftarrow U_0^k$$

for i=0:N-1 do

вычисляем $G(T_i, T_{i+1}, U_i^{k+1})$

$$U_{i+1}^{k+1} = F(T_i, T_{i+1}, U_i^k) + G(T_i, T_{i+1}, U_i^{k+1}) - G(T_i, T_{i+1}, U_i^k)$$

end for

if сходимость

выход

end if

Вычисляем $F(T_n, T_{n+1}, U_{n-1}^{k+1})$ параллельно на N процессорах, по одному отрезку на процессор.

$$k \leftarrow k + 1$$

end while

Для проверки эффективности алгоритма были проведены тесты на задаче нахождения орбиты Аренсторфа, которая представляет собой замкнутую кривую. Всего было произведено две серии тестов, первая – для достижения одинаковой точности. Вторая – тесты с одинаковым шагом сетки.

Все тесты производились на компьютере с двухъядерном процессоре Intel Core 2 Duo с тактовой частотой каждого ядра 2 ГГц и оперативной памятью 4 Гб.

В тестах сравнивались:

- последовательный алгоритм метода Рунге-Кутты 4-го порядка аппроксимации;

- парареальный алгоритм при запуске на одном ядре процессора;

- парареальный алгоритм при запуске на 2-х ядрах процессора.

Первая серия – тесты на достижение одинаковой точности. В таблице показана зависимость времени выполнения в секундах от количества витков вычисляемого решения.

Таблица

Алгоритм	1	2	3	4
Р-К 4	0.2	1.11	6.83	38.96
Парареальный на 1 ядре	0.39	1.37	8.63	18.02
Парареальный на 2 ядрах	0.29	1.15	6.42	9.83

Вторая серия – тесты с одинаковым шагом сетки $h=0.0003$. Ниже на рисунках приводятся результаты такого тестирования (точное решение показано прерывистой линией):

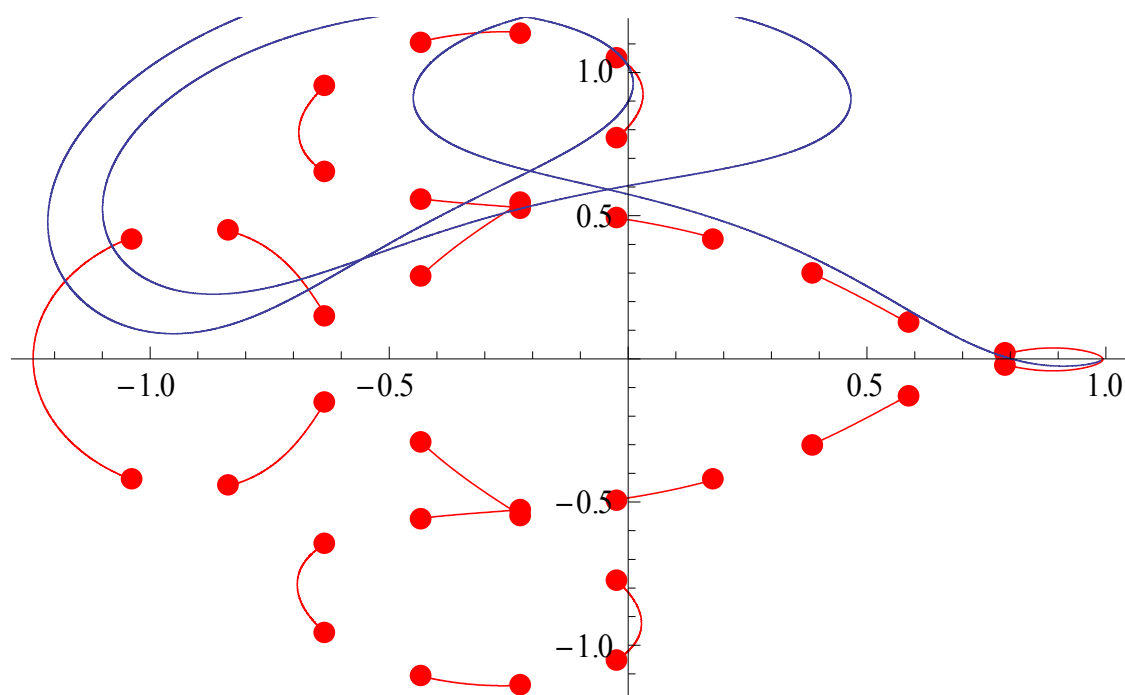


Рис. 1. Решение методом Рунге-Кутты 4-го порядка точности, количество витков – 3, время выполнения – 6.352 сек.

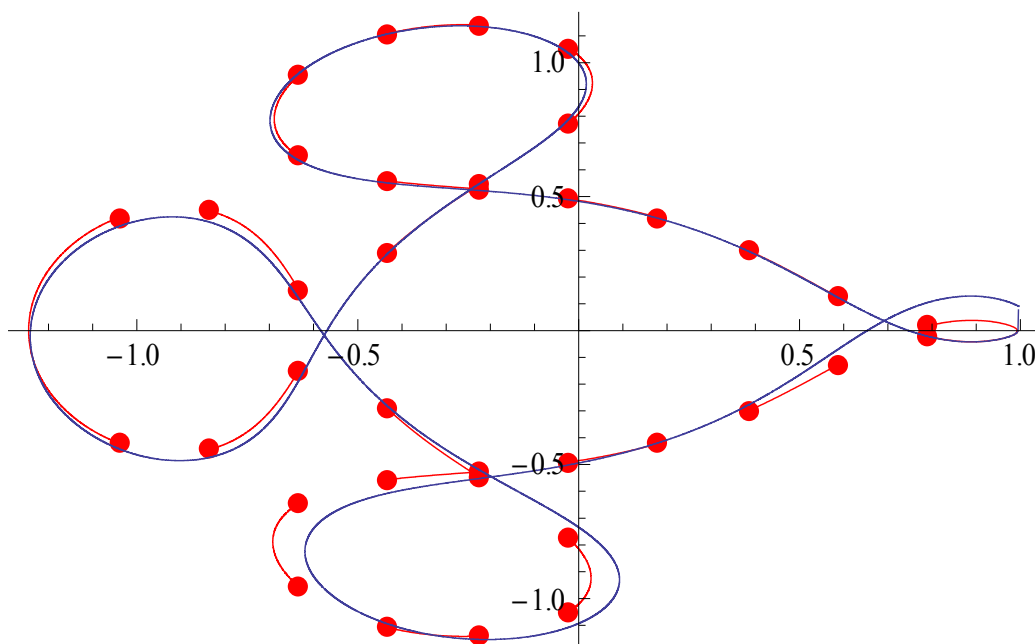


Рис. 2. Решение парареальным алгоритмом, количество витков – 4.
Время выполнения: на одном ядре – 10.36 сек, на двух ядрах – 6.504 сек

Полученные результаты сравнения работы разных алгоритмов показали, что парареальный алгоритм ведет себя эффективнее, чем классический метода Рунге-Кутты 4-го порядка точности.

Во второй серии тестов примечательным является результат, полученный при вычислении с шагом сетки 0.0003, когда метод Рунге-Кутты 4-го порядка аппроксимации смог проделать только два витка, с сильной погрешностью в самом начале 3-его, а парареальный алгоритм почти закончил четвертый виток.

При этом необходимо отметить относительно невысокую скорость работы парареального алгоритма на сетке малой размерности. Это можно объяснить тем, что время тратится на инициализацию потоков, которые при малом количестве вычисляемых шагов излишни.

Литература

1. *Gunnar A. Staff and E.M. Rønquist* Stability of the parareal algorithm, Proceedings of the 15th International Conference on Domain Decomposition Methods (Berlin 2003). // Lecture Notes in Computational Science and Engineering. Vol. 40. Springer. 2004.
2. *M.J. Gander and E. Hairer* Nonlinear Convergence Analysis for the Parareal Algorithm. // Proceedings of the 17th International Conference on Domain Decomposition Methods. 2007.