

УДК 519.71

ДЕКОМПОЗИЦИОННАЯ РЕАЛИЗАЦИЯ СХЕМЫ ГОРНЕРА ДЛЯ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ МНОГОМЕРНЫХ МНОГОЧЛЕНОВ¹⁾

© 2020 г. А. П. Афанасьев^{1,*}, С. М. Дзюба^{2,**}

¹ 127994 Москва, Большой Каретный пер., 19, стр. 1, ИППИ РАН, Россия;

² 170026 Тверь, наб. Афанасия Никитина, 22, ТвГТУ, Россия

*e-mail: apa@iitp.ru

**e-mail: sdzyuba@mail.ru

Поступила в редакцию 26.06.2017 г.

Переработанный вариант 02.09.2019 г.

Принята к публикации 17.10.2019 г.

Приведена декомпозиционная реализация схемы Горнера для вычисления значений многомерных многочленов, сводящая задачу к вычислению по схеме Горнера последовательности значений одномерных многочленов. Изучена возможность использования этой схемы в распределенной компьютерной среде. В качестве примера работы схемы рассмотрена проблема построения приближенных аналитических решений дифференциальных уравнений с полиномиальной правой частью. Библ. 11. Фиг. 1. Табл. 1.

Ключевые слова: многомерные многочлены, схема Горнера, декомпозиция вычислительного процесса, дифференциальные уравнения с полиномиальной правой частью.

DOI: 10.31857/S0044466920020027

1. ВВЕДЕНИЕ

Рассмотрим многомерный многочлен

$$p_n(\mathbf{x}) = \sum_{i_1=0}^N \dots \sum_{i_n=0}^N a_{i_1 \dots i_n} x_1^{i_1} \dots x_n^{i_n}, \quad (1.1)$$

где $\mathbf{x} = (x_1, \dots, x_n)$ есть n -мерный действительный вектор, $a_{i_1 \dots i_n}$ — постоянные действительные коэффициенты многочлена и N — некоторое натуральное число, представляющее собой максимальную степень, которая достигается по крайней мере по одному из переменных x_1, \dots, x_n .

Проблема вычисления значений многочлена (1.1) достаточно актуальна. Этими многочленами описываются различные математические объекты такие, как правые части обыкновенных дифференциальных уравнений, ограничения задач нелинейного программирования, начальные и граничные условия задач математической физики и многие другие (см., например, [1]–[4]).

Хорошо известно, что вычисление значений многочленов вида (1.1) одного переменного удобно осуществлять с помощью схемы Горнера, позволяющей во многих случаях уменьшить верхнюю оценку ошибки вычислений (см., например, [1], [5]–[7]). Что касается многомерных многочленов, то на сегодняшний день известны несколько вычислительных алгоритмов, реализующих различные обобщения схемы Горнера (см., например, [1], [8]–[11]). Здесь помимо преимуществ одномерной схемы появляется возможность декомпозиции вычислительного процесса. Так, например, в работе [9] показано, что при наличии $O(N)$ процессоров вычислительный процесс может быть декомпозирован на $O(\log_2(N) \log_2(n))$ шагов параллельных вычислений.

В работах [10], [11] приведена и подробно изучена еще одна из возможных реализаций схемы Горнера. В этих работах схема Горнера сводится к преобразованию исходного многочлена и его представления в виде комбинации мономов. Однако результаты этих работ не учитывают и не

¹⁾Работа выполнена при поддержке РФФ (проект 16-11-10352). Исследование выполнено при финансовой поддержке Минобрнауки РФ, проект № 2.1777.2017/4.6.

используют тот факт, что одномерная схема Горнера хорошо изучена; в частности, в настоящее время существуют весьма эффективные программы для вычисления значений одномерных многочленов. Более того, в [10], [11] не рассматривается вопрос о распараллеливании и декомпозиции вычислительного процесса.

Целью настоящей работы является построение декомпозиционной реализации схемы Горнера для вычисления значений многомерных многочленов (1.1). Особое внимание будет уделено описанию соответствующего алгоритма, ориентированного на сведение вычислительного процесса к последовательно-параллельному вычислению значений одномерных многочленов.

2. ДЕКОМПОЗИЦИОННАЯ РЕАЛИЗАЦИЯ СХЕМЫ ГОРНЕРА

Схема Горнера для одномерных многочленов хорошо известна и изучается во многих работах (см., например, [5]). Суть ее состоит в следующем.

Рассмотрим действительный одномерный многочлен

$$p_1(x) = \sum_{i=0}^N a_i x^i, \quad (2.1)$$

где a_i – некоторые действительные числа и x – действительное переменное. Согласно [5] схема Горнера для многочлена (2.1) имеет следующий вид:

$$p_1(x) = ((\dots((a_N x + a_{N-1})x + a_{N-2})x + \dots + a_2)x + a_1)x + a_0. \quad (2.2)$$

Рассмотрим обобщение схемы Горнера для многочлена (1.1). При $i_1 = 0, \dots, N$ определим функции $(x_2, \dots, x_n) \rightarrow \alpha_{i_1}(x_2, \dots, x_n)$. Для этого положим

$$\alpha_{i_1}(x_2, \dots, x_n) = \sum_{i_2=0}^N \dots \sum_{i_n=0}^N a_{i_1 i_2 \dots i_n} x_2^{i_2} \dots x_n^{i_n}, \quad i_1 = 0, \dots, N. \quad (2.3)$$

Тогда имеем

$$p_n(\mathbf{x}) = ((\dots((\alpha_N(x_2, \dots, x_n)x_1 + \alpha_{N-1}(x_2, \dots, x_n))x_1 + \alpha_{N-2}(x_2, \dots, x_n))x_1 + \dots + \alpha_2(x_2, \dots, x_n))x_1 + \alpha_1(x_2, \dots, x_n))x_1 + \alpha_0(x_2, \dots, x_n). \quad (2.4)$$

Каждая функция $(x_2, \dots, x_n) \rightarrow \alpha_{i_1}(x_2, \dots, x_n)$, задаваемая равенством (2.3), является многомерным многочленом переменных x_2, \dots, x_n . Для вычисления значений этих многочленов при $i_1, i_2 = 0, \dots, N$ определим функции $(x_3, \dots, x_n) \rightarrow \alpha_{i_1 i_2}(x_3, \dots, x_n)$. Для этого положим

$$\alpha_{i_1 i_2}(x_3, \dots, x_n) = \sum_{i_3=0}^N \dots \sum_{i_n=0}^N a_{i_1 i_2 i_3 \dots i_n} x_3^{i_3} \dots x_n^{i_n}, \quad i_1, i_2 = 0, \dots, N.$$

Тогда имеем

$$\begin{aligned} \alpha_{i_1}(x_2, \dots, x_n) = & ((\dots((\alpha_{i_1 N}(x_3, \dots, x_n)x_2 + \alpha_{i_1, N-1}(x_3, \dots, x_n))x_2 + \\ & + \alpha_{i_1, N-2}(x_3, \dots, x_n))x_2 + \dots + \alpha_{i_1 2}(x_3, \dots, x_n))x_2 + \alpha_{i_1 1}(x_3, \dots, x_n))x_2 + \alpha_{i_1 0}(x_3, \dots, x_n), \\ & i_1 = 0, \dots, N. \end{aligned}$$

Действуя аналогичным образом, для произвольного значения $k = 1, \dots, n-1$ положим

$$\alpha_{i_1 \dots i_k}(x_{k+1}, \dots, x_n) = \sum_{i_{k+1}=0}^N \dots \sum_{i_n=0}^N a_{i_1 \dots i_k i_{k+1} \dots i_n} x_{k+1}^{i_{k+1}} \dots x_n^{i_n}, \quad i_1, \dots, i_k = 0, \dots, N.$$

Тогда несложно заметить, что для определения функций $(x_k, \dots, x_n) \rightarrow \alpha_{i_1 \dots i_{k-1}}(x_k, \dots, x_n)$ при $i_1, \dots, i_{k-1} = 0, \dots, N$ справедливо рекуррентное соотношение

$$\begin{aligned} \alpha_{i_1 \dots i_{k-1}}(x_k, \dots, x_n) = & ((\dots((\alpha_{i_1 \dots i_{k-1} N}(x_{k+1}, \dots, x_n)x_k + \alpha_{i_1 \dots i_{k-1}, N-1}(x_{k+1}, \dots, x_n))x_k + \\ & + \alpha_{i_1 \dots i_{k-1}, N-2}(x_{k+1}, \dots, x_n))x_k + \dots + \alpha_{i_1 \dots i_{k-1} 2}(x_{k+1}, \dots, x_n))x_k \alpha_{i_1 \dots i_{k-1} 1}(x_{k+1}, \dots, x_n))x_k + \\ & + \alpha_{i_1 \dots i_{k-1} 0}(x_{k+1}, \dots, x_n), \quad i_1, \dots, i_{k-1} = 0, \dots, N. \end{aligned} \quad (2.5)$$

При этом

$$\alpha_{i_1 \dots i_{n-1}}(x_n) = \sum_{i_n=0}^N a_{i_1 \dots i_n} x_n^{i_n} = ((\dots((a_{i_1 \dots i_{n-1} N} x_n + a_{i_1 \dots i_{n-1} N-1}) x_n + a_{i_1 \dots i_{n-1} N-2}) x_n + \dots + a_{i_1 \dots i_{n-1} 2}) x_n + a_{i_1 \dots i_{n-1}}) x_n + \alpha_{i_1 \dots i_{n-1} 0}, \quad i_1, \dots, i_{n-1} = 0, \dots, N. \quad (2.6)$$

Таким образом, имеем соотношения (2.4)–(2.6), определяющие обобщенную схему Горнера для вычисления значений многочлена (1.1).

Обобщенная схема Горнера во многих случаях позволяет упростить вычисления. Прямым подсчетом устанавливается, что вычисление значений многочлена (1.1) по схеме (2.4)–(2.6) эквивалентно вычислению значений

$$\mathcal{N} = \sum_{i=0}^{n-1} (N+1)^i$$

одномерных многочленов. Более того, число умножений по обобщенной схеме Горнера равно

$$H(N, n) = (N+1)^n - 1.$$

Число же умножений по “тривиальному” алгоритму (т.е. алгоритму прямого вычисления по формуле (1.1) при условии, что величина y^k вычисляется как $y^k = y y^{k-1}$) равно

$$T(N, n) = n(N+1)^n + nN.$$

Уже только это говорит о том, что чем больше значения n и N , тем эффективнее становится использование обобщенной схемы Горнера.

Процесс вычисления значений многочлена (1.1) по предложенной выше схеме сводится к процессу вычисления значений одномерных многочленов. Вычисление же значений одномерных многочленов по схеме Горнера в настоящее время реализовано на самом высоком уровне (см., например, [1], [5]–[7]).

3. АЛГОРИТМ

Для реализации описанной выше обобщенной схемы Горнера будем использовать следующий алгоритм, состоящий из n шагов и ориентированный на декомпозицию вычислительного процесса.

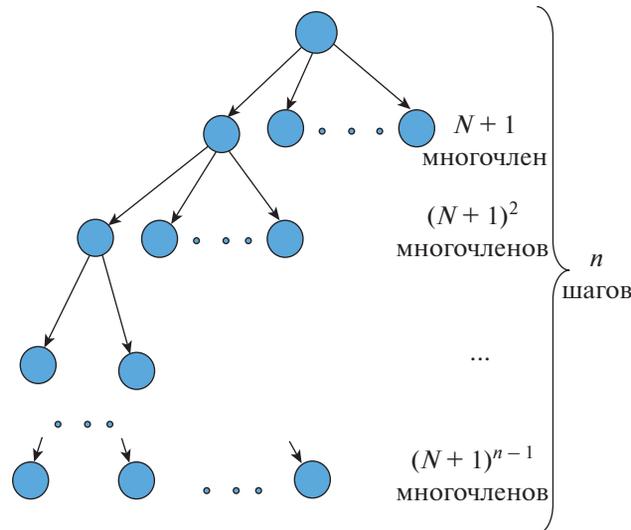
Шаг 1. Здесь осуществляется вычисление значений функций $x_n \rightarrow \alpha_{i_1 \dots i_{n-1}}(x_n)$ по формуле (2.6). Каждая из этих функций, очевидно, может строиться независимо. Поэтому процесс построения может быть естественным образом декомпозирован на параллельное вычисление значений всех этих функций.

Шаги 2, ..., n-1. На данных шагах для $k = n-1, \dots, 2$ по формуле (2.5) вычисляются значения функций $(x_k, \dots, x_n) \rightarrow \alpha_{i_1 \dots i_{k-1}}(x_k, \dots, x_n)$. Как и на шаге 1, каждая из этих функций может строиться независимо, т.е. здесь процесс построения в общем случае также может быть естественным образом декомпозирован на параллельное вычисление значений всех этих функций.

Шаг n. Данный шаг наиболее простой: на нем по формуле (2.4) вычисляются значения функции $\mathbf{x} \rightarrow p_n(\mathbf{x})$.

Условная схема организации работы описанного выше алгоритма приведена на фиг. 1. Как видно из фиг. 1, при наличии $(N+1)^{n-1}$ процессоров значения всех одномерных многочленов на каждом шаге алгоритма могут быть вычислены параллельно. Следовательно, вычисление значений многочлена (1.1) эквивалентно вычислению значений n одномерных многочленов.

В общем случае значение $(N+1)^{n-1}$ может быть достаточно велико. Поэтому наиболее эффективно представленный алгоритм может быть реализован в распределенной компьютерной среде с большим числом независимых узлов.



Фиг. 1. Условная схема работы алгоритма.

4. ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ С ПОЛИНОМИАЛЬНОЙ ПРАВОЙ ЧАСТЬЮ

В качестве одного из важнейших примеров использования обобщенной схемы Горнера обратимся к проблеме построения приближенных аналитических решений дифференциальных уравнений с полиномиальной правой частью.

Рассмотрим систему обыкновенных дифференциальных уравнений, векторная запись которой имеет вид

$$\dot{\mathbf{x}} = \mathbf{p}_n(\mathbf{x}), \tag{4.1}$$

где $\mathbf{x}(t) = (x_1(t), \dots, x_n(t))$ – действительная векторная функция действительного переменного t , а $\mathbf{p}_n = (p_1, \dots, p_n)$ – действительная векторная функция, каждый элемент которой p_i является n -мерным многочленом переменных x_1, \dots, x_n .

Для построения решения $\mathbf{x} = \xi(t)$ системы (4.1) с начальным условием

$$\xi(0) = \xi_0$$

удобно учитывать специальный вид правой части данной системы (см., например, [1]–[4]). Наиболее простым и естественным является метод построения приближенного аналитического решения с использованием его разложения в ряд Тейлора (см., например, [1], [2]). В этом случае имеем

$$\xi(t) = \xi_0 + \sum_{i=1}^l \psi_i(\xi_0)t^i + o(|t|^l),$$

где $\psi_i(\xi_0)$ – соответствующие векторные n -мерные многочлены компонент вектора ξ_0 (см., например, [1], [2]). Другими словами, с некоторой заданной точностью, определяющей количество слагаемых l , решение $\mathbf{x} = \xi(t)$ можно аппроксимировать приближенным равенством

$$\xi(t) \approx \xi_0 + \sum_{i=1}^l \psi_i(\xi_0)t^i. \tag{4.2}$$

Соотношение (4.2) дает приближение к решению $\xi(t)$ в виде времени t и векторного n -мерного многочлена вектора ξ_0 . Однако, как известно, радиус сходимости ряда

$$\sum_{i=1}^l \psi_i(\xi_0)t^i$$

Таблица 1

Число членов разложения l	Декомпозиция	Время вычисл.
30	Да	2.401 с
30	Нет	4.541 с
40	Да	5.589 с
40	Нет	14.802 с

во многих случаях достаточно мал. Поэтому оценки значений функции $\xi(t)$ будем строить по формуле

$$\xi_{j+1} = \xi_j + \sum_{i=1}^l \psi_i(\xi_j) h^i, \quad j = 0, \dots, J, \quad (4.3)$$

где h – шаг по t .

Для вычисления значений каждого из многочленов ψ_i в формуле (4.3) нами использовалась обобщенная схема Горнера, как с декомпозицией вычислительного процесса, так и без декомпозиции. Вычислительные эксперименты, проведенные на типовых уравнениях (уравнение Ван-дер-Поля, система Лоренца и др.), показали, что использование декомпозиции в сочетании с обобщенной схемой Горнера позволяет при сохранении точности сократить время расчетов в несколько раз.

Для иллюстрации сказанного обратимся к проблеме построения решений системы Лоренца

$$\dot{x} = \sigma(y - x), \quad \dot{y} = rx - y - xz, \quad \dot{z} = xy - bz, \quad (4.4)$$

при классических значениях параметров данной системы:

$$\sigma = 10, \quad r = 28, \quad b = 8/3.$$

Для начального условия

$$x(0) = 1, \quad y(0) = 1, \quad z(0) = 1$$

вычисления проводились для $J = 1000$ с шагом $h = 0.01$ на кластере параллельных вычислений (4 процессора). В ходе вычислительного эксперимента осуществлялась эмуляция мантиксы в сто десятичных знаков (использовалась библиотека MPFR (https://en.wikipedia.org/wiki/GNU_MPFR)). Данные о времени вычислительного процесса сведены в табл. 1.

Система (4.3) является системой с полилинейной правой частью; поэтому для нее максимальная степень многочлена ψ_i равна i (см., например, [1]). Несложный анализ табл. 1 показывает, что использование декомпозиционной реализации обобщенной схемы Горнера даже в простейшем случае сокращает время вычислений. Разница во времени начинает увеличиваться уже при переходе с $l = 30$ до $l = 40$.

5. ОБСУЖДЕНИЕ

Отличительной особенностью описанной в настоящей работе обобщенной схемы Горнера является очевидная возможность декомпозиции вычислительного процесса. При этом сама обобщенная схема Горнера (даже без декомпозиции вычислительного процесса) во многих случаях позволяет уменьшить верхнюю оценку ошибки по сравнению с прямым вычислением многочлена.

Алгоритм, реализующий эту схему, сводится к последовательно-параллельному вычислению значений одномерных многочленов. Отличительная особенность такой реализации состоит в том, что здесь возможна очевидная декомпозиция вычислительного процесса, которая может быть осуществлена в распределенной компьютерной среде.

СПИСОК ЛИТЕРАТУРЫ

1. *Афанасьев А.П.* Продолжение траекторий в оптимальном управлении. Труды ИСА РАН. М.: КомКнига, 2006.
2. *Афанасьев А.П., Тарасов А.С.* Квазианалитическое решение систем дифференциальных уравнений с полиномиальными правыми частями // Сб. трудов ИСА РАН “Пробл. вычисл. в распределенной среде: распределенные приложения, коммуникационные системы, матем. модели и оптимизация”. Т. 25. С. 165–183. М.: КомКнига, 2006.
3. *Афанасьев А.П., Дзюба С.М.* Метод построения приближенных аналитических решений систем обыкновенных дифференциальных уравнений с полиномиальной правой частью // Ж. вычисл. матем. и матем. физ. 2015. Т. 55. № 10. С. 1694–1702.
4. *Афанасьев А.П., Дзюба С.М.* Построение минимальных множеств дифференциальных уравнений с полиномиальной правой частью // Дифференц. уравнения. 2015. Т. 51. № 11. С. 1411–1419.
5. *Мак-Кракен Д., Дорн У.* Численные методы и программирование на ФОРТРАНЕ. М.: Мир, 1977.
6. *Левитин А.В.* Схема Горнера и возведение в степень // Алгоритмы. Введение в разработку и анализ. С. 284–291. М.: Вильямс, 2006.
7. *Cormen T.H., Leiserson C.E., Rivest R.L., Stein C.* Introduction to the algorithms. New York: MIT Press and McGraw-Hill, 2009.
8. *Диканев Т.В.* Вычисление полинома от нескольких переменных // <http://www.tvd-home.ru/numerical/polynomial>
9. *Dowling M.L.* A fast parallel Horner algorithm // SIAM J. Comp. 1990. Vol. 19. No. 1. P. 133–142.
10. *Peña J.M., Sauer T.* On the multivariate Horner scheme // SIAM J. Numer. Anal. 2000. Vol. 37. No. 4. P. 1186–1197.
11. *Peña J.M., Sauer T.* On the multivariate Horner scheme II: Running Error Analysis // Computing (Vienna/New York). 2000. Vol. 65. P. 313–322.