Decomposition Algorithm of Generalized Horner Scheme for Multidimensional Polynomial Evaluation

Alexander P. Afanasiev[1,2,3,4] Sergei M. Dzyuba[5] Irina I. Emelyanova[5]
Elena V. Putilina[1]

[1]Institute for Information Transmission Problems,
Russian Academy of Sciences, Nakhimovsky Prospekt 36-1, Moscow 117218, Russia
[2]National Research University Higher School of Economics, Myasnitskaya str., 20, Moscow 101000, Russia
[3]National University of Science and Technology MISiS, Leninskiy Prospekt 4, Moscow 119049, Russia
[4]Lomonosov Moscow State University, GSP-1, Vorobievy Gory, Moscow 119991, Russia
[5]Tver State Technical University, Af. Nikitin emb. 22, Tver 170026, Russia
E-mail: apa@iitp.ru, sdzyuba@mail.ru, emelyanova-123@yandex.ru, elena@iitp.ru

Abstract

We suggested a decomposition algorithm of multidimensional polynomials evaluation performing according to the generalized Horner scheme. The generalized Horner scheme is described in detail; we also demonstrate that in the vast majority of real situations this scheme can be decomposed easily and naturally. As a use-example, we study the problem of evaluation of classic quasipolynomials.

1 Introduction

Let's consider a multidimensional polynomial

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

where $\mathbf{x} = (x_1, \dots, x_n)$ is an n-dimensional real data vector, $a_{i_1 \dots i_n}$ are constant real coefficients of the polynomial and N is some natural number representing the maximal degree attained by at least one of the variables $x_1, \dots, x_n$.

The problem of the polynomial evaluation (1.1) is rather popular. It is explained by the fact that different mathematical objects such as the right sides of ordinary differential equations, constraints of nonlinear programming problems, initial and boundary conditions of mathematical physics problems and many others are described by these polynomials (see [Afanasiev, 2005]–[Afanasiev & Dzyuba, 2015b]).

It is well-known that polynomial evaluation of the form (1.1) of one variable is easy to perform in terms of the Horner scheme allowing in many cases to reduce the number of machine operations resulting in reduction of computation error (see [Afanasiev, 2005], [McCracken & Dorn, 1965]–[Cormen et al., 2009]). As for multidimensional polynomials several computational algorithms representing different generalizations of Horner scheme to a multidimensional case are known nowadays (see [Afanasiev, 2005]).

In: Yu. G. Evtushenko, M. Yu. Khachay, O. V. Khamisov, Yu. A. Kochetov, V.U. Malkova, M.A. Posypkin (eds.): Proceedings of the OPTIMA-2017 Conference, Petrovac, Montenegro, 02-Oct-2017, published at http://ceur-ws.org

Here we should point out an important work [Dowling, 1990], presenting a multidimensional performance of the generalized Horner scheme, this performance applying parallel computations. Besides, the work [Dowling, 1990] provides an important evaluation of parallel computations steps as a function of n and N parameters and of the number of the processors applied.

The distinctive feature of the indicated algorithms is that the multidimensional analog of Horner scheme, in contrast to the scalar one, not only reduces the amount of computation and reduces the error, but also represents additional opportunities. A namely, a possibility of computation process decomposition accompanies to the advantages of the one-dimensional scheme (see [Afanasiev, 2005]). However, all known results either are attributed to special cases of polynomial or have no detailed extended characterization.

The main goal of this paper is the construction and analysis of complete generalized Horner scheme for multidimensional polynomials evaluation (1.1). Special focus will be on the description of the scheme and the algorithm related to the scheme, especially oriented on decomposing of computation process (see [Afanasiev, 2005]).

As it will be demonstrated, this variant of the scheme is focused on being performed in a distributed computer environment. Additionally, we shall note that the stated algorithm was developed to meet the needs of the optimization theory in connection with the issue of constructing approximate solutions for the problem of control of nonlinear systems via quadratic criteria within long time intervals (see [Afanasiev et al., 2016]).

2 Generalized Horner Scheme

The Horner scheme for one-dimensional polynomials is well known and is under study in many papers (see [Afanasiev, 2005]). The point is as follows.

Let's consider a real one-dimensional polynomials

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

where a_i are some real numbers and x is a real variable. According to [5], the Horner scheme for polynomial (2.1) appears as follows:

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

Remark 1. Let's clarify the meaning of the Horner scheme use. For this purpose let's consider a complete polynomial, i.e. a polynomial in which all a_i coefficients are different from zero.

Let's assume that for the polynomial evaluation by (2.1) the innermost brackets are expanded first. Further it is easily seen that N multiplications and N additions are required by the Horner scheme for (2.1) polynomial evaluation. Therewith, the number of machine operations for the evaluation by (2.1) equal to $N(N+1)/2$, if only each x degree is obtained by sequential multiplication by x, i.e. $x^k = x^{k-1}x$ (see [McCracken & Dorn, 1965]).

This means that the greater N value the closer to complete is the polynomial (2.1) the Horner scheme use (2.2) becomes more effective, which allows in similar cases to reduce a number of machine operations essentially and thus to reduce computation error.

Let's illustrate the generalization of the scheme (2.2) meant for multidimensional polynomials evaluation of the form (1.1) on the following example.

Example 1. Let's consider a classic real polynomial

$$p_2(x, y) = \sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij} x^i y^j \tag{2.3}$$

of x and y real variables. For the polynomial evaluation (2.3) let's state y variable and assume

$$\alpha_i(y) = \sum_{j=0}^{N} a_{ij} y^j, \quad i = 0, \dots, N. \tag{2.4}$$

Then if one consider with y defined as polynomial of one x real variable, then, acting by analogy to (2.1) and (2.2), we hold:

$$p_2(x, y) = ((\dots((\alpha_N(y)x + \alpha_{N-1}(y))x + \alpha_{N-2}(y))x + \dots$$

$$\ldots + \alpha_2(y))x + \alpha_1(y))x + \alpha_0(y). \tag{2.5}$$

It is easily seen that with any y defined the scheme (2.5) is a primitive generalization of the one-dimensional Horner scheme to polynomial (2.3). To obtaining a finer generalization let's note that each $y \to \alpha_i(y)$ function is one-dimensional polynomial. Thus, applying the one-dimensional Horner scheme to polynomial (2.4), we assume

$$\alpha_i(y) = ((\ldots((a_{iN}y + a_{i,N-1})y + a_{i,N-2})y + \ldots + a_{i2})y + a_{i1})y + a_{i0},$$

$$i = 0, \ldots, N. \tag{2.6}$$

Thus (2.5) and (2.6) equalities provide generalized Horner scheme (1.1) for polynomial evaluation (2.3). It is apparent from these equalities show that in the general case Horner scheme use allows to considerably reduce computation process in comparison to brute force computation by (2.3).

Going now over to construction of generalized Horner scheme for polynomial (1.1) in the general case let's state $x_2, \ldots, x_n$ variables and assume

$$\alpha_{i_1}(x_2, \ldots, x_n) = \sum_{i_2=0}^{N} \ldots \sum_{i_n=0}^{N} a_{i_1 i_2 \ldots i_n} x_2^{i_2} \ldots x_n^{i_n}, \quad i_1 = 0, \ldots, N. \tag{2.7}$$

Then acting by analogy to (2.1) and (2.2) we hold:

$$p_n(\mathbf{x}) = ((\ldots((\alpha_N(x_2, \ldots, x_n)x_1 + \alpha_{N-1}(x_2, \ldots, x_n))x_1 +$$

$$+\alpha_{N-2}(x_2, \ldots, x_n))x_1 + \ldots + \alpha_2(x_2, \ldots, x_n))x_1 +$$

$$+\alpha_1(x_2, \ldots, x_n))x_1 + \alpha_0(x_2, \ldots, x_n). \tag{2.8}$$

Let's now note that each $(x_2, \ldots, x_n) \to \alpha_{i_1}(x_2, \ldots, x_n)$ function defined by equality (2.7) is a multidimensional polynomial of $x_2, \ldots, x_n$ variables. For these polynomials evaluation let's state $x_3, \ldots, x_n$ variables and assume

$$\alpha_{i_1 i_2}(x_2, \ldots, x_n) = \sum_{i_3=0}^{N} \ldots \sum_{i_n=0}^{N} a_{i_1 i_2 i_3 \ldots i_n} x_3^{i_3} \ldots x_n^{i_n}, \quad i_1, i_2 = 0, \ldots, N.$$

Then we hold

$$\alpha_{i_1}(x_2, \ldots, x_n) = ((\ldots((\alpha_{i_1 N}(x_3, \ldots, x_n)x_2 +$$

$$+\alpha_{i_1, N-1}(x_3, \ldots, x_n))x_2 + \alpha_{i_1, N-2}(x_3, \ldots, x_n))x_2 + \ldots$$

$$\ldots + \alpha_{i_1 2}(x_3, \ldots, x_n))x_2 + \alpha_{i_1 1}(x_3, \ldots, x_n))x_2 + \alpha_{i_1 0}(x_3, \ldots, x_n),$$

$$i_1 = 0, \ldots, N.$$

Acting in a similar way for $k = 1, \ldots, n-1$ arbitrary value, let's assume

$$\alpha_{i_1 \ldots i_k}(x_{k+1}, \ldots, x_n) =$$

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

It follows that for all $k = 2, \ldots, n-1$ values

$$\alpha_{i_1 \ldots i_{k-1}}(x_k, \ldots, x_n) = (\ldots((\alpha_{i_1 \ldots i_{k-1} N}(x_{k+1}, \ldots, x_n)x_k +$$

$$+\alpha_{i_1 \ldots i_{k-1}, N-1}(x_{k+1}, \ldots, x_n))x_k + \alpha_{i_1 \ldots i_{k-1}, N-2}(x_{k+1}, \ldots, x_n))x_k + \ldots)x_k +$$

$$+\alpha_{i_1 \ldots i_{k-1} 0}(x_3, \ldots, x_n), \quad i_1, \ldots, i_{k-1} = 0, \ldots, N. \tag{2.9}$$

recurrence relation is correct.

Provided that

$$\alpha_{i_1 \ldots i_{n-1}}(x_n) = \sum_{i_n=0}^{N} a_{i_1 \ldots i_n} x_n^{i_n} =$$

$$= ((\dots((a_{i_1\dots i_{n-1}N}x_n + a_{i_1\dots i_{n-1},N-1})x_n + a_{i_1\dots i_{n-1},N-2})x_n + \dots$$

$$\dots + a_{i_1\dots i_{n-1}2})x_n + a_{i_1\dots i_{n-1}1})x_n + \alpha_{i_1\dots i_{n-1}0},$$

$$i_1, \dots, i_{n-1} = 0, \dots, N. \tag{2.10}$$

The relations (2.8)–(2.10) define the generalized Horner scheme for polynomial evaluation (1.1). In many cases this scheme allows to reduce computation. As will be seen in section 3, in the general case scheme (2.9) and (2.10) admits decomposing, and the validity of the following statement is established by simple computation.

The number of operations for polynomial evaluation (1.1) according to the generalized Horner scheme is equivalent to the number of operations for evaluation of not more than

$$\mathcal{N} = \sum_{i=0}^{n-1} (N+1)^i \tag{2.11}$$

one-dimensional polynomials of the form (2.1) by the Horner scheme.

3 Algorithm

For the purpose of implementation of the above-described generalized Horner scheme we will use an n-steps algorithm focused on the computation process decomposition.

Step 1. Here we evaluate the functions $x_n \to \alpha_{i_1\dots i_{n-1}}(x_n)$ by (2.10). Evidently, each of these functions can be constructed independently. That's why the process of construction can be decomposed naturally into parallel evaluations of all these functions.

Steps $2, \dots, n-1$. At these steps for $k = n-1, \dots, 2$ by (2.9) we evaluate the functions $(x_k, \dots, x_n) \to \alpha_{i_1\dots i_{k-1}}(x_k, \dots, x_n)$. Like at step 1, each of these functions can be built independently, i.e. here the process of construction can be, in the general case, also decomposed naturally into parallel evaluations of all these functions.

Step n. This step is the simplest one: at this step we evaluate the functions $\mathbf{x} \to p_n(\mathbf{x})$ by (2.8).

Remark 2. Evidently, the higher the values of n and N are, the greater effect is produced by the computation process decomposition on parallel nodes. In other words, the generalized Horner scheme efficiency for the nearly complete polynomials can be additionally boosted by the natural decomposition of the computation process.

Remark 3. In the course of operation the necessity of polynomial evaluation (1.1) at various $\mathbf{x}$ vector values can occur. Obviously, every such computation doesn't depend on others and these computations can be performed in parallel on independent nodes. As it will be shown below in section 4, in some cases the latter ensures essential simplification of the classic problem of finding the zeros of quasipolynomials (see ex. 2).

4 Evaluation of the Simplest Quasipolynomial

As one more example of the use of the generalized Horner scheme let's consider the problem of classical quasipolynomials evaluation.

Example 2. Let's consider

$$\tilde{p}_2(t) = p_2(t, \exp t) \tag{4.1}$$

function of t real variable where p_2 is the polynomial of $x = t$ and $y = \exp t$ variables described in the example 1. The study of zeros of this function (in general terms, of a quasipolynomial) is essential to the Lyapunov theory of stability of motion (see [Pontryagin, 1955]). Let's here turn to more simple problem of its evaluation.

Considering that p_2 is polynomial of $x = t$ and $y = \exp t$ variables then acting by analogy to (2.5) and (2.6), we can write

$$\tilde{p}_2(t) = (\dots((\tilde{\alpha}_N(t)\exp t + \tilde{\alpha}_{N-1}(t))\exp t + \tilde{\alpha}_{N-2}(t))\exp t + \dots)\exp t + \tilde{\alpha}_0(t), \tag{4.2}$$

where

$$\tilde{\alpha}_i(t) = (\dots((a_{iN}t + a_{i,N-1})t + a_{i,N-2})t + \dots)t + a_{i0}, \quad i = 0, \dots, N, \tag{4.3}$$

which defines the generalized Horner scheme for the quasipolynomial evaluation.

Thus according to (4.2) and (4.3) relations it is easy to notice that the generalized Horner scheme described in the example 1 can be in some cases used efficiently use for classical quasipolynomial evaluation.

In order to verify this, let's suppose that the roots of the quasipolynomial (4.3) are localized within a certain interval $[a, b]$. Let's assume some natural number k and state

$$\varepsilon = \frac{(b-a)}{k}$$

and

$$t_i = i\varepsilon, \quad i = 0, \dots, k. \tag{4.4}$$

It seems clear that evaluation of the quasipolynomial (4.1) at any point of the set (4.4) can be performed independently of the others, i.e. in parallel, for instance, on independent nodes. Thus, providing there are enough independent nodes, finding the zeros of the quasipolynomial (4.1) requires no great timing budgets. Therewith, as it was stated in Ex. 1, performing the evaluation by means of the generalized Horner scheme allows, in many cases, to reduce the number of machine operations. The latter reduces the error and computation time.

5 Discussion

The distinctive feature of the generalized Horner scheme described herein is the obvious possibility of computation process decomposition.

According to section 2, the computation process decomposition is always possible in case of availability the additive component distinctive in the polynomial (1.1). This condition is simple and natural because its failure means that in practice (1.1) represents a direct analogy of a one-dimensional polynomial.

In other words in almost all real situations the algorithm representing the generalized Horner scheme admits decomposition. Moreover, the level of the algorithm decomposition is easily rated.

Actually, let's consider for simplicity only the first stage of the algorithm. At step 1 of this stage the symbolic construction of $x_n \to \alpha_{i_1 \dots i_{n-1}}(x_n)$ function is realized provided that each of this functions can be constructed independently. A number of these functions as it is easy to see can get to $(N+1)^{(n-1)}$. It follows that the computation process at this step can in general terms decompose into $(N+1)^{(n-1)}$ of parallel computation processes.

In a similar way, at step 2 of the stage 1 the computation process can be decomposed into $(N+1)^{(n-2)}$ of parallel computation processes and so on.

Thus, according to remark 1, if the total number of one-dimensional polynomials equals to $\mathcal{N}$ (see 2.11), then the total number of machine operations of addition and multiplication is $2N\mathcal{N}$. Proceeding from the above, if there are $(N+1)^{(n-1)}$ processors, then the total number of parallel addition and parallel multiplication operations, featured during the decomposition of the parallel computation processes, will amount to $2Nn$, where $n << \mathcal{N}$ in the general case.

It should be noted that the number $(N+1)^{(n-1)}$ can be rather large. That's why the implementation of the above-described algorithm is focused, in the general case, on a distributed computer environment featuring a rather large number of processors.

Acknowledgements

This work was supported by Russian Scientific Foundation, Project 16-11-10352.

References

[Afanasiev, 2005] Afanasiev, A.P. (2005). *Extension of trajectory toward optimal control.* Moscow: KomKniga/URSS.

[Afanasiev & Tarasov, 2006] Afanasiev, A.P., Tarasov, A.S. (2006). Quasi-analytic system solution of differential equations with polynomial right side. *Coll. of works of ISA RAN Problems of estimation in a distributed environment: distributed systems, communications systems, symbolic models and optimization, (25)*, 165–183.

[Afanasiev & Dzyuba, 2015a] Afanas'ev, A.P., Dzyuba, S.M. (2015). Method for constructing approximate analytic solutions of differential equations with a polynomial right-hand side. *Comp. Math. and Math. Phys., 55(10)*, 1665–1673.

[Afanasiev & Dzyuba, 2015b] Afanas'ev, A.P., Dzyuba, S.M. (2015). Construction of minimal sets of differential equations with a polynomial right hand-side. *Diff. Eqns, 51(11)*, 1403–1412.

[McCracken & Dorn, 1965] McCracken, D., Dorn, W. (1965). *Numerical methods and FORTRAN programming.* New York: John Willey and Sons.

[Levitin, 2006] Levitin, A.V. (2006). *Algorithms. Introduction into development and analysis.* Moscow: Williams.

[Cormen et al., 2009] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2009). *Intorduction to the algorithms.* New York: MIT Press and McGraw-Hill.

[Dowling, 1990] Dowling M. L. (1990). A fast parallel Horner algorithm. *SIAM Journal on Computing, 19(1)*, 133–142.

[Afanasiev et al., 2016] Afanasiev, A.P., Dzyuba, S.M., Emelyanova, I.I., Ramazanov, A.B. (2016). Optimal control with feedback of some class of nonlinear systems via quadratic criteria. *Appl. Comp. Math., 19(1)*, 78–87.

[Pontryagin, 1955] Pontryagin, L.S. (1955). On the zeroes of some elementary transcendental functions. *Amer. Math. Soc. Transl. Ser., 2(1)*, 95–110.