Aleksandr CARIOW, Galina CARIOWA

West Pomeranian University of Technology, Szczecin

An algorithm for complex-valued vector-matrix multiplication

***Abstract**. In this note we present the algorithm for vector-matrix product calculating for vectors and matrices whose elements are complex numbers.*

***Streszczenie.** W artykule został przedstawiony zracjonalizowany algorytm wyznaczania iloczynu wektorowo-macierzowego, dla danych będących liczbami zespolonymi. Proponowany algorytm wyróżnia się w stosunku do metody naiwnej zredukowaną złożonością multiplikatywną. Jeśli metoda naiwna wymaga wykonania 4MN mnożeń oraz 2M(2N-1) dodawań liczb rzeczywistych to proponowany algorytm wymaga tylko 3MN mnożeń oraz N+M(5N-1) dodawań. (**Zracjonalizowany algorytm wyznaczania iloczynu wektorowo-macierzowego, dla danych będących liczbami zespolonymi**)*

Słowa kluczowe: iloczyn wektorowo-macierzowy, liczby zespolone, szybki algorytm.
Keywords: vector-matrix product, complex-valued data, Fast algorithm.

Introduction

Most of the computation algorithms which are used in digital signal, image and video processing, computer graphics and vision and high performance supercomputing applications have vector-matrix multiplication as the kernel operation [1, 2]. For this reason, the rationalization of these operations is devoted to numerous publications [3-8]. In some cases, elements of the multiplied matrices and vectors are complex numbers [9]. Some interesting solutions related to the rationalization of the complex-valued matrix transformations have already been obtained [10-13]. However, the rationalized algorithms of complex-valued vector-matrix multiplications has not yet been published. For this reason, in this paper, we propose such algorithm.

The complex-valued vector-matrix product may be defined as:

$$\mathbf{Z}_{M\times 1} = \mathbf{Y}_{M\times N}\mathbf{X}_{N\times 1} \tag{1}$$

where $\mathbf{X}_{N\times 1} = [x_0, x_1, ..., x_{N-1}]^{\mathrm{T}}$ - N-dimensional input vector, $\mathbf{Z}_{N\times 1} = [z_0, z_1, ..., z_{M-1}]^{\mathrm{T}}$ - N-dimensional output vector and

$$\mathbf{Y}_{M\times N} = \begin{bmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,N-1} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M-1,0} & y_{M-1,1} & \cdots & y_{M-1,N-1} \end{bmatrix},$$

where $x_n = a_n + jb_n$, $z_m = e_m + jf_m$, $y_{m,n} = c_{m,n} + jd_{m,n}$ and $n = 0,1,...,N-1$, $m = 0,1,...,M-1$.

In this expression a_n, b_n, $c_{m,n}$, $d_{m,n}$, e_m and f_m are real numbers and j is the imaginary unit, satisfying $j^2 = -1$.

It is well known, that complex multiplication requires four real multiplications and two real additions, because:

$$(a + jb)(c + jd) = ac - bd + j(ad + bc). \tag{2}$$

Consequently, we can observe that the schoolbook computation of (1) requires MN complex multiplications ($4MN$ real multiplications) and $2M(2N-1)$ real additions. However, it is possible to perform the complex multiplication with three real multiplication and five real additions, because [13]:

$$(a + jb)(c + jd) = ac - bd + j[(a+b)(c+d) - ac - bd]. \tag{3}$$

Taking into account this fact the expression (1) can be calculated using the $3MN$ multiplications and $M(7N-2)$ additions of real numbers. The proposed algorithm has the same number of real multiplications. The number of real additions, however, is less. Below we consider the synthesis algorithm in detail.

Development of the algorithm

First, we rewrite the vector $\mathbf{X}_{N\times 1} = [x_0, x_1, ..., x_{N-1}]^{\mathrm{T}}$ in a following form:

$$\mathbf{X}_{2N\times 1} = [a_0, b_0, a_1, b_1, ..., a_{N-1}, b_{N-1}]^{\mathrm{T}}$$

and vector $\mathbf{Z}_{M\times 1} = [z_0, z_1, ..., z_{M-1}]$ - in a following form:

$$\mathbf{Z}_{2M\times 1} = [e_0, f_0, e_1, f_1, ..., e_{M-1}, f_{M-1}]^{\mathrm{T}}.$$

Next, we introduce some auxiliary matrices:

$$\mathbf{P}^{(0)}_{N(2M+1)\times 2N} = \left[\begin{array}{c} \mathbf{1}_{M\times 1} \otimes \mathbf{I}_{2N} \\ \hline \mathbf{I}_N \otimes \overline{\mathbf{1}}_{1\times 2} \end{array}\right],$$

where $\mathbf{1}_{M\times N}$ - is an $M \times N$ matrix of ones (a matrix where every element is equal to one) and $\overline{\mathbf{1}}_{1\times 2} = [1 \mid -1]$,

$$\mathbf{P}^{(1)}_{3MN\times N(2M+1)} = \mathbf{I}_{2MN} \oplus (\mathbf{1}_{M\times 1} \otimes \mathbf{I}_N),$$

$\mathbf{I}_N$ - is an identity $N \times N$ matrix and signs „$\otimes$" and „$\oplus$" – denote tensor product and direct sum of two matrices respectively [14],

$$\mathbf{A}^{(0)}_{3M\times 3MN} = (\mathbf{I}_M \otimes (\mathbf{1}_{1\times N} \otimes \mathbf{I}_2)) \oplus (\mathbf{I}_M \otimes \mathbf{1}_{1\times N}),$$

$$\mathbf{A}^{(1)}_{2M\times 3M} = [(\mathbf{I}_M \otimes \mathbf{I}_2) \mid (\mathbf{I}_M \otimes \mathbf{1}_{2\times 1})].$$

Using the above matrices the computational procedure for calculating the scalar product can be written as follows:

$$\mathbf{Z}_{2M\times 1} = \mathbf{A}^{(1)}_{2M\times 3M}\mathbf{A}^{(0)}_{3M\times 3MN}\mathbf{D}_{3MN} \times \tag{4}$$
$$\times \mathbf{P}^{(1)}_{3MN\times N(2M+1)}\mathbf{P}^{(0)}_{N(2M+1)\times 2N}\mathbf{X}_{2N\times 1},$$
$$\mathbf{D}_{3MN} = \mathbf{D}'_{2MN} \oplus \mathbf{D}''_{MN},$$

$$\mathbf{D}'_{2MN} = diag\{\widetilde{\mathbf{D}}_{2N}^{(0)}, \widetilde{\mathbf{D}}_{2N}^{(1)}, ..., \widetilde{\mathbf{D}}_{2N}^{(M-1)}\},$$

$$\widetilde{\mathbf{D}}_{2N}^{(m)} = \bigoplus_{n=0}^{N-1} (s'_{m,n} \oplus s''_{m,n})$$

where $s'_{m,n} = c_{m,n} - d_{m,n}$ and $s''_{m,n} = c_{m,n} + d_{m,n}$,

$$\mathbf{D}''_{MN} = diag\{\mathbf{d}_N^{(0)}, \mathbf{d}_N^{(1)}, ..., \mathbf{d}_N^{(M-1)}\},$$

$$\mathbf{d}_N^{(m)} = diag(d_{m,0}, d_{m,1}, ..., d_{m,N-1}),\ m = 0,1,...,M-1.$$

If the elements of $\mathbf{D}_{3MN}$ placed vertically without disturbing the order and written in the form of the vector $\mathbf{D}_{3MN\times 1}$, then they can be calculated using the following computational procedure:

(5) $$\mathbf{D}_{3MN\times 1} = \mathbf{P}''_{3MN} \widetilde{\mathbf{H}}_{3MN} \mathbf{P}'_{3MN\times 2MN} \mathbf{Y}_{2MN\times 1},$$

where

$$\widetilde{\mathbf{H}}_{3MN} = (\mathbf{I}_{MN} \otimes \mathbf{H}_2) \oplus \mathbf{I}_{MN},\ \mathbf{P}''_{3MN} = (\mathbf{I}_{MN} \otimes \mathbf{J}_2) \oplus \mathbf{I}_{MN},$$

$$\mathbf{P}'_{3MN\times 2N} = \begin{bmatrix} \mathbf{I}_{2MN} \\ \hline \mathbf{I}_{MN} \otimes [0 \mid 1] \end{bmatrix},\ \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ \mathbf{J}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

$$\mathbf{Y}_{2MN\times 1} = [\mathbf{Y}_{2N}^{(0)}, \mathbf{Y}_{2N}^{(1)}, ..., \mathbf{Y}_{2N}^{(M-1)}]^{\mathrm{T}},$$

$$\mathbf{Y}_{2N}^{(m)} = [c_{m,0}, d_{m,0}, c_{m,1}, d_{m,1}, ..., c_{m,N-1}, d_{m,N-1}]^{\mathrm{T}}.$$

For example, let us consider the case of $N = 4$ and $M = 3$.

$$\mathbf{Z}_{6\times 1} = \mathbf{A}_{6\times 9}^{(1)} \mathbf{A}_{9\times 36}^{(0)} \mathbf{D}_{36} \mathbf{P}_{36\times 28}^{(1)} \mathbf{P}_{28\times 8}^{(0)} \mathbf{X}_{8\times 1},$$

$$\mathbf{X}_{8\times 1} = [a_0, b_0, a_1, b_1, a_2, b_2, a_3, b_3]^{\mathrm{T}},$$

$$\mathbf{Z}_{2M\times 1} = [e_0, f_0, e_1, f_1, e_2, f_2]^{\mathrm{T}},$$

$$\mathbf{P}_{28\times 8}^{(0)} = \begin{bmatrix} \mathbf{1}_{3\times 1} \otimes \mathbf{I}_8 \\ \hline \mathbf{I}_4 \otimes \bar{\mathbf{1}}_{1\times 2} \end{bmatrix},\ \mathbf{P}_{36\times 28}^{(1)} = (\mathbf{I}_{24} \oplus (\mathbf{1}_{3\times 1} \otimes \mathbf{I}_4)).$$

$$\mathbf{D}_{36} = \mathbf{D}'_{24} \oplus \mathbf{D}''_{12},\ \mathbf{D}'_{24} = diag\{\widetilde{\mathbf{D}}_8^{(0)}, \widetilde{\mathbf{D}}_8^{(1)}, \widetilde{\mathbf{D}}_8^{(2)}\},$$

$$\widetilde{\mathbf{D}}_8^{(0)} = \bigoplus_{n=0}^{3} (s'_{0,n} \oplus s''_{0,n}) = diag\begin{bmatrix} c_{0,0} - d_{0,0} \\ \hline c_{0,0} + d_{0,0} \\ \hline c_{0,1} - d_{0,1} \\ \hline c_{0,1} + d_{0,1} \\ \hline c_{0,2} - d_{0,2} \\ \hline c_{0,2} + d_{0,2} \\ \hline c_{0,3} - d_{0,3} \\ \hline c_{0,3} + d_{0,3} \end{bmatrix},$$

$$\mathbf{A}_{9\times 36}^{(0)} = (\mathbf{I}_3 \otimes (\mathbf{1}_{1\times 4} \otimes \mathbf{I}_2)) \oplus (\mathbf{I}_3 \otimes \mathbf{1}_{1\times 4}),$$

$$\widetilde{\mathbf{D}}_8^{(1)} = \bigoplus_{n=0}^{3} (s'_{1,n} \oplus s''_{1,n}) = diag\begin{bmatrix} c_{1,0} - d_{1,0} \\ \hline c_{1,0} + d_{1,0} \\ \hline c_{1,1} - d_{1,1} \\ \hline c_{1,1} + d_{1,1} \\ \hline c_{1,2} - d_{1,2} \\ \hline c_{1,2} + d_{1,2} \\ \hline c_{1,3} - d_{1,3} \\ \hline c_{1,3} + d_{1,3} \end{bmatrix},$$

$$\widetilde{\mathbf{D}}_8^{(2)} = \bigoplus_{n=0}^{3} (s'_{2,n} \oplus s''_{2,n}) = diag\begin{bmatrix} c_{2,0} - d_{2,0} \\ \hline c_{2,0} + d_{2,0} \\ \hline c_{2,1} - d_{2,1} \\ \hline c_{2,1} + d_{2,1} \\ \hline c_{2,2} - d_{2,2} \\ \hline c_{2,2} + d_{2,2} \\ \hline c_{2,3} - d_{2,3} \\ \hline c_{2,3} + d_{2,3} \end{bmatrix},$$

$$\mathbf{A}_{6\times 9}^{(1)} = \left[\mathbf{I}_6 \mid (\mathbf{I}_3 \otimes \mathbf{1}_{2\times 1})\right],\ \mathbf{D}''_{12} = diag\{\mathbf{d}_4^{(0)}, \mathbf{d}_4^{(1)}, \mathbf{d}_4^{(2)}\},$$

$$\mathbf{d}_4^{(0)} = diag(d_{0,0}, d_{0,1}, d_{0,2}, d_{0,3}),$$

$$\mathbf{d}_4^{(1)} = diag(d_{1,0}, d_{1,1}, d_{1,2}, d_{1,3}),$$

$$\mathbf{d}_4^{(2)} = diag(d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}),$$

$$\mathbf{D}_{36\times 1} = \mathbf{P}''_{36} \widetilde{\mathbf{H}}_{36} \mathbf{P}'_{36\times 24} \mathbf{Y}_{24\times 1},\ \widetilde{\mathbf{H}}_{36} = (\mathbf{I}_{12} \otimes \mathbf{H}_2) \oplus \mathbf{I}_{12},$$

$$\mathbf{P}'_{36\times 24} = \begin{bmatrix} \mathbf{I}_{24} \\ \hline \mathbf{I}_{12} \otimes [0 \mid 1] \end{bmatrix},\ \mathbf{P}''_{36} = (\mathbf{I}_{12} \otimes \mathbf{J}_2) \oplus \mathbf{I}_{12},$$

$$\mathbf{Y}_{24\times 1} = [\mathbf{Y}_8^{(0)}, \mathbf{Y}_8^{(1)}, \mathbf{Y}_8^{(2)}]^{\mathrm{T}},$$

$$\mathbf{Y}_8^{(0)} = [c_{0,0}, d_{0,0}, c_{0,1}, d_{0,1}, c_{0,2}, d_{0,2}, c_{0,3}, d_{0,3}]^{\mathrm{T}},$$
$$\mathbf{Y}_8^{(1)} = [c_{1,0}, d_{1,0}, c_{1,1}, d_{1,1}, c_{1,2}, d_{1,2}, c_{1,3}, d_{1,3}]^{\mathrm{T}},$$
$$\mathbf{Y}_8^{(2)} = [c_{2,0}, d_{2,0}, c_{2,1}, d_{2,1}, c_{2,2}, d_{2,2}, c_{2,3}, d_{2,3}]^{\mathrm{T}}.$$

The graph-structural model for realization of proposed algorithm is illustrated in Figure 1. In turn, Figure 2 shows a graph-structural model for computing elements of the matrix $\mathbf{D}_{3N}$ in accordance with the procedure (5). In this paper, the graph-structural models are oriented from left to right. Note [15] that the circles in these figures show the operation of multiplication by a real number (variable) inscribed inside a circle. Straight lines in the figures denote the operation of data transfer. At points where lines converge, the data are summarized. (The dash-dotted lines indicate the subtraction operation). We use the usual lines without arrows specifically so as not to clutter the picture. In turn, the rectangles indicate the matrix-vector multiplications with the 2×2 Hadamard matrices.

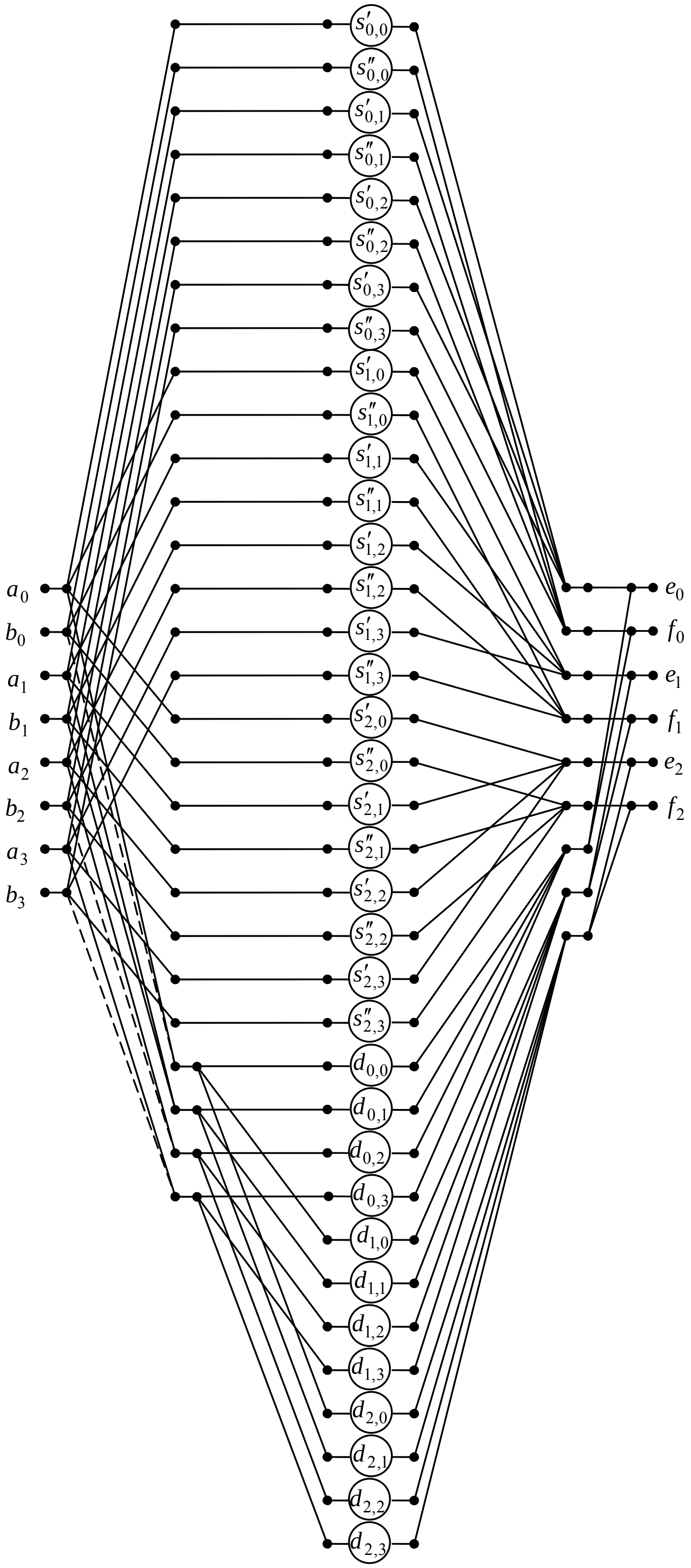

Fig.1. Graph-structural model of complex-valued vector-matrix product computation in according to procedure (4) for N=4, M=3

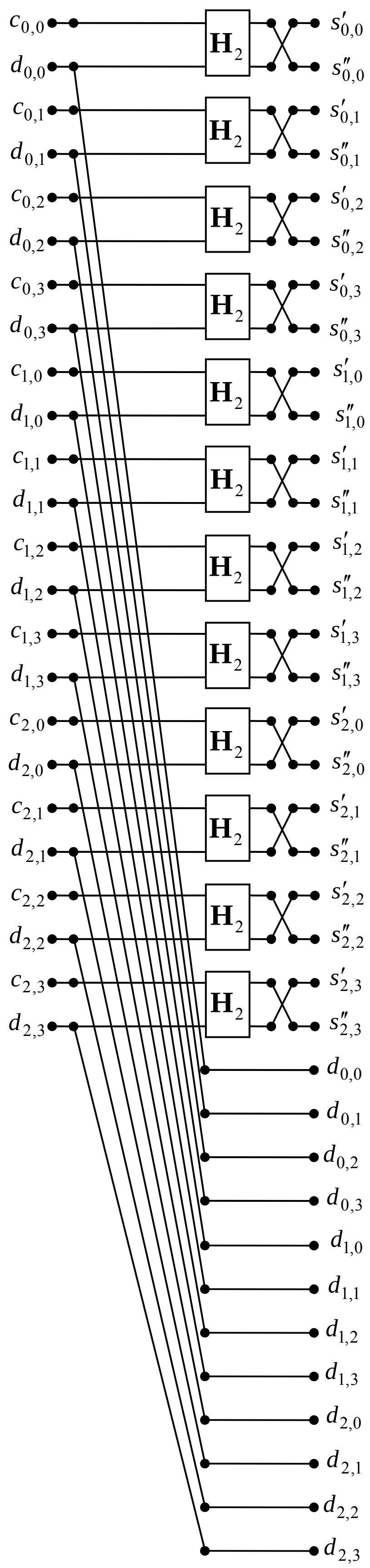

Fig.2. The graph-structural model of computation process organization for vector $\mathbf{D}_{36\times1}$ elements calculating corresponding to (5) for N=4, M=3.

Discussion of computational complexity

Described algorithm for computing the vector-matrix product of complex-valued data requires $3MN$ multiplications and $N+M(5N-1)$ additions of real numbers. Compared to the schoolbook way of computing it gives 25% reduction in multiplicative complexity due to the increase by 50% of the additive. Table 1 shows the number of multiplications and additions of real numbers for the three different methods: for schoolbook method in according to formula (1), for creative method in according to formula (2), and for proposed algorithm in according to procedure (3). As can be seen, the developed algorithm has the same multiplicative complexity as the creative method (1), but has a lower additive complexity. It should be noted that in some applications, the matrix elements are constants [16-18]. In this case the diagonal matrix $\mathbf{D}_{3MN}$ elements can be calculated and stored in the memory of the computing unit in advance. Then the number of additions in the implementation of the proposed algorithm is even more reduced. As a result, compared to the naive method of computing the number of multiplications is reduced by 25% and the number of additions remains almost the same. This means that the proposed algorithm has some advantages and deserves the right to exist.

Table 1. Estimates for the computational complexity of the discussed methods of calculation complex-valued vector-matrix products

M/N	Multiplicative complexity			Additive complexity		
	(1)	(2)	(3)	(1)	(2)	(3)
1/2	8	6	6	6	12	11
1/3	12	9	9	10	19	17
1/4	16	12	12	14	26	23
1/5	20	15	15	18	33	29
1/6	24	18	18	22	40	35
2/2	16	12	12	12	24	20
2/3	24	18	18	20	38	31
2/4	32	24	24	28	52	42
2/5	40	30	30	36	66	53
2/6	48	36	36	44	80	64
3/2	24	18	18	18	36	29
3/3	36	27	27	20	57	45
3/4	48	36	36	42	78	61
3/5	60	45	45	54	99	77
3/6	72	54	54	66	120	93
4/2	32	24	24	24	48	38
4/3	48	36	36	40	76	59
4/4	64	48	48	56	104	80
4/5	80	60	60	72	132	101
4/6	96	72	72	88	160	122
5/2	40	30	30	30	60	47
5/3	60	45	45	50	95	73
5/4	80	60	60	70	130	99
5/5	100	75	75	90	165	125
5/6	120	90	90	110	200	151
6/2	48	36	36	36	72	56
6/3	72	54	54	60	114	87
6/4	96	72	72	84	156	118
6/5	120	90	90	108	198	149
6/6	144	108	108	132	240	180

Concluding remarks

The article presents an effective algorithm for complex-valued vector-matrix product computation. The idea of constructing the algorithm uses the opportunity to calculate the product of complex numbers with three multiplications and five additions of real numbers. However, in comparison with the direct use of the realization of products of complex numbers calculated by the creative way in accordance with (2), the proposed algorithm has a lower additive complexity. This allows the effective use of parallelization computational process of matrix multiplication and results in a reduction in computation time.

Wydanie publikacji zrealizowano przy udziale środków finansowych otrzymanych z budżetu Województwa Zachodniopomorskiego.

REFERENCES

[1] Blahut R.E., Fast algorithms for digital signal processing, Addison-Wesley Publishing company, Inc. 1985.
[2] Pratt. W. K. Digital Image Processing (Second Edition), John Wiley & Sons, New York, 1991.
[3] Goto, K. and van de Geijn, R. Anatomy of high-performance matrix multiplication. ACM Trans. Math. Softw. 2008, v. 34, No. 3, pp.1-25.
[4] Valsalam V., Skjellum A., A framework for high-performance matrix multiplication based on hierarchical abstractions, algorithms and optimized low-level kernels, Concurrency and Computation Practice and Experience. 2002, v. 14, No.10, Publisher: Wiley Online Library, pp. 805-839
[5] Varman P.J., Ramakrishnan I.V., "Synthesis of an Optimal Family of Matrix Multiplication Algorithms on Linear Arrays," IEEE Transactions on Computers, vol. 35, no. 11, pp. 989-996, Nov. 1986
[6] Gunnels John A., Gustavson Fred G., Henry Greg M., van de Geijn Robert A., A family of high-performance matrix multiplication algorithms, Proceeding PARA'04 Proceedings of the 7th international conference on Applied Parallel Computing: state of the Art in Scientific Computing, v. 2073, No. 2, Publisher: Springer-Verlag, pp. 51-60.
[7] Delorimier M. and Dehon A., “Floating-Point sparse matrix-vector multiply for FPGAs,” in Proc. of the IEEE Intl. Symposium on Field Programmable Gate Arrays, 2005, pp. 75-85.
[8] Fujimoto N., “Dense matrix-vector multiplication on the CUDA architecture,” Parallel Processing Letters, 2008, vol. 18, No. 4, pp. 511-530.
[9] Barazesh B., Michalina J., and Picco A. A VLSI signal processor with complex arithmetic capability. *IEEE Transactions on Circuits and Systems*, May 1988, 35(5) pp.495–505.
[10] Fam A. T., Efficient complex matrix multiplication, *IEEE Transactions on Computers*, 1988, v. 37, no. 7, pp. 877-879.
[11] Connolly, F.T., Yagle, A.E., Fast algorithms for complex matrix multiplication using surrogates, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1989, v. 37 no. 6, pp. 938 – 939.
[12] Li, Guoqiang, Liu, Liren, Complex-valued matrix-vector multiplication using twos complement representation. *Optics Communications*, v. 105, no. 3-4, pp. 161-166.
[13] Knuth D.E., The Art Of Computing Programing, Volume 2, Semi-numerical Algorithms, Addison-Wesley, Reading, MA, USA, Sec-ond Ed., 1981.
[14] Regalia P. A. and Mitra K. S., Kronecker Products, Unitary Matrices and Signal Processing Applications, *SIAM Review*. 1989, v. 31, no. 4, pp. 586-613.
[15] Ţariov A., Algorithmic aspects of calculation rationalization in digital signal processing. West Pomeranian University of Technology, Szczecin, 2011 (in Polish).
[16] Gustafsson O., Ohlsson H., and Wanhammar L., Low-complexity constant coefficient matrix multiplication using a minimum spanning tree approach, Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004, June 9 - 11, 2004, Espoo, Finland, pp.141-144.
[17] Boullis, N., Tisserand, A., Some optimizations of hardware multiplication by constant matrices, *IEEE Transactions on Computers*, 2005, v. 54 no 10, pp. 1271 – 1282.
[18] Kinane A., Muresan V., Towards an optimized VLSI design algorithm for the constant matrix multiplication problem. In: Proc. IEEE International Symposium on Circuits and Systems ISCAS-2006, 2006, pp. 5111 – 5114.

***Authors**: D.Sc. Aleksandr Cariow, PhD Galina Cariowa, Department of Computer Architectures and Telecommunications, Faculty of Computer Sciences, West Pomeranian University of Technology, Szczecin, ul. Żołnierska 51, 71-210 Szczecin, E-mail: atariov@wi.zut.edu.pl, gtariova@wi.zut.edu.pl*