Fast Algorithms: a Science, an Art, and a Craft

Aleksandr Cariow
West Pomeranian University of Technology Szczecin
Faculty of Computer Science and Information Technology
Szczecin, Poland
acariow@wi.zut.edu.pl

***Abstract*— This paper offers the strategies for the synthesis of fast algorithms for computing the matrix-vector products. The concrete example of synthesis of fast algorithm for matrix-vector multiplication is demonstrated in the speech. The example offered allows to track all the stages of construction of the algorithm which was rationalized from the point of view of number multiplication minimization. It is claimed that the process of constructing fast algorithms is both a science and an art and a craft.**

***Index Terms*—Fast algorithms, matrix-vector multiplication, digital signal processing.**

I. Introduction

Fast algorithms are a field of computer science that studies algorithms for implementing various kinds of computational problems using as few mathematical (and other) operations as possible. The development of the theory and practice of constructing fast algorithms has long been in direct dependence on progress in the design and production of electronic computing equipment. We can safely say that it is the imperfection of computers of the first, second and third generations that contributed to the emergence of fast algorithms. For the sake of justice, it should be noted that the instruction system of the first generation computers contained all the full necessary set of commands required for the implementation of mathematical calculations. However, if such operations as addition and subtraction were performed during one clock cycle, then, for example, the multiplication command required the implementation of a rather long sequence of addition and shift operations in accordance with the rules for multiplying binary numbers. This sequence of operations is usually hard coded on the ferrite rings in the block of the computer's ROM and stored there as a firmware. Clearly, the implementation of such a microprogram required much more time than the execution of the addition operation or memory access operation. Thus, it turned out that the time of realization of multiplication became the main factor limiting the speed of solving applied problems. This fact stimulated the search and development of methods and various algorithmic tricks that allow to reduce the number of multiplication in the implementation of certain numerical methods. It is within the framework of this direction that fast algorithms of digital data processing are developed and applied [1].

II. A Bit of History of Fast Algorithms

The ancestors of fast computations with a certain degree of conventionality can be considered German mathematicians K. Runge and K. Gauss, who were looking for ways to reduce the number of arithmetic operations in carrying out various kinds of mathematical calculations [1]. Well-known, for example, Gauss's algorithmic trick, which allows to calculate the product of two complex numbers with just three multiplications and five additions of real numbers [2]. However, as the beginning of the era of the most notable achievements in the field of fast computations, one can accept the emergence of the "divide and conquer" method developed in 1960 by Anatoly Alekseevich Karatsuba, demonstrated by him on the example of synthesis of a new efficient algorithm for fast multiplication of polynomials [3, 4].

The next revolutionary event in the scientific world was the development and publication in 1965 of the Fast Fourier Transform (FFT) algorithm of the authorship of J. Cooley and J. Tukey, obtained in essence also using the "divide and conquer" method. The emergence of this algorithm was a turning point in the development of the theory and practice of digital signal and image processing, as well as a number of other areas of science and technology, since it was possible to drastically reduce the number of arithmetic operations in calculating the discrete Fourier transform [5].

Later numerous "fast" algorithms for calculation of convolutions and correlations of digital sequences, discrete transformations in various orthogonal bases, and many others were appeared [6-8]. Among other things, Strassen's and Winograd's algorithms for multiplying the matrices, the Toom-Cook and Fürer's algorithms for multiplying large integers, and many others [9], which have become the "classics" of fast computations, should be singled out.

III. Explanation

The main advantage of all the "fast" algorithms was a radical reduction in the multiplication operations (reducing the multiplicative complexity) in comparison with the "naive" algorithms. However, in a number of cases, a decrease in the number of multiplication operations leads to an increase in the number of additions (additive complexity) and almost always to an increase in the complexity of controlling the calculation process, as well as to an increase in data transfer operations, which previously no one paid much attention to because of their insignificant, in comparison with multiplication, execution time. With the development of the production

technology of the microelectronic base of electronic computers, as well as with the appearance of VLSIs with built-in hardware multipliers that allow performing a multiplication command during one clock cycle, the value of fast algorithms has been somewhat diminished. Suddenly it turned out that the reduction in multiplications in fast algorithms causing the growth of addition operations and data transfer operations under conditions when the execution time of these operations is comparable may also have a negative effect.

Practice has shown that, at least in a number of cases, "naive" approaches, based on time-consuming in terms of the number of arithmetic operations performed, but more simple in terms of organization of calculation and implementing of data addressing procedures, may be more effective than their "fast" modifications. This allowed all sorts of dilettantes and skeptics to assert about the further inexpediency of searching and applying algorithmic solutions that reduce the computational complexity of mathematical calculations.

It should nevertheless be noted that in fact, in the case when a computer or other computing device already contains a built-in hardware multiplier, reducing the number of multiplication operations due to a disproportionate increase in additions can lead to negative consequences. Nevertheless, in the design of specialized processors, especially processors with parallelization of computations in which a number of parallel multiplying units are supposed to exist, the problem of minimizing the number of these blocks remains to be actual. This is because if the hardware complexity of an binary adder increases linearly with operand size, then the hardware complexity of a binary multiplier grows quadratically. The multiplier, compared to the adder, occupies considerably more space on the crystal, consumes much more energy, and releases much more heat. It is clear that the developer of such a processor will strive to ensure that its structure contains as few blocks of multiplication as possible. In this case, the searching for algorithmic solutions leading to a reduction in hardware costs is extremely topical. From this point of view, the development of fast algorithms is economically justified and technically feasible.

It should be noted that there is still no universal methodology for designing fast algorithms. The most famous and interesting solutions were obtained on the basis of consideration of particular properties and unique features of specific problems. For example, the FFT algorithm was developed by taking into account the properties of periodicity and multiplicativity of discrete exponential functions, the algorithm of fast cyclic convolution - due to the proof that the convolution of two sequences can be calculated as the product of the FFT coefficients of these sequences.

Anyway, the development of a fast algorithm requires the developer to have a deep understanding of the problem to be solved, as well as broad theoretical knowledge. Such a state of affairs may cause difficulties for engineering and technical personnel and specialists who have rich practical experience, but do not have sufficient theoretical training, and in some cases even induce unwillingness to independently develop such algorithms.

Nevertheless, it should be recognized that the process of creating a fast algorithm is extremely interesting and creative. Much here depends not only on the depth of knowledge and the level of theoretical preparation of the developer, but also on his intuition and ingenuity. Not the least role is played also by the accumulated experience and the availability of skills in solving such problems. Therefore, we can state with full confidence that the design of fast algorithms is a science, an art, and a craft.

IV. Conclusion

The report discusses a simple and practical approach [10-11] to the development of fast algorithms for matrix-vector multiplications. The main attention is focused on this type of operations, since the need to quickly calculate vector-matrix products with different matrix nuclei arises when solving a huge number of applied problems related to digital processing of data in radios and sonars, navigation, telecommunications, image recognition, scene analysis, machine graphics, etc. Without pretending to be completely universal, the proposed approach yet has a sufficient set of properties that allow to unify, formalize and even to automate the development of fast algorithms in interactive mode [12]. With the help of the developed approach, a number of effective algorithmic solutions have been developed that make it possible to reduce the execution time for solving various applied problems and/or to simplify the structures of processing units [13-27].

Acknowledgment

I would like to thank Dr Tatiana Șestakov for a fruitful discussion of my views which enabled me to correct some of the assumptions and conclusions, as well as even more convinced in my own right.

References

[1] R. E. Blahut, Fast Algorithms for Digital Signal Processing. Addison-Wesley, 1985.

[2] W. K. Pratt, Digital Image Processing. 4th Edition, John Wiley & Sons, Inc. Hoboken, New Jersey, 2007.

[3] H.J. Nussbaumer, Fast Fourier Transform and Convolution Algorithms, Springer-Verlag, 1982.

[4] S. Burrus, T. W. Parks, J. F. Potts, DFT/FFT and Convolution Algorithms and Implementation, John Wiley & Sons, 1985

[5] R. Tolimieri, M. An, C. Lu, Algorithms for Discrete Fourier Transform and Convolution, Springer-Verlag, New York, 1989.

[6] T. K. Moon, Wynn C. Stirling, Mathematical methods and algorithms for signal processing. Prentice-Hall, 2000.

[7] P. A. Regaliat and S. K., Mitra, "Kronecker Products, Unitary Matrices and Signal Processing Applications", Siam Review, 31, No. 4, pp. 586-613, 1989.

[8] H. K. Garg, Digital Signal Processing Algorithms: Number Theory, Convolution, Fast Fourier Transforms, and Applications, CRC Press. 1998.

[9] G. Bi and Y. Zeng, Transforms and Fast Algorithms for Signal Analysis and Representations, Birkhäuser, Basel, 2004.

[10] A. Țariov, Algorithmic aspects of computing rationalization in Digital Signal processing. West Pomeranian University Press, (In Polish). 2012.

[11] A. Cariow, G. Cariowa, "Algorithm for multiplying two octonions". Radioelectronics and Communications Systems (Allerton Press, Inc. USA), 55, No 10, pp. 464-473, 2012.

[12] B. Andreatto, A. Cariow, "Automatic generation of fast algorithms for matrix-vector multiplication", International Journal of Computer Mathematics, 2017, v. 95, no. 3, pp.626-644.

[13] M Gliszczyński., A. Ţariov Szybki algorytm splotu kołowego dla $N = 2^m$. Pomiary Automatyka Kontrola, 2009, 55, nr 8, pp. 566-568.

[14] A. Ţariov, G. Ţariova, Aspekty algorytmiczne redukcji liczby bloków mnożących w układzie do obliczania iloczynu dwóch kwaternionów, Pomiary, Automatyka, Kontrola, 2010, 56, nr 7 str. 688-69.

[15] A. Ţariov, G. Ţariova, Aspekty algorytmiczne organizacji jednostki procesorowej do mnożenia liczb Cayleya, Elektronika: konstrukcje, technologie, zastosowania, 2010, 51, Nr 11, str. 104-108.

[16] A. Cariow, G. Cariowa, "An algorithm for complex-valued vector-matrix multiplication". Electrical Review, R 88, No 10b, pp. 213-216, 2012.

[17] D. Majorkowska-Mech, A. Cariow, "An algorithm for discrete fractional Hadamad transform with reduced arithmetical complexity". Electrical Review, R 88, No 11a, pp. 70-76, 2012.

[18] A. Cariow, M. Gliszczyński, "Fast algorithms to compute matrix-vector products for Toeplitz and Hankel matrices". Electrical Review, R 88, No 8, 166-171. 2012.

[19] A. Cariow, G. Cariowa, "An algorithm for fast multiplication of sedenions". Information Processing Letters, 113, pp. 324–331, 2013.

[20] A. Cariow, G. Cariowa, "An algorithm for multiplication of Dirac numbers". Journal of Theoretical and Applied Computer Science, 7, no. 4, pp. 26-34, 2013.

[21] A. Cariow, G. Cariowa, "Algorithmic tricks for reducing the complexity of FDWT/IDWT basic operations implementation". International Journal of Image, Graphics and Signal Processing, 6, no. 10, pp. 1-9, 2014.

[22] A. Cariow, G. Cariowa, "An Algorithm for Fast Multiplication of Pauli Numbers". Advances in Applied Clifford Algebras, Advances in Applied Clifford Algebras, March 2015, v. 25, no 1, pp. 53-63.

[23] A. Cariow, D. Majorkowska-Mech, "Fast algorithm for discrete fractional Hadamard transform". Numerical Algorithms, March 2015, v. 68, no 3, pp. 585-600.

[24] D. Majorkowska-Mech, A. Cariow, A Low-Complexity Approach to Computation of the Discrete Fractional Fourier Transform, Circuits, Systems, and Signal Processing, 2017 v. 36, no. 204. pp. 1-27

[25] A. Cariow, G. Cariowa, M. Witczak, A FPGA-Oriented Fully Parallel Algorithm for multiplying dual quaternions, Measurement Automation Monitoring, No 7, pp. (2015). Measurement Automation Monitoring, Jul. 2015, vol. 61, no. 07 pp. 370-372.

[26] A. Cariow, G. Cariowa, On the Multiplication of Biquaternions, Soft Computing in Computer and Information Science: Advances in Intelligent Systems and Computing, vol. 342, 2015, pp. 423-434.

[27] A. Cariow, G. Cariowa, M. Chicheva. Hardware-Efficient Schemes of Quaternion Multiplying Units for 2D Discrete Quaternion Fourier Transform Processors, Measurement Automation Monitoring, 2017, vol. 63, no 06, pp. 201-208.