Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 51

XXVI Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE 2016

Oddział Gdański PTETiS

IMPROVED MAGNITUDE ESTIMATION OF COMPLEX NUMBERS USING ALPHA MAX AND BETA MIN ALGORITHM

Robert SMYK[1], Maciej CZYŻAK[2]

1. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
   tel.: +48 58 347 13 32 e-mail: robert.smyk@pg.gda.pl
2. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
   tel.: +48 58 347 15 02 e-mail: maciej.czyzak@pg.gda.pl

Abstract: The paper presents an improved algorithm for calculating the magnitude of complex numbers. This problem, which is a special case of square rooting, occurs for example, in FFT processors and complex FIR filters. The proposed method of magnitude calculation makes use of the modified alpha max and beta min algorithm. The improved version of the algorithm allows to control the maximum magnitude approximation error by using an adequate number of approximation regions. In this way it is possible to reduce the maximum error to 3.95% for one region, and 0.24% and 0.06% for four and eight regions, respectively. This algorithm in its basic form requires only two multiplications by a constant and one addition which are preceded by the choice of greater of two arguments with respect to their absolute values. The improved version requires one general division to determine the proper approximation region. The algorithm implementation issues are considered in the accompanying paper.

Keywords: alpha max beta min algorithm, complex number magnitude.

1. INTRODUCTION

Calculation of magnitude of complex numbers is the common task in processing of information from devices that supply complex signals. As examples may serve Synthetic Aperture Radars (SAR) for imaging the surface [1], and spectrum analyzers [2]. In such devices usually the Fast Fourier Transform (FFT) processor is used that outputs the real and the imaginary part of i-th harmonic. In order to obtain spectrum samples, magnitudes of harmonics have to be calculated in real time. The circuit that performs this task must be able to operate at high speed and with a possibly small delay. These requirements can be fulfilled only by the limited category of algorithms. There is a constant need to search for new algorithms or modifications of existing ones that would be able to cope with contemporary and future tasks.

In many cases the FFT processor operates in the pipeline mode and magnitude should be calculated in the same manner. This means that magnitude of complex samples at the output of the FFT processor should be calculated within a prescribed time interval. Timing requirements usually exclude the use of a floating-point coprocessor, therefore an application specific hardware has to be designed and implemented. Alternatively, specific FPGAs can be applied as, for example, Xilinx xc7vx690t, that allows to attain maximum pipelining frequency of 281 MHz for the 8192 point FFT in the pipelined streaming mode and 396 MHz for the 256 point radix-2 burst mode FFT with the 12-bit fixed-point input range [10]. These frequencies correspond to 3.6 ns and 2.5 ns pipelining rates. However, there exist FFT/DSP processors with the pipelining frequency over 1 GHz, as for example, TMS320C6678 from Texas Instruments [11] that operates at 1.25 GHz. The pipelining rate in this case is equal to 0.8 ns. Such processors impose severe requirements on the magnitude computation circuit with respect to speed and fine-grained structure of the circuit. The proposed design permits to attain this goal. In addition to time requirements, the crucial problem is the maximum magnitude error. This error is the result of the applied approximation algorithm and properties of fixed-point arithmetic. A small approximation error can be achieved for numbers from the given number range by using the known iterative algorithms through the use of a large number of iterations. However, this can be achieved at the expense of an increased computation time which can be unacceptable. The main contribution of this work is an approach that allows to calculate magnitude without iterations and without exceeding an acceptable error. The solution is based on an improved version of the alpha max and beta min algorithm [3]. The magnitude error of this algorithm can be controlled by selecting a proper number of approximation regions. It may be noted that an increased number of regions gives the smaller maximum error. In the literature a derivation of the approximation with two regions is given [3]. In this work we show an improved version which is the extended version of the algorithm from [4]. This version allows to use any number of approximation regions. In order to determine the number of the needed approximation regions we present a relationship between the maximum error and the number of approximation regions. This relationship allows to find the number of regions for the prescribed maximum error. The crucial issue in the improved version of alpha max beta min algorithm is the choice of the proper approximation region. The proper region is selected using the quotient of arguments.

In Section 1 we review magnitude calculation algorithms, in Section 2 we present the generalized magnitude approximation algorithm and in Section 3 we analyze the algorithm error.

2. REVIEW OF MAGNITUDE CALCULATION ALGORITHMS

A magnitude calculation algorithm usually includes squaring of arguments, addition and square rooting. Squaring is the relatively simple operation and can be performed by multiplication or for small-length binary words by look-up and addition. The most demanding operation is square-rooting. Therefore we review only these magnitude calculation algorithms that perform square-rooting in the form suitable for the fast hardware implementation.

Square rooting algorithms which are commonly used for hardware implementations can be divided into two distinct categories: iterative and non-iterative algorithms. The iterative algorithms are often used but they usually require a considerable number of iterations to obtain the square root with an acceptable accuracy. The non-iterative algorithms are highly desirable but the maximum acceptable error bound is difficult to ensure. The iterative algorithms can be divided into three main groups: Newton-Raphson [5] method based algorithms, digit-by-digit algorithms [6] and CORDIC algorithms [7].

In the algorithms based on the Newton-Raphson method an estimate of the square root is obtained in each iteration. The main disadvantage of these algorithms is the dependence of the error upon the number of iterations and the starting point. These algorithms can be transformed into a non-iterative form but the general division and multiplication are needed that slow down execution of the algorithm. Moreover, the calculation error for a fixed number of stages strongly depends upon the choice of the starting point.

In digit-by-digit algorithms one-digit result is produced in every step of calculations by inspecting the shifted partial remainder which is derived from previous digit selections. Two forms are known: the restoring algorithm and the non-restoring algorithm which does not restore the remainder and requires less hardware resources. In practice the modified versions of non-restoring binary shift-subtract algorithms are used. Sutikno et al. [8] presented such algorithm and its FPGA implementation that uses subtraction only and appending of 01 or 11 sequence. The algorithm for n-bit argument requires $\lceil n/2 \rceil$ steps. Their FPGA implementation uses about 256 LUTs (Altera FPGA APEX 20KE).

The CORDIC based algorithms belong to the third class of iterative algorithms, which can be adopted for the efficient square rooting computation. The CORDIC hardware uses two adders and two shift registers at each computational stage. The number of stages depends on the type of arithmetic used and the dynamic range of the input signal. For small dynamic ranges, for example 11-12 bit, six to eight computational stages may be required to achieve the small error of the result. For greater dynamic ranges, for example 16-bit, the number of stages may increase to 11-12. After completing the basic computations the magnitude normalization is needed, being a multiplication by a suitable constant. This constant is a product of factors emerging at every CORDIC step due to the nature of the algorithm.

If the radicand is the sum of squares, the alpha max and beta min algorithm [4] can be applied. This algorithm in its original version uses up to two approximation regions and allows to compute the magnitude of a complex number without division and without iterations. In principle, until recently, there existed two versions of the algorithm. One version allows to compute magnitude with the error not exceeding 3.95% and the second with 1%. The initial version of the magnitude calculation algorithm based on the alpha max beta min was presented in [4]. However, this work contains derivation of the general form of magnitude calculation algorithm with appropriate corrections. Moreover, an extended analysis of the magnitude error resulting from the improper choice of the approximation region is presented.

3. GENERALIZED ESTIMATION ALGORITHM

The problem is to compute $R=\sqrt{P^2+Q^2}$, where P and Q represent the quadrature pair, i.e., the real and imaginary part of a complex number. Define

$$x=\max\left(|P|,|Q|\right), \tag{1}$$

$$y=\min\left(|P|,|Q|\right). \tag{2}$$

We may formulate the approximation problem in the following manner. We have to approximate $\sqrt{x^2+y^2}$ as the linear combination of x and y in the following form

$$\sqrt{x^2+y^2}=\alpha x+\beta y\,. \tag{3}$$

Using $\dfrac{y}{x}=\tan\theta$ and inserting it into (3) we obtain

$$\sqrt{1+\tan^2\theta}=\alpha+\beta\tan\theta\,. \tag{4}$$

After substituting and rearrangement of terms we receive

$$1=\alpha\cos\theta+\beta\sin\theta\,. \tag{5}$$

Eq. (5) is fulfilled for each θ and a suitable pair (α,β). Our aim is to approximate the right-hand side for $\theta\in[0,\pi/4]$ by using one or several, suitably chosen, pairs of (α,β). For this purpose we divide the interval into the certain number of regions $(\theta_{s_i},\theta_{e_i}), i=1,2,...,n$ and assign to each region a pair (α_i,β_i) that will give an approximation of (5) with an acceptable error. The approximation error for the i-th region is given by

$$e_i(\theta,\alpha,\beta)=(\alpha-\alpha_i)\cos\theta+(\beta-\beta_i)\sin\theta\,,\ i=1,2,...,n\,, \tag{6}$$

where (α,β) are exact values that should be used for the given pair (x,y) and (α_i,β_i) are the values assumed for the i-th region.

Inserting (5) into (6) we obtain

$$e_i(\theta)=1-\alpha_i\cos\theta-\beta_i\sin\theta\,,\ i=1,2,...,n\,. \tag{7}$$

As we see from (6) and (7) the magnitude approximation problem has been reduced to an approximation of a constant equal to one. The right-hand side may approximate the constant in various ways. It can be, for example, the mean-square approximation, the least-square, equiripple or minimax approximation. We will use the equiripple approximation, because it provides the guaranteed peak error. As we have two variables we shall use two equations by imposing the requirement of the same absolute error at both ends of the interval and at the midpoint. It is evident that the absolute error will depend on the size of the approximation interval.

For the assumed number of regions we fix the certain values of $r = y/x$ as boundary points, where we switch the respective set of α and β coefficients (Fig. 1). We assume that each approximation interval starts at θ_s and ends at θ_e. The overall magnitude estimation procedure includes the following steps as in Algorithm 1. Before the Algorithm 1 is performed, we assume the number of approximation regions that directly affects the maximum approximation error.

Algorithm 1 Computation of magnitude of complex number

```
Input: $P, Q$,
1. Determine of $x = \max(|P|, |Q|)$, $y = \min(|P|, |Q|)$ for $(P, Q)$ pair.
2. Calculate $r = y/x$.
3. Determine the $i$-th current approximation region, based on $r$, and determine the proper pair $(\alpha_i, \beta_i)$.
4. Compute the magnitude as $\alpha_i x + \beta_i y$.
```

Now we will focus on derivation of general formulas for calculating of α and β. In order to estimate the magnitude with a controlled error level we assume that the approximation error for both ends of the interval and at a certain angle $\theta_{\max}$ within the interval should be equal. Then the following equations have to be fulfilled

$$|e(\theta_s)| = |e(\theta_{\max})|, \tag{8a}$$

$$|e(\theta_e)| = |e(\theta_{\max})|, \tag{8b}$$

where $\theta_{\max} = 0.5(|\theta_s| + |\theta_e|)$ is chosen as the midpoint of the interval. Using (6) and (7) in (8a) and (8b) respectively we receive two equations

$$1 - \alpha \cdot \cos\theta_s - \beta \cdot \sin\theta_s = 1 - \alpha \cdot \cos\theta_{\max} - \beta \cdot \sin\theta_{\max}, \tag{9a}$$

$$1 - \alpha \cdot \cos\theta_e - \beta \cdot \sin\theta_e = -1 + \alpha \cdot \cos\theta_{\max} + \beta \cdot \sin\theta_{\max}. \tag{9b}$$

It is easily seen that after solving these equations we get the general formulas for α and β

$$\alpha = \frac{2(\sin\theta_e - \sin\theta_s)}{D_1 + D_2}, \tag{10a}$$

$$\beta = \frac{2(\sin\theta_e - \sin\theta_s)}{D_1 + D_2} \cdot \frac{\cos\theta_s - \cos\theta_e}{\sin\theta_e - \sin\theta_s}, \tag{10b}$$

where $D_1 = (\cos\theta_e - \cos\theta_{\max})(\sin\theta_e - \sin\theta_s)$ and $D_2 = (\cos\theta_s - \cos\theta_e)(\sin\theta_e - \sin\theta_{\max})$.

Using (10a) and (10b) we may calculate α and β coefficients for any number of approximation regions. In Table 1 the precalculated values of the coefficients for up to eight approximation regions are given, which covers the full range of practical applications of the algorithm.

Table 1 Precalculated values of the coefficients for up to eight approximation regions

Num. of regions	Angle range for θ = arctan(Q/P)	α, β coefficients	Maximum error in region
1	[0, π/4]	α= 0.9604, β= 0.3978	3.95%
2	[0, π/8)	α= 0.9903, β= 0.1970	0.97%
	[π/8, π/4)	α= 0.8395, β= 0.5610	
3	[0, π/12)	α= 0.9957, β= 0.1311	0.42%
	[π/12,π/6)	α= 0.9278, β= 0.3843	
	[π/6,π/4]	α= 0.7968, β= 0.6114	
4	[0,π/16)	α= 0.9976, β= 0.0983	0.24%
	[π/16,π/8)	α= 0.9592, β= 0.2910	
	[π/8,3π/16)	α= 0.8840, β= 0.4725	
	[3π/16,π/4]	α= 0.7749, β= 0.6359	
5	[0,π/20)	α= 0.9984, β= 0.0785	0.15%
	[π/20,π/10)	α= 0.9738, β= 0.2338	
	[π/10,3π/20)	α= 0.9253, β= 0.3832	
	[3π/20,π/5)	α= 0.8539, β= 0.5233	
	[π/5,π/4)	α= 0.7615, β= 0.6504	
6	[0,π/24)	α= 0.9989, β= 0.0655	0.10%
	[π/24,π/12)	α= 0.9818, β= 0.1953	
	[π/12,π/8)	α= 0.9479, β= 0.3218	
	[π/8,π/6)	α= 0.8978, β= 0.4428	
	[π/6,5π/24)	α= 0.8324, β= 0.5562	
	[5π/24,π/4]	α= 0.7526, β= 0.6601	
7	[0,π/28)	α= 0.9992, β= 0.0561	0.07%
	[π/28,π/14)	α= 0.9866, β= 0.1676	
	[π/14, 3π/28)	α= 0.9616, β= 0.2770	
	[3π/28, π/7)	α= 0.9246, β= 0.3829	
	[π/7,5π/28)	α= 0.8759, β= 0.4841	
	[5π/28,3π/14	α= 0.8160, β= 0.5791	
	[3π/14,π/4)	α= 0.7462, β= 0.6668	
8	[0,π/32)	α= 0.9994, β= 0.0491	0.06%
	[π/32,π/16)	α= 0.9898, β= 0.1468	
	[π/16,3π/32)	α= 0.9706, β= 0.2431	
	[3π/32,π/8)	α= 0.9421, β= 0.3371	
	[π/8,5π/32)	α= 0.9045, β= 0.4278	
	[5π/32,3π/16)	α= 0.8582, β= 0.5144	
	[3π/16,7π/32)	α= 0.8037, β= 0.5961	
	[7π/32,π/4]	α= 0.7414, β= 0.6720	

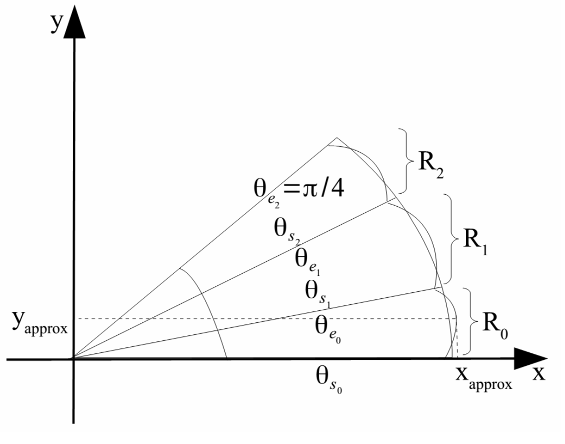

Fig 1. Approximation scheme using three regions

3. ALGORITHM ERROR ANALYSIS

The absolute approximation error $e_{\max}$ can be expressed as the absolute difference between the exact and approximated values at the start and endpoint of each region

$$e_{\max} = |1 - \alpha \cdot \cos\theta_e - \beta \cdot \sin\theta_e| = |1 - \alpha \cdot \cos\theta_s - \beta \cdot \sin\theta_s|. \tag{11}$$

The maximum peak error emerges at both ends of a region and in the midpoint. For the selected θ_s and θ_e it depends directly on the approximation coefficients. They can be obtained using (10a), (10b) and substituting coefficients from Table 1. The maximum peak error decreases when the number of regions is increased, which is depicted in Fig. 2. For three regions the error is just below 0.5 %, and further decrease of the maximum peak error by a factor of two is possible by adding the additional region. Ultimately, for eight regions the maximum peak error does not exceed 0.06 %, which is close to the floating-point result. It may be stated that rather not more than eight regions would be required in practical implementations of the algorithm.

In Fig. 3 a comparison of the magnitude computed using the floating-point and the algorithm that uses eight approximation regions is shown. Due to small values of the approximation error the curves are almost identical. The resulting approximation error is shown in Fig. 4. We may observe the increase of the approximation error in the vicinity of the boundaries of neighboring regions. This leads to the conclusion that boundaries of approximation regions should be determined with high accuracy. In fact, this may have an impact on additional inaccuracy in calculations because of the use of fixed-point arithmetic.

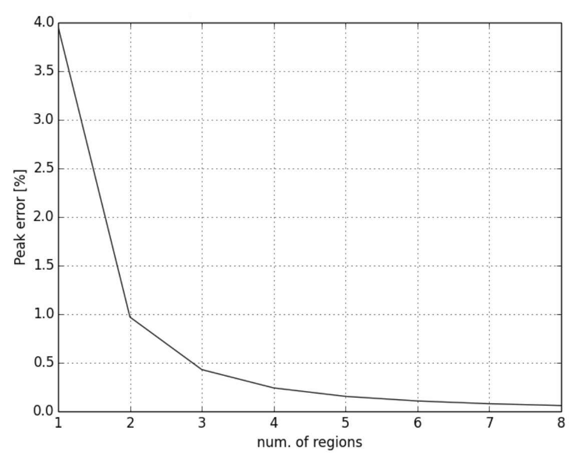

Fig. 2. Maximum peak error for selected numbers of approximation regions

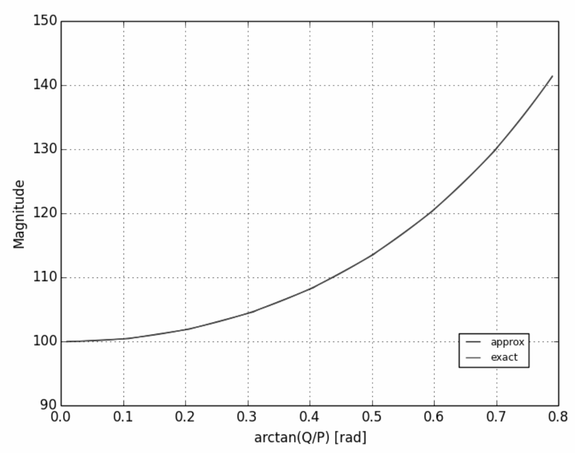

Fig. 3. Comparison of approximated (8 regions) and floating point magnitude

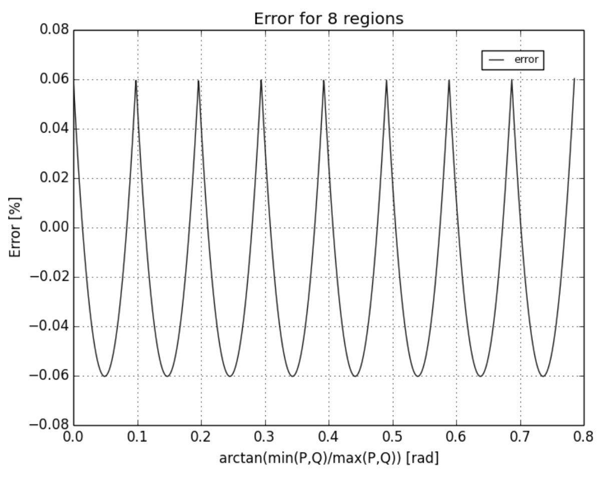

Fig. 4. The difference between approximated and exact magnitude for calculations with use of 8 regions

The total error resulting from the magnitude approximation implementation using Algorithm 1 has two components. The first one is the algorithm error and the second is the incorrect region choice error. The first error depends on the number of regions assumed for implementation. The second error results from the erroneous choice of the neighboring region instead of the correct one. The algorithm error can be controlled only by increasing number of approximation regions, while the region choice error depends on the accuracy of the estimate $\hat{r}$ of $r = y/x$ for the given (x, y) pair. It can be remarked that if general division would be applied it would lead to an ineffective implementation because of its computational complexity.

4. SUMMARY

The paper presents a detailed derivation of the improved version of alpha max plus beta min algorithm for noniterative calculation of the magnitude of complex numbers. The generalized version of the algorithm permits to control the maximum approximation error by using an adequate number of approximation regions with the proper pair of coefficients α and β assigned to each region. The improved version of the algorithm allows to control the maximum magnitude approximation error by using an adequate number of approximation regions. In this way it is possible to reduce the maximum error to 3.95% for one region, and 0.24% and 0.06% for four and eight regions, respectively. The selection of the region is made by using an approximate quotient of arguments. We have also given the detailed analysis of the relationship between the magnitude approximation error and the improper choice of the approximation region.

5. REFERENCES

1. Stimson, G.W. Introduction to Airborne Radar, 2 edition ed.; SciTech Publishing, 2014.
2. Hassanieh, H.; Shi, L.; Abari, O.; Hamed, E.; Katabi, D. GHz-wide sensing and decoding using the sparse Fourier transform. Proceedings of IEEE International Conference on Computer Communications, INFOCOM 2014, pp. 2256–2264, 2014.

3. Filip, A. Linear approximations to $\sqrt{x^2 + y^2}$ having equiripple error characteristics. IEEE Transactions on Audio and Electroacoustics 1973, 21, pp. 554–556.
4. Czyzak, M.; Smyk, R. FPGA realization of an improved alpha max plus beta min algorithm. Poznan University of Technology Academic Journals. Electrical Engineering 2014, 80, pp. 151-160.
5. Kosheleva, O. Babylonian method of computing the square root: Justifications based on fuzzy techniques and on computational complexity. 2009 Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2009, pp. 1–6.
6. Ercegovac, M. On Digit-by-Digit Methods for Computing Certain Functions. Conference Record of the 41th Asilomar Conference on Signals, Systems and Computers, ACSSC 2007, pp. 338–342.
7. Meher, P.; Valls, J.; Juang, T.B.; Sridharan, K.; Maharatna, K. 50 Years of CORDIC: Algorithms, Architectures, and Applications. IEEE Transactions on Circuits and Systems I: Regular Papers 2009, 56, pp. 1893–1907.
8. Sutikno, T. An Efficient Implementation of the Non Restoring Square Root Algorithm in Gate Level. International Journal of Computer Theory and Engineering 2011, 3, pp. 1793–8201.
9. Smyk R., Czyżak M., Implementation of improved magnitude estimation of complex numbers using alpha max and beta min algorithm, this issue.
10. Xilinx, Performance and Resource Utilization for Fast Fourier Transform v9.0, Vivado Design Suite Release 2016.3, 2016
11. Xiaohui Li, Blinka E., Very large FFT for TMS320C6678 processors, Texas Instruments, 2015.

ULEPSZONY ALGORYTM APROKSYMACJI MODUŁU LICZBY ZESPOLONEJ Z WYKORZYSTANIEM METODY ALPHA MAX BETA MIN

W artykule przedstawiono ulepszony algorytm aproksymacji modułu liczby zespolonej. Wyznaczanie modułu liczby zespolonej wymagane jest przykładowo przy realizacji FFT i filtracji cyfrowej sygnałów zespolonych. Jest to specjalny przypadek obliczania pierwisatka kwadratowego. Wersja ulepszona algorytmu umożliwia pełną kontrolę maksymalnego błędu wyznaczania modułu liczby zespolonej. Możliwe jest to dzięki wyprowadzeniu ogólnej postaci algorytmu dla dowolnej liczby regionów aproksymacji. Umożliwia to redukcję wspomnianego błędu aproksymacji z 3,95% dla jednego regionu, do przykładowo 0,24% dla czterech regionów i 0,06% dla ośmiu regionów aproksymacji. Proponowana metoda bazuje na zmodyfikowanej wersji algorytmu alpha max beta min. Algorytm ten wymaga najpierw porównania wartości bezwzględnych części rzeczywistej i części urojonej liczby zespolonej w celu wyznaczenia większej z nich. Następnie algorytm w wersji podstawowej z jednym regionem aproksymacji konieczne jest wykonanie tylko dwóch mnożeń przez stałą oraz jednego sumowania. W wersji ulepszonej wykonywane jest dodatkowe dzielenie celem wyznaczenia odpowiedniego regionu aproksymacji. Zastosowano tu beziteracyjny algorytm dzielenia. Szczegółowe zagadnienia związane z implementacją układową ulepszonej wersji algorytmu zostały przedstawione w artykule towarzyszącym.

Słowa kluczowe: algorytm alpha max beta min, aproksymacja modułu liczby zespolonej.