УДК 519.7
© Л. Мороз[1], 2012

ІТЕРАЦІЙНІ ФОРМУЛИ ДЛЯ CORDIC-МЕТОДУ

Розглянуто широковживаний ітераційний CORDIC- метод обчислення елементарних функцій. Досліджена проблема деформації модуля початкового вектора. Обчислено коефіцієнт деформації модуля вектора для формул різного порядку.

The widely usage iterative CORDIC- method for evaluation of elementary function had considered. The problem of deformation modulus of original vector researched. The rate of deformation modulus of the vector for formulas of different orders have been calculated.

1. ВСТУП

Ітераційний метод CORDIC – COordinate Rotation DIgital Computer, є одним з найпопулярніших методів обчислення багатьох елементарних функцій. Незважаючи на те, що методу уже понад 50 років, він продовжує інтенсивно розвиватись [6,8,10] .

2. МЕТА РОБОТИ

Метою даної роботи є запропонування узагальненого підходу до знаходження ітераційних формул різного порядку точності для CORDIC-методу та ефективного підвищення точності обчислень за рахунок використання коригуючих формул.

3.ОГЛЯД ДОСЛІДЖЕНЬ

Широковживаний ітераційний метод обчислення багатьох елементарних функцій під назвою CORDIC базується на очевидних тригонометричних перетвореннях повороту одиничного вектора в площині координат X-Y. Вихідний вектор $\left[X^{'}\ ,Y^{'}\right]^{T}$ одержується обертанням початкового вектора $\left[X\ ,Y\ \right]^{T}$ в декартовій системі координат на кут φ з допомогою виконання такої матричної операції (при обертанні вектора проти годинникової стрілки):

$$\begin{bmatrix} X^{'} \\ Y^{'} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

або

[1] Національний університет «Львівська політехніка»

$$\begin{cases} X' = X \cos\varphi - Y \sin\varphi = \cos\varphi(X - Y \tan\varphi) \\ Y' = Y \cos\varphi + X \sin\varphi = \cos\varphi(Y + X \tan\varphi) \end{cases} \quad (1)$$

або

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \cos\varphi \begin{bmatrix} 1 & -\tan\varphi \\ \tan\varphi & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}.$$

У методі CORDIC обертання вектора на кут φ – це ітераційний процес, який складається з мікроротацій, під час яких початковий вектор повертається на попередньо обчислені елементарні кути α_i. Будь-який кут φ може бути представлений елементарними кутами α_i з відповідною точністю у $m-1$ розрядів. Сума знакозмінних кутів α_i наближає даний кут φ :

$$\varphi = \sum_{i=0}^{m-1} \sigma_i \alpha_i + \varphi_m \ , \quad (2)$$

де $\sigma_i \in \{-1,1\}$-оператор вибору (decision operator) напрямку обертання, φ_m- залишок, який визначає абсолютну похибку наближення кута φ.

Якщо виразити кожен елементарний кут α_i через арктангенс співвідношення двох простих змінних s_i та c_i, а саме $\alpha_i = \arctan(s_i / c_i)$, то рівняння системи (1) на кожному кроці $i+1$, де $i = 0,1,2,...,m-1$, можна звести до вигляду:

$$\begin{cases} X_{i+1} = p_i (X_i - \sigma_i Y_i s_i / c_i) \\ Y_{i+1} = p_i (Y_i + \sigma_i X_i s_i / c_i) \end{cases} \quad (3)$$

причому $\quad p_i = \cos\alpha_i = c_i / \sqrt{s^2{}_i + c^2{}_i}$.

Систему (3) можна переписати у вигляді

$$\begin{cases} X_{i+1} = (X_i c_i - \sigma_i Y_i s_i) / \sqrt{s^2{}_i + c^2{}_i} \\ Y_{i+1} = (Y_i c_i + \sigma_i X_i s_i) / \sqrt{s^2{}_i + c^2{}_i} \end{cases} \quad (4)$$

Позначимо через k_i

$$k_i = \sqrt{s_i^2 + c_i^2} \tag{5}$$

З врахуванням цього даний кут φ наближається таким чином :

$$\varphi \approx \sum_{i=0}^{m-1} \sigma_i \alpha_i \ , \tag{6}$$

$\sigma_i \in \{-1,1\}$-оператор вибору (decision operator) напрямку обертання,

$$\alpha_i = \arctan(s_i / c_i) \ . \tag{7}$$

Тоді система (4) набере вигляду

$$\begin{cases} X_{i+1} = \dfrac{1}{k_i}(X_i c_i - \sigma_i Y_i s_i) \\ Y_{i+1} = \dfrac{1}{k_i}(Y_i c_i + \sigma_i X_i s_i) \end{cases} \tag{8}$$

Якщо відкинути з (8) масштабуючий коефіцієнт $1/k_i$, то отримаємо спрощений ітераційний CORDIC. Ротації в такому випадку називають псевдоротаціями, а рівняння приймають вигляд

$$\begin{cases} x_{i+1} = (x_i c_i - \sigma_i y_i s_i) \\ y_{i+1} = (y_i c_i + \sigma_i x_i s_i) \end{cases} \tag{9}$$

Системи рівнянь (3),(4),(8) пов’язані з системою (9) наступним чином :

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \frac{1}{k_i} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix}, \tag{10}$$

або

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = k_i \begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} \tag{11}$$

Зауважимо що таке спрощення веде до деформації модуля початкового вектора $\left[X, Y \right]^T$, причому коефіцієнт деформації k_i на кожній ітерації описується рівнянням (5).

Якщо провести всі m ітерацій за алгоритмом (8), то в результаті отримаємо:

$$\begin{aligned} X_m &= X \cos(\sum_{i=0}^{m-1} \sigma_i \alpha_i) - Y \sin(\sum_{i=0}^{m-1} \sigma_i \alpha_i) \\ Y_m &= Y \cos(\sum_{i=0}^{m-1} \sigma_i \alpha_i) + X \sin(\sum_{i=0}^{m-1} \sigma_i \alpha_i) \end{aligned} \quad (12)$$

Якщо ж провести всі m ітерацій за алгоритмом (9), то отримаємо:

$$\begin{aligned} x_m &= K_m \left[X \cos(\sum_{i=0}^{m-1} \sigma_i \alpha_i) - Y \sin(\sum_{i=0}^{m-1} \sigma_i \alpha_i) \right] \\ y_m &= K_m \left[Y \cos(\sum_{i=0}^{m-1} \sigma_i \alpha_i) + X \sin(\sum_{i=0}^{m-1} \sigma_i \alpha_i) \right] \end{aligned} \quad (13)$$

Тут K_m -результуючий коефіцієнт деформації, який визначається з рівняння

$$K_m = \prod_{i=0}^{m-1} k_i = \prod_{i=0}^{m-1} (\sqrt{s_i^2 + c_i^2}) \quad (14)$$

При виборі виразів для c_i, s_i ставиться вимога, щоб операції їх множення на відповідні змінні (9) могла бути замінені на операцію зсуву та додавання-віднімання.

Найпростішими і найпоширенішими виразами є такі:

$$c_i = 1, \qquad s_i = \varepsilon, \quad (15)$$

де $\varepsilon = 2^{-i}$, а елементарний кут α_i буде дорівнювати $\arctan(\varepsilon)$. Назвемо формули (15) формулами першого порядку точності [7]. Тоді рівняння (3)-(7) описують традиційний ітераційний CORDIC в коловому режимі (circular mode або rotation mode)

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i y_i \varepsilon \\ y_{i+1} &= y_i + \sigma_i x_i \varepsilon \end{aligned} \quad (16)$$

$$z_{i+1} = z_i - \sigma_i \alpha 1_i \quad (17)$$

де $\sigma_i = \begin{cases} -1 \; if \; z_i < 0 \\ +1 \; if \; z_i \geq 0 \end{cases} \qquad i = 0,1,2,...,m-1$, (18)

$$\alpha 1_i = \arctan(\varepsilon), \tag{19}$$

причому

$$x_0 = X \;,\; y_0 = Y \;, z_0 = \varphi. \tag{20}$$

Тут присутні лише операції зсуву, додавання чи віднімання та зчитування з пам'яті наперед обчислених кутів α_i. Коефіцієнт деформації модуля вектора на кожному кроці $k1_i$ при цьому складає

$$k1_i = \sqrt{1+\varepsilon^2}\,. \tag{21}$$

Якщо повернути початковий вектор за таким алгоритмом на кут φ,то після m ітерацій отримаємо при початкових умовах $x_0 = 1,\, y_0 = 0$ наступний результат:

$$\begin{aligned} x_m &= K1_m \cos\varphi \\ y_m &= K1_m \sin\varphi \\ z_m &= \varphi_m \approx 0 \\ z_0 &= \varphi \end{aligned} \tag{22}$$

де $K1_m = \prod_{i=0}^{m-1} k1_i = \prod_{i=0}^{m-1} \sqrt{1+\varepsilon^2}$ - результуючий коефіцієнт деформації модуля вектора. Для його компенсації використовують початкове значення $x_0 = 1/K1_m$ (якщо використовується алгоритм зі сталим , наперед заданим, числом кроків – тобто $\sigma_i \in \{-1,1\}$). В цьому випадку отримаємо після m ітерацій, що

$$\begin{aligned} x_m &= X_m = \cos\varphi \\ y_m &= Y_m = \sin\varphi \end{aligned}\;, \tag{23}$$

тобто обчислення проводяться з граничною для даного методу точністю (без врахування похибок, зумовлених обмеженою розрядністю, заокругленнями).

Слід сказати, що процес відпрацювання вхідного кута φ, що описується рівняннями (16)-(20) – збіжний . Недоліком такого підходу є необхідність проведення всіх m ітерацій.

Можна отримати значно менший коефіцієнт деформації на кожній ітерації, якщо задати $c_i = 1 - \varepsilon^2 / 2$ і знайти вираз для s_i, виходячи з умови

$$s_i = \sqrt{1 - c^2{}_i} \tag{24}$$

$$s_i = \sqrt{1 - c^2{}_i} = \frac{1}{2}\sqrt{4\varepsilon^2 - \varepsilon^4} \tag{25}$$

Наближене значення для s_i має вигляд:

$$s_i \approx \varepsilon - \varepsilon^3 / 8 - \varepsilon^5 / 128 \tag{26}$$

Якщо взяти

$$s_i = \varepsilon, \ c_i = 1 - \varepsilon^2 / 2, \tag{27}$$

то елементарний кут можна зобразити як $\arctan\left[\varepsilon / (1 - \varepsilon^2 / 2)\right]$. Це формули другого порядку точності. Ітераційний процес матиме такий вигляд:

$$x_{i+1} = x_i (1 - \varepsilon^2 / 2) - \sigma_i y_i \varepsilon$$

$$y_{i+1} = y_i (1 - \varepsilon^2 / 2) - \sigma_i x_i \varepsilon \tag{28}$$

$$z_{i+1} = z_i - \sigma_i \alpha 2_i \tag{29}$$

де $$\sigma_i = \begin{cases} -1 \ if \ z_i < 0 \\ +1 \ if \ z_i \geq 0 \end{cases} \quad i = 0, 1, 2, ..., m - 1, \tag{30}$$

$$\alpha 2_i = \arctan\left[\varepsilon / (1 - \varepsilon^2 / 2)\right]. \tag{31}$$

Слід відмітити, що процес відпрацювання вхідного кута φ (рівняння (29)-(31)) із $\sigma_i \in \{-1, 1\}$ є незбіжним [5,6].

Коефіцієнт деформації модуля вектора на кожному кроці складає

$$k2_i = \sqrt{1 + \varepsilon^4 / 4}. \tag{32}$$

Якщо ж взяти

$$s_i \approx \varepsilon - \varepsilon^3 / 8, \ c_i = 1 - \varepsilon^2 / 2, \tag{33}$$

то елементарний кут можна зобразити як $\arctan\left((\varepsilon - \varepsilon^3 / 8) / (1 - \varepsilon^2 / 2)\right)$. Це формули третього порядку точності. Ітераційний процес матиме такий вигляд:

$$x_{i+1} = x_i(1-\varepsilon^2/2) - \sigma_i y_i(\varepsilon - \varepsilon^3/8)$$
$$y_{i+1} = y_i(1-\varepsilon^2/2) + \sigma_i y_i(\varepsilon - \varepsilon^3/8) \quad (34)$$

$$z_{i+1} = z_i - \sigma_i \alpha 3_i \quad (35)$$

де
$$\sigma_i = \begin{cases} -1 \; if \; z_i < 0 \\ +1 \; if \; z_i \geq 0 \end{cases} \quad i = 0,1,2,...,m-1, \quad (36)$$

$$\alpha 3_i = \arctan\left[(\varepsilon - \varepsilon^3/8)/(1-\varepsilon^2/2)\right]. \quad (37)$$

Коефіцієнт деформації модуля вектора на кожному кроці складає

$$k3_i = \sqrt{1+\varepsilon^6/64}. \quad (38)$$

Аналогічно можна вивести формули четвертого, п'ятого, шостого та вищих порядків точності. Вони зведені у таблицю 1. Зауважимо, що вони в основному збігаються з результатами досліджень, проведених в роботах [1-9].

Таблиця 1

№	Порядок формули	c_i	s_i	k_i
1	*I*	$c_i = 1$	$s_i = \varepsilon$	$k1_i = \sqrt{1+\varepsilon^2}$
2	*II*	$c_i = 1-\varepsilon^2/2$;	$s_i = \varepsilon$	$k2_i = \sqrt{1+\varepsilon^4/4}$
3	*III*	$c_i = 1-\varepsilon^2/2$;	$s_i = \varepsilon - \varepsilon^3/8$	$k3_i = \sqrt{1+\varepsilon^6/2^6}$
4	*IV*	$c_i = 1-\varepsilon^2/2$;	$s_i = \varepsilon - \varepsilon^3/8 - \varepsilon^5/128$	$k4_i = \sqrt{1+\varepsilon^8/2^9+\varepsilon^{10}/2^{14}}$
5	*V*	$c_i = 1-\varepsilon^2/2+\varepsilon^4/8$;	$s_i = \varepsilon - \varepsilon^3/4 + \varepsilon^5/32$	$k5_i = \sqrt{1+\varepsilon^{10}/2^{10}}$
6	*VI*	$c_i = 1-\varepsilon^2/2+\varepsilon^4/16-\varepsilon^6/512$;	$s_i = \varepsilon - 3\varepsilon^3/16 + \varepsilon^5/64$	$k6_i = \sqrt{1+\varepsilon^{12}/2^{18}}$
7	*VI–I*	$c_i = 1-\varepsilon^2/2-\varepsilon^3/2-\varepsilon^4/4+\varepsilon^6/8$;	$s_i = \varepsilon-\varepsilon^2/2-\varepsilon^4/4-\varepsilon^5/4$	$k6I_i = \sqrt{1+\varepsilon^{12}/2^6}$
8	*VII*	$c_i = 1-\varepsilon^2/2+\varepsilon^4/8-\varepsilon^6/32$;	$s_i = \varepsilon-\varepsilon^3/4-\varepsilon^5/16-\varepsilon^7/128$	$k7_i = \sqrt{1+\varepsilon^{14}/2^{14}}$

В таблиці 2 показані значення відхилення коефіцієнта деформації модуля вектора від одиниці для деяких формул різного порядку точності в залежності від значення $\varepsilon = 2^{-i}$:

$$\Delta_n = k(n) - 1,$$

де n - порядок формули.

З таблиці 2 випливає, що, наприклад, перехід від формул першого порядку точності до формул другого порядку вже навіть для $i = 4 \div 8$ зменшує абсолютну похибку обчислень на декілька десяткових порядків. Звичайно, що таке підвищення точності досягається за рахунок додаткових апаратурних затрат, однак дає можливість підняти ще й швидкодію за рахунок вибору оператора $\sigma_i \in \{0,1\}$ або $\sigma_i \in \{-1,0,1\}$, чого не дозволяли зробити формули першого порядку точності.

Таблиця 2

№	ε	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
1	$\varepsilon=2^{-1}$	0.1180	0.7782e-2	0.1221e-3	0.3844e-5	0.4768e-6
2	$\varepsilon=2^{-2}$	0.3077e-1	0.4882e-3	0.1907e-5	0.1493e-7	0.4657e-9
3	$\varepsilon=2^{-3}$	0.7782e-2	0.3052e-4	0.2980e-7	0.5824e-10	0.4547e-12
4	$\varepsilon=2^{-4}$	0.1951e-2	0.1907e-5	0.4657e-9	0.2274e-12	0.4441e-15
5	$\varepsilon=2^{-5}$	0.4882e-3	0.1192e-6	0.7276e-11	0.8882e-15	0.4337e-18
6	$\varepsilon=2^{-6}$	0.1221e-3	0.7451e-8	0.1137e-12	0.3469e-17	0.4235e-21
7	$\varepsilon=2^{-7}$	0.3052e-4	0.4657e-9	0.1776e-14	0.1355e-19	0.4136e-24
8	$\varepsilon=2^{-8}$	0.7629e-5	0.2910e-10	0.2776e-16	0.5294e-22	0.4039e-27
9	$\varepsilon=2^{-9}$	0.1907e-5	0.1819e-11	0.4337e-18	0.2068e-24	0.3944e-30
10	$\varepsilon=2^{-10}$	0.4768e-6	0.1137e-12	0.6776e-20	0.8078e-27	0.3852e-33
11	$\varepsilon=2^{-11}$	0.1192e-6	0.7105e-14	0.1059e-21	0.3155e-29	0.3762e-36
12	$\varepsilon=2^{-12}$	0.2980e-7	0.4441e-15	0.1654e-23	0.1233e-31	0.3673e-39
13	$\varepsilon=2^{-13}$	0.7451e-8	0.2776e-16	0.2585e-25	0.4815e-34	0.3587e-42
14	$\varepsilon=2^{-14}$	0.1863e-8	0.1735e-17	0.4039e-27	0.1881e-36	0.3503e-45
15	$\varepsilon=2^{-15}$	0.4657e-9	0.1084e-18	0.6311e-29	0.7347e-39	0.3421e-48
16	$\varepsilon=2^{-16}$	0.1164e-9	0.6776e-20	0.9861e-31	0.2870e-41	0.3341e-51
17	$\varepsilon=2^{-17}$	0.2910e-10	0.4235e-21	0.1541e-32	0.1121e-43	0.3263e-54
18	$\varepsilon=2^{-18}$	0.7276e-11	0.2647e-22	0.2407e-34	0.4379e-46	0.3186e-57
19	$\varepsilon=2^{-19}$	0.1819e-11	0.1654e-23	0.3762e-36	0.1711e-48	0.3111e-60
20	$\varepsilon=2^{-20}$	0.4547e-12	0.1034e-24	0.5877e-38	0.6682e-51	0.3039e-63
…						
24	$\varepsilon=2^{-24}$	0.1776e-14	0.1578e-29	0.3503e-45	0.1556e-60	0.2764e-75

4. МЕТОД ПІДВИЩЕННЯ ТОЧНОСТІ ОБЧИСЛЕНЬ

Спробуємо підвищити точність обчислень (зменшити значення коефіцієнта деформації модуля вектора) на кожній ітерації для будь-якого порядку формул. Покажемо це на прикладі формул першого порядку точності.

Для цього розглянемо рівняння (10) з врахуванням (15):

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \frac{1}{k_i} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \frac{1}{\sqrt{1+\varepsilon^2}} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix}. \qquad (39)$$

Значення $\frac{1}{\sqrt{1+\varepsilon^2}}$ можна замінити його наближеним значенням $1-\varepsilon^2/2$. Тоді ітерації (9) ,або (17)-(20) для даного випадку, будуть доповнені наступними коригуючими ітераціями:

$$\begin{aligned} x'_{i+1} &= x_{i+1} \ (1-\varepsilon^{2)}/2) \\ y'_{i+1} &= y_{i+1} \ (1-\varepsilon^{2)}/2) \end{aligned} . \qquad (40)$$

Очевидно, що після такого коригування отримані значення x'_{i+1}, y'_{i+1} будуть значно ближчими до точних значень X_{i+1}, Y_{i+1} і ці скориговані значення знову включаються в ітераційний процес (16)-(20) як змінні

$$\begin{aligned} x_{i+1} &= x'_{i+1} \\ y_{i+1} &= y'_{i+1} \end{aligned} .$$

Значення α_i при цьому залишаються без змін, оскільки s_i та c_i зміняться пропорційно на однакове значення $(1-\varepsilon^2/2)$. Однак коефіцієнт деформації зменшиться до значення $k1'_i = \sqrt{(1-\varepsilon^2/2)^2(s^2{}_i + c_i^2)}/c_i = \sqrt{1-3*\varepsilon^4/4+\varepsilon^6/4}$.

Аналогічно можна скоригувати значення змінних x_{i+1}, y_{i+1} для будь-якого порядку формул. В загальному випадку коригування величин x_{i+1}, y_{i+1} можна зобразити формулами:

$$\begin{aligned} x'_{i+1} &= x_{i+1} \ (1-b) \\ y'_{i+1} &= y_{i+1} \ (1-b) \end{aligned} , \qquad (41)$$

або в розгорнутому вигляді

$$x'_{i+1} = (c_i x_i - \sigma_i s_i y_i)(1-b\ )$$
$$y'_{i+1} = (c_i y_i + \sigma_i s_i x_i)(1-b\ ) \qquad (42)$$

Коефіцієнт деформації у такому випадку буде дорівнювати

$$k1'_i = \sqrt{(1-b)^2 (s^2{}_i + c_i^2)}\,, \qquad (43)$$

а значення b для різного порядку формул зведені в таблицю 3.

Таблиця 3

№	Порядок формули	b	$k'(n)_i$
1	*I*	$b = \varepsilon^2 / 2$	$k'1_i = \sqrt{1-3/4\ \varepsilon^4 + 1/4\ \varepsilon^6}$
2	*II*	$b\ = \varepsilon^4 / 2^3$	$k'2_i\ = \sqrt{1-3/2^6\ \varepsilon^8 + 1/2^8\ \varepsilon^{12}}$
3	*III*	$b\ = \varepsilon^6 / 2^7$	$k'3_i = \sqrt{1-3/2^{14}\ \varepsilon^{12} + 1/2^{20}\ \varepsilon^{18}}$
4	*IV*	$b\ = \varepsilon^8 / 2^{10}$	$k'4_i = \sqrt{1+1/2^{14}\ \varepsilon^{10} - 3/2^{20}\ \varepsilon^{16}}$
5	*V*	$b\ = \varepsilon^{10} / 2^{11}$	$k'5_i = \sqrt{1-3/2^{22}\ \varepsilon^{20} + 1/2^{32}\ \varepsilon^{30}}$
6	*VI*	$b\ = \varepsilon^{12} / 2^{19}$	$k'6_i = \sqrt{1+2^{-38}\ \varepsilon^{24}}$
7	*VI–I*	$b\ = \varepsilon^{12} / 2^7$	$k'6{-}I_i = \sqrt{1+2^{-14}\ \varepsilon^{24}}$
8	*VII*	$b\ = \varepsilon^{14} / 2^{15}$	$k'7_i = \sqrt{1-3/2^{30}\ \varepsilon^{28} + 2^{-44}\varepsilon^{42}}$

Зауважимо, що реалізація формул коригування (36) на кожній ітерації проста і включає лише операцію зсуву та віднімання.

Для прикладу наведемо значення коефіцієнта деформації для різних формул при $\varepsilon = 2^{-4}$ до (Δ_n) і після (Δ'_n) коригування:

Δ_1=0.1951e-2 Δ_2=0.1907e-5 Δ_3=0.4657e-9 Δ_4=0.2274e-12 Δ_5=0.4441e-15

Δ'_1=0.5715e-5 Δ'_2=0.5457e-11 Δ'_3=0.3253e-18 Δ'_4=0.2776e-16 Δ'_5=0.2958e-30.

Як бачимо відбулось суттєве (на порядки) зменшення значення коефіцієнта деформації модуля вектора.

5. ВИСНОВКИ

Використовуючи запропоновані узагальнені підходи до створення формул різного порядку точності та коригуючих формул для них, можна будувати високоточні CORDIC – обчислювачі для відтворення широкого класу елементарних функцій.

1.F.S. Lim, Y.S. Wong, M.Rahman.Circular interpolators for numerical control: A comparison of the modified DDA techniques and an LSI interpolator, Computers in Industry, Vol. 18,Issue1,pp.41-52,1992. 2.Hong Tao. A high precision Digital Differential Analyzer for circle generation.- In Fundamental Algorithms for Computer Graphics, Earnshaw R.A., (Ed.), Vol.17 of NATO ASI Series F. Springer, pp. 239-256,1985. 3.Bergren C.A. A simple algorithm for circular interpolation.- Control Engineering, Vol.18, No 9, pp. 57-59, 1971. 4. 7. Аристов В.В. Функциональные макрооперации. Основы итерационных алгоритмов. – Киев: "Наукова думка", 1992.-280с. 5.K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture",IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 11,pp. 1463–1474, Nov. 2005. 6.Francisco J. Jaime, Miguel A. Sánchez, Javier Hormigo, Julio Villalba, and Emilio L. Zapata Enhanced Scaling-Free CORDIC. IEEE transactions on circuits and systems–i: regular papers, vol. 57, no. 7, july 2010, pp. 1654–1662. 7.G. Hekstra. Evaluation of Fast Rotation Methods. Journal of VLSI Signal Processing 25, pp.113–124, 2000. 8. Pramod K. Meher and S.Y.Park. CORDIC Designs for Fixed Angle of Rotation. IEEE transactions on very large scale integration (VLSI) systems. Issue: 99, pp.1-12 .7 March,2012. 9. R.F.Eschenbach, B.M.Oliver: An efficient coordinate rotation algorithm, IEEE Trans. Comput. C 27 (12) (1978) 1178-1180. 10. Lakshmi B. and Dhar A. S. CORDIC Architectures: A Survey. VLSI Design. Volume 2010, Article ID 794891, 19 pages, January 8, 2010.