Hindawi Publishing Corporation
Modelling and Simulation in Engineering
Volume 2014, Article ID 916539, 6 pages
http://dx.doi.org/10.1155/2014/916539

Research Article

A Comparison of Standard One-Step DDA Circular Interpolators with a New Cheap Two-Step Algorithm

Leonid Moroz,[1] Jan L. Cieśliński,[2] Marta Stakhiv,[1] and Volodymyr Maksymovych[1]

[1] *Lviv Polytechnic National University, S. Bandery Street 12, Lviv 79013, Ukraine*
[2] *Uniwersytet w Białymstoku, Wydział Fizyki, ul. Lipowa 41, 15-424 Białystok, Poland*

Correspondence should be addressed to Jan L. Cieśliński; janek@alpha.uwb.edu.pl

Received 28 June 2013; Accepted 21 October 2013; Published 20 January 2014

Academic Editor: Jing-song Hong

We present and study existing digital differential analyzer (DDA) algorithms for circle generation, including an improved two-step DDA algorithm which can be implemented solely in terms of elementary shifts, addition, and subtraction.

1. Introduction

Digital interpolation algorithms are widely used in machine tools with numerical control, graphics displays and plotters, and manipulation robots. Circles and circular arcs frequently appear in computer graphics, computer-controlled printing, and automated control; see [1]. One of most popular methods for generation of circles and arcs is known as digital differential analyzer (DDA). Digital circular interpolators (or angular sweep generators) based on DDA approach are widely used [2–6]. There are many papers where characteristics research results of such interpolators are systematized [7, 8].

In this paper a new improved two-step algorithm for DDA circle generation is presented. In paper [9] the idea of applying a Nystrom two-step scheme to circle generation appeared for the first time. Our next paper [10] developed theoretical and geometric aspects of this method. In the present paper we focus on practical and experimental issues. Moreover, here we consider circles of arbitrary radius R, while in [9, 10] we assumed $R = 1$.

The accuracy of this method is higher than the accuracy of other known algorithms. Because of its simplicity (it uses only elementary shift, addition, and subtraction); this method can also be used in numerical control, planning mechanisms, and so forth.

2. DDA Algorithms

The general class of DDA algorithms for circles generation is based on obvious trigonometric transformations describing rotation of vector R in the coordinate plane x-y (Figure 1);

$$\begin{aligned} x_{n+1} &= R\sin(\alpha+\Delta\alpha) = R\sin\alpha\cos\Delta\alpha + R\cos\alpha\sin\Delta\alpha \\ &= x_n\cos\Delta\alpha + y_n\sin\Delta\alpha, \\ y_{n+1} &= R\cos(\alpha+\Delta\alpha) = R\cos\alpha\cos\Delta\alpha + R\sin\alpha\sin\Delta\alpha \\ &= y_n\cos\Delta\alpha - x_n\sin\Delta\alpha. \end{aligned} \tag{1}$$

Advantages of DDA algorithms include simplicity and high speed in generating circle point coordinate x_{n+1}, y_{n+1}.

Substituting $\Delta\alpha = \varepsilon = 2^{-m}$, where m is an integer (usually $m \geq 3$), we rewrite (1) as

$$x_{n+1} = x_n\cos\varepsilon + y_n\sin\varepsilon, \qquad y_{n+1} = y_n\cos\varepsilon - x_n\sin\varepsilon, \tag{2}$$

which, obviously, can be expressed by a rotation matrix A:

$$A = \begin{bmatrix} \cos\varepsilon & -\sin\varepsilon \\ \sin\varepsilon & \cos\varepsilon \end{bmatrix}. \tag{3}$$

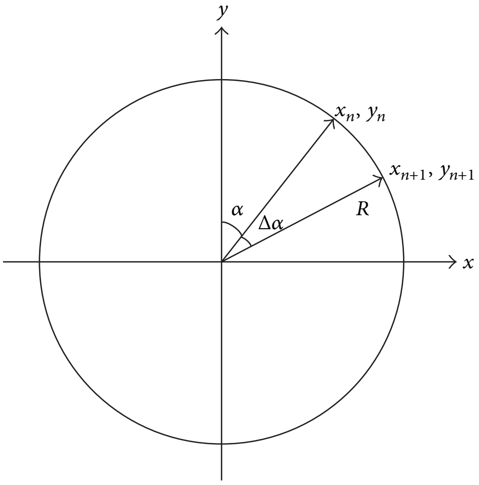

Figure 1: Rotation of vector R in the coordinate plane x-y.

To avoid expensive computation of trigonometric functions DDA algorithms use simpler (cheaper) expressions instead of $\cos\varepsilon$ and $\sin\varepsilon$. For example, by replacing trigonometric functions by truncated Taylor series one gets so-called simultaneous DDA algorithms [3]. The determinant of the rotation matrix is 1. Matrices of DDA algorithms have different determinants and $\det A \approx 1$ only approximately. The closer it equals 1 the more accurate is the corresponding circular interpolator [2]. Approximating $\cos\varepsilon \approx 1$ and $\sin\varepsilon \approx \varepsilon$, we obtain the simplest (and least accurate) DDA algorithm [2, 11]:

$$x_{n+1} = x_n + \varepsilon y_n, \qquad y_{n+1} = y_n - \varepsilon x_n, \qquad \det A = 1 + \varepsilon^2. \tag{4}$$

Much more accurate algorithm is obtained using $\cos\varepsilon \approx 1 - (\varepsilon^2/2)$ and $\sin\varepsilon \approx \varepsilon$ [7, 12, 13]:

$$x_{n+1} = \left(1 - \frac{\varepsilon^2}{2}\right) x_n + \varepsilon y_n, \qquad y_{n+1} = \left(1 - \frac{\varepsilon^2}{2}\right) y_n - \varepsilon x_n,$$
$$\det A = 1 + \frac{\varepsilon^4}{4}. \tag{5}$$

Approximating $\cos\varepsilon$ and $\sin\varepsilon$ with first two terms of Taylor series, we obtain the following algorithm [14]:

$$\begin{aligned} x_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) x_n + \left(\varepsilon - \frac{\varepsilon^3}{6}\right) y_n, \\ y_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) y_n - \left(\varepsilon - \frac{\varepsilon^3}{6}\right) x_n, \\ \det A &= 1 - \frac{\varepsilon^4}{4} + \frac{\varepsilon^6}{36}. \end{aligned} \tag{6}$$

Another second order scheme, different form truncated Taylor series, is proposed in [2]:

$$\begin{aligned} x_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) x_n + \left(\varepsilon - \frac{\varepsilon^3}{4}\right) y_n, \\ y_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) y_n - \left(\varepsilon - \frac{\varepsilon^3}{4}\right) x_n, \\ \det A &= 1 - \frac{\varepsilon^4}{4} + \frac{\varepsilon^6}{16}. \end{aligned} \tag{7}$$

All the above DDA algorithms have systematic radial error at each step, which is defined as

$$\Delta R_n = \sqrt{x_n^2 + y_n^2} - R. \tag{8}$$

In [7], given the recurrence relations for algorithms that do not have radial errors

$$\begin{aligned} x_{n+1} &= \frac{4 - \varepsilon^2}{4 + \varepsilon^2} x_n + \frac{4\varepsilon}{4 + \varepsilon^2} y_n, \\ y_{n+1} &= \frac{4 - \varepsilon^2}{4 + \varepsilon^2} y_n - \frac{4\varepsilon}{4 + \varepsilon^2} x_n. \end{aligned} \tag{9}$$

However, implementation of these algorithms requires floating point operations [7]. Replacing the coefficients of (9) with the first two terms of Taylor series [2], we obtain

$$\frac{4 - \varepsilon^2}{4 + \varepsilon^2} \approx 1 - \frac{\varepsilon^2}{2}, \qquad \frac{4\varepsilon}{4 + \varepsilon^2} \approx \varepsilon - \frac{\varepsilon^3}{4}. \tag{10}$$

As a result we obtain, obviously, algorithm given by (7).

A lower level of radial error with much simpler hardware implementation can be achieved using the following algorithm [3, 13]:

$$\begin{aligned} x_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) x_n + \left(\varepsilon - \frac{\varepsilon^3}{8}\right) y_n, \\ y_{n+1} &= \left(1 - \frac{\varepsilon^2}{2}\right) y_n - \left(\varepsilon - \frac{\varepsilon^3}{8}\right) x_n, \\ \det A &= 1 + \frac{\varepsilon^6}{64}. \end{aligned} \tag{11}$$

Another scheme worthwhile to be mentioned is the so-called “magic circle” interpolator [5, 7, 15]:

$$\begin{aligned} x_{n+1} = x_n + \varepsilon y_n, \qquad y_{n+1} &= \left(1 - \varepsilon^2\right) y_n - \varepsilon x_n, \\ \det A &= 1. \end{aligned} \tag{12}$$

This interpolator is more accurate than the simplest scheme (4) and requires the use of only two multipliers. However, in practice it is not used for large ε, because then the generated “circle” becomes an ellipse [2, 15].

Table 1: Maximum values of the absolute radial error ($R = 2^m$).

Number of algorithms	$m = 3$	$m = 5$	$m = 6$	$m = 8$	$m = 10$	$m = 12$
(4)	3.972	3.332	3.234	3.159	3.145	3.142
(5)	0.012	$0.771e-3$	$0.192e-3$	$0.120e-4$	$0.749e-6$	$0.468e-7$
(6)	$-0.41e-2$	$-0.257e-3$	$-0.641e-4$	$-0.400e-5$	$-0.25e-6$	$-0.156e-7$
(7)	−0.012	$-0.771e-3$	$-0.192e-3$	$-0.120e-4$	$-0.74e-6$	$-0.468e-7$
(11)	$0.12e-4$	$0.47e-7$	$0.29e-8$	$0.11e-10$	$0.45e-13$	$0.17e-15$
(12)	0.262	0.253	0.251	0.250	0.250	0.250
(13)	0.262	0.252	0.251	0.250	0.250	−0.250
(14)	−0.063	−0.015	$-0.781e-2$	$-0.195e-2$	$-0.48e-3$	$-0.122e-3$
(15)	−2.677	−3.020	−3.079	−3.124	−3.136	−3.140
(16)	$0.18e-7$	$0.31e-6$	$0.12e-5$	$0.19e-4$	$0.30e-3$	$0.49e-2$

There is also a group of sequential DDA algorithms [12]. Here are some of them:

$$x_{n+1} = x_n + \varepsilon y_n, \qquad y_{n+1} = y_n - \varepsilon x_{n+1}, \tag{13}$$

$$\begin{aligned} x_{n+1} &= x_n + \varepsilon y_n, & y_{n+1} &= y_n - \varepsilon x_{n+1}, \\ y_{n+2} &= y_{n+1} - \varepsilon x_{n+1}, & x_{n+2} &= x_{n+1} + \varepsilon y_{n+2}, \end{aligned} \tag{14}$$

$$x_{n+1} = \left(1 - \frac{\varepsilon^2}{2}\right) x_n + \varepsilon y_n, \qquad y_{n+1} = \left(1 - \frac{\varepsilon^2}{2}\right) y_n - \varepsilon x_{n+1}. \tag{15}$$

Two-step algorithm:

$$x_{n+2} = x_{n+1} \cos\varepsilon - x_n, \qquad y_{n+2} = y_{n+1} \cos\varepsilon - y_n \tag{16}$$

with initial conditions: $x_0 = R$, $y_0 = 0$, $x_1 = R\cos\varepsilon$, and $y_1 = R\sin\varepsilon$, is known as a "direct" or biquad form of the circular interpolator [14]. Note that its implementation needs only two multipliers.

In the process of circle points generation radial errors accumulate, and the law of accumulation is specific to a given algorithm. Note that in most cases, the circle radius is chosen to satisfy condition $2^{m-1} \leq R \leq 2^m$ [2, 12, 15]. Obviously, the greatest error will occur when $R = 2^m$. In Table 1 we give maximum values of the absolute radial systematic errors for $R = 2^m$ and $n = 2\pi(1/\varepsilon) = [2\pi 2^m]$, that is, for the full circle generation. Initial conditions are chosen to be $x_0 = 0$, $y_0 = R$.

Apart from radial error an important role in such devices (especially in computer numerical control (CNC) interpolators) is played by the chord error E_{ch} [7], which should not exceed one basic length unit (BLU). Unlike radial error, the chord error does not accumulate. In [7] it is shown that for faster circle point generation for most accurate methods (5)–(7), (9) $\Delta\alpha$ can be chosen from condition $\Delta\alpha = \sqrt{8/R}$ at $E_{\text{ch}} = 1$ (we recall that $\Delta\alpha \equiv \varepsilon$); that is

$$R \leq 2^{2m+3} E_{\text{ch}} \quad \text{or} \quad R \leq 2^{2m+3}. \tag{17}$$

Since the CNC systems radius can reach value $R_{\max} = 10^8$ BLU, the CNC interpolators angles range $\Delta\alpha = \varepsilon = 2^{-m}$, where $m = \overline{3, 12}$ overlapping all possible radius values range (the though adopted algorithms can work and at $m > 12$).

In Table 2 we give maximum values of absolute radial systematic error for most accurate algorithms in case $R = 2^{2m+3}$.

From these results we see that algorithm (11) is the most accurate. Therefore, in next sections we choose algorithm (11) in order to compare it with two new circular interpolators introduced in [10].

3. New DDA Algorithms

In the proposed two-step DDA algorithm the values x_{n+2} and y_{n+2} are determined by the following formula:

$$x_{n+2} = x_n - 2 \cdot \sin\varepsilon \cdot y_{n+1}, \qquad y_{n+2} = y_n + 2 \cdot \sin\varepsilon \cdot x_{n+1}. \tag{18}$$

This is an exact algorithm which can be derived as follows (for another approach, see [10]). We consider the rotation from the point x_n, y_n to x_{n+2}, y_{n+2}; see Figure 2. Hence, we obtain the following equations:

$$x_{n+1} = x_n \cos\varepsilon - y_n \sin\varepsilon, \qquad y_{n+1} = y_n \cos\varepsilon + x_n \sin\varepsilon, \tag{19}$$

$$\begin{aligned} x_{n+2} &= x_{n+1} \cos\varepsilon - y_{n+1} \sin\varepsilon, \\ y_{n+2} &= y_{n+1} \cos\varepsilon + x_{n+1} \sin\varepsilon. \end{aligned} \tag{20}$$

From (20) we obtain values x_{n+1} and $-y_{n+1}$:

$$\begin{aligned} x_{n+1} &= \frac{y_{n+2} - y_{n+1}\cos\varepsilon}{\sin\varepsilon}, \\ -y_{n+1} &= \frac{x_{n+2} - x_{n+1}\cos\varepsilon}{\sin\varepsilon}. \end{aligned} \tag{21}$$

Substituting values given by (21) into (20) we get

$$\begin{aligned} x_{n+2} - 2x_{n+1}\cos\varepsilon + x_n &= 0, \\ y_{n+2} - 2y_{n+1}\cos\varepsilon + y_n &= 0. \end{aligned} \tag{22}$$

Here we recognize the exact discretization of the classical harmonic oscillator equation [16, 17]. From (22) it follows that

$$x_{n+1} = \frac{x_{n+2} + x_n}{2\cos\varepsilon}, \qquad y_{n+1} = \frac{y_{n+2} + y_n}{2\cos\varepsilon}. \tag{23}$$

Table 2: Maximum values of the absolute radial error ($R = 2^{2m+3}$).

Number of algorithms	$m = 3$	$m = 5$	$m = 6$	$m = 8$	$m = 10$	$m = 12$
(5)	0.7975	0.1973	$0.984e - 1$	$0.246e - 1$	$0.614e - 2$	$0.153e - 2$
(6)	−0.2642	$-0.657e - 1$	$-0.328e - 1$	$-0.818e - 2$	$-0.205e - 2$	$-0.511e - 3$
(7)	−0.7932	−0.1972	$-0.984e - 1$	$-0.246e - 1$	$0.614e - 2$	$0.153e - 2$
(11)	$0.78e - 3$	$0.12e - 4$	$0.15e - 5$	$0.23e - 7$	$0.37e - 9$	$0.57e - 11$

Substituting obtained values in (21), we get

$$x_{n+1} = \frac{y_{n+2} - \left(\left(y_{n+2} + y_n\right)/2\cos\varepsilon\right)\cos\varepsilon}{\sin\varepsilon},$$
$$-y_{n+1} = \frac{x_{n+2} - \left(\left(x_{n+2} + x_n\right)/2\cos\varepsilon\right)\cos\varepsilon}{\sin\varepsilon} \qquad (24)$$

which can easily be reduced to

$$x_{n+1} = \frac{y_{n+2} - y_n}{2\sin\varepsilon}, \qquad -y_{n+1} = \frac{x_{n+2} - x_n}{2\sin\varepsilon}, \qquad (25)$$

that is, to (18). The algorithm works with initial conditions: $x_0 = R$, $y_0 = 0$, $x_1 = R\cos\varepsilon$, and $y_1 = R\sin\varepsilon$, where $R = 2^m$ (m is an integer). Approximating $\sin\varepsilon \approx \varepsilon$ one can transform (18) into another two-step algorithm:

$$x_{n+2} = x_n - 2\cdot\varepsilon\cdot y_{n+1}, \qquad y_{n+2} = y_n + 2\cdot\varepsilon\cdot x_{n+1}, \quad (26)$$

with $x_0 = R$, $y_0 = 0$, $x_1 = R\sqrt{1-\varepsilon^2}$, $y_1 = R$. This algorithm was first proposed in [9]; see also [10].

Functional diagram of interpolator which reproduces (26) is shown in Figure 3. Interpolator contains 6 registers (Register 1, Register 2, Register 3, Register 4, Output—$x_{n+2,}$, and Output—y_{n+2}), 2 multipliers (Right Shift Register 1 Right Shift Register 2), subtracter and adder.

In this scheme, at the beginning (before the interpolator starts) values of x_0 and y_0 are recorded in Register 2 and Register 4, respectively, while values of x_1 and y_1 are recorded in Register 1 and Register 3. Then, values of x_{n+1} and x_n are stored in Register 1 and Register 2, and values of y_{n+1} and y_n. are stored in Register 3 and Register 4. Right Shift Register 1 and Right Shift Register 2 perform multiplication of values y_{n+1} and x_{n+1} by 2ε. Thus, at outputs of subtractor and adder parallel binary codes are formed in accordance with (26). Rounded values are recorded in Output x_{n+2} and Output y_{n+2}.

Table 3 contains maximum values of absolute radial systematic error of proposed algorithm, given by (26), after full circle simulation in the case

$$d = 24, \qquad d = 32,$$
$$\varepsilon = 2^{-m}, \quad R = 2^m, \quad m = 3, \ldots, 10, \qquad (27)$$
$$n = 2\pi\frac{1}{\varepsilon},$$

where d is structure elements datapath (in bits) and n is number of iterations. Table 4 contains the same data for case $R = 2^{2m+3}$. Note that results given in Tables 1 and 2 do not include round-off errors; hence they do not depend on d.

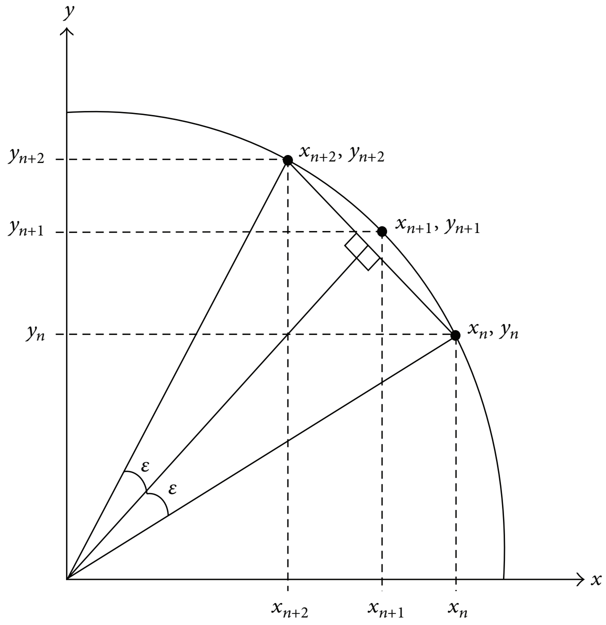

Figure 2: Generation of 3 subsequent points along the circle.

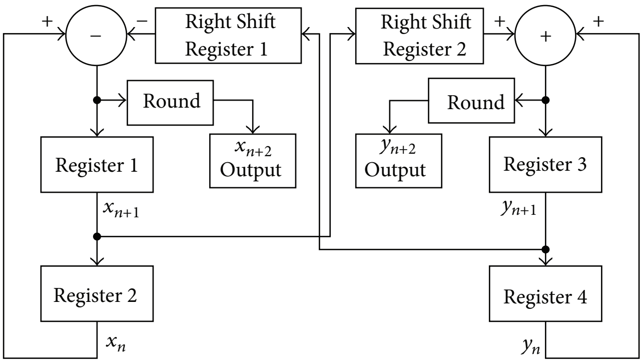

Figure 3: Functional diagram for the circular interpolator (26).

Analogical data for algorithm given by (11) are given in Table 5 (we consider, as before, case (27)). Table 6 contains the same data for $R = 2^{2m+3}$.

Comparing errors in Tables 3 and 5, and in Tables 4 and 6, we see that the absolute radial errors of our algorithm (26) (Tables 3 and 4) are smaller than errors of well-known algorithm (11) (see Tables 5 and 6). For example, we can compare values of errors at $R = 2^m$ ($m = 10$) and $d = 24$ given in Tables 3 and 5. Consequently, we will get that an absolute

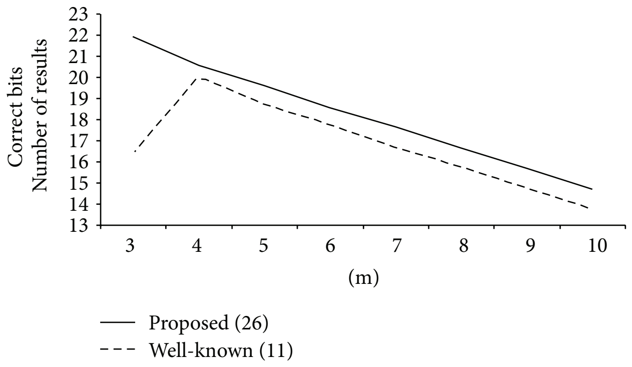

Figure 4: Number of correct bits of result as a function of m at $R = 2^m$ and $d = 24$.

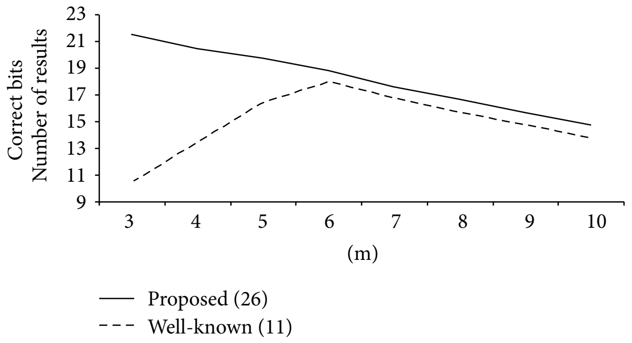

Figure 5: Number of correct bits of result as a function of m at $R = 2^{2m+3}$ and $d = 24$.

Table 3: Maximum values of the absolute radial error ($R = 2^m$) for proposed Algorithm (26).

m	Absolute radial error			
	d = 24		d = 32	
	Max	Min	Max	Min
3	7.712e − 8	−2.567e − 7	4.377e − 10	−9.137e − 10
4	1.215e − 7	−6.462e − 7	1.284e − 9	−2.666e − 9
5	2.458e − 7	−1.267e − 6	9.422e − 10	−5.362e − 9
6	4.643e − 7	−2.645e − 6	2.223e − 9	−8.292e − 9
7	7.045e − 7	−4.942e − 6	3.614e − 9	−1.856e − 8
8	1.655e − 6	−9.678e − 6	6.936e − 9	−3.794e − 8
9	3.258e − 6	−1.898e − 5	1.245e − 8	−7.470e − 8
10	6.758e − 6	−3.761e − 5	2.510e − 8	−1.456e − 7

radial error is smaller by (max(Table 5))/(max(Table 3)) = $(1.267e-5)/(6.792e-6) = 1.876$ times.

We also see that the smaller is m, the greater is the difference between the corresponding errors of both algorithms. For example, for ($m = 3$) and $d = 32$ we obtain that an absolute radial error is smaller by (max(Table 6))/(max(Table 4)) = $(7.781e-4)/(4.533e-10) \approx 1716523$ times.

Finally, Figures 4, 5, 6, and 7 present numbers of correct bits as a function of m, for $R = 2^m$ and $R = 2^{2m+3}$ and for $d = 24$ and $d = 32$, respectively.

Table 4: Maximum values of the absolute radial error ($R = 2^{2m+3}$) for proposed Algorithm (26).

m	Absolute radial error			
	d = 24		d = 32	
	Max	Min	Max	Min
3	6.965e − 8	−3.866e − 7	4.533e − 10	−1.671e − 9
4	1.768e − 7	−8.276e − 7	4.654e − 10	−3.550e − 9
5	3.010e − 7	−1.176e − 6	1.260e − 9	−5.494e − 9
6	4.245e − 7	−2.292e − 6	1.923e − 9	−9.138e − 9
7	8.948e − 7	−5.040e − 6	3.707e − 9	−1.927e − 8
8	1.454e − 6	−9.834e − 6	8.918e − 9	−3.704e − 8
9	3.424e − 6	−1.936e − 5	1.307e − 8	−7.389e − 8
10	6.792e − 6	−3.648e − 5	2.792e − 8	−1.437e − 7

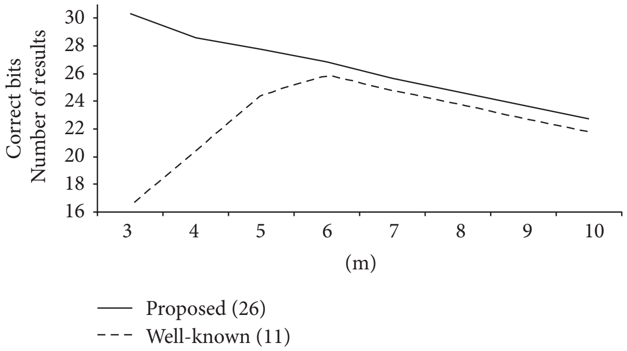

Figure 6: Number of correct bits of result as a function of m at $R = 2^m$ and $d = 32$.

Table 5: Maximum values of the absolute radial error ($R = 2^m$) for Algorithm (11).

m	Absolute radial error			
	d = 24		d = 32	
	Max	Min	Max	Min
3	1.214e − 5	0	1.215e − 5	0
4	9.641e − 7	−8.277e − 7	7.525e − 7	0
5	5.496e − 7	−2.386e − 6	4.552e − 8	0
6	5.508e − 7	−4.660e − 6	5.781e − 9	−1.682e − 8
7	1.235e − 6	−9.674e − 6	6.069e − 9	−3.579e − 8
8	4.009e − 6	−1.858e − 5	1.524e − 8	−7.158e − 8
9	6.478e − 6	−3.651e − 5	2.034e − 8	−1.484e − 7
10	1.268e − 5	−7.285e − 5	5.564e − 8	−2.853e − 7

Presented figures clearly show that the proposed algorithm (26) yields greater number of correct bits in comparison to algorithm (11).

4. Conclusion

A new circle generation DDA algorithm has been proposed. In terms of hardware implementation and precision of circle generation the proposed DDA algorithm for circle interpolator is more efficient as compared with existing algorithms. We hope that the proposed algorithm will find some applications

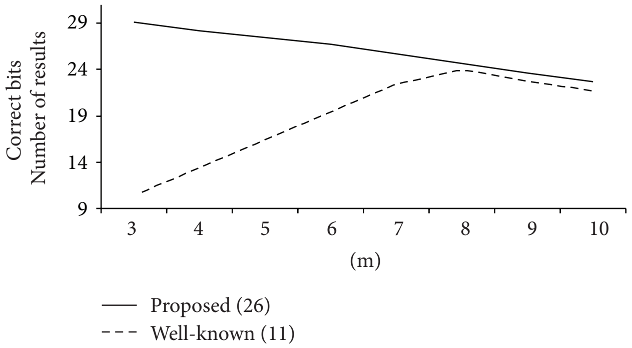

Figure 7: Number of correct bits of result as a function of m at $R = 2^{2m+3}$ and $d = 32$.

Table 6: Maximum values of the absolute radial error ($R = 2^{2m+3}$) for Algorithm (11).

m	Absolute radial error $d = 24$		$d = 32$	
	Max	Min	Max	Min
3	$7.780e-4$	0	$7.781e-4$	0
4	$9.650e-5$	0	$9.632e-5$	0
5	$1.165e-5$	0	$1.204e-5$	0
6	$2.469e-6$	$-3.933e-6$	$1.501e-6$	0
7	$1.731e-6$	$-9.497e-6$	$1.878e-7$	0
8	$2.359e-6$	$-1.942e-5$	$3.598e-8$	$-6.302e-8$
9	$5.371e-6$	$-3.779e-5$	$2.655e-8$	$-1.421e-7$
10	$1.057e-5$	$-7.478e-5$	$5.028e-8$	$-2.914e-7$

in computer graphics (display screens), computer-controlled printing, automated control, and so forth.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The second author (Jan L. Cieśliński) is partly supported by the National Science Centre (NCN) Grant no. 2011/01/B/ST1/05137.

References

[1] W. Wang and C. Wang, "Difference method for generation of circular arcs and ellipses," *Computer-Aided Design*, vol. 21, no. 1, pp. 33–37, 1989.

[2] N. Matsushiro, "A new digital differential analyzer for circl generation," *IEICE Transactions on Information and Systems*, vol. E81-D, no. 2, pp. 239–242, 1998.

[3] I. Chami, "A high precision digital differential analyzer for circular interpolation," *Tishreen University Journal*, vol. 29, no. 1, 2007.

[4] H. Hama and T. Okumoto, "Fast transform into polar coordinate by generalized DDA," *Journal of the Institute of Television Engineers of Japan*, vol. 47, no. 1, pp. 97–100, 1993.

[5] X. Y. Fan, Y. H. Guo, and S. C. Li, "New digital differential analyzer interpolation algorithm," *Advanced Science Letters*, vol. 6, no. 1, pp. 692–695, 2012.

[6] X. L. Yan, W. L. Wang, and N. Xue, "Research and simulation of an efficient circular interpolation algorithm," *Advanced Materials Research*, vol. 542-543, pp. 1204–1208, 2012.

[7] O. Masory and Y. Koren, "Reference-word circular interpolators for CNC systems," *Journal of Engineering for Industry*, vol. 104, no. 4, pp. 400–405, 1982.

[8] Y. Katayama, H. Harasaki, and M. Yano, "Parallel algorithm of˜ three dimensional digital differential analyzer," in *Proceedings of the Annual Convention ITE Japan*, pp. 185–186, 1993.

[9] L. V. Moroz, "A method for studying characteristics of digital sine-cosine generators," *Information Extraction and Processing*, vol. 36, no. 112, pp. 84–90, 2012 (Ukrainian).

[10] J. L. Cieśliński and L. V. Moroz, "Fast exact digital differential analyzer for circle generation," http://arxiv.org/abs/1304.4974.

[11] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, McGraw-Hill, 1979.

[12] F. S. Lim, Y. S. Wong, and M. Rahman, "Circular interpolators for numerical control: a comparison of the modified DDA techniques and an LSI interpolator," *Computers in Industry*, vol. 18, no. 1, pp. 41–52, 1992.

[13] R. F. Eschenbach and B. M. Oliver, "An efficient coordinate rotation algorithm," *IEEE Transactions on Computers*, vol. 27, no. 12, pp. 1178–1180, 1978.

[14] H. Hasegawa and Y. Okada, "Apparatus and method for controlling a moving vehicle utilizing a digital differential analysis circuit," US Patent 5379353, 1995.

[15] C. Pokorny Computer Graphics, *An Object Oriented Approach to the Art and Science*, Franklin, Beedle & Associates Publishing, 1994.

[16] J. L. Cieśliński and B. Ratkiewicz, "On simulations of the classical harmonic oscillator equation by difference equations," *Advances in Difference Equations*, vol. 2006, Article ID 40171, 2006.

[17] J. L. Cieśliński, "On the exact discretization of the classical harmonic oscillator equation," *Journal of Difference Equations and Applications*, vol. 17, no. 11, pp. 1673–1694, 2011.