IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **31** (2020) 045005 (9pp) https://doi.org/10.1088/1361-6501/ab4d1d

A simple unified branch-and-bound algorithm for minimum zone circularity and sphericity errors

Yu Zheng

Tencent Robotics X, Shenzhen, Guangdong Province, People's Republic of China

E-mail: petezheng@tencent.com

Received 17 June 2019, revised 27 September 2019
Accepted for publication 11 October 2019
Published 8 January 2020

Abstract

This paper presents a simple branch-and-bound (B&B) algorithm to compute the minimum zone circularity/sphericity error, which is formulated as the problem of finding the centers of two concentric circles/spheres that contain given points between them and have minimal difference in their radii. For any square domain of the center, a lower bound on the minimal radius difference attainable over the domain is proposed, and this monotonically increases as the domain is divided into square subdomains. Along with the branching operation, a domain will become infinitely small and provide only a center of concentric circles/spheres such that its lower bound is equal to the radius difference. Using such a lower bound, therefore, the B&B algorithm guarantees the computing of the exact circularity/sphericity error and it is proven that the accuracy of the computed result can be easily specified by the termination tolerance of the algorithm as needed. In addition, the algorithm is very efficient and easy to implement.

Keywords: roundness, circularity, sphericity, form error, branch-and-bound

(Some figures may appear in colour only in the online journal)

1. Introduction

Circular parts are components of many industrial products and widely exist in manufacturing. However, no part is perfectly circular in practice and errors can be introduced by various sources in machining processes, such as clamping distortion, spindle run-out, poor bearings in the lathe or grinding wheel spindle, misalignment and deflection of shafts, presence of dirt and chips on clamping surfaces, imbalance, heat, vibration, etc. In assembly of circular parts, it is not enough to consider only the dimensional tolerances on their diameters; the geometric form of circularity needs to be considered. Circularity, which is also known as roundness, is one of the most important geometric forms for circular parts and is used to measure how close the shape of a part is to a perfect circle. To evaluate the circularity error of a part, a set of data points first needs to be collected from the part's surface. To do this, the part is placed on a turntable and the probe of a measuring instrument is placed at the circular element of the part to be measured. While the part is rotated, the profile of the circular element or a set of measured data points is captured. Then, the circularity error can be estimated with respect to a chosen criterion. There are four different criteria for circularity error evaluation: least square circle (LSC), minimum circumscribed circle (MCC), maximum inscribed circle (MIC), and minimum zone circle (MZC). Of these criteria, the LSC is the simplest and can be calculated analytically, but the circularity error is often exaggerated. The MZC is aimed at finding a pair of concentric circles with minimum radial separation to contain all the measurement points, which gives a more reasonable circularity error evaluation. The MZC is essentially a nonlinear optimization problem and computing its exact solution is much harder. A similar problem exists for spherical parts, which are also ubiquitous in manufacturing. Analogous to the criteria for circularity error evaluation, there are four equivalent criteria for sphericity error evaluation: least square sphere (LSS), minimum circumscribed sphere (MCS), maximum inscribed sphere (MIS), and minimum zone sphere (MZS).

So far, there are a number of methods of circularity error evaluation. Based on computational geometry involving a convex hull and Voronoi diagram, Roy and Zhang [1, 2] derived that the MZC center is located at one vertex of the farthest and nearest Voronoi diagrams, which are the intersections of the farthest and nearest Voronoi convex sets of all data points, respectively. Huang [3] further divided the vertices into X- and Y-types and showed that only the X-type vertex can give the optimal solution. Using the concepts of convex hulls and equidistant (Voronoi) and equiangular diagrams, Samuel and Shunmugam [4] proposed another method for MZC error evaluation based on computational geometric techniques using outer and inner convex hulls. Li and Shi [5–7] further used convex hulls with a curvature technique in computing MCC, MIC, and MZC errors. Based on the outer and inner convex hulls, software was developed to determine the MCC, MIC, and MZC errors from measurement points [8]. Jywe *et al* [9] constructed three geometric models to select the exact control points for MCC, MIC, and MZC errors. Noticing that the circularity error can be determined from a small number of critical data points, Huang [10] proposed a new strategy for improving computational efficiency by collecting the farthest and nearest data points from the current minimum radial separation center until all collected data points meet an optimal condition. In addition, the evaluation of the MZC error was often treated algebraically as an optimization problem and solved using various techniques, such as iterative search approaches [11–17], evolutionary algorithms [18, 19], a particle swarm optimization algorithm [20] and a linear programming method [21]. Recently, Rhinithaa *et al* [22] conducted a comparative study of several selected algorithms and a new geometric algorithm using the reflection mapping technique.

Sphericity error evaluation is actually a mathematical generalization of circularity error evaluation from 2D space to 3D space. However, the one-dimension increase makes the problem much harder to solve. Huang [23] extended the method based on Voronoi diagrams [3] to MZS error evaluation. Similarly to the work [9], Chen and Liu [24] proposed four models of control points to evaluate the MZS error. Samuel and Shunmugam [25, 26] expanded their method [4] for MZC error evaluation to the MZS error. Lei *et al* [27] extended the search algorithm [16] to sphericity error evaluation. Fan and Lee [28] cast the problem of MZS error evaluation into the problem of minimum potential energy of a simulated mechanical system and proposed a search algorithm for the optimal MZS solution around the LSS solution. Wen and Song [29] proposed an immune evolutionary algorithm for the MZS error. Huang *et al* [30] computed the MZS error using a search algorithm with diverse search directions and adaptive step sizes. In addition, an online validation service to test minimum zone algorithms has been provided by TraCIM (Traceability for Computationally-Intensive Metrology) [31], an international head organization for metrological algorithm validation.

In this paper, a unified branch-and-bound (B&B) algorithm is proposed to compute the MZC/MZS errors. The MZC/MZS error can be generally written as

$$e \triangleq \min_{\boldsymbol{p}_0 \in Q} \left\{ \max_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{p}_0\| - \min_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{p}_0\| \right\} \tag{1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^d$, $i = 1, 2, ..., n$ are n measurement points in d-dimensional space $\mathbb{R}^d$, $\boldsymbol{p}_0 \in \mathbb{R}^d$ is the center of the concentric circles/spheres, Q is the solution domain, and $d = 2$ or 3. While taking the domain Q to be a square region that is believed to contain the globally optimal solution, a lower bound on the MZC/MZS error attainable over the domain is derived and proven to monotonically increase as the square domain is divided into smaller square subdomains. With this lower bound, applying the B&B technique to sequentially dividing the domain with the smallest lower bound yields that the smallest lower bound is guaranteed to converge to the MZC/MZS error. Furthermore, the accuracy of the algorithm's result is proven to be within a user-specified tolerance for stopping the division of a domain, which provides users with the freedom to control the computation accuracy as needed. In addition, the proposed algorithm is efficient. It takes several milliseconds on a modern PC to compute the MZC/MZS error on a thousand data points distributed over a whole circle/sphere, or tens to hundreds of milliseconds even if the data points are gathered in a small sector. In comparison with the existing algorithms, which often need sophisticated mathematics, the most distinctive feature of the proposed algorithm is that it is extremely simple and can be implemented in several lines of computer programs by anyone with basic algebraic knowledge. Its performance has been verified on a number of point sets and shown to be in accordance with other methods.

The rest of this paper is organized as follows. Section 2 presents the algorithm, followed by numerical examples showing its performance in section 3. Section 4 concludes this paper.

2. Algorithm

Let $\boldsymbol{p}_0^*$ be a globally optimal solution such that

$$e = \max_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{p}_0^*\| - \min_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{p}_0^*\|. \tag{2}$$

In general, the point $\boldsymbol{p}_0^*$ is not unique and all of them give the same minimal value e. The proposed algorithm in this paper is aimed at finding one such point for which e is attained. Assume that Q is a square domain containing $\boldsymbol{p}_0^*$. Let $\boldsymbol{q}_0$ be the center of Q and $D(Q)$ the diagonal length of Q, both of which can be easily calculated for a given square domain Q. Let $\tilde{e}(Q)$ be defined as

$$\tilde{e}(Q) \triangleq \max_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_0\| - \min_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_0\|. \tag{3}$$

By the triangular inequalities for the triangle formed by $\boldsymbol{p}_i$, $\boldsymbol{q}_0$, and $\boldsymbol{p}_0^*$, i.e. $\|\boldsymbol{p}_i - \boldsymbol{q}_0\| \leqslant \|\boldsymbol{p}_i - \boldsymbol{p}_0^*\| + \|\boldsymbol{p}_0^* - \boldsymbol{q}_0\|$ and $\|\boldsymbol{p}_i - \boldsymbol{q}_0\| \geqslant \|\boldsymbol{p}_i - \boldsymbol{p}_0^*\| - \|\boldsymbol{p}_0^* - \boldsymbol{q}_0\|$, from equations (2) and (3) it follows that

$$\tilde{e}(Q) \leqslant e + 2\|\boldsymbol{p}_0^* - \boldsymbol{q}_0\|. \tag{4}$$

Since $\boldsymbol{p}_0^*$ is contained in Q, the following inequality holds:

$$\|\boldsymbol{p}_0^* - \boldsymbol{q}_0\| \leqslant D(Q)/2. \tag{5}$$

From equations (4) and (5) it then follows that

$$\tilde{e}(Q) \leqslant e + D(Q). \tag{6}$$

Let $l(Q)$ be defined as below, and from equation (6) it follows that

$$l(Q) \triangleq \tilde{e}(Q) - D(Q) \leqslant e. \tag{7}$$

Hence, $l(Q)$ gives a lower bound on the value e over domain Q. It is noteworthy that such a lower bound can be obtained for any square domain containing the globally optimal solution $\boldsymbol{p}_0^*$ to the problem (1). For a domain not containing $\boldsymbol{p}_0^*$, $l(Q)$ can still be defined as equation (7) but it would not be bounded above by e since the inequality (5) does not necessarily hold any more.

Algorithm 1 describes the proposed B&B algorithm for e, which sequentially divides domains into smaller ones such that a sufficiently small domain is attained and gives the globally optimal solution $\boldsymbol{p}_0^*$ to the problem (1). At every iteration of the algorithm, one domain Q^* for which $l(Q^*)$ is the smallest among all domains is selected from the list $\mathcal{L}$ of domains and divided into 2^d square subdomains (namely, four quadrants and eight octants in the case of $d = 2$ and 3, respectively), using lines/planes through its center $\boldsymbol{q}_0^*$ orthogonal to the coordinate axes. Then, the new domains $Q_j's$ are added to $\mathcal{L}$, and Q^* is removed from $\mathcal{L}$. It should be noted that the union of domains in $\mathcal{L}$ is always equal to the initial domain Q and there is at least one domain in $\mathcal{L}$ containing the optimal solution $\boldsymbol{p}_0^*$ as long as Q contains it. Since domains here are closed and adjacent domains share borders, on occasion the point $\boldsymbol{p}_0^*$ could lie on the shared border of two or more domains. However, this does not affect the convergence of the proposed algorithm, as proven by the following.

Let Q_j be one of the subdomains of Q^* and $\boldsymbol{q}_j$ the center of Q_j. It can be proven that $l(Q^*) \leqslant l(Q_j)$. First, it is easy to derive $\|\boldsymbol{q}_j - \boldsymbol{q}_0^*\| = D(Q_j)/2$ and $D(Q^*) = 2D(Q_j)$. Then, similarly to the derivation of equation (6) from equation (3), by the triangular inequalities $\|\boldsymbol{p}_i - \boldsymbol{q}_j\| - \|\boldsymbol{q}_j - \boldsymbol{q}_0^*\| \leqslant \|\boldsymbol{p}_i - \boldsymbol{q}_0^*\| \leqslant \|\boldsymbol{p}_i - \boldsymbol{q}_j\| + \|\boldsymbol{q}_j - \boldsymbol{q}_0^*\|$ it can be deduced that

$$\begin{aligned}\tilde{e}(Q^*) &= \max_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_0^*\| - \min_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_0^*\| \\ &\leqslant \max_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_j\| - \min_{i\in[1,n]} \|\boldsymbol{p}_i - \boldsymbol{q}_j\| + 2\|\boldsymbol{q}_j - \boldsymbol{q}_0^*\| \\ &= \tilde{e}(Q_j) + D(Q_j).\end{aligned} \tag{8}$$

Therefore, from equations (7) and (8) it follows that

$$\begin{aligned}l(Q^*) &= \tilde{e}(Q^*) - D(Q^*) \\ &\leqslant \tilde{e}(Q_j) + D(Q_j) - D(Q^*) \\ &= \tilde{e}(Q_j) - D(Q_j) \\ &= l(Q_j).\end{aligned} \tag{9}$$

This confirms that $l(Q^*)$ monotonically increases during the iteration of the B&B algorithm. At the same time, the inequality (7) holds for any square domain containing the globally optimal solution $\boldsymbol{p}_0^*$ and there exists a domain in the list $\mathcal{L}$ containing $\boldsymbol{p}_0^*$. In addition, $l(Q^*)$ is the minimum for all domains in $\mathcal{L}$. Hence, $l(Q^*)$ is bounded above by e. When Q^* is infinitely small and a singleton, $l(Q^*) = \tilde{e}(Q^*) = e$. Therefore, it is guaranteed that Q^* will converge to the globally optimal solution $\boldsymbol{p}_0^*$ to the problem (1), and the value $\tilde{e}(Q^*)$ to e. Moreover, from the inequality (6) an elegant property for the final Q^* by the stopping criteria $D^* \leqslant \epsilon$ can be deduced:

$$0 \leqslant \tilde{e}(Q^*) - e \leqslant \epsilon \tag{10}$$

where $\tilde{e}(Q^*)$ and e represent the computed and true values of the MZC/MZS error, respectively, and ϵ is a user-specified non-negative scalar representing the termination tolerance of the algorithm. Equation (10) implies that the error in the result and the accuracy of the algorithm can be easily controlled through ϵ. Another major advantage of the proposed algorithm is that it is very simple and easy to implement.

There are two user-specified parameters for the algorithm, namely the initial domain Q and the termination tolerance ϵ. In practice, the minimum requirement for choosing an initial domain is that it should contain the optimal solution $\boldsymbol{p}_0^*$. Since there is no other constraint on its choice, the initial domain can be taken to be arbitrarily big as long as it is believed to contain $\boldsymbol{p}_0^*$. On this basis, a smaller initial domain is more beneficial and can reduce the number of iterations and the computation time required by the algorithm to compute $\boldsymbol{p}_0^*$. While measuring a practical object, one may have the object's dimensions or an estimation, which could help determine an appropriate initial domain. In addition, ϵ can be set to match the precision of measured data points or the requirement in practical scenarios.

Algorithm 1. Circularity/sphericity evaluation.

Input: $\boldsymbol{p}_i$, $i = 1, 2, ..., n$
Output: Circularity/sphericity error e and center $\boldsymbol{p}_0^*$
1: Select a domain Q and calculate $D(Q)$
2: $\mathcal{L} \leftarrow \emptyset$
3: $D^* \leftarrow D(Q)$ and $Q^* \leftarrow Q$
4: **while** $D^* > \epsilon$ **do**
5: Divide domain Q^* into 2^d subdomains Q_j
6: Calculate $D(Q_j)$, $\tilde{e}(Q_j)$, and $l(Q_j)$ for each Q_j
7: Add all Q_j's to $\mathcal{L}$
8: $Q^* \leftarrow$ any domain in $\mathcal{L}$ in which $l(Q_j)$ is minimal
9: $D^* \leftarrow$ the value $D(Q_j)$ of Q^*
10: Remove Q^* from $\mathcal{L}$
11: **end while**
12: **return** $\tilde{e}(Q^*)$ and the center of Q^*

3. Examples

In this section, the performance of algorithm 1 is verified with numerical examples. The algorithm has been implemented in MATLAB and run on a desktop with an Intel Core i7-6700 3.40 GHz CPU and 16 GB RAM on which its CPU running times for the following examples were collected.

Example 1. The purpose of this example is to verify the computational accuracy and efficiency of algorithm 1 for the MZC error. In this example, $n - 8$ points are randomly generated between two circles centered at the origin with radii of 9.5 and 10.5, and another eight points are taken at $[-9.50^T]$, $[9.50^T]$,

Table 1. Comparison of different algorithms for circularity evaluation.

Data set	Method	x_c (mm)	y_c (mm)	e (μm)
Appendix A [9]	[9]	—	—	8.5
	[18]	—	—	8.5
	[14]	0.035 615	−0.052 9295	8.537 46
	[7]	0.0356	−0.0529	8.6
	[19]	—	—	8.5
	[20]	0.035 615 098 379 2411	−0.052 929 337 039 9840	8.536 709 71
	[17]	0.035 614 97	−0.052 929 48	8.54
	Ours	0.035 614 971 221	−0.052 929 481 201	8.537 464 355
Table C2 [4]	[4]	40.0007	50.0015	29.816
	[13]	40.000 739	50.001 530	29.280 1747
	[7]	40.0007	50.0015	29.3
	Ours	40.000 739 456 991	50.001 530 105 661	29.280 174 780
Appendix A [10]	[10]	0.005 36	0.007 88	957.35
	[18]	0.005 36	0.007 88	957.35
	[14]	0.005 355 47	0.007 880 73	957.354
	[7]	0.005 35	0.007 91	957.42
	[16]	0.005 3758	0.007 7843	957.4438
	[17]	0.005 3467	0.007 909 06	957.42
	Ours	0.005 346 707 309	0.007 909 059 150	957.419 945 646
Table B1 [13]	[13]	82.990 941	97.008 387	38.231
	[18]	82.990 941	97.008 387	38.231
	[14]	82.9909	97.0084	38.2309
	[6]	82.9909	97.0084	38.2
	[7]	82.9909	97.0084	38.2
	[19]	—	—	38.231
	[16]	82.990 9411	97.008 3872	38.231 04
	[17]	82.990 941 40	97.008 3875	38.2304
	Ours	82.990 941 049 445	97.008 387 267 061	38.230 943 982

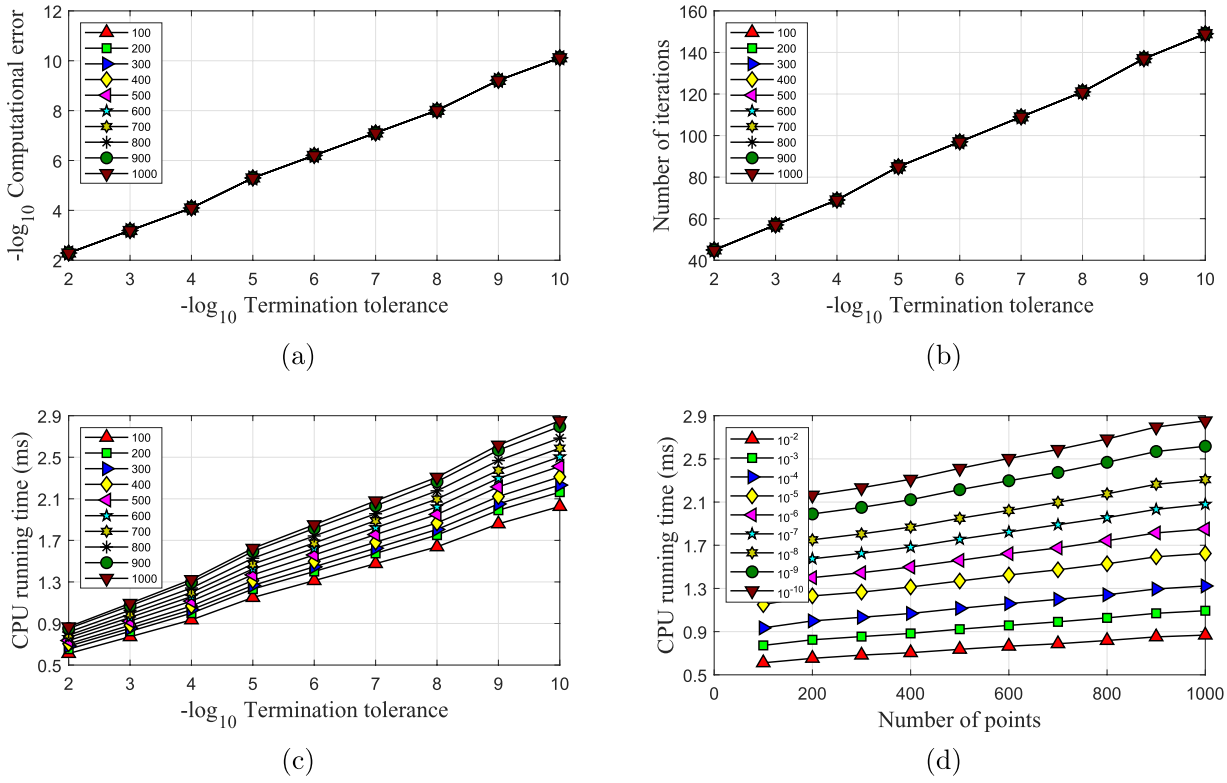

Figure 1. Performance of algorithm 1 initialized by the smallest bounding box to compute the MZC error.

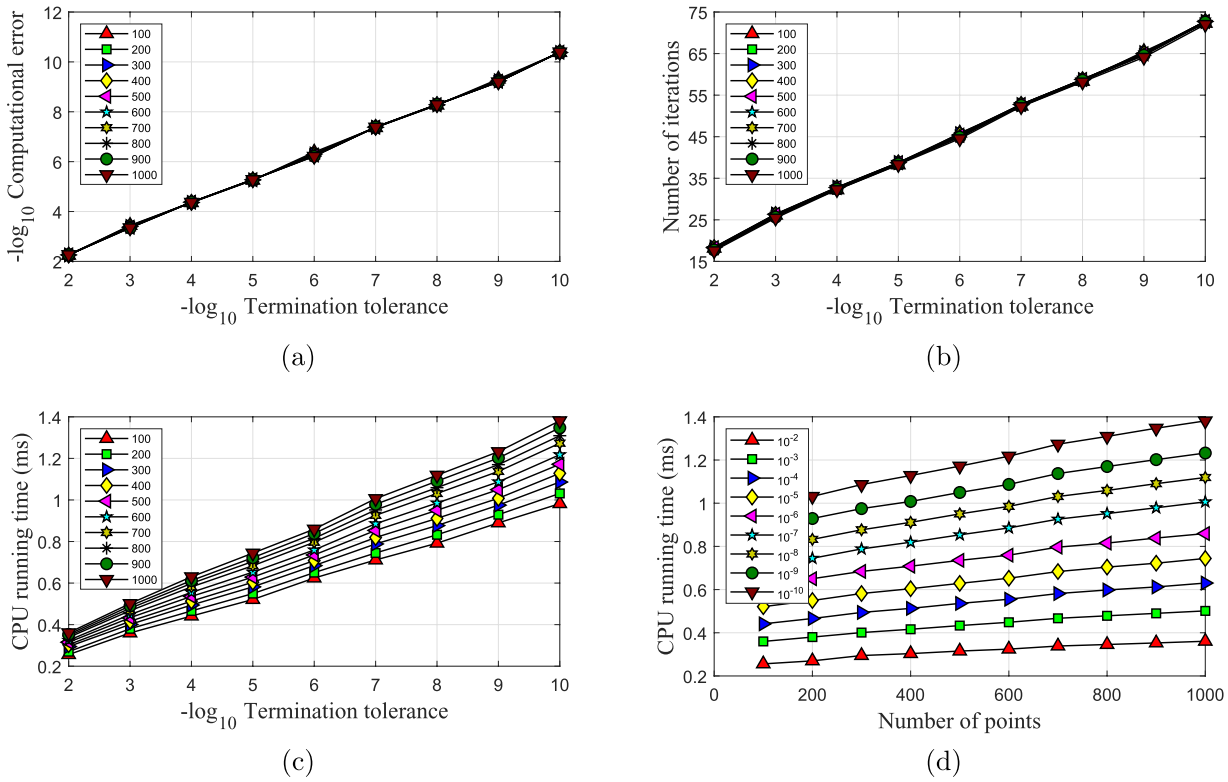

Figure 2. Performance of algorithm 1 initialized by the least square solution to compute the MZC error.

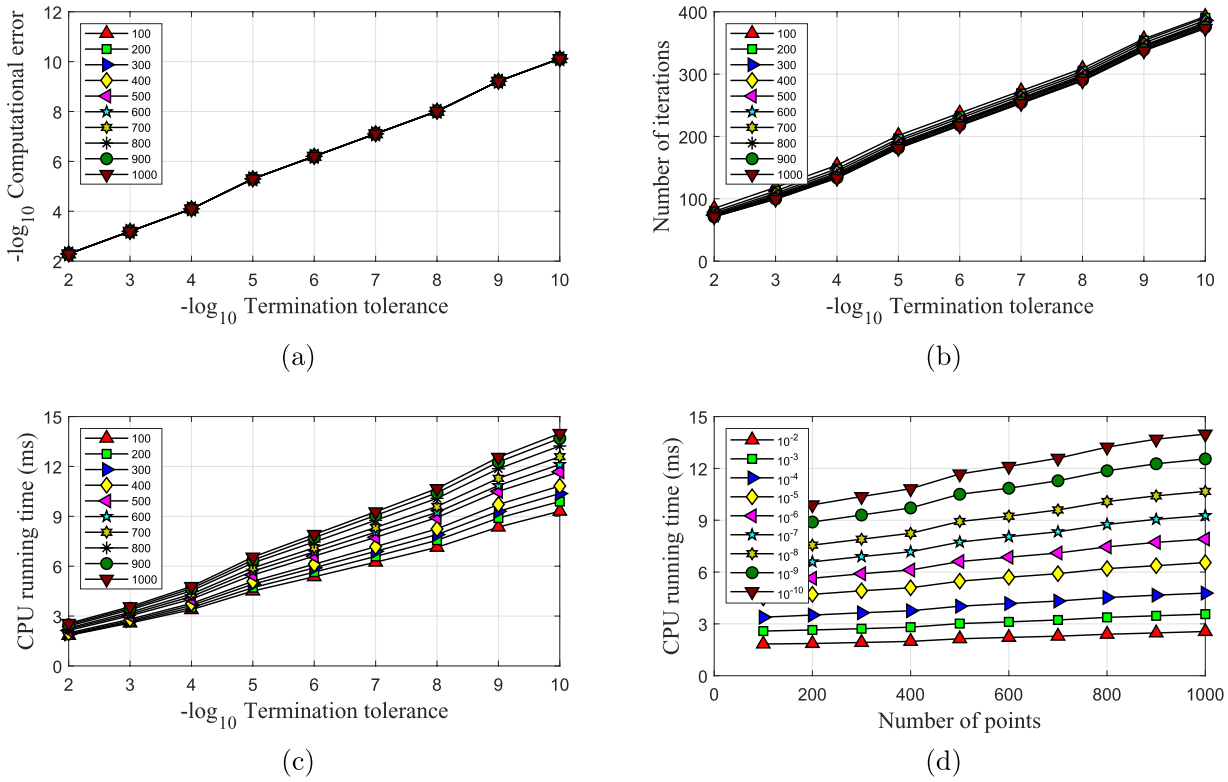

Figure 3. Performance of algorithm 1 initialized by the smallest bounding box to compute the MZS error.

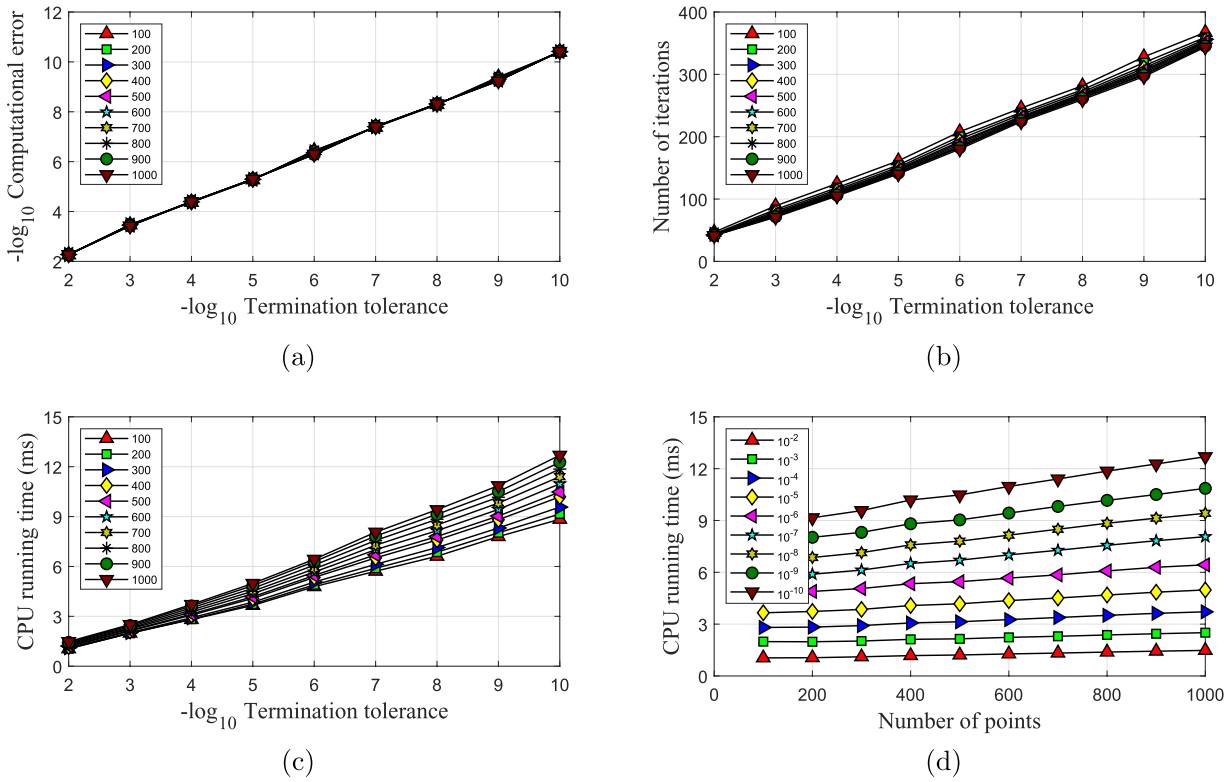

Figure 4. Performance of algorithm 1 initialized by the least square solution to compute the MZS error.

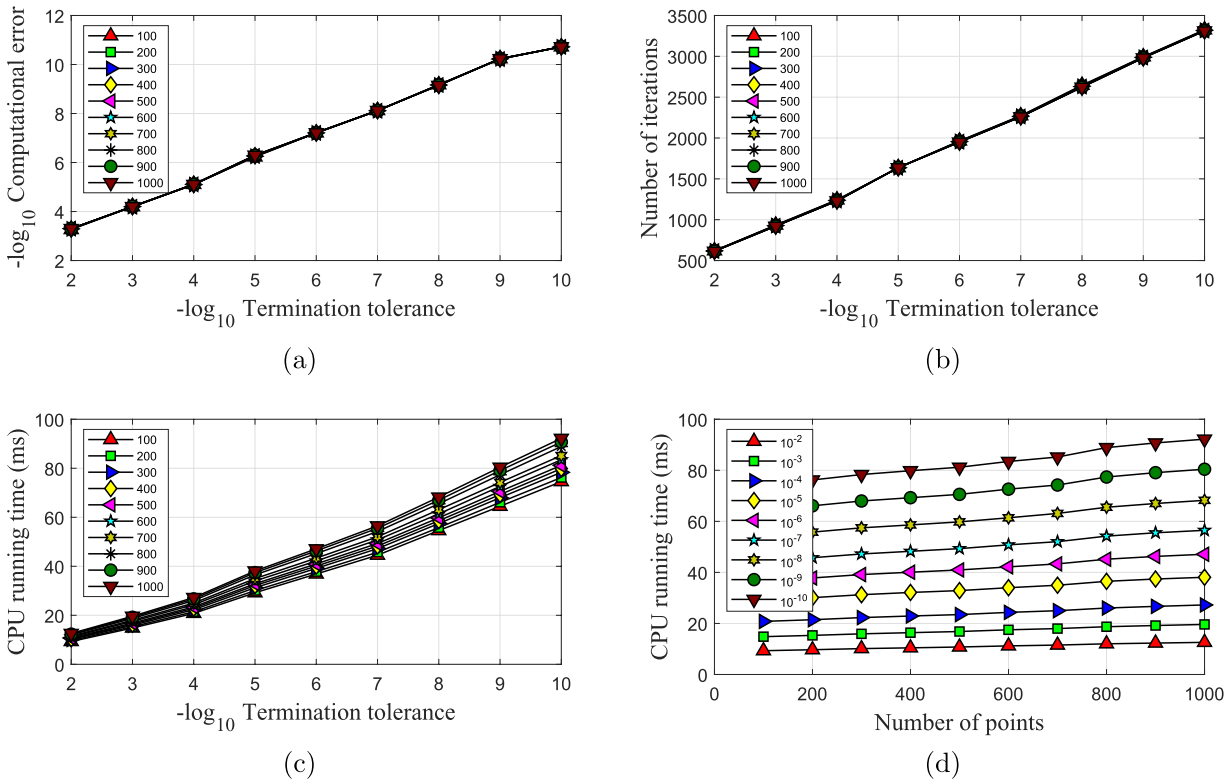

Figure 5. Performance of algorithm 1 to compute the MZC error in the case where data points are distributed in a small sector.

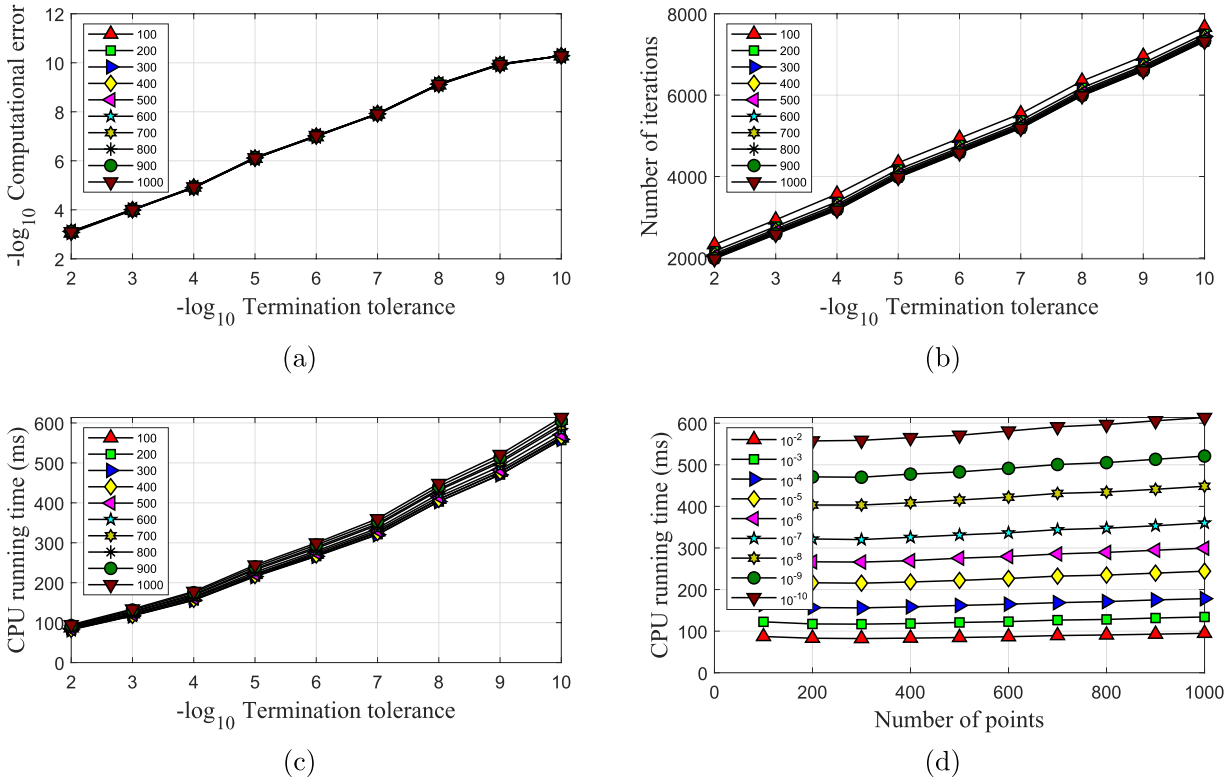

Figure 6. Performance of algorithm 1 to compute the MZS error in the case where data points are distributed in a small sector.

Table 2. Comparison of different algorithms for sphericity evaluation.

Data set	Method	x_c (mm)	y_c (mm)	z_c (mm)	e (μm)
Table 1 [28]	[28]	—	—	—	7.66
	[26]	0.001 118 08	0.000 414 94	−0.000 172 66	7.660 12
	[29]	0.002 495	−0.000 097	0.000 479	7.660
	[30]	0.002 504 16	−0.000 096 13	0.000 481 53	7.660 19
	Ours	0.002 504 156 298	−0.000 096 126 920	0.000 481 529 278	7.660 194 938
Example 1 [24]	[24]	0.003 911	0.002 535	0.004 562	8.327
	[29]	0.003 910	0.002 536	0.004 556	8.327
	[27]	0.003 901	0.002 624	0.004 427	8.336
	[30]	0.003 910 96	0.002 534 75	0.004 561 46	8.326 84
	Ours	0.003 910 946 122	0.002 534 741 343	0.004 561 521 573	8.326 841 243
Example 2 [24]	[24]	0.003 509	−0.003 305	−0.000 292	9.67
	[29]	0.003 506	−0.003 308	−0.000 292	9.669
	[27]	0.003 506	−0.003 289	−0.000 266	9.672
	[30]	0.003 509 03	−0.003 305 22	−0.000 292 24	9.666 21
	Ours	0.003 508 991 629	−0.003 305 242 365	−0.000 292 261 803	9.666 208 620
Table 1 [25]	[25]	40.001 41	30.000 24	60.000 59	2.828 42
	[30]	40.001 414 21	30.000 242 65	60.000 585 80	2.828 43
	Ours	40.001 414 215 042	30.000 242 636 914	60.000 585 787 641	2.828 429 604

$[0 - 9.5^T]$, $[09.5^T]$, $[-10.50^T]$, $[10.50^T]$, $[0 - 10.5^T]$, and $[010.5^T]$ so that the ground truth of the MZC error is known to be 1. The number of points n increases from 100 to 1000 and the termination tolerance ϵ for the algorithm decreases from 10^{-2} to 10^{-10}. For each combination of n and ϵ, the algorithm is tested on 1000 random point sets to collect the average error in the computed results compared with the ground truth, as well as the average number of iterations and CPU running time of the algorithm.

When the initial domain Q for algorithm 1 is taken to be the smallest square bounding box of all data points, the performance of the algorithm is depicted in figure 1. From figure 1(a) it can be seen that the level of accuracy is proportional to and bounded below by $-\log_{10}\epsilon$, as stated by equation (10). From

figure 1(b), the number of iterations increases linearly with $-\log_{10}\epsilon$ and seems independent of the number of points, since all the plotted lines overlap with each other. From figures 1(c) and (d), the CPU running time of the algorithm is linear with $-\log_{10}\epsilon$ and the number of points, respectively.

The initial domain Q can also be taken to be a square centered at the LSC center with a size of twice the LSC error. By doing this, the number of iterations and the CPU running time of the algorithm are reduced by half, as shown in figures 2(b)–(d), whereas the computational accuracy remains at the same level, as shown in figure 2(a).

Example 2. To verify the performance of algorithm 1 for the MZS error, a similar setting to Example 1 is used; that is, randomly generating $n-12$ points between two origin-centered spheres with radii of 9.5 and 10.5, and arranging 12 points at the intersections of the spheres with the x, y, and z axes of the coordinate frame such that the ground truth of the MZS error is 1. The algorithm's performance with the two aforementioned different initialization strategies in Example 1 is shown in figures 3 and 4, respectively. Again, the level of accuracy and the number of iterations are linear with $-\log_{10}\epsilon$, while the CPU running time is linear with $-\log_{10}\epsilon$ or the number of points. In this case, nevertheless, the initialization with the least square solution helps only a little to improve the computational efficiency of the algorithm.

Example 3. In this example, the proposed algorithm is applied to the case where data points are distributed in a small sector between the concentric circles/spheres centered at the origin as in the previous two examples. By describing a circle with the polar coordinates (r,ϕ), in this case, r is varied from 9.5 to 10.5 and ϕ from 0 to $\pi/4$ to generate random data points. Six points at $(9.5,0)$, $(9.5,\pi/8)$, $(9.5,\pi/4)$, $(10.5,0)$, $(10.5,\pi/8)$, and $(10.5,\pi/4)$ are added to the data points so that the ground truth of the MZC error is exactly 1. Similarly, by using the spherical coordinates (r,θ,ϕ), data points for the MZS test are generated within $r\in[9.5,10.5]$, $\theta\in[0,\pi/4]$, and $\phi\in[0,\pi/4]$. Again, to ensure a unit MZS error, ten points at $(9.5,0,0)$, $(9.5,0,\pi/4)$, $(9.5,\pi/4,0)$, $(9.5,\pi/4,\pi/4)$, $(9.5,\pi/8,\pi/8)$, $(10.5,0,0)$, $(10.5,0,\pi/4)$, $(10.5,\pi/4,0)$, $(10.5,\pi/4,\pi/4)$, and $(10.5,\pi/8,\pi/8)$ are imposed.

The initial domain for the algorithm is taken to be a square domain centered at the LSC/LSS center. However, it has been noticed that the LSC/LSS center can significantly deviate from the MZC/MZS center in the case of partially distributed points. Hence, the size of the initial domain is taken to be four times the radius of the LSC/LSS, which is believed to be large enough to contain the MZC/MZS center. Figures 5 and 6 show the algorithm's performance in the two cases, respectively. It can be seen that the proposed algorithm can still compute the MZC/MZS errors accurately, but its required number of iterations and computation time are notably greater than in the previous examples.

Example 4. In this example, the proposed algorithm is run on several existing data point sets and compared with other methods. The selected data point sets are the most popular ones that have been frequently used to test circularity/sphericity error algorithms. Tables 1 and 2 display the results of our algorithm together with the results taken directly from the original work. Since the original work provide results with different approximations, we set the termination tolerance $\epsilon=10^{-12}$ for our algorithm to attain an accuracy level higher than all the others. Then, it can be seen that the accuracy of our algorithm is comparable with others and even higher in many cases.

4. Conclusion

In this paper, a unified algorithm for minimum zone circularity/sphericity errors based on the B&B technique has been presented. For any square domain, a lower bound on the minimal radial difference between two concentric circles/spheres centered in the domain sandwiching given points is proposed. It is proven that the lower bound monotonically increases when the domain is divided into smaller square subdomains. At every iteration of the algorithm, the subdomain with the smallest lower bound is chosen and further divided. As the algorithm iterates and the subdomains become smaller, the smallest lower bound monotonically increases and converges to the exact value of the desired error as long as the initial domain contains the solution. The proposed algorithm is accurate, efficient, and very simple. In addition, a parallel implementation of the algorithm could further improve its computational efficiency as it usually does for B&B algorithms.

ORCID iDs

Yu Zheng https://orcid.org/0000-0002-4617-3252

References

[1] Roy U and Zhang X 1992 Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error *Comput. Aided Des.* **24** 161–8
[2] Roy U and Zhang X 1994 Development and application of Voronoi diagrams in the assessment of roundness error in an industrial environment *Comput. Ind. Eng.* **26** 11–26
[3] Huang J 1999 An exact solution for the roundness evaluation problems *Precis. Eng.* **23** 2–8
[4] Samuel G L and Shunmugam M S 2000 Evaluation of circularity error from coordinate measurement data using computational geometric techniques *Precis. Eng.* **24** 251–63
[5] Li X M and Shi Z Y 2008 Development and application of convex hull in the assessment of roundness error *Int. J. Mach. Tools Manuf.* **48** 135–9
[6] Li X M and Shi Z Y 2008 Application of convex hull in the assessment of roundness error *Int. J. Mach. Tools Manuf.* **48** 711–4
[7] Li X M and Shi Z Y 2010 Evaluation of roundness error from coordinate data using curvature technique *Measurement* **43** 164–8
[8] Gadelmawla E S 2010 Simple and efficient algorithms for roundness evaluation from the coordinate measurement data *Measurement* **43** 223–35
[9] Jywe W-Y, Liu C-H and Chen C-K 1999 The min-max problem for evaluating the form error of a circle *Measurement* **26** 273–82
[10] Huang J 2001 A new strategy for circularity problems *Precis. Eng.* **25** 301–8

[11] Wang M, Cheraghi S H and Masud A S M 1999 Circularity error evaluation: theory and algorithm *Precis. Eng.* **23** 164–76
[12] Dhanish P B 2002 A simple algorithm for evaluation of minimum zone circularity error from coordinate data *Int. J. Mach. Tools Manuf.* **42** 1589–94
[13] Zhu L M, Ding H and Xiong Y L 2003 A steepest descent algorithm for circularity evaluation *Comput. Aided Des.* **35** 255–65
[14] Cui C C, Fan W and Huang F G 2010 An iterative neighborhood search approach for minimum zone circularity evaluation from coordinate measuring machine data *Meas. Sci. Technol.* **21** 027001
[15] Lei X Q, Zhang C Y, Xue Y J and Li J S 2011 Roundness error evaluation algorithm based on polar coordinate transform *Measurement* **44** 345–50
[16] Lei X Q, Pan W M, Tu X P and Wang S F 2014 Minimum zone evaluation for roundness error based on geometric approximating searching algorithm *MAPAN-J. Metrol. Soc. India* **29** 143–9
[17] Srinivasu D S and Venkaiah N 2017 Minimum zone evaluation of roundness using hybrid global search approach *Int. J. Adv. Manuf. Technol.* **92** 2743–54
[18] Wen X L, Xia Q G and Zhao Y B 2006 An effective genetic algorithm for circularity error unified evaluation *Int. J. Mach. Tools Manuf.* **46** 1770–7
[19] Wang D X, Song A G, Wen X L and Wang F L 2012 Circularity error evaluation based on differential evolution algorithm *Appl. Mech. Mater.* **143–144** 416–21
[20] Du C-L, Luo C-X, Han Z-T and Zhu Y-S 2014 Applying particle swarm optimization algorithm to roundness error evaluation based on minimum zone circle *Measurement* **52** 12–21
[21] Nouira H and Bourdet P 2014 Evaluation of roundness error using a new method based on a small displacement screw *Meas. Sci. Technol.* **25** 044012
[22] Rhinithaa P T, Selvakumar P, Nikhil S, Vysyaraju A, Deepak L and Jose M 2018 Comparative study of roundness evaluation algorithms for coordinate measurement and form data *Precis. Eng.* **51** 458–67
[23] Huang J 1999 An exact minimum zone solution for sphericity evaluation *Comput. Aided Des.* **31** 845–53
[24] Chen C-K and Liu C-H 2000 A study on analyzing the problem of the spherical form error *Precis. Eng.* **24** 119–26
[25] Samuel G L and Shunmugam M S 2001 Evaluation of sphericity error from coordinate measurement data using computational geometric techniques *Comput. Methods Appl. Mech. Eng.* **190** 6765–81
[26] Samuel G L and Shunmugam M S 2002 Evaluation of sphericity error from form data using computational geometric techniques *Int. J. Mach. Tools Manuf.* **42** 405–16
[27] Lei X Q, Gao Z B, Duan M D and Pan W M 2015 Method for sphericity error evaluation using geometry optimization searching algorithm *Precis. Eng.* **42** 101–12
[28] Fan K-C and Lee J-C 1999 Analysis of minimum zone sphericity error using minimum potential energy theory *Precis. Eng.* **23** 65–72
[29] Wen X L and Song A G 2004 An immune evolutionary algorithm for sphericity error evaluation *Int. J. Mach. Tools Manuf.* **44** 1077–84
[30] Huang J Z, Jiang L, Zheng H X, Xie M Z and Tan J B 2019 Evaluation of minimum zone sphericity by combining single-space contraction strategy with multi-directional adaptive search algorithm *Precis. Eng.* **55** 189–216
[31] Hutzschenreuter D, Härtig F, Wendt K, Lunze U and Löwe H 2015 Online validation of Chebyshev geometric element algorithms using the TraCIM-system *J. Mech. Eng. Autom.* **5** 94–111