

Research Article

Calculating the Mean Amplitude of Glycemic Excursions from Continuous Glucose Data Using an Open-Code Programmable Algorithm Based on the Integer Nonlinear Method

Xuefei Yu,¹ Liangzhuo Lin,¹ Jie Shen,² Zhi Chen,² Jun Jian,³
Bin Li,¹ and Sherman Xuegang Xin ⁴

¹Southern Medical University, Guangzhou, China

²The Third Affiliated Hospital of Southern Medical University, Guangzhou, China

³Guangzhou Power Supply Bureau Co., Ltd., Huadu Branch, Guangzhou, China

⁴School of Medicine, South China University of Technology, Guangzhou 520006, China

Correspondence should be addressed to Sherman Xuegang Xin; xinxg@scut.edu.cn

Received 20 September 2017; Revised 18 January 2018; Accepted 7 February 2018; Published 8 March 2018

Academic Editor: Jesús Picó

Copyright © 2018 Xuefei Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The mean amplitude of glycemic excursions (MAGE) is an essential index for glycemic variability assessment, which is treated as a key reference for blood glucose controlling at clinic. However, the traditional “ruler and pencil” manual method for the calculation of MAGE is time-consuming and prone to error due to the huge data size, making the development of robust computer-aided program an urgent requirement. Although several software products are available instead of manual calculation, poor agreement among them is reported. Therefore, more studies are required in this field. In this paper, we developed a mathematical algorithm based on integer nonlinear programming. Following the proposed mathematical method, an open-code computer program named MAGECAA v1.0 was developed and validated. The results of the statistical analysis indicated that the developed program was robust compared to the manual method. The agreement among the developed program and currently available popular software is satisfied, indicating that the worry about the disagreement among different software products is not necessary. The open-code programmable algorithm is an extra resource for those peers who are interested in the related study on methodology in the future.

1. Introduction

Clinical researches have suggested that high glycemic variability may cause more serious damage to the body than high level stable blood glucose [1], which relates to the development of diabetic complication [2–6] and the increase of mortality in critically serious patients without diabetes [7, 8]. In such circumstances, how to quantitatively evaluate the glycemic variability in diabetic blood glucose monitor is essential for the clinical diagnosis and treatment. Several indexes have been proposed for the quantitative evaluation of the glycemic variability, such as MBG, SDBG, IQR, LAGE, M-value, %CV, J-index, IGC, GRADE, MODD, LBG, HBGI, ADRR, TI, LI, PGS, CONGA, and MAGE [9–16]. Presently, clinicians and researchers trend to choose MAGE as the preferred index [17, 18], making it a “popular standard”

in the quantitative evaluation of the within-day glycemic variability.

MAGE is an arithmetic average of either the upward or downward of all glycemic excursions exceeding the threshold (standard deviation of blood glucose (SDBG) obtained from all blood glucose concentrations within 24-hour period); the direction of the calculation is determined by the first countable excursion [19–21]. Following the original definition by Service et al. [19], clinicians and researchers applied the “ruler and pencil” graphical approach to calculate MAGE. However, this kind of manual approach is time-consuming and error-prone, when dealing with huge amount of data typically characterized by 288 observations in 5 minutes apart over a 24-hour period, generated from continuous glucose monitoring (CGM). Thus, the development of computer-aided programs to calculate MAGE becomes an urgently needed task.

To address the urgent need above, several automatic programs have been developed [22–25], making great contribution to the automatic calculation of MAGE. However, it was pointed out that the agreement among them is poor [26]. Therefore, more studies are required to explore the automatic calculation of MAGE. Currently available programs can be roughly divided into two groups. One group has detailed descriptions about algorithm with graphical display, such as the automated algorithm described by Baghurst [22] and the computer software described by Fritzsche et al. (Fritzsche) [23]. The other group only provides executable software for automatic calculation, without detailed descriptions of the algorithms used in the software, such as the web-based application “GlyCulator” [24] and the Excel-based workbook “EasyGV” [25]. However, to the best of our knowledge, all of these methods did not provide programmable open codes, which are important resources required for peers to implement more studies on methodology in this field.

Therefore, in this report, we developed a mathematical algorithm based on integer nonlinear programming method. Following the proposed mathematical method, a computer-aided program named MAGECAA v1.0 was developed. The code of our program is open; if peers are interested in it, please contact xuefeiyu@smu.edu.cn for downloading. To validate the developed program, comparison study was implemented using blood glucose CGM datasets obtained from T1D, T2D, and gestational diabetes patients against the manual method (MAGEo) and other currently popular software products.

2. Materials and Methods

2.1. The Proposed Mathematical Algorithm. Let $f(t)$ ($t = t_1, t_2, \dots, t_L$) represent the discrete blood glucose values obtained in CGM; then f is a discrete function defined in time set $\{t_1, t_2, \dots, t_L\}$. Let SDBG be the standard deviation of $f(t)$ ($t = t_1, t_2, \dots, t_L$). A graph depicting the glycemic variability can be formed by connecting all the discrete values of function f . When the difference of a peak and an adjacent nadir exceeds SDBG, the corresponding peak is labeled as a valid peak. The key point for the calculation of MAGE is to correctly count the valid peak or nadir.

Amplitude is the difference of functional values in a peak and a nadir of the function graph. A valid amplitude is labeled and counted when it is bigger than SDBG. To compute MAGE, these valid amplitudes of function f should be firstly searched. From the mathematical perspective, a peak or nadir should be an extreme point of the function. These extreme points related to valid amplitudes are called valid extreme points. The MAGE computation problem can be solved by calculating all valid extreme points.

Suppose that $\{t_1, t_2, \dots, t_{l_N}\}$ is the sequence of all extreme points of function f . We know that the local maximum points and local minimum points of function f should be arranged as follows: interleaved between a maximum point and a minimum point. For simplicity, $\{t_1, t_2, \dots, t_{l_N}\}$ is denoted as $\{1, 2, \dots, N\}$, and $\{n_1, n_2, \dots, n_K\}$ is an arbitrary valid subsequence of sequence $\{1, 2, \dots, N\}$. These local

maximum points and local minimum points can be staggered in subsequence $\{n_1, n_2, \dots, n_K\}$. Then

$$\begin{aligned} 1 \leq n_1 < n_2 < \dots < n_K \leq N, \\ (-1)^{n_{k+1}-n_k} = -1, \quad k = 1, 2, \dots, K-1, \end{aligned} \quad (1)$$

where $2 \leq K \leq N$. This equation ensures that the selected extreme points are staggered.

Thus, the above MAGE computation problem can be transformed to an integer nonlinear programming (INLP) problem.

$$\begin{aligned} \arg \max_{K, n_1, n_2, \dots, n_K} Z_K(n_1, n_2, \dots, n_K) \\ = \arg \max_{K, n_1, n_2, \dots, n_K} \sum_{k=1}^{K-1} |f(t_{n_{k+1}}) - f(t_{n_k})|, \end{aligned} \quad (2)$$

$$\begin{aligned} \text{subject to } |f(t_{n_{k+1}}) - f(t_{n_k})| \geq \text{SDBG}, \\ k = 1, 2, \dots, K-1, \\ 1 \leq n_1 < n_2 < \dots < n_K \leq N, \\ (-1)^{n_{k+1}-n_k} = -1, \quad k = 1, 2, \dots, K-1. \end{aligned} \quad (3)$$

According to the principle of INLP problem, function (2) should have an optimum solution $\{n_1^*, n_2^*, \dots, n_K^*\}$ and an optimum value Z_K^* with respect to a constant K ; it should also have an optimum value Z_{K-1}^* for constant $K-1$, and

$$Z_{K-1}^* \leq Z_K^*. \quad (4)$$

Suppose that K^* is the maximal K with respect to the optimal solution of function (2), and then the optimal solution $\{n_1^*, n_2^*, \dots, n_{K^*}^*\}$ represents the valid extreme points to compute MAGE.

If the extreme point n_1^* is the local minimum point, then

$$\begin{aligned} \text{MAGE}_+ &= \frac{1}{[K^*/2]} \sum_{k=1}^{[K^*/2]} |f_{n_{(k-1)*2+2}} - f_{n_{(k-1)*2+1}}|, \\ \text{MAGE}_- &= \frac{1}{[(K^*-1)/2]} \sum_{k=2}^{[(K^*-1)/2]} |f_{n_{(k-1)*2+1}} - f_{n_{(k-1)*2}}|, \end{aligned} \quad (5)$$

$$\text{MAGE} = \text{MAGE}_+,$$

$$\text{MAGE}_a = \frac{\text{MAGE}_+ + \text{MAGE}_-}{2},$$

If the extreme point n_1^* is the local maximum point, then

$$\begin{aligned} \text{MAGE}_+ &= \frac{1}{[(K^*-1)/2]} \sum_{k=2}^{[(K^*-1)/2]} |f_{n_{(k-1)*2+1}} - f_{n_{(k-1)*2}}|, \\ \text{MAGE}_- &= \frac{1}{[K^*/2]} \sum_{k=1}^{[K^*/2]} |f_{n_{(k-1)*2+2}} - f_{n_{(k-1)*2+1}}|, \end{aligned} \quad (6)$$

$$\text{MAGE} = \text{MAGE}_-,$$

$$\text{MAGE}_a = \frac{\text{MAGE}_+ + \text{MAGE}_-}{2}.$$

Now, our object is to get the optimal solution $\{n_1^*, n_2^*, \dots, n_K^*\}$ of function (2). If the INLP problem is solved by enumeration algorithm, the amounts of different subsequences $\{n_1, n_2, \dots, n_K\}$ should be $\binom{N}{K}$. For $3 \leq K \leq N$, the amounts of all subsequences are

$$\binom{N}{N} + \binom{N}{N-1} + \dots + \binom{N}{3} \approx 2^N, \quad (7)$$

It is difficult to directly solve function (2) when N is large. A faster optimization algorithm is usually used to solve the above INLP problem. For example, a penalty function

$$\arg \min_{n_1, n_2, \dots, n_K} Y_K(n_1, n_2, \dots, n_K) = \arg \min_{K, n_1, n_2, \dots, n_K} \left[-\sum_{k=1}^{K-1} |f_{n_{k+1}} - f_{n_k}| + \sum_{k=1}^{K-1} \mu \max\{0, g_k\}^2 + \sum_{k=1}^{K-1} \lambda h_k^2 \right], \quad (9)$$

where $3 \leq K \leq N$ and μ_k and λ_k ($k = 1, \dots, K-1$) are penalty coefficients and tend to be $+\infty$.

When the optimum value $Y_K^* > 0$, problem (9) has no optimal solution.

Differential evolution (DE) algorithm, a faster optimization algorithm proposed by Storn and Price [29], is a simple but powerful population-based stochastic search technique to solve global optimization problems over continuous domains. Many researchers modified DE algorithm to improve its performance when it was applied to a specific problem [30–33]. The idea of a modified DE algorithm proposed by Lin [34] to solve mixed-integer nonlinear programming problem is used here.

DE searches for a global optimal point in an n -dimensional hyperspace. Let \bar{S} be the K -dimensional search space of the INLP problem under consideration. The DE evolves a population of NP n -dimensional individual vectors, that is, solution candidates, $\mathbf{N}_{iK} = (n_{i1}, \dots, n_{iK}) \in \bar{S}, i = 1, \dots, NP$, from one generation to the next. The evolution begins with a randomly initialized population of n -dimensional integer parameter vectors in space $\{1, 2, \dots, N\}^K$. In each vector, integer parameters are sorted in an ascending order. Each vector forms a candidate solution to the unconstrained optimization problem. At each generation G , DE employs the mutation and crossover operations to produce a trial vector \mathbf{U}_{iK}^G for each individual vector \mathbf{N}_{iK}^G , also called target vector, in the current population.

The details of the employment of the DE algorithm to solve the above INLP problem are as follows from (a) to (d).

(a) *Initialization.* A randomly initialized population is created to cover the entire search space uniformly as in the following form:

$$\mathbf{N}_{iK}^0 = (1, 1, \dots, 1) + \text{NINT}[(\rho_{i1}, \rho_{i2}, \dots, \rho_{iK}) \times (N-1)], \quad (10)$$

where ρ_{ij} is a random number in the range $[0, 1]$ and $\text{NINT}[\mathbf{B}]$ is expressed as the nearest integer vector to real vector \mathbf{B} .

algorithm [27, 28] was used here to transform the above INLP problem (2) to an unconstrained optimization problem as follows.

Set

$$\begin{aligned} g_k(n_1, n_2, \dots, n_K) &= \text{SDBG} - |f_{n_{k+1}} - f_{n_k}|, \\ h_k(n_1, n_2, \dots, n_K) &= (-1)^{n_{k+1} - n_k} + 1, \end{aligned} \quad (8)$$

$$k = 1, 2, \dots, K-1.$$

The new unconstrained integer optimization problem becomes

(b) *Mutation Operation.* For each target vector \mathbf{N}_{iK}^G at generation G , randomly sample three other individuals $\mathbf{N}_{r_1K}^G, \mathbf{N}_{r_2K}^G$, and $\mathbf{N}_{r_3K}^G$ from the same generation, where r_1, r_2 , and r_3 are random and mutually different integers generated over the range $[1, NP]$, which should be different from the current trial vector's index i . Then an associated mutant vector $\mathbf{V}_{iK}^G = (v_{i1}^G, \dots, v_{iK}^G)$ can be generated by using strategy:

$$\mathbf{V}_{iK}^G = \mathbf{N}_{r_1K}^G + \text{NINT}[F \times (\mathbf{N}_{r_2K}^G - \mathbf{N}_{r_3K}^G)], \quad (11)$$

where F is a factor in $[0, 1]$ for scaling differential vectors.

(c) *Crossover Operation.* The crossover operation is applied to each pair of the generated mutant vector \mathbf{V}_{iK}^G and its corresponding target vector \mathbf{N}_{iK}^G to generate a trial vector $\mathbf{U}_{iK}^G = (u_{i1}^G, u_{i2}^G, \dots, u_{iK}^G)$.

$$u_{ij}^G = \begin{cases} v_{ij}^G, & \text{rand}[0, 1] \leq \text{CR or } j = j_{\text{rand}}, \\ n_{ij}^G, & \text{otherwise,} \end{cases} \quad (12)$$

$$j = 1, 2, \dots, K,$$

where $\text{CR} \in [0, 1]$ is a crossover constant that is determined by users. j_{rand} is a randomly chosen index in $[1, K]$ which ensures that u_{ij}^G gets at least one parameter from v_{ij}^G . The integer parameters of target vector \mathbf{N}_{iK}^G are also sorted from small to big.

(d) *Selection Operation.* The trial vector \mathbf{U}_{iK}^G is compared to its corresponding target vector \mathbf{N}_{iK}^G using the greedy criterion to decide whether a member of generation $G+1$ existed or not. If vector \mathbf{U}_{iK}^G yields a smaller cost function value $Y(\mathbf{U}_{iK}^G)$ than $Y(\mathbf{N}_{iK}^G)$, then \mathbf{N}_{iK}^{G+1} is set to \mathbf{U}_{iK}^G ; otherwise, the old vector is retained. The operation is expressed as follows:

$$\mathbf{N}_{iK}^{G+1} = \begin{cases} \mathbf{U}_{iK}^G, & \text{if } Y(\mathbf{U}_{iK}^G) < Y(\mathbf{N}_{iK}^G), \\ \mathbf{N}_{iK}^G, & \text{otherwise.} \end{cases} \quad (13)$$

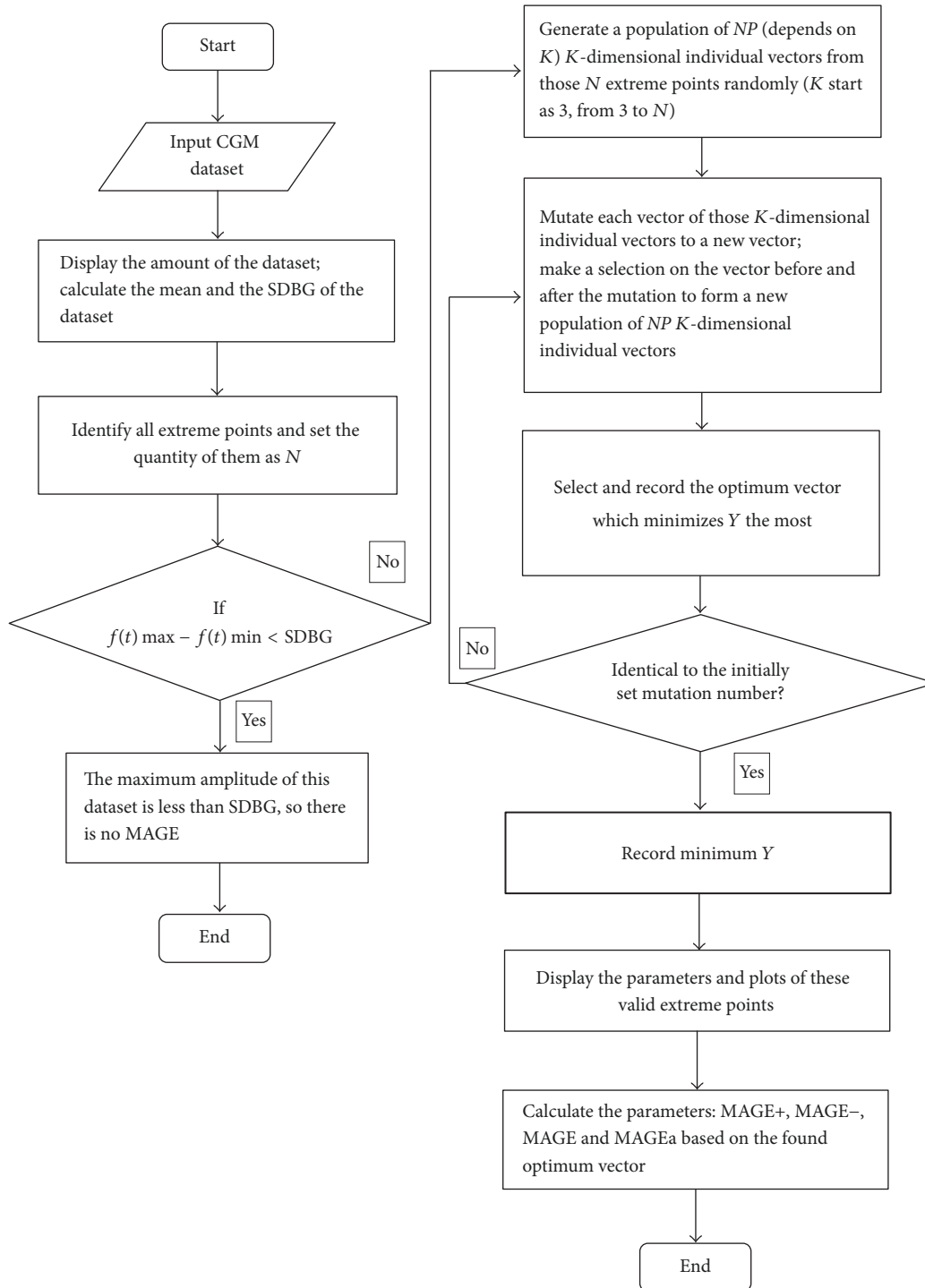


FIGURE 1: Flow of the algorithm of the program of MAGECAA v1.0.

The above (b) to (d) steps are repeated till the evolution times arrived to certain number (general $G = 200$); from the last evolutionary generation, the individual vector \mathbf{N}_{iK}^G with the smallest value of the objective function Y is the optimal solution of the problem of the current K -dimensional extreme point combination. The algorithm will search K from 3 to N till the objective function Y obtains the minimum

value so as to obtain the optimal extreme point combination $\{n_1^*, n_2^*, \dots, n_{K^*}^*\}$ and then use formulas (5)~(6) to calculate the MAGE value. Figure 1 shows the entire algorithm.

2.2. *The Developed Program Based on the Proposed Mathematical Method.* Based on the proposed mathematical method, a computer automated program named MAGECAA v1.0

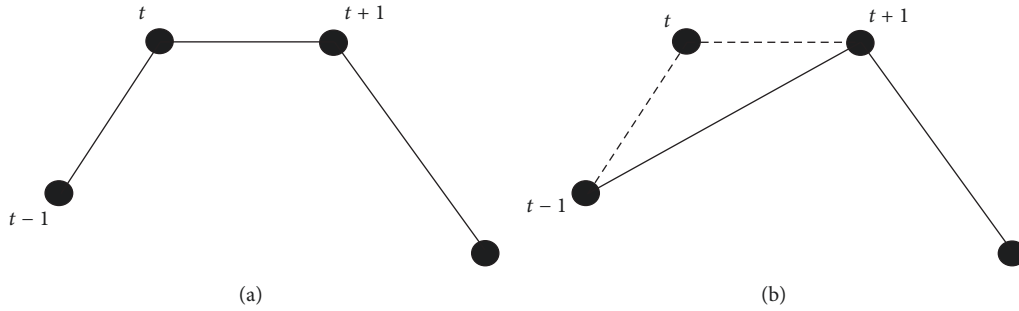


FIGURE 2: (a) shows turning points t and $t + 1$ to be equal in values (not being extreme points) but there is an amplitude from peak point $t - 1$ to nadir point t . (b) shows the combination of points t and $t + 1$ as one point $t + 1$ or remove point t ; then let point $t + 1$ be a local maximum point.

was developed. The MAGE calculation program can be described as a process that selects valid extreme points from a time-ordered set of glucose concentrations whose adjacent differences are all greater than the threshold (typically 1 SDBG obtained from 24-hour period blood glucose concentrations). It can be summarized as finding the optimum vector combination solution of the valid extreme points and using INLP to establish the mathematic method, which can be solved by differential evolution (DE) algorithm. Once all valid extreme points of countable excursions have been identified, the MAGE is determined by $MAGE_+$ or by $MAGE_-$, depending on the direction of the first countable excursion. For more in-depth understanding, it depends on the first valid extreme point of the vector combination, because that point indicates the direction. In addition, the average of both $MAGE_+$ and $MAGE_-$, designated as $MAGE_a$, is also calculated. MAGECAA v1.0 is based on INLP and has several different outputs: SDBG, $MAGE_+$, $MAGE_-$, MAGE, $MAGE_a$, and so forth. Besides it also needs plot to show all valid extreme points joined by straight lines; MATLAB (MathWorks®, USA) is chosen as the programming environment accordingly. Generally, some data points could not be extreme point according to the mathematical definition, like points shown in Figure 2, which shows turning points t and $t + 1$ which are equal in values (not being extreme points) but there is an amplitude from peak point $t - 1$ to nadir point t , so we combine points t and $t + 1$ as one point $t + 1$ or remove point t and then let point $t + 1$ be a local maximum point. If the points appear to be opposite, then point $t + 1$ should be a local minimum point.

MAGECAA v1.0 consists of the following major modules: (1) import CGM data and calculate the SDBG as the threshold; (2) identify all extreme points; (3) find the optimum vector combination solution of valid extreme points; and (4) display the calculated parameters and plots.

2.3. Collection of CGM Data. The CGM datasets obtained by using CGMS® Gold™ (Medtronic®, USA), collected from clinical treatment, are used to evaluate the proposed program. All CGM datasets were provided by the Third Affiliated Hospital of Southern Medical University. Only complete 24-hour CGM data were selected for comparison study. All patients have provided their written informed consent. 5 CGM recordings from 3 T1D patients, 116 CGM measurements contributed by 58 T2D patients, and 127 CGM

measurements based on gestational diabetes patients have been collected. Outpatients had been treated with either diet, oral hypoglycemic agents, oral hypoglycemic agents plus insulin, or insulin alone, depending on their glycemic control.

2.4. Data Analysis. The validation of MAGECAA v1.0 was implemented by comparison against MAGEo and MAGEc. A doctor who has been well trained in using the original manual method to analyze CGM data was invited from Department of Endocrinology of the Third Affiliated Hospital of Southern Medical University, and he did not know the effect of MAGECAA v1.0. Our research team analyzed the complete patient population ($n = 248$) CGM measurements using MAGECAA v1.0 and two other popular software products, EasyGV and Fritzsche, whereas the doctor performed a manual analysis on a randomly drawn sample ($n = 60$). As calculator, Fritzsche allows the analyzer to choose whether or not to consider the first and/or the final glucose value of the CGM trace as a start or end point of a glucose excursion; it will result in four different MAGE values per CGM trace. To keep following the original description of the calculation of MAGE and for the comparison of the different software products, the first and final glucose values were both taken into account.

2.5. Statistical Analyses. Spearman's correlation analysis was applied in evaluating the relationship between the MAGE values obtained by different methods with respect to the same patients. Bland-Altman plots were used to represent the agreement of the methods [35, 36]. $P < 0.01$ was considered significant.

3. Results

Figure 3 presents the screenshot of the graphical user interface of the developed software named MAGECAA v1.0.

3.1. Comparison of MAGECAA v1.0 with the Original Manual Method. As shown in Figure 4, Spearman's correlation analysis identified that there is a highly significant linear correlation between MAGEc and MAGEo ($r = 0.998$, resp.; $P < 0.01$), and the mean difference found in Bland-Altman plot was -0.03 ± 0.21 , which was statistically significantly small. This result indicated that the developed program was robust compared with the manual method.

TABLE 1: The correlation coefficient and difference of the results separately calculated using MAGEc, Fritzsche, and EasyGV software.

	MAGEc		Fritzsche	
	r	mean \pm SD (mmoles/L ⁻¹)	r	mean \pm SD (mmoles/L ⁻¹)
MAGEc	—	—	—	—
Fritzsche	0.987	0.14 \pm 0.55	—	—
EasyGV	0.926	0.52 \pm 1.17	0.926	0.38 \pm 1.12

Note. $P < 0.01$.

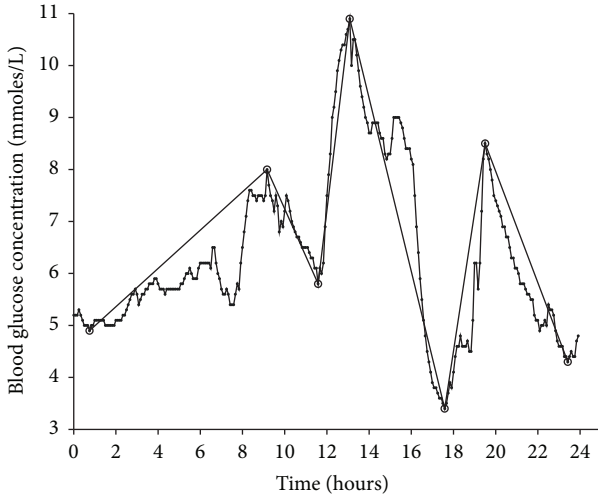


FIGURE 3: Screenshot of the graphical user interface of the MAGECAA v1.0 based on the MATLAB⁵ programming environment for the mean amplitude of glycemic excursions (MAGE) calculation. After computation, 24-hour continuous glucose monitoring profiles are shown in the plots with all valid extreme points joined by straight lines. Besides the calculation of MAGE, it also calculates the standard deviation of blood glucose (SDBG), the average of all upward valid excursions (MAGE₊), the average of all downward valid excursions (MAGE₋), and the average of all valid excursions (MAGEa). The CGM data were collected from patients with type 1 diabetes. After calculation, the results are as follows: SDBG = 1.63 mmoles/L, MAGE₊ = 4.43 mmoles/L, MAGE₋ = 4.63 mmoles/L, MAGEc = 4.43 mmoles/L (the first account excursion is from nadir to peak), and MAGEa = 4.53 mmoles/L.

3.2. Agreement of MAGECAA v1.0 with Fritzsche and EasyGV. To evaluate the agreement of MAGECAA v1.0 with two currently available popular software products, that is, Fritzsche and EasyGV, we did pairwise comparison of MAGE among them. Table 1 shows the correlation coefficients for MAGE calculation between the calculators, which ranged from 0.926 to 0.987 ($P < 0.01$ for all). Figure 5 shows the Bland-Altman plot among the three software products; the dashed lines represent the 95% confidence limit of the differences between the two methods, and the solid line demonstrates that the mean difference between the methods is close to 0, indicating that little difference existed among them. The mean differences are 0.14 ± 0.55 (MAGEc versus Fritzsche), 0.52 ± 1.17 (MAGEc versus EasyGV), and 0.38 ± 1.12 (Fritzsche versus EasyGV), showing the good agreement between these three software products.

3.3. Comparison of MAGEc with MAGEa. In the original definition of MAGE, its direction of the calculation is determined by the first countable excursion. So MAGEc = MAGE₊ or MAGEc = MAGE₋; it is somewhat arbitrary and ignores half of the valid excursions. However, MAGEa represents the average of MAGE₊ and MAGE₋; it does not consider the direction, thus involving all the valid excursions.

To explore whether MAGEa may be a more useful index than MAGEc which depends on MAGE₊ or MAGE₋, the relationship between MAGEc and MAGEa was tested by using all the 248 CGM measurements via Spearman's correlation analysis and Bland-Altman analysis. As shown in Figure 6, the correlation coefficient was $r = 0.993$ ($P < 0.01$) and the mean difference was -0.05 ± 0.48 . The results showed that little difference was observed between using MAGEa and MAGEc, indicating that MAGEa may be a useful index for evaluating glycemic variability.

4. Discussions

We developed a computer-aided open-code program named MAGECAA v1.0 based on INLP algorithm for automatic calculation of MAGE. Compared with the existing methods, the proposed novel method turns to search the optimal solution of the combination of extreme points from overall CGM measurements instead of searching adjacent extreme points from the beginning to the end step by step. As for the computational time, if used for one person, the proposed method is comparable with currently available methods; if used for batch calculation, the proposed method is more powerful. The programmable open codes are useful for the study of methodology for automatic calculation of MAGE. The comparison study using MAGECAA v1.0 against manual method indicated that the agreement is satisfied. The pairwise comparison study between MAGECAA v1.0 and two other available software products, Fritzsche and EasyGV, based on Spearman's correlation analysis and Bland-Altman plots, demonstrated that the agreements between them met the requirement. Our study showed that the worry about the disagreement among the currently available popular software products is not necessary, which is different from the proposal by Sechterberger et al. [26]. The reason may be the increased amount of CGM data used in our research.

The MAGE value depends on MAGE₊ or MAGE₋, following the direction of the first accountable glucose excursion. Considering the fact that only one direction of glucose excursion is adopted in current popular software, unavoidably resulting in the omission of the other directions of valid

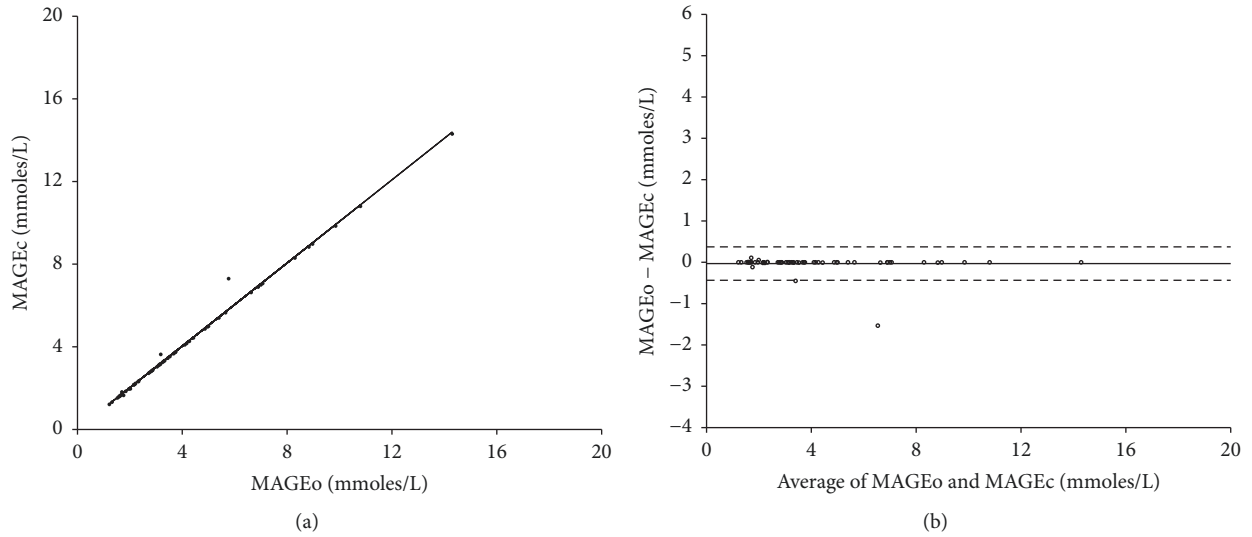


FIGURE 4: (a) Correlation of the mean amplitude of glycemic excursions values (MAGE) obtained from the proposed program (MAGEc) and the original manual methods (MAGEo). The data of 60 continuous glucose monitoring (CGM) measurements were randomly chosen from all collected CGM data. (b) Bland-Altman plot shows the difference between MAGEc and MAGEo on y -axis and the mean of the two computed indices on x -axis. The dashed lines represent the 95% confidence limit of the differences between the two methods, and the solid line indicates that the mean difference between the methods is close to 0.

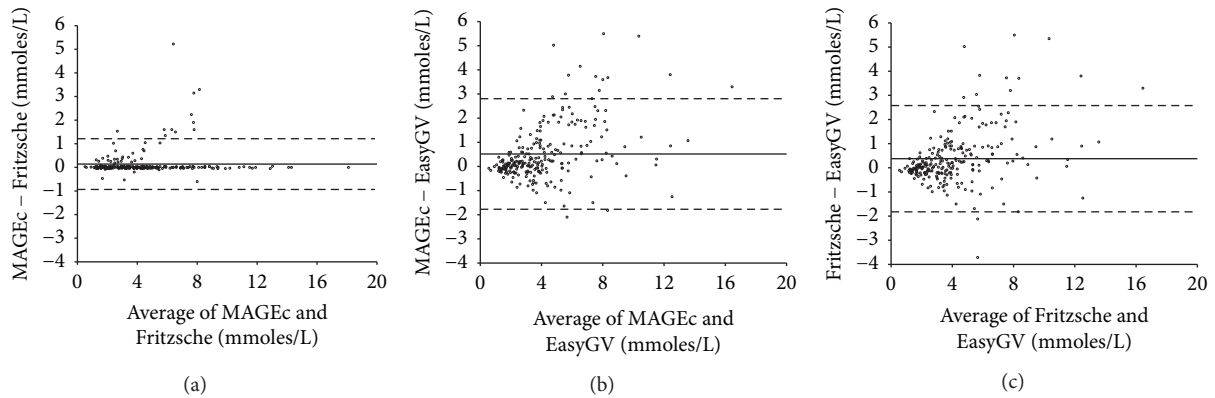


FIGURE 5: Bland-Altman plots showing the mean difference between MAGEc, EasyGV, and Fritzsche when applied to the same continuous glucose monitoring datasets.

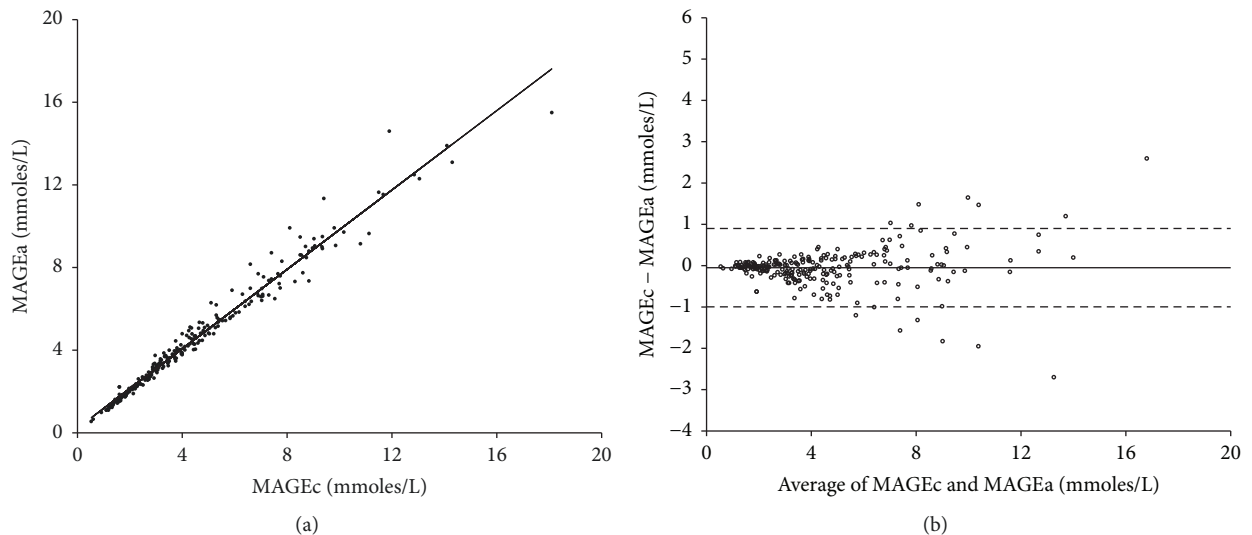


FIGURE 6: (a) Correlation between the mean amplitudes of glycemic excursions computerized (MAGEc) and the average of all valid glycemic excursions (MAGEa) for the total collected 248 continuous glucose monitoring measurements. (b) Bland-Altman plot shows the mean difference between MAGEc and MAGEa.

excursions, we implemented an extra experiment in which data from both directions of the valid glycemic excursions are utilized. MAGE_a was used to represent the mean of MAGE₊ and MAGE₋ for the calculation of MAGE. As shown by our data, a close linear correlation between MAGE_a and MAGE_c was observed, indicating that the difference between MAGE₊ and MAGE₋ is significantly small. Therefore, we proposed that MAGE_a might be another suitable parameter to quantify glycemic variability.

To conclude, an open-code software program named MAGECAA v1.0 for automatic calculation of MAGE based on a mathematical algorithm has been proposed and evaluated. The programmable open codes are useful for further methodology study in the future. The comparison study indicated that the agreement among the proposed software and existing software is satisfied, and the worry about the disagreement among currently available different popular software products is not necessary.

Abbreviations

MAGE:	Mean amplitude of glycemic excursions
SDBG:	Standard deviation of blood glucose
CGM:	Continuous glucose monitoring
Fritzsche:	Fritzsche et al.'s software
INLP:	Integer nonlinear programming
MAGE _o :	Calculated values obtained from the original manual method
MAGE _c :	Calculated values obtained from MAGECAA v1.0
MAGE ₊ :	The average of all upward valid glycemic excursions
MAGE ₋ :	The average of all downward valid glycemic excursions
MAGE _a :	The average of all downward validated glycemic excursions
DE:	Differential evolution.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was partially supported by the National Key Research and Development Program of China (nos. 2016YFC0100800 and 2016YFC0100801), the National Natural Science Foundation of China (Grants nos. 61671229 and 61528102), Science and Technology Program of Guangdong, China (nos. 2015B020214006 and 2016A050502026), Guangdong Natural Science Foundation (no. 2015A030313234), and Science and Technology Program of Guangzhou, China (no. 201704020091).

References

- [1] L. Monnier, E. Mas, C. Ginet et al., "Activation of oxidative stress by acute glucose fluctuations compared with sustained chronic

hyperglycemia in patients with type 2 diabetes," *The Journal of the American Medical Association*, vol. 295, no. 14, pp. 1681–1687, 2006.

- [2] I. B. Hirsch and M. Brownlee, "The effect of glucose variability on the risk of microvascular complications in type 1 diabetes: Response to Kilpatrick et al. and Bolli [9]," *Diabetes Care*, vol. 30, no. 1, pp. 186–187, 2007.
- [3] J. Bragd et al., "Can glycemic variability, as calculated from blood glucose self-monitoring, predict the development of complications in type 1 diabetes over a decade?" *Diabetes & Metabolism*, vol. 34, no. 6, pp. 612–616, 2008.
- [4] J. Škrha et al., "Glucose variability, HbA1c and microvascular complications," *Reviews in Endocrine & Metabolic Disorders*, vol. 17, no. 1, pp. 1–8, 2016.
- [5] M. Prázný et al., "Short-term study and long-term glycemic variability and its relationship to microvascular complications of diabetes," *Vnitřní Lekarství*, vol. 62, no. Supply 4, 85 pages, 2016.
- [6] M. Caprnda, "Glycemic variability and Vascular Complication in Patients with Type 2 Diabetes Mellitus," *Folia Med*, vol. 59, no. 3, pp. 270–278, 2017.
- [7] S. Eslami, Z. Taherzadeh, M. J. Schultz, and A. Abu-Hanna, "Glucose variability measures and their effect on mortality: a systematic review," *Intensive Care Medicine*, vol. 37, no. 4, pp. 583–593, 2011.
- [8] J. Hermanides, T. M. Vriesendorp, R. J. Bosman, D. F. Zandstra, J. B. Hoekstra, and J. H. Devries, "Glucose variability is associated with intensive care unit mortality," *Critical Care Medicine*, vol. 38, no. 3, pp. 838–842, 2010.
- [9] J. Schlichtkrull, O. Munck, and M. Jersild, "The M-Value, an Index of Blood-sugar Control in Diabetics," *Journal of Internal Medicine*, vol. 177, no. 1, pp. 95–102, 1965.
- [10] J. M. Wojcicki, "Mathematical descriptions of the glucose control in diabetes therapy. Analysis of the Schlichtkrull 'M'-value," *Hormone and Metabolic Research*, vol. 27, no. 1, pp. 1–5, 1995.
- [11] J. Wójcicki, "'M'-Index. A New Proposition of the Assessment of Current Glucose Control in Diabetic Patients," *Hormone and Metabolic Research*, vol. 27, no. 01, pp. 41–42, 1995.
- [12] B. P. Kovatchev, D. J. Cox, L. A. Gonder-Frederick, and W. Clarke, "Symmetrization of the blood glucose measurement scale and its applications," *Diabetes Care*, vol. 20, no. 11, pp. 1655–1658, 1997.
- [13] N. R. Hill, P. C. Hindmarsh, R. J. Stevens, I. M. Stratton, J. C. Levy, and D. R. Matthews, "A method for assessing quality of control from glucose profiles," *Diabetic Medicine*, vol. 24, no. 7, pp. 753–758, 2007.
- [14] B. P. Kovatchev, E. Otto, D. Cox, L. Gonder-Frederick, and W. Clarke, "Evaluation of a new measure of blood glucose variability in diabetes," *Diabetes Care*, vol. 29, no. 11, pp. 2433–2438, 2006.
- [15] D. Rodbard, T. Bailey, L. Jovanovic, H. Zisser, R. Kaplan, and S. K. Garg, "Improved quality of glycemic control and reduced glycemic variability with use of continuous glucose monitoring," *Diabetes Technology & Therapeutics*, vol. 11, no. 11, pp. 717–723, 2009.
- [16] I. B. Hirsch, A. K. Balo, K. Sayer, A. Garcia, B. A. Buckingham, and T. A. Peyser, "A simple composite metric for the assessment of glycemic status from continuous glucose monitoring data: Implications for clinical practice and the artificial pancreas," *Diabetes Technology & Therapeutics*, vol. 19, pp. S38–S48, 2017.

- [17] L. Monnier, C. Colette, and D. R. Owens, "Glycemic variability: The third component of the dysglycemia in diabetes. Is it important? how to measure it?" *Journal of Diabetes Science and Technology*, vol. 2, no. 6, pp. 1094–1100, 2008.
- [18] D. Rodbard, "New and improved methods to characterize glycemic variability using continuous glucose monitoring," *Diabetes Technology & Therapeutics*, vol. 11, no. 9, pp. 551–565, 2009.
- [19] F. J. Service, G. D. Molnar, J. W. Rosevear, E. Ackerman, L. C. Gatewood, and W. F. Taylor, "Mean amplitude of glycemic excursions, a measure of diabetic instability," *Diabetes*, vol. 19, no. 9, pp. 644–655, 1970.
- [20] F. J. Service and R. L. Nelson, "Characteristics of glycemic stability," *Diabetes Care*, vol. 3, no. 1, pp. 58–62, 1980.
- [21] F. J. Service, P. C. O'Brien, and R. A. Rizza, "Measurements of glucose control," *Diabetes Care*, vol. 10, no. 2, pp. 225–237, 1987.
- [22] P. A. Baghurst, "Calculating the mean amplitude of glycemic excursion from continuous glucose monitoring data: An automated algorithm," *Diabetes Technology & Therapeutics*, vol. 13, no. 3, pp. 296–302, 2011.
- [23] G. Fritzsche, K.-D. Kohnert, P. Heinke, L. Vogt, and E. Salzsieder, "The use of a computer program to calculate the mean amplitude of glycemic excursions," *Diabetes Technology & Therapeutics*, vol. 13, no. 3, pp. 319–325, 2011.
- [24] D. Czerwoniuk, W. Fendler, L. Walenciak, and W. Mlynarski, "GlyCulator: A glycemic variability calculation tool for continuous glucose monitoring data," *Journal of Diabetes Science and Technology*, vol. 5, no. 2, pp. 447–451, 2011.
- [25] N. R. Hill et al., "Normal Reference Range for Mean Tissue Glucose and Glycemic Variability Derived from Continuous Glucose Monitoring for Subjects Without Diabetes in Different Ethnic Groups," *Diabetes Technology & Therapeutics*, vol. 13, no. 9, pp. 921–928, 2011.
- [26] M. K. Sechterberger, Y. M. Luijck, and J. H. Devries, "Poor agreement of computerized calculators for mean amplitude of glycemic excursions," *Diabetes Technology & Therapeutics*, vol. 16, no. 2, pp. 72–75, 2014.
- [27] M. Anjidani and S. Effati, "Steepest descent method for solving zero-one nonlinear programming problems," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 197–202, 2007.
- [28] M. Kılınç et al., "Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON," *Optimization Methods & Software*, pp. 1–23, 2017.
- [29] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [30] S. Kukkonen and C. A. Coello Coello, "Generalized differential evolution for numerical and evolutionary optimization," *Studies in Computational Intelligence*, vol. 663, pp. 253–279, 2017.
- [31] P. Rakshit, "Uncertainty Management in Differential Evolution Induced Multiobjective Optimization in Presence of Measurement Noise," *IEEE Transactions on Systems Man Cybernetics Systems*, vol. 44, no. 7, pp. 922–937, 2017.
- [32] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 991–998, June 2005.
- [33] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 2, pp. 1785–1791, IEEE, September 2005.
- [34] Y. C. Lin, "A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems," *Computer Mathematics with Applications*, vol. 47, no. 8–9, pp. 1295–1307, 2004.
- [35] J. M. Bland and D. G. Altman, "Measuring agreement in method comparison studies," *Statistical Methods in Medical Research*, vol. 8, no. 2, pp. 135–160, 1999.
- [36] D. Giavarina, "Understanding Bland Altman analysis," *Biochemia Medica*, vol. 25, no. 2, pp. 141–151, 2015.



Hindawi

Submit your manuscripts at www.hindawi.com

