Research on the Dragon Dance Formation Based on Spiral Motion Model and Collision Detection

Jiazong Han*, #, Yinuo Su#, Ziheng Ding#

School of Mathematics, Shandong University, Jinan, China, 250100

*Corresponding author: 202200091033@mail.sdu.edu.cn

#These authors contributed equally.

Abstract. This study presents a mathematical model of the motion of a dragon dance team moving along an Archimedean spiral path. The motion of 224 points representing the dragon dance team is described using parametric equations, with the velocity of each point determined through the Runge-Kutta method. The study investigates the dynamics of the dragon head and body segments as they spiral inward, with a focus on maintaining fixed distances between consecutive points. Additionally, collision detection between the dragon's benches is implemented to prevent overlap. The termination time of the motion is determined by identifying the first collision between the dragon head and the body segment. The results demonstrate the application of numerical methods to model complex dynamic systems and ensure smooth movement without collision. The first collision occurs at 432.18 seconds, during the dragon's thirteenth loop. This study provides a valuable theoretical and practical basis for the motion analysis of complex dynamic systems by accurately simulating the movement of a dragon dance team along an Archimedean spiral and proposing a novel collision detection method.

Keywords: Dragon dance, Spiral motion model, Collision detection, Runge-Kutta method.

1. Introduction

The dragon dance is a traditional Chinese performance where a team of dancers manipulates a dragon-shaped figure along a predetermined path[1, 2]. In recent years, there has been growing interest in the mathematical modeling of such performances to enhance precision and coordination. The challenge of modeling the motion of the dragon dance team lies in its dynamic and intricate movement, often requiring complex algorithms to ensure smooth trajectories and avoid collisions among the dancers[1, 3]. Existing studies primarily focus on basic kinematic modeling and trajectory optimization but often overlook the intricate spatial constraints between the dancers, such as maintaining fixed distances between consecutive points. Additionally, while some approaches incorporate collision avoidance, they generally rely on simplified models that do not fully capture the dynamic nature of the motion, resulting in less accurate simulations and potentially unrealistic collision detection.

This study proposes a novel approach to modeling the formation of a dragon dance team as they move along an Archimedean spiral. The mathematical model accounts for both the movement of individual points along the spiral path and the spatial constraints between consecutive points. Furthermore, we employ collision detection techniques to prevent overlapping between dancers as they move, which is crucial for maintaining the aesthetic quality and safety of the performance. Unlike existing methods, this study introduces a more detailed model that incorporates velocity relationships and more precise spatial constraints, ensuring smoother motion and more accurate collision predictions.

2. Analysis of the Spiral Motion Model of a Dragon Dance Team

The data in this paper originates from http://www.mcm.edu.cn. To better describe the motion process of the dragon dance team on the spiral, this study first consider 224 points on an Archimedean spiral[3, 4]. Point A starts at the initial position and moves toward the spiral's center at a speed of 1 m/s. The straight-line distance between the first point and the second point is fixed at 2.86 meters, while

the distance between all other consecutive points remains constant at 1.65 meters. All points always remain on the spiral. The goal is to determine the position and velocity of each point from the initial moment to 300 seconds.

This study first defines the parametric equation of the curve in terms of the position vector $\vec{r}(\alpha)$as follows:

$$\vec{r}(\alpha) = \frac{p}{2\pi}\alpha(\cos\alpha\hat{\imath} + \sin\alpha\hat{\jmath}) \tag{1}$$

where α is the spiral parameter and $\hat{\alpha}\{i\}$ represents the spiral parameter at time t for the i-th point.

To construct the arc length equation, this study first specify the arc length s(α) of the Archimedean spiral, as shown below:

$$s(\alpha) = \frac{p}{4\pi}\left(\alpha\sqrt{1+\alpha^2} + \ln\left(\alpha + \sqrt{1+\alpha^2}\right)\right) \tag{2}$$

where α is the spiral parameter and p is the pitch of the spiral.

For the motion equations of the points, the i-th point (1≤i≤N) satisfies the following integral equation related to the arc length s(α) based on physical principles:

$$s(\alpha_0) - s\left(\alpha_i(t)\right) = \int_{(i-1)\Delta t}^{t} |\vec{v_i}(\alpha)| d\alpha, t \geq (i-1)\Delta t \tag{3}$$

where $|\vec{v_i}(\alpha)|$ is the magnitude of the velocity vector for the i-th point at time t, Δt represents the time interval for consecutive points entering the spiral (1.65 seconds), and the right-hand side indicates the integral of the absolute value of the velocity from the time the (i−1)-th point entered to the current moment.

The distance constraints between points are constructed as follows. For the leading points A and B, the distance between them must always equal the fixed distance d_1, represented by the following equation:

$$\left|\vec{r}\left(\alpha_1(t)\right) - \vec{r}\left(\alpha_2(t)\right)\right| = d_1, \quad t \geq \Delta t \tag{4}$$

where d_1 is the fixed distance between the handles of the dragon's head. Similarly, for any two consecutive points i and i+1(2≤i<N), the following relationship holds:

$$\left|\vec{r}\left(\alpha_i(t)\right) - \vec{r}\left(\alpha_{i+1}(t)\right)\right| = d_2, \quad t \geq i\Delta t \tag{5}$$

where d_2 is the fixed distance between all other consecutive points, set to 1.65 meters.

For the velocity relationships, the leading points A and B satisfy the condition that the difference in their position vectors and the difference in their velocity vectors are perpendicular to each other, as illustrated in figure 1:

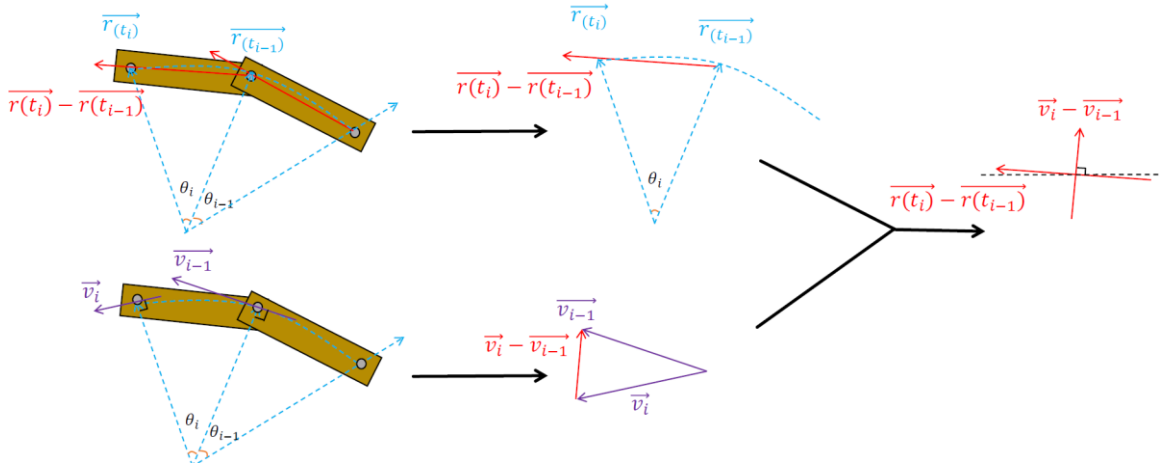

Figure 1 Speed and position vector relationship diagram

$$\left(\vec{r}(\alpha_1) - \vec{r}(\alpha_2)\right) \cdot (\vec{v_1} - \vec{v_2}) = 0, \quad t \geq \Delta t \tag{6}$$

Similarly, for any two consecutive points iii and i+1(2≤i<N), the following condition holds:

$$\left(\vec{r}(\alpha_i) - \vec{r}(\alpha_{i+1})\right) \cdot (\overrightarrow{v_i} - \overrightarrow{v_{i+1}}) = 0, \quad t \geq i\Delta t \tag{7}$$

In summary, the following system of equations can be derived for each discrete time step dt:
For points A and B (i=2):

$$\begin{cases} s(\alpha_0) - s(\alpha_2(t)) = \int_{\Delta t}^{t} |\vec{v}_2(\alpha)| d\alpha \\ |\vec{r}(\alpha_1(t)) - \vec{r}(\alpha_2(t)| = d_1 \\ (\vec{r}(\alpha_1) - \vec{r}(\alpha_2)) \cdot (\vec{v}_1 - \vec{v}_2) = 0 \end{cases} \tag{8}$$

For other points (3≤i≤224):

$$\begin{cases} s(\alpha_0) - s(\alpha_i(t)) = \int_{(i-1)\Delta t}^{t} |\vec{v}_i(\alpha)| d\alpha \\ |\vec{r}(\alpha_i(t)) - \vec{r}(\alpha_{i-1}(t))| = d_2 \\ (\vec{r}(\alpha_i) - \vec{r}(\alpha_{i-1})) \cdot (\vec{v}_i - \vec{v}_{i-1}) = 0 \end{cases} \tag{9}$$

To solve this complex system of equations, the Runge-Kutta method was adopted, a highly efficient numerical technique widely used for solving ordinary differential equations, whether linear or nonlinear[5, 6]. The core idea of this method is to estimate the change in the solution over each discrete time step through multiple function evaluations, thereby progressively approximating the true solution of the equations[7, 8]. Among its many variants, the fourth-order Runge-Kutta method (RK4) was chosen due to its balance of computational efficiency and accuracy, making it particularly suitable for long-term tracking of complex dynamic systems[9, 10].

The initial values of the independent variable t and the dependent variable $\vec{r}$(t) were set as t0=0 and $\vec{r}(\alpha_0)$=(8.8.0). With a time step h=1 second, k1, k2, k3, and k4 were iteratively calculated to update the approximate value of $\vec{r}$(α). Specifically:

$$\begin{aligned} k_1 &= hf\left(t_0, \overrightarrow{r(\alpha_0)}\right) \\ k_2 &= hf\left(t_0 + \frac{h}{2}, r(\alpha_0) + \frac{k_1}{2}\right) \\ k_3 &= hf\left(t_0 + \frac{h}{2}, r(\alpha_0) + \frac{k_2}{2}\right) \\ k_4 &= hf(t_0 + h, r(\alpha_0) + k_3) \\ r_1 &= r(\alpha_0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{10}$$

Repeating this iterative process, the value of $\vec{r}$(t) at each time step is obtained.

Table 1 Speed results at each point

Time (s)	Dragon Head (m/s)	1st Segment of Dragon Body (m/s)	51st Segment of Dragon Body (m/s)	101st Segment of Dragon Body (m/s)	151st Segment of Dragon Body (m/s)	201st Segment of Dragon Body (m/s)	Dragon Tail (m/s)
0	1	1.004248	1.003341	1.002681	1.002181	1.001788	1.00164
60	1	1.000037	0.999721	0.999502	0.999342	0.999219	0.999174
120	1	1.000052	0.999618	0.999334	0.999134	0.998986	0.998932
180	1	1.000079	0.999445	0.999063	0.998807	0.998624	0.998558
240	1	1.000132	0.999121	0.998577	0.998238	0.998006	0.997925
300	1	0.987264	0.990235	0.991631	0.992443	0.992974	0.993154

Table 2 Results of each point location

Point	0 s	60 s	120 s	180 s	240 s	300 s
Dragon Head x (m)	8.8	5.799209	-4.08489	-2.96361	2.594494	4.420274
Dragon Head y (m)	0	-5.77109	-6.30448	6.09478	-5.35674	2.320429
1st Segment of Dragon Body x (m)	8.621967	6.968526	-2.42036	-4.5223	4.136928	3.29038
1st Segment of Dragon Body y (m)	1.85146	-4.32462	-7.1345	5.079843	-4.31727	3.797907
51st Segment of Dragon Body x (m)	-9.31681	-8.34003	-4.73483	3.801309	5.360683	-6.12866
51st Segment of Dragon Body y (m)	2.329246	3.488743	6.986336	6.802887	-4.63647	1.477716
101st Segment of Dragon Body x (m)	1.931766	4.830292	4.501171	0.896346	-5.67563	-5.62135
101st Segment of Dragon Body y (m)	-10.1467	-8.53562	-8.08892	-8.62511	-5.70019	4.756265
151st Segment of Dragon Body x (m)	10.97548	7.425166	3.354965	2.004683	3.900548	7.519336
151st Segment of Dragon Body y (m)	0.826846	7.450784	9.428455	9.254184	7.99726	3.485755
201st Segment of Dragon Body x (m)	5.461938	-5.77513	-10.4435	-9.65493	-8.05724	-7.99359
201st Segment of Dragon Body y (m)	10.28435	9.578722	2.354333	-3.30164	-5.35846	-4.38837
Dragon Tail x (m)	-6.18499	6.558019	10.99739	8.064617	4.182751	2.777887
Dragon Tail y (m)	-10.1836	-9.40516	-0.16754	6.741063	9.082899	9.04443

Using the Runge-Kutta method, the position and velocity of the entire dragon dance team are determined at each second from the initial moment to 300 seconds. Table 1 presents the position coordinates of selected points, with the dragon head starting at A(8.8,0), while Table 2 shows the magnitude of the velocity vector for selected points, where the dragon head maintains a constant velocity of 1 m/s.

3. Collision Detection on the Dragon Dance Team's Spiral Path

Furthermore, as the dragon dance team spirals inward along the defined spiral path, it is necessary to determine the termination time of the spiraling motion to avoid collisions between benches. Specifically, a collision is deemed to occur if the rectangular areas formed by the four corner points of the dragon head's bench and the four corner points of any other bench overlap.

Given the position vector of the bench handles on the spiral, as well as the distance of 27.5 cm from the handle to the nearest bench edge, and a bench width of 30 cm, it is required to represent the coordinates of the four corner points of the dragon head's bench. To do so, auxiliary vectors need to be defined, including the unit vector in the longitudinal direction of the dragon head:

$$\vec{u}_1 = \frac{\vec{r}(\alpha_1(t)) - \vec{r}(\alpha_2(t))}{\left|\vec{r}(\alpha_1(t)) - \vec{r}(\alpha_2(t))\right|} = (u_1^x, u_1^y) \tag{11}$$

Unit vector of dragon head width:

$$\vec{v}_1 = (-u_1^y, u_1^x) \tag{12}$$

Similarly, to represent the coordinates of the four corner points of the i-th bench, it is necessary to define the unit vector in the length direction of the bench:

$$\vec{u_i} = \frac{\vec{r}(\alpha_i(t)) - \vec{r}(\alpha_{i+1}(t))}{\left|\vec{r}(\alpha_i(t)) - \vec{r}(\alpha_{i+1}(t))\right|} = (u_i^x, u_i^y) \tag{13}$$

Unit vector in the board width direction:

$$\vec{v_i} = (-u_i^y, u_i^x) \tag{14}$$

This shows that the two corner points of faucet A are:

$$\begin{cases} (x_1, y_1) = \vec{r}(\alpha_1(t)) + 27.5\vec{u_1} + 15\vec{v_1} \\ (x_2, y_2) = \vec{r}(\alpha_2(t)) - 27.5\vec{u_1} - 15\vec{v_1} \end{cases} \tag{15}$$

The two corner points of the i-th bench B are:

$$\begin{cases} (x_3, y_3) = \vec{r}(\alpha_i(t)) + 27.5\vec{u_i} + 15\vec{v_i} \\ (x_4, y_4)\,\vec{r}(\alpha_{i+1}(t)) - 27.5\vec{u_i} - 15\vec{v_i} \end{cases} \tag{16}$$

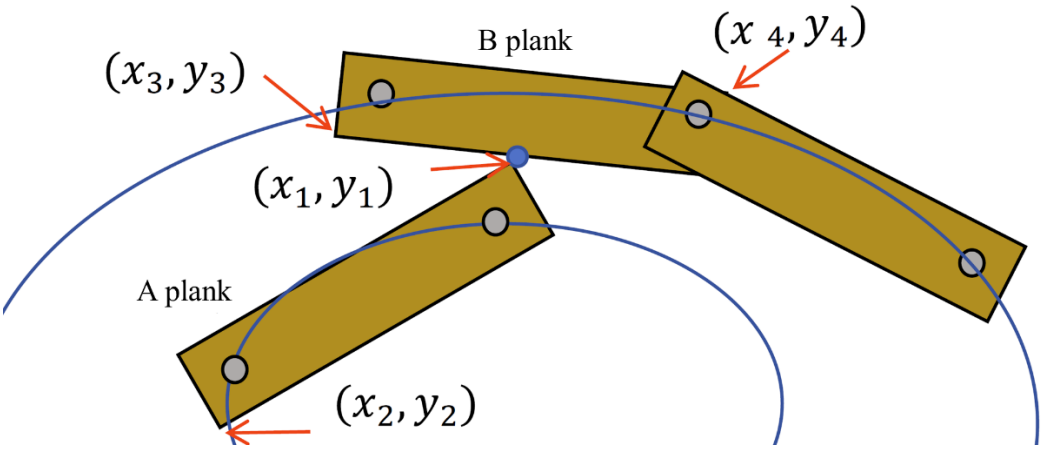

Figure 2 A B plank board collision diagram

This paper considers two rectangles, A and B, each defined by the coordinates of their bottom-left and top-right corners. As shown in Figure 2. Rectangle A has its bottom-left corner at (x1, y1) and its top-right corner at (x2, y2), while Rectangle B has its bottom-left corner at (x3, y3) and its top-right corner at (x4, y4). To determine if these two rectangles overlap, this paper first examines the scenario in which they do not overlap. There are four possible situations where the rectangles do not overlap: when Rectangle A is entirely to the left of Rectangle B, which occurs if x2<x3; when Rectangle A is entirely to the right of Rectangle B, which occurs if x1>x4; when Rectangle A is entirely above Rectangle B, which occurs if y1>y4; and when Rectangle A is entirely below Rectangle B, which occurs if y2<y3. If any of these conditions are true, the two rectangles do not overlap. This logic can be expressed as a Boolean expression.

$$Do\ not\ overlap = (x_2 < x_3)||(x_1 > x_4)||(y_1 > y_4)||(y_2 < y_3) \tag{17}$$

Therefore, the condition for the two rectangles to overlap is the negation of the above expression.

$$\text{overlap} = !\,\text{do not overlap} = !((x_2 < x_3)||(x_1 > x_4)||(y_1 > y_4)||(y_2 < y_3)) \tag{18}$$

By applying De Morgan's law, this expression can be rewritten as:

$$\text{overlap} = (x_2 \geqslant x_3)\&\&(x_1 \leqslant x_4)\&\&(y_1 \leqslant y_4)\&\&(y_2 \geqslant y_3) \tag{19}$$

This is the final formula for determining whether two rectangles overlap. For multiple rectangles, each pair of rectangles must be compared. If any two rectangles overlap, then an overlap exists. Using a traversal algorithm, the program was run and the first collision time was found to be t = 432.18 seconds. At this moment, the head of the dragon collided with the sixth segment of the body while the dragon was progressing to the thirteenth loop (as shown in Figure 3). The positions and velocities of the other points in the team at this time are provided in Table 3.

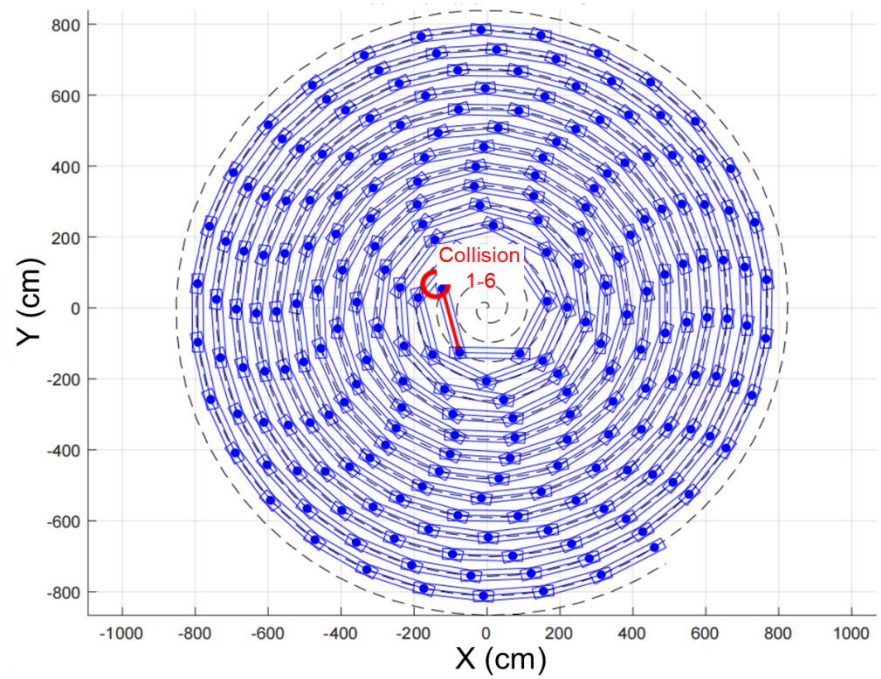

Figure 3 Diagram of the Positions of Points in the Dragon Dance Team at the Time of Collision

Table 3: Positions and Velocities of Selected Points in the Dragon Dance Team

Point	x-coordinate (m)	y-coordinate (m)	Speed (m/s)
Dragon Head	0.340443	1.737525	1
1st Segment of Dragon Body	-1.43665	1.188444	0.962836
51st Segment of Dragon Body	-0.35705	-4.24014	0.962779
101st Segment of Dragon Body	-4.24804	3.816914	0.96285
151st Segment of Dragon Body	-6.79664	0.950858	0.962881
201st Segment of Dragon Body	-0.82641	7.803097	0.962898
Dragon Tail (Rear)	8.211347	-0.71479	0.962903

Table 3 and figure 3 presents the positions and velocities of selected points in the dragon dance team at the time of the first collision. The dragon head, with a constant velocity of 1 m/s, is positioned at (0.340443, 1.737525). As move along the body, the velocity gradually decreases, with the 1st segment at 0.962836 m/s and the 151st segment at 0.962881 m/s, reflecting the smooth transition in speed due to the spiral path. The tail of the dragon maintains a similar velocity to the body segments, indicating synchronized movement. This consistency in velocity is crucial for ensuring fluid motion and preventing collisions during the performance.

4. Conclusion

This study presents a mathematical model for simulating the motion of a dragon dance team along an Archimedean spiral, with a focus on both motion dynamics and collision detection. Using the Runge-Kutta method, the positions and velocities of the team's formation were tracked over time,

considering the fixed distances between consecutive points to maintain the structure of the formation as it spirals inward.

In addition, collision detection was integrated into the model to prevent overlap between benches placed along the spiral path. By defining the positions of the benches and utilizing De Morgan's law, an efficient algorithm was developed to check for potential collisions between the dragon's formation and the benches. The analysis revealed the critical point in time when the first collision occurs between segments of the formation, demonstrating the effectiveness of the collision detection approach.

This research provides a detailed framework for modeling and simulating coordinated movement systems, with applications beyond the dragon dance, including areas such as robotics, animation, and traffic flow management. The methods introduced here can be applied to ensure smooth and safe operations in dynamic, multi-point systems, where collision avoidance is crucial.

References

[1] LIANG Y, KRUTASAEN W. Comparative Cultural Analysis and Modern Reflections on Chinese Dragon Dance[J]. Journal of Roi Kaensarn Academi, 2024,9(7): 504-516.

[2] ZHANG Q. Exploration and Analysis on the Development of Chinese Dragon Dance and the Training System of Dragon Dancing in Universities[J]. Journal of Humanities, Arts and Social Science, 2023,7(2): 461-464.

[3] LEI F, LIU Q, SHEN X, et al. Optimization of Bench Dancing Dragon Team Motion Based on Collision Detection Model and Numerical Simulation[J]. Advances in Computer, Signals and Systems, 2024,8(6): 112-121.

[4] de SOUSA A C, DECKERS E, CLAEYS C, et al. On the assembly of Archimedean spiral cavities for sound absorption applications: Design, optimization and experimental validation[J]. Mechanical Systems and Signal Processing, 2021,147: 107102.

[5] WU Y, CHEN Y, WEI Y. Stochastic runge-kutta methods: Provable acceleration of diffusion models[J]. arXiv preprint arXiv:2410.04760, 2024.

[6] FU Z, YANG J. Energy-decreasing exponential time differencing Runge–Kutta methods for phase-field models[J]. Journal of Computational Physics, 2022,454: 110943.

[7] ATTAULLAH Y, ALYOBI S, AL-DUAIS F S, et al. On the comparative performance of fourth order Runge–Kutta and the Galerkin–Petrov time discretization methods for solving nonlinear ordinary differential equations with application to some mathematical models in epidemiology[J]. AIMS Math, 2023,8(2): 3699-3729.

[8] SOUTHWORTH B S, KRZYSIK O, PAZNER W. Fast solution of fully implicit Runge--Kutta and discontinuous Galerkin in time for numerical PDEs, part II: Nonlinearities and DAEs[J]. SIAM Journal on Scientific Computing, 2022,44(2): A636-A663.

[9] AIDA M. Fourth-order runge-kutta method for solving applications of system of first-order ordinary differential equations[J]. Enhanced Knowledge in Sciences and Technology, 2022,2(1): 517-526.

[10] SIMANGUNSONG L, MUNGKASI S. Fourth order Runge-Kutta method for solving a mathematical model of the spread of HIV-AIDS[C]//. AIP Conference Proceedings: AIP Publishing, 2021.