Data Computing Model and Software Development for Construction Setting-out of Line Curve

Mingduan Zhou1,*, Bohong Ma1, Miao Wang2, Wenyao Zhang1 and Binbin Wang3

1School of Geometrics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China
2School of Civil and Transportation Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China
3Shanxi Railway Engineering Survey Co., Ltd, Xian, China

*Corresponding author email: mdzhou@bucea.edu.cn

Abstract. In view of the fact that the line design is often a cross combination of multi-curves, it is difficult to meet the middle-pile surveying data computing of multi-curves combination for the existing tool software. In this paper, a line pile-point surveying data computing software (LPLS) suitable for multi-curves combination is designed and developed, and the flowchart of software design and development is given. Through the computing and analysis of engineering examples, the main functional modules of LPLS software are verified, and the correct and reliable computing results are obtained.

1. Introduction
Circular curve is a common linear element in railway engineering. Circular curve should be designed for highway or railway regardless of the angle. When the traffic carrier enters the circular curve from the straight line, the driver should gradually change the steering angle and transition from the straight line to the circular curve [1]. The curvature of the traffic carrier driving on the curve may be constantly changing. In order to adapt to the change of curvature, a transition curve needs to be set. In addition, the profile center-line can be composed of many slope segments with different slopes. Whether it is circular curve, transition curve or vertical curve, the computing of point location measurement data for a single curve is relatively simple. With Visual C++ 6.0 as the development platform, the software of the transition curve coordinate for construction setting-out of line curve was designed and developed [2]. The pile-point coordinate of the transition curve was computing using the table characteristics of Excel [3]. The program for calculating the transition curve coordinate was designed and implemented using VB and MATLAB [4]. Construction setting-out data of the transition curve was calculated quickly using the student function calculator [5-6]. However, there are always the combinations of complex kinds of curves. It is difficult to meet the calculation of point location measurement data of many kinds of curve cross combinations for the existing software tools. In this paper, a data computing software for construction setting-out of line curve (that is LPLS software) based on VS2010 development platform and using C# programming language is designed and developed, and the flowchart of the designed and developed software is given, which is of great practical significance and practical value to improve the working efficiency of the data computing of construction setting-out of line curve. The paper is divided into four sections. The first section is the “Introduction”. “Construction setting-out data computing model” formulates the proposed model to solve the data computing of construction setting-out. “Software design and implementation” gives the flowchart of the LPLS software. “Numerical verification” includes the experiments and accuracy verification. The
last section “Conclusions” draws the conclusions.

2. Construction Setting-out Data Computing Model

2.1. Coordinates Computing Model of Circular Curve

The principle of coordinate computing of circular curve was shown in figure 1 [7-9]. Suppose M was one of the points on the circular curve, the coordinate computing steps of the model were as follows.

Figure 1. Circular curve construction setting-out principle

Setp1: the coordinate of the center O was calculated. Firstly, the coordinate azimuth from ZY to YZ could be calculated.

\[
\alpha_{ZY \rightarrow YZ} = \arctan \left( \frac{Y_{YZ} - Y_{ZY}}{X_{YZ} - X_{ZY}} \right)
\]

And then, the center angle corresponding to the inferior arc from ZY to YZ could be calculated.

\[
\theta = \left[ \frac{L_{ZY} - L_{YZ}}{R} \right] \times \left( \frac{180^\circ}{\pi} \right)
\]

By formula (2), angle \( \beta \) could be obtained.

\[
\beta = (180^\circ - \theta)/2
\]

According to formula (1) and (3), the coordinate azimuth from point ZY to point O could be obtained.

\[
\alpha_{ZY \rightarrow O} = \alpha_{ZY \rightarrow YZ} + \beta
\]

By equation (4), the coordinates of the center O cloud be calculated.

\[
X_O = X_{ZY} + R \cos \alpha_{ZY \rightarrow O}, \quad Y_O = Y_{ZY} + R \sin \alpha_{ZY \rightarrow O}
\]

Step 2: the coordinate azimuth from the center O to ZY was calculated.

\[
\alpha_{O \rightarrow ZY} = \alpha_{ZY \rightarrow O} - 180^\circ
\]

Step 3: the coordinate azimuth from the center O to M on the circular curve was calculated. Firstly, the arc length \( l_M \) from ZY to M was calculated.

\[
l_M = L_M - L_{ZY}
\]

Where, \( L_M \) was the mileage of point M, \( L_{ZY} \) was the mileage of ZY. And then, the center angle corresponding to the inferior arc from ZY to M was calculated.

\[
\theta_M = \left( l_M / R \right) \times \left( \frac{180^\circ}{\pi} \right)
\]

According to formula (6) and (8), the coordinate azimuth from O to M was obtained.

\[
\alpha_{O \rightarrow M} = \alpha_{O \rightarrow ZY} - \theta_M
\]
Step 4: the pile-point coordinate of $M$ was calculated. According to formula (9), the coordinates of $M$ was calculated.

$$X_M = X_O + R \cos \alpha_{O \rightarrow M}, \quad Y_M = Y_O + R \sin \alpha_{O \rightarrow M}$$

(10)

2.2. Coordinates Computing Model of Circular Curve with Transition Curve

The principle of coordinate computing of circular curve with transition curve in the independent coordinate system was shown in figure 2 [7-9]. If $K$ was one of the points on the transition curve, the coordinate calculation steps of the model in the construction coordination system were as follows.

**Figure 2.** Circular curve with transition curve construction setting-out principle

Step 1: the tangent angle $\beta_K$ of $K$ and the deflection $\delta_K$ of $K$ were calculated in respectively.

$$\beta_K = \left[ \frac{L_K}{(2RL_h)} \right] \times (180^\circ / \pi)$$

(11)

$$\delta_K = \frac{L_K}{3} = \left[ \frac{L_K^2}{(6RL_h)} \right] \times (180^\circ / \pi)$$

(12)

Where, $L_K$ was the arc length from $K$ to $ZH$, $L_h$ was the length of transition curve, $R$ was the radius of circular curve.

Step 2: the deflection from $ZH$ to $K$ was calculated.

$$\alpha_K = \alpha_{ZH \rightarrow JD} + \delta_K$$

(13)

Step 3: the chord length $C_k$ from $K$ to $ZH$ was calculated.

$$C_k = \sqrt{x_k^2 + y_k^2} = \sqrt{\left[ L_K - \frac{L_R}{(40R^2L_h)} \right]^2 + \left[ \frac{L_R}{(6RL_h)} \right]^2}$$

(14)

Step 4: According to formula (13) and (14), the coordinate of $K$ in the construction coordinate system was calculated.

$$X_K = X_{ZH} + C_k \cdot \cos \alpha_K, \quad Y_K = Y_{ZH} + C_k \cdot \sin \alpha_K$$

(15)

On the other hand, if $Q$ was an arbitrary point of the circular curve with transition curve in the independent coordination was shown in figure 2, the coordinate calculation steps of model were as follow.

Step 1: the tangent azimuth at $Q$ point was calculated.

$$\beta_Q = \beta_0 + \left[ \frac{(L_Q - L_h)}{R} \right]$$

(16)

Where, $\beta_0 = L_h/(2R)$ was the tangent azimuth of HY, $L_Q$ was the arc length from Q to ZH.

Step 2: the tangent increment $q$ of transition curve and inward shift value of circular curve $\Delta R$ was calculated.
\[ q = L_i / 2 - L_h^1 / (240R^2) \]  \hspace{1cm} (17) \\
\[ \Delta R = L_h^1 / (24R) - L_h^i / (2384R^3) \]  \hspace{1cm} (18)

Step 3: according to the equation (16) ~ (18), the coordinates of Q in independent system could be obtained.

\[ x_Q = R \sin \beta_Q + q, y_Q = R(1 - \cos \beta_Q) + \Delta R \]  \hspace{1cm} (19)

Step 4: the deflection angle from Q to ZH and the arc length from Q to ZH by equation (19) was calculated.

\[ \alpha_Q = \alpha_{ZH} + \delta_Q \]  \hspace{1cm} (20)

\[ C_Q = \sqrt{x_Q^2 + y_Q^2} \]  \hspace{1cm} (21)

Step 5: the coordinate of Q in the construction coordinate system by equations (20) and (21) was calculated.

\[ X_Q = X_{ZH} + C_Q \cos \alpha_Q, Y_Q = Y_{ZH} + C_Q \sin \alpha_Q \]  \hspace{1cm} (22)

2.3. Elevation Computing Model of Vertical Curve

The principle of elevation computing of vertical curve was shown in figure 3 [7-9]. Suppose \( V \) was one of the points on the vertical curve, the elevation calculation steps of the model were as follows.

**Figure 3.** Vertical curve construction setting-out principle

Step 1: the basic elements of vertical curve consist of \( \alpha \), \( T \), \( L \) and \( E \) was calculated as follows.

\[ \alpha = \Delta, i = i_1 - i_2, T = R \tan(\alpha/2) = (\alpha R)/2 \approx (R \Delta)/2, L = R \alpha = R |\Delta| \approx 2T, E = T^2/(2R) \]  \hspace{1cm} (23)

Step 2: the mileage of starting and ending points was calculated.

\[ M_O = M_V - T, M_D = M_O + L \]  \hspace{1cm} (24)

Where, \( M_O \) was the starting mileage of vertical curve, \( M_V \) was the mileage of grade change point, \( M_D \) was the end mileage of vertical curve.

Step 3: the tangent elevation of any point on the slope line was calculated.

\[ H_V = H_y + d_y \times i \]  \hspace{1cm} (25)

Where, \( d_y = M_V - M_O \), \( H_V \) was tangent elevation of any point on the slope line, \( H_y \) was the elevation of the slope change point.

Step 4: the elevation difference between each point of vertical curve and corresponding tangent line \( h \) was calculated.
\[ h = \frac{d^2}{(2R)} \]  

(26)

Step 5: the design elevation of any point \( V \) of vertical curve was calculated.

\[ H_{CV} = H_V^t + y, H_{RV} = H_V^t - y \]  

(27)

Where, \( H_{CV} \) was the design elevation of a station on a concave vertical curve, \( H_{RV} \) was the design elevation of a station on a convex vertical curve, \( H_V^t \) was the tangent elevation of this point.

3. Software Design and Implementation

According to the data computing model of construction setting-out, the data computing software for construction setting-out of line curve (short as LPLS) based on VS2010 development platform using C# programming language was designed and developed. The main interface design of LPLS software was shown in figure 4. According to the figure 4, the program models consisted of curve-element computing module, curve configuration module, coordinate computing module and elevation computing module were established respectively. The structure of LPLS program module was shown in figure 5.

![Figure 4. The main interface design of LPLS](image)

![Figure 5. The structure of LPLS program module](image)

![Figure 6. The flowchart of coordinate computing module](image)

![Figure 7. The flowchart of evaluation computing module](image)

From figure 5, the curve-element computing module consisted of curve length, tangent length and computing function realization of the difference of the tangent and curve. The curve configuration module consisted of route start point and end point, mileage coordinates, intersection coordinates and mileage, the radius of circular curve and the length of transition curve. The coordinate computing module realized the computing function of the main-point coordinate and the middle-pile coordinate of line curve based on the curve-element computing module and the curve configuration module. The elevation computing module realized the computing function of design-elevation with middle-pile of line curve. Among them, the core modules of LPLS software were the coordinate computing module and the elevation computing module. The realization flowchart of the coordinate computing module was shown in figure 6. The realization flowchart of the elevation computing module was shown in figure 7.
4. Numerical Verification
In order to analyze and verify the correctness and validity of the data computing model and the module of LPLS software, the middle-pile data of a certain section mileage of railway engineering was selected.

4.1. Coordinates Computing and Verification
According to the coordinate computing formulas and steps in section 2.1-2.2, the coordinate computing function module (LPLS_H) was established by using the C# programming language. Based on the LPLS software, the calculation results of the coordinates of the interval of each kilometer of DK6+813~ DK9+430 were obtained as shown in figure 8.

From figure 8, in order to verify the correctness of coordinates of pile-point, mileage points of transition curve was selected randomly, circular curve and other five curve segments for manual verification calculation, the results of manual verification calculation with the results of LPLS software calculation were compared. The statistical results of the difference were shown in table 1.

From the table 1, it could be seen that the coordinate results of the pile-point on the five curves intercepted by the LPLS software were consistent with the results of the manual verification calculation (ignoring the manual calculation error). It showed that the LPLS_H module was correct and effective.

| Curve segment   | Mileage/m | LPLS_X/m  | LPLS_Y/m  | Manual_X/m | Manual_Y/m | ΔX/m | ΔY/m |
|-----------------|-----------|-----------|-----------|------------|------------|------|------|
| Transitive curve part (1) | DK6+8 | 60 | 509 | 527 | 510 | 527 | 0.0001 | 0.0000 |
| Circular curve part | DK6+8 | 99 | 842 | 748 | 840 | 748 | 0.0002 | 0.0000 |
| Transitive curve part (2) | DK7+0 | 20 | 746 | 552 | 748 | 553 | 0.0000 | 0.0000 |
| Circular curve part (1) | DK7+0 | 90 | 348 | 183 | 346 | 183 | 0.0000 | 0.0000 |
| Transitive curve part (2) | DK7+1 | 21 | 844 | 115 | 847 | 114 | 0.0000 | 0.0000 |
| Circular curve part (1) | DK7+1 | 57 | 535 | 334 | 533 | 335 | 0.0000 | 0.0000 |
| Transitive curve part (2) | DK7+4 | 60 | 498 | 540 | 500 | 539 | 0.0000 | 0.0000 |
| Circular curve part (1) | DK7+4 | 60 | 734 | 141 | 733 | 141 | 0.0001 | 0.0002 |
| Transitive curve part (2) | DK7+6 | 10 | 930 | 924 | 931 | 926 | 0.0000 | 0.0000 |
| Transitive curve part (1) | DK7+8 | 40 | 675 | 075 | 675 | 075 | 0.0000 | 0.0000 |
| Circular curve part | DK7+8 | 60 | 376 | 063 | 378 | 064 | 0.0000 | 0.0000 |
| Transitive curve part (2) | DK8+0 | 90 | 716 | 905 | 717 | 904 | 0.0000 | 0.0001 |
| Circular curve part | DK8+4 | 10 | 785 | 048 | 788 | 048 | 0.0000 | 0.0000 |
| Transitive curve part (1) | DK8+5 | 20 | 266 | 587 | 265 | 584 | 0.0000 | 0.0000 |
| Circular curve part | DK8+5 | 60 | 839 | 838 | 839 | 841 | 0.0000 | 0.0000 |
| Transitive curve part (2) | DK8+5 | 0.0003 | 0.0003 |
| Min | -0.000 | -0.000 |
| Average | 0.0000 | 0.0000 |
4.2. Elevation Computing and Verification

According to the elevation computing formulas and steps in section 2.3, the elevation computing function module (LPLS_V) was established by using the C# programming language. Based on the LPLS software, the elevation results of DK503+500~DK504+436 were obtained as shown in figure 9. From figure 9, in order to verify the correctness of pile elevation results in LPLS software, the mileage DK503+750 of variable slope point 1, the elevation of 1405.530m, the radius of 50000m, the mileage DK504+450 of variable slope point 2, the elevation of 1408.260m, the radius of 50000m, the mileage DK505+980 of variable slope point 3, the elevation of 1408.72m and the radius of 80000m were intercepted for manual verification, and 7 mileage points were randomly selected. The statistical results of the difference between manual calculation and LPLS software calculation could be seen from table 2.

Table 2. The comparison results of difference between LPLS and manual calculation

| Order number | Mileage /m | LPLS elevation/m | Manual elevation/m | ΔH/m |
|--------------|------------|------------------|--------------------|------|
| 1            | DK503+526  | *6.314           | *6.314             | 0.000|
| 2            | DK503+578  | *6.132           | *6.132             | 0.000|
| 3            | DK503+760  | *5.569           | *5.569             | 0.000|
| 4            | DK503+994  | *6.482           | *6.482             | 0.000|
| 5            | DK504+358  | *7.901           | *7.901             | 0.000|
| 6            | DK504+370  | *7.948           | *7.948             | 0.000|
| 7            | DK504+436  | *8.205           | *8.205             | 0.000|

From the table 2, it could be seen that the calculation results of 7 mileage points arbitrarily selected on the vertical curve calculated by the LPLS software were consistent with the calculation results of manual verification. It showed that the LPLS_V module established was correct and effective.

5. Conclusions

In view of the fact that the line design is often a cross combination of multiple curves, it was difficult to meet the data computing for construction setting-out of multi-type curve combination for the existing tool. Based on VS2010 development platform, the data computing of construction setting-out software (LPLS) suitable for multi-type curve was designed and developed by using C# programming language. The data computing models of circular curve, circular curve with transition curve and vertical curve for construction setting-out were given in detail and the corresponding program modules (LPLS_H and LPLS_V) were established. Through the numerical verification of the railway engineering, the correctness of the modules of LPLS_H and LPLS_V were verified respectively, and it can be used to calculate the construction setting-out data of highway, railway and other line curve, which is of great significance and practical value to improve the efficiency of construction setting-out and profile design of line curve.
Acknowledgments
This work was financially supported by the China Construction Education Association education teaching research project (No.2019009) and General project of science and technology plan of Beijing Municipal Education Commission (No.SQKM201710016005).

References
[1] Minglin Zhong, Eryu Zhu. 2013. Development of emergency track beam alignment for rapid track beam replacement of straddle monorail transit, Journal of Transportation Engineering, Vol.139, No.4, pp.416-423.
[2] Haibiao Tong. 2012. Design of data calculation software for coordinate method of road transition curve. Shanxi Architecture. Vol.38, No.8, pp.284-285.
[3] Kai Ji. 2014. Calculation of pile coordinates in transition curve based on Excel. Anhui Architecture, No 5, pp.229-230
[4] Shuaishuai Zhang, Shengwen Yu, Mingcui Wang et al. 2014. Coordinate calculation of arbitrary transition curve based on VB and MATLAB. Journal of Shandong University of science and technology (Natural Science Edition). Vol.33, No.4, pp.76-80.
[5] Youbao Guo. 2016. A fast method of calculating circle curve with student function calculator. Technology vision. No.23, pp.156-156.
[6] Youbao Guo. 2017. A fast method of calculating circle curve with student function calculator. Heilongjiang transportation technology. No.10, pp.17-18.
[7] Xiuzhong Chen, Yukui Chan, Rongyao Jin. 2013. Engineering surveying. Tsinghua University Press, Beijing.
[8] Xiaolie Gu, Feng Bao, Xiaojun Cheng. 2011. Surveying (The Fourth Edition). Tongji University Press, Shanghai, China.
[9] Yongqi Chen. 2016. Engineering surveying. Surveying and Mapping Press, Beijing, China.