Cant Change Calculation Methods - Equivalence Verification

Rail Geometry Analysis

1 Problem Statement

Given interpolated rail coordinates at chainage ch_c for left and right rails, verify that two methods for calculating cant change produce identical results:

Method A: Calculate cant, then compute change in cant
Method B: Calculate coordinate deltas, then compute cant change directly

2 Input Dataset

Assume we have interpolated rail coordinates available:

$$\text{Left Rail:} \quad X(ch_c,0)_{\mathrm{L}}, Y(ch_c,0)_{\mathrm{L}}, Z(ch_c,0)_{\mathrm{L}} \quad \text{(baseline)} \tag{1}$$

$$X(ch_c,m)_{\mathrm{L}}, Y(ch_c,m)_{\mathrm{L}}, Z(ch_c,m)_{\mathrm{L}} \quad \text{(current)} \tag{2}$$

$$\text{Right Rail:} \quad X(ch_c,0)_{\mathrm{R}}, Y(ch_c,0)_{\mathrm{R}}, Z(ch_c,0)_{\mathrm{R}} \quad \text{(baseline)} \tag{3}$$

$$X(ch_c,m)_{\mathrm{R}}, Y(ch_c,m)_{\mathrm{R}}, Z(ch_c,m)_{\mathrm{R}} \quad \text{(current)} \tag{4}$$

3 Method A: Cant-First Approach

3.1 Step 1: Calculate Baseline Cant

$$\mathrm{Cant}(ch_c,0) = Z(ch_c,0)_{\mathrm{L}} - Z(ch_c,0)_{\mathrm{R}} \tag{5}$$

3.2 Step 2: Calculate Current Cant

$$\mathrm{Cant}(ch_c,m) = Z(ch_c,m)_{\mathrm{L}} - Z(ch_c,m)_{\mathrm{R}} \tag{6}$$

3.3 Step 3: Calculate Cant Change

$$\Delta\mathrm{Cant}(ch_c,m)_A = \mathrm{Cant}(ch_c,m) - \mathrm{Cant}(ch_c,0) \tag{7}$$

Substituting the cant definitions:

$$\Delta\mathrm{Cant}(ch_c,m)_A = [Z(ch_c,m)_{\mathrm{L}} - Z(ch_c,m)_{\mathrm{R}}] - [Z(ch_c,0)_{\mathrm{L}} - Z(ch_c,0)_{\mathrm{R}}] \tag{8}$$

4 Method B: Delta-First Approach

4.1 Step 1: Calculate Z-Coordinate Deltas

$$\Delta Z(ch_c,m)_{\mathrm{L}} = Z(ch_c,m)_{\mathrm{L}} - Z(ch_c,0)_{\mathrm{L}} \tag{9}$$

$$\Delta Z(ch_c,m)_{\mathrm{R}} = Z(ch_c,m)_{\mathrm{R}} - Z(ch_c,0)_{\mathrm{R}} \tag{10}$$

4.2 Step 2: Calculate Cant Change Directly

$$\Delta\text{Cant}(ch_c, m)_B = \Delta Z(ch_c, m)_\text{L} - \Delta Z(ch_c, m)_\text{R} \tag{11}$$

Substituting the delta definitions:

$$\Delta\text{Cant}(ch_c, m)_B = [Z(ch_c, m)_\text{L} - Z(ch_c, 0)_\text{L}] - [Z(ch_c, m)_\text{R} - Z(ch_c, 0)_\text{R}] \tag{12}$$

5 Equivalence Proof

5.1 Algebraic Expansion

Expand Method A result:

$$\begin{align}
\Delta\text{Cant}(ch_c, m)_A &= [Z(ch_c, m)_\text{L} - Z(ch_c, m)_\text{R}] - [Z(ch_c, 0)_\text{L} - Z(ch_c, 0)_\text{R}] \tag{13}\\
&= Z(ch_c, m)_\text{L} - Z(ch_c, m)_\text{R} - Z(ch_c, 0)_\text{L} + Z(ch_c, 0)_\text{R} \tag{14}
\end{align}$$

Expand Method B result:

$$\begin{align}
\Delta\text{Cant}(ch_c, m)_B &= [Z(ch_c, m)_\text{L} - Z(ch_c, 0)_\text{L}] - [Z(ch_c, m)_\text{R} - Z(ch_c, 0)_\text{R}] \tag{15}\\
&= Z(ch_c, m)_\text{L} - Z(ch_c, 0)_\text{L} - Z(ch_c, m)_\text{R} + Z(ch_c, 0)_\text{R} \tag{16}
\end{align}$$

5.2 Rearranging Terms

Rearrange Method A:

$$\Delta\text{Cant}(ch_c, m)_A = Z(ch_c, m)_\text{L} - Z(ch_c, 0)_\text{L} - Z(ch_c, m)_\text{R} + Z(ch_c, 0)_\text{R} \tag{17}$$

Compare with Method B:

$$\Delta\text{Cant}(ch_c, m)_B = Z(ch_c, m)_\text{L} - Z(ch_c, 0)_\text{L} - Z(ch_c, m)_\text{R} + Z(ch_c, 0)_\text{R} \tag{18}$$

5.3 Conclusion

$$\boxed{\Delta\text{Cant}(ch_c, m)_A = \Delta\text{Cant}(ch_c, m)_B} \tag{19}$$

VERIFIED: Both methods produce identical results.

6 General Formula

The equivalent expressions can be written as:

$$\Delta\text{Cant}(ch_c, m) = [\Delta Z(ch_c, m)_\text{L} - \Delta Z(ch_c, m)_\text{R}] = [\text{Cant}(ch_c, m) - \text{Cant}(ch_c, 0)] \tag{20}$$

7 Implementation Notes

- **Method A** is conceptually clearer: calculate geometry parameter, then find change
- **Method B** is computationally efficient: work directly with coordinate differences
- Both methods are mathematically equivalent due to linearity of the cant calculation
- Choose based on implementation preference or computational efficiency requirements

8 Extension to Twist Calculations

Since twist is the difference in cant over a distance interval, the same equivalence principle applies.

8.1 Twist Definition

For chainages ch_w and $ch_w + \Delta ch$:

$$(ch_w, \Delta ch, m) = \text{Cant}((, ch_) w + \Delta ch, m) - \text{Cant}((, ch_) w, m) \tag{21}$$

8.2 Twist Change Methods

Method A (Twist-First):

$$ch_w \Delta chm_A = (ch_w, \Delta ch, m) - (ch_w, \Delta ch, 0) \tag{22}$$

Method B (Delta-First):

$$ch_w \Delta chm_B = [\Delta\text{Cant}((, ch_) w + \Delta ch, m) - \Delta\text{Cant}((, ch_) w, m)] \tag{23}$$

where ΔCant(, v)alues are calculated using the cant change methods above.

8.3 Twist Equivalence

Since twist is a linear combination of cant values, and cant change calculations are equivalent:

$$\boxed{ch_w \Delta chm_A = ch_w \Delta chm_B} \tag{24}$$

VERIFIED: Twist change methods are equivalent by transitivity of the cant equivalence.

8.4 Computational Efficiency

Method B for Twist is particularly efficient as it:

- Reuses the ΔZ calculations from cant analysis
- Avoids computing intermediate cant and twist values
- Directly computes twist change from coordinate deltas

9 Summary

Both cant and twist change calculations demonstrate mathematical equivalence between parameter-first and delta-first approaches. The choice depends on computational efficiency needs and conceptual clarity preferences.