A Review of Geometric Integrity Criteria for Military Standards – 31000A
Duygu Sap

TR-19-001
September 2019
A Review of Geometric Integrity Criteria for Military Standards - 31000A

Duygu Sap*

International Computer Science Institute,
Berkeley, CA 94704, USA.

September 2019

Abstract

In this article, we focus on the geometric integrity criteria presented in MIL-STD 31000A[7]. We present a general review of the geometric integrity criteria and recommend a method that focuses on the investigation of geometric and topological property violations to identify defects. We propose ways to improve the geometric integrity criteria classes and the recommended correction methods provided by MIL-STD 31000A. We point out how the categorization of the tests based on dimensionality arguments or experimental observations creates repetition in testing. We show that the heuristic checks complicate the testing procedure by letting the designer or the user diverge from investigating the inherent reasons of the invalid geometry. We examine the curve criteria as a local application of our analysis.

We highlight that although the tests are defined based on dimensionality arguments, the threshold definitions disregarded this approach, and dependency needed to be introduced over the threshold values of the geometric components of different dimensions. We present how the inconsistencies in thresholds may result in an unintentional creation of a faulty design or allow a defect to go unnoticed during a model validation. We suggest some alternative threshold values for the aerospace and automotive industries' geometric criteria to eliminate some existing inconsistencies.

*duygu.sap@icsi.berkeley.edu
Contents

1 Introduction 3

2 Overview of MIL-STD 31000A 3
   2.1 Motivation 3
   2.2 Geometry Validation 4
   2.3 Product Geometric Integrity Criteria 6

3 Review of Geometric Integrity Criteria 7
   3.1 Redundancy in Test Classification 8
   3.2 Proposed Test Classification 9
   3.3 Remarks 11
   3.4 Review of the Thresholds for the Aerospace & Automotive Industries 12
      3.4.1 Defects created within threshold 13
      3.4.2 Indetectable defects 13

4 Evaluation of the Curve Criteria 15
   4.1 Curve Criteria 7.1.1 - Large Segment Gap 15
   4.2 Curve Criteria 7.1.4 - Tiny Curve or Segment 16
   4.3 Curve Criteria 7.1.5 - Indistinct Curve Knots 16
   4.4 Curve Criteria 7.1.6 - Self-Intersecting Curve 17
   4.5 Curve Criteria 7.1.7 - Embedded Curves 17
   4.6 Curve Criteria 7.1.8 - Excessively High-Degree Curve
      & Curve Criteria 7.1.9 - Fragmented Curve 18
   4.7 Curve Criteria 7.1.10 - Wavy Planar Curve 19
   4.8 Curve Criteria 7.1.11 - Small Radius of Curvature 20

5 Conclusion 20

List of Figures

1 Aerospace & automotive industries’ threshold values [7] 12
2 A 1D gap of allowable size creating a defect 14
3 Indetectable feature loss 14
4 Curve Criteria 7.1.1 [7] 16
5 Curve Criteria 7.1.4 [7] 16
6 Curve Criteria 7.1.5 [7] 17
7 Curve Criteria 7.1.6 [7] 17
8 Curve Criteria 7.1.7 [7] 17
9 Curve Criteria 7.1.8 and 7.1.9 [7] 18
10 Curve Criteria 7.1.8 [7] 19
11 Curve Criteria 7.1.9 [7] 19
12 Curve Criteria 7.1.10 [7] 19
13 Curve Criteria 7.1.11 [7] 20
1 Introduction

MIL-STD 31000A investigates 3D model validation in three classes: geometry, annotations, and attributes[7]. In this review, we focus on the geometry validation that discovers the defects in geometric entities and the errors in the relations among these entities, and examines model topology.

The recommended practices in [7] are based on currently available, commercial software tools for the inspection and validation of 3D CAD models. Validation procedures provided are necessary but not complete.

In this review, we point out how the categorization of the tests based on dimensionality arguments and experimental observations creates repetition in testing [1](See Section 2.3 and 3.1). We show that the heuristic checks complicate the testing procedure since they let the designer or the user diverge from investigating the inherent reasons of the invalid geometry (See Section 4.6). Moreover, we highlight that although the tests are defined based on dimensionality arguments, the threshold definitions disregard this approach, and dependency needs to be introduced over the threshold values of the geometric components of different dimensions[1]. We show how the existing inconsistencies in the thresholds for the aerospace and automotive industries could yield unexpected issues and suggest a way to overcome such issues by modifying some of the existing thresholds (See Section 3.4).

In Section 2, we provide a summary of the geometry validation procedure discussed in MIL-STD 31000A. In Section 3, we present a general review of the geometric integrity criteria and the recommended methods for identifying defects in [7] and we propose ways to improve the existing procedures. In Section 4, we provide an examination of the curve criteria as a local application of our analysis in Section 3. In Section 5, we remark the key points of our analysis and provide a summary of this review.

2 Overview of MIL-STD 31000A

In this section, we provide a summary of the military standards (31000A) for the geometric integrity criteria [7].

2.1 Motivation

Prime contractors on significant weapons programs use 3D CAD software to design, analyze, manufacture, and inspect the mechanical components and structures of their products. Anyone who may edit or use a model is responsible for the maintenance of the model integrity during the model’s creation and/or revision. Currently, 3D model data is not allowed to be used as master product data within the Department of Defense (DoD), contractors must submit 2D design drawings as permanent records throughout each system’s life.

Drawings are less useful than 3D models in an equipment’s long-term life activities such as manufacturing of spare parts, design improvements, and analysis
of systems if unexpected failures occur. Therefore, the capability of accepting 3D models instead of or in addition to 2D drawings is essential for DoD. Formal, algorithmic validation processes are needed since 3D models may contain subtle errors that can prevent them from being used by downstream applications such as numerically-controlled manufacturing, finite element analysis, and inspection with coordinate-measuring devices.

MIL-STD-31000A provides technical guidance to the government and its contractors for the validation of digital 3D models before their distribution to the government agencies and subcontractors during the design, manufacturing, and logistical-support phases of the life of a weapons system. It identifies the 3D digital model defects and lists the kind of issues each defect may cause in computer applications that use the models. Besides, it provides recommended corrections and acceptance criteria. The recommendations may be applied to 3D models in the proprietary native formats of various CAD/CAM and CAE applications or to equivalent derivative forms of models that have been translated to STEP, compact visualization formats, and proprietary commercial CAD formats. Currently, there is no commercial software that checks all classifications of model data quality, therefore, MIL-STD 31000A has been designed in a way that would allow the addition of new checks and acceptance criteria in future releases.

2.2 Geometry Validation

Defects in geometry and topology may be caused by the following:

- Designers may create invisible flaws unintentionally during the iterative design process.
- CAD software may create defects while trying to resolve the geometry and topology of the model features.
- Data translation software may introduce defects during the conversion of proprietary CAD formats to standard formats, such as IGES or STEP, or the proprietary formats of other software vendors.

Defects due to data translation may result from programming errors or the use of slightly different algorithms for generating model geometry by different systems. Problems may also be caused when the resolution or precision to which a CAD system computes complex geometric intersections differs in the originating and receiving systems. Defects in geometry or topology of models may cause translation failures. 3D model parts may not join to form valid solids in the receiving system after translation. Model faces or parts may get lost.

A well-known translation validation technique used to ensure the geometric accuracy utilizes a set of sampling points chosen on the model surfaces to measure the deviations of the translated surfaces from the original surfaces. This method allows assuring that the variation in the surfaces is within an acceptable range. The main challenge of this method is the difficulty in determining the density
and the distribution of the sampling points. A fast deterministic method for sampling point generation on NURBS curves and surfaces [4] is introduced by using Greville points, which are the points located at average knot values on NURBS geometry. However, this method is applicable only if a NURBS geometry represents the model. On the other hand, there is no need to define sampling points on analytic surfaces such as conic sections since it suffices to determine the parameters of surfaces to identify the curve of intersection for two intersecting analytic surfaces. The commonly used allowable deviation by contractors is \[ \pm 0.01 \text{ mm} \]. However, the number of points required to be tested is usually so large that evaluating the deviations of the points from their corresponding surfaces is too challenging using an interactive approach. Automated data checking software is usually used for identifying defects, and healing software may be needed for correcting errors.

The process for checking 3D models in either an automated or an interactive mode is as follows:

1. The operator determines the quality criteria to be checked and sets the appropriate threshold values in the software.

2. The software checks every part model for errors, and generates a report. In the presence of defects, one of the following steps for correcting errors is taken before the release of the models.
   
   i. Return the models to the designer for correction
   
   ii. Heal the model defects via an automated interactive healing software.

In the case when the parametric model features need to be preserved or the defects are too severe for the healing software to deal with, (i) should be executed. When models are translated from one CAD format to another, feature data may be lost. In such cases, (ii) is executed to correct minor geometric and topological defects that occur during the translation.

Validation procedures are also needed for ensuring that there is an accurate correspondence between a model retrieved from the long-term archive and the original model before it was saved in the archive. Such procedures should be performed independently from the formats used in long-term archives. Currently, the European Association of Aerospace Industries – Standardization (ASD-STAN) and the ProSTEP iViP Association develop and test the standards for long-term archival storage of 3D models.

It is more efficient in terms of time and cost to determine the quality of models and the recommended threshold values for their use in downstream applications before their translation to other formats. The recommended threshold values for each model quality criteria depend on the application 3D CAD model data is needed for. For example, an acceptable threshold for design may be unacceptable for data exchange or finite element analysis.

The following procedure is recommended for determining the data acceptance criteria and their appropriate thresholds by DoD.
Identify the downstream application and the methodology for the reuse of the models with its acceptance criteria.

Collect a representative model set that contains the standard features and tolerances in each design data category.

Process the models through the downstream application(s) and classify them based on the satisfaction of the acceptance criteria.

For every potential data quality criterion, determine the extreme values in the acceptable and unacceptable model groups and derive the thresholds that identify the limits of these groups.

2.3 Product Geometric Integrity Criteria

The defects in curves, surfaces, edges, and edge loops are categorized as follows [7]:

1. Curve Criteria
   - i. Large segment gap – \( G_0 \) discontinuity [3]
   - ii. Non-tangent curves or segments – \( G_1 \) discontinuity [3]
   - iii. Non-smooth curves or segments – \( G_2 \) discontinuity [3]
   - iv. Tiny curve or segment
   - v. Indistinct curve knots
   - vi. Self-intersecting curve
   - vii. Embedded curves
   - viii. Excessively high-degree curve
   - ix. Fragmented curve
   - x. Wavy planar curve
   - xi. Small radius of curvature

2. Surface Criteria
   - i. Large gap between surfaces or patches – \( G_0 \) discontinuity
   - ii. Non-tangent surfaces or patches – \( G_1 \) discontinuity
   - iii. Non-smooth surfaces or patches – \( G_2 \) discontinuity
   - iv. Tiny surface or patch
   - v. Indistinct surface knots
   - vi. Self-intersecting surface
   - vii. Embedded surfaces
   - viii. Excessively high-degree surface
   - ix. Fragmented surface
x. Wavy surface
xi. Small surface radius of curvature
xii. Narrow surface or patch
xiii. Relatively-narrow neighboring patches
xiv. Degenerate surface boundary
xv. Degenerate surface corner
xvi. Unused patches
xvii. Folded surface

3. Edge Criteria
   i. Tiny edge
   ii. Fragmented edge
   iii. Inconsistent edge on curve

4. Edge Loop Criteria
   i. Large edge gap
   ii. Non-tangent edges
   iii. Non-smooth edges
   iv. Self-intersecting loop
   v. Sharp edge angle
   vi. Inconsistent edge in loop

Similar categorizations are provided for faces, shells, and solid bodies. In each category, effects of the defects on the CAD quality data in applications such as FE analysis, NC manufacturing, data exchange, design, and drafting are listed along with the recommended correction methods.

3 Review of Geometric Integrity Criteria

In this section, we propose a more efficient and compact test classification procedure than the one presented in [7]. We highlight the equivalency among the defect classes and the interdependence of the criteria used for geometric integrity. Thus, we show the redundancy in the testing procedures in [7]. Moreover, we point out the inconsistencies that result from the heuristic threshold determination process that presumably yielded the thresholds accepted by the aerospace and automotive industries. We illustrate some potential outcomes of such inconsistencies and provide some suggestions on how to eliminate them.
3.1 Redundancy in Test Classification

There are two kinds of redundancies in the defect, therefore, the test classifications. These redundancies are due to:

1. the dimensional and topological interdependence of the model parts
2. the heuristic definitions used for identifying the defects

(1) could be observed in the broad intersection of tests listed for different geometric or topological entities: curves, surfaces, edges & edge loops, and shells. In Section 2.3, we show that the defect types listed for curves are a subset of the defect types listed for surfaces. This is due to the dimensional dependence between curves and surfaces. It is sufficient to point out a necessary reduction in dimension while defining defects for curves given the list of defects for surfaces instead of deriving a separate list of defects for curves. On the other hand, there is a similar redundancy that arises from the intersection of geometric and topological entities. Shells can be classified as surfaces to a broad extent, and edges and edge loops can be classified as curves. The following lists some of the shell criteria for a comparison with the surface criteria listed in Section 2.3:

i. Large face gap
ii. Non-tangent faces
iii. Non-smooth faces
iv. Self-intersecting shell

(2) results in the definition of multiple defect types that result from the violation of a single geometric property. For example, 1(i), 2(i), and 4(i) listed in Section 2.3 and (i) listed in the shell criteria above are defects that are all due to $G_0$ discontinuity. Moreover, we note that the redundancy in testing due to (2) can also be observed in the descriptions of the potential errors the defects may cause in applications and the recommended practices to be followed for fixing these errors. Here is an example: Consider the defects 1(i) and 2(i).

Effects of 1(i) on various applications are stated as follows [7]:

- Design or drafting: When curves are offset, gaps between segments may be enlarged or overlapping segments may intersect. Cross-sectional views may fail.
- Data exchange: Faces associated with curves may not trim properly.
- Finite element analysis: Unwanted mesh elements may be produced. Automatic mesh generation may fail.
- Numerically controlled manufacturing: Gouges or abrupt changes in angle may occur when cutting curve surfaces.

Similarly, the effects of 2(i) on the same set of applications are stated as follows [7]:

8
• Design and drafting: Surfaces may not combine to produce a solid model
• Data exchange: faces may not trim properly.
• Finite element analysis: Mesh algorithms may fail or produce excessive numbers of elements.
• Numerically controlled manufacturing: Tool path software may fail. Tool paths may gouge the work piece.

Also, the recommended practices for fixing defects of type 1(i) and 2(i) are stated as follows [7]:

• Recommended correction for 1(i): Limit the distance between end points to less than the recommended values. The preferred correction method is extending or trimming one or both entities.
• Recommended correction for 2(i): Extend and trim one or both surfaces to reduce the maximum gap width.

As can be seen from above, the effects of these defects, and the recommended corrections for these defects coincide. This is due to the fact that these defects are results of the violation of the same geometric property.

3.2 Proposed Test Classification

The geometry checks could be considered as subject to the:

1. Curve Criteria
2. Surface Criteria
3. Solid Body Criteria

The following criteria (i) and (ii) are based on the assumption that curves and surfaces are represented by Brep. They could be considered within the categories (1) and (2).

i Edge Criteria

ii Face Criteria

However, this would allow the elimination of redundancy in testing only to some extent. For example, this would help us avoid defining separate tests for searching for tiny surfaces and tiny faces. However, it does not help us avoid searching for the same kind of geometric or topological fault on a surface via distinct tests. It would be more efficient if the tests are classified based on the geometric and topological properties rather than geometric and topological components of models. Validation based on geometric and topological property checks is a technique that has been used by several commercial software tools [2, 5, 6]. However, the existing methods are pretty immature and suffer from the
aforementioned testing redundancies because they lack the analytical approach we provide in this review.

Here is a possible categorization of some tests:

- Large segment gap ($G_0$ discontinuity)
- Large gap between surfaces and patches ($G_0$ discontinuity)
- Large face gap ($G_0$ discontinuity)
- Large edge gap ($G_0$ discontinuity)
- Non-tangent curves or segments ($G_1$ discontinuity)
- Non-tangent surfaces or patches ($G_1$ discontinuity)
- Non-tangent edges ($G_1$ discontinuity)
- Non-tangent faces ($G_1$ discontinuity)
- Tiny curve or segment (Local feature)
- Narrow surface or patch (Local feature)
- Tiny surface or patch (Local feature)
- Tiny face (Local feature)
- Narrow face (Local feature)

This kind of categorization would allow eliminating the repetitive analysis of the effects of defects of the same kind on the same applications and repetitive derivation of recommended corrections for the same kind of defects. It would further yield a more comprehensive analysis. The same geometric or topological fault may occur in different forms and this approach would allow considering all such potential forms in a compact form. This is especially important in the case when a model is to be used in a downstream application. For example, $G_0$ or $G_1$ continuity may be required by a certain application. Ensuring that one potential condition that would yield $G_0$ or $G_1$ discontinuity does not take place is not equivalent to ensuring that all such conditions are absent, thus, the model can be used in this application. In this regard, a more efficient categorization of the model defect checks could be listed as follows:

1. $G_0$ discontinuity: Large gaps
2. $G_1$ discontinuity: Non-tangency, Self-intersecting shapes, Folded surface, Sharp corners: Sharp face angle, Degenerate surface corner
3. $G_2$ discontinuity: Non-smoothness
4. Incorrect NURBS degree: High degree shapes, Fragmented shapes, Wavy shapes
5. Inconsistent orientation: Inconsistent face in shell, Inconsistent edge on curve, Inconsistent edge in loop, Inconsistent face on surface

6. Ill-defined shape: Embedded features

7. Small local features: Tiny/narrow shape parts, Indistinct knots, Small radius of curvature, Degenerate surface boundary

### 3.3 Remarks

In this section, we provide a list of remarks on the equivalency of the defects listed in [7] along with comments on some of the recommended corrections.

- A self-intersecting surface is a special case of a folded surface.

- When a self-intersecting surface/curve is trimmed as recommended, it may turn into a surface with a sharp boundary/corner, thus, create a different kind of defect.

- Edge loop criteria can be considered as a part of edge criteria.

- A shell consists of faces and edges and it can be considered as a primitive form of a surface with a possible non-manifold geometry. Thus, there is no need for a separate shell criteria.

- Indistinct curve knots can be considered as a consequence of tiny knot curves, which are the curves holding the knot points.

We note that the following comments could be considered in improving some of the recommended corrections:

- High degree shapes and fragmented shapes result from the incorrect choice of the polynomial degree in defining the shapes. The degree could be approximated by checking the highest order of $G$ continuities of the relevant shapes.

- Wavy shapes include too many inflection points or saddle points. In the presence of a wavy planar curve, the designer could choose a curve with fewer inflection points if possible or use the inflection points in determining the sign intervals of the curvature $\kappa$ and split the curve at the zeros of $\kappa$.

- Self-intersecting curves mentioned in [7] are results of the wrong offsetting with $d > r_{\text{min}}$ in the concave region, where $d$ denotes the offset distance, and $r_{\text{min}}$ indicates the minimum radius of curvature.
3.4 Review of the Thresholds for the Aerospace & Automotive Industries

In this section, we point out some inconsistencies that result from the heuristic threshold determination process. Figure 1 shows the recommended thresholds for the aerospace and automotive industries.
We note that a large curve (or segment) gap, tiny surface (or patch) and narrow face thresholds highlighted in Figure 1 yield some inconsistencies. The large gap threshold could be set as $(1e - 1) \, mm$ max or the tiny surface area could be set as $(1e - 4) \, mm^2$ min or the narrow face threshold could be set as $(1e - 1) \, mm$ min to eliminate these inconsistencies.

Such inconsistencies in thresholds may result in the following cases:

i. Defects created within threshold: Unintentional creation of a defect or a faulty design

ii. Indetectable defects: A defect or a faulty design passing the validation tests

In Sections 3.4.1 and 3.4.2, we exemplify these cases, thus, illustrate some of the potential outcomes of the aforementioned inconsistencies.

3.4.1 Defects created within threshold

In this section, we show that the large curve (or segment) gap threshold allows the creation of a 2D hole which is too small to be covered by a tiny face. We also point out that such a defect cannot be classified as a tiny feature defect. Thus, the recommended correction for tiny features would not apply.

In Figure 2, the Boolean union of a prism with two rectangular boxes is illustrated. Suppose only the tip of the prism where it intersects with the boxes is hollow such that a 2D hole is created on the intersection plane (See Figure 3a). We note that the gap between the edges of the boxes is $(9e - 3) \, mm$, i.e., the gap is within the allowable threshold (See Figure 1) and the circular cross-section on the intersection plane belongs to a circle with a diameter of $(2e - 2) \, mm$. Thus, the area of the 2D hole with two straight edges and two circular edges is approximately $(2e - 4) \, mm^2$. Note that this 2D hole is smaller than a valid tiny face. Thus, it cannot be covered with a tiny face.

On the other hand, if the correction method for invalid tiny faces could be applied in this scenario, the gap would need to be removed. However, no validation test would invoke that criterion in this scenario. Thus, a defect is created within threshold and could go unnoticed, and it may cause unexpected failures due to the design in which this part is used. Such a defect could also occur after a translation.

However, if the threshold for a tiny face or a 2D hole was set as $(1e - 4) \, mm^2$, then this 2D hole could be covered with a tiny face or a 2D hole of this size would not be an unexpected feature in the design.

3.4.2 Indetectable defects

In this section, we show that the inconsistency between the tiny face and narrow face thresholds could allow a loss of a feature and such a defect may be indetectable.

Figure 3 shows a simple transformer design where the coils have diameters $(1e - 2) \, mm$. Suppose the wires are folded tightly to the core. Thus, each of
the four sides of the core has a face width $(1e-2) \text{mm}$. Note that these faces would pass the narrow face check (See Figure 1). Suppose this transformer was located on a square plate with side length $(7e-2) \text{mm}$. Then, the area of this plate would be $(4.9e-4) \text{mm}^2$, which is less than the allowable tiny face size. As a result, this tiny face would be eliminated by enlarging or extending the neighboring surfaces as recommended in [7]. In the meanwhile, the transformer would be corrupted. This defect may be indetectable through design checks since the rest of the design was faithful to the thresholds. However, if the narrow face width threshold was changed to $(1e-1) \text{mm}$, then the smallest plate that would accommodate this transformer would be of size $(4e-2 + \delta) \text{mm}^2$ for some $\delta > 0$. This small plate would pass the tiny surface test and prevent such a defect from occurring.

We note that the sizes chosen for the transformer in this example may be unrealistically small. The goal of this example is to show that a minor feature loss may be harmful in the long run if it cannot be detected.

![Figure 2: A 1D gap of allowable size creating a defect](image1)

![Figure 3: Indetectable feature loss](image2)
4 Evaluation of the Curve Criteria

In this section, we present a review specifically on the curve criteria provided by [7] as a local application of our analysis. Detailed reports of other criteria listed in [7] can be derived similarly.

We show that the defects presented in 11 different categories based on an empirical approach in [7] could be presented in 6 categories, namely, \(G_0\), \(G_1\), \(G_2\) discontinuities, small local features, incorrect polynomial degree for curves, ill-definedness, by using an analytic approach. Precisely, we show how to classify the curve defects as geometric or topological property violations. We also evaluate the correction methods recommended for these issues in [7].

In Sections 4.1 and 4.4, we present the defects that could be classified as \(G_0\) discontinuity. The defect shown in Section 4.3 could be considered in the same category, namely, small local (of below permissible size) feature, like the one presented in Section 4.2. Section 4.5 illustrates an ill-defined curve. In Section 4.6, we present two defects resulting from a single error, namely, the incorrect choice of polynomial degree. These defects are stated as different kinds of defects in [7] with recommended corrections of one leading to the emergence of the other. Similarly, the defect in Section 4.7 could turn into one of the defects presented in Section 4.6 once the recommended correction is executed. Section 4.8 illustrates another form of defect that could be considered in the same class as the one presented in Section 4.2.

We note that the Curve Criteria 7.1.2 and 7.1.3 in [7] are omitted for they do not require reevaluations.

4.1 Curve Criteria 7.1.1 - Large Segment Gap

*Recommended correction:* Limit the distance between end points to less than the recommended values. The preferred correction method is extending or trimming one or both entities [7].

As can be seen in Figure 4, this issue may occur in three different forms, therefore, we provide three separate corrections as follows:

1. \(G_0\) discontinuity ⇒ Limit the distance between the end points

2. \(G_0\) discontinuity due to overlapping ⇒ Trim one of the curves by \(d(p_1, p_{21})\), where \(p_1\) is the endpoint of segment 1 and \(p_2\) is the projection of the endpoint of segment 2 onto segment 1

3. \(G_0\) discontinuity due to overlapping after offsetting ⇒ Choose an offset value \(\epsilon\) such that \(\kappa_{\text{max}} < \frac{1}{\epsilon_{\text{pedem}}\text{ where } \kappa_{\text{max}}\text{ is the maximum signed curvature of the curve}}\)

We note that offsetting would actually require \(G_1\) continuity which would guarantee \(G_0\) continuity. Thus, if offsetting was not attempted in the absence of \(G_1\) continuity, the third scenario would not be encountered.
4.2 Curve Criteria 7.1.4 - Tiny Curve or Segment

*Recommended correction:* Remove the tiny entity and extend or trim one or both entities to eliminate the gap [7].

This issue could be considered in the small local feature category. However, we also note that curves of length less than a threshold (tiny) could be automatically removed (as recommended) which would turn this scenario into the scenario where the Curve Criteria 7.1.1 applies.

4.3 Curve Criteria 7.1.5 - Indistinct Curve Knots

*Recommended correction:* Remove one of the knot points. If the CAD software does not permit knot removal, recreate the spline with the proper number of knots [7].

Note that recreating the spline with a different number of knot points does not guarantee that the same issue will not be encountered. Clear definitions for the proper number of knot points and proper locations for knot points need to be provided. Thus, we suggest defining a minimum knot interval size and recreating the spline with the necessary number of knot points.
As mentioned in Section 3.3, this defect can be classified as a tiny feature defect, so Curve Criteria 7.1.4 could be applied to remove the tiny knot interval.

### 4.4 Curve Criteria 7.1.6 - Self-Intersecting Curve

**Recommended correction:** Recreate the offset curve in a way that does not produce a self-intersection. Trim the self-intersecting portion of the curve [7].

The intersection in Figure 7 is just the rotated version of the overlap in Section 4.1 which results from the same operation, namely, offsetting. This defect could be identified as a $G_0$ discontinuity issue and could be corrected as in the Curve Criteria 7.1.1.

### 4.5 Curve Criteria 7.1.7 - Embedded Curves

**Recommended correction:** Delete one of the double elements with the fewest dependent relationships. Reassign relationships, if any, to the remaining curve [7].
This defect is due to the ill-definedness of the curve. The recommended correction stated above must be executed.

4.6 Curve Criteria 7.1.8 - Excessively High-Degree Curve & Curve Criteria 7.1.9 - Fragmented Curve

Recommended correction for 7.1.8: Avoid using polynomial curves of high degree. Subdivide high-degree curves into sets of adjacent curves with lower degree [7].

Recommended correction for 7.1.9: Replace the fragmented curve with one that consists of fewer segments [7].

Figure 9: Curve Criteria 7.1.8 and 7.1.9 [7]

Note that the recommended practice of 7.1.8 could easily create the issue described in 7.1.9, and vice versa.
The defect in 7.1.9 may be caused by data translation that approximates a higher degree curve with a set of lower-degree curves and the defect in 7.1.8 may result from the use of an excessively high-degree polynomial curve instead of a piecewise lower-degree polynomial curve, i.e., a fragmented curve (See Figure 10 and Figure 11).
This shows that there needs to be agreed-upon maximum and minimum degrees of polynomial degrees and how to pick one needs to be specified.
This is an important example for it shows how a recommended correction has a tendency to create a defect itself. It emphasizes the need to eliminate the case-by-case study approach in correcting errors. The causes stated and the corrections recommended decouple for there is a single error in these separate defect scenarios, namely, the ambiguity in the choice of the appropriate polynomial degree.
Effects of condition on CAD data quality:

Design and drafting: High-degree curves are susceptible to unwanted undulations and unpredictable behavior when small changes are made.

Data exchange: The receiving system may be incapable of interpreting the high degree curves. Translators may fragment the curve into multiple tiny or discontinuous segments.

Finite element analysis: Automatic mesh generation may fail because software can’t recognize high degree curves. Discontinuous segments may produce excessively fine mesh elements.

Numerically controlled manufacturing: Tool path generation may fail because software can’t recognize high-degree continuous curves. Unwanted undulations may occur in the finished work piece.

Recommended correction: Avoid using polynomial curves of high degree. **Subdivide high-degree curves into sets of adjacent curves with lower degree.**

Figure 10: Curve Criteria 7.1.8 [7]

Effects of condition on CAD data quality:

Design and drafting: Fragmented curves increase model size and are hard to control.

Data exchange: The receiving system may be incapable of accepting the large number of segments.

Finite element analysis: Large numbers of segments may produce excessively fine mesh elements.

Numerically controlled manufacturing: Increases model size unnecessarily.

Recommended correction: Replace the fragmented curve with one that consists of fewer segments.

Figure 11: Curve Criteria 7.1.9 [7]

### 4.7 Curve Criteria 7.1.10 - Wavy Planar Curve

![Wavy Planar Curve](image)

**Recommended correction:** Replace the wavy curve with one that has fewer undulations[7].

The wavy curve could be classified as a high-degree curve. As mentioned in Sec-
tion 3.3, the wavy curve could be split into segments at the inflection points to reduce the number of undulations and have the sign of the curvature piecewise constant on the curve. Note that, this correction, as well as the correction recommended in [7], may turn this issue into the fragmented curve issue described in Section 4.6.

This issue could also be addressed by referring to the moving frame of the curve.

4.8 Curve Criteria 7.1.11 - Small Radius of Curvature

Figure 13: Curve Criteria 7.1.11 [7]

Recommended correction: Replace curve segments that have less than the minimum acceptable radius[7].

Note that this defect could be considered as a tiny feature defect and classified in the same category as the one in Section 4.2. The match between the recommended corrections is also supportive of this proposed approach.

5 Conclusion

In this article, we focused on the geometric integrity criteria presented in [7]. We provided a summary of the geometry validation procedure discussed in MIL-STD 31000A. We presented a general review of the geometric integrity criteria and recommended a method that focuses on the investigation of geometric and topological property violations to identify defects. We proposed ways to improve the geometric integrity criteria definitions and the recommended correction methods provided by [7].

We pointed out how the categorization of the tests based on dimensionality arguments or experimental observations creates repetition in testing. We showed that the heuristic checks complicate the testing procedure by letting the designer or the user diverge from investigating the inherent reasons of the invalid geometry. We examined the curve criteria as a local application of our analysis. Moreover, we highlighted that although the tests are defined based on dimensionality arguments, the threshold definitions disregarded this approach, and dependency needed to be introduced over the threshold values of the geometric components of different dimensions. We presented how the inconsistencies in thresholds may result in an unintentional creation of a faulty design or allow a defect to go unnoticed during a model validation. We also suggested some alternative threshold values for the aerospace and automotive industries' geometric criteria to eliminate some existing inconsistencies.
Acknowledgements

This research was funded in part by DARPA contracts HR0011620042 and HR0011623402. I would like to thank Vadim Shapiro from the University of Wisconsin-Madison, USA and the International Computer Science Institute, Berkeley, USA for his support.

References


