

Article

Affine Invariance of Bézier Curves on Digital Grid

Miklós Hoffmann ^{1,2,*}  and Ede Troll ¹¹ Institute of Mathematics and Computer Science, Eszterházy Károly Catholic University, 3300 Eger, Hungary; troll.ede@uni-eszterhazy.hu² Department of Data Science and Visualisation, University of Debrecen, 4032 Debrecen, Hungary

* Correspondence: hoffmann.miklos@uni-eszterhazy.hu; hoffmann.miklos@inf.unideb.hu

Abstract

Affine invariance is one of the most fundamental properties of free-form curves, ensuring that transformations such as translation, scaling, rotation, and shearing preserve the essential characteristics of the geometric shape. It is exploited by almost every software that uses such curves. However, this property only holds in a theoretical, mathematical sense. The transformation of a curve calculated and displayed on computers using finite precision arithmetic and representation may not be fully identical to the curve calculated from the transformed control points. This deviation, even pixel-level inaccuracy, can cause problems in various applications, such as Computer-Aided Geometric Design, medical image processing, numerical computations, and font design, where this level of error can have serious consequences. In this paper, we study and demonstrate the extent and nature of this deviation using geometric and statistical tools on a cubic Bézier curve. We provide practical methods to mitigate this inaccuracy and decrease the error level using fast and simple alternative computations of the curve, taking advantage of the symmetry of the basis functions, elevating the degree of the curve, and using reparametrization to evaluate the curve on integer values. The effectiveness of these alternatives is evaluated by statistical methods based on 500,000 transformations.

Keywords: free-form curves; Bézier curve; affine transformation; geometric modeling; inaccuracy

MSC: 68U05; 68U07; 68U10; 65D17; 65D18



Academic Editor: Gabriel Eduard Vilcu

Received: 7 October 2025

Revised: 9 November 2025

Accepted: 14 November 2025

Published:

Citation: Hoffmann, M.; Troll, E. Affine Invariance of Bézier Curves on Digital Grid. *Mathematics* **2025**, *1*, 0. <https://doi.org/>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Bézier curve was originally developed in the automotive industry for computer-aided design purposes. The basic construction is very intuitive and user-friendly—the control points are freely given by the designer, and the curve is calculated as a weighted sum of these points multiplied by some blending functions, in this case, by the well-known Bernstein–Bézier basis functions, grounded in Bernstein polynomials (see, e.g., [1,2]). This has proven to be such a successful construction that it has also been used in many other fields, such as image processing, computer graphics and animation, the aerospace industry, robotics, font design, and optics. But these functions are also fundamental tools in computational mathematics, particularly in the finite element method (FEM), for solving differential equations. In all cases, precise computation and well-defined invariant properties are essential in effective application.

Its geometric construction, precisely because of its high degree of design freedom, together with some other advantageous properties of the curve, has become a kind of

golden standard for the further development of free-form curves. The B-spline curve, the non-uniform rational B-spline (NURBS) curve, and curves using various trigonometric blending functions or even functions forming a basis of other function spaces all inherited the important and useful properties of the Bézier curve.

Affine invariance is one fundamental property of these free-form curves, ensuring that transformations such as translation, scaling, rotation, and shearing preserve the essential characteristics of the geometric shape. Let the Bézier curve be defined by the control points $\mathbf{p}_i$, ($i = 0, \dots, n$) as

$$\mathbf{s}(t) = \sum_{i=0}^n B_i^n(t) \mathbf{p}_i(t), \quad t \in [0, 1]$$

where $B_i^n(t)$ represent the Bernstein polynomials of degree n . If A is an affine transformation, then invariance under A means that

$$A \left(\sum_{i=0}^n B_i^n(t) \mathbf{p}_i(t) \right) = \sum_{i=0}^n B_i^n(t) A(\mathbf{p}_i(t)), \quad t \in [0, 1]. \quad (1)$$

The property is a direct consequence of the fact that the non-negative blending functions, the Bernstein polynomials, form a partition of unity, which holds for the blending functions of every other type of free-form curve as well.

Affine invariance is exploited by almost every software that uses such curves in a way that the transformation of curves defined by control points is calculated by transforming only the control points and evaluating the new curve on the transformed control points—the resulting shape will be the same as if the curve itself, including all its points, had been transformed.

However, this property holds only in the theoretical, mathematical sense. The transformation of a curve calculated and displayed on computers using finite precision arithmetic and representation may not be fully identical to the curve calculated on the transformed control points.

This deviation, even pixel-level inaccuracy, can cause problems in various applications, where this level of error can have serious consequences, such as escalation of errors in further calculations or loss of the property of digital convexity.

In this paper, we study and demonstrate the extent and nature of this deviation using geometric and statistical tools, and we suggest practical methods to mitigate this inaccuracy and decrease the error level through fast and simple alternative computations of the curve.

2. Related Work

Numerical stability and errors of evaluation algorithms for parametric curves and surfaces (including Bézier, B-spline, and NURBS models) have been studied in several papers. A recent comprehensive survey reviews how evaluation complexity, algorithmic choices, and numerical stability interact in practical evaluators for parametric curves and surfaces [3]. The survey is about complexity/stability trade-offs but does not investigate invariance failures under floating point.

A study of the numerical stability of B-spline functions, including error tables for interpolation by cubic splines but also by higher degree splines, is presented in [4], and a study of multi-degree splines is presented in [5].

A direct analysis of Bézier evaluation, error bounds, and stability, including a review of Bernstein-based and Bézier-type evaluations and error analyses (forward, backward, and running error) for de Casteljau-type algorithms, can be found in [6], and a study on rational Bézier curves is presented in [7]. A compensated algorithm for the evaluation of polynomials in Bernstein form (hence Bézier curves) under finite-precision arithmetic

is provided in [8], also discussing forward error analysis and numerical experiments for evaluation algorithms.

However, these papers do not directly address transformations of the curve and do not explicitly analyze the numerical stability and floating-point error of parametric curves against the affine invariance property, that is, how the floating point effects interact with affine transformations of control polygons and display grids.

Other important topics in recent works are B-splines, NURBS, and Boundary Element Methods (BEMs) within Isogeometric Analysis (IGA). IGA unifies CAD geometry and numerical simulation via spline bases. This integration is especially attractive for boundary-only formulations that operate directly on CAD curves and surfaces, i.e., the isogeometric Boundary Element Method (IgA-BEM). In acoustics, Simpson *et al.* introduced an IgA-BEM based on T-splines together with a regularized Burton–Miller formulation for Helmholtz problems, demonstrating accuracy and robustness while eliminating geometry clean-up and meshing [9]. Most recently, Desiderio *et al.* presented hierarchical matrix technology for multi-patch IgA-BEM applied to three-dimensional Helmholtz problems, showing that industrial-scale acoustic analyses can be performed directly on CAD geometry [10]. These results show that accurate evaluation of spline bases and geometry maps underpins quadrature/collocation, normals, and fast solvers in IgA-BEM, and establish the centrality of spline bases in precision-sensitive pipelines (including IgA-BEM).

Although our experiments in this paper focus on cubic Bézier curves, the proposed remedies are basis-agnostic and carry over to the spline bases used in IgA-BEM: integer parameter evaluation and symmetry-aware splitting apply at the element level, and degree elevation is a standard spline operation that improves numerical behavior without altering geometry. In this sense, our findings complement IgA-BEM developments by addressing a fundamental numerical assumption—affine invariance under floating-point arithmetic—whose violation can subtly shift quadrature or collocation locations and, in turn, affect assembly and solver accuracy.

Whereas prior studies focused on the stability of evaluation or devise precision-savvy renderers, our work quantifies and mitigates the loss of affine invariance itself on digital grids, provides practical drop-in strategies (with implementations that match typical GPU/graphics precision), and validates them at scale with new diagnostics that expose asymmetric error behavior along the parameter domain. These observations and remedies extend the state of the art in a way that is directly applicable for many applications, including both geometric modeling and precision-sensitive rendering pipelines.

3. Importance of Precision in Affine Invariance

Although affine invariance can also be considered as a continuous evolution flow of a shape in the affine scale-space [11], in this paper, we consider affine invariance under a singular transformation.

In this section, we present a brief survey of some of the above-mentioned fields, where the affine invariance of the Bézier curve and the Bernstein polynomials play essential roles in applications. We pay special attention to the importance of (in)accuracy, since affine invariance is not only a theoretical principle but also a practical tool for effective computation.

3.1. Computer-Aided Geometric Design

Bézier curves are integral to Computer-Aided Geometric Design, where designers create complex shapes and surfaces. A high level of accuracy of Bézier curves is critical in computer-aided design for several reasons. CAD models are often used as inputs for manufacturing processes, such as CNC machining, 3D printing, or laser cutting, where

tolerances are extremely tight (for a recent overview, see [12]). High-level accuracy ensures that the curves are rendered and interpreted exactly as designed and do not exhibit artifacts such as jagged edges or discontinuities, which can compromise the visual and functional quality of the design; thus, errors are minimized in the final product.

Designers also frequently zoom in and out while working on a model. High-level accuracy ensures that the representation of Bézier curves remains consistent regardless of the zoom level, providing confidence in the fidelity of the design.

In Computer-Aided Geometric Design, accurate computation and geometric properties of curves, such as convexity, intersections, or collisions, are of utmost importance. These are also crucial for operations such as trimming, joining, or Boolean operations. High-level precision ensures that these calculations remain reliable throughout the design process, independently of the applied transformations.

Many industries, such as automotive and consumer electronics, demand curves with high aesthetic quality, such as class-A Bézier curves [13]. Even very small inaccuracies in the curve can lead to visible imperfections, reducing its appeal or functionality.

Pixel-level accurate Bézier curves contribute to efficient rendering in visualization software, ensuring that the model looks correct in real-time applications, such as virtual prototyping or augmented reality previews. Errors or inaccuracies in the design may require additional post-processing or correction downstream, which increases cost and time. Alternative Bézier curve representations, discussed in Section 5, reduce these issues.

3.2. Numerical Computations

In numerical computations, Bézier basis functions are employed to construct trial and test function spaces to solve differential equations [14,15]. This has also been generalized by using B-spline or NURBS basis functions [16,17]. Their inherent affine invariance ensures that numerical solutions remain consistent under affine transformations, enhancing computational accuracy and reliability.

Affine invariance contributes to the stability of numerical methods in several ways. In finite element methods, the system matrix formed from shape functions can be ill-conditioned if the basis functions are not invariant under affine transformations. This can lead to inaccurate solutions or numerical instability. But this is also essential in terms of error propagation control. If a numerical scheme lacks affine invariance, small changes in the domain shape due to transformations can cause errors to accumulate differently across different configurations. Finally, it can also affect robustness, since numerical solvers often require different refinement strategies; affine invariance ensures that accuracy does not degrade when transforming elements.

3.3. Image Processing and Medical Applications

The affine invariance property of free-form curves is also extensively used in affine contour matching and image registration, where, especially in medical image processing, pixel-level accuracy is of utmost importance [18].

Pixel-level accuracy of curves is also essential in applications where the inflection point of the curve, or in the case of a closed curve, its digital convexity, plays a prominent role. In extreme cases, the latter properties can depend on a single pixel [19,20]. Thus, although the affine transformation preserves convexity in principle, it is possible that a convex region bounded by a closed Bézier curve will not be convex after the affine transformation when the shape is evaluated using the transformed control points.

In medical applications, such as the design of custom implants or prosthetics, extremely precise modeling is required at the pixel level, where imprecision is often measured in thousandths of a millimeter [21]. But the design process often involves scaling or shear-

ing the model to different sizes, making affine invariance critical in keeping the shape and precision level as accurate as possible [22–24].

3.4. Font Design and Silhouette

Digital font designers often use Bézier curves to define character shapes. Affine invariance guarantees that characters can be scaled, rotated, or sheared without unwanted distortion, ensuring consistent appearance across various media and devices [25,26]. The exploitation of affine invariance is especially relevant in large-scale vector font generation in an efficient and scalable way, such as in the recent studies of [25], where a Chinese font component dataset is created to synthesize characters by combining components using an affine transformation-based method. A Component Affine Transformation Regressor is trained to regress affine transformation matrix for each individual component image, and this is applied to its corresponding component to produce the transformed version with appropriate scale and position. It is important to note that fonts, in their rasterized form, typically consist of relatively few pixels; therefore, its accuracy is of great importance.

Affine invariance and its pixel-level accuracy also play similarly crucial roles in silhouette vectorization; recent studies intended to utilize techniques that promote affine invariance in order to fully explore mathematically and geometrically meaningful features of the silhouette, notably focusing on even subpixel accuracy in reconstruction [27].

In summary, Bézier curves with a high level of accuracy ensure high precision, visual quality, and functional reliability independently of the applied transformations, which are all essential in the demanding workflows of several applications. In the next section, we will discuss error measurements and potential alternative methods to compute the curve in order to reduce affine inaccuracy. The results are application-independent. Therefore, they can be applied in any of the fields described in this section, and they can also be incorporated into the classical graphics pipeline.

4. Methodology of Error Measurement of Transformations

Our main aim was to evaluate the level of degradation of the affine invariance property mentioned above and determine the possibility of reducing potential errors. Our investigations were carried out on a Bézier curve with a degree of 3, but these results can easily be generalized to higher degrees, to other basis functions, and to other free-form curve types. To avoid possible bias and unintended coincidences, both the Bézier curves and the transformations themselves were randomly generated, and this was repeated on a large number of samples, as described below.

The source code of the statistical analysis was written in C#, which uses the IEEE-754 [28] standard to represent single precision floating point numbers. Math calculations (e.g., trigonometric functions) were performed with the floating point MathF class provided by .NET 8.0.22. We chose the single floating point representation because most graphics software (Unity, Blender, etc.) and Graphics Processing Units (GPUs) themselves use this specific number representation.

Our measurements were first performed on general affine transformation. Affine transformations are represented as

$$T = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

where $a_{11}, a_{12}, a_{21}, a_{22} \in \mathbb{R}$ and $\det(T) \neq 0$.

We compare the points of the two transformed curves, where one is computed by transforming only the control points and the other one is computed by direct transformation. Given the original curve $\mathbf{s}(t)$, let $\mathbf{s}^d(t)$ be the curve whose points were transformed directly,

one by one, and let $\mathbf{s}^c(t)$ be the curve where only control points were transformed, and the curve is calculated by using these transformed control points.

The discrepancy between the two curves, $\mathbf{s}^d(t)$ and $\mathbf{s}^c(t)$, can be measured by the average length of deviation vectors:

$$\Delta \mathbf{s} = \frac{\sum |\mathbf{s}^d(t) - \mathbf{s}^c(t)|}{N},$$

where $t \in [0, 1]$ and $N \in \mathbb{N}$ are the numbers of parameter values where the curve has been evaluated. The difference between the two curves can be visually observed in an enlarged vectorized form in Figure 1.

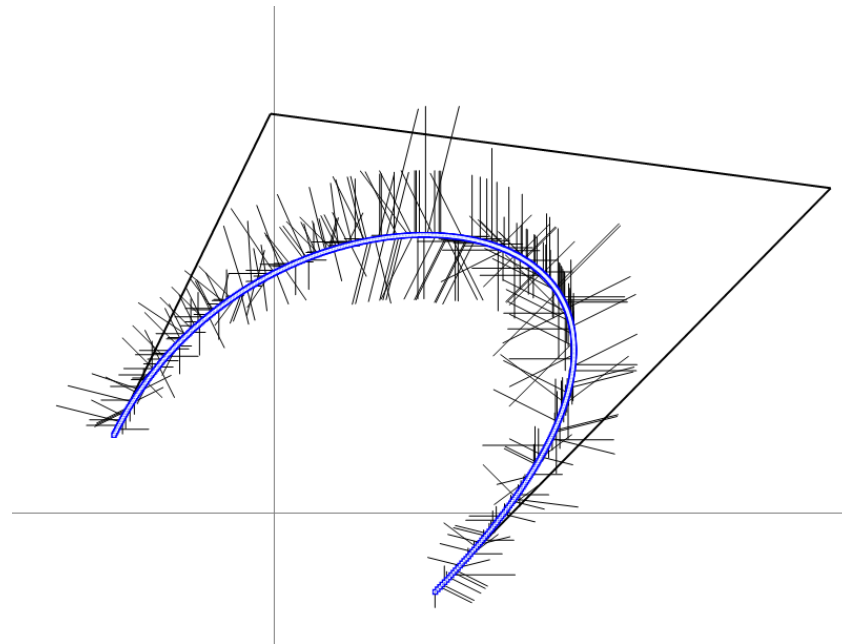


Figure 1. Deviation vectors (enlarged) between $\mathbf{s}^d(t)$ and $\mathbf{s}^c(t)$ after a 60° rotation.

Since we also noticed a constantly present and measurable difference in the length of the deviation vectors in the first and second halves of the curve, we further introduced the following notations to measure the same deviation separately for the first and second parts of the curve.

$$\Delta \mathbf{s}^{\leftarrow} = \frac{\sum |\mathbf{s}^d(t^{\leftarrow}) - \mathbf{s}^c(t^{\leftarrow})|}{N}$$

$$\Delta \mathbf{s}^{\rightarrow} = \frac{\sum |\mathbf{s}^d(t^{\rightarrow}) - \mathbf{s}^c(t^{\rightarrow})|}{N},$$

where $t^{\leftarrow} \in [0, 0.5]$ and $t^{\rightarrow} \in [0.5, 1]$. To measure the difference between the first and second parts, we use the notation $\Delta \mathbf{s}^{\leftrightarrow} = |\Delta \mathbf{s}^{\leftarrow} - \Delta \mathbf{s}^{\rightarrow}|$. Our last measurement, specifically important for computer image processing, is the average number of different pixels per curve, denoted by $\Delta[\mathbf{s}]$.

For our statistics, we used a uniformly distributed pseudo-random generator to generate the coordinates of the control points $\mathbf{p}_i$ of the curve and coefficients a_{ij} of the affine transformation performed. The results of our statistics are calculated on 500,000 randomly defined and transformed curves overall. The basis functions were evaluated in a classical way. The details of the statistical analysis are described in Section 6, but the overall results are summarized in Table 1.

Table 1. Average error per curve. $\Delta[s]$: the number of different pixels; Δs : the difference between the curves; $\Delta s^{\leftarrow}$: the difference between the first halves of the curves; $\Delta s^{\rightarrow}$: the difference between the second halves of the curves; $\Delta s^{\leftrightarrow}$: the difference between the two parts of the curves.

$\Delta[s]$	Δs	$\Delta s^{\leftarrow}$	$\Delta s^{\rightarrow}$	$\Delta s^{\leftrightarrow}$
3.376701111	2.87663569	1.532766562	1.343869128	0.188897435

We make the following observations from this statistical measurement:

- The discrepancy (Δs) is definitely present;
- The average number of misplaced pixels $\Delta[s]$ is more than 3, that is, practically every affine transformation can yield multiple occurrences of pixel-level inaccuracy;
- There is a significant difference ($\Delta s^{\leftrightarrow}$) between the level of discrepancy of the first and second parts of the curve ($\Delta s^{\leftarrow}$ and $\Delta s^{\rightarrow}$).

The alternative computational methods of the curve presented in the following sections aim to reduce the total length of the difference vectors, minimize the occurrence of different pixels, and reduce the discrepancies between $t \in [0, 0.5]$ and $t \in [0.5, 1]$.

5. Alternative Methods to Compute the Bézier Curve

In this section, we aim to provide various practical solutions to reduce the discrepancy between the two transformed curves mentioned above.

We present three independent approaches to mitigate the effect of inaccuracy: first, using the symmetry property of the Bernstein-Bézier basis functions; second, elevate the degree of the curve from 3 to 4; third, transform the basis functions from the usual domain of definition $[0, 1]$ to $[0, N]$ in order to evaluate the functions over integer values.

5.1. Using Symmetric Basis Functions

It is well known that the polynomial basis functions $B_i^n(t)$ of the Bézier curve are symmetric (see [2]). Taking advantage of the symmetry of the basis functions, the curve

$$\mathbf{s}(t) = \sum_{i=0}^n B_i^n(t) \mathbf{p}_i, t \in [0, 1]$$

can be described as

$$\begin{aligned} \mathbf{s}^{\downarrow}(t) &= \mathbf{s}^{\leftarrow}(t^{\leftarrow}) \cup \mathbf{s}^{\rightarrow}(t^{\leftarrow}), \text{ or} \\ \mathbf{s}^{\uparrow}(t) &= \mathbf{s}^{\leftarrow}(t^{\rightarrow}) \cup \mathbf{s}^{\rightarrow}(t^{\rightarrow}), \end{aligned}$$

where

$$\begin{aligned} \mathbf{s}^{\leftarrow}(t^{\leftarrow}) &= \sum_{i=0}^n B_i^n(t^{\leftarrow}) \mathbf{p}_i & \mathbf{s}^{\leftarrow}(t^{\rightarrow}) &= \sum_{i=0}^n B_{n-i}^n(t^{\rightarrow}) \mathbf{p}_i \\ \mathbf{s}^{\rightarrow}(t^{\leftarrow}) &= \sum_{i=0}^n B_{n-i}^n(t^{\leftarrow}) \mathbf{p}_i, & \mathbf{s}^{\rightarrow}(t^{\rightarrow}) &= \sum_{i=0}^n B_i^n(t^{\rightarrow}) \mathbf{p}_i, \end{aligned}$$

where $t^{\leftarrow} \in [0, 0.5]$ and $t^{\rightarrow} \in [0.5, 1]$.

5.2. Elevate the Degree

There is a standard procedure in computer graphics to change (elevate) the degree of the curve, while the geometric shape is preserved (see [2]). In the cubic case, the original Bézier curve has four control points: $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$. To elevate the degree by 1, we need five suitably positioned control points $\mathbf{q}_i, (i = 0, \dots, 4)$. To preserve the endpoint interpolation,

let us define the new endpoints as $\mathbf{q}_0 = \mathbf{p}_0$ and $\mathbf{q}_4 = \mathbf{p}_3$. For the other control points, the convex combination $\mathbf{q}_i = \frac{i}{4}\mathbf{p}_{i-1} + \left(1 - \frac{i}{4}\right)\mathbf{p}_i$, ($i = 1, 2, 3$) is applied. We investigated what happens for the previously mentioned metrics when the original cubic curve is replaced by the fourth-degree curve.

5.3. Transformed Basis Functions

The Bézier curve $\mathbf{s}(t)$ (and its basis functions) are evaluated over the interval $[0, 1]$. In the geometric modeling and visualization algorithms, we compute $N \in \mathbb{N}$ points over this interval, so that we define a set of parameters $t_0 < t_1 < \dots < t_{N-1} \in [0, 1]$, where $t_0 = 0$ and $t_{N-1} = 1$. For the reasons discussed above, these values generate an incremental error. The essence of our new approach is to transform the curve into the domain of definition $[0, N]$, and therefore, the resulting series of parameters $t_0^d < t_1^d < \dots < t_{N-1}^d \in [0, N]$ will have all the elements as integers. Even for incremental algorithms, integers can be correctly represented by the floating point number type and the addition operator. This does not mean, of course, that the error is fully eliminated since, during the transformation, additional operations (division) appear in the calculation of the curve, which also generates errors.

The basis functions of the Bézier curve are $B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$, where $i = 0, 1, \dots, n$. The domain transformation of the curve and its basis functions can be defined as

$$\bar{\mathbf{s}}(T) = \sum_{i=0}^n \bar{B}_i^n\left(\frac{T}{N}\right),$$

where $N \in \mathbb{N}$, $N > 0$ and $T = 0, 1, \dots, N$.

In the case of cubic curves, $\bar{\mathbf{s}}(T) = \sum_{i=0}^3 \bar{B}_i^3\left(\frac{T}{N}\right)$, where

$$\begin{aligned}\bar{B}_0^3(T) &= 1 - \frac{3T}{N} + \frac{3T^2}{N^2} - \frac{T^3}{N^3}, \\ \bar{B}_1^3(T) &= \frac{3T}{N} - \frac{6T^2}{N^2} + \frac{3T^3}{N^3}, \\ \bar{B}_2^3(T) &= \frac{3T^2}{N^2} - \frac{3T^3}{N^3}, \\ \bar{B}_3^3(T) &= \frac{T^3}{N^3},\end{aligned}$$

where $T = 0, 1, 2, \dots, N$. The behavior of the curves is also investigated in the case of an increased degree, applying degree elevation for the modified domain. Accordingly, the transformed shape of the fourth-degree curve is $\bar{\mathbf{s}}(T) = \sum_{i=0}^4 \bar{B}_i^4\left(\frac{T}{N}\right)$, where

$$\begin{aligned}\bar{B}_0^4(T) &= 1 - \frac{4T}{N} + \frac{6T^2}{N^2} - \frac{4T^3}{N^3} + \frac{T^4}{N^4}, \\ \bar{B}_1^4(T) &= \frac{4T}{N} - \frac{12T^2}{N^2} + \frac{12T^3}{N^3} - \frac{4T^4}{N^4}, \\ \bar{B}_2^4(T) &= \frac{6T^2}{N^2} - \frac{12T^3}{N^3} + \frac{6T^4}{N^4}, \\ \bar{B}_3^4(T) &= \frac{4T^3}{N^3} - \frac{4T^4}{N^4}, \\ \bar{B}_4^4(T) &= \frac{T^4}{N^4},\end{aligned}$$

where $T = 0, 1, 2, \dots, N$. It is important to note that although the transformed basis functions could be simplified, from an implementation point of view, the above form has proven to be the most efficient in terms of reducing errors.

6. Statistics

For the statistics, pseudo-randomly generated control points of the curve were calculated over the domain $[-l, l] \times [-l, l] \in \mathbb{R}^2$, where $l \in \{10, 100, 1000\}$. For each l , the elements of the affine transformation matrix $a_{11}, a_{12}, a_{21}, a_{22}$ are generated over the interval $[-k, k]$, where $k \in \{10, 100, 1000\}$. The calculation was performed twice for each curve, once evaluating the function at 500 and once at 1000 points. Overall, 500,000 randomly generated curves and transformations were evaluated. The results presented are the averages of these measurements. As shown in Figure 2, the total length of the difference vectors increases significantly as the value of l increases.

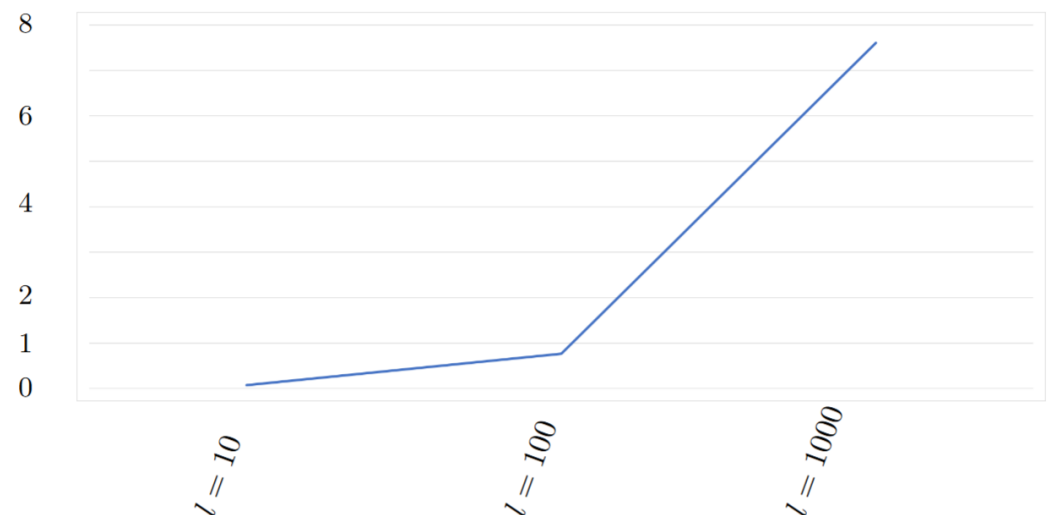


Figure 2. The increasing sum of the difference vectors as a function of the value of l .

Using the curve computational methods discussed in the previous chapter, in the case of randomized affine transformations, the improvements shown in Table 2 were achieved.

Table 2. Different solutions to reduce the errors in case of affine transformations. Third-degree cases that improved the original result (red) are marked in blue, and fourth-degree cases in green. We indicated the best performance (least error) in bold. Notations: $\mathbf{s}$: curve computed by the original method; $\bar{\mathbf{s}}$: curve with transformed domain of definition; $\mathbf{s}^\downarrow$ and $\mathbf{s}^\uparrow$: curves computed using the symmetry of the basis functions. $\Delta[\mathbf{s}]$: the number of different pixels; $\Delta\mathbf{s}$: the difference between the curves; $\Delta\mathbf{s}^{\leftarrow}$: the difference between the first halves of the curves; $\Delta\mathbf{s}^{\rightarrow}$: the difference between the second halves of the curves; $\Delta\mathbf{s}^{\leftrightarrow}$: the difference between the two parts of the curves.

		$\Delta[\mathbf{s}]$	$\Delta\mathbf{s}$	$\Delta\mathbf{s}^{\leftarrow}$	$\Delta\mathbf{s}^{\rightarrow}$	$\Delta\mathbf{s}^{\leftrightarrow}$
degree 3	original curve $\mathbf{s}$	3.376701	2.876635	1.532766	1.343869	0.188897
	domain transf. $\bar{\mathbf{s}}$	3.370310	2.871893	1.532652	1.339241	0.193411
	symmetric 1 $\mathbf{s}^\uparrow$	3.385122	2.884936	1.541032	1.343904	0.197127
	symmetric 2 $\mathbf{s}^\downarrow$	3.370470	2.871731	1.534461	1.337270	0.197190
degree 4	original curve $\mathbf{s}$	3.225944	2.754859	1.515982	1.238877	0.277105
	domain transf. $\bar{\mathbf{s}}$	3.216758	2.749627	1.515860	1.233767	0.282093
	symmetric 1 $\mathbf{s}^\uparrow$	3.232057	2.761692	1.523491	1.238201	0.285290
	symmetric 2 $\mathbf{s}^\downarrow$	3.219706	2.749931	1.517510	1.232420	0.285089

One of the main questions of our research was whether the methods we tested could reduce the total length of the mismatch vectors caused by performing the transformations in different ways and whether it was possible to reduce the number of different pixels found. The results are presented in Figures 3 and 4.

It is important to note that although most of the solutions we tested decreased the error measures, they negatively affected the deviation between the two sides of the curve (see Figure 5).

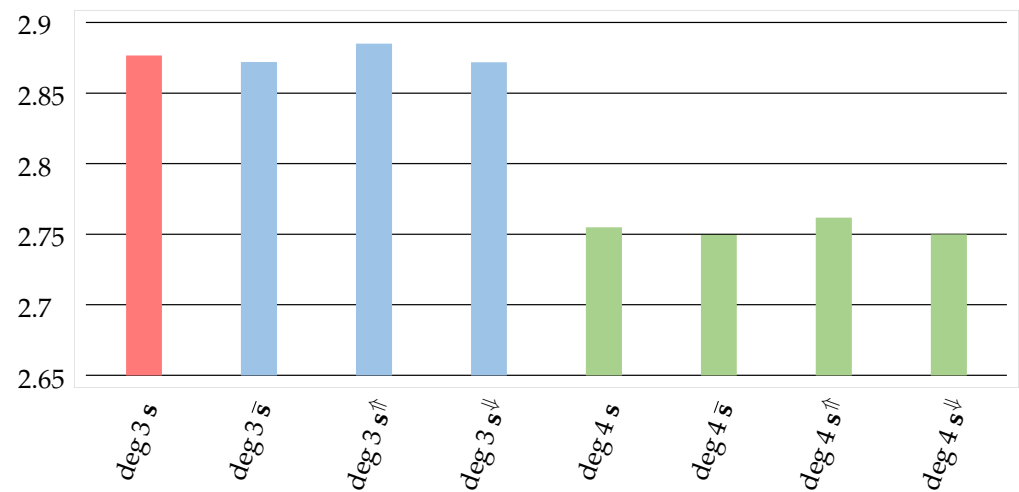


Figure 3. Δs in case of various computational methods.

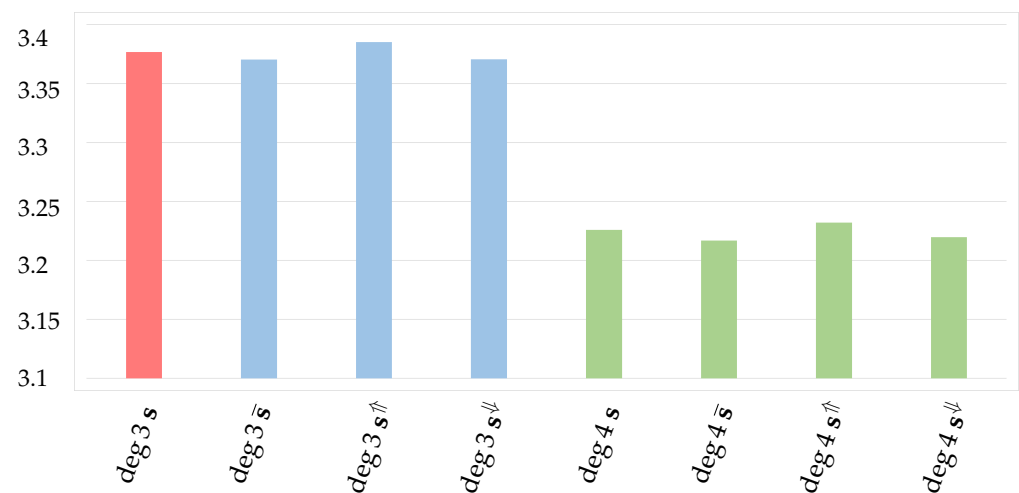


Figure 4. $\Delta[s]$ in case of various computational methods.

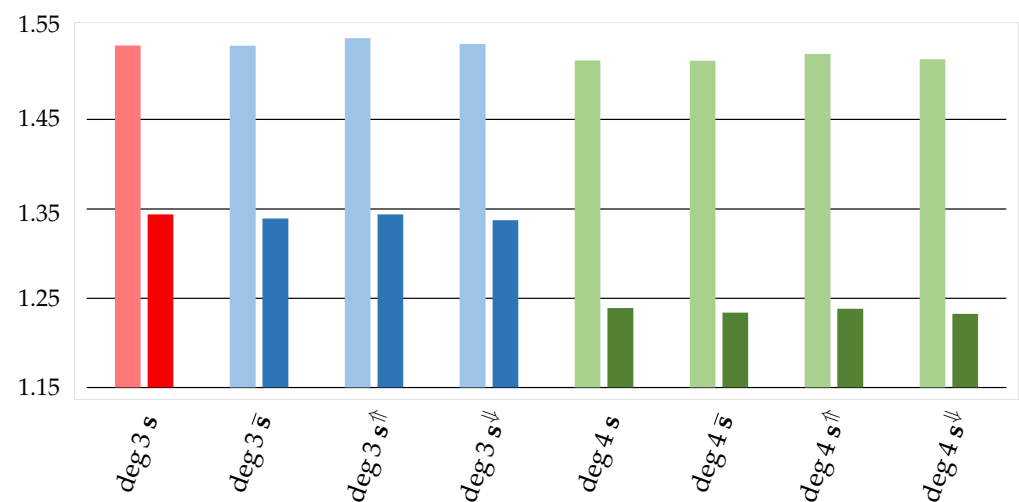


Figure 5. Comparison of $\Delta s^{\leftarrow}$ (left columns) and $\Delta s^{\rightarrow}$ (right columns) in case of various computational methods. The first half of the curve has significantly larger error in each case.

Methods for Verifying Statistical Results

To verify the statistical results, we computed the sum of the lengths of the point-to-point deviation vectors for each curve and then performed comparative statistical analysis. All our methods were compared to the original deg 3 $\mathbf{s}$ method. The results of the F-tests are shown in Table 3. The standard deviations are similar for the 3-degree curves but differ significantly for the 4-degree curves. Therefore, we used a paired t -test for the 3-degree curves and Welch's t -test for the 4-degree curves, as shown in Table 3.

Table 3. Comparison of statistical test methods and results of validation. The analysis always takes the original cubic curve as a reference point.

		F-Test	p -Value	Method	Average
degree 3	original curve $\mathbf{s}$				3.37679
	domain transf. $\bar{\mathbf{s}}$	0.921477	0.564279	paired t -test	3.37031
	symmetric 1 $\mathbf{s}^{\uparrow}$	0.933111	0.875981	paired t -test	3.38512
	symmetric 2 $\mathbf{s}^{\downarrow}$	0.572099	0.166039	paired t -test	3.37047
degree 4	original curve $\mathbf{s}$	0.019836	8.3088×10^{-151}	Welch's t -test	3.22594
	domain transf. $\bar{\mathbf{s}}$	0.037078	3.3378×10^{-159}	Welch's t -test	3.21675
	symmetric 1 $\mathbf{s}^{\uparrow}$	0.020748	1.0004×10^{-153}	Welch's t -test	3.23205
	symmetric 2 $\mathbf{s}^{\downarrow}$	0.003809	1.2761×10^{-137}	Welch's t -test	3.21971

Based on the tests, it is evident that for the 4-degree curves, the expected values differ significantly from the original method in all cases. In these cases, our methods substantially reduced the error rate according to the average values. It is also important to note that we performed additional tests on higher-degree curves. For fifth-degree curves, our methods further reduced the error rate compared to the degree 4 case, although the improvement was less pronounced than in the case of elevating the degree from 3 to 4. We did not perform measurements for higher degrees.

Although the paired t -test did not reveal a statistically significant difference between the means in cubic cases, the direction and consistency of the results across all parameter ranges suggest that these differences are systematic rather than random. The lack of statistical significance is primarily due to the extremely small absolute magnitude of the deviations—typically on the order of 10^{-5} – 10^{-4} per curve point—which makes it difficult for classical significance tests to detect an effect even when it consistently appears in the same direction.

From a geometric perspective, however, these small pointwise discrepancies accumulate along the curve, and the total reduction in $\Delta \mathbf{s}$ becomes measurable on the scale of 10^{-3} , corresponding to a visible, though subtle, improvement in affine consistency. Since the variance of the results remains stable and the mean difference is negative (i.e., smaller errors) except in the case of deg 3 $\mathbf{s}^{\uparrow}$, both of the methods of deg 3 $\mathbf{s}^{\downarrow}$ and deg 3 $\bar{\mathbf{s}}$ can be regarded as slightly better than the baseline and practically superior in preserving affine invariance.

We also performed another set of tests using the third-degree methods. For all curves examined, we selected the third-degree method that resulted in the lowest error. In 77.86% of the cases, one of our methods produced a smaller error. In these cases, the average error reduction was 0.075, and the maximum error reduction was 0.819. For curves where none of our degree-preserving methods improved the result, the difference between the original method and our best method was 0.383. This shows that in cases where our new degree-preserving methods did not reduce the error, the resulting deterioration was smaller than the typical improvement they achieved.

In summary, the statistical tests clearly validate the superiority of the methods applying 4-degree curves. And while the tests do not indicate a significant mean difference at the

cubic level, the observed pattern of consistently lower total deviation lengths provides practical evidence that the $\deg 3 \bar{s}$ and $\deg 3 s^{\downarrow}$ methods improve the stability and geometric fidelity of the curves under affine transformations.

7. Discussion

We examined the affine invariance, an essential property of free-form curves which theoretically always holds but is burdened with errors in finite precision practice, in the case of cubic Bézier curves. As we have seen in the previous sections, there are several methods that can be used to reduce this error.

The degree elevation method clearly has the greatest positive effect on the error rate, but it is not always advantageous to use it for other reasons, for example, if we want to keep the degree of 3 for later compatibility. That said, degree elevation is the method that can be clearly recommended if the precision of the affine transformation is absolutely important.

The method of transforming the domain of definition also reduces the error with good efficiency, the essence of which is that the basis functions are evaluated on integer values. Its advantage is that it keeps the degree at 3, and it is a very simple procedure. However, its disadvantage compared to the method that exploits the symmetry of the Bernstein–Bézier basis functions is that the formula and computation of the basis functions must be changed, while in most programs, this is carried out using a procedure imported from a standard CAD library.

Finally, exploiting the symmetry of the basis functions can also reduce the error. This method has the advantage of preserving the degree and allowing the standard Bernstein–Bézier function formula to be used. However, the disadvantage is that we have to be very careful in which direction we exploit this symmetry. The error occurs to a different extent on the two halves of the curve, so if we use the “worse” half, we will only worsen the error rate even more (see the case of $s^{\uparrow}$ in the results). If, on the other hand, we use the “better” half of the curve, then this method can also reduce the error (see $s^{\downarrow}$).

The above methods can of course be easily extended to higher-degree Bézier curves, and theoretically, they can also work for other types of spline curves. However, a higher degree also makes the Bézier curve more rigid and less flexible to design. Therefore, increasing the degree to above 4 is not recommended and was not included in the detailed analysis, although it does increase the accuracy somewhat further. The extent of this work to other curve types and to surfaces may form the basis of future research.

Author Contributions: Conceptualization, M.H. and E.T.; methodology, M.H. and E.T.; software, E.T.; validation, E.T.; statistical analysis, M.H. and E.T.; writing—original draft preparation, M.H. and E.T.; writing—review and editing, M.H.; visualization, E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable, since the study did not involve humans or animals.

Informed Consent Statement: Not applicable, since the study did not involve humans.

Data Availability Statement: The raw dataset supporting the conclusions of this article will be made available by the authors upon request.

Acknowledgments: The authors thank the anonymous reviewers for their helpful comments that improved the quality of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CAD	Computer-Aided Design
CAGD	Computer-Aided Geometric Design
NURBS	Non-Uniform Rational B-Spline

References

- Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*; Elsevier: Amsterdam, The Netherlands, 2014.
- Piegl, L.; Tiller, W. *The NURBS Book*; Springer: Berlin/Heidelberg, Germany, 2012.
- Yan, L. Evaluation Algorithms for Parametric Curves and Surfaces. *Mathematics* **2025**, *13*, 2248.
- Rashidinia, J.; Sharifi, S. Survey of B-spline functions to approximate the solution of mathematical problems. *Math. Sci.* **2012**, *6*, 48.
- Beccari, C.V.; Casciola, G. Stable numerical evaluation of multi-degree B-splines. *J. Comput. Appl. Math.* **2022**, *400*, 113743.
- Delgado, J.; Mainar, E.; Peña, J.M. On the accuracy of de Casteljau-type algorithms and Bernstein representations. *Comput. Aided Geom. Des.* **2023**, *106*, 102243.
- Fuda, C.; Ramanantoanina, A.; Hormann, K. A comprehensive comparison of algorithms for evaluating rational Bézier curves. *Dolomites Res. Notes Approx.* **2024**, *17*, 56–79.
- Hermes, D. Compensated de Casteljau algorithm in K times the working precision. *Appl. Math. Comput.* **2019**, *357*, 57–74.
- Simpson, R.N.; Scott, M.A.; Taus, M.; Thomas, D.C.; Lian, H. Acoustic isogeometric boundary element analysis. *Comput. Methods Appl. Mech. Eng.* **2014**, *269*, 265–290. <https://doi.org/10.1016/j.cma.2013.10.026>.
- Desiderio, L.; D’Inverno, G.A.; Sampoli, M.L.; Sestini, A. Hierarchical matrices for 3D Helmholtz problems in the multi-patch IgA-BEM setting. *Eng. Comput.* **2025**, *41*, 2021–2042. <https://doi.org/10.1007/s00366-025-02144-w>.
- Sapiro, G.; Tannenbaum, A. Affine invariant scale-space. *Int. J. Comput. Vis.* **1993**, *11*, 25–44.
- Kyratsis, P.; Tzotzis, A.; Davim, J.P. A Review on CAD-Based Programming for Design and Manufacturing. In *CAD-based Programming for Design and Manufacturing*; Springer Nature: Cham, Switzerland, 2025; pp. 1–26.
- Farin, G. Class a Bézier curves. *Comput. Aided Geom. Des.* **2006**, *23*, 573–581.
- Ainsworth, M.; Andriamaro, G.; Davydov, O. Bernstein–Bézier finite elements of arbitrary order and optimal assembly procedures. *SIAM J. Sci. Comput.* **2011**, *33*, 3087–3109.
- Bhatti, M.I.; Bracken, P. Solutions of differential equations in a Bernstein polynomial basis. *J. Comput. Appl. Math.* **2007**, *205*, 272–280.
- Zhu, C.G.; Kang, W.S. Numerical solution of Burgers–Fisher equation by cubic B-spline quasi-interpolation. *Appl. Math. Comput.* **2010**, *216*, 2679–2686.
- Hughes, T.J.; Cottrell, J.A.; Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 4135–4195.
- Wang, Y.; Teoh, E.K. 2D affine-invariant contour matching using B-spline model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1853–1858.
- Tarsissi, L.; Kenmochi, Y.; Djerroumi, H.; Coeurjolly, D.; Romon, P.; Borel, J.P. Algorithms for pixelwise shape deformations preserving digital convexity. In *Proceedings of the International Conference on Discrete Geometry and Mathematical Morphology*; Springer: Cham, Switzerland, 2022; pp. 84–97.
- Tarsissi, L.; Kenmochi, Y.; Romon, P.; Coeurjolly, D.; Borel, J.P. Convexity preserving deformations of digital sets: Characterization of removable and insertable pixels. *Discret. Appl. Math.* **2023**, *341*, 270–289.
- Wakjira, Y.; Kurukkal, N.S.; Lemu, H.G. Reverse engineering in medical application: Literature review, proof of concept and future perspectives. *Sci. Rep.* **2024**, *14*, 23621.
- Leordean, D.; Radu, S.; Frătilă, D.; Berce, P. Studies on design of customized orthopedic endoprostheses of titanium alloy manufactured by SLM. *Int. J. Adv. Manuf. Technol.* **2015**, *79*, 905–920.
- Gumiel, Y.; Rudek, M.; Bichinho, G.L.; Da Silveira, M.; Canciglieri, O. Anatomic prosthesis modelling based on Descriptors by Cubic Bézier Curves. In *Proceedings of the 45th International Conference on Computers & Industrial Engineering*, Metz, France, 28–30 October 2015.
- Thiselton, J.; Hanekom, T. Parameterisation and prediction of intra-canal cochlear structures. *Ann. Biomed. Eng.* **2024**, *52*, 695–706.
- Song, J.; You, W.; Shi, S.; Guo, S.; Sun, L.; Wang, W. Efficient and Scalable Chinese Vector Font Generation via Component Composition. *arXiv* **2024**, arXiv:2404.06779. <https://doi.org/10.48550/arXiv.2404.06779>.

26. Dube, M.; Gupta, N. Font Design Through RQT Bézier Curve Using Shape Parameters. In *Proceedings of the Advances in Mathematical Modelling, Applied Analysis and Computation*; Singh, J., Anastassiou, G.A., Baleanu, D., Cattani, C., Kumar, D., Eds.; Springer: Singapore, 2023; pp. 319–335.
27. He, Y.; Kang, S.H.; Morel, J.M. Silhouette vectorization by affine scale-space. *J. Math. Imaging Vis.* **2022**, *64*, 41–56.
28. IEEE Std 754-2008. IEEE Standard for Floating-Point Arithmetic. **2008**, 1–70.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.