

Topological Integrity for Dynamic Spline Models During Visualization of Big Data *

Hugh P. Cassidy, Thomas J. Peters, Horea Ilies, and Kirk E. Jordan

Abstract In computer graphics and scientific visualization, B-splines are common geometric representations. A typical display method is to render a piecewise linear (PL) approximation that lies within a prescribed tolerance of the curve. In dynamic applications it is necessary to perturb specified points on the displayed curve. The distance between the perturbed PL structure and the perturbed curve it represents can change significantly, possibly changing the underlying topology and introducing unwanted artifacts to the display. We give a strategy to perturb the curve smoothly and keep track of the error introduced by perturbations. This allows us to refine the PL curve when appropriate and avoid spurious topological changes. This work is motivated by applications to visualization of Big Data from simulations on high performance computing architectures.

1 Introduction

In geometric modeling B-splines are frequently used to model complex geometric objects [5]. The spline models are smooth structures but PL approximations are typically used to render the spline. Aeronautical, automotive and chemical simulations rely on topological algorithms to provide mathematically correct visualization. These topological algorithms typically enforce that the display curve (i.e. the PL structure) will preserve crucial topological characteristics [3, 14]. A sufficiently refined PL model will preserve topological characteristics of the initial static model. But as points on the PL model are perturbed over the course of the simulation, the

Hugh P. Cassidy, Thomas J. Peters and Horea Ilies
University of Connecticut, Storrs, CT

Kirk E. Jordan
T.J. Watson Research Center, IBM, Cambridge, MA

* The three UConn authors were partially supported by NSF grants CMMI 1053077 and CNS 0923158. T. J. Peters was also partially supported by an IBM Faculty Award and IBM Doctoral Fellowships. All statements here are the responsibility of the author, not of the National Science Foundation nor of IBM.

PL model may diverge significantly from the smooth model that it represents. This may introduce topological artifacts to the display, resulting in a flawed image that could mislead domain scientists.

Our formal analysis is motivated by graphics experiments, which are summarized in Experiment 1 (Section 3). We observed that the PL approximation used for graphics could be perturbed for more time steps, *while still preserving ambient isotopic equivalence* than might be expected from previously published bounds [8]. This data-specific *a posteriori* analysis led us to question whether we could develop *rigorous, predictive* methods for the permissible number of time steps. A method based upon second centered differences is developed for that predictive capability to support efficient frame generation, where this new method is motivated by Experiment 1, with a formal analysis in Example 4.

Many perturbation strategies are possible, but in dynamic visualization, retaining differentiability over time is often desirable, so our predictive method is presented in the context of a representative differentiable perturbation strategy. However, the formal analysis is quite general, and other perturbation strategies could easily be integrated by a user interested in other applications. Our exposition first uses a non-differentiable strategy to introduce some central concepts within this simplified context, but the ensuing differentiable strategy is then used in the rest of the development. Our distinctive contributions are analyses of the amount of error introduced by each perturbation. This error can be monitored and the PL model can be refined as necessary to avoid unwanted topological changes. For ease of notation the investigation below is performed on Bézier curves, however the analysis is identical for general B-spline curves [5]. The motivating graphics experiments are summarized in Section 3 and a representative analysis is presented as Example 4 in Section 7.

2 Background, Motivation and Notation

In this section we introduce some fundamental definitions and notation.

2.1 Curves and control polygons

Definition 1. A degree d Bézier curve with control points $X = \{q_0, \dots, q_d\}$ is given by

$$c(t) = \sum_{i=0}^d \binom{d}{i} (i-t)^{d-i} q_i$$

where the PL curve connecting q_0, \dots, q_d is called the control polygon of c .

A subdivision algorithm operates on X to generate two PL curves, each having $d+1$ vertices, denoted, respectively as X_L and X_R , as shown in Figure 1. The union $X_L \cup X_R$ is also a control polygon for c but lies closer to c than the original control polygon. This process can be repeated to obtain a PL graphical approximation that is within a prescribed distance, ϵ_d , of the curve c .

Definition 2. Given the polygon generated by $X = \{q_0, \dots, q_d\}$, the second centered difference of a given control point q_i is defined as

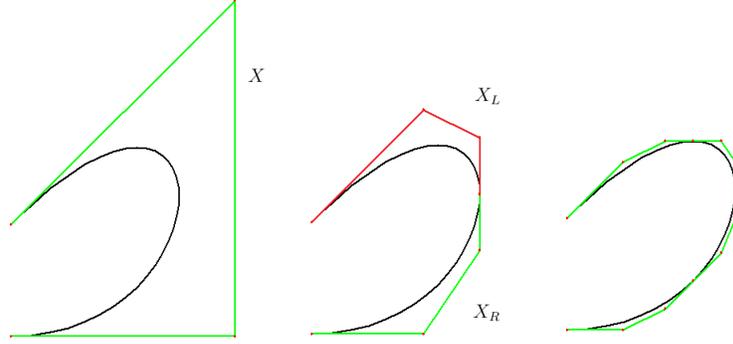


Fig. 1 Subdivision produces refined PL approximation.

$$\Delta_2 q_i = q_{i-1} - 2q_i + q_{i+1}.$$

We define $\Delta_2 q_0 = \Delta_2 q_d = 0$. The *maximal second centered difference* of the polygon generated by X is given by

$$\|\Delta_2 X\|_\infty = \max_{0 \leq i \leq d} \|\Delta_2 q_i\|.$$

Given a degree d curve with control points $X = \{q_0, \dots, q_d\}$, after α uniform subdivisions the maximal distance between the control polygon and the curve is given by [15]

$$\left(\frac{1}{2}\right)^{2\alpha} \|\Delta_2 X\|_\infty N_\infty(d).$$

Here $N_\infty(d) = \frac{\lceil d/2 \rceil \lfloor d/2 \rfloor}{2d}$. Note that this distance is actually attained [15]. So subdividing α times guarantees that the PL structure is within the specified tolerance for display, ε_d , where

$$\alpha = \left\lceil -\frac{1}{2} \log_2 \left(\frac{\varepsilon_d}{\|\Delta_2 X\|_\infty N_\infty(d)} \right) \right\rceil$$

2.2 Equivalence relation

The traditional measure of topological equivalence is homeomorphism. Homeomorphic equivalence does not capture the embedding of a curve within \mathbb{R}^3 , for example, all simple closed curves are homeomorphic even though there can be fundamentally different embeddings.

We use the stronger equivalence of *ambient isotopy* to also preserve embedding of c in \mathbb{R}^3 . Different knot types are not ambient isotopic.

Definition 3. Two subspaces, X and Y , of \mathbb{R}^n are said to be ambient isotopic if there exists a continuous function $H : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n$ such that

1. $H(\cdot, 0)$ is the identity on \mathbb{R}^n ,
2. $H(X, 1) = Y$, and
3. $\forall t \in [0, 1], H(\cdot, t)$ is a homeomorphism.

2.3 Related Work

Molecular simulations are run on high performance computing (HPC) architectures, often generating petabytes of data, initiating a typical ‘Big Data’ problem. This data output is too voluminous for standard numerical analytic techniques and dynamic visualization has become a common zero-th order analysis. The supportive dynamic visualization techniques are well-established [12, 13] and will not be addressed further. The vitally important and novel support from this work is to provide rigorously proven numerical assurances that the frames being viewed have appropriate approximation in order to avoid topological artifacts in the images that could prove misleading to the domain scientists [9, 4]. To establish context for this work, a brief overview will be given of the three primary facets of *supportive mathematics, geometric models and molecular simulations*. The emphasis here is upon the new mathematics to meet the new Big Data challenges posed by the recent prevalence of these petabytes of simulation output, where this emerging mathematics is developing a blend of theory and experimentation.

At the highest level of viewing this work, there are so many tools available for molecular visualization, that it suffices to provide two broad summary portals [12, 13]. Often protein data is of interest, which appears publicly in an international resource [2]. The indicated resources do not directly provide geometric models of the molecules visualized – only images are produced.

The molecular simulation research [16, 21, 19, 20, 18] closely aligns with the work presented here, with [21] being of particular interest because of its use of splines to model molecules, as also assumed here. Alternate geometric representations have been considered [10, 11, 17] for molecules, but the choice of splines here is offered as a very broad, fundamental representation, which could be examined for adaptation to these alternate representations. The more contemporary Big Data issues had not yet appeared when this earlier work had already been completed.

The emphasis here upon geometric representations echoes much work in computer-aided geometric design [5]. In particular, this dynamic molecular visualization has been synergistically pursued with an emerging virtual reality (VR) engineering design laboratory [7]. A fascinating common use is of 1-dimensional geometry to model the molecule writhing proteins and design features [6], where the latter application is integrated with a constraint solver.

Motivating Applications The mathematics proven here was motivated by design of dynamic visualization for molecular simulations in HPC. As a zero-th order analysis, a dynamic visualization is synchronized with the ongoing simulation. The graphics at each frame are displayed by PL approximations, raising the possibility that an image could show an intersection on a writhing molecule where none occurs on the more accurate spline model. The isotopic analysis presented is designed to integrate the necessary numerical accuracy with sufficient performance for dynamic visualization. Subdivision is chosen for the PL approximation, but the analysis presented here could easily be adapted to other PL approximation techniques, such as PL interpolation through selected points on the curve. Proteins are typical objects of interest, modeled as spline curves. Public data bases [2] provide spatial co-ordinates for interpolation to create a spline model. However, there can easily be hundreds of thousands of such co-ordinates, so that interpolation by a single segment spline would be also be on the order of hundreds of thousands — typically prohibitive for

interactive graphics, where much lower degree is preferred (often as low as degree 3, but rarely higher than degree 8). Sufficiently accurate, low degree models can be created by the composite curves [5] used here. Since these geometric molecular models are not readily available in the public resources [2, 12, 13] prototype software is also being developed to provide those models, but reports on those tools will appear elsewhere.

The presented mathematical analysis has guided our algorithmic design so that we are now confident that we can use splines of sufficiently low degree, while maintaining desired topological characteristics. It remains to integrate these topology preserving techniques into the supporting dynamic visualizations discussed. That full experimental work is beyond the scope of the present paper and remains as subject for future publications.

3 Graphics Efficiency Experiment

The efficient use of PL approximations in dynamic visualizations has previously appeared [8], as a way to ensure correct graphics topology during animation, as previously presented relative to isotopic equivalence. That previous strategy [8] will now be briefly summarized, where this work adds the additional perspective of practical limits on the number of frames where this aggressive strategy can be invoked. As perspective on the extreme data and performance demands of this environment, it is instructive to note the order of 30 - 60 frames per second to synchronize dynamic visualization with a simulation producing peta-bytes of output.

During simulation, the molecule moves as reflected by movement of a spline. Each frame will use PL approximation. Here are two graphics display options to consider:

Option 1: At each time step, perturb the spline and create a new PL approximation for display.

Option 2: Create a PL approximation of the spline at some initial time step. Continue to perturb this PL approximation until it is no longer sufficiently accurate for graphics display.

Clearly, Option 2 can eliminate the approximation algorithm at some time steps. The previous work [8] provided existence theorems for maintaining isotopic equivalence during continued perturbation of these PL approximations. This work refines [8] by now providing specific numerical analyses to show *exactly* how many subsequent frames can invoke this aggressive strategy, *before* it becomes necessary to create a new PL approximation to ensure ongoing topological fidelity between the spline and its graphics approximation.

A representative graphics experiment will be summarized to show implications of Option 2. A sufficient² perturbation bound [8, Proposition 5.2] to preserve ambient isotopy is $(1/2)v$, with v defined as the minimal distance between points and edges of a PL curve [1]. With the control points here, we note that $(1/2)v = 1/2$. We will show, later, that this upper bound, while sufficient to preserve ambient isotopy, leaves open the possibility of more aggressive perturbation strategies.

² There is an obvious typographical error [8, Proposition 5.2].

Experiment 1 Consider the non-self intersecting C^1 composite cubic Bézier curve in \mathbb{R}^2 , as depicted on the left hand side of Figure 2. The following points together with their reflection through the line $y = 3$ form the control points:

$$(0, 6), (1, 5), (2, 4.5), (3, 5.25), (4, 6), (5, 7), (6, 8), (7, 9), (9, 10), (11, 11), (13, 12), (15, 13), \\ (17, 13.35), (19, 13.7), (21, 13), (22, 12), (23, 11), (24, 10), (24.5, 8), (25, 6), (25, 4), (25, 3).$$

The control polygon is green with red control points, the underlying curve is black. Perturbing p_u and p_v over ten time steps introduces a self intersection to the PL structure that is not present in the underlying spline curve, as illustrated on the right hand side of Figure 2. For brevity of presentation, the example of Figure 2 presents the graphics of the original and perturbed Bézier curves to show that both are non-self-intersecting, which can be rigorously verified [1]. We return to this example for a detailed analysis in Section 6.3.

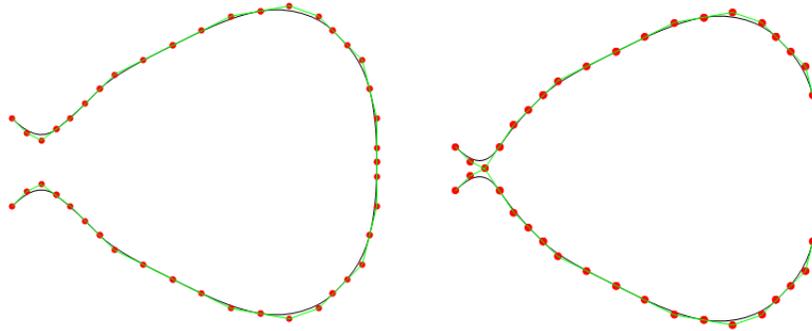


Fig. 2 Spurious self-intersection in PL structure.

We note that the previous bound with of $(1/2)v = 1/2$ would have guaranteed that the first 5 time steps were permissible. When these visual experiments showed that topological fidelity could be preserved until the 10th step, we pursued a deeper analysis to explicate identification of this longer preservation of topology.

4 Notation for perturbation analysis

We now define the notation required for the perturbation analysis.

We shall examine n time steps denoted $\{t_1, \dots, t_n\}$, t_0 denotes the time at initialization.

We assume that we are given a refined control polygon so that it is within ϵ_d of the represented curve. Note if α subdivisions are required then, from the original set of control points $\{q_0, \dots, q_d\}$, there are generated w control points where $w = 2^\alpha d + 1$. Denote the subdivided, *but unperturbed*, control polygon by

$$X_0 = \{p_0, p_1, \dots, p_w\}.$$

Let X_i denote the perturbed control polygon at time t_i . Assume we are supplied with a $(w + 1) \times n$ perturbation matrix, Γ , where each row contains perturbation vectors for a corresponding control point and each column contains the perturbation vectors for all control points at the corresponding time step, i.e.

$$\Gamma = \begin{matrix} & t_1 & t_2 & \dots & t_n \\ \begin{matrix} p_0 \\ p_1 \\ \vdots \\ p_w \end{matrix} & \begin{pmatrix} \gamma_{0,1} & \gamma_{0,2} & \dots & \gamma_{0,n} \\ \gamma_{1,1} & \gamma_{1,2} & \dots & \gamma_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{w,1} & \gamma_{w,2} & \dots & \gamma_{w,n} \end{pmatrix} \end{matrix}$$

where $\gamma_{i,j}$ denotes the perturbation vector applied to p_i at time t_j (may be the zero vector). Let $\delta_j p_i$ denote the coordinates of the point that originated at p_i at t_j , i.e.

$$\delta_j p_i = p_i + \sum_{k=1}^j \gamma_{i,k}.$$

5 Non-differentiable Perturbations

In cases where maintaining differentiability of the curve is not required, we may simply perturb each point by the prescribed vector. At t_0 we are given X_0 and Γ as described above. At each t_i we can calculate X_i from Γ and X_{i-1} .

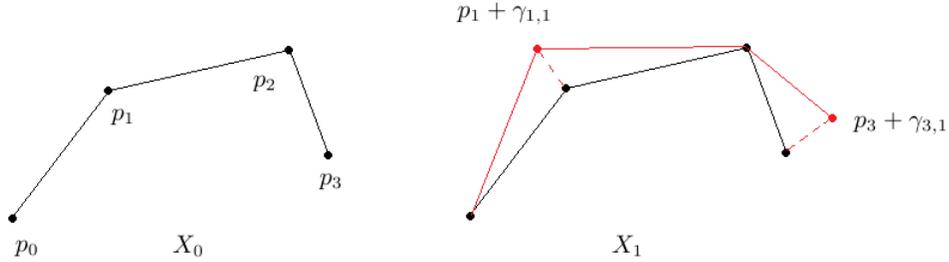


Fig. 3 Perturbation over a single time step.

Example 1.

Given the points $X_0 = \{p_0, p_1, p_2, p_3\}$ and the perturbation matrix,

$$\Gamma = \begin{bmatrix} 0 & 0 \\ \gamma_{1,1} & \gamma_{1,2} \\ 0 & 0 \\ \gamma_{3,1} & \gamma_{3,2} \end{bmatrix}$$

as depicted in Figure 3. We can calculate

$$X_1 = \{\delta_1 p_0, \delta_1 p_1, \delta_1 p_2, \delta_1 p_3\} = \{p_0, p_1 + \gamma_{1,1}, p_2, p_3 + \gamma_{3,1}\}, \text{ and}$$

$$X_2 = \{\delta_2 p_0, \delta_2 p_1, \delta_2 p_2, \delta_2 p_3\} = \{p_0, p_1 + \gamma_{1,1} + \gamma_{1,2}, p_2, p_3 + \gamma_{3,1} + \gamma_{3,2}\}.$$

5.1 Perturbing a single point

First we consider perturbing a single point over a single time step. At initialization we have

$$X_0 = \{p_0, \dots, p_j, \dots, p_w\}.$$

Note that

$$\|\Delta_2 X_0\|_{\infty} N_{\infty}(d) \leq \epsilon_d$$

Let p_j be the point being perturbed. At time t_1 the point p_j is perturbed to $p_j + \gamma_{j,1}$ and all other points remain in their original positions.

$$X_1 = \{p_0, \dots, \delta_1 p_j, \dots, p_w\} = \{p_0, \dots, p_j + \gamma_{j,1}, \dots, p_w\}$$

The only second differences affected are $\Delta_2(p_{j-1})$, $\Delta_2(p_j)$ and $\Delta_2(p_{j+1})$.

$$\|\Delta_2 X_1\|_{\infty} = \max\{\|\Delta_2 X_0\|_{\infty}, \|\Delta_2(\delta_1 p_{j-1})\|, \|\Delta_2(\delta_1 p_j)\|, \|\Delta_2(\delta_1 p_{j+1})\|\}$$

where

$$\begin{aligned} \Delta_2(\delta_1 p_{j-1}) &= p_{j-2} - p_{j-1} + p_j + \gamma_{j,1} = \Delta_2(p_{j-1}) + \gamma_{j,1}, \\ \Delta_2(\delta_1 p_j) &= \Delta_2(p_j) - 2\gamma_{j,1}, \text{ and} \\ \Delta_2(\delta_1 p_{j+1}) &= \Delta_2(p_{j+1}) + \gamma_{j,1}. \end{aligned}$$

This approach extends easily to n time steps

$$\|\Delta_2 X_n\|_{\infty} = \max\{\|\Delta_2 X_0\|_{\infty}, \|\Delta_2(\delta_n p_{j-1})\|, \|\Delta_2(\delta_n p_j)\|, \|\Delta_2(\delta_n p_{j+1})\|\}$$

where $\Delta_2(\delta_n p_{j-1}) = \Delta_2(p_{j-1}) + \sum_{i=1}^n \gamma_{j,i}$, $\Delta_2(\delta_n p_j) = \Delta_2(p_j) - 2\sum_{i=1}^n \gamma_{j,i}$ and $\Delta_2(\delta_n p_{j+1}) = \Delta_2(p_{j+1}) + \sum_{i=1}^n \gamma_{j,i}$.

5.2 Perturbing multiple points

To perturb multiple points over multiple time steps, using the information supplied by Γ , sort the points being perturbed into adjacency chains, i.e. sets of adjacent control points denoted Q_0, \dots, Q_s where each Q_i contains either a single point or a list of adjacent points to be perturbed. This is necessary as chains of different length have different effects on the second differences that involve points in that chain. Let $|Q_i| = u$. If $u = 1$ then this is treated as in the single point case above. If $u = 2$ then we write $Q_i = \{p_k, p_{k+1}\}$, and we compute the affected centered differences as follows:

$$\Delta_2(\delta_n p_{k-1}) = \Delta_2(p_{k-1}) + \sum_{j=1}^n \gamma_{k,j}$$

$$\begin{aligned}\Delta_2(\delta_n p_k) &= \Delta_2(p_k) + \sum_{j=1}^n (\gamma_{k+1,j} - 2\gamma_{k,j}) \\ \Delta_2(\delta_n p_{k+1}) &= \Delta_2(p_{k+1}) + \sum_{j=1}^n (\gamma_{k,j} - 2\gamma_{k+1,j}) \\ \Delta_2(\delta_n p_{k+2}) &= \Delta_2(p_{k+2}) + \sum_{j=1}^n \gamma_{k+1,j}\end{aligned}$$

If $u \geq 3$ then $Q_i = \{p_k, \dots, p_{k+v}\}$ for some $v \geq 2$. The affected centered differences are computed:

$$\begin{aligned}\Delta_2(\delta_n p_{k-1}) &= \Delta_2(p_{k-1}) + \sum_{j=1}^n \gamma_{k,j} \\ \Delta_2(\delta_n p_k) &= \Delta_2(p_k) + \sum_{j=1}^n (\gamma_{k+1,j} - 2\gamma_{k,j}) \\ &\vdots \\ \Delta_2(\delta_n p_s) &= \Delta_2(p_s) + \sum_{j=1}^n (\gamma_{s-1,j} - 2\gamma_{s,j} + \gamma_{s+1,j}) \\ &\vdots \\ \Delta_2(\delta_n p_{k+v}) &= \Delta_2(p_{k+v}) + \sum_{j=1}^n (\gamma_{k+v-1,j} - 2\gamma_{k+v,j}) \\ \Delta_2(\delta_n p_{k+v+1}) &= \Delta_2(p_{k+v+1}) + \sum_{j=1}^n \gamma_{k+v,j}\end{aligned}$$

6 Differentiable Perturbations

It may be desirable to maintain a degree of differentiability either for appearances, analysis or both. We define a perturbation strategy that guarantees C^1 continuity (assuming the original curve is at least C^1).

6.1 Perturbation strategy

We are given a composite Bézier curve to perturb. Recall that a *junction point* is a point where curve segments meet. We identify three types of point:

- *Type 1*: A point adjacent to a junction point.
- *Type 2*: A junction point.
- *Type 3*: A point that is neither a junction point nor adjacent to a junction point.

To maintain C^1 continuity we must require that the tangent edges with a shared junction point be collinear and have the same length [5].

Type 1

If we perturb a type 1 point in order to satisfy the C^1 criteria we perturb the junction point to the midpoint of the line segment joining its adjacent points. This approach is illustrated in the following example, where p_2 is being perturbed, relative to the junction point of p_3 .

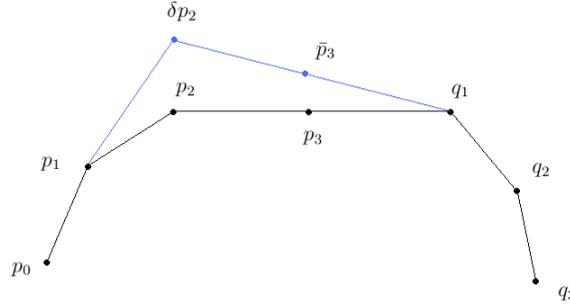


Fig. 4 Type 1.

Example 2. Given a composite cubic control polygon with sub polygons $\{p_0, p_1, p_2, p_3\}$ and $\{p_3, q_1, q_2, q_3\}$ as depicted in Figure 4. If we perturb p_2 by a vector γ :

$$p_2 \rightarrow \delta p_2 = p_2 + \gamma,$$

then to maintain C^1 continuity we perturb p_3 as follows:

$$p_3 \rightarrow \bar{p}_3 = \frac{\delta p_2 + q_1}{2}.$$

Type 2

To maintain C^1 differentiability when perturbing a type 2 point we must also perturb its adjacent points by the same vector so the tangent edges are collinear and have the same length and are collinear.

Example 3. Here we have a composite cubic control polygon with sub polygons $\{p_0, p_1, p_2, p_3\}$ and $\{p_3, q_1, q_2, q_3\}$ as shown in Figure 5. Perturbing p_3 by γ has the following effect:

$$\begin{aligned} p_2 &\rightarrow \delta p_2 = p_2 + \gamma, \\ p_3 &\rightarrow \delta p_3 = p_3 + \gamma, \\ q_1 &\rightarrow \delta q_1 = q_1 + \gamma. \end{aligned}$$

Type 3

Since Type 3 points do not effect tangent edges we can just perturb them as normal without perturbing neighboring points.

p_{j-1}, p_j and p_{j+1} are each perturbed by $\gamma_{j,i}$ at time t_i , $i \in \{1, \dots, n\}$.
At time t_n we have

$$\|\Delta_2 X_n\|_\infty = \max\{\|\Delta_2 X_0\|_\infty, \max\{\|\Delta_2(\delta_n p_k)\|\}_{k=j-2}^{j+2}\}.$$

The changes to the second centered differences are as follows:

$$\begin{aligned}\Delta_2(\delta_n p_{j-2}) &= \Delta_2(p_{j-2}) + \sum_{k=1}^n \gamma_{j,k} \\ \Delta_2(\delta_n p_{j-1}) &= \Delta_2(p_{j-1}) - \sum_{k=1}^n \gamma_{j,k} \\ \Delta_2(\delta_n p_j) &= 0 \\ \Delta_2(\delta_n p_{j+1}) &= \Delta_2(p_{j+1}) - \sum_{k=1}^n \gamma_{j,k} \\ \Delta_2(\delta_n p_{j+2}) &= \Delta_2(p_{j+2}) + \sum_{k=1}^n \gamma_{j,k}\end{aligned}$$

Type 3. If we are perturbing a type 3 point then at time t_n we have

$$\|\Delta_2 X_n\|_\infty = \max\{\|\Delta_2 X_0\|_\infty, \|\Delta_2(\delta_n p_{j-1})\|, \|\Delta_2(\delta_n p_j)\|, \|\Delta_2(\delta_n p_{j+1})\|\},$$

with the changes in second centered differences:

$$\begin{aligned}\|\Delta_2(\delta_n p_{j-1})\|_\infty &= \|\Delta_2(p_{j-1}) + \sum_{i=1}^n \gamma_{j,i}\|_\infty \\ \|\Delta_2(\delta_n p_j)\|_\infty &= \|\Delta_2(p_j) - 2 \sum_{i=1}^n \gamma_{j,i}\|_\infty \\ \|\Delta_2(\delta_n p_{j+1})\|_\infty &= \|\Delta_2(p_{j+1}) + \sum_{i=1}^n \gamma_{j,i}\|_\infty\end{aligned}$$

6.3 Perturbing multiple points

Let p_j and p_{j+2} be type 1 point, so p_j is a type 2. We consider the illustrative case where p_j, p_{j+1} and p_{j+2} are each being perturbed over n time steps:

$$\begin{aligned}p_j &\rightarrow p_j + \sum_{k=1}^n (\gamma_{j,k} + \gamma_{j+1,k}) \\ p_{j+1} &\rightarrow \frac{1}{2} \left(p_j + p_{j+2} + \sum_{k=1}^n (\gamma_{j,k} + 2\gamma_{j+1,k} + \gamma_{j+2,k}) \right) \\ p_{j+2} &\rightarrow p_{j+2} + \sum_{k=1}^n (\gamma_{j+1,k} + \gamma_{j+2,k})\end{aligned}$$

The effect on the second differences is as follows:

$$\begin{aligned}\Delta_2(\delta_n p_{j-1}) &= \Delta_2 p_{j-1} + \sum_{k=1}^n (\gamma_{j,k} + \gamma_{j+1,k}) \\ \Delta_2(\delta_n p_j) &= p_{j-1} - \frac{3}{2} p_j + \frac{1}{2} p_{j+2} + \sum_{k=0}^n \left(-\frac{3}{2} \gamma_{j,k} - \gamma_{j+1,k} + \frac{1}{2} \gamma_{j+2,k} \right) \\ \Delta_2(\delta_n p_{j+1}) &= 0 \\ \Delta_2(\delta_n p_{j+2}) &= \frac{1}{2} p_j - \frac{3}{2} p_{j+2} + p_{j+3} + \sum_{k=1}^n \left(\frac{1}{2} \gamma_{j,k} - \gamma_{j+1,k} - \frac{3}{2} \gamma_{j+2,k} \right) \\ \Delta_2(\delta_n p_{j+3}) &= \Delta_2 p_{j+3} + \sum_{k=1}^n (\gamma_{j+1,k} + \gamma_{j+2,k})\end{aligned}$$

7 An Example Predictive Analysis

Our predictive method is now applied to formalize the empirical observations of Experiment 1, explicating extensions beyond previous bounds [8].

Example 4. The cubic Bézier curve of Example 1 was specifically synthesized to permit more aggressive PL graphics perturbations than previously known [8]. Given control points $X_0 = \{(0, 6), (1, 5), (2, 4.5), \dots, (2, 1.5), (1, 1), (0, 0)\}$. Denote $\{(0, 6), (1, 5), (2, 4.5)\}$ by $U = \{p_{u-2}, p_{u-1}, p_u\}$ and $\{(2, 1.5), (1, 1), (0, 0)\}$ by $V = \{p_v, p_{v+1}, p_{v+2}\}$. Let the display tolerance, $\varepsilon_d = 1.9167$. The maximal distance between the curve and the control polygon is $5/12$ which we trivially note is less than the given ε_d . Say we wish to perturb the points in U and V over 10 time steps with perturbation vectors $\{\gamma_{u-2,k}\}_{k=1}^{10} = \{\gamma_{u-1,k}\}_{k=1}^{10} = \{\gamma_{u,k}\}_{k=1}^{10}$ and $\{\gamma_{v,k}\}_{k=1}^{10} = \{\gamma_{v+1,k}\}_{k=1}^{10} = \{\gamma_{v+2,k}\}_{k=1}^{10}$ where

$$\begin{aligned}\{\gamma_{u,k}\}_{k=1}^{10} &= \{ (0, 5/20), (0, 4/20), (0, 4/20), (0, 4/20), (0, 3/20), \\ &\quad (0, 2/20), (0, 1/20), (0, 1/20), (0, 1/20), (0, 5/20) \}, \text{ and} \\ \{\gamma_{v,k}\}_{k=1}^{10} &= \{ (0, -5/20), (0, -4/20), (0, -4/20), (0, -4/20), (0, -3/20), \\ &\quad (0, -2/20), (0, -1/20), (0, -1/20), (0, -1/20), (0, -5/20) \}\end{aligned}$$

For this curve, $1/2v = 1/2$, a value which is clearly exceeded after 5 steps of this strategy. Since previous criteria [8] were only sufficient, the rest of this example demonstrates that greater perturbation is possible to support efficiency in Strategy 2. Since the analysis for points in U and V is identical we shall focus on V . Notice that

$$\sum_{k=1}^{10} \gamma_{v,k} = \left(0, \frac{3}{2} \right).$$

Since p_v is a type 2 point, the junction point $p_j = (3, 3/4)$ will also be perturbed as described above. Denote the control point following p_j by p_{j+1} .

$$p_v = (2, 3/2) \rightarrow \delta_{10} p_v = (2, 3)$$

$$p_j = (3, 3/4) \rightarrow \delta_{10} p_j = (3, 3/2)$$

We require that for each i ,

$$\|\Delta_2 X_i\|_\infty + \left\| \sum_{j=1}^i \gamma_{X,j} \right\| < \epsilon_d.$$

Here $\sum \gamma_{X,j}$ is the sum of the perturbation vectors applied to the control point that yields $\|\Delta_2 X_i\|_\infty$. These quantities are easily calculated using the analysis above. It is easy to see that $\|\Delta_2 X_i\|_\infty + \left\| \sum_{j=1}^i \gamma_{X,j} \right\| < \epsilon_d$ for $i = 1, \dots, 9$. At the ninth time step we have

$$\|\Delta_2 X_9\|_\infty + \left\| \sum_{j=1}^9 \gamma_{X,j} \right\| = 0.65 + 1.25 = 1.9 < \epsilon_d$$

At the tenth time step

$$\|\Delta_2 X_{10}\|_\infty + \left\| \sum_{j=1}^{10} \gamma_{X,j} \right\| = 0.667 + 1.5 = 2.167 > \epsilon_d$$

Observing this we are now aware of the need to refine the control polygon by subdivision. Note that [3] and [14] allow us to determine the amount of subdivision required so that an ambient isotopic approximation is guaranteed

8 Conclusions and Future Work

For dynamic visualization of molecular simulations it is important to ensure that the rendered curve and the underlying spline are ambient isotopic at each time step. That global bounds on these perturbations can be exceeded if only local perturbations are executed is obvious, the performance imperatives for dynamic visualization make such data-specific refinements relevant, as is explored here. This can be achieved by keeping track of changes to the second centered differences and applying further subdivision as required.

The above analysis was performed for B-spline curves, the surface case was not pursued but we expect that the results can be extended to B-spline surfaces easily.

The molecules modeled certainly have 3-dimensional structure that is not captured by the 1-dimensional spline models. The reduction in dimension was chosen to support the performance demands of dynamic visualization of an ongoing simulation producing peta-bytes of output, while still being able to capture essential topological characteristics needed for zero-th order analyses. A similar reduction of dimension was undertaken to simplify engineering design studies [6]. The user identifies boundaries, that are modeled as 1-dimensional curves, as abstractions to convey design intent. This low-order geometry affords interactive manipulation and constraint satisfaction. Emerging VR techniques rely upon hand and finger gestures to express design variations. It would be desirable to adapt such gestures to interactive steering of these molecular simulations, providing further opportunities to share these research perspectives. Indeed, some of the required emphasis on graphics manipulation is being pursued, concurrently, under associate technology transfer projects [7] for gesture based editing during production of computer animations in the film making industry. An ideal outcome would be the effective merging from

these three fronts of molecular simulation, engineering design and film making – as remains the subject of planned activities.

Acknowledgements

The authors thank the referees, both for the conference presentation, as well as for this final book, for their helpful and constructive comments, which led to many improvements.

References

1. Lars-Erik Andersson, Thomas J. Peters, Neil F. Stewart, and S. M. Doney. Polyhedral perturbations that preserve topological form. *Computer Aided Geometric Design*, 12(8):785–799, 1995.
2. Anonymous. The protein data bank. <http://www.rcsb.org/pdb/home/home.do>, 2013.
3. H.P. Cassidy, T.J. Peters, and K.E. Jordan. Dynamic computational topology for piecewise linear curves. In *Proceedings of the Canadian Conference on Computational Geometry 2012*, pages 279 – 284, Charlottetown, P.E.I., August 8 - 10, 2012 2012.
4. Tiago Etienne, Luis Gustavo Nonato, Carlos Eduardo Scheidegger, Julien Tierny, Thomas J. Peters, Valerio Pascucci, Robert M. Kirby, and Cláudio T. Silva. Topology verification for isosurface extraction. *IEEE Trans. Vis. Comput. Graph.*, 18(6):952–965, 2012.
5. G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practicle Guide*. Academic Press, San Diego, CA, 2nd edition, 1990.
6. Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. iwires: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph.*, 28(3):33:1–33:10, July 2009.
7. H. Iliès. MRI: Development of a gesture based virtual reality system for research in virtual worlds. NSF award 0923158. <http://www.nsf.gov>, 2009.
8. K. E. Jordan, L. E. Miller, E. L. F. Moore, T. J. Peters, and A. C. Russell. Modeling time and topology for animation and visualization with examples on parametric geometry. *Theoretical Computer Science*, 405:41–49, 2008.
9. Robert M. Kirby and Cláudio T. Silva. The need for verifiable visualization. *IEEE Computer Graphics and Applications*, 28(5):78–83, 2008.
10. R. Kolodny, L. Guibas, M. Levitt, and P. Koehl. Inverse kinematics in biology: The protein loop closure problem. *Jour. Robotics Research*, 24:151–162, 2005.
11. R. Kolodny, P. Koehl, L. Guibas, and M. Levitt. Small libraries of protein fragments model native protein structures accurately. *Journal of Molecular Biology*, 323:297–307, 2002.
12. Eric Martz and Trevor D. Kramer. World index of molecular visualization resources.
13. Eric Martz and Trevor D. Kramer. Molviz. <http://molviz.sdsc.edu/>, 2013.
14. L. Miller, E.L.F. Moore, T.J. Peters, and A.C. Russell. Topological neighborhoods for spline curves : Practice and theory. In P. Hertling et al., editors, *Reliable Implementation of Real Number Algorithms: Theory and Practice*, volume 5045 of *LNCS*, pages 149–161. Springer, 2008.
15. D. Nairn, J. Peters, and D. Lutterkort. Sharp, quantitative bounds on the distance between a polynomial piece and its Bézier control polygon. *Computer Aided Geometric Design*, 16:613–631, 1999.
16. Gomathi Ramachandran and Tamar Schlick. Solvent effects on supercoiled DNA dynamics explored by Langevin dynamics simulations. *Phys. Rev. E*, 51(6):6188–6203, Jun 1995.
17. Daniel Russel and Leonidas Guibas. Exploring protein folding conformations using spanners. In *Pacific Symposium on Biocomputing*, pages 40–51, 2005.
18. T. Schlick. Dynamic simulations of biomolecules. <http://www.searlescholars.net/people/1991/schlick.html>.
19. T. Schlick. Modeling superhelical DNA: recent analytical and dynamic approaches. *Current Opinion in Structural Biology*, 5:245–262, 1995.
20. T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, New York, 2002.
21. T. Schlick and Wilma K. Olson. Trefoil knotting revealed by molecular dynamics simulations of supercoiled DNA. *Science*, 21:1110 – 1115, 1992.