Fast L1-Ck polynomial spline interpolation algorithm with shape-preserving properties

Eric Nyiri, Olivier Gibaru, Philippe Auquiert

Fast $L_1 - C^k$ polynomial spline interpolation algorithm with shape-preserving properties

Eric Nyiri[1], Olivier Gibaru[1,2], Philippe Auquiert[1]

[1] *Arts et Metiers ParisTech, L2MA, 8 Boulevard Louis XIV, 59046 Lille Cedex*
[2] *INRIA Lille-Nord-Europe, 40, avenue Halley 59650 Villeneuve d'Ascq, France*

Abstract

In this article, we address the interpolation problem of data points per regular L_1-spline polynomial curve that is invariant under a rotation of the data. We iteratively apply a minimization method on five data, belonging to a sliding window, in order to obtain this interpolating curve. We even show in the C^k-continuous interpolation case that this local minimization method preserves well the linear parts of the data, while a global L_p ($p \geq 1$) minimization method does not in general satisfy this property. In addition, the complexity of the calculations of the unknown derivatives is a linear function of the length of the data whatever the order of smoothness of the curve.

Key words: L_1 spline, interpolation, shape preserving, smooth spline

Introduction

In geometric modelling, a common requirement is that the computational curves 'preserve shape', which means the curves express the geometric properties of the interpolated data in accordance with human perception. These geometric properties are variously interpreted as linearity, monotonicity, convexity and smoothness. Conventional splines, which are calculated by minimizing the square of the L_2 norm of the second partial derivatives of a cubic piecewise polynomial interpolant, represent sufficiently "smooth" data quite well. However, they often have extraneous, nonphysical oscillations when used for interpolation of data with abrupt changes.

Recently, a new kind of splines called cubic L_1 splines has arisen (Cf. [5], [6], [7], [9], [15], [17]). Cubic L_1 splines, which are calculated by minimizing the L_1 norm of the second derivatives of a C^1-smooth piecewise cubic interpolant, 'preserve the shape' of data with abrupt changes (Cf. [14], [16]). In [1] and [3],

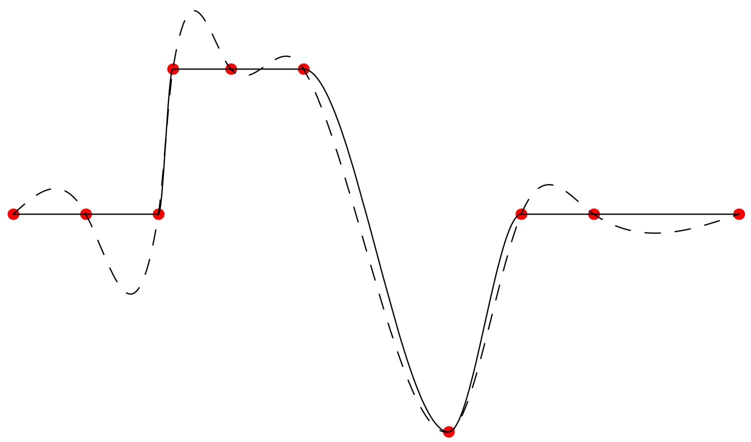


Fig. 1. L_2 (dotted line) versus L_1(solid line) global interpolations

we show that the univariate interpolating cubic L_1 spline of a set of points lying over a Heaviside function entirely agrees with the function (i.e the two half lines) except at the jump.

Although many different interpretations of shape preservation can be found in literature (Cf. [13], [23], [24]), there is no widely accepted quantitative description of shape preservation. In the present paper, we accept the observation made by most observers that L_1 splines preserve shape well as justification for working on improving the algorithm to calculate L_1 splines and the shape preserving properties (Cf Fig. 1).

The L_1 spline is issued from the minimization of a nonlinear functional. Since nonlinear programming procedures for minimizing the L_1 spline functional are not yet practical for global data interpolation, a discretization of this functional is commonly used. Minimization of the discretized L_1 spline functional which is a nonsmooth convex programming problem, leads to solving of an overdetermined linear system that can be reduced to a linear program for which many methods are available. In literature, a compressed primal affine method has been the most common choice. This method is based on the primal affine algorithm by Vanderbei, Meketon and Freedman (Cf. [27], [28], [25], [26]) and it is described in [16]. The primal-dual algorithm [17,20,24,28] is widely considered to be the most efficient and robust interior-point method.

We introduce a local minimization method based on a sliding window defined over five points so as to interpolate data points smoothly. This local method allows us to define a fast computational algorithm issued from the algebraic calculus of the exact solution over five points only. Furthermore, we show that a five points window allows us to preserve the linear parts of the data points while in general the global method does not satisfy this property. Moreover, we show that if we apply this strategy iteratively we can define C^k-continuous spline curves with a linear complexity of the calculations. The spline curve solutions also have shape-preserving properties.

This paper is organized as follows. In Section 1, we give some results concerning C^1-continuous cubic L_1 spline interpolation on three and five points. Based on these results, in Section 2 we define a new interpolation strategy with a sliding five points window to create a local L_1C^1 interpolating method. By applying iteratively this method we are able to construct C^k-continuous interpolating spline curves in $\mathbb{R}^d$(with $d \geq 1$). In Section 3, we demonstrate that the linear parts of the data are preserved and we also give some other properties. Some conclusions will be drawn in the last section.

1 The C^1-continuous cubic L_1 spline interpolation on five points

Let $a = u_1 < u_2 <\cdots< u_n = b$ be an arbitrary and strictly monotonic partition of the finite real interval $[a, b]$. In [2], we deal with the parametric case where we wish to interpolate a set of data points $P_1, \ldots, P_n$ belonging to $\mathbb{R}^d$ (with $d \geq 1$). The u_i are chosen according to the classical chordal partition (see [12] Section 4.4.1 page 201). This choice seems to give the best results in most data configurations. The C^1 interpolating cubic spline curve is calculated by minimizing the L_1-norm of the second derivative vector of the spline. If we denote by Δ the classical forward difference operator, we showed in [2] that the solution to this problem is obtained by minimizing the following functional

$$E(T_1, \ldots, T_n) = \sum_{i=1}^{n-1} \int_{-\frac{1}{2}}^{\frac{1}{2}} \|\Delta T_i + 6t(T_{i+1} + T_i - \frac{2}{\Delta u_i}\Delta P_i)\|_1 dt \tag{1}$$

where the $T_i \in \mathbb{R}^d$ are the first order derivative vectors at points P_i for $i = 1, \ldots, n$. As $E(T_1, \ldots, T_n)$ is not strictly convex, then its minima are not necessarily unique. To reduce the set of solutions, Lavery in [14] added a 'regularization'term so as to select the derivative vectors T_i which are as short as possible in the L_1-norm. Consequently, a C^1-continuous cubic L_1 spline is obtained by minimizing the following functional

$$E(T_1, \ldots, T_n) + \varepsilon \sum_{i=1}^{n} |T_i|, \tag{2}$$

where ε is a strictly positive real. As this problem is also nonlinear, this functional is discretized by using the midpoint rule method for each integral. the resulting problems raised by the L_1-minimization of linear systems [1] are solved by the Vanderbei, Meketon and Freedman primal affine algorithm defined in [28] and outlined in [16].

[1] One linear system for each coordinate

From now on, we shall be interested in calculating the exact solutions to the minimization problem (1) when we have a set of five points. To do so first, we shall study the three-point case.

1.1 Univariate cubic L_1C^1 interpolation over three points

The following lemma gives the exact solution to the minimization of (1) with $n = 3$.

Lemma 1 *Let $(u_i, z_i)_{i=1,2,3}$ be three couples of real values where $u_1 < u_2 < u_3$ and the slopes be defined by $h_i = \dfrac{\Delta z_i}{\Delta u_i}$ for $i = 1, 2$. Let*

$$\min_{(b_2,b_3)\in \mathbb{R}^2} \Phi(b_1, b_2, b_3) \tag{3}$$

with

$$\Phi(b_1, b_2, b_3) = \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_1 + 6t(b_2 + b_1 - 2h_1)|dt + \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_2 + 6t(b_3 + b_2 - 2h_2)|dt, \tag{4}$$

be a univariate C^1-continuous L_1 cubic spline interpolation minimization problem where b_1, b_2 and b_3 are the first derivative values at the three points. The solutions to (3) are

$$\begin{array}{l} a)\ \textit{if } b_1 \textit{ is comprised between } h_1 + \frac{\sqrt{10}+1}{3}(h_2 - h_1) \textit{ and } h_1 \textit{ then} \\ b_2 = h_1 + \frac{\sqrt{10}-1}{3}(b_1 - h_1),\ b_3 = h_2 + \frac{\sqrt{10}-5}{5}(h_1 - h_2) + \frac{5-2\sqrt{10}}{5}(b_1 - h_1), \\ b)\ \textit{if } b_1 \textit{ is comprised between } h_1 \textit{ and } h_1 + \frac{\sqrt{10}-5}{5}(h_2 - h_1) \textit{ then} \\ b_2 = h_1 - \frac{5+\sqrt{10}}{3}(b_1 - h_1),\ b_3 = h_2 + \frac{\sqrt{10}-5}{5}(h_1 - h_2) + (b_1 - h_1), \\ c)\ \textit{otherwise } b_2 = b_3 = h_2. \end{array} \tag{5}$$

PROOF. Function $\Phi(b_1, b_2, b_3)$ is the sum of two positive convex continuous functions. The minimal value $\frac{2(\sqrt{10}-1)}{3}|b_2 - h_2|$ of the second integral according to the variables (b_2, b_3) is obtained for $b_3 - h_2 = \frac{\sqrt{10}-5}{5}(b_2 - h_2)$. By using the following variables $x = b_1 - h_1$ and $y = b_2 - h_1$, we can infer from Lemma 4 of [3] that

$$\min_{(b_2,b_3)\in \mathbb{R}^2} \Phi(b_1, b_2, b_3) = \min_{y\in \mathbb{R}} H(x, y), \tag{6}$$

where

$$H(x,y)=\begin{cases}\frac{2\left(\sqrt{10}-1\right)}{3}\left|y+h_1-h_2\right|+\left|y-x\right| \quad if \;\; \left|y-x\right|\geq 3\left|x+y\right|, \\ \frac{2\left(\sqrt{10}-1\right)}{3}\left|y+h_1-h_2\right|+\dfrac{3}{2}\left|x+y\right|+\dfrac{\left(y-x\right)^2}{6\left|x+y\right|} \;\; else.\end{cases} \tag{7}$$

If we consider the following function $\varphi_{h_1,h_2}(x)=\min\limits_{y\in\mathbb{R}} H(x,y)$, then after some calculations we obtain for any $x\in\mathbb{R}$ that the minimal values are given for

$$y=\begin{cases}\min\left(h_2-h_1,\min\left(\frac{\sqrt{10}-1}{3}x,-\frac{\sqrt{10}+5}{3}x\right)\right) \text{ if } h_2-h_1<0, \\ \min\left(h_2-h_1,\max\left(\frac{\sqrt{10}-1}{3}x,-\frac{\sqrt{10}+5}{3}x\right)\right) \text{ else.}\end{cases} \tag{8}$$

Case a) : If we assume that $h_2-h_1<0$ then from (8) we infer that for any $b_1\in\left[h_1+\frac{\sqrt{10}+1}{3}\left(h_2-h_1\right),h_1\right]$, $b_2=h_1+y=h_1+\frac{\sqrt{10}-1}{3}\left(b_1-h_1\right)$ and $b_3=h_2+\frac{\sqrt{10}-5}{5}\left(b_2-h_2\right)=h_2+\frac{\sqrt{10}-5}{5}\left(h_2-h_1\right)+\frac{5-2\sqrt{10}}{5}\left(b_1-h_1\right)$. The other cases are obtained from (8) similarly. Then the solutions to (3) given by (5) are satisfying.

In the following subsection we shall calculate the subdifferential of the continuous convex function $\varphi_{h_1,h_2}(x)=\min\limits_{y\in\mathbb{R}} H(x,y)$ defined in (6). Let us define the following functions :

$$\begin{aligned}&\varphi^1_{h_1,h_2}(x)=-\tfrac{3}{2}\left(x+h_2-h_1\right)-\tfrac{\left(h_2-h_1-x\right)^2}{6\left(x+h_2-h_1\right)},\\ &\varphi^2_{h_1,h_2}(x)=\tfrac{8-4\sqrt{10}}{3}x+\tfrac{2\left(\sqrt{10}-1\right)}{3}\left(h_1-h_2\right),\;\varphi^3_{h_1,h_2}(x)=\tfrac{2\left(\sqrt{10}-1\right)}{3}\left(h_1-h_2\right),\\ &\varphi^4_{h_1,h_2}(x)=\varphi^1_{h_1,h_2}(x),\;\varphi^5_{h_1,h_2}(x)=x-h_2+h_1,\;\varphi^6_{h_1,h_2}(x)=-\varphi^1_{h_1,h_2}(x).\end{aligned}$$

Consequently according to (7) and (8), we can infer that

$$\varphi_{h_1,h_2}(x)=\sigma\varphi^k_{h_1,h_2}(x) \text{ if } x\in\left[\min\left(\sigma x^{k-1}_{h_1,h_2},\sigma x^k_{h_1,h_2}\right),\max\left(\sigma x^{k-1}_{h_1,h_2},\sigma x^k_{h_1,h_2}\right)\right], \tag{9}$$

where $\sigma=\{1 \text{ if } h_2-h_1\geq 0 \text{ and } -1 \text{ otherwise}\}$ and

$$\begin{cases}x^0_{h_1,h_2}=-\infty,x^1_{h_1,h_2}=\frac{\sqrt{10}+1}{3}\left(h_2-h_1\right),x^2_{h_1,h_2}=0,x^3_{h_1,h_2}=\frac{\sqrt{10}-5}{5}\left(h_2-h_1\right),\\ x^4_{h_1,h_2}=-\frac{1}{2}\left(h_2-h_1\right),x^5_{h_1,h_2}=-2\left(h_2-h_1\right),x^6_{h_1,h_2}=+\infty.\end{cases} \tag{10}$$

From now on, we shall be interested in giving the solutions to the following univariate L_1C^1 interpolation problem on five points. Let $(u_i, z_i)_{i=1,\ldots,5}$ be five couples of real values where $u_1 < \cdots < u_5$ and the slopes be defined by $h_i = \dfrac{\Delta z_i}{\Delta u_i}$ for $i = 1, \ldots, 5$. Hence, the univariate L_1C^1 spline solution is obtained from

$$\min_{(b_1,\ldots,b_5)\in\mathbb{R}^5} \sum_{i=1}^{4} \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_i + 6t(b_{i+1} + b_i - 2h_i)|dt$$

where the b_i are the derivative values of the spline at u_i. This functional is the sum of positive and convex continuous functions. It can be written by

$$\begin{aligned} &\min_{b_3\in\mathbb{R}} \left(\min_{(b_2,b_1)\in\mathbb{R}^2} \Phi(b_3, b_2, b_1) + \min_{(b_4,b_5)\in\mathbb{R}^2} \Phi(b_3, b_4, b_5) \right) \\ &= \min_{b_3\in\mathbb{R}} \varphi_{h_2,h_1}(b_3 - h_2) + \varphi_{h_3,h_4}(b_3 - h_3) \end{aligned} \tag{11}$$

where Φ is defined by (4) in the previous lemma and φ_{h_i,h_j} by (9). Let us denote by $\partial\varphi_{h_i,h_j}(x)$ the subdifferential of $\varphi_{h_i,h_j}(x)$ at x (Cf. [4],[11]). Since (11) is convex and continuous its subdifferential is compact and nonempty.

Let us define $d_g(x) = \min \partial\varphi_{h_2,h_1}(x - h_2) + \min \partial\varphi_{h_3,h_4}(x - h_3)$ and $d_d(x) = \max \partial\varphi_{h_2,h_1}(x - h_2) + \max \partial\varphi_{h_3,h_4}(x - h_3)$ respectively the left and right derivative values of (11) at x. We define a sorted list $\{\beta_k\}_{k=1\ldots10}$ from the abscissa $\left(x^j_{h_3,h_4} + h_3, x^j_{h_2,h_1} + h_2\right)_{j=1,\ldots,5}$. As function (11) is convex the minimal value is obtained for any b_3 such that $d_g(b_3).d_d(b_3) \le 0$. Consequently b_3 is between $\alpha_1 = \min_{k\in\{1,\ldots,10\}}(\beta_k$ such that $d_g(\beta_k)d_d(\beta_k) \le 0$ or $d_d(\beta_k)d_g(\beta_k + 1) < 0)$ and $\alpha_2 = \max_{k\in\{1,\ldots,10\}}(\beta_k$ such that $d_g(\beta_k)d_d(\beta_k) \le 0$ or $d_d(\beta_k - 1)d_g(\beta_k) < 0)$. Three cases can thus be identified:

(1) $\alpha_1 = \alpha_2$: the solution is unique and $b_3 = \alpha_1 = \alpha_2$.
(2) $\alpha_1 \ne \alpha_2$, $d_d(\alpha_1) = 0$ and $d_g(\alpha_2) = 0$: The solutions for b_3 are $[\alpha_1, \alpha_2]$. In this case we choose $b_3 = \min_{x\in[\alpha_1,\alpha_2]} \left|x - \frac{h2+h3}{2}\right|$ so as to preserve linear parts when possible.
(3) $\alpha_1 \ne \alpha_2$, $d_d(\alpha_1) \ne 0$ and $d_g(\alpha_2) \ne 0$: The value of b_3 is unique and it belongs to $]u_1, u_2[$. We calculate this value by using a dichotomic search algorithm.

Since b_3 is obtained, we calculate b_1, b_2, b_4 and b_5 by using Lemma 1.

2 Local cubic $L_1^k C^k$ interpolation method

To define a C^k-continuous parametric spline curve with degree $2k+1$ which interpolates a set of points $(P_i)_{i=1,\ldots,n}$, one must define the derivative vectors up to the k^{th} order at these points. We propose to calculate these derivative vectors by applying iteratively for each coordinates the previous L_1C^1 interpolation algorithm within windows which contain only five data for each derivative order. In the following subsections we shall give more detail about this method.

2.1 Local cubic L_1C^1 interpolation method

From now on we shall consider the C^1 case. We define a five-point sliding window on a set of points and we calculate the derivative vector only for the middle point (Cf. Fig. 2). By translating the window, point by point over all the data, we obtain a derivative vector at each interpolation point. Hence, we are able to construct a cubic L_1-spline.

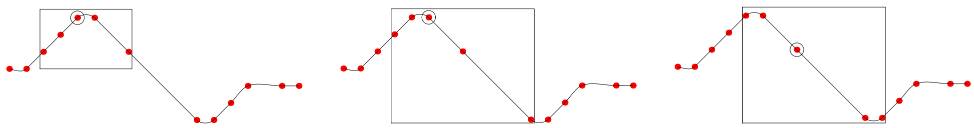


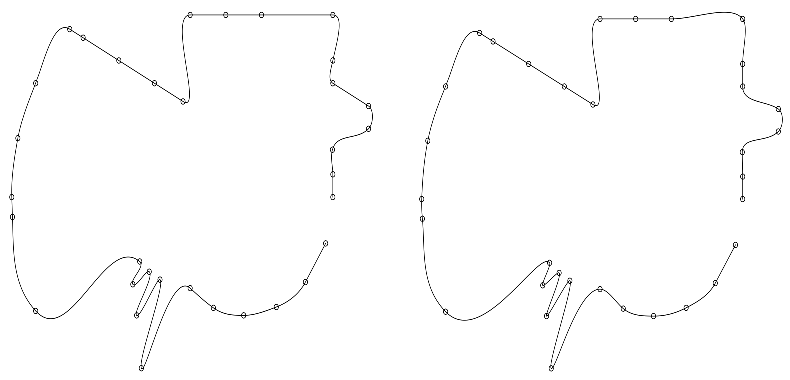
Fig. 2. Sliding window over the sets of points

If we visually compare the results obtained by Lavery's L_1C^1 global method and our local one (see Figure 3), we can see that they are quite identical. The two parts which differ are a corner [2] and a point at the bottom [3].

Since the L_1 minimization algorithm does not produce invariant curves with respect to the rotation of the data, we proposed in [2] to get it by using a local change of coordinates for 2D data points. On each interval $[u_i, u_{i+1}]$, we define a coordinate system $(P_i, \overrightarrow{u_i}, \overrightarrow{v_i})$ such that P_{i+1} coordinates in this system are $(\frac{\|\overrightarrow{P_iP_{i+1}}\|}{\sqrt{2}}, \frac{\|\overrightarrow{P_iP_{i+1}}\|}{\sqrt{2}})$. This method which has been developed for the global L_1C^1 interpolation method is quite costly in computing time as the dimension of the matrix issued from the primal affine algorithm is multiplied by two and this local change of coordinates cannot easily be extended to $d > 2$ dimensional data. In our local algorithm, another reason why this change of coordinates cannot be used is that the functional to minimize thus obtained changed for each set of five points. Here, we simply propose to apply a local change of coordinates to the five points belonging to the sliding window before applying

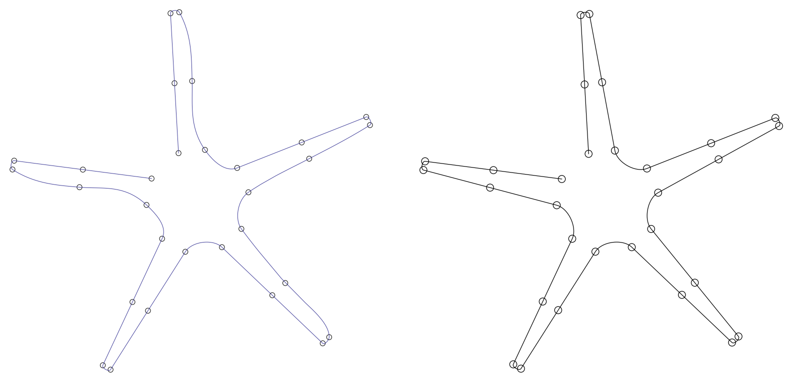
[2] where the minimization problem has a range of solutions for the derivative vectors

[3] because the minimization functions are quiet different



a.Global Method　　　　b.Local Method

Fig. 3. L_1C^1 interpolation



Global Method　　　　Local Method

Fig. 4. Star L_1C^1 interpolation

the minimization method. Consequently, the invariance is satisfied and the method minimizes the functional (1) on this new frame. For instance with 3D data points, we define an orthonormal frame based on the points belonging to each sliding window . For any pair of sets of five points equivalent up to a rotation, this local change of coordinate method gives the same derivative vector up to the rotation for the middle point of the sets. As we only keep this middle value to construct the L_1-spline solution, that result provides a coherent shape on the curve.

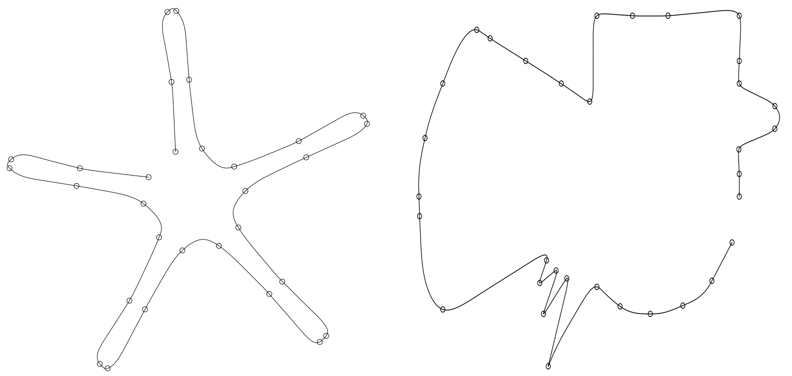
2.2 Local $L_1^kC^k$ interpolation method ($k \geq 2$)

In [2], a global L_1C^2 method was given so as to construct a parametric quintic spline which interpolates data points. In this method, we need to minimize the following function

$$\sum_{i=1}^{n-1} \frac{1}{\Delta u_i} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left|\alpha_i(t) T_{i+1} + \beta_i(t) T_i + \gamma_i(t) M_{i+1} + \delta_i(t) M_i + \eta_i(t) \Delta P_i\right| dt + \epsilon_1 \sum_{i=1}^{n} |T_i| + \epsilon_2 \sum_{i=1}^{n} |M_i| \tag{12}$$

where the M_i are the second derivative vectors at points P_i, $\alpha_i(t) = \frac{3}{2} + 15t - 6t^2 - 60t^3$, $\beta_i(t) = -\frac{3}{2} + 15t + 6t^2 - 60t^3$, $\gamma_i(t) = \left(-\frac{1}{4} - \frac{3}{2}t + 3t^2 + 10t^3\right)\Delta u_i$, $\delta_i(t) = \left(-\frac{1}{4} + \frac{3}{2}t + 3t^2 - 10t^3\right)\Delta u_i$ and $\eta_i(t) = \frac{-30t+120t^3}{\Delta u_i}$. Here, ϵ_1 and ϵ_2 are positive reals.

If we study the resulting curves thus obtained (See Fig. 5) we can see that they are smooth and do not oscillate too much.

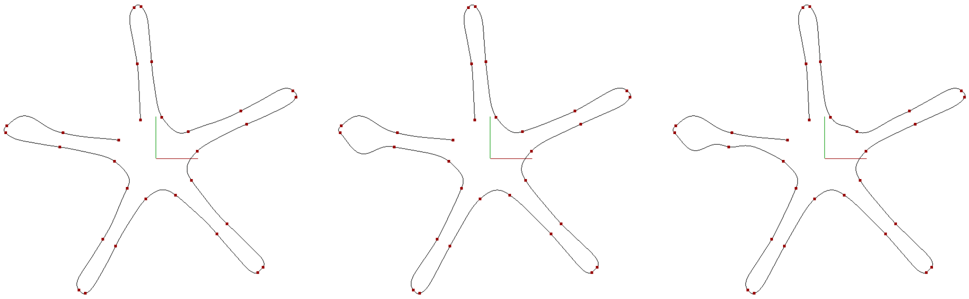


a b

Fig. 5. Global L_1C^2 interpolation method

Therefore we have tested a sliding window method with the primal affine algorithm on the sets of points as in the L_1C^1 case. We thought that this method could improve the result so that we could study the functional to minimize. On the contrary, the curves show more oscillations (See Fig. 6)

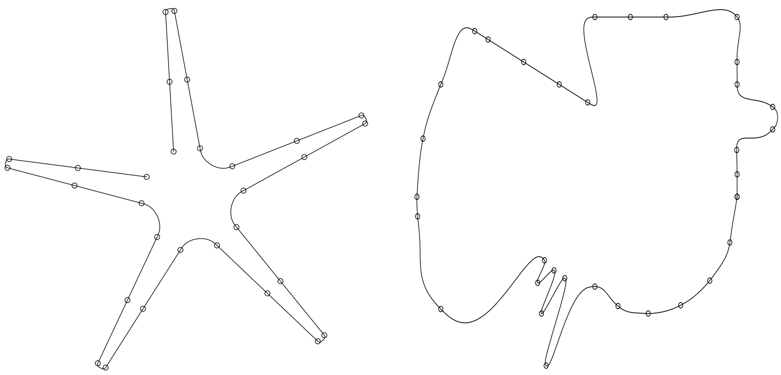
Nevertheless, we found that the local L_1C^1 interpolation method produces good spline curvature results between the data points even if the spline curves are only C^1continuous at the data points. Like Lavery in [20], we decided to use the first derivative vectors obtained by our local L_1C^1 interpolation method, but with another scope. In his article, Lavery calculated the first derivative vectors T_i^{Cubic} by minimizing (2) over (u_i, P_i) and then he found the second derivative vectors M_i minimizing (12) by using $(u_i, P_i, T_i = T_i^{Cubic})$. This two-step procedure allows to reduce the complexity of the minimization calculus but it prevents using our previous studies over cubic splines. We propose a new



5 point window 7 point window 9 point window

Fig. 6. Local L_1C^2 interpolation method with windows changing size

C^2 method, denoted by $L_1^2C^2$ which is obtained by applying twice the local L_1C^1 interpolation method. We firstly apply it on the data points (u_i, P_i) so as to obtain the first derivative vectors T_i. We apply it again onto the $(u_{i,}T_i)$ in order to calculate the second derivative vectors M_i. As we can see in Figure 7, the quintic spline curve solution has shape preserving properties. As we show further down, our $L_1^kC^k$ method allows us to benefit from the 'shape preserving'property of the local L_1C^1 algorithm.

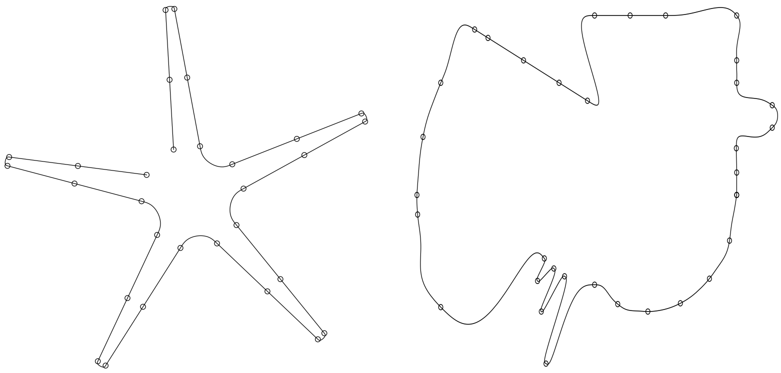


a b

Fig. 7. $L_1^2C^2$ interpolation method

This $L_1^2C^2$ sliding window method needs a nine-points sliding window. Actually, we need $P_{i-2}, \ldots, P_{i+2}$ so as to calculate vector T_i and each vector M_i is calculated from vectors $T_{i-2}, \ldots, T_{i+2}$.

If we want to construct C^k-continuous splines (with $k \geq 2$), we have to calculate the derivative vectors up to the k^{th} order. To do so, we can simply use our local L_1C^1 method repeated k times, which is noted $L_1^kC^k$ for $k \geq 1$ (consequently $L_1^1C^1 = L_1C^1$). As our local L_1C^1 algorithm has shape preserving properties, we shall think that this iterative method will produce smooth high degree spline curves. As we can see in Figure 8, the curves are C^3-continuous and they preserve the data well.



a b

Fig. 8. local $L_1^3C^3$ interpolation method

3 Properties of the local L_1C^1 algorithm

3.1 Linear shape preservation

For any arbitrary set of points, our local L_1C^1 produces linear curve parts when up to three points lie on a line, contrary to the global L_1C^1 solution curve (Cf. Fig. 9).

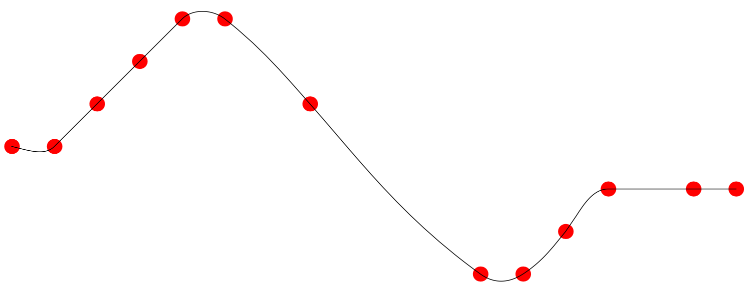
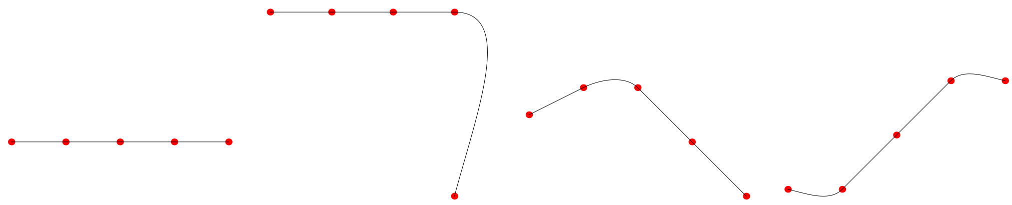


Fig. 9. Global L_1C^1 interpolation

In the univariate case, we showed in [1] that for interpolation points belonging to the Heaviside function, where three of them are located on one of the half-line of this function, the derivative vectors at these points are necessarily collinear to this line. Actually, in the parametric case over some sets of five points, we can see in Figure 10 that the L_1C^1 cubic interpolation produces derivative vectors which are collinear when three points lie on a line.

To demonstrate this property, we first study the univariate case. By using the fact that the u_i are fixed according to the chordal parametrization, we then



5 aligned points 4 aligned points 3 aligned points 3 aligned points

Fig. 10. L_1C^1 interpolation over 5 points

show that linear parts are preserved in $\mathbb{R}^d$ (with $d \geq 1$), which implies that the derivative vectors T_i at such points are collinear.

Lemma 2 *Let $(u_i, \alpha_i)_{i=1,\dots,5}$ be five couples of real values where $u_1 < u_2 < \dots < u_5$. We denote by $h_i = \dfrac{\Delta\alpha_i}{\Delta u_i}$ for $i = 1, \dots, 4$ the slopes between the points. The minimization problem*

$$\min_{(b_1,\dots,b_5)\in\mathbb{R}^5} \sum_{i=1}^{4} \left(\int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_i + 6t(b_{i+1} + b_i - 2h_i)|dt \right) \tag{13}$$

has the following solutions :
a) if $h_i = h_{i+1} = h$ for $i = 1$ (resp. $i = 3$) except the case $h_j = h_{j+1} \neq h$ for $j = 3$ (resp. $j = 1$) then $b_3 = b_i = b_{i+1} = h$,
b) if $h_2 = h_3 = h$ and $(h_1 - h)(h_4 - h) < 0$ then $b_1 = h_1 + h_4 + \frac{\sqrt{10}-5}{5}(h - h_1)$, $b_2 = b_3 = b_4 = h$ and $b_5 = h_4 + \frac{\sqrt{10}-5}{5}(h - h_4)$,
c) if $h_2 = h_3 = h$ and $(h_1 - h)(h_4 - h) > 0$ then $b_2 = b_4 = h$ and b_3 is comprised between h and $\min\left(h + \frac{\sqrt{10}-5}{5}(h_4 - h), h + \frac{\sqrt{10}-5}{5}(h_4 - h)\right)$,
d) if $h_1 = h_2$ and $h_3 = h_4$ with $h_1 \neq h_3$ (we call it the corner case) then $b_1 = b_2 = h_1$, $b_4 = b_5 = h_3$ and b_3 is comprised between h_1 and h_3.
Moreover if we add $\left|b_3 - \frac{h_2+h_3}{2}\right|$ to (13) then the solution is unique in each cases and $b_3 = \frac{h_2+h_3}{2}$.

PROOF. From (11), we know that this minimization problem (13) can be written as follows :

$$\min_{b_3\in\mathbb{R}} \varphi(b_3) = \varphi_{h_2,h_1}(b_3 - h_2) + \varphi_{h_3,h_4}(b_3 - h_3)$$

where the φ_{h_i,h_j} are positive convex functions. Since $h_i = h_{i+1} = h$ (for $i = 1$ or $i = 3$) then $\varphi_{h_i,h_{i+1}}(b_3 - h) = \frac{5}{3}|b_3 - h|$. As the minimal value of this strictly convex function at $b_3 = h$ is equal to zero, consequently whatever the values of the other slopes (except for case d)), φ has a unique minimum at $b_3 = h$. Hence, property a) holds. Case d) : If $h_1 = h_2$ and $h_3 = h_4$ with $h_1 \neq h_3$ then $\varphi(b_3) = \frac{5}{3}|b_3 - h_1| + \frac{5}{3}|b_3 - h_3|$ is minimal for any b_3 comprised between h_1 and h_3. Case b) and c) : The minimal value of $\varphi_{h_2,h_1}(b_3 - h_2)$ is obtained for b_3

comprised between h and $h+\frac{\sqrt{10}-5}{5}(h_1-h)$. Similarly for $\varphi_{h_3,h_4}(b_3-h_3)$ the minimum value is obtained for b_3 comprised between h and $h+\frac{\sqrt{10}-5}{5}(h_4-h)$. Moreover, their minimal values are equal. For $(h_1-h)(h_4-h)>0$, these intervals overlap. Hence, the minimum of φ is obtained for b_3 comprised between h and $\min\left(h+\frac{\sqrt{10}-5}{5}(h_4-h), h+\frac{\sqrt{10}-5}{5}(h_4-h)\right)$. If $(h_1-h)(h_4-h)<0$ then the minimal value of φ is only obtained for $b_3=h$. Consequently, if we add $\left|b_3-\frac{h_2+h_3}{2}\right|$ to the minimization function φ, then for each case, $b_3=\frac{h_2+h_3}{2}$ is the unique solution.

The linear parts of the data can be preserved in each cases. For case d) there are two such solutions. The following proposition allows us to extend the previous univariate study to the parametric case.

Proposition 3 *Let $P_1, P_2,\ldots,P_5$ be five data points. We associate to each point P_i a real value u_i such that $u_1<u_2<\cdots<u_5$. The u_i are chosen according to the chordal partition. If at least three data points are aligned then the minimization problem (1) has a unique interpolation L_1C^1 cubic spline which preserves the linear part except in the corner problem where there are two such solutions.*

PROOF. In this case, we have to minimize (1) with $n=5$. As we use the L_1-norm, this minimization can be done on each coordinate separately. In addition, the chordal partition allows us to keep the same value for the slope between each consecutive coordinate issued from aligned data points. Consequently, the result can therefore be inferred by using the previous proposition.

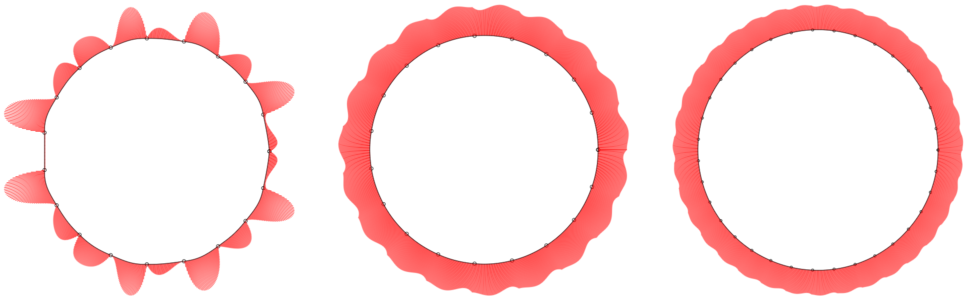
Similarly to the C^1-continuous case, the $L_1^2C^2$ method gives quintic spline with linear parts when the data points lie on a line. Actually, the first use of the local L_1C^1 algorithm always gives first derivative vectors which are collinear with lines defined by the aligned data points. Hence, the second use of this algorithm on these vectors also gives collinear vectors. This property can be extended to $L_1^kC^k$ interpolation methods which are simply obtained by iterating this process.

3.2 Good curvature

In Figure 11, we have created curves that interpolate twenty points which lie on a circle. As we can see, the local method gives good curvatures [4] which are

[4] the curvature of the spline is represented by an offset according to the curvature value at some points

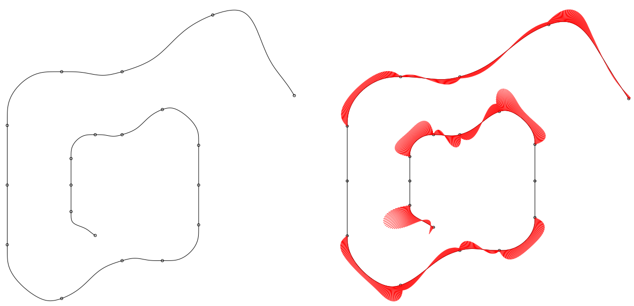
almost monotonous contrary to the solution of the global L_1C^2 interpolation case.



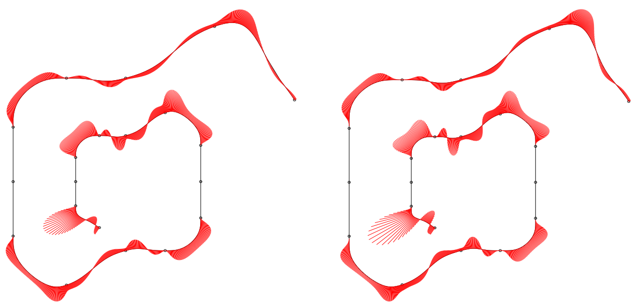
a. Global L_1C^2 b. Local $L_1^2C^2$ c. Local $L_1^3C^3$

Fig. 11. Curvatures for circle approximation by a "closed" 20 points set

The splines obtained by local $L_1^kC^k$, $k \geq 2$ method are relatively smooth without oscillations (see. Figure 12). This is partly due to the fact that we have a smooth set of derivative vectors by applying many times our local L_1C^1 algorithm . We can see in Figure 12.c that the splines do not oscillate too much despite the high degree of the spline curves[5] .



a. Local $L_1^2C^2$



b. Local $L_1^3C^3$ c. Local $L_1^5C^5$

Fig. 12. Curvatures for a snail like data set interpolated by local $L_1^kC^k$ methods

[5] eleventh degree spline for C^5-continuity

3.3 Computational complexity

In our local five-point window algorithm, the time needed to calculate each derivative vector is small and almost constant. Hence, the time needed to calculate the global spline curve is linear and we can interpolate large data sets of points. Moreover, our original iterative approach allows us to calculate C^k continuous interpolation spline curves with a linear time complexity. Indeed, to create a C^k interpolation spline curve of degree $2k+1$, we need to apply the local L_1C^1 method, which allows us to preserve this linear complexity, k times consecutively on the successive data.

In addition to this, our local L_1C^1 algorithm can be parallelized on multi-processor computers by distributing the computation of each five-point window over the processors. That is possible as each calculus is independent at each stage. Furthermore, for the C^k continuous interpolating curves, the iterative use of the local L_1C^1 method allows us to start the calculus of the first k^{th} derivative vector when five $k-1^{th}$ derivative vectors are known. Consequently, we only need a sliding window of $4k+1$ points for a C^k spline. This allows us to accelerate the calculus.

For example, to compute the local $L_1^2C^2$ quintic spline solution, we only need a nine-point sliding window so as to be able to parallelize the calculus.

4 Conclusion

We have defined a local method which is efficient to interpolate sets of data points by univariate cubic splines using the L_1 norm. This method keep the shape-preserving properties without having the drawback of the commonly used global L_1 method : Our method use algebraic results which are faster and more stable than the numerical approximations used before. We can use it to calculate C^k-continuous curves with good curvature despite their high degrees. Furthermore, it is very simple to parallelize the calculus. If we want to reduce the time spent on calculations on a very large number of data, we need a large number of processors. Because our methods can be used in $\mathbb{R}^d$, the interpolation of data by spline curves from various domains (Cf. [8], [10]) is possible keeping the good properties of our algorithm. We currently use the $L_1^2C^2$ interpolation method for enhanced trajectory planning for machining with industrial six-axis robots (Cf. [22]).

5 Acknowledgments

The authors thank the referees for their helpful suggestions and comments.

References

[1] P. Auquiert, Interpolation de points par des splines L_1 régulières, Phd Thesis, Université de Valenciennes et du Hainaut-Cambrésis, LAMAV, (20 décembre 2007).

[2] P. Auquiert, O. Gibaru, E. Nyiri, C^1 and C^2-continuous polynomial parametric L_p splines ($p \geq 1$), Comp. Aided Geom. Design 24 (2007), 373–394.

[3] P. Auquiert, O. Gibaru, E. Nyiri, On the cubic L_1 spline interpolant to the Heaviside function, Numer. Algor. 46 (2007), 321–332.

[4] D. Azé, Eléments d'analyse convexe et variationnelle, Ellipses, (1997).

[5] H. Cheng, S.-C. Fang, J.E. Lavery, Univariate cubic L_1 splines—A geometric programming approach, Math. Methods Oper. Res. 56 (2002) 197–229.

[6] H. Cheng, S.-C. Fang, J.E. Lavery, An efficient algorithm for generating univariate cubic L1 splines, Comput. Optim. Appl. 29 (2004) 219–253.

[7] H. Cheng, S.-C. Fang, J.E. Lavery, Shape-preserving properties of univariate cubic L_1 splines, J. Comput. Appl. Math. 174 (2005) 361–382.

[8] O. Gibaru, Tensorial rational surfaces with base points via massic vectors. SIAM J. Numer. Anal. 42, n°4 (2004), 1415-1434.

[9] D.E. Gilsinn, J.E. Lavery, Shape-preserving, multiscale fitting of bivariate data by cubic L_1 smoothing splines, in: C.K. Chui, L.L. Schumaker, J. Stöckler (Eds.), Approximation Theory X:Wavelets, Splines, and Applications,Vanderbilt University Press, Nashville, TN, (2002), pp. 283–293.

[10] L. Han, L.L. Schumaker, Fitting monotone surfaces to scattered data using C1 piecewise cubics, SIAM J. Numer. Anal. 34 (1997) 569–585.

[11] J.B. Hiriart-Urruty, C. Lemaréchal, Fundamentals of Convex Analysis, Springer, (2001).

[12] Hoschek, J., Lasser, D. Fundamentals of computer Aided Geometric Design, A.K. Peters, Wellesley, 1993.

[13] F. Kuijt, R. van Damme, Shape preserving interpolatory subdivision schemes for nonuniform data, J. Approx. Theory 114 (2002) 1–32.

[14] J.E. Lavery, Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines, Comput. Aided Geom. Design 17 (2000), 319–336.

[15] J.E. Lavery, Shape-preserving, multiscale fitting of univariate data by cubic L1 smoothing splines, Comput. Aided Geom. Design 17 (2000) 715–727.

[16] J.E. Lavery, Shape-preserving, multiscale interpolation by bi- and multivariate cubic L_1 splines, Comput. Aided Geom. Design 18 (2001) 321–343.

[17] J.E. Lavery, The state of the art in shape preserving, multiscale modeling by L1 splines, in: M.L. Lucian, M. Neamtu (Eds.), Proceedings of SIAM Conference on Geometric Design Computing, Nashboro Press, Brentwood, TN, (2004), pp. 365–376.

[18] J.E. Lavery H. Cheng, S.-C. Fang, Shape-preserving properties of univariate cubic L_1 splines, Journal of Computational and Applied Mathemetics 174 (2005), 361–382.

[19] J.E. Lavery, Shape-preserving, first-derivative-based parametric and non parametric cubic L_1 spline curves, Comput. Aided Geom. Design 23 (2006), 276–296.

[20] J.E. Lavery, Shape-preserving univariate cubic and higher-degree L_1 splines with function-value-based and multistep minimization principles, Computer Aided Geom. Design 26 (2009), 1–16.

[21] I.J. Lustig, R.E. Marsten, D.F. Shanno, Interior point methods for linear programming: computational state of the art, ORSA J. Comput. 6 (1994) 1–14.

[22] A. Olabi, R. Bare, E. Nyiri, O. Gibaru, L_1 paramatric interpolation and feedrate planing for machining robots, The Swedish Production Symposium Göteborg, Sweden, (2-3 december 2009)

[23] J. Peters, Smoothness, fairness and the need for better multisided patches, in: R. Goldman, R. Krasauskas (Eds.), Topics in Algebraic Geometry and Geometric Modeling, Contemporary Mathematics, vol. 334, American Mathematical Society, Providence, RI, (2003), pp. 55–64.

[24] F.I. Utreras, The variational approach to shape preservation, in: P.J. Laurent, A. Le Mehauté, L.L. Schumaker (Eds.), Curves and Surfaces, Academic Press, NewYork, 1991, pp. 461–476.

[25] R.J. Vanderbei, Affine-scaling for linear programs with free variables, Math. Program. 43 (1989) 31–44.

[26] R.J. Vanderbei, LOQO: An interior point code for quadratic programming, Statistics and Operations Research Technical Report SOR-94-15, Princeton University, (1995).

[27] R.J. Vanderbei, Linear Programming: Foundations and Extensions, second ed., Kluwer Academic, Boston, (2001).

[28] R.J.Vanderbei, M.J. Meketon, B.A. Freedman,A modification of Karmarkar's linear programming algorithm, Algorithmica 1 (1986) 395–407.

[29] Y.Wang, S.-C. Fang, J.E. Lavery, H. Cheng, A geometric programming approach for bivariate cubic L1 splines, Comp. Math. Appl. 49 (2005) 481–514.

[30] Y. Wanga, S.-C.Fang, J.E. Lavery, A compressed primal-dual method for generating bivariate cubic L1 splines, Journal of Computational and Applied Mathematics 201 (2007) 69–87.