A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics

C. Beccari [a], G. Casciola [b], L. Romani [b,*]

[a] *Department of Pure and Applied Mathematics, University of Padova, Via G. Belzoni 7, 35131 Padova, Italy*

[b] *Department of Mathematics, University of Bologna, P.zza di Porta San Donato 5, 40127 Bologna, Italy*

Abstract

In this paper we propose a non-stationary C^1-continuous interpolating 4-point scheme which provides users with a single tension parameter that can be either arbitrarily increased, to tighten the limit curve towards the piecewise linear interpolant between the data points, or appropriately chosen in order to represent elements of the linear spaces spanned respectively by the functions $\{1, x, x^2, x^3\}$, $\{1, x, e^{sx}, e^{-sx}\}$ and $\{1, x, e^{\mathrm{i}sx}, e^{-\mathrm{i}sx}\}$. As a consequence, for special values of the tension parameter, such a scheme will be capable of reproducing all conic sections exactly.

Exploiting the reproduction property of the scheme, we derive an algorithm that automatically provides the initial tension parameter required to exactly reproduce a curve belonging to one of the previously mentioned spaces, whenever the initial data are uniformly sampled on it.

The performance of the scheme is illustrated by a number of examples that show the wide variety of effects we can achieve in correspondence of different tension values.

Key words: Subdivision; Interpolation; Curves; Locality; Tension control; Conics reproduction

* Corresponding author.

Email addresses: beccari@dm.unibo.it (C. Beccari), casciola@dm.unibo.it (G. Casciola), romani@dm.unibo.it (L. Romani).

1 Introduction

Subdivision is an efficient method for generating smooth curves and surfaces in Computer Aided Geometric Design (CAGD). A univariate subdivision process defines a curve as the limit of a sequence of refinements performed on an initial polyline. If the points generated at each refinement level are retained at all successive levels, the scheme is said to be interpolatory (Dyn, 2002). The important schemes for applications should allow to control the shape of the limit curve and be capable of reproducing families of curves widely used in Computer Graphics, such as conic sections and polynomials. Although a wide variety of schemes has been proposed in the literature, the difficulty of combining the above requirements prevented the diffusion of interpolatory schemes in applications. Indeed, the schemes proposed up to now either possess a tension parameter to control the tightness of the limit curve (Dyn et al., 1987; Dyn et al., 2005) or are able to reproduce circles (Zhang, 1996; Ivrissimtzis et al., 2002; Jena et al., 2003) and generate conic sections (Dyn et al., 2003). However, none of them possesses an intuitive shape parameter that, gradually increased, allows to increase the tightness of the limit curve and for special values associated with the initial data, allows to represent all conic sections exactly. The only one which possesses a tension parameter that, appropriately chosen, allows to reproduce circles as well, is the non-stationary scheme presented in Morin et al. (2001). Anyway, this is not interpolatory. Aim of this work is therefore to generate an interpolatory subdivision scheme with a tension parameter, that is capable of reproducing circles and all other conic sections exactly whenever such a parameter has been chosen correctly.

The paper is structured as follows. In Section 2 we derive a C^1-continuous subdivision scheme which allows us to represent cubic polynomials as well as a certain class of hyperbolic and trigonometric functions, related to the definition of conic sections. Next, in Section 3, we show its special property of conics reproduction and we derive an algorithm that automatically provides the initial tension parameter required to exactly reproduce a curve belonging to one of the previously mentioned families, whenever the initial data are uniformly sampled on it. In Section 4 we propose an endpoint rule for generating open curves with the same regularity. Finally in Section 5 we demonstrate the role of the tension parameter by a few examples.

2 Definition of the scheme

In this section we are going to define an interpolating 4-point scheme that captures three different curve schemes which are capable of representing elements in the class of cubic polynomials, hyperbolic functions and trigonometric functions.
For the sake of conciseness, in the remainder of this paper we will indicate with V_0, V_s and $V_{\mathrm{i}s}$ the spaces spanned respectively by the functions $\{1, x, x^2, x^3\}$, $\{1, x, e^{sx}, e^{-sx}\}$ and $\{1, x, e^{\mathrm{i}sx}, e^{-\mathrm{i}sx}\}$.

Observe that, depending on the value of t^2 (where t is either a positive real or imaginary constant), the solutions of the differential equation $D^4 \cdot -t^2 D^2 \cdot = 0$ are linear combinations of the functions in V_0 (whenever $t = 0$), V_s (whenever $t = s$, $s > 0$) or $V_{\mathrm{i}s}$ (whenever $t = \mathrm{i}s$, $s > 0$). Hence, due to the identities

$$\cosh(sx) = \tfrac{1}{2}(e^{sx} + e^{-sx}), \quad \sinh(sx) = \tfrac{1}{2}(e^{sx} - e^{-sx}),$$

$$\cos(sx) = \tfrac{1}{2}(e^{\mathrm{i}sx} + e^{-\mathrm{i}sx}), \quad \sin(sx) = \tfrac{1}{2\mathrm{i}}(e^{\mathrm{i}sx} - e^{-\mathrm{i}sx}),$$

the common insertion rule which unifies the three cases will be obtained by interpolation with a function from the linear space spanned by $\{1, x, e^{tx}, e^{-tx}\}$. In particular, whenever $t = 0$ such an insertion rule will reproduce cubic polynomials, whenever $t = s > 0$ it will reproduce hyperbolic functions and whenever $t = \mathrm{i}s$, $s > 0$, it will reproduce trigonometric functions. In this way, interpolating the data $(2^{-k}h, p^k_{j+h})$, $h = -1, 0, 1, 2$, by a function of the form

$$f(x) = a_0 + a_1 x + a_2 e^{tx} + a_3 e^{-tx}, \tag{1}$$

we get the following system of equations

$$\begin{cases} f(-\frac{1}{2^k}) = p^k_{j-1} \\ f(0) = p^k_j \\ f(\frac{1}{2^k}) = p^k_{j+1} \\ f(\frac{1}{2^{k-1}}) = p^k_{j+2} \end{cases}$$

from which it follows

$$\begin{cases} p^k_{j-1} = a_0 - a_1 \frac{1}{2^k} + a_2\ e^{-\frac{t}{2^k}} + a_3\ e^{\frac{t}{2^k}} \\ p^k_j = a_0 + a_2 + a_3 \\ p^k_{j+1} = a_0 + a_1 \frac{1}{2^k} + a_2\ e^{\frac{t}{2^k}} + a_3\ e^{-\frac{t}{2^k}} \\ p^k_{j+2} = a_0 + a_1 \frac{1}{2^{k-1}} + a_2\ e^{\frac{t}{2^{k-1}}} + a_3\ e^{-\frac{t}{2^{k-1}}}. \end{cases} \tag{2}$$

Then, solving (2) with respect to a_0, a_1, a_2, a_3, we get for any $t \neq 0$ and, whenever $t = \mathrm{i}s$, for any s not a multiple of π, the following expression for the coefficients in (1):

$$a_0 = \frac{\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}}\right) p^k_j - p^k_{j-1} - p^k_{j+1}}{e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2}$$

$$a_1 = 2^k \frac{\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} + 1\right)(p_{j+1}^k - p_j^k) + p_{j-1}^k - p_{j+2}^k}{e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2}$$

$$a_2 = \frac{\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2\right)(-p_j^k + 2p_{j+1}^k - p_{j+2}^k) + \left(e^{-\frac{t}{2^k}} - 1\right)^2 (p_{j-1}^k - 2p_j^k + p_{j+1}^k)}{\left(e^{-\frac{t}{2^k}} - e^{\frac{t}{2^k}}\right)\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2\right)^2}$$

$$a_3 = \frac{\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2\right)(p_j^k - 2p_{j+1}^k + p_{j+2}^k) - \left(e^{\frac{t}{2^k}} - 1\right)^2 (p_{j-1}^k - 2p_j^k + p_{j+1}^k)}{\left(e^{-\frac{t}{2^k}} - e^{\frac{t}{2^k}}\right)\left(e^{-\frac{t}{2^k}} + e^{\frac{t}{2^k}} - 2\right)^2}.$$

To get the insertion rule now, we only need to compute the value of the interpolating function $f(x)$ at the grid point $\frac{1}{2^{k+1}}$, defining the new point $p_{j+\frac{1}{2}}^k$ as a linear combination of the four consecutive points p_{j-1}^k, p_j^k, p_{j+1}^k, p_{j+2}^k in the last set:

$$f\left(\frac{1}{2^{k+1}}\right) = \left[\frac{1}{2} + \frac{1}{2\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}}\right)\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}} + 2\right)}\right](p_j^k + p_{j+1}^k)$$

$$-\frac{1}{2\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}}\right)\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}} + 2\right)}(p_{j-1}^k + p_{j+2}^k) \equiv p_{j+\frac{1}{2}}^k.$$

In this way we will define the set of points at the $(k+1)$-th level of refinement, as:

$$\begin{aligned} p_{2j}^{k+1} &= p_j^k \\ p_{2j+1}^{k+1} &= p_{j+\frac{1}{2}}^k = \left(\frac{1}{2} + w^{k+1}\right)(p_j^k + p_{j+1}^k) - w^{k+1}(p_{j-1}^k + p_{j+2}^k) \end{aligned} \tag{3}$$

where

$$w^{k+1} = \frac{1}{2\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}}\right)\left(e^{\frac{t}{2^{k+1}}} + e^{-\frac{t}{2^{k+1}}} + 2\right)}. \tag{4}$$

Remark 1 *The subdivision scheme defined in (3)-(4) turns out to be a special case of the general exponential reproducing schemes proposed by Dyn et al. (2003). Note that when $t = 0$ the weight w^{k+1} is well-defined and turns out to be $\frac{1}{16}$, in such a way that the subdivision rules (3) reduce to the 4-point Dubuc-Deslauriers insertion rules (Dubuc, 1986; Deslauriers and Dubuc, 1989), which reproduce cubic polynomials. This is due to the fact that, for $t = 0$, the solutions of the differential equation $D^4 \cdot -t^2 D^2 \cdot = 0$ are exactly cubic polynomials.*

Proposition 2 *Let $t_k = \frac{t}{2^k}$, where whenever $t = \mathrm{i}s$, we assume $s \in (0, \pi)$, and define*

$$v^k = \frac{1}{2}(e^{t_k} + e^{-t_k}). \tag{5}$$

Then for any $k \geq 0$ the parameters v^k and v^{k+1} defined as in (5) satisfy the recurrence

$$v^{k+1} = \sqrt{\frac{1 + v^k}{2}}. \tag{6}$$

PROOF. Equation (6) follows from the fact that

$$\sqrt{\tfrac{1+v^k}{2}} = \tfrac{1}{2}\sqrt{2 + e^{t_k} + e^{-t_k}} = \tfrac{1}{2}\left(e^{\frac{t_k}{2}} + e^{-\frac{t_k}{2}}\right) = \tfrac{1}{2}\left(e^{t_{k+1}} + e^{-t_{k+1}}\right) = v^{k+1}. \quad \square$$

Remark 3 *Note that the recurrence relation defined in (6) satisfies the property*

$$\lim_{k \to +\infty} v^k = 1.$$

We now express the weights in (3) in terms of v^{k+1} so that, given any arbitrary tension value v^0 in $(-1, +\infty)$, and exploiting (6) to update it at each refinement level, we can generate an interpolating limit curve whose shape is easily controlled by the choice of v^0.

Definition 4 *Given a set of control points $P^0 = \{p_j^0 \mid j \in \mathbb{Z}\}$ at refinement level 0 and an arbitrary initial tension parameter $v^0 \in (-1, +\infty)$, we define a subdivision scheme that generates a new set of control points $P^{k+1} = \{p_j^{k+1} \mid j \in \mathbb{Z}\}_{k \geq 0}$ at the $(k+1)$-th level of refinement, by the subdivision rules (3) with weight*

$$w^{k+1} = \frac{1}{8v^{k+1}(1 + v^{k+1})}, \qquad k \geq 0, \tag{7}$$

where for any $k \geq 0$ the sequence v^{k+1} in (7) is recursively defined through equation (6).

Note that for any choice of the initial tension value v^0 in the range $(-1, +\infty)$, the recurrence in (6) is always well-defined and $v^{k+1} > 0$ for any $k \geq 0$. Furthermore $\lim_{k \to +\infty} w^k = \frac{1}{16}$.

Remark 5 *The subdivision scheme defined in (3)-(7) generates C^1-continuous limit curves for any choice of the initial tension parameter $v^0 \in (-1, +\infty)$. This follows from the convergence analysis results in Dyn and Levin (1995), as explained for the general exponentials reproducing schemes proposed in Dyn et al. (2003).*

3 Reproduction of conic sections

In the first part of this section we are going to show that, choosing correctly the initial tension parameter v^0, the subdivision scheme defined in (3)-(7) allows to reproduce the classes of cubic polynomials, hyperbolic functions and trigonometric functions identified respectively by the spaces V_0, V_s and $V_{\mathrm{i}s}$. Successively, in subsection 3.2, we will illustrate a procedure that allows to automatically compute the special tension value required to reproduce curves from the above three classes whenever the initial points are uniformly sampled on them.

3.1 The initial tension parameters for conics reproduction

Observe that, defining v^0 as in (5), that is $v^0 = \frac{1}{2}(e^t + e^{-t})$, and assuming $t = 0$, $t = s$ (with $s > 0$) and $t = \mathrm{i}s$ (with $s \in (0, \pi)$), we get respectively:

- $v^0 = 1$, hence $v^k = 1$ and $w^k = \frac{1}{16}$ for all $k \geq 1$;
- $v^0 = \cosh(s) > 1$, hence $v^k = \cosh(\frac{s}{2^k})$ and $w^k = \frac{1}{16\cosh\left(\frac{s}{2^k}\right)\cosh^2\left(\frac{s}{2^{k+1}}\right)}$ for all $k \geq 1$;
- $v^0 = \cos(s) \in (-1, 1)$, hence $v^k = \cos(\frac{s}{2^k})$ and $w^k = \frac{1}{16\cos\left(\frac{s}{2^k}\right)\cos^2\left(\frac{s}{2^{k+1}}\right)}$ for all $k \geq 1$ (thus the scheme in (3)-(7) coincides with the interpolating 4-point scheme on the circle presented in Ivrissimtzis et al. (2002)).

In this way, starting from a set of points uniformly sampled on a function in V_0, V_s or $V_{\mathrm{i}s}$ the value of the parameter v^0 identifies the space to which the limit function generated by the scheme belongs. More precisely,

- if $v^0 = 1$, then the limit curve belongs to V_0, namely the linear space spanned by cubic polynomials;
- if $v^0 > 1$ then $v^0 = \cosh(s)$ for some $s \in \mathbb{R}^+$, hence the limit curve belongs to V_s with $s = \operatorname{acosh}(v^0)$;
- if $v^0 \in (-1, 1)$ then $v^0 = \cos(s)$ for some $s \in (0, \pi)$, hence the limit curve belongs to $V_{\mathrm{i}s}$ with $s = \operatorname{acos}(v^0)$.

As a consequence, the following result holds.

Proposition 6 *Choosing the initial tension parameter $v^0 = \cosh(su)$, $s, u > 0$, the subdivision scheme defined in (3)-(7) reproduces exactly the hyperbolic functions $f(x) = \cosh(sx)$ and $f(x) = \sinh(sx)$ whenever the given data points (ju, p_j^0), $j \in \mathbb{Z}$ lie on such functions.*
Analogously, choosing the initial tension parameter $v^0 = \cos(su)$, $su \in (0, \pi)$, the subdivision scheme defined in (3)-(7) reproduces exactly the trigonometric functions $f(x) = \cos(sx)$ and $f(x) = \sin(sx)$ whenever the given data points (ju, p_j^0), $j \in \mathbb{Z}$

lie on such functions.

PROOF. The result above follows from the construction of the scheme. □

Corollary 7 *By taking as initial data the points* $p_j^0 = (a\cosh(ju), b\sinh(ju))$, $j \in \mathbb{Z}$, $u > 0$, *equidistant in the parameter* u *on the parametric representation of the hyperbola, and choosing the initial tension parameter* $v^0 = \cosh(u)$, *the resulting limit curve is the hyperbola itself (see Fig.2 where* $a = b = u = 1$*).*
Analogously, if we take as initial data four points $p_j^0 = (a\cos(ju), b\sin(ju))$, $j \in \mathbb{Z}$, $u \in (0, \pi)$, *equidistant in the parameter* u *on the parametric representation of the ellipse with center* 0 *and radii* a, b, *by choosing the initial tension parameter* $v^0 = \cos(u)$, *the resulting limit curve is exactly the ellipse itself (see Fig.2 where* $a = 4$, $b = 2$, $u = \frac{\pi}{2}$*). In particular, when* $a = b$, *the resulting limit curve is exactly the circle of radius* a *(see Fig.5 where* $a = b = 1$, $u = \frac{\pi}{2}$*).*

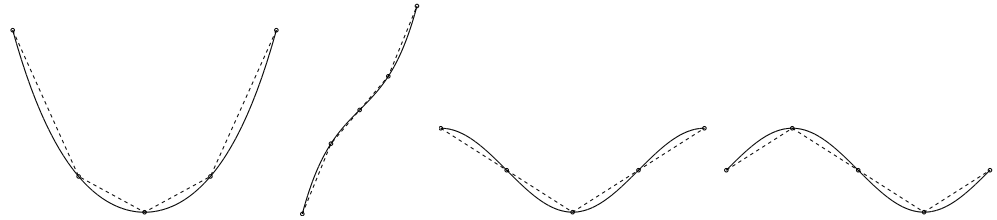

Fig. 1. Reproduction of hyperbolic and trigonometric functions: $f(x) = \cosh(x)$, $f(x) = \sinh(x)$, $f(x) = \cos(x)$, $f(x) = \sin(x)$.

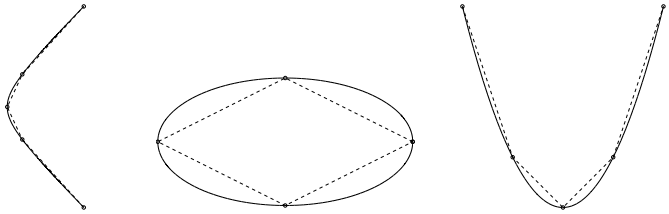

Fig. 2. Reproduction of conic sections: hyperbola, ellipse, parabola.

Remark 8 *Figs. 1-2 have been obtained by extending uniformly on each side the open control polygon* $P^0 = \{p_j^0 \mid j = 0, ..., 4\}$ *by two extra segments whose endpoints lie on the curve. Whenever two auxiliary points have been defined for each endpoint, we can deal with open polygons leaving the subdivision rules (3)-(7) unmodified. In this way the open curve generated in the limit through (3)-(7) with* $j = -2, ..., 6$, *trivially turns out to have* p_0^0 *as first endpoint and* p_4^0 *as the last one (see Section 4).*

3.2 Automatical computation of the initial tension parameter

In this subsection we are going to derive an algorithm that, starting by a sufficient number of equally-spaced data, is capable to automatically compute the initial ten-

sion parameter v^0 such that, if the initial points are sampled from a curve belonging to one of the spaces V_0, V_s or $V_{\mathrm{i}s}$, by applying the scheme defined by (3)-(7) with the so computed tension, we will be able to reproduce the curve from which those points are sampled.
The following algorithm provides a brief sketch of the procedure one could exploit in order to compute the value of the tension parameter that should be used to reproduce a curve belonging to one of the above three spaces, whether the initial points belong to it.

Algorithm in pseudo-code:

[1] Consider an initial set of equally-spaced points $\{ju, p_j\}_{j=0,\dots,2n}$ $(n \geqslant 4)$ for some positive u
[2] for $j = 1, ..., n-2$
  [2.1] consider the quadruple of even-indexed points
    $p_{2j-2}, p_{2j}, p_{2j+2}, p_{2j+4}$
  [2.2] determine w_j by solving componentwise the equation given by

$$p_{2j+1} = \frac{p_{2j} + p_{2j+2}}{2} + w_j \left(\frac{p_{2j} + p_{2j+2}}{2} - \frac{p_{2j-2} + p_{2j+4}}{2} \right) \tag{8}$$

  [2.3] if (8) has no solution $(w_{j,x} \neq w_{j,y})$
    stop: the initial set of points belongs to none of the specified classes
  else
    store the value of w_j
[3] compare the values of w_j obtained for all j in the previous step:
  if $w_j = w$, $\forall j = 1, ..., n-2$
    [3.1] compute the tension parameter $v^0 = \frac{\sqrt{2w(2w+1)}}{4w} - \frac{1}{2}$
  else
    the initial set of points belongs to none of the specified curve types

The proposed procedure does not require to know a priori whether the initial points lie on a curve of a prescribed space. Moreover, by step 3, it is clear that, if it does not stop before, this method can come successfully to an end only when all the coefficients w_j stored at step 2.3 have the same value. This is always true if the initial points are sampled from a curve either in V_0, V_s or $V_{\mathrm{i}s}$, and, if this is the case, the algorithm allows to determine the tension v_0 necessary to reproduce the given curve. In particular, when the initial data belong either to V_s or $V_{\mathrm{i}s}$, such a tension value uniquely identifies the corresponding space through the parameter s defined as $s = \frac{\operatorname{acosh}(v^0)}{u}$ or $s = \frac{\operatorname{acos}(v^0)}{u}$ respectively.
On the other hand, by applying the proposed algorithm to points that belong to a curve of none of these spaces, the procedure could eventually yield a value of tension, even so, by applying the proposed scheme with such a tension, it is not

possible to predict anything about what kind of curve we will get.

Remark 9 *Taking a look at step 3. it is clear that, in order to possess all the initial data necessary for the computation described at this stage of the procedure, we need to produce at least two weights to compare. To this aim, we need to start from a minimal number of nine initial points. This is due to the fact that the described algorithm cannot be applied to compute the points* p_1 *and* p_{n-2}*, since we do not possess a well defined two-neighborhood around these points.*

4 A subdivision rule for curve endpoints

In case of open curves, rules (3)-(7) can be applied only on the interior of the curve, while for the endpoints we should include an alternative rule. Since the two endpoints can be treated analogously, it will be sufficient to address our attention only on one side.
To this aim we observe that, if we define just two auxiliary points p^0_{-2}, p^0_{-1} in the coarsest polygon $P^0 = \{p^0_j \mid j = 0, ..., n\}$, each new point p^{k+1}_{2j+1} has a well-defined 2-neighborhood and the open curve generated in the limit through (3)-(7) with $j = -1, ..., n$ trivially turns out to have p^0_0 as first endpoint and to be C^1-continuous.
However, the extension of this strategy to surface subdivision, implies the definition of two rings of points around the boundary control net. Thus, to avoid so many computations when subdividing the first edge $\overline{p^k_0, p^k_1}$, we propose here a special rule for computing the point p^{k+1}_1 independently of the two auxiliary points p^0_{-2}, p^0_{-1}; as said above, to work out the endpoint rule for the last edge, it will be sufficient to proceed analogously.
Let $\overline{p^k_0, p^k_1}$ be the first edge of the non-refined polygon $P^k = \{p^k_j \mid j = 0, ..., 2^k n\}$. Once defined an auxiliary point p^k_{-1}, we can compute the point p^{k+1}_1 through (3)-(7) with $j = 0$, applying subdivision to the subpolygon p^k_{-1}, p^k_0, p^k_1, p^k_2.
We choose here the following extrapolatory rule:

$$p^k_{-1} = 2p^k_0 - p^k_1 \tag{9}$$

since the three curve schemes (corresponding to the cubic, the hyperbolic and the trigonometric cases) are all capable of representing linear functions. In this way the additional refinement rule for the endpoint can be expressed as the following stationary linear combination of points from the non-extrapolated open polygon p^k_0, p^k_1, p^k_2:

$$p^{k+1}_1 = \left(\frac{1}{2} - w^{k+1}\right) p^k_0 + \left(\frac{1}{2} + 2w^{k+1}\right) p^k_1 - w^{k+1} p^k_2. \tag{10}$$

Proposition 10 *Rule (10) does not affect the convergence of the original scheme to a continuously differentiable limit.*

PROOF. It is sufficient to show that, taken $p^0_{-2} = 2p^0_0 - p^0_2$ and $p^0_{-1} = 2p^0_0 - p^0_1$, and refining the polygon P^0 through (3)-(7), after k rounds of subdivision the expression of the point p^k_{-1} turns out to coincide with (9). $\square$

5 Applications and examples

The following examples show open and closed curves which pass through a set of given points. The control polygons (corresponding to the piecewise linear curve between the given points) are drawn by a dashed line, and the smooth curves obtained by our algorithm by a full line.

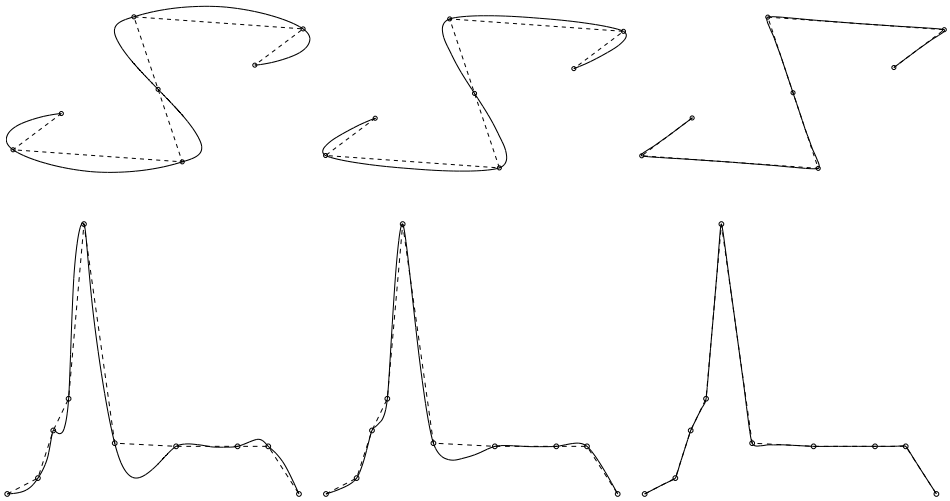

Fig. 3. Increasing the tightness (tension) of an open curve.

Figure 3, depicting the open limit curves obtained with $v^0 = -0.4,\ 1,\ 1000$, demonstrates the increase in the tightness (tension) of the curve with the increase in v^0.

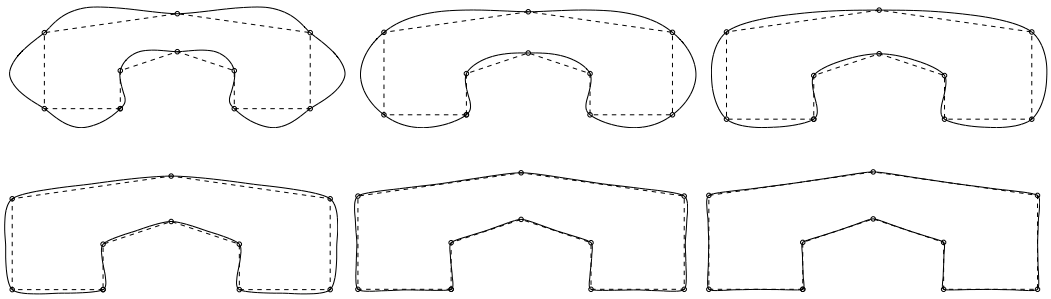

Fig. 4. Increasing the tightness (tension) of a closed curve.

Analogously, Figure 4, depicting the closed limit curves obtained with $v^0 = -0.5,\ 0,\ 1,\ 5,\ 50,\ 500$, demonstrates the increase in the tightness (tension) of the curve with

the increase in v^0.
The following figures show the effect of the tension parameter v^0 when our algorithm is applied on a regular N-sided control polygon inscribed in the unit circle.

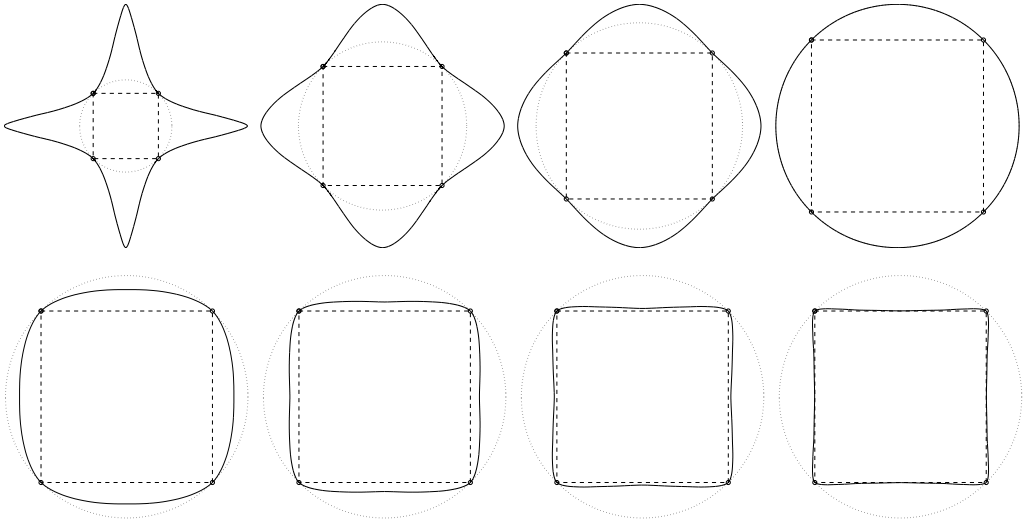

Fig. 5. Interpolation of the vertices of a square with the uniform tension controlled interpolating 4-point scheme defined by the following values of the parameter v^0: -0.95, -0.75, -0.5, 0, 1, 5, 25, 500.

While choosing $v^0 < 1$ the parameter acts as a looseness and, smaller it is, looser the limit curve is, for high values of the tension parameter v^0, the limit curve tends to shrink to the initial control polygon.
In addition, whenever we choose $v^0 = \cos(\frac{2\pi}{N})$, in the limit we obtain exactly the unit circle (see Figs. 5, 6).

6 Conclusions and Future Work

This paper describes a simple and efficient non-stationary subdivision scheme for curve interpolation depending on a single tension parameter, that is capable of reproducing conic sections exactly whenever such a parameter is chosen correctly. The subdivision algorithm (3)-(7) is actually an insertion algorithm since all the points at stage k are carried over to stage $k+1$ and new points are inserted in between the old ones. Evidently, the resulting limit curve interpolates the initial points. The local nature of the scheme, the possibility of reproducing cubic polynomials as well as certain classes of hyperbolic and trigonometric functions, and the control of the tension by the associated parameter, are important features for curve design. The algorithm proposed here is unique in combining these five ingredients: subdivision, locality, interpolation, global tension control, reproduction of conic sections.

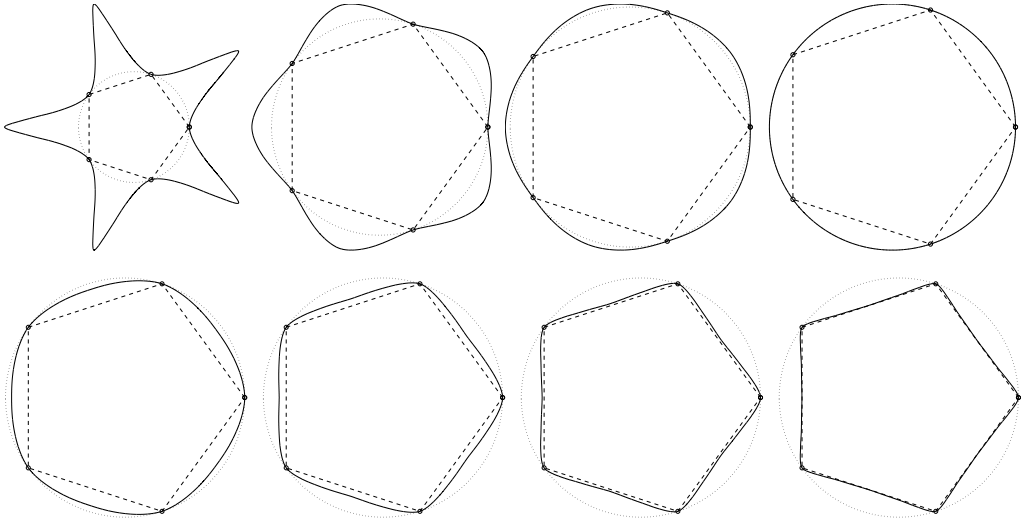

Fig. 6. Interpolation of the vertices of a regular pentagon with the uniform tension controlled interpolating 4-point scheme defined by the following values of the parameter v^0: -0.5, -0.25, 0, $\cos(\frac{2\pi}{5})$, 1, 5, 25, 500.

An interesting generalization of this proposal could include the possibility of working with a different tension parameter v^0 for each segment of the initial polygon P^0. In this way, since during each subdivision step each segment is split into two new segments, these two will inherit a new tension via equation (6). The resulting subdivision scheme will allow therefore different tensions on distinct curve segments. The curve scheme proposed can also be naturally extended to tensor-product surfaces. Next step will be therefore generalizing the univariate scheme to a surface scheme over arbitrary quadrilateral meshes.

Acknowledgements

This research was supported by MIUR-PRIN 2004 and by University of Bologna "Funds for selected research topics". Many thanks go to the anonymous reviewers for their helpful comments. The authors are also grateful to Nira Dyn for her precious suggestions.

References

Deslauriers, G., Dubuc, S., 1989. Symmetric iterative interpolation processes. Constr. Approx. 5, 49-68.

Dubuc, S., 1986. Interpolation through an iterative scheme. J. Math. Anal. Appl. 114, 185-204.

Dyn N., Levin D., Gregory J.A., 1987. A 4-point interpolatory subdivision scheme

for curve design. Computer Aided Geometric Design 4, 257-268.

Dyn, N., Levin, D., 1995. Analysis of asymptotically equivalent binary subdivision schemes. J. Math. Anal. Appl. 193, 594-621.

Dyn, N., 2002. Interpolatory subdivision schemes. In: Iske, A., Quak, E., Floater, M.S. (Eds.), Tutorials on Multiresolution in Geometric Modelling. Springer-Verlag, 25-50.

Dyn, N., Levin, D., Luzzatto, A., 2003. Exponentials Reproducing Subdivision Schemes. Found. Comput. Math. 3, 187-206.

Dyn, N., Floater, M.S., Hormann, K., 2005. A C^2 Four-Point Subdivision Scheme with Fourth Order Accuracy and its Extensions. In: Dæhlen, M., Mørken, K., Schumaker, L.L. (Eds.), Mathematical Methods for Curves and Surfaces: Tromsø 2004. Nashboro Press, 145-156.

Ivrissimtzis, I.P., Dodgson, N.A., Hassan, M.F., Sabin, M.A., 2002. On the geometry of recursive subdivision. Intern. J. Shape Modeling 8(1), 23-42.

Jena, M.K., Shunmugaraj, P., Das, P.C., 2003. A non-stationary subdivision scheme for curve interpolation. Anziam J. 44(E), 216-235.

Morin, G., Warren, J., Weimer, H., 2001. A subdivision scheme for surfaces of revolution. Computer Aided Geometric Design 18, 483-502.

Zhang, J., 1996. C-curves: an extension of cubic curves. Computer Aided Geometric Design 13, 199-217.