Interpolatory ternary subdivision

Talk given in the mathematics postgraduate seminar at Stellenbosch on 11 May 2012

M.D. van der Walt

First of all, let me describe the concept of subdivision. Suppose that we want to draw or design some curve or surface, using a computer graphic tool. We would like the curve or surface to be smooth, so as to be pleasing to the eye. So suppose we start with a sequence of points in $\mathbb{R}^s$, say $\mathbb{R}^2$, for example. We call these points control points; they are points chosen by the user to determine the shape of the desired curve. Now we want to add points iteratively to the picture. Then, with each iteration the points become more dense, until it resembles the desired curve. We call this curve the limit curve.

A lot of attention has been given to finding a way of doubling the number of points at each iteration, for example in the book Wavelet Subdivision Methods by Prof. De Villiers and Charles Chui. In their book, they define a way of doubling the number of points at each iteration, by shifting each existing point to a new position ((swart blokkie), and also adding one new point for each existing point (hollow circle). These processes are defined by taking combinations of neighbouring control points, using certain coefficients, or weights.

The limit curve we obtain in this way, has the parametric representation

$$ F_c(t) = \sum_j c_j \phi(t - j), $$

where the points $c_j$ are the initial control points, and it turns out that $\phi$, the basis function of the limit curve, is a refinable function. A refinable function is a function that can be generated from finitely many integer shifts of its scaled formulation, as in

$$ \phi(x) = \sum_j p_j \phi(kx - j), \quad x \in \mathbb{R}. $$

We say a function is 2-refinable or a function has refinement factor 2 if $k=2$; 2-refinable functions form the basis of the limit curve when we double the number of points at each iteration, as above. The sequence $\{p_j\}$ is a finitely-supported sequence, and it is called the refinement sequence, and the terms of the refinement sequence $\{p_j\}$ will form the
coefficients or weights when forming combinations of neighbouring points.

In my studies, I have focused my attention on finding ways to triple the number of points in our picture at each iteration. In other words, I want to define a subdivision method where each existing point is shifted to a new position, and two new points are added for each existing point. This means that we will rather be using 3-refinable functions as basis functions for the limit curve, that is, the refinement factor \( k = 3 \).

Throughout my work, I followed the guidelines concerning subdivision methods based on 2-refinable functions, trying to adapt the results for ternary subdivision methods. I will cover the concepts regarding binary subdivision schemes, and then show how it needs to be adapted for ternary subdivision. I have decided not to go into too much detail regarding the proofs of the results; I would rather like to go through the flow of the argument.

In practice, it is often necessary to require that the limit curve passes through the initial control points. For example, the control points could represent data points, and we do not want the data to be lost. This means we need to require that the initial control points are kept fixed, and not moved to a new position, as I indicated above. I have focused on this situation in my thesis. For this situation, we want to construct an interpolatory subdivision scheme, using interpolatory refinable functions as basis functions for the limit curve. An interpolatory refinable function has the extra requirement that

\[
\phi(j) = \delta_j, \quad (3)
\]

where \( \delta_j \) is the Kronecker delta sequence, that is,

\[
\delta_j := \begin{cases} 
1, & j = 0, \\
0, & j \in \mathbb{Z} \setminus \{0\}.
\end{cases} \quad (4)
\]

Now, our task is to show that there exists such an interpolatory refinable function to be used as basis function for the limit curve. We will first derive a necessary condition on the refinement sequence \( \{p_j\} \) for this interpolatory refinable function \( \phi \) to exist. Then we will need to show that this condition is also sufficient, that is, we will need to show that if the refinement sequence satisfies the condition, then there exists an interpolatory refinable function \( \phi \).

First considering binary subdivision schemes, we can prove that a necessary condition for \( \phi \) to be interpolatory and refinable, is

\[
p_{2j} = \delta_j, \quad (5)
\]

where \( \{p_j\} \) refers to the refinement sequence. A refinement sequence with this constraint is called an interpolatory refinement sequence. This is equivalent to saying

\[
P(z) + P(-z) = 1, \quad (6)
\]

2
where $P$, a Laurent polynomial, is defined by

$$P(z) := \frac{1}{2} \sum_j p_j z^j, \quad z \in \mathbb{C} \setminus \{0\}. \quad (7)$$

So, a necessary condition to find an interpolatory refinable function $\phi$ as basis function for the limit curve, is that the Laurent polynomial $P$ of the refinement sequence \{$p_j$\} solves this polynomial equation. Note that the coefficients of $z$ in the polynomial equation, $+1$ and $-1$, are the square roots of unity.

Moving over to ternary subdivision schemes, we have the same requirement for $\phi$ to be an interpolatory refinable function. This condition now implies

$$p_{3j} = \delta_j, \quad (8)$$

which is equivalent to saying

$$P(z) + P(\alpha z) + P(\alpha^2 z) = 1, \quad (9)$$

where the coefficients of $z$, namely $1, \alpha, \alpha^2$, now refer to the cube roots of unity, so that

$$\alpha := e^{\frac{2\pi i}{3}}, \quad (10)$$

and where $P$ is defined by

$$P(z) := \frac{1}{3} \sum_j p_j z^j, \quad z \in \mathbb{C} \setminus \{0\}. \quad (11)$$

So we need to find a Laurent polynomial $P$ that solves this polynomial equation.

Now, a natural requirement is that the subdivision scheme has a certain polynomial reproduction property, that is, if the original control points lie on a parametric straight line or parabola or some other polynomial, then the limit curve we obtain is also a parametric straight line or parabola, to preserve smoothness in some way; that is,

$$\sum_j f(j) \phi(t - j) = f(t). \quad (12)$$

We can prove that this is equivalent to saying that the Laurent polynomial $P$ has the form

$$P(z) = \left(\frac{1 + z}{2}\right)^m R(z), \quad (13)$$

for some $m \geq 2$, where $R$ is a polynomial such that

$$R(1) = 1, \quad R(-1) \neq 0. \quad (14)$$

So substituting this formulation into our polynomial equation, we obtain

$$\left(\frac{1 + z}{2}\right)^m R(z) + \left(\frac{1 - z}{2}\right)^m R(-z) = 1, \quad (15)$$
For ternary subdivision schemes, the polynomial reproduction property is equivalent to saying
\[ P(z) = \left(\frac{1+z+z^2}{3}\right)^m R(z), \]
for some \( m \geq 2 \), where \( R \) is a polynomial such that
\[ R(1) = 1, \quad R(\alpha) \neq 0, \quad R(\alpha^2) \neq 0. \]
so that
\[ \left(\frac{1+z+z^2}{3}\right)^m R(z) + \left(\frac{1+\alpha z+\alpha^2 z^2}{3}\right)^m R(\alpha z) + \left(\frac{1+\alpha^2 z+\alpha z^2}{3}\right)^m R(\alpha^2 z) = 1. \]

In other words, if we assume that \( P \) has the form (16), the necessary condition (9) has the equivalent form (18).

For binary subdivision schemes, we will find a polynomial \( R \in \pi_{m-1-\lfloor 2m-1/2 \rfloor} \) that solves the polynomial equation (15), while for ternary subdivision schemes, we will find a polynomial \( R \in \pi_{m-1-\lfloor 3m-1/2 \rfloor} \) that solves the polynomial equation (18). In both cases, we can show that this is the minimum degree polynomial that solves the polynomial equation.

Furthermore, in order to center the refinement sequence \( \{p_j\} \) obtained by solving the polynomial equation, we multiply the equation by \( z^d \), with
\[ d := \lfloor \frac{3m-1}{2} \rfloor, \]
to obtain
\[ \left(\frac{1+z+z^2}{2}\right)^m U(z) + (-1)^d \left(\frac{1-z}{2}\right)^m U(-z) = z^d, \]
with
\[ U(z) = z^d R(z). \]

For refinement factor 3, we multiply with \( z^d \) straight through as well, and then to keep the format the same, we need factors \( \alpha^d \) and \( \alpha^{2d} \) in the second and third terms, to obtain
\[ \left(\frac{1+z+z^2}{3}\right)^m U(z) + \alpha^{2d} \left(\frac{1+\alpha z+\alpha^2 z^2}{3}\right)^m U(\alpha z) + \alpha^d \left(\frac{1+\alpha^2 z+\alpha z^2}{3}\right)^m U(\alpha^2 z) = z^d, \]
with
\[ U(z) = z^d R(z). \]

So, the necessary condition on the refinement sequence \( \{p_j\} \) for the existence of the interpolatory refinable function \( \phi \), has the equivalent formulation: The polynomial \( U \), which will determine the Laurent polynomial \( P \), must satisfy this polynomial equation. And this on its own is an interesting mathematical problem: Can we find a polynomial \( U \) that solves a polynomial equation with this specific form?
To solve the polynomial equation for binary subdivision schemes, we use the Euclidean algorithm and Division algorithm to show that there exist polynomials $U$ and $V$ in $\pi_{m-1}$ such that

$$(\frac{1+z}{2})^m U(z) + (\frac{1-z}{2})^m V(z) = z^d,$$

(24)
after which we apply arguments based on the degrees of the polynomials involved to show that $V(z) = (-1)^d U(-z)$. For refinement factor 3, it is possible to show (using repeated applications of the Euclidean and Division algorithms) that there exist polynomials $U$, $V$ and $W$ in $\pi_{m-1}$ such that

$$(\frac{1+z+z^2}{3})^m U(z) + (\frac{1+\alpha z+\alpha^2 z^2}{3})^m V(z) + (\frac{1+\alpha^2 z+\alpha z^2}{3})^m W(z) = z^d,$$

(25)
but we could not find a way to show that $V(z) = \alpha^2 U(z)$ and $W(z) = \alpha U(z)$. So we needed to think of another way to find a polynomial $U$ that solves the polynomial equation.

Another approach is to define the polynomial $Q$ by

$$Q(z) := (\frac{1+z}{2})^m U(z) + (-1)^d (\frac{1-z}{2})^m U(-z) - z^d,$$

(26)
so that our problem has the equivalent formulation: Find a polynomial $U \in \pi_{m-1}$ such that

$$Q(z) := (\frac{1+z}{2})^m U(z) + (-1)^d (\frac{1-z}{2})^m U(-z) - z^d = 0.$$

(27)

Now, we step away from our first set of equivalent statements for the moment, and solve this new problem, using a similar method of deriving a necessary condition and proving that it is sufficient. It is interesting to note that we apply this type of reasoning twice in the argument: We have two instances where we want to prove the existence of some function – in the beginning, we want to prove the existence of an interpolatory refinable function $\phi$ to be used as basis function for the limit curve, and here we want to prove the existence of a polynomial $U$ that satisfies some polynomial equation. In both cases, we first derive a necessary condition to be satisfied by the functions $\phi$ and $U$. The necessary condition shows us which kind of functions are possible solutions to the problem. But then we also need to show that these solutions actually do solve the problem, that is, that they are sufficient. Then we have completed the circle.

So we first assume that there exists a polynomial $U \in \pi_{m-1}$ such that $Q = 0$, and we derive a necessary condition on $U$. If $Q = 0$, we can differentiate both sides of the equation $m - 1$ times, and put $z = 1$. We note that since the second term has a factor $(\frac{1-z}{2})^m$, all the derivatives up to $m - 1$ will contain one of these factors $(\frac{1-z}{2})$, so that all the derivatives at $z = 1$ will vanish, meaning we only have to derive the first term, using the Leibniz formula, and $z^d$, and set it equal to 0. In this way, we can find all the derivatives up to $m - 1$ of $U$, and by Taylor’s theorem we then have an expression for $U$, that is,

$$U(z) = \sum_{j=0}^{m-1} \frac{U^{(j)}(1)}{j!} (z - 1)^j,$$

(28)
with
\[ U^{(n)}(1) = \binom{d}{n} n! - \sum_{l=1}^{n} U^{(n-l)}(1) \binom{n}{l} (l!)^2 2^{-l}, \quad n = 0, \ldots, m - 1. \] (29)

So, we can show that if \( Q = 0 \), then \( U \) must be of this form. Now we need to show the converse. So we assume that \( U \) has the above formulation, with all the derivatives up to \( m - 1 \) of \( U \) at \( z = 1 \) calculated as I explained earlier. This will imply that
\[ Q^{(j)}(1) = 0, \quad j = 0, \ldots, m - 1; \] (30)

that is how we derived the formulation of \( U \). Now we note that if we replace \( z \) by \(-z\) in the definition of \( Q \), we obtain
\[ Q(-z) = (-1)^d Q(z). \] (31)

so that, by deriving the above equation, and since \( Q^{(j)}(1) = 0, \quad j = 0, \ldots, m - 1 \), we also have
\[ Q^{(j)}(-1) = 0, \quad j = 0, \ldots, m - 1. \] (32)

By combining these, we see that we have \( 2m \) Hermite interpolation conditions. Then, by the Hermite polynomial interpolation theorem, we know that there exists a unique polynomial in \( \pi_{2m-1} \) which satisfies these conditions. Now the zero polynomial satisfies these conditions, and \( Q \in \pi_{2m-1} \) also satisfies these conditions. So \( Q \) must be the zero polynomial.

So we have found a polynomial \( U \in \pi_{m-1} \) that satisfies the polynomial equation.

Now, this solution method can be adapted for the refinement factor 3 case. We define \( Q \) to be the polynomial
\[ Q(z) := \left( \frac{1+z+z^2}{3} \right)^m U(z) + \alpha^{2d} \left( \frac{1+\alpha z+\alpha^2 z^2}{3} \right)^m U(\alpha z) + \alpha^d \left( \frac{1+\alpha^2 z+\alpha z^2}{3} \right)^m U(\alpha^2 z) - z^d, \] (33)

so that our problem has the equivalent formulation: Find a polynomial \( U \in \pi_{m-1} \) such that
\[ Q(z) := \left( \frac{1+z+z^2}{3} \right)^m U(z) + \alpha^{2d} \left( \frac{1+\alpha z+\alpha^2 z^2}{3} \right)^m U(\alpha z) + \alpha^d \left( \frac{1+\alpha^2 z+\alpha z^2}{3} \right)^m U(\alpha^2 z) - z^d = 0. \] (34)

We first assume that there exists a polynomial \( U \in \pi_{m-1} \) such that \( Q = 0 \), and we derive a necessary condition on \( U \). If \( Q = 0 \), we can differentiate both sides of the equation \( m - 1 \) times, and put \( z = 1 \). We note that, since the second and third terms contain the factors \( \left( \frac{1+\alpha z+\alpha^2 z^2}{3} \right)^m \) and \( \left( \frac{1+\alpha^2 z+\alpha z^2}{3} \right)^m \), all the derivatives up to \( m - 1 \) will contain one of these factors, which vanish at \( z = 1 \), as \( \alpha^3 = 1 \). So we only need to
differentiate the first term, using the Leibniz formula and the binomial theorem, and \( z^d \), and set it equal to 0. In this way, we can find all the derivatives up to \( m - 1 \) of \( U \), and by Taylor’s theorem we then have

\[
U(z) = \sum_{j=0}^{m-1} \frac{U^{(j)}(1)}{j!} (z - 1)^j,
\]

with

\[
U^{(n)}(1) = \binom{d}{n} n! - \left( \frac{1}{3} \right)^n \sum_{l=1}^{n} \binom{n}{j} j! \sum_{j=0}^{m} \sum_{k=0}^{l} \binom{2j}{l-k} \binom{m-j}{k} (2)^{m-j-k} U^{(j)}(1),
\]

\( n = 0, \ldots, m - 1 \). (36)

So, we can show that if \( Q = 0 \), then \( U \) must be of this form. Now we need to show the converse. So we assume that \( U \) has the above formulation, with all the derivatives up to \( m - 1 \) of \( U \) at \( z = 1 \) calculated as I explained earlier. This will imply that

\[
Q^{(j)}(1) = 0, \quad j = 0, \ldots, m - 1;
\]

that is how we derived the formulation of \( U \). Now if we replace \( z \) by \( \alpha z \) and \( \alpha^2 z \) in the equation for \( Q \), we see that

\[
Q(\alpha z) = \alpha^d Q(z); \quad Q(\alpha^2 z) = \alpha^{2d} Q(z).
\]

This implies that we also have

\[
Q^{(j)}(\alpha) = 0, \quad j = 0, \ldots, m - 1, \quad Q^{(j)}(\alpha^2) = 0, \quad j = 0, \ldots, m - 1.
\]

By combining these, we see that we have \( 3m \) Hermite interpolation conditions. Then, by the Hermite polynomial interpolation theorem, we know that there exists a unique polynomial in \( \pi_{3m-1} \) which satisfies these conditions. Now the zero polynomial satisfies these conditions, and \( Q \in \pi_{3m-1} \) also satisfies these conditions. So \( Q \) must be the zero polynomial.

So we have found a polynomial \( U \in \pi_{m-1} \) that satisfies the polynomial equation.

To recap, we can now substitute the formula for \( U \) into the formula for \( R \), and the formula for \( R \) into the formula for \( P \), to obtain the solution to our polynomial equation. Now remember, this equation is a necessary condition for \( \phi \) to be an interpolatory refinable function. We have not proved that \( \phi \) exists yet. We still need to show that the subdivision scheme based on this specific refinement sequence we obtained is convergent, because only then are we guaranteed that a limit function \( \phi_p \) exists, and we need to show that this \( \phi_p \) is an interpolatory refinable function.
To prove convergence, we first need to define the cascade operator.

For binary subdivision, the cascade operator is defined by

\[(C_p f)(x) := \sum_j p_j f(2x - j), \quad x \in \mathbb{R}, \quad (40)\]

for any continuous function \( f \), and we define the cascade algorithm based on the cascade operator by

\[h_0 := h; \quad h_r := C_p h_{r-1}, \quad r = 1, 2, \ldots, \quad (41)\]

where \( h \) is the hat function, a continuous, interpolatory refinable function.

The theory of cascade operators translates to ternary subdivision; we define the cascade operator by

\[(C_p f)(x) := \sum_j p_j f(3x - j), \quad x \in \mathbb{R}, \quad (42)\]

for any continuous function \( f \), and the cascade algorithm is defined in the same way as for binary subdivision, that is,

\[h_0 := h; \quad h_r := C_p h_{r-1}, \quad r = 1, 2, \ldots. \quad (43)\]

We need the notion of the cascade algorithm, since we can prove that if the cascade algorithm converges to a certain limit function \( h_p \), then the corresponding subdivision scheme is convergent with limit function \( \phi_p := h_p \). Moreover, since the cascade operator preserves the refinability and interpolatory conditions of the initial hat function, we can prove that the limit function will be an interpolatory refinable function.

So, to prove that the interpolatory refinable limit function \( \phi_p \) exists, we need to prove that the cascade algorithm with respect to the refinement sequence \( \{p_j\} \) converges.

For binary subdivision schemes, this can be shown for a general (not necessarily interpolatory) subdivision scheme, by deriving a condition for the sequence of functions generated by the cascade algorithm, \( \{h_r\} \), to be a Cauchy sequence. Now we can prove that the support intervals of all the functions generated by the cascade algorithm will be contained in an interval \([a, b]\), and that all these functions are continuous. This implies that the Cauchy sequence \( \{h_r\} \) will converge with the maximum norm; so we know the cascade algorithm is convergent.

However, we could not adapt this condition for ternary subdivision schemes - the proof rests, among other things, on the use of trigonometric identities, which do not hold for refinement factor 3.

We decided to rather restrict our attention to interpolatory subdivision schemes, to exploit the interpolatory condition on the refinement sequence \( \{p_j\} \). We found another
sufficient condition for the sequence of functions generated by the cascade algorithm to be a Cauchy sequence. We define the polynomial $B_2$ by

$$B_2(z) = \sum_j b_j z^j := \frac{P(z)}{(1 + z)^2}. \quad (44)$$

If the value

$$\max \left\{ \sum_j |b_{2j}|, \sum_j |b_{2j+1}| \right\}$$

is less than 1, then the sequence $\{h_r\}$ will be a Cauchy sequence, and thus it will converge to some limit function $h_p$, so that the subdivision scheme is also convergent with limit function $\phi_p := h_p$, and $\phi_p$ is an interpolatory refinable function.

This idea can be adapted for ternary subdivision schemes. We define the polynomial $B_2$ by

$$B_2(z) = \sum_j b_j z^j := \frac{P(z)}{(1 + z + z^2)^2}. \quad (46)$$

If the value

$$\max \left\{ \sum_j |b_{3j}|, \sum_j |b_{3j+1}|, \sum_j |b_{3j+2}| \right\}$$

is less than 1, then the sequence $\{h_r\}$ will be a Cauchy sequence, and thus it will converge to some limit function $h_p$, so that the subdivision scheme is also convergent with limit function $\phi_p := h_p$, and $\phi_p$ is an interpolatory refinable function.

To conclude, we derived what the refinement sequences should look like to provide us with an interpolatory refinable function $\phi$, and we also showed that the subdivision scheme based on these refinement sequences do converge, so that the interpolatory refinable function $\phi$ does exist, completing the argument.

A graphical illustration of this result for ternary subdivision schemes, for the case when $m = 3$, can be seen in the figure below. The first picture displays the initial control points. At each iteration, we can see that two new points are added for each existing point, and the initial control points are kept fixed. After sufficiently many iterations, the subdivision scheme converges, and we obtain the desired limit curve, which passes through the control points.
Figure 1: Ternary subdivision for $m = 3$. 