

Local subdivision of Powell-Sabin splines

Hendrik Speleers

Paul Dierckx

Stefan Vandewalle

Report TW 424, March 2005



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Local subdivision of Powell-Sabin splines

Hendrik Speleers

Paul Dierckx

Stefan Vandewalle

Report TW 424, March 2005

Department of Computer Science, K.U.Leuven

Abstract

We present an algorithm for local subdivision of Powell-Sabin spline surfaces. The construction of such a spline is based on a particular PS-refinement of a given triangulation. We build the new triangulation on top of this PS-refinement by applying a $\sqrt{3}$ -subdivision scheme on a local part of the domain. To avoid degeneration we introduce a simple heuristic for refinement propagation, driven by a parameter. This parameter manages the trade-off between the mesh quality and the refinement localization.

Keywords : Powell-Sabin splines, subdivision, adaptive refinement, CAGD
AMS(MOS) Classification : Primary : 65D17, Secondary : 65D07

Local subdivision of Powell-Sabin splines

Hendrik Speleers, Paul Dierckx and Stefan Vandewalle

*Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium*

Abstract

We present an algorithm for local subdivision of Powell-Sabin spline surfaces. The construction of such a spline is based on a particular PS-refinement of a given triangulation. We build the new triangulation on top of this PS-refinement by applying a $\sqrt{3}$ -subdivision scheme on a local part of the domain. To avoid degeneration we introduce a simple heuristic for refinement propagation, driven by a parameter. This parameter manages the trade-off between the mesh quality and the refinement localization.

Keywords: Powell-Sabin splines, subdivision, adaptive refinement, CAGD
AMS classification: 65D07, 65D17, 68U07

1 Introduction

The tensor product B-spline representation [9, 3] is nowadays commonly used in many computer aided geometric design packages, because of its compactness, ease of implementation, and computational efficiency. With the aid of the tensor product B-spline control net surfaces can be locally adapted in a geometrically intuitive and flexible way. Tensor product splines have, however, a definite drawback: they are restricted to regular rectangular meshes. These limit the shape of the domain and often preclude an adequate adaptive mesh refinement.

An alternative is to consider piecewise polynomials on triangulations. Bernstein-Bézier polynomial patches [8] can be used to this purpose. Unfortunately, imposing smoothness conditions between the triangular patches usually results in a large number of non-trivial relations between the coefficients. Other candidates are Powell-Sabin (PS-)splines [15]. These C^1 -continuous quadratic splines can be represented in a compact normalized B-spline basis with an intuitive geometrical interpretation involving tangent control triangles [4]. These properties ensure their effectiveness in a wide range of application domains. Willemans [20] used Powell-Sabin splines for smoothing scattered data. Their application in surface modelling was analysed by Windmolders [21]. Powell-Sabin spline wavelets are developed in [23, 18]. Recently, Powell-Sabin splines were also applied as quasi-interpolants [14], and as finite elements [16] for solving partial differential equations.

A natural question that comes up in many such applications is how to represent a given function defined on a particular mesh on a refinement of that mesh. The procedures to do so are called subdivision schemes and allow, e.g., an increased local control for surface manipulation. The schemes introduced by Catmull and Clark [1], and by Doo and Sabin [6] marked the beginning of subdivision for surface modelling. Many more schemes like the ones by Loop [13] and by Kobbelt [10], the Butterfly [7] and the $\sqrt{3}$ -subdivision [11] scheme, and further variants have since become popular. Windmolders and Dierckx [22] solved the subdivision problem for uniform Powell-Sabin splines with a dyadic scheme, and Vanraes *et al.* presented a triadic subdivision scheme for general Powell-Sabin splines in [19]. Both schemes perform a global refinement at every level of

subdivision. When an increased surface resolution is only required in a small part of the surface, global subdivision may lead to excessive computational and storage costs. In such case, a locally adaptive subdivision strategy is called for. The development of such a strategy for Powell-Sabin splines has remained an open problem in the CAGD literature. In this paper, we address that problem and we derive a local subdivision scheme for general Powell-Sabin splines. It is based on a $\sqrt{3}$ -subdivision scheme to refine a local neighbourhood, and is combined with a refinement propagation heuristic in order to improve the quality of the refined triangulation.

The paper is organized as follows. Section 2 reviews some general concepts of polynomials on triangulations, and recalls the definition of the Powell-Sabin spline space. The section also covers the relevant aspects of the construction of a normalized B-spline basis and its Bernstein-Bézier representation. In the remaining sections we treat the local subdivision scheme. Section 3 shows how the triangulation is locally refined, and in section 4 we develop the corresponding subdivision rules. Finally, section 5 concludes with some remarks and points out possible application domains.

2 Powell-Sabin splines

2.1 Bivariate polynomials in the Bernstein-Bézier representation

Consider a non-degenerate triangle $\rho(V_1, V_2, V_3)$, defined by vertices V_i with Cartesian coordinates $(x_i, y_i) \in \mathbb{R}^2$, $i = 1, 2, 3$. Any point $(x, y) \in \mathbb{R}^2$ can be expressed in terms of its barycentric coordinates $\tau = (\tau_1, \tau_2, \tau_3)$ with respect to ρ . Let Π_m denote the linear space of bivariate polynomials of total degree less than or equal to m . Any polynomial $p_m(x, y) \in \Pi_m$ on the triangle ρ has a unique representation of the form

$$p_m(x, y) = b_\rho^m(\tau) = \sum_{|\lambda|=m} b_\lambda B_\lambda^m(\tau). \quad (2.1)$$

Here, $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ is a multi-index of length $|\lambda| = \lambda_1 + \lambda_2 + \lambda_3$, and

$$B_\lambda^m(\tau) = \frac{m!}{\lambda_1! \lambda_2! \lambda_3!} \tau_1^{\lambda_1} \tau_2^{\lambda_2} \tau_3^{\lambda_3} \quad (2.2)$$

is a Bernstein-Bézier polynomial defined on the triangle [8]. The coefficients b_λ are called the Bézier ordinates of $p_m(x, y)$. By associating each ordinate b_λ with the point $(\frac{\lambda_1}{m}, \frac{\lambda_2}{m}, \frac{\lambda_3}{m})$ in the triangle, we can display this Bernstein-Bézier representation schematically as in Figure 1 for the case $m = 2$. The points $(\frac{\lambda}{m}, b_\lambda)$ are called control points for the surface $z = b_\rho^m(\tau)$ and the piecewise linear interpolant at these points is the Bézier control net. This control net is tangent to the polynomial surface at the three vertices of the triangle.

Polynomials in their Bernstein-Bézier representation can be evaluated efficiently using the de Casteljau algorithm, i.e.,

$$b(\tau) = b_{0,0,0}^m(\tau) \quad (2.3a)$$

with

$$b_{\lambda_1, \lambda_2, \lambda_3}^0(\tau) = b_{\lambda_1, \lambda_2, \lambda_3}, \quad |\lambda| = m \quad (2.3b)$$

$$b_{\lambda_1, \lambda_2, \lambda_3}^r(\tau) = \tau_1 b_{\lambda_1+1, \lambda_2, \lambda_3}^{r-1}(\tau) + \tau_2 b_{\lambda_1, \lambda_2+1, \lambda_3}^{r-1}(\tau) + \tau_3 b_{\lambda_1, \lambda_2, \lambda_3+1}^{r-1}(\tau), \quad |\lambda| = m - r \text{ and } r = 1, \dots, m. \quad (2.3c)$$

The algorithm is numerically stable and has many interesting properties [8]. For example, the derivation of the continuity conditions on neighbouring triangles is straightforward.

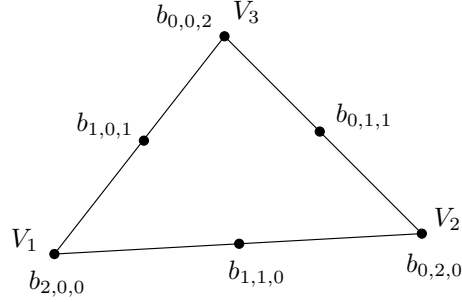


Figure 1: Schematic representation of a quadratic bivariate polynomial by means of its Bézier ordinates b_λ with $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ and $|\lambda| = 2$.

2.2 The Powell-Sabin spline space

Consider a simply connected subset $\Omega \in \mathbb{R}^2$ with polygonal boundary $\partial\Omega$. Assume a conforming triangulation Δ of Ω is given, consisting of t triangles $\rho_j, j = 1, \dots, t$, and having vertices V_k with Cartesian coordinates $(x_k, y_k), k = 1, \dots, n$. The Powell-Sabin refinement Δ^* of Δ partitions each triangle ρ_j into six smaller triangles with a common vertex Z_j . This partition is defined algorithmically as follows:

1. Choose an interior point Z_j in each triangle ρ_j , so that if two triangles ρ_i and ρ_j have a common edge, then the line joining Z_i and Z_j intersects the common edge at a point R_{ij} between its vertices.
2. Join each point Z_j to the vertices of ρ_j .
3. For each edge of the triangle ρ_j
 - (a) which belongs to the boundary $\partial\Omega$: join Z_j to an arbitrary point on that edge;
 - (b) which is common to a triangle ρ_i : join Z_j to R_{ij} .

Figure 2(a) displays a triangulation with 8 elements, and a corresponding PS-refinement containing 48 triangles. The space of piecewise quadratic polynomials on Δ^* with global C^1 -continuity is called the Powell-Sabin spline space:

$$S_2^1(\Delta^*) := \left\{ s \in C^1(\Omega) : s|_{\rho_j^*} \in \Pi_2, \rho_j^* \in \Delta^* \right\}. \quad (2.4)$$

Each of the $6t$ triangles resulting from the PS-refinement is the domain triangle of a quadratic Bernstein-Bézier polynomial, i.e., with $m = 2$ in equations (2.1) and (2.2). Powell and Sabin [15] proved that the following interpolation problem

$$s(V_l) = f_l, \quad \frac{\partial s}{\partial x}(V_l) = f_{x,l}, \quad \frac{\partial s}{\partial y}(V_l) = f_{y,l}, \quad l = 1, \dots, n. \quad (2.5)$$

has a unique solution $s(x, y) \in S_2^1(\Delta^*)$ for any given set of n $(f_l, f_{x,l}, f_{y,l})$ -values. It follows that the dimension of the Powell-Sabin spline space $S_2^1(\Delta^*)$ equals $3n$.

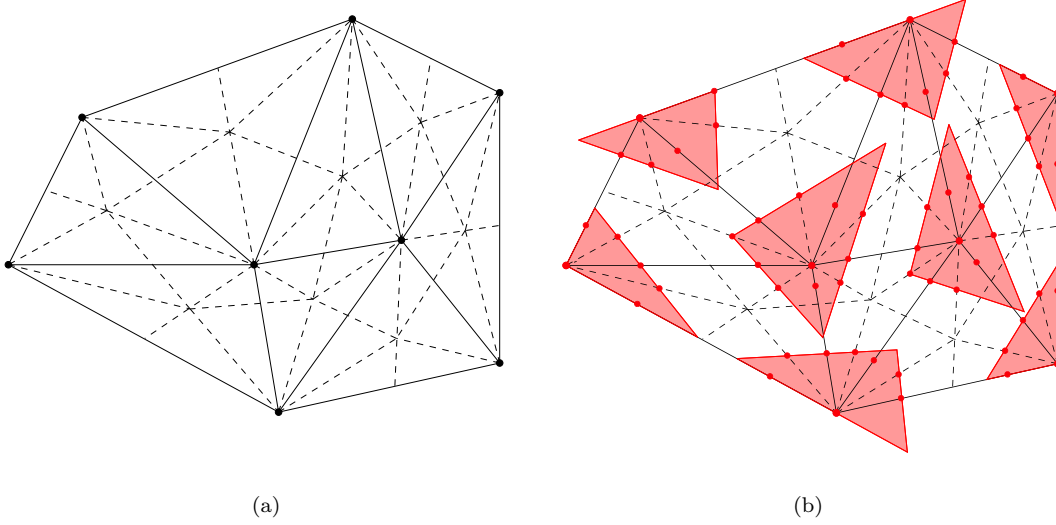


Figure 2: (a) A PS-refinement Δ^* (dashed lines) of a given triangulation Δ (solid lines); (b) the PS-points (bullets) and a set of suitable PS-triangles (shaded).

2.3 A normalized B-spline representation

A procedure for the construction of a locally supported basis for $S_2^1(\Delta^*)$ was developed in [4]. With each vertex V_i three linearly independent triplets $(\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$, $j = 1, 2, 3$ are associated. The basis function $B_i^j(x, y)$ can be found as the unique solution of the interpolation problem (2.5) with all $(f_l, f_{x,l}, f_{y,l}) = (0, 0, 0)$ except for $l = i$, where $(f_i, f_{x,i}, f_{y,i}) = (\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}) \neq (0, 0, 0)$. It is easy to see that this B-spline has a local support, because $B_i^j(x, y)$ vanishes outside the so-called molecule M_i of V_i , meaning the union of all triangles containing V_i . Every Powell-Sabin spline can then be represented as

$$s(x, y) = \sum_{i=1}^n \sum_{j=1}^3 c_{i,j} B_i^j(x, y). \quad (2.6)$$

The basis forms a convex partition of unity on Ω if

$$B_i^j(x, y) \geq 0, \quad \text{and} \quad \sum_{i=1}^n \sum_{j=1}^3 B_i^j(x, y) = 1, \quad (2.7)$$

for all $(x, y) \in \Omega$. This property, together with the local support of the Powell-Sabin B-splines, lies at the basis of their computational effectiveness for CAGD applications [21].

Dierckx [4] has presented a geometrical way to derive and construct such a normalized basis:

1. For each vertex $V_i \in \Delta$, identify the corresponding PS-points. Those are defined as the midpoints of all edges in the PS-refinement Δ^* containing V_i . The vertex V_i itself is also a PS-point. In Figure 2(b) the PS-points are indicated as bullets.
2. For each vertex V_i , find a triangle $t_i(Q_{i,1}, Q_{i,2}, Q_{i,3})$ that contains all the PS-points of V_i . Denote its vertices as $Q_{i,j}(X_{i,j}, Y_{i,j})$. The triangles t_i , $i = 1, \dots, n$ are called PS-triangles. We remark that the PS-triangles are not uniquely defined. Figure 2(b) shows some valid PS-triangles. One possibility for their construction [4] is to calculate a triangle of minimal area.

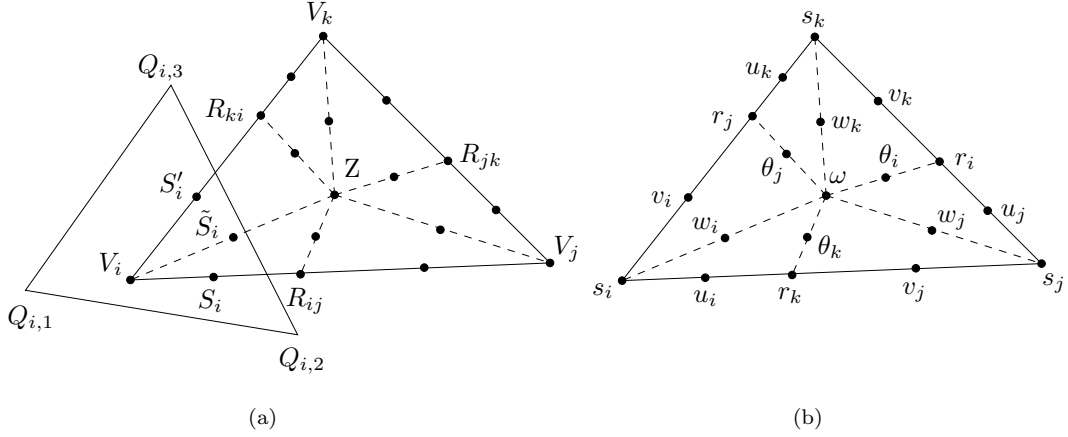


Figure 3: (a) PS-refinement of triangle $\rho(V_i, V_j, V_k)$ together with PS-triangle $t_i(Q_{i,1}, Q_{i,2}, Q_{i,3})$ of vertex V_i ; (b) schematic representation of the Bézier ordinates of a Powell-Sabin spline.

Computationally, this problem leads to a quadratic programming problem. An alternative solution is given in [17], where the sides of the PS-triangle are found by connecting two PS-points. From a practical point of view, other choices may be more appropriate. A particular choice of the PS-triangles can, e.g., simplify the treatment of boundary conditions [16] when PS B-splines are used in a finite element PDE solution procedure. For quasi-interpolation [14] the corners of the PS-triangle must be chosen on the edges of the triangulation.

3. The three linearly independent triplets $(\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$, $j = 1, 2, 3$ are derived from the PS-triangle t_i of a vertex V_i as follows:

$$\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}) \text{ are the barycentric coordinates of } V_i \text{ with respect to } t_i, \quad (2.8a)$$

$$\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3}) = (Y_{i,2} - Y_{i,3}, Y_{i,3} - Y_{i,1}, Y_{i,1} - Y_{i,2})/E, \quad (2.8b)$$

$$\gamma_i = (\gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3}) = (X_{i,3} - X_{i,2}, X_{i,1} - X_{i,3}, X_{i,2} - X_{i,1})/E, \quad (2.8c)$$

$$\text{where } E = \begin{vmatrix} X_{i,1} & Y_{i,1} & 1 \\ X_{i,2} & Y_{i,2} & 1 \\ X_{i,3} & Y_{i,3} & 1 \end{vmatrix}.$$

Note that $|\alpha_i| = 1$ and $|\beta_i| = |\gamma_i| = 0$. The fact that the PS-triangle t_i contains the PS-points of the vertex V_i guarantees the positivity property of (2.7). We define the control points as $C_{i,j} = (Q_{i,j}, c_{i,j})$ and the control triangles as $T_i(C_{i,1}, C_{i,2}, C_{i,3})$. One can easily prove that the control triangle T_i is tangent to the surface $z = s(x, y)$ at the vertex V_i .

2.4 The Bernstein-Bézier representation of a Powell-Sabin spline

Consider a domain triangle $\rho(V_i, V_j, V_k) \in \Delta$ with its PS-refinement Δ^* , and consider the points indicated in Figure 3(a), with the following barycentric coordinates: $V_i(1, 0, 0)$, $V_j(0, 1, 0)$, $V_k(0, 0, 1)$, $Z(a_i, a_j, a_k)$, $R_{ij}(\lambda_{ij}, \lambda_{ji}, 0)$, $R_{jk}(0, \lambda_{jk}, \lambda_{kj})$, and $R_{ki}(\lambda_{ik}, 0, \lambda_{ki})$. On each of the six triangles in Δ^* the Powell-Sabin spline (2.6) is a quadratic polynomial, that can be represented in its Bernstein-Bézier formulation by means of Bézier ordinates. In [5] the values of these Bézier ordinates are

derived. The outcome is schematically represented in Figure 3(b), with

$$s_i = \alpha_{i,1}c_{i,1} + \alpha_{i,2}c_{i,2} + \alpha_{i,3}c_{i,3} \quad (2.9a)$$

$$u_i = L_{i,1}c_{i,1} + L_{i,2}c_{i,2} + L_{i,3}c_{i,3} \quad (2.9b)$$

$$v_i = L'_{i,1}c_{i,1} + L'_{i,2}c_{i,2} + L'_{i,3}c_{i,3} \quad (2.9c)$$

$$w_i = \tilde{L}_{i,1}c_{i,1} + \tilde{L}_{i,2}c_{i,2} + \tilde{L}_{i,3}c_{i,3} \quad (2.9d)$$

As was shown in [4], the values $(\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3})$, $(L_{i,1}, L_{i,2}, L_{i,3})$, $(L'_{i,1}, L'_{i,2}, L'_{i,3})$ and $(\tilde{L}_{i,1}, \tilde{L}_{i,2}, \tilde{L}_{i,3})$ are the barycentric coordinates of the PS-points V_i , S_i , S'_i , and \tilde{S} with respect to the PS-triangle $t_i(Q_{i,1}, Q_{i,2}, Q_{i,3})$. The other Bézier ordinates can be found from the continuity conditions of the Powell-Sabin spline, e.g.,

$$r_k = \lambda_{ij}u_i + \lambda_{ji}v_j \quad (2.9e)$$

$$\theta_k = \lambda_{ij}w_i + \lambda_{ji}w_j \quad (2.9f)$$

$$\omega = a_iw_i + a_jw_j + a_kw_k \quad (2.9g)$$

In this Bernstein-Bézier representation Powell-Sabin splines can easily be manipulated using the de Casteljau algorithm (2.3).

3 Local refinement of Powell-Sabin triangulations

Subdivision produces the new B-spline representation (2.6) of a Powell-Sabin surface on a refinement of the given triangulation. In this section we propose a local refinement strategy using the $\sqrt{3}$ -refinement scheme. Afterwards, in section 4, we determine the corresponding control triangles so as to preserve the original surface.

3.1 Local refinement scheme

A central difficulty in any adaptive refinement is the joining of triangles from different refinement levels such that the resulting triangulation is conforming. The $\sqrt{3}$ -subdivision scheme [11, 12] handles this requirement in a natural way. The scheme proceeds as follows:

1. Split every triangle into three subtriangles by inserting a new vertex V_{ijk} inside the old triangle $\rho(V_i, V_j, V_k)$, and connect it to the surrounding old vertices. For example, in the PS-case the new vertex could be located at the interior point Z_{ijk} .
2. Flip each edge adjacent to two refined triangles of the original mesh in order to rebalance the new triangulation. These edges connect now two new vertices instead of two original vertices.

These two steps are illustrated in Figure 4. From the construction it follows that the refined triangulation does not preserve the original edges except at the boundary. Yet, if the new vertex V_{ijk} is chosen as the interior point Z_{ijk} of the PS-refinement, the new edges still belong to the old PS-refinement. When we apply the $\sqrt{3}$ -refinement scheme twice, we obtain a triadic split, as shown in Figure 5. Every original edge is trisected and each original triangle is split into nine subtriangles. Vanraes *et al.* [19] developed a global triadic subdivision scheme for a Powell-Sabin surface based on the $\sqrt{3}$ -refinement. Here, we will use the $\sqrt{3}$ -refinement to obtain a local subdivision scheme.

In order to make subdivision possible, the interior points of the $\sqrt{3}$ -refined triangulation must be chosen such that the new PS-refinement contains the edges of the original PS-refinement.

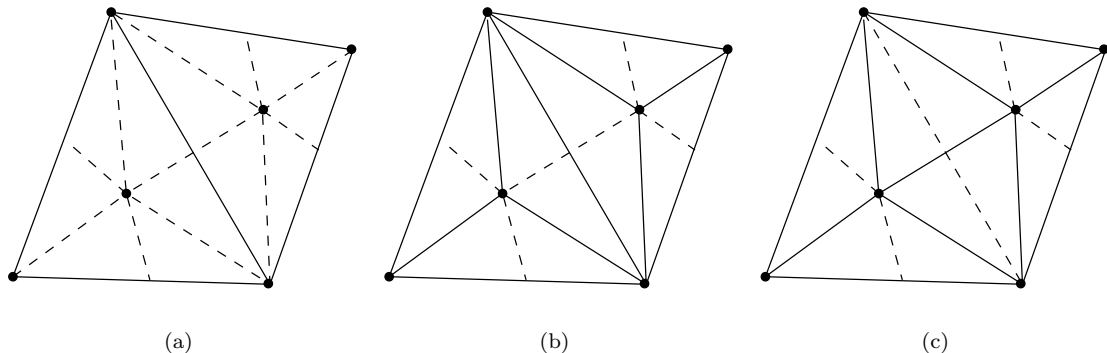


Figure 4: Principle of $\sqrt{3}$ -subdivision. (a) PS-refinement of two neighbouring triangles. (b) Place a new vertex at the position of the interior points and connect with the triangle corners. (c) Flip the edge adjacent to the two refined triangles.

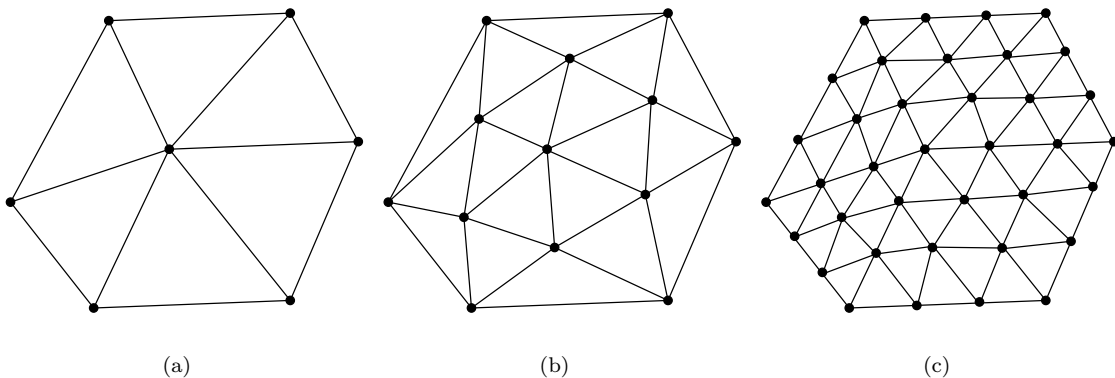


Figure 5: Applying $\sqrt{3}$ -subdivision twice results in triadic subdivision.

We illustrate our procedure by means of the example displayed in Figure 6. The original triangulation consists of the three triangles $\rho(V_i, V_j, V_l)$, $\rho(V_j, V_k, V_l)$ and $\rho(V_i, V_l, V_m)$. The $\sqrt{3}$ -refinement scheme is applied to the first two triangles, and adds the vertices V_{ijl} and V_{jkl} . Afterwards, the boundary triangle $\rho(V_k, V_l, V_{jkl})$ is bisected by inserting vertex V_{kl} at the position of point R_{kl} on the boundary edge V_k - V_l . This leads to a refined triangulation with 8 triangles, as indicated by the solid lines in Figure 6.

We now detail the construction of a compatible PS-refinement for the refined triangulation. We need to distinguish between three different types of triangles. First, we consider the new triangles generated in the second step of the $\sqrt{3}$ -refinement scheme, i.e., those containing a ‘flipped’ edge. Their interior points must be situated on the edge of the original triangulation. For instance, the interior point of triangle $\rho(V_j, V_{jkl}, V_{ijl})$ should be written as $Z_{j11} = \omega_{j11}V_j + (1 - \omega_{j11})R_{jl}$ with $0 < \omega_{j11} < 1$. The remaining new triangles lie at the boundary of the locally refined domain. For the new triangles containing two original vertices, the interior points must be chosen at the only edge of the original PS-refinement inside the considered triangles. For instance, triangle $\rho(V_i, V_j, V_{ijl})$ has as interior point $Z_{ij} = \omega_{ij}V_{ijl} + (1 - \omega_{ij})R_{ij}$ with $0 < \omega_{ij} < 1$. The interior points of the new triangles with a single original vertex can be chosen freely, i.e., Z_{kl1} in $\rho(V_k, V_{kl}, V_{jkl})$.

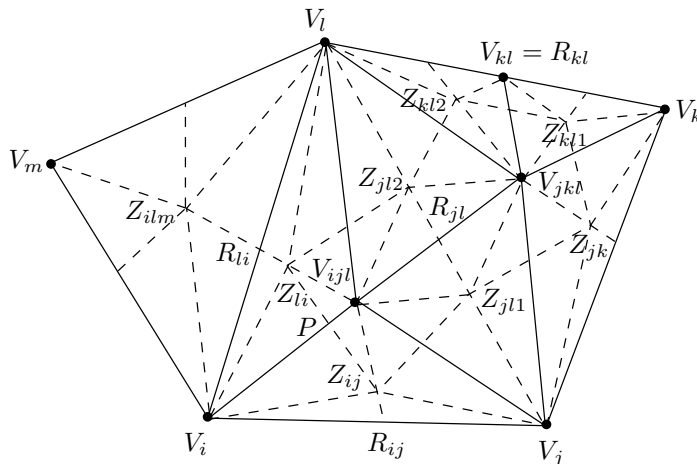


Figure 6: PS-refinement of a locally $\sqrt{3}$ -refined triangulation with additional bisection of boundary triangle $\rho(V_k, V_i, V_jkl)$.

The $\sqrt{3}$ -subdivision scheme can be applied to any subset of triangles without propagating outside this subset. However, the refined triangulation and its PS-refinement may contain poorly shaped triangles with small angles. Well-conditioned triangles are important, e.g., in finite element methods where error bounds depend on the minimal angle of the triangulation. Two undesirable situations can occur:

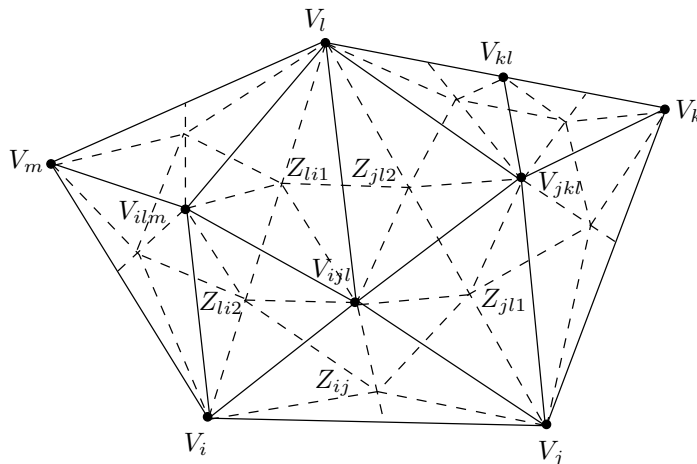
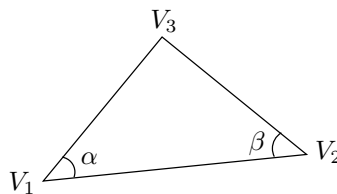
1. The new triangles at the boundary of the local refinement have too small angles. For example, in Figure 6, the new triangle $\rho(V_i, V_{ijl}, V_l)$ has smaller angles at the boundary edge V_i - V_l than the old triangle $\rho(V_i, V_j, V_l)$. After a few refinement levels, these angles may become unacceptably small.
2. The constrained position of the new interior points may be problematic. Imagine an initial triangulation, obtained by moving vertex V_i in the triangulation of Figure 6 to the right. Then triangle $\rho(Z_{ij}, V_{ijl}, Z_{li})$ becomes narrower. At some point no valid PS-refinement may be possible anymore, i.e., when the line Z_{ij} - Z_{li} does not intersect the edge V_i - V_{ijl} .

To deal with these problems, we will look for an expansion of the refinement domain where the $\sqrt{3}$ -refined triangles remain acceptable. We propose a simple refinement propagation heuristic driven by one parameter.

3.2 A simple heuristic for refinement propagation

To decide whether a triangle $\rho(V_1, V_2, V_3)$ is undesirable, we will set up a criterion based on a function $\mathcal{G}(\rho)$ that measures the quality of the shape. The function must be independent of the scale. We propose to use a value proportional to the area of the triangle divided by the square of the length of the longest side. This measure is easy to calculate and is related to the small angles of the triangle (see Figure 8):

$$\mathcal{G}(\rho) = \frac{4}{\sqrt{3}} \frac{\text{area}(\rho)}{\|V_2 - V_1\|^2} = \frac{2}{\sqrt{3}} \left(\frac{1}{\tan \alpha} + \frac{1}{\tan \beta} \right)^{-1} \quad (3.1)$$

Figure 7: PS-refinement of a locally $\sqrt{3}$ -refined triangulation (continued).Figure 8: The function $\mathcal{G}(\rho)$ measures the quality of triangle $\rho(V_1, V_2, V_3)$.

with α and β the angles adjacent to the longest edge V_1 - V_2 of triangle ρ . A low value of \mathcal{G} indicates the presence of small angles, as depicted in Figure 9. We can easily verify that $0 < \mathcal{G} \leq 1$. The upper bound is attained in the case of equilateral triangles.

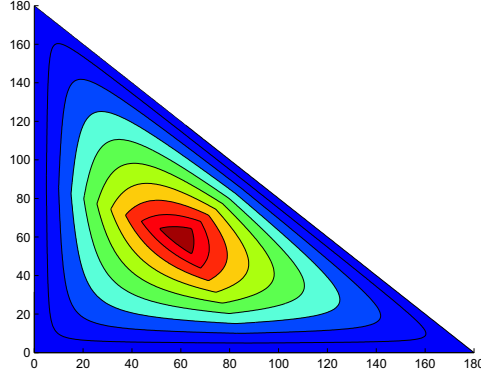
We propose now a heuristic for refinement propagation based on \mathcal{G} , to deal with both undesirable situations mentioned in section 3.1. Referring to Figure 6, we will split the neighbour of triangle $\rho(V_i, V_j, V_l)$ adjacent to the side V_i - V_l if

$$\min\{\mathcal{G}(\rho(V_i, V_l, V_{ijl})), \mathcal{G}(\rho(Z_{li}, Z_{ij}, V_{ijl})), \mathcal{G}(\rho(Z_{li}, Z_{jl2}, V_{ijl}))\} < \delta, \quad (3.2)$$

with $0 < \delta < 1$ a user-defined threshold parameter. A high value of δ will generate well-conditioned triangles, but will spread out the refinement. A low value will keep the locality at the cost of leaving poorly shaped triangles.

Referring to Figures 6 and 7, we explain how the propagation heuristic copes with the mentioned undesirable situations. The inequality (3.2) tests whether triangles $\rho(V_i, V_l, V_{ijl})$, $\rho(Z_{li}, Z_{ij}, V_{ijl})$, and $\rho(Z_{li}, Z_{jl2}, V_{ijl})$ are poorly shaped. By splitting the neighbouring triangle $\rho(V_i, V_l, V_m)$ in Figure 6, we can improve the quality of the three triangles. The edge V_i - V_l is flipped, and two new triangles $\rho(V_i, V_{ijl}, V_{ilm})$ and $\rho(V_l, V_{ilm}, V_{ijl})$ are created, as shown in Figure 7. Both triangles have a more favorable shape than $\rho(V_i, V_l, V_{ijl})$. The triangles in the PS-refinement are also improved, e.g., triangle $\rho(Z_{li}, Z_{ij}, V_{ijl})$ is replaced by $\rho(Z_{ij}, Z_{li2}, V_{ijl})$. When this triangle is not yet good enough, one may also decide to split the triangle adjacent to the side V_i - V_j .

We can characterize the latter improvement by the barycentric coordinates of $P(\eta, 0, 1 - \eta)$ with respect to triangle $\rho(V_i, V_j, V_{ijl})$, where P is the intersection point of the lines V_i - V_{ijl} and Z_{ij} - Z_{li} (see Figure 6). Assume that the interior points of $\rho(V_i, V_j, V_{ijl})$ and $\rho(V_l, V_i, V_{ijl})$ are chosen

Figure 9: Contour plot of $\mathcal{G}(\rho)$ in function of two angles of triangle ρ .

with the same ω_i , i.e., $Z_{ij} = \omega_i V_{ijl} + (1 - \omega_i) R_{ij}$ and $Z_{li} = \omega_i V_{ijl} + (1 - \omega_i) R_{li}$. If we include the two neighbouring triangles into the refinement domain, the considered interior points become $Z_{ij1} = \omega_i V_i + (1 - \omega_i) R_{ij}$ and $Z_{li2} = \omega_i V_i + (1 - \omega_i) R_{li}$. Calculating the barycentric coordinates of the new intersection point $P^*(\eta^*, 0, 1 - \eta^*)$ results in the relation $\eta^* = \eta + \omega_i$. The increased value of η^* will guarantee that the resulting triangle is less narrow. It follows that by means of a good choice of ω_i and the split of the adjacent triangles, we can cope with the undesirable situation.

Finally, we state the refinement propagation strategy algorithmically. Suppose we want to refine a triangulation Δ by splitting all triangles in $\text{SUBSET} \subset \Delta$. The strategy iteratively enlarges the subset in order to try to satisfy the imposed quality constraints. We calculate the quality function \mathcal{G} for two types of triangles: `TYPE1` stands for triangles in the original PS-refinement, e.g., triangle $\rho(V_i, V_l, V_{ijl})$ in Figure 6, and `TYPE2` for triangles in the new PS-refinement, e.g., triangle $\rho(Z_{li}, Z_{ij}, V_{ijl})$. The refinement propagation algorithm works as follows:

1. For each boundary edge of a triangle in `SUBSET`: prepare the decision on whether to include the neighbouring triangle into the refinement region by calculating the \mathcal{G} -values of three triangles. These triangles can be identified in analogy to the ones in (3.2) for triangle $\rho(V_i, V_j, V_l)$. One of these triangles is of `TYPE1`; two are of `TYPE2`. Add the values to the list `G`.
2. (*Propagation*) While `G` contains \mathcal{G} -values smaller than δ :
 - (a) Determine the element with the minimal \mathcal{G} -value in `G`.
 - (b) If the element is of `TYPE1` then:
 - When the neighbouring triangle exists and is not yet in `SUBSET`:
 - Add the neighbouring triangle to `SUBSET`
 - Choose ω_i
 - Update `G`, by adding the \mathcal{G} -values associated with the new boundary edges.
 - (c) Else:
 - When at least one of the two adjacent triangles exists and is not yet in `SUBSET`:
 - Add the valid adjacent triangles to `SUBSET`
 - Choose ω_i
 - Update `G`, by adding the \mathcal{G} -values associated with the new boundary edges.
 - (d) Remove the element from `G`.
3. Apply the $\sqrt{3}$ -refinement scheme to the triangles of `SUBSET`.

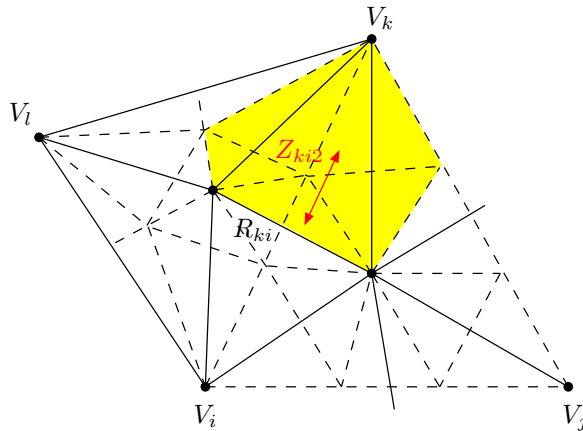


Figure 10: The choice of ω_i for the interior point $Z_{ki2} = \omega_i V_k + (1 - \omega_i) R_{kii}$ influences the quality of the shaded triangles in the PS-refinement.

Some remarks are in order here:

1. Although the algorithm strives to generate triangles that are as equilateral as possible, it cannot guarantee that the imposed quality requirement is satisfied everywhere. In particular, the quality of the locally refined mesh is bounded by that of the original triangulation and its PS-refinement.
2. The refinement strategy is guaranteed to terminate in a finite number of steps. The propagation definitively stops when SUBSET contains all triangles of the given triangulation.
3. Further fine tuning of the algorithm is possible, for example in step (2c): split only one adjacent triangle if that already leads to acceptable boundary triangles.
4. The quality of the result is influenced by the value of ω_i . Finding the best value is a hard global optimization problem which is not attractive to solve. Figure 10 shows the triangles in the PS-refinement that are directly influenced by the choice of a single ω_i . For simplicity, we can take the fixed value $\omega_i = 1/3$, as proposed in [19], which is inspired by the case of equilateral triangles. Otherwise, we could think of simplifying the general optimization problem by varying only one interior point at a time, while fixing the positions of the surrounding interior points, e.g. with $\omega_i = 1/3$.

3.3 Refinement near the boundary

When the refinement propagation reaches a domain boundary triangle, we cannot flip the boundary edge because the triangle has no neighbour. One possibility is to bisect the triangle by adding a new vertex at the boundary side, e.g., vertex V_{kl} in Figure 6, which must coincide with the PS-refinement point R_{kii} . With this approach, the boundary angle remains as small as it was, but the new triangle is not so wide anymore.

An alternative solution is the creation of an artificial vertex outside the domain. That allows to flip the boundary edge and to form two well-conditioned triangles. On a further refinement level it is possible to remove this artificial vertex. In Figure 11 the artificial vertex V_{ij}^a is added to the mesh of Figure 6. To enable a flip of edge V_i-V_j , vertex V_{ij}^a must be chosen on the line $V_{ijl}-R_{ij}$. The new triangles $\rho(V_i, V_{ij}^a, V_{ijl})$ and $\rho(V_j, V_{ijl}, V_{ij}^a)$ lie partly outside of the domain. When we

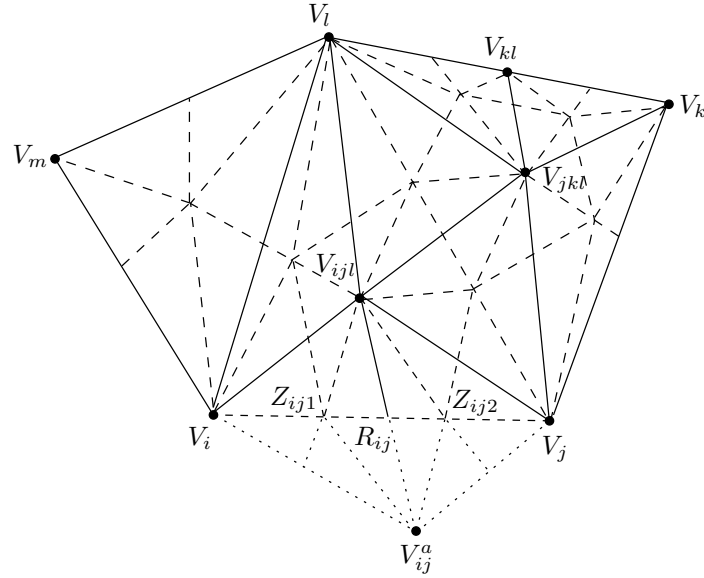


Figure 11: Splitting at the boundary side V_i-V_j introduces an artificial vertex V_{ij}^a outside the boundary, chosen at the line $V_{ijl}-R_{ij}$. The PS-refinement is indicated with dashed lines.

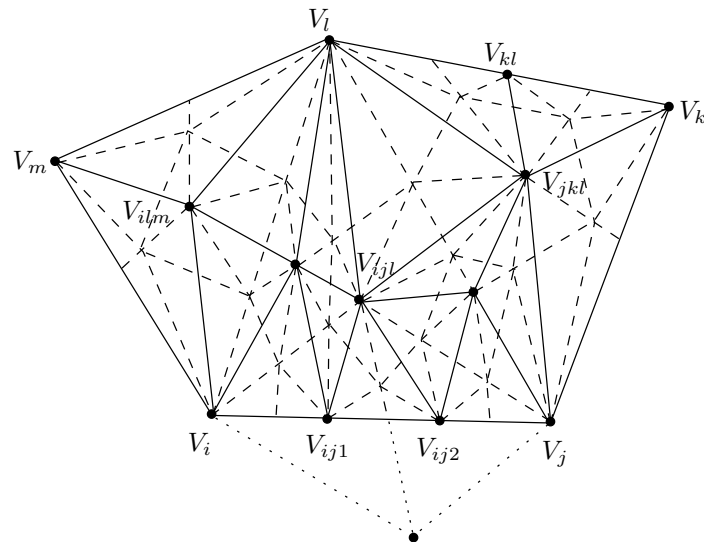


Figure 12: A new refinement at the boundary removes the artificial vertex V_{ij}^a from Figure 11.

number of vertices					
level	global (triadic)	local refinement			
		$\delta = 0.6$	$\delta = 0.3$	$\delta = 0.2$	$\delta = 0.1$
0	21	21	21	21	21
1		61	34	29	27
2	154	154	65	44	33
3		460	91	56	39
4	1297	1297	150	69	45
5		3889	206	84	51
6	11422	11422	285	95	57
7		34264	461	107	63
8	102061	102061	903	121	69

Table 1: Number of vertices for the triangulations successively refined by the global triadic subdivision scheme, and by the vertex-centered local subdivision scheme with different δ -values.

overall minimal angle/mean minimal angle					
level	global (triadic)	local refinement			
		$\delta = 0.6$	$\delta = 0.3$	$\delta = 0.2$	$\delta = 0.1$
0	26.6/37.4	26.6/37.4	26.6/37.4	26.6/37.4	26.6/37.4
1		18.9/39.4	17.1/37.6	17.1/36.3	12.5/35.8
2	20.6/39.4	20.6/39.4	17.1/35.5	16.7/34.0	8.7/34.5
3		18.9/39.5	17.1/36.1	16.7/34.3	8.7/34.2
4	15.7/39.3	15.7/39.3	17.1/35.6	15.2/33.5	8.7/33.4
5		15.8/39.5	17.1/36.2	14.8/33.9	8.7/33.1
6	12.8/39.2	12.8/39.2	17.1/36.2	14.8/33.5	8.7/32.6
7		13.0/39.2	17.1/36.3	14.2/33.4	8.7/32.1
8	11.0/39.1	11.0/39.1	17.1/36.2	14.2/33.2	8.7/31.7

Table 2: Overall minimal angle and the mean of the minimal angle of the triangles, for successively refined triangulations by the global triadic subdivision scheme and by the vertex-centered local subdivision scheme with different δ -values.

refine these triangles again, as in Figure 12, the artificial vertex V_{ij}^a is not needed anymore. Note that this strategy is reminiscent of a classical construction of one-dimensional B-splines near an interval boundary. Also there, knots may be defined outside the domain [2].

3.4 A numerical example

For the numerical example, we will assume that an initial subset of triangles has been marked for refinement. In data fitting and finite element applications, this subset is typically determined by an error analysis. For CAGD it is more common to use a vertex-centered approach, i.e., all triangles in the molecule of a selected vertex are put in the refinement set. The choice of a proper δ -value in (3.2) may be inspired by the application. In CAGD a small δ -value may be sufficient if a PS-spline has to be locally refined. When solving a PDE with finite elements, the error is often spread out over a large part of the domain. Then, a larger value for δ is advisable.

The vertex-centered refinement scheme is illustrated in Figures 14 and 15. The initial mesh is taken from [5] and is shown in Figure 13. We applied the scheme to the central vertex $V(0.5, 0.5)$. When the refinement propagation reaches the boundary, we used artificial vertices. In Figure 14,

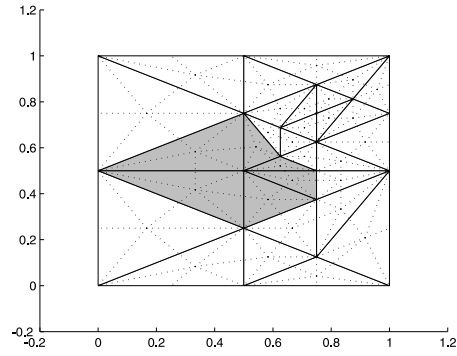
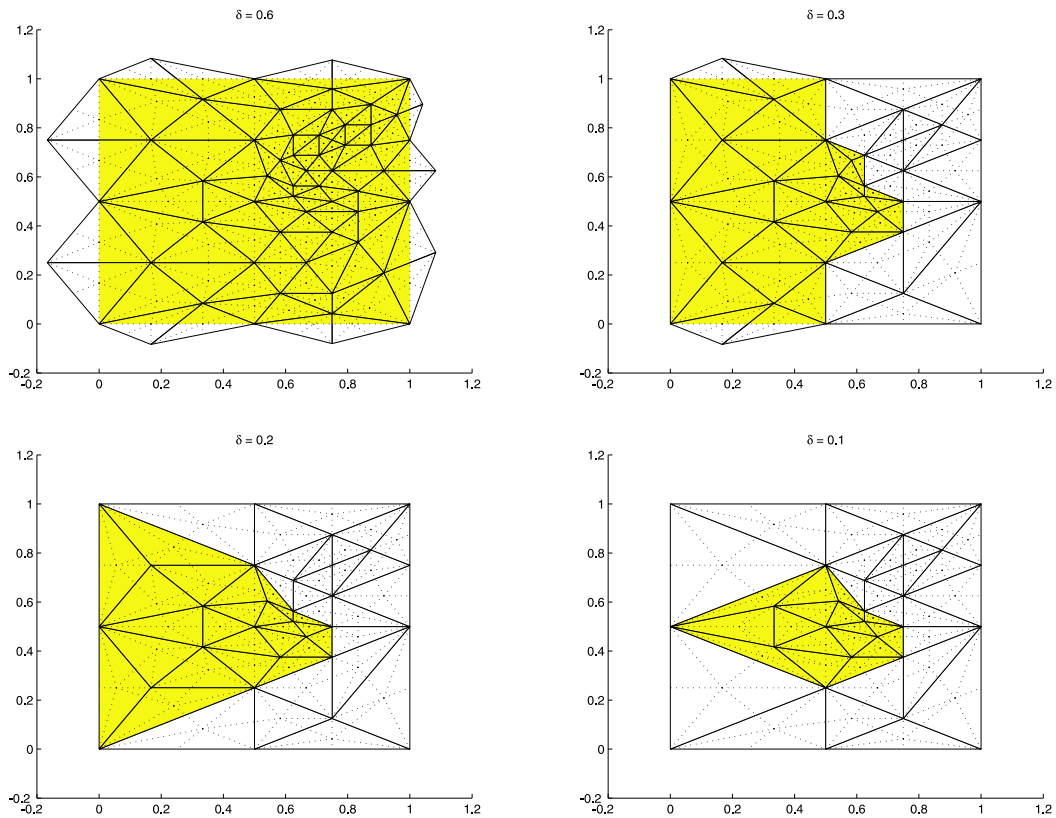


Figure 13: Initial triangulation, with the shaded triangles marked for refinement.

Figure 14: Example of the vertex-centered adaptive refinement scheme applied to vertex $V(0.5, 0.5)$ of the mesh in Figure 13, with different values for δ . The refined triangles are shaded, and the PS-refinement is indicated with dotted lines.

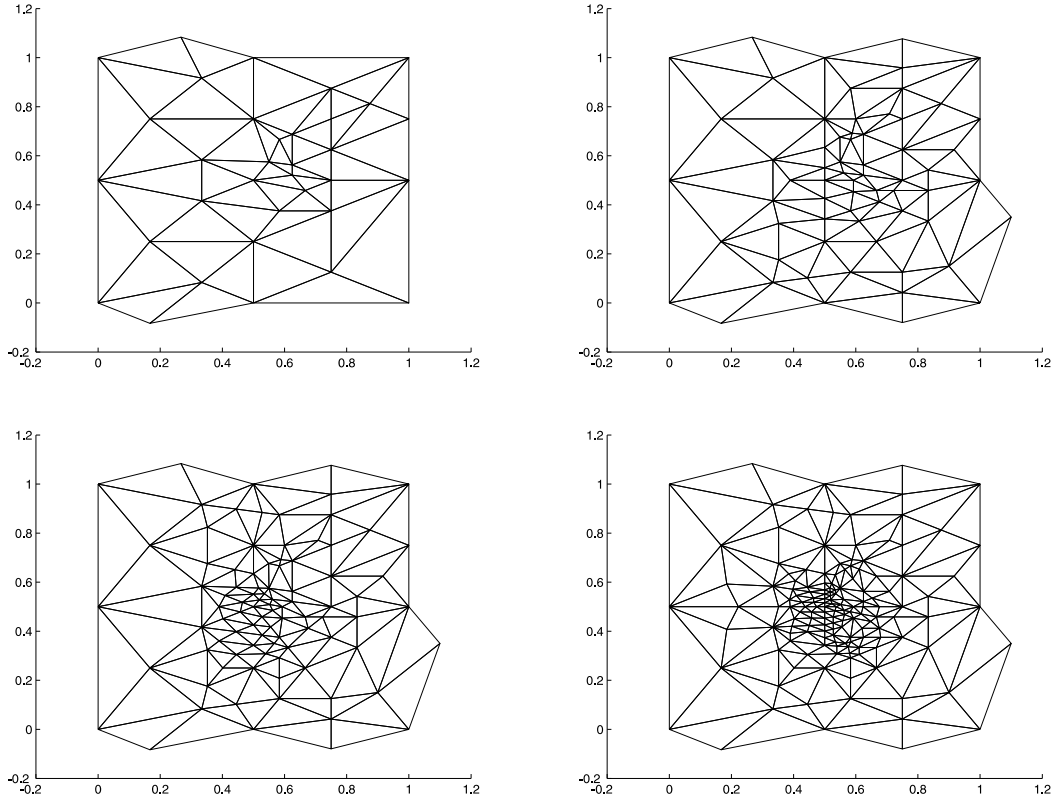


Figure 15: Example of the vertex-centered adaptive refinement scheme applied four times to vertex $V(0.5, 0.5)$ of the mesh in Figure 13 with $\delta = 0.3$.

δ is varied to show its influence. The small value $\delta = 0.1$ only refines the molecule of the vertex at the centre of the mesh; larger values of δ lead to the refinement of more triangles. With $\delta = 0.6$ the whole triangulation is refined. In Figure 15 we apply the vertex-centered scheme successively to the same vertex with the value $\delta = 0.3$. The dimensions of the successively refined Powell-Sabin spline spaces are 63, 102, 195, 273 and 450. In Table 1 we compare the growth of the number of vertices for the global triadic scheme from [19] with our local scheme. Table 2 gives the minimal angle of the corresponding meshes, and the mean value of the minimal angle in the triangulations.

We observe that two refinement levels with $\delta = 0.6$ are similar to one global triadic refinement level. This explains the appearance of the identical value sets in the second and third columns of Table 2. A small value of δ strongly limits the propagation of the refinement, at the cost of a small minimal angle of the triangulation. Larger values of δ enlarge the minimal angle of the mesh, up to a certain value, but generates more vertices. The minimal angle of a globally $\sqrt{3}$ -refined triangulation is an indication of the attainable improvement. However, some choices of δ can do better, e.g., in the eighth refinement level $\delta = 0.3$ leads to a much better minimal angle than $\delta = 0.6$ (see Table 2).

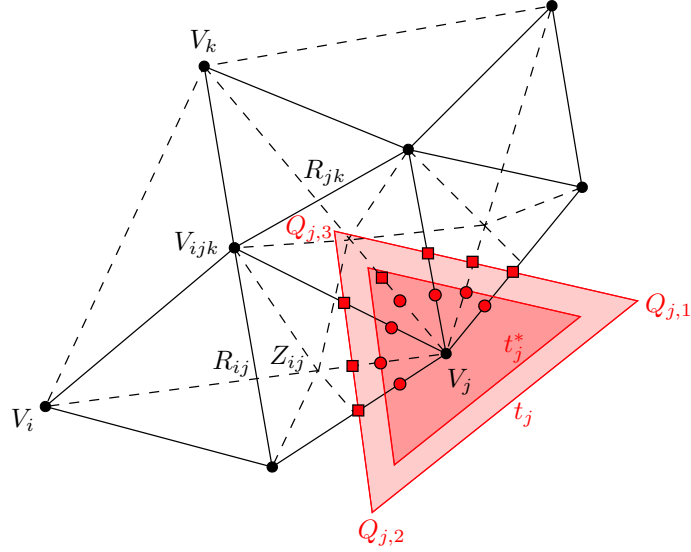


Figure 16: Rescaling the original PS-triangle t_j gives a new PS-triangle t_j^* . The PS-points of V_j in the original triangulation are indicated with squares, the ones in the refined mesh with bullets.

4 Construction of the locally refined B-spline representation

In this section we will explain how to derive the B-spline coefficients corresponding to the locally refined triangulation, of a Powell-Sabin surface when the B-spline coefficients were given on the original triangulation. Our refinement algorithm applies global $\sqrt{3}$ -subdivision on a subset of the triangulation. Hence, we can take over the subdivision rules developed by Vanraes *et al.* in [19]. Only at the boundary of the refinement domain a reformulation of some rules is needed.

4.1 The new control triangles of the vertices in the original triangulation

For the original vertices one can reuse the old PS-triangles and control triangles. This is a valid choice because any new PS-point in the refined triangulation lies closer to the considered original vertex. However, often an optimization is possible. When all triangles in the molecule of the vertex are refined, it is possible to determine a smaller PS-triangle by rescaling the original one. Denote the new PS-triangle of vertex V_j as $t_j^*(Q_{j,1}^*, Q_{j,2}^*, Q_{j,3}^*)$. To find the appropriate scaling factor, we compare the positions of the old and new PS-points. Figure 16 shows a part of the original and refined molecule of V_j together with the corresponding PS-points. Suppose each interior point inside the refined molecule is written as $Z_{ij} = \omega_{ij}V_j + (1 - \omega_{ij})R_{ij}$. The scaling factor ω is then found as the minimal value of the ω_{ij} . The new corners are given by

$$Q_{j,m}^* = \omega V_j + (1 - \omega)Q_{j,m}, \quad m = 1, 2, 3, \tag{4.1}$$

and the corresponding control points are calculated as [19]:

$$C_{j,1}^* = (\omega\alpha_{j,1} + 1 - \omega)C_{j,1} + \omega\alpha_{j,2}C_{j,2} + \omega\alpha_{j,3}C_{j,3} \tag{4.2a}$$

$$C_{j,2}^* = \omega\alpha_{j,1}C_{j,1} + (\omega\alpha_{j,2} + 1 - \omega)C_{j,2} + \omega\alpha_{j,3}C_{j,3} \tag{4.2b}$$

$$C_{j,3}^* = \omega\alpha_{j,1}C_{j,1} + \omega\alpha_{j,2}C_{j,2} + (\omega\alpha_{j,3} + 1 - \omega)C_{j,3}. \tag{4.2c}$$

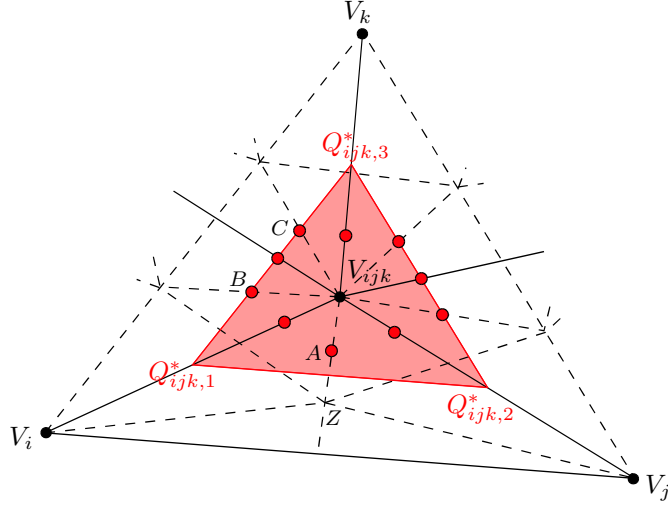


Figure 17: The PS-triangle t_{ijk}^* for vertex $V_{ijk}(= Z_{ijk})$ is found by rescaling the original triangle $\rho(V_i, V_j, V_k)$. The corresponding PS-points are marked as bullets.

4.2 The new control triangles of vertices at the original interior points

For each refined triangle $\rho(V_i, V_j, V_k)$ a new vertex V_{ijk} at the position of the interior point Z_{ijk} is added to the triangulation. Its PS-triangle $t_{ijk}^*(Q_{ijk,1}^*, Q_{ijk,2}^*, Q_{ijk,3}^*)$ can be chosen as

$$Q_{ijk,1}^* = (V_{ijk} + V_i)/2, \quad Q_{ijk,2}^* = (V_{ijk} + V_j)/2, \quad \text{and,} \quad Q_{ijk,3}^* = (V_{ijk} + V_k)/2. \quad (4.3)$$

In [19] is proven that t_{ijk}^* is a valid PS-triangle in the case where the three neighbouring triangles are refined. The PS-triangle is also valid when one or more adjacent triangles are not refined. Figure 17 illustrates the case where the triangle adjacent to side V_i - V_j is not refined. Let the interior point Z have the barycentric coordinates (a, b, c) with respect to triangle $\rho(V_i, V_j, V_{ijk})$. PS-point A can then be written as $A = aQ_{ijk,1}^* + bQ_{ijk,2}^* + cV_{ijk}$, which proves that the point lies inside the PS-triangle.

Note that the corners of PS-triangle t_{ijk}^* coincide with the PS-points \tilde{S}_i, \tilde{S}_j and \tilde{S}_k of the original triangulation, as indicated in Figure 3(a). In [19] it is shown that the corresponding Bézier ordinates w_i, w_j and w_k are valid control points. From equation (2.9d) we may then conclude that

$$C_{ijk,1}^* = \tilde{L}_{i,1}C_{i,1} + \tilde{L}_{i,2}C_{i,2} + \tilde{L}_{i,3}C_{i,3} \quad (4.4a)$$

$$C_{ijk,2}^* = \tilde{L}_{j,1}C_{j,1} + \tilde{L}_{j,2}C_{j,2} + \tilde{L}_{j,3}C_{j,3} \quad (4.4b)$$

$$C_{ijk,3}^* = \tilde{L}_{k,1}C_{k,1} + \tilde{L}_{k,2}C_{k,2} + \tilde{L}_{k,3}C_{k,3}. \quad (4.4c)$$

4.3 The new control triangles for boundary refinement

For the refinement of a boundary triangle $\rho(V_i, V_j, V_k)$ two strategies were proposed in section 3.3. They introduce either a new vertex at the boundary side, or an artificial vertex outside of the domain. The subdivision rules for both situations are now developed.

The first case is shown in Figure 18. The new vertex V_{ij} coincides with the PS-refinement point R_{ij} at the boundary edge V_i - V_j . Choosing the new interior point Z_{ij1} at the line R_{ij1} - V_{ijk} and

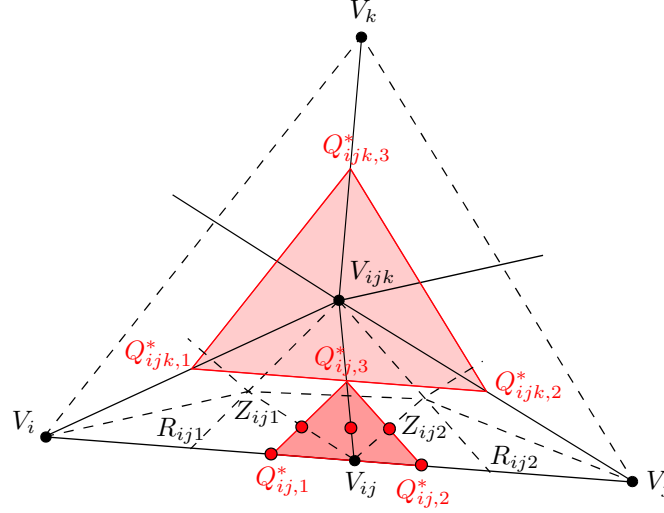


Figure 18: The PS-triangle t_{ij}^* for the boundary vertex $V_{ij}(= R_{ij})$. The corresponding PS-points are marked as bullets.

$Z_{ij,2}$ at $R_{ij,2}-V_{ijk}$, the new PS-triangle $t_{ij}^*(Q_{ij,1}^*, Q_{ij,2}^*, Q_{ij,3}^*)$ can be determined as

$$Q_{ij,1}^* = (V_{ij} + R_{ij,1})/2, \quad Q_{ij,2}^* = (V_{ij} + R_{ij,2})/2, \quad \text{and} \quad Q_{ij,3}^* = (V_{ij} + V_{ijk})/2. \quad (4.5)$$

Obviously this PS-triangle is valid. To find an expression for the new control points with respect to the old ones, we first reformulate (4.5). Corner $Q_{ij,3}^*$ can be calculated as

$$Q_{ij,3}^* = \lambda_{ij}Q_{ijk,1}^* + \lambda_{ji}Q_{ijk,2}^*, \quad (4.6)$$

with λ_{ij} and λ_{ji} defined as in (2.9e). Since point $R_{ij,1}$ can be chosen freely at the edge V_i-V_{ij} , we may write $R_{ij,1} = \nu_i V_i + (1 - \nu_i)V_{ij}$. Let $S_i = (V_i + V_{ij})/2$ and $S_j = (V_j + V_{ij})/2$, then we find

$$Q_{ij,1}^* = \nu_i S_i + (1 - \nu_i)V_{ij} = (\nu_i + \lambda_{ij} - \nu_i \lambda_{ij})S_i + \lambda_{ji}(1 - \nu_i)S_j. \quad (4.7)$$

For $Q_{ij,2}^*$ we obtain an analogous expression. We refer to Figure 3 to recall that the points S_i , S_j and $Q_{ij,3}^*$ are Bézier points. Because of the tangent property of the Bézier control net for each of the triangles in the PS-refinement, the corresponding Bézier ordinates u_i , v_j and θ_k form a plane tangent to the surface at $R_{ij} = V_{ij}$. Using equations (2.9b)-(2.9c) and (4.6)-(4.7), we find the following values for the control points

$$C_{ij,1}^* = (\nu_i + \lambda_{ij} - \nu_i \lambda_{ij})(L_{i,1}C_{i,1} + L_{i,2}C_{i,2} + L_{i,3}C_{i,3}) + \lambda_{ji}(1 - \nu_i)(L'_{j,1}C_{j,1} + L'_{j,2}C_{j,2} + L'_{j,3}C_{j,3}) \quad (4.8a)$$

$$C_{ij,2}^* = \lambda_{ij}(1 - \nu_j)(L_{i,1}C_{i,1} + L_{i,2}C_{i,2} + L_{i,3}C_{i,3}) + (\nu_j + \lambda_{ji} - \nu_j \lambda_{ji})(L'_{j,1}C_{j,1} + L'_{j,2}C_{j,2} + L'_{j,3}C_{j,3}) \quad (4.8b)$$

$$C_{ij,3}^* = \lambda_{ij}C_{ijk,1}^* + \lambda_{ji}C_{ijk,2}^*. \quad (4.8c)$$

In the second situation, a triangulation with an artificial vertex V_{ij}^a is used, as shown in Figure 19. The artificial vertex is put at the line $V_{ijk}-R_{ij}$ such that $R_{ij} = \nu V_{ij}^a + (1 - \nu)V_{ijk}$ for some value ν , e.g., $\nu = 1/2$. The corresponding PS-triangle $t_{ij}^{a*}(Q_{ij,1}^{a*}, Q_{ij,2}^{a*}, Q_{ij,3}^{a*})$ can be chosen as

$$Q_{ij,1}^{a*} = (V_{ij}^a + V_i)/2, \quad Q_{ij,2}^{a*} = (V_{ij}^a + V_j)/2, \quad \text{and} \quad Q_{ij,3}^{a*} = V_{ij}^a. \quad (4.9)$$

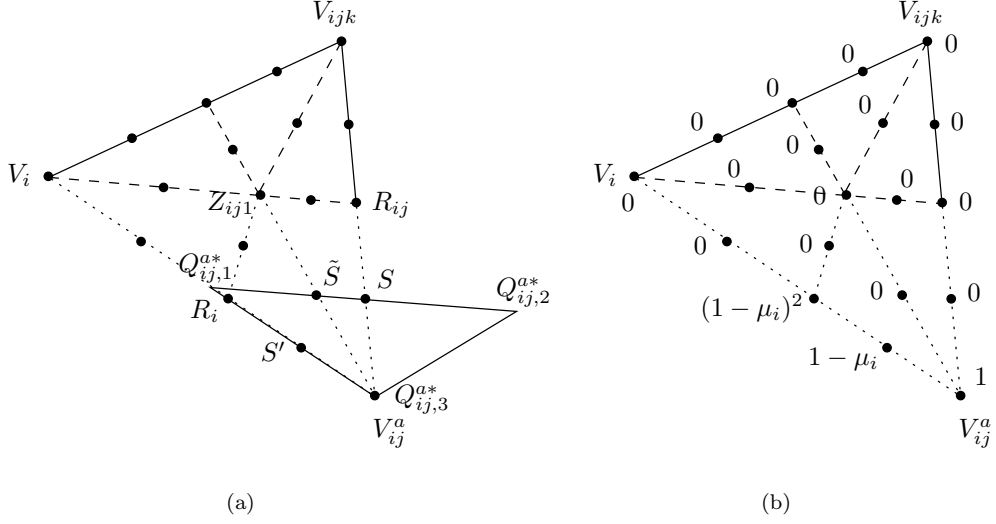


Figure 20: (a) PS-refinement of triangle $\rho(V_i, V_{ij}^a, V_{ijk})$ together with PS-triangle t_{ij}^{a*} ; (b) Bézier ordinates of basis spline $B_{ij}^{a*,3}$.

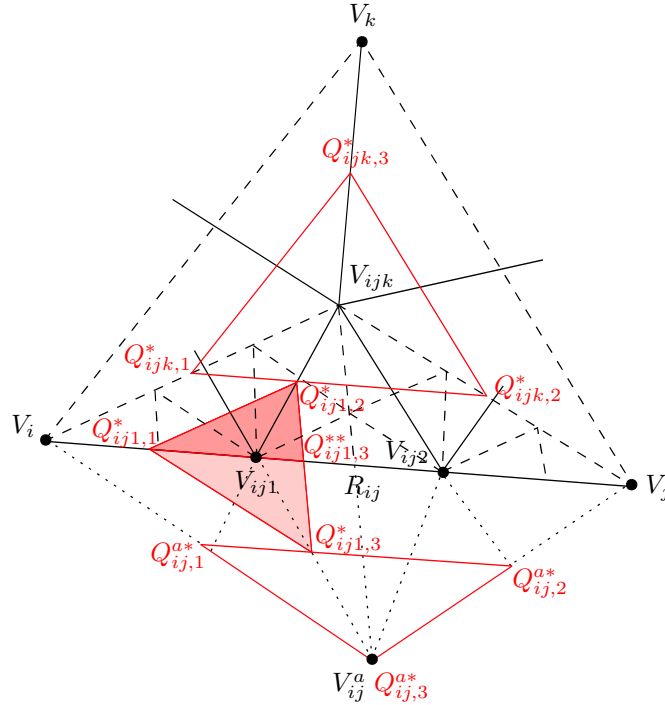


Figure 21: The PS-triangle t_{ij1}^* for the boundary vertex V_{ij1} , using the artificial PS-triangle t_{ij}^{a*} .

In a subsequent refinement level we can remove the artificial vertex V_{ij}^a . This is illustrated in Figure 21. Valid control triangles at boundary vertices such as V_{ij1} can be determined by simply applying (4.3) and (4.4). Note that a further optimization is possible for the PS-triangle $t_{ij1}^*(Q_{ij1,1}^*, Q_{ij1,2}^*, Q_{ij1,3}^*)$. The triangle can be replaced by the smaller PS-triangle $t_{ij1}^{**}(Q_{ij1,1}^*, Q_{ij1,2}^*, Q_{ij1,3}^{**})$. Using the relation $R_{ij} = \nu V_{ij}^a + (1 - \nu)V_{ijk}$, we can immediately derive the value for $C_{ij1,3}^{**}$, i.e., $C_{ij1,3}^{**} = \nu C_{ij1,3}^* + (1 - \nu)C_{ij1,2}^*$.

5 Concluding remarks

In this paper, we have proposed a subdivision algorithm for calculating the B-spline representation of a Powell-Sabin spline on a local refinement of a given triangulation. The subdivision strategy is based on the $\sqrt{3}$ -subdivision scheme. This scheme could be used to refine a local part of the domain without disturbing the neighbouring triangles. However, as has been shown, this procedure may introduce poorly shaped triangles. This problem has been dealt with by adding a refinement propagation heuristic. When a triangle fails to satisfy a certain quality requirement, an adjacent triangle is also refined. This results in an expansion of the refinement domain. If this refinement propagation reaches the boundary, an artificial vertex outside the domain is inserted into the triangulation. In the next refinement level the artificial vertex can be removed again. For this local refinement scheme, the corresponding subdivision rules have been developed. One of the advantages of our algorithm is that both global and local subdivision are possible. When the global subdivision scheme is applied twice, the triadic subdivision of [19] is exactly recovered.

The subdivision algorithm provides the user some further freedom. The splitting conditions are driven by the parameter δ , which steers the trade-off between balancing the mesh quality and localizing the refinement. A good value for this parameter is application-dependent. For CAGD one can take a low value for δ ; in finite element applications, larger values may be more appropriate. The generated new PS-triangles are not optimal in the sense of minimal area [4]. Control points associated with a minimal area PS-triangle are as close as possible to the surface, which is advantageous for local editing. Still, our control triangles are not too large and can be calculated with a minimal cost.

In many application domains local subdivision is of interest. It is, of course, an important ingredient of surface modelling. Local subdivision results in an efficient time and memory usage for representing complex surfaces. The new basis functions after subdivision have a smaller support and give the designer more local control for manipulating surfaces. The locality of the scheme ensures that the dimension of the subdivided space stays reasonable. The scheme is also useful for data fitting and finite element applications. The original spline space is a subspace of the refined one. Hence, we are guaranteed of a better approximation with an increased local resolution when a least squares data fitting approach is applied. When using an iterative solver in a finite element context, subdivision allows a trivial nested iteration strategy, where an initial fine mesh approximation is produced from a coarse mesh solution. Multiresolution techniques need subdivision too. Wavelets can be developed by means of the lifting scheme with subdivision as the prediction step and an extra update step. Geometric multigrid uses subdivision to make a prediction at the finer level. A number of applications for local subdivision scheme proposed in this paper are currently subject of further investigation.

Acknowledgement

Hendrik Speleers is funded as a Research Assistant of the Fund for Scientific Research Flanders (Belgium).

References

- [1] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Design*, 10(6):350–355, 1978.
- [2] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972.
- [3] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford, 1993.
- [4] P. Dierckx. On calculating normalized Powell-Sabin B-splines. *Comput. Aided Geom. Design*, 15(3):61–78, 1997.
- [5] P. Dierckx, S. Van Leemput, and T. Vermeire. Algorithms for surface fitting using Powell-Sabin splines. *IMA J. Numer. Anal.*, 12:271–299, 1992.
- [6] D. Doo and M.A. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Comput. Aided Design*, 10:356–360, 1978.
- [7] N. Dyn, J. Gregory, and D. Levin. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graphics*, 9(2):160–169, 1990.
- [8] G. Farin. Triangular Bernstein-Bézier patches. *Comput. Aided Geom. Design*, 3(2):83–127, 1986.
- [9] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Academic Press, Boston, 1990.
- [10] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Proceedings of Eurographics*, pages 409–420, 1996.
- [11] L. Kobbelt. $\sqrt{3}$ -Subdivision. In *Computer Graphics Proceedings, Annual Conference Series*, pages 103–112. ACM SIGGRAPH, 2000.
- [12] U. Labsik and G. Greiner. Interpolatory $\sqrt{3}$ -subdivision. In *Proceedings of the 21th European Conference on Computer Graphics*, volume 19 of *Computer Graphics Forum*, pages 131–138, Cambridge, 2000.
- [13] C.T. Loop. Smooth subdivision surfaces based on triangles. Master’s thesis, Dep. Comp. Science, University of Utah, 1987.
- [14] C. Manni and P. Sablonnière. Quadratic spline quasi-interpolants on Powell-Sabin partitions. Technical Report IRMAR 04-16, 2004.
- [15] M.J.D. Powell and M.A. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Softw.*, 3:316–325, 1977.
- [16] H. Speleers, P. Dierckx, and S. Vandewalle. Numerical solution of partial differential equations with Powell-Sabin splines. *J. Comp. Appl. Math.*, in press, 2005.
- [17] E. Vanraes, P. Dierckx, and A. Bultheel. On the choice of the PS-triangles. Technical Report 353, Dep. Comp. Science, K.U. Leuven, 2003.
- [18] E. Vanraes, J. Maes, and A. Bultheel. Powell-Sabin spline wavelets. *Int. J. Wav. Multires. Inf. Proc.*, 2(1):23–42, 2004.
- [19] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Automatic construction of control triangles for subdivided Powell-Sabin splines. *Comput. Aided Geom. Design*, 21(7):671–682, 2004.

- [20] K. Willemans and P. Dierckx. Surface fitting using convex Powell-Sabin splines. *J. Comput. Appl. Math.*, 56:263–282, 1994.
- [21] J. Windmolders. *Powell-Sabin splines for computer aided geometric design*. PhD thesis, Dep. Comp. Science, K.U. Leuven, 2003.
- [22] J. Windmolders and P. Dierckx. Subdivision of uniform Powell-Sabin splines. *Comput. Aided Geom. Design*, 16:301–315, 1999.
- [23] J. Windmolders, E. Vanraes, P. Dierckx, and A. Bultheel. Uniform Powell-Sabin spline wavelets. *J. Comp. Appl. Math.*, 154(1):125–142, 2003.